



**TÉCNICO**  
LISBOA

## **On Rotor Noise Optimization for Urban Air Mobility**

**Olívia Raquel Leite Pinto**

Thesis to obtain the Master of Science Degree in

### **Aerospace Engineering**

Supervisors: Prof. Fernando José Parracho Lau  
Prof. Frederico José Prata Rente Reis Afonso

#### **Examination Committee**

Chairperson: Prof. Filipe Szolnoky Ramos Pinto Cunha  
Supervisor: Prof. Prof. Fernando José Parracho Lau  
Member of the Committee: Prof. Edgar Caetano Fernandes

**April 2022**



This thesis is dedicated to my mother.



## **Declaration**

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.



## Acknowledgments

Firstly, I would like to thank my thesis supervisors, Professor Fernando Lau and Professor Frederico Afonso, for all the support and guidance provided throughout this thesis and for their constant availability. I would also like to thank Professor Afzal Suleman for the opportunity provided to exchange weekly presentations with himself and his partners at University of Victoria and by their valuable input.

To my parents, for always supporting me.

I truly appreciate having been able to share this process with my colleague Gabriele Bossotto with whom an exchange of meetings and knowledge happened throughout the entire process and to Bernardo Leite for his company in the computer office.

I would also like to thank Filipe Tenreiro, Francisco Santos and Laura Santos for not only being there as friends, but also for the exchange of knowledge developed throughout this thesis. I have also to thank Hugo Faustino, Inês Miguel and José Fernandes for being there for me through this master. Likewise, I have to give a special thanks to Deema Krupka, Maria Ribeiro, Miguel Domingos and Vanessa Lopes, for their support as friends.



## Resumo

A presente tese de mestrado refere-se a um estudo de otimização do ruído produzido por rotores no contexto de mobilidade aérea urbana.

A importância do presente estudo está alicerçada no crescimento global previsto para este segmento de aviação nas grandes cidades e no conseqüente investimento na diminuição do ruído produzido pelo mesmo. Tomando em consideração os rotores constituírem a parte maioritária do ruído emitido pelas aeronaves estudadas, as quais são totalmente elétricas, o design destes componentes apresenta-se como um ponto fulcral na minimização do ruído supramencionado.

Para a simulação realizada aquando do processo de otimização, um código open-source de aerodinâmica e aeroacústica (FLOWUnsteady) é integrado no módulo de otimização criado especificamente para a presente tese. As ferramentas aeroacústicas do presente código englobam a Formulação 1A de Farassat e as equações empíricas de Brooks, Pope e Marcolini, para a predição do ruído tonal e atonal, respectivamente. O código do simulador é, por conseqüente, aplicado na previsão do ruído que é posteriormente reduzido através da utilização de um algoritmo de otimização, conduzindo à geometria ótima proposta, tendo sido atingida uma redução de 16,21% em termos de Energy Averaged Overall Sound Pressure Level (EAOSPL).

**Palavras-chave:** Aeroacústica, Otimização de Ruído, Mobilidade Aérea Urbana, Rotores.



## **Abstract**

This master thesis project presents an optimization study of the noise produced by rotors in the context of Urban Air Mobility. The importance of this study is based on the expected growth of Urban Air Mobility in larger cities across the globe and on the consequent effort to decrease the noise emitted by this type of aircraft. Provided that rotors constitute a major part of the noise produced by this aviation segment and given the fact that the aircraft are intended to be fully electric, the design of these components is of paramount importance for noise minimization.

For the simulation accomplished within the optimization process, an open-source aerodynamic and aeroacoustic code (FLOWUnsteady, or FLight, Optimization, and Wind Unsteady) was integrated within an optimization module created specifically for the current thesis. These tools were based on the Formulation 1A of Farassat (PSU-WOPWOP) and the Brooks, Pope and Marcolini airfoil noise model (FLOWNoise), being utilized for the prediction of the tonal and atonal noise, respectively. The simulation code is applied to predict rotor noise which is posteriorly reduced by the means of an optimization algorithm, leading to the proposed optimal rotor design, with a reduction of 16,21% of the Energy Averaged Overall Sound Pressure Level (EAOSPL) having been achieved.

**Keywords:** Aeroacoustics, Noise Optimization, Urban Air Mobility, Rotors.



# Contents

Declaration . . . . .	v
Acknowledgments . . . . .	vii
Resumo . . . . .	ix
Abstract . . . . .	xi
List of Tables . . . . .	xv
List of Figures . . . . .	xvii
Nomenclature . . . . .	xix
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Topic Overview . . . . .	1
<b>2 Background</b>	<b>3</b>
2.1 Urban Air Mobility Noise . . . . .	3
2.2 Noise Regulation . . . . .	4
2.3 Sources of Rotor Noise . . . . .	5
2.3.1 Discrete-frequency Noise . . . . .	5
2.3.2 Broadband Noise . . . . .	6
2.4 The Ffwoocs Williams-Hawkings equation . . . . .	7
2.4.1 Derivation . . . . .	7
2.5 Formulation 1A of Farassat . . . . .	9
2.6 Numerical Algorithms for Acoustic Noise . . . . .	14
2.7 Brooks, Pope and Marcolini airfoil noise model . . . . .	15
2.7.1 Turbulent Boundary Layer - Trailing Edge . . . . .	15
2.7.2 Laminar Boundary Layer - Vortex Shedding . . . . .	19
2.7.3 Tip Vortex . . . . .	20
2.7.4 Trailing-Edge Bluntness - Vortex Shedding . . . . .	21
2.7.5 Directivity . . . . .	22
2.8 Brown and Harris model . . . . .	23
2.8.1 Rotational noise . . . . .	23
2.8.2 Vortex noise . . . . .	24

2.9	Vortex Lattice Method (VLM)	24
2.10	Blade Element Momentum Theory (BEMT)	28
2.10.1	Corrections to the BEMT model	30
2.10.2	Iteration Procedure	32
2.11	Rotor on Rotor Interactions	33
<b>3</b>	<b>Methodology</b>	<b>36</b>
3.1	FLOWUnsteady Software Description	36
3.1.1	Single-rotor Noise Example	39
3.1.2	Dynamic Code	39
3.1.3	Optimization	42
3.2	Verification and Validation	53
3.3	Code Approximations and Limitations	55
<b>4</b>	<b>Results</b>	<b>57</b>
4.1	Introduction	57
4.2	Computational	58
4.2.1	Basis Blade	58
4.2.2	Optimized Blade	60
4.2.3	Comparison between cases: n2k1, n5k1, and n6k2	63
<b>5</b>	<b>Conclusions</b>	<b>72</b>
5.1	Achievements	72
5.2	Future Work	73
	<b>Bibliography</b>	<b>73</b>
<b>A</b>	<b>Market Study of UAM aircraft</b>	<b>81</b>
<b>B</b>	<b>Dynamic Movement Code</b>	<b>82</b>
<b>C</b>	<b>Geometry CSV files</b>	<b>87</b>
C.1	Basis Rotor	87
C.2	First order two point interpolation	89
C.3	First order five point interpolation	90
C.4	Second order six point interpolation	91
<b>D</b>	<b>Acoustic Optimization Results</b>	<b>94</b>

# List of Tables

3.1	Design constraints of the optimization code. . . . .	53
4.1	Conditions for the optimization scenario. . . . .	57
4.2	Properties of the air used during the optimizations. . . . .	57
4.3	Optimization results. . . . .	60
4.4	Optimization results for the radius at the tip of the blade. . . . .	64
4.5	Optimization results: study of the impact of the number of individuals on a two-bladed rotor optimization. . . . .	69
4.6	Optimization results: study of the effect of the number of blades on the rotor optimization process. . . . .	70
A.1	Market Study of UAM aircraft, adapted from [86]. . . . .	81
C.1	Radius of the tip, radius of the hub and number of blades. . . . .	87
C.2	Airfoil geometry CSV file. . . . .	87
C.3	Blade file. . . . .	87
C.4	Spanwise CSV of the chord distribution. . . . .	88
C.5	Spanwise CSV of the height distribution. . . . .	88
C.6	Spanwise CSV of the pitch distribution. . . . .	88
C.7	Spanwise CSV of the sweep distribution. . . . .	88
C.8	Radius of the tip, radius of the hub and number of blades. . . . .	89
C.9	Blade file. . . . .	89
C.10	Spanwise CSV of the chord distribution. . . . .	89
C.11	Spanwise CSV of the height distribution. . . . .	89
C.12	Spanwise CSV of the pitch distribution. . . . .	89
C.13	Spanwise CSV of the sweep distribution. . . . .	90
C.14	Radius of the tip, radius of the hub and number of blades. . . . .	90
C.15	Blade file. . . . .	90
C.16	Spanwise CSV of the chord distribution. . . . .	90
C.17	Spanwise CSV of the height distribution. . . . .	91
C.18	Spanwise CSV of the pitch distribution. . . . .	91
C.19	Spanwise CSV of the sweep distribution. . . . .	91

C.20 Radius of the tip, radius of the hub and number of blades. . . . .	91
C.21 Blade file. . . . .	92
C.22 Spanwise CSV of the chord distribution. . . . .	92
C.23 Spanwise CSV of the height distribution. . . . .	92
C.24 Spanwise CSV of the pitch distribution. . . . .	92
C.25 Spanwise CSV of the sweep distribution. . . . .	93

# List of Figures

2.1	UAM factors of influence on noise annoyance [2]. . . . .	3
2.2	Subsonic flow conditions producing airfoil blade self-noise [19]. . . . .	6
2.3	Propagation direction of primary radiation of different noise sources [18]. . . . .	7
2.4	Definition of the moving surface implicitly as $f(x) = 0$ [21]. . . . .	8
2.5	Angles in the directivity functions [25]. . . . .	23
2.6	Definition of the $\Theta$ angle [26]. . . . .	24
2.7	Wing in vortice lattice method: coordinate system, elemental panels and horseshoe vortices [31]. . . . .	25
2.8	Nomenclature relative to the velocity that is induced by a finite-length segment [31]. . . . .	25
2.9	Geometrical scheme of a "typical" horseshoe vortex [31]. . . . .	26
2.10	Vector elements used for the computation of the induced velocities by the horseshoe in the VLM [31]. . . . .	27
2.11	Blade element [33]. . . . .	28
2.12	Blade element velocities [33]. . . . .	29
2.13	Classic $C_T$ curve and two proposed corrections for loss coefficient $F=0.9$ [37]. . . . .	32
2.14	Plots of the instantaneous vorticity contours [40]. . . . .	34
2.15	Plots of the average flow field streamline (the white line represents the rotors) [40]. . . . .	34
2.16	Multirotor configuration [39]. . . . .	35
2.17	Overall Sound Pressure Level (OASPL) directivity of the multirotor depending on separation distances [39]. . . . .	35
3.1	Flowchart of the FLOWUnsteady code for the quasi-steady solver, adapted from [51]. . . . .	38
3.2	Scheme of the Single-rotor Noise example of FLOWUnsteady. . . . .	39
3.3	Quadrotor body (B) and inertial (E) frames [56]. . . . .	41
3.4	Schematic representation of a population, a chromosome and a gene [59]. . . . .	45
3.5	Flowchart of the optimization code. . . . .	50
3.6	OASPL (dB) obtained for observers in a circular array of microphones located at a $90^\circ$ plane with the rotor's plane at 1.905 meters distance from the rotor's hub, both experimentally by Zawodny et al. [81] and by the FLOWUnsteady software with BEMT simulation [53] for a rotor performing in hover at 5400 RPM. . . . .	54

3.7	SPL (dB) obtained for an observer position at an angle of $-45^\circ$ of an $90^\circ$ plane with the rotor's plane at 1.905 meters distance from the rotor's hub, both experimentally by Zawodny et al. [81] and by the FLOWUnsteady software with BEMT simulation [53], for a rotor performing in hover at 5400 RPM. . . . .	54
3.8	SPL (dB) obtained for an observer at a $90^\circ$ plane with the rotor's plane, positioned at a 2.3 meters distance from the rotor's hub, for a rotor in hover at 4000 RPM [54]. . . . .	55
3.9	A-OASPL(dB) obtained for observers in a circular array of microphones located at a $90^\circ$ plane with the rotor's plane and at 1.905 meters distance from the rotor's hub both experimentally by Zawodny et al. [81] and by the FLOWUnsteady software with BEMT simulation [53]. . . . .	56
4.1	Geometry of the original rotor. . . . .	59
4.2	Geometry of the optimized two-bladed rotor with a $n2k1$ spline optimization. . . . .	61
4.3	Noise results of the optimized blade. . . . .	62
4.4	Distributed loads along the blade spans. . . . .	65
4.5	Geometry of the optimized two-rotor with an $n5k1$ spline optimization. . . . .	66
4.6	Geometry of the optimized two-bladed rotor with an $n6k2$ spline optimization. . . . .	66
4.7	Noise results of the optimized blades. . . . .	67
4.8	SPL Spectrum. . . . .	68
4.9	Comparison of the optimized OASPL for two-bladed rotor optimizations with a different number of individuals. . . . .	69
4.10	Geometry of a rotor with an $n2k1$ spline optimization for a 3 and a 4-blade rotor. . . . .	70
4.11	Noise results for the parametric study of the effect of the number of blades. . . . .	71
D.1	Optimization OASPL results for a 3 point spline two-bladed rotor. . . . .	94
D.2	Optimization OASPL results for a 4 point spline two-bladed rotor. . . . .	94
D.3	Optimization OASPL results for a 5 point spline two-bladed rotor. . . . .	95
D.4	Optimization OASPL results for a 6 point spline two-bladed rotor. . . . .	95

# Nomenclature

## Acronyms

A-OASPL A-weighted Overall Sound Pressure Level.

BEMT Blade Element Momentum Theory.

BET Blade Element Theory.

BPF Blade Passing Frequency.

BPM Brooks, Pope and Marcolini.

CAD Computed Aided Design.

CFD Computational Fluid Dynamics.

CNC Computed Numerical Control.

CTR Civil Tiltrotor.

DE Differential Evolution.

DLMS Direct Metal Laser Sintering.

DLP Digital Light Processing.

DOD Drop On Demand.

EA Evolutionary Algorithms.

EAOASPL Energy Averaged Overall Sound Pressure Level.

EBM Electron Beam Melting.

EP Evolutionary Programming.

ES Evolution Strategies.

eVTOL electric Vertical Take-off and Landing.

FDM Fusion Deposition Modeling.

FF2020 Flying Forward 2020.

FFF Fused Filament Fabrication.

FLOWUnsteady FLight, Optimization, and Wind Unsteady.

FW-H Ffwocs Williams-Hawkings.

GA Genetic Algorithms.

ICAO International Civil Aviation Organization.

LBL-VS Laminar Boundary Layer - Vortex Shedding.

NASA National Aeronautics and Space Administration.

OASPL Overall Sound Pressure Level.

ODM On-Demand Mobility.

RPM Rotations Per Minute.

SATS Small Aircraft Transportation System.

SEL Sound Exposure Level.

SLA Stereolithographic.

SLM Selective Laser Melting.

SLS Selective Laser Sintering.

SPIV Stereo Particle Image Velocimetry.

SPL Sound Pressure Level.

STOL Short Take-Off and Landing.

TBL Turbulent Boundary Layer.

TBL-TE Turbulent Boundary Layer - Trailing Edge.

TE Trailing Edge.

TEB-VS Turbulent Edge Bluntness - Vortex Shedding.

UA Unmanned Aircraft.

UAM Urban Air Mobility.

UAV Unmanned Aircraft Vehicle.

VLM Vortex Lattice Method.

VPM Vortex Particle Method.

VS Vortex Shedding.

VTOL Vertical Take-Off and Landing.

### Greek symbols

$\alpha$	Angle of attack.
$\beta$	Angle of side-slip.
$\delta(\cdot)$	Dirac delta function.
$\delta^*$	Boundary layer displacement thickness.
$\eta$	Lagrangian coordinates.
$\Gamma$	Vortex circulation.
$\kappa$	Thermal conductivity coefficient.
$\mu$	Molecular viscosity coefficient.
$\nabla$	Vector differential operator.
$\Omega r$	Tangential velocity.
$\Omega$	Angular velocity of rotor.
$\partial$	Partial derivative.
$\phi$	Inflow angle.
$\Phi_e, \Theta_e$	Angles used in directivity functions.
$\pi$	Pi (3.14159...).
$\psi$	Trailing edge solidity angle.
$\rho$	Fluid density.
$\sigma$	Rotor solidity.
$\square^2$	Wave or D'Alembertian operator in the three dimensional space.
$\sum$	Sum.
$\tau$	Emission time.
$\tau_B$	External torque vector in the body frame.
$\theta$	(i) Geometric angle of attack.
$\theta$	(ii) Azimuth angle.

### Roman symbols

$[I]$  Inertia matrix.

$[R]$	Rotation matrix.
$\dot{L}$	Rate of change of angular momentum.
$\dot{M}$	Rate of change of momentum.
$B(t)$	Bézier curves.
$b_{i,n}(t)$	Bernstein polynomials.
$C$	Continuity of a function.
$C_d$	2D drag coefficient.
$C_l$	2D lift coefficient.
$C_M$	3D moment coefficient.
$C_a$	Non-dimensional force component in the axial direction.
$C_r$	Non-dimensional force component in the radial direction.
$CR_i$	Crossover probability.
$D$	Dimension of each individual vector.
$d$	Distance between neighbouring rotor tips.
$F$	Loss correction factor.
$F_{hub}$	Hub loss correction factor.
$F_i$	Mutation factor.
$F_{tip}$	Tip loss correction factor.
$G$	Vehicle's state variables.
$I(., *)$	Interpolation operator.
$N_{i,k}$	Basis Cox-de Boor functions.
$NP$	Population size.
$P_i$	Bézier control points.
$R$	Rotor radius.
$r$	Radial position.
$s$	Smoothing factor.
$S(t)$	B-splines.
$D$	Rotor diameter.

$p$  Pressure.

$p'$  Acoustic pressure.

$(R_c)_0$  Reference Reynolds number based on chord length.

$\bar{D}_l, \bar{D}_h$  Low and high directivity functions.

$\vec{V}$  Velocity vector.

$a, a'$  Axial and radial induction factor.

$A, B, C$  Spectral shape functions.

$A_R, B_R$  Interpolation factors for the TBL-TE functions.

$c$  Rotor chord.

$c_0$  Speed of sound.

$F_D$  Drag Force.

$F_L$  Lift Force.

$G_1$  Peak level function for LBL-VS noise.

$G_2$   $R_c$ -dependence for LBL-VS noise peak amplitude.

$G_3$  Angle dependence for  $G_2$  function.

$G_4$  Peak level shape function for TEB-VS noise.

$G_5$  Spectral shape function for TEB-VS noise.

$h$  Blunt thickness.

$H(f)$  Heaviside function.

$J$  (i) Advance ratio.

$J$  (ii) Bessel function.

$k$  Polynomial order.

$K_1, K_2$  Amplitude functions for the TBL-TE.

$K_i$  Approximation of an integral over the panel.

$L$  (i) Lift.

$L$  (ii) Distance from the quadcopter's centre to any of its propeller's geometric centres.

$l$  Length of separation due to tip vortex.

$LE$  Leading edge.

$LE_x$	Leading edge twist.
$LE_z$	Leading edge height.
$M$	Mach number.
$m$	Harmonic number.
$M_c$	Convective mach number.
$M_r$	Mach number in the radiation direction.
$n$	Number of interpolation points.
$NB$	Number of rotor blades.
$p$	Pressure.
$p_A$	A-weighted sound pressure.
$p_{ij}$	Compressive stress tensor.
$p_{ref}$	Reference sound pressure.
$Q$	Torque.
$r$	Geometric distance.
$R_c$	Reynolds number based on chord length.
$r_e$	Effective observer distance.
$S$	Surface of discontinuity.
$St$	Strouhal number.
$St_{peak}$	Peak Strouhal number.
$T$	Thrust.
$t$	TE thickness (degree of bluntness).
$t^*$	Observer time.
$T_{ij}$	Lighthill's stress tensor.
$U$	Free-stream velocity.
$u, v, w$	Velocity Cartesian components.
$V$	Kinematic velocity.
$W$	Relative flow velocity.
$x, y, z$	Position cartesian components.

## Subscripts

$\infty$	Free-stream condition.
$\mathbb{R}^3$	Real coordinate space.
<i>avg</i>	Average.
<i>B</i>	Body frame.
<i>d</i>	Friction.
<i>e</i>	Emission.
<i>i, j, k</i>	Computational indexes.
<i>k<sub>d</sub></i>	Friction constant.
<i>L</i>	Loading.
<i>max</i>	Maximum.
<i>min</i>	Minimum.
<i>n</i>	Normal component.
<i>p</i>	Pressure side.
<i>ref</i>	Reference.
<i>ret</i>	Retarded.
<i>s</i>	Suction side.
<i>T</i>	Thickness.
<i>tip</i>	Tip of the blade.
<i>x, y, z</i>	Cartesian components.
ref	Reference condition.

## Superscripts

*	Adjoint.
kin	Kinematic.
L	Lower.
T	Transpose.
U	Upper.



# Chapter 1

## Introduction

Urban air mobility (UAM) is concerned with operational concepts and vehicles regarding the provision on demand or scheduled air transportation services for both cargo and passengers within a metropolitan area [1]. This concept proposes the creation of transportation systems in metropolitan areas that are short-ranged (point-to-point), using vertical take-off and landing (VTOL) or short take-off and landing (STOL) aircraft.

### 1.1 Motivation

By 2050 it is expected that over two thirds of the world's population lives in cities [1]. This increase on urbanization leads to a search for new means of transportation and solutions that not only sustain the increased number of passengers and cargo, but also provide geographic coverage, speed and environmental efficiency [1]. To this extent, UAM is a proposed alternative to the classic means of transportation that could solve the problems stated above, having the increasing congestion in cities as its primary market driver. It is important to underline that the proposed implementation depends on the ability to minimize the noise generated by VTOL aircraft for the UAM segment. This last topic will be further approached in chapter 2.

### 1.2 Topic Overview

"Urban Air Mobility (UAM) refers to a set of vehicles and operational concepts proposed to provide on-demand or scheduled air transportation services within a metropolitan area to overcome increasing surface congestion" [2].

The first commercial helicopter air transportation company transported passengers to and from New York City's three major airports, starting on 1953. In a decade, Los Angeles, San Francisco and Chicago had scheduled carriers as well. These companies provided connection services between major airports and nearby city centers mostly, and their operations went from under 155,000 annual passengers in 1957 to over 1.2 million in 1967. Apart from these scheduled urban air carriers (one for each city mentioned),

there were also over 100 helicopter air taxis that provided pre-booked, intra-city transportation [1].

Despite the initial success of the helicopter airlines, all of them were forced either to reduce or end services as a result of safety issues and economic challenges, namely fatal accidents that damaged the companies' reputations [1].

After the conclusion of the helicopter airline operations in the 1970s, the American government interest in VTOL applied for military applications and the airlines' interest in sub-regional air transportation were the main responsible for VTOL development [1].

During the 1980s and 1990s, Civil Tiltrotors (CTRs) were viewed as an alternative aircraft to the helicopter. They benefited from higher cruise speeds and longer range. CTRs were not able to emerge to provide passenger services, but numerous helicopter charter companies managed several scales around the world [1]. The CTR concept did not mature into an aircraft certified for commercial use, but its technologies were the basis for the tiltrotor aircraft being concerned in UAM applications.

From 2000 to 2005, the National Aeronautics and Space Administration (NASA) pursued the development of a Small Aircraft Transportation System (SATS) with the aim of using four to nine passenger propeller or jet aircraft, operating between regional, reliever, and general aviation airports or heliports. Two new classes of aircraft were enclosed within the SATS concept: very light jets and personal air vehicles. However, most commercial SATS operators were not able to reach economic viability and after the 2008 financial crisis several ceased operations; nevertheless there were diverse commuter airlines that thrived [1]: these airlines provide short distance, inter-city flights, where demand is either too small for mainline services or distances are too short for the use of large aircraft to be justifiable.

In 2010, NASA presented an idea of an "On-Demand Mobility" (ODM) system that relied upon small, electric aircraft and autonomy to conduct UAM operations in proximity to a metropolitan area [1].

In annex A, some of the world's leading UAM aircraft's characteristics are presented.

With regards to the history of UAM in Europe, Germany has three important UAM companies, namely, Lilium, Volocopter and Wingcopter. The first is the producer of Lilium jet. The second constitutes the first VTOL company to get the 'Design Organization Approval' from the European Aviation Safety Authority. The third is an award winning manufacturer of electric vertical take-off and landing (eVTOL) aircraft. From the Netherlands comes Airbus, the producer of the eVTOL Vahana and that is currently working on its second eVTOL airtaxi, the city Airbus. In Sweden there is also HeartAirspace, an air mobility startup whose first aircraft, ES-19, is certified for commercial operation by 2025 [3].

Regarding European projects with the aim of promoting UAM, one must recognize the importance of the Horizon 2020. This constitutes an European framework that was implemented from 2014 to 2020 with the aim of funding research and innovation [4]. This program promoted 27 new projects in the areas of mobility for growth, automated road transport and green vehicles, with urban drone traffic management and integration research projects heavily featured [5]. Within these were the Flying Forward 2020 (FF2020), that incorporated UAM within the geospatial infrastructure of cities [6], and CORUS-XUAM, that planned for several countries to host UAM demonstrations [7].

# Chapter 2

## Background

### 2.1 Urban Air Mobility Noise

A significant growth of UAM is expected in larger cities across the globe, with this new aviation system helping to achieve sustainability and decarbonization levels [8]. Within UAM, the section of eVTOL aircraft is projected to grow into a \$1.5 trillion industry by the year 2040 [9].

In regards of noise, when comparing small single-engine, manned fixed wing aeroplanes with small Unmanned Aircraft (UA), the second has noise levels 6 to 9 dB lower [10].

“The scalability of the current air traffic control system, the availability of aviation ground infrastructure, and acceptability of aircraft noise to the local communities have been identified as three operational constraints that may limit the implementation or growth of Urban Air Mobility (UAM) systems” [2]. Thus, there has been an increased focus on the noise emitted both by aircraft and helicopters, and on the research to predict and minimize noise emissions [2].

For the Horizon 2020 framework, there are two key criteria that should be implemented in order to achieve sustainable UAM: decrease the environmental footprint, and control both the noise and visual pollution [11], [8].

Figure 2.1 presents the influence of both acoustic and non-acoustic factors that influence a person’s annoyance to UAM noise operations, showing the influence components that act through the mechanisms to determine community acceptance of UAM noise.

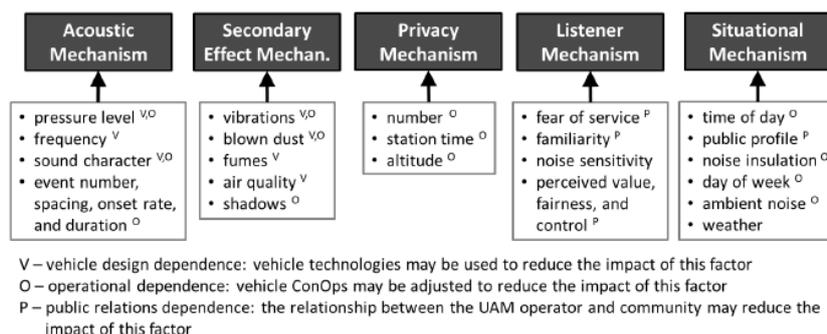


Figure 2.1: UAM factors of influence on noise annoyance [2].

The mechanisms that affect noise annoyance from UAM are, thus, the following:

1. Acoustic Mechanism - captures the audible aspects of noise and their impacts on annoyance;
2. Secondary Effect Mechanism - this mechanism provides a description of the role of non-auditory sensory impacts of aircraft operations upon annoyance;
3. Privacy Mechanism - these type of concerns may lead to action against low flying helicopters and aircraft;
4. Listener Mechanism - describes the qualities of the individual listener that influence their annoyance to aircraft noise;
5. Situational Mechanism - describes diverse qualities of the environment in which the noise is generated, propagated and received.

In respect to the noise emitted by UAM, it is perceived as being the strongest predictor of acceptance of civil drones, that is, people who are concerned about noise show the least acceptance to drones [12], [8]. This is further aggravated by the fact that the sound generated by aviation is perceived as being the most annoying [13], [8].

There are several approaches that will potentially reduce noise at the source for eVTOL of 3 to 5 dB (reduced tip speed, swept blades and increased blade count), of 1 to 2 dB, (reduced blade loading) and of 5-10 dB (optimizing the approach path) [8]. UAM introduces other noise problems, such as the possibility of acoustic interference between different rotors of the aircraft [14]. Adding to the previous contributions to total noise, quadcopters are controlled by varying the speed of its rotors individually, which leads to more content at higher frequencies, when compared to conventional aircraft [15], [8]. Therefore the high frequency noise is likely to constitute an important part of the annoyance due to the operation of small quadcopters [14].

## 2.2 Noise Regulation

The International Civil Aviation Organization (ICAO) lacks the providing of guidance to help its Member States on the creation of regulatory framework in order to regulate and oversight UAM [16]. Currently, most original equipment manufacturers contingent on general aviation and small helicopter standards for UAM regulation[16]. Thus, noise requirements for current aircraft are being applied to Unmanned Aircraft assuming that similar airframes and propulsion systems are being used [10]. It should be borne in mind that the appearance of new Unmanned Aircraft urges the elaboration of new noise and emission standards [10].

The following regulations presented are accordingly to the International Standards and Recommendations of ICAO for not exceeding 3 175 kg maximum certificated take-off mass [17]. In these, the maximum noise levels are presented in terms of sound exposure level  $L_{AE}(SEL)$ , which consists on the level, in decibels, of the time integral of squared A-weighted sound pressure ( $p_A$ ), over a certain

period of time or event, with reference to the square of the standard reference sound pressure ( $p_0$ ) - 20  $\mu Pa$ - and a reference duration of one second,

$$SEL = 10 \log \frac{1}{t_0} \int_{t_1}^{t_2} \left( \frac{p_A(t)}{p_0} \right)^2 dt, \quad (2.1)$$

where  $t_0$  is the reference integration time of one second and  $(t_2 - t_1)$  is the integration time interval. The integration time must not be less than the 10 dB-down period through which  $L_A(t)$  first rises to 10 dB(A) below its maximum value and last falls below 10 dB(A) of its maximum value.

In chapter 11.4, section 1 of volume 1 (Aircraft Noise) of annex 16 to the Convention on International Civil Aviation [17], in reference to the maximum noise level, for helicopters whose type certificate was submitted on or after 11 November 1993 or for derived versions of a helicopter for which the application for certification of the change in type design was submitted on or after 11 November 1993 (excluding those specifically designed and used for agricultural, firefighting or external load-carrying purposes), the maximum noise levels shall not exceed 82 decibels of SEL for helicopters with maximum certificated take-off mass of up to 788 kg. This value increases linearly with the logarithm of the helicopter mass at a rate of 3 decibels per doubling of mass thereafter. In section 2, it is mentioned that for all helicopters, including their derived versions, for which the application for the type certificate was submitted on or after 21 March 2002, the regulation is similar, substituting the 788 kg for 1417 kg [17].

Regarding the noise certification reference procedures, they should follow the reference atmospheric conditions:

1. constant atmospheric pressure of 1 013.25 hPa;
2. constant ambient air temperature of 25°C;
3. constant relative humidity of 70 per cent;
4. wind null.

Furthermore, with the rise of new aircraft, the creation of additional noise and emission standards urges.

## 2.3 Sources of Rotor Noise

The sources of rotor noise may be divided into two different groups: discrete frequency and broadband aerodynamic noise, each of them comprising several noise mechanisms.

### 2.3.1 Discrete-frequency Noise

The discrete frequency noise comprises the deterministic components of **thickness** (due to the displacement of the fluid by the rotor blade), **loading** (caused by the accelerating force of the fluid generated by the moving blade), **blade-vortex interaction** (resultant of the interaction of a shed tip

vortex with a following blade) and **high-speed impulsive** (associated with transonic flow around the blade) noises.

### 2.3.2 Broadband Noise

Broadband noise consists of non-deterministic loading noise sources, comprising turbulence ingestion, blade-wake interaction and blade self-noise.

Broadband noise is due to the association between the blade loading with turbulent flow on or near the blade surface [18].

Broadband self-noise is the total of noise produced when an airfoil encounters smooth nonturbulent inflow [19]. The five self-noise mechanisms, representing the interaction between the blade and the turbulence it induced, whose sound pressure levels were predicted by semi-empirical functions in the Brooks, Pope and Marcolini (BPM) method -described in section 2.7-, are subsequently presented, for subsonic flow conditions.

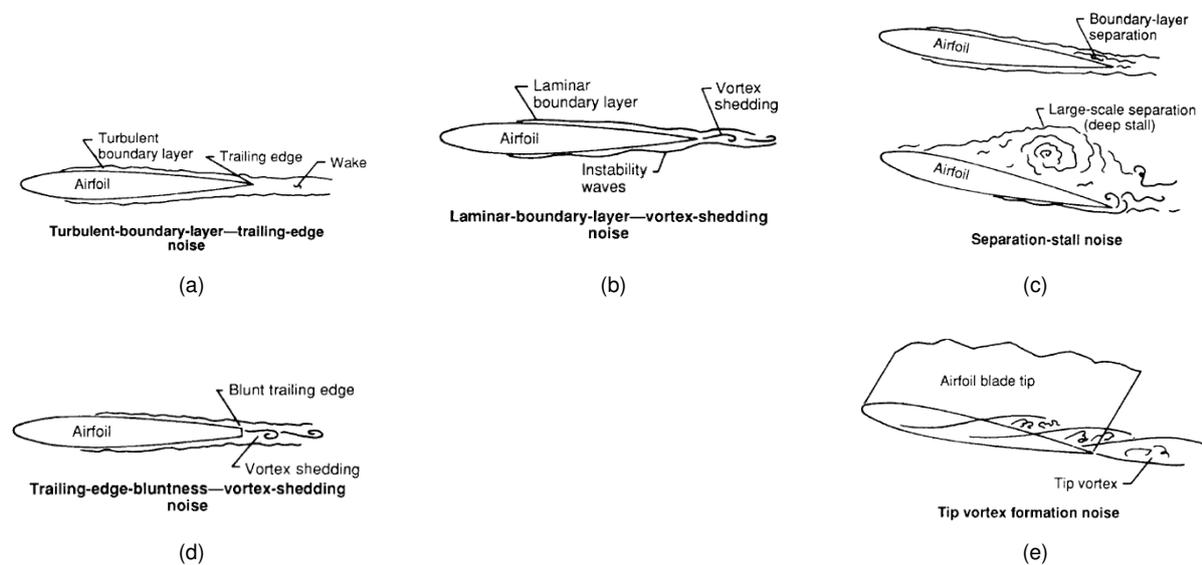


Figure 2.2: Subsonic flow conditions producing airfoil blade self-noise [19].

**Turbulent boundary layer - trailing edge (TBL-TE) noise:** At high Reynolds number ( $R_c$ ) - based on chord length -, there will be turbulent boundary layers (TBL) developed over the airfoil. The passing of this turbulence over the trailing edge (TE) will produce noise. When at stall, the noise can increase by 10dB relative to TBL-TE noise at low angles of attack [19]. A scheme of TBL-TE noise is shown in figure 2.2(a).

**Laminar boundary layer - vortex shedding (LBL-VS) noise:** At low  $R_c$ , a LBL exists over at least most of one side of an airfoil. Its instability will lead to vortex shedding (VS) and associated noise from the TE [19]. A scheme of LBL-VS noise is shown in figure 2.2(b).

**Separation stall noise:** At very high angles of attack, the flow separated near the TE originates a large-scale separation (deep stall), leading the airfoil to radiate low-frequency noise [19]. A scheme of separation stall noise is shown in figure 2.2(c).

**Trailing edge bluntness - vortex shedding noise (TEB-VS):** When the angle of attack differs from zero, the flow can separate on the suction side after a blunt TE, producing TE noise, derived from the shed turbulent vorticity. For rotor blades, the bluntness will probably be smaller than the boundary-layer thickness [19]. A scheme of trailing edge bluntness-vortex shedding noise is shown in figure 2.2(d).

**Tip vortex formation noise:** Another noise source comes from the formation of the tip vortex, containing highly turbulent flow that occurs near the tip of the lifting blades [19]. A scheme of tip vortex formation noise is shown in figure 2.2(e).

Figure 2.3 represents a scheme of the propagation direction of primary radiation of different noise sources.

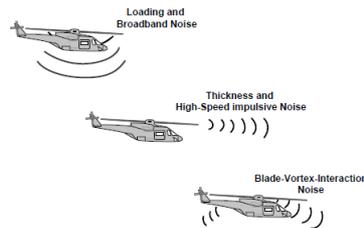


Figure 2.3: Propagation direction of primary radiation of different noise sources [18].

## 2.4 The Ffwocs Williams-Hawkings equation

Ffwocs Williams and Hawkings, on their paper "Sound Generation by Turbulence and Surfaces in Arbitrary Motion" [20], generalize the Lighthill's acoustic analogy approach, including the effects of general types of surfaces and motions. Through the mathematical theory of distributions, the authors were able to rearrange the Navier-Stokes equations into the form of an inhomogenous wave equation with a quadrupole source distribution in the volume surrounding the body and a monopole and dipole sources on the body surface. In current days, the Ffwocs Williams-Hawkings (FW-H) equation on the time-domain integral formulation is used vastly to predict deterministic rotor noise [18].

### 2.4.1 Derivation

The derivation present in this subsection is based on references [18, 20].

The aerodynamic sound theory is based on the equations of mass and momentum conservation of a compressible fluid. From these equations, one can obtain an inhomogenous wave equation, governing the propagation of sound waves outside a given closed region. In order to obtain spatial homogeneity, the following situation may be considered: an unbounded liquid that is partitioned into regions by mathematical surfaces; the last correspond to the real surface. This solution presents discontinuities at the surfaces due to the fact that usually the internal motion assumed does not match the exterior flow at the boundaries, having to be introduced mass and momentum sources that will act as sound generators. Thus, the equations used will be the conservation equations with sources, and are valid all through space, making the equation homogeneous.

Let  $f(x, t) = 0$  be an equation of form that defines a moving surface of discontinuity  $S$ , as depicted in figure 2.4, that can move in an arbitrary fashion and change its shape or orientation.  $S$  encloses the noise generating region of the flow with velocity  $v$ , that divides a volume of fluid into regions 1 and 2, in such way that  $f < 0$  in the exterior of the surface and  $f > 0$  in its interior.  $f(x, t)$  is defined such that  $\nabla f = n$  is the outward unit normal vector.

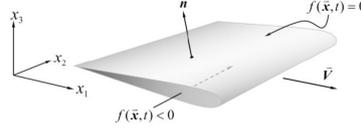


Figure 2.4: Definition of the moving surface implicitly as  $f(x) = 0$  [21].

This assumption leads to the generalized mass equation

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_i} (\bar{\rho} u_i) = [\rho(u_i) - v_i]_{(2)}^{(1)} \delta(f) \frac{\partial f}{\partial x_i}, \quad (2.2)$$

where  $[\ ]_{(2)}^{(1)}$  represents the difference of the contents between regions 2 and 1, and  $\rho$  represents the fluid's density. The generalized equation of momentum becomes

$$\frac{\partial}{\partial t} (\bar{\rho} u_i u_j) + \frac{\partial}{\partial x_j} (\bar{\rho} u_i + \bar{p}_{ij}) = [p_{ij} + \rho u_i (u_j - v_j)]_{(2)}^{(1)} \delta(f) \frac{\partial f}{\partial x_j}, \quad (2.3)$$

where  $p_{ij}$  is the compressive stress tensor.

Hence, equations 2.2 and 2.3 give the general forms of the equations governing the unbounded fluid, being valid through space.

The shear stress tensor  $p_{ij}$  has the same mean value  $p_0 \delta_{ij}$  in regions 1 and 2 and, consequently, this constant disappears from equation 2.3, and  $\bar{p}_{ij}$  can be interpreted as being the difference of the stress tensor and its mean value thus being equal to 0 in the interior of  $S$ . Another assumption made is that  $S$  is impermeable, making  $u_n = v_n$  in the exterior region, since the normal velocity of the fluid is equal to the normal velocity of the surface. Taking all of the above into consideration, the mass and momentum equations now become, respectively,

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_i} (\bar{\rho} u_i) = \rho_0 v_i \delta(f) \frac{\partial f}{\partial x_i} \quad (2.4a)$$

and

$$\frac{\partial \bar{\rho} u_i}{\partial t} + \frac{\partial}{\partial x_j} (\bar{\rho} u_i u_j + \bar{p}_{ij}) = p_{ij} \delta(f) \frac{\partial f}{\partial x_j}. \quad (2.4b)$$

Eliminating  $\bar{\rho} u_i$  from equations 2.4, with the aim of obtaining the wave equation governing the generation and propagation of sound, one obtains the following equation:

$$\left( \frac{\partial^2}{\partial t^2} - c_0^2 \frac{\partial^2}{\partial x_i^2} \right) (\bar{\rho} - \rho_0) = \frac{\partial^2 \bar{T}_{ij}}{\partial x_i \partial x_j} - \frac{\partial x}{\partial x_i} \left( p_{ij} \delta(f) \frac{\partial f}{\partial x} \right) + \frac{\partial}{\partial t} \left( \rho_0 v_n \delta(f) \frac{\partial f}{\partial x_i} \right), \quad (2.5)$$

where the dependent variable has been changed to the generalized density perturbation  $\bar{\rho} - \rho_0$  (which

constitutes a measure of the sound amplitude) and  $T_{ij}$  is a generalized function equal to Lighthill's stress tensor  $T_{ij} = \rho u_i u_j + p_{ij} - c_0^2(\rho - \rho_0)\delta_{ij}$  outside any surfaces and zero within them. Equation 2.5 is obtained by taking the derivative  $\partial/\partial t$  of equation 2.4a and the derivative  $\partial/\partial x_i$  of equation 2.4b. The obtained equations are then subtracted. Finally, the term  $\partial p/\partial x_i \partial x_j$  must be subtracted from both sides of the obtained equation.

Equation 2.5 shows that there are three possible source distributions for sound generation: a distribution of acoustic quadrupoles of strength density  $T_{ij}$  through the region that is exterior to the surfaces; surface distributions of acoustic dipoles of strength density  $p_{ij}n_j$  and also further surface distributions of sources, essentially monopoles, in the case of the existence of moving surfaces.

## 2.5 Formulation 1A of Farassat

The deduction of Formulation 1A of Farassat present on this section is based upon reference [21]. This formulation constitutes a solution to the FW-H equation with surface sources only when the surface moves at subsonic speed.

In order to simplify the derivation of Formulation 1A of Farassat, equation 2.5 is going to be presented in the following way, in the present section:

$$\square^2 p' = \frac{\partial}{\partial t}[\rho_0 v_n] \delta(f) - \frac{\partial}{\partial x_i} [p n_i \delta(f)] + \frac{\partial^2}{\partial x_i \partial x_j} [H(f) T_{ij}]. \quad (2.6)$$

In equation 2.6,  $\square^2$  is the wave or D'Alembertian operator in the three dimensional space and it's equal to  $\square^2 = (1/c_0^2)(\partial^2/\partial t^2) - \nabla^2$  and  $H(f)$  is the Heaviside function. In equation 2.6, it is assumed that  $f = 0$  encloses the moving surface in a way that the quadrupoles that produce non-negligible noise are enclosed within it and thus there is no need to integrate them outside of that same surface.

Throughout the inspection of equation 2.6, one reaches the conclusion that the objective will be solving the two following equations:

$$\square^2 p'_T = \frac{\delta}{\delta t} [\rho_0 v_n \delta(f)], \text{ Thickness Noise Equation,} \quad (2.7)$$

and

$$\square^2 p'_L = -\frac{\delta}{\delta x_i} [p n_i \delta(f)], \text{ Loading Noise Equation.} \quad (2.8)$$

Equations 2.7 and 2.8 are in the form of:

$$\square^2 p' = Q(x, t) \delta(f). \quad (2.9)$$

Therefore, in order to have the solution of the wave equation with sources on a moving surface, equation 2.9 must be solved. The solution of this equation is obtained through the use of the free-space Green's function, given by:

$$G(x, t; y, \tau) = \begin{cases} 0 & \tau > t, \\ \delta(\tau - t + r/c_0)/4\pi r & \tau < t. \end{cases} \quad (2.10)$$

In equation 2.10,  $r = |x - y|$  and  $(x, \tau)$  and  $(y, \tau)$  are the observer and the source space-time variables, respectively.  $\delta(\cdot)$  represents the Dirac delta function. The symbol  $g = \tau t + r/c$  is normally used.

Applying the free-space Green's function, given by 2.10, the following formal solution for equation 2.9 is obtained:

$$4\pi p'(x, t) = \int Q(y, t)\delta(f)\frac{\delta(g)}{r}dyd\tau. \quad (2.11)$$

The x-frame and the y-frame are fixed to the undisturbed medium. Another system of coordinates fixed relative to the surface is introduced. This new system, denominated as Lagrangian, uses the variable  $\eta$ , so the trajectory of a point fixed on the surface is given by:

$$y = y(\eta, \tau), \quad (2.12)$$

and the inverse transformation by:

$$\eta = \eta(y, \tau). \quad (2.13)$$

Equations 2.12 and 2.13 represent isometric transformations, as they only involve translations and rotations, having, therefore, Jacobians equal to unity:

$$\det\left(\frac{\partial y}{\partial \eta}\right) = 1, \quad (2.14)$$

and

$$\det\left(\frac{\partial \eta}{\partial y}\right) = 1. \quad (2.15)$$

Firstly, applying the transformation  $y \rightarrow \eta$  to equation 2.11, one obtains:

$$\begin{aligned} 4\pi p' &= \int Q(y(\eta, \tau), \tau)\delta(f)\frac{\delta(g)}{r|\det(\partial\eta/\partial y)|}d\eta d\tau \\ &= \int Q(\eta, \tau)\delta(f)\frac{\delta(g)}{r}d\eta d\tau. \end{aligned} \quad (2.16)$$

Secondly, the transformation  $\tau \rightarrow g$  is applied. It is needed to compute the Jacobian of this transformation  $\partial g/\partial \tau$ :

$$\frac{\partial g}{\partial \tau} = 1 + \frac{1}{c_0} \frac{\partial r}{\partial y_i} \frac{\partial y_i}{\partial \tau} = 1 - \frac{\hat{r}_i v_i}{c_0} = 1 - M_r, \quad (2.17)$$

where  $M_r = \hat{r}_i v_i/c_0$  is the Mach number at point  $\eta$  in the radiation direction at time  $\tau$ ,  $\hat{r}_i = (x - y)/r$  is the component of unit radiation vector and  $v_i = \partial y_i(\eta, \tau)/\partial \tau$  the component of the velocity  $v$  at point  $\eta$  with respect to the y-frame.

Through equations 2.16 and 2.17, one gets:

$$4\pi p'(x, t) = \int Q(\eta, \tau) \delta(f) \frac{\delta(g)}{r|\partial g/\partial \tau|} d\eta dg = \int_{\mathbb{R}^3} \left( \frac{Q(\eta, \tau)}{r|1 - M_r|} \delta(f) \right)_{g=0} d\eta. \quad (2.18)$$

The surface  $g = \tau - t + r/c_0 = 0$  is simply a sphere centered in  $x$ , with the surface given by the following expression:  $|x - y| = c_0(t - \tau)$ . This geometric solid is known as collapsing sphere, since, accordingly to Green's function, the wave equation is different from zero when  $\tau < t$ , thus as  $\tau$  increases from minus infinite to  $t$ , the sphere radius will shrink from infinitely large until zero.

Writing  $\tau_e = \tau(x, t; \eta)$ , and  $y_e = y(\eta, \tau_e)$ , gives:

$$r_e \equiv |x - y_e| = c_0(t - \tau_e), \quad (2.19)$$

where  $\tau_e$  is the emission time,  $y_e$  the emission position, and  $r_e$  the emission distance of the source point  $\eta$  in respect to the observer position  $x$ .

With the presented notation, equation 2.18, can be rewritten as:

$$4\pi p'(x, t) = \int_{\mathbb{R}^3} \frac{Q(\eta, \tau_e)}{r_e(1 - M_{r_e})} \delta(f) d\eta. \quad (2.20)$$

From reference [21], for an arbitrary integrable function, one obtains:

$$\int_{\mathbb{R}^3} q(y) \delta(f) dy = \int_{f=0} q(y) dS, \quad (2.21)$$

which allows to rewrite 2.20, in the following manner:

$$4\pi p' = \int_{f=0} \left[ \frac{Q(y, \tau)}{r(1 - M_r)} \right]_{ret} dS, \quad (2.22)$$

where  $ret$  stands for retarded time.

Using the results presented, formulation 1 of Farassat can be obtained.

Through the resolution of equation 2.9, presented in 2.22, the thickness noise of Formulation 1 is obtained:

$$4\pi p'_T(x, t) = \frac{\partial}{\partial t} \int_{f=0} \left[ \frac{\rho_0 v_n}{r(1 - M_r)} \right]_{ret} dS. \quad (2.23)$$

In equation 2.23, the order of the derivative was changed, based on the Leibniz rule of differentiation under an integral sign, that states:

$$\frac{\partial}{\partial t} \int Q(y, \tau) \delta(f) \frac{\delta(g)}{r} dy d\tau = \int Q(y, \tau) \delta(f) \frac{\partial}{\partial t} \left( \frac{\delta(g)}{r} \right) dy d\tau. \quad (2.24)$$

The loading noise term will now be deduced. Firstly, equation 2.8 must be written using the free-space Green's function:

$$4\pi p'_L(x, t) = -\frac{\partial}{\partial x_i} \int l_i \delta(f) \frac{\delta(g)}{r} dy d\tau = -\int l_i \delta(f) \frac{\partial}{\partial x_i} \left( \frac{\delta(g)}{r} \right) dy d\tau. \quad (2.25)$$

Using the following identity

$$\frac{\partial}{\partial x_i} \left( \frac{\delta(g)}{r} \right) = -\frac{1}{c_0} \frac{\partial}{\partial t} \left( \frac{\hat{r}_i \delta(g)}{r} \right) - \frac{\hat{r}_i \delta(g)}{r^2} \quad (2.26)$$

on equation 2.25, one obtains:

$$\begin{aligned} 4\pi p'_L(x, t) &= \frac{1}{c_0} \int l_i \delta(f) \frac{\partial}{\partial t} \left( \frac{\hat{r}_i \delta(g)}{r} \right) dy d\tau + \int l_i \hat{r}_i \delta(f) \frac{\delta(g)}{r^2} dy d\tau \\ &= \frac{1}{c_0} \frac{\partial}{\partial t} \int l_i \hat{r}_i \delta(f) \frac{\delta(g)}{r} dy d\tau + \int l_i \hat{r}_i \delta(f) \frac{\delta(g)}{r^2} dy d\tau \end{aligned} \quad (2.27)$$

Replacing the result obtained in 2.27 in 2.9, the formulation of the loading noise for formulation 1 of Farassat is obtained:

$$4\pi p'_L(x, t) = \frac{1}{c_0} \frac{\partial}{\partial t} \int \left[ \frac{l_r}{r(1 - M_r)} \right]_{ret} dS + \int \left[ \frac{l_r}{r^2(1 - M_r)} \right]_{ret} dS, \quad (2.28)$$

where  $l_r = l_i \hat{r}_i$  and  $l_i = p_{ij} n_j$ .

Formulation 1 of Farassat is obtained through the sum of equations 2.23 and 2.28.

Formulation 1A will now be derived based on reference [21]. This Formulation eliminates the time derivative present in Formulation 1, since it decreases the speed and accuracy in computational noise computations [18].

The chain rule is used for the purpose of eliminating the observer time derivative of Formulation 1:

$$\left( \frac{\partial}{\partial t} \right) = \left[ \frac{\partial \tau(x, t; \eta)}{\partial t} \frac{\partial}{\partial \tau} \right]_{\tau=\tau_e}, \quad (2.29)$$

where  $\tau(x, t; \eta)$  is the solution of

$$\tau - t + |x - y(\eta, \tau)|/c_0 = 0. \quad (2.30)$$

The partial derivation of equation 2.30 with respect to  $t$  gives:

$$\left( \frac{\partial \tau}{\partial t} \right)_{x, \eta} - 1 + \frac{1}{c_0} \left( \frac{\partial r}{\partial t} \right)_{x, \eta} = \left( \frac{\partial \tau_e}{\partial t} \right)_{x, \eta} - 1 - M_r \left( \frac{\partial \tau}{\partial t} \right)_{(x, \eta)} = 0, \quad (2.31)$$

where  $M_r$  is the Mach number of point  $\eta$  in the radiation at time  $\tau$  and the following relationships were used:

$$\left( \frac{\partial r}{\partial t} \right)_{(x, \eta)} = \left( \frac{\partial r}{\partial \tau} \right)_{(x, \eta)} \left( \frac{\partial \tau}{\partial t} \right)_{(x, \eta)}, \quad (2.32a)$$

and

$$\left( \frac{\partial r}{\partial \tau} \right)_{(x, \eta)} = \frac{\partial r}{\partial y_i} \left( \frac{\partial y_i}{\partial \tau} \right)_{(x, \eta)} = -\hat{r}_i v_i = -v_r, \quad (2.32b)$$

where  $\hat{r}_i$  is the component of unit radiation vector and  $v_r$  is the velocity of the point  $\eta$  in the radiation direction.

Using equation 2.32 on equation 2.31 gives:

$$\left(\frac{\partial \tau}{\partial t}\right)_{(x,\eta)} = \frac{1}{1 - M_r}. \quad (2.33)$$

This will result in:

$$\frac{\partial}{\partial t}[q(x, y, \tau)]_{ret} = \frac{\partial}{\partial t}[q(x, y, (\eta, \tau), \tau_e(x, t; \eta))] = \left[ \frac{1}{1 - M_r} \frac{\partial q(x, y, \tau)}{\partial \tau} \right]_{ret}. \quad (2.34)$$

Using the following algebra results:

$$\frac{\partial r}{\partial \tau} = \frac{\partial r}{\partial y_i} \frac{\partial y_i(\eta, \tau)}{\partial \tau} = -\hat{r} \cdot v = -v_r, \quad (2.35a)$$

$$\frac{\partial \hat{r}_i}{\partial \tau} = \frac{\hat{r}_i v_r - v_i}{r}, \quad (2.35b)$$

$$\frac{\partial M_r}{\partial \tau} = \frac{1}{c_0 r} (r_i \dot{v}_i + v_r^2 - v^2) = \hat{r}_i \dot{M}_i + \frac{c_0(M_r^2 - M^2)}{r}, \quad (2.35c)$$

and the time derivative present in equation 2.34 on Formulation 1 of Farassat, the thickness ( $p'_T$ ) and loading ( $p'_L$ ) noise components of Formulation 1A are obtained:

$$4\pi p'_T(x, t) = \int_{f=0} \left[ \frac{\rho_0 \dot{v}_n}{r(1 - M_r)^2} + \frac{\rho_0 v_n \hat{r}_i \dot{M}_i}{r(1 - M_r)^3} \right]_{ret} dS + \int_{f=0} \left[ \frac{\rho_0 c_0 v_n (M_r - M^2)}{r^2 (1 - M_r)^3} \right]_{ret} dS, \quad (2.36a)$$

and

$$4\pi p'_L(x, t) = \frac{1}{c_0} \int_{f=0} \left[ \frac{\dot{l}_r}{r(1 - M_r)^2} \right]_{ret} dS + \frac{1}{c} \int_{f=0} \left[ \frac{l_r (r \dot{M}_r + c_0 M_r - c_0 M^2)}{r^2 (1 - M_r)^3} \right]_{ret} dS + \int_{f=0} \left[ \frac{l_r - l_M}{r^2 (1 - M_r^2)} \right]_{ret} dS, \quad (2.36b)$$

respectively.

Formulation 1A of Farassat is given by adding these two components, that is:

$$p'(x, t) = p'_L(x, t) + p'_T(x, t). \quad (2.37)$$

Equations 2.36 were written separating near ( $1/r^2$ ) and far-field ( $1/r$ ) terms, where far-field terms are the ones at a distance from where a sound source can be considered as if it is producing a plane wave with the sound pressure and velocity in phase.

According to Bretner et al [22], the loading can be estimated by a chordwise compact loading approximation if the frequency range is low enough that the observer cannot distinguish chordwise variations, which leads to a more efficient computational code and facilitates its input, since the distribution of the aerodynamic loading along the chordwise distribution of the aerodynamic loading is not always available.

## 2.6 Numerical Algorithms for Acoustic Noise

In this section, numerical algorithms possible of being used to calculate the integrals present in Formulation 1A of Farassat are presented. The numerical algorithms are based on reference [18].

The **generic retarded-time formulation** is:

$$4\pi p'(x, t) = \int_{f=0} \left[ \frac{Q(y, \tau)}{r|1 - M_r|} \right]_{ret} dS, \quad (2.38)$$

where the integral is calculated at the surface  $f = 0$  and the integrand is evaluated at the emission (retarded) time,  $\tau_e = t - r_e/c_0$ .

The **mid-panel quadrature method** constitutes the most common method of evaluating retarded-time integrals and its expression is the following:

$$4\pi p'(x, t) \approx \sum_{i=1}^N \left[ \frac{Q(y_i, t - r_i/c_0)}{r_i|1 - M_r|_i} \right] \Delta S_i, \quad (2.39)$$

where the surface  $S$  is divided into  $N$  panels and the integrand is calculated at the center of each panel ( $y_i$ ) at the point's retarded time. Through equation 2.30, a history of  $p'$  can be developed by performing an evaluation of equation 2.39 after choosing the observer's time and position, and choosing the next observer's time. This approximation works well if the source strength variation along the panel is approximately linear and the variation of the retarded time along the panel is negligible.

The **high-accuracy quadrature method** constitutes a refinement of the mid-panel quadrature method, where more points are used for a greater accuracy. This method is represented by equation 2.40:

$$4\pi p'(x, t) \approx \sum_{i=1}^N \left( \sum_{j=1}^{n_i} \alpha_j \left[ \frac{Q(y_j, t - r_j/c_0)}{r_j|1 - M_r|_j} \right]_{ret} |J|_j \right) \Delta S_i, \quad (2.40)$$

where  $\alpha_j$  represents the quadrature weight coefficient and  $|J|_j$  the Jacobian of the transformation.

The use of a larger number of points leads to the overcoming of the limitations of the mid-panel algorithm, with the negative point of increasing the computational effort.

In the **source-time-dominant algorithm**, as opposed to the other algorithms previously presented, where the primary time was regarded as being the observer's time, the source time is regarded as being the primary time, considering the central panel point  $y_i$ . In order to find the observer's time, the root of the equation  $t - \tau - |x(t) - y_i(\tau)|/c_0 = 0$  must be found. A negative consequence is that by using a sequence of source times, the sequence of observer times will be unequally spaced, requiring an interpolation in order to be able to sum the contributions from the source panels at the same observer times.

The algorithm is given by:

$$4\pi p'(x, t^*) \approx \sum_{i=1}^N I(K_i(t), t^*), \quad (2.41)$$

where  $I(\cdot, t^*)$  is an interpolation operator and  $t^*$  is the desired observer time.

K is the approximation of the integral over the panel, defined by:

$$K_i(t) = \frac{Q(y_i, \tau)}{r_i |1 - M_r|_i} \Delta S_i. \quad (2.42)$$

This algorithm has several advantages over the time-dominant algorithms previously presented. Firstly, the fact that it is source time-dominant, in opposition to observer time-dominant, presents a clear advantage when the input aerodynamic data are obtained through Computational Fluid Dynamics (CFD), since these data are normally written in a source-time referential, thus there is no need to interpolate the data. Another advantage is that the solution process of this algorithm is apposite to parallel computation. This algorithm also requires significantly less operations for a maneuvering rotor noise prediction [23].

## 2.7 Brooks, Pope and Marcolini airfoil noise model

The BPM airfoil noise model defined semi-empirical models for prediction of noise, being initially defined for NACA 0012 [24] for angles of attack from  $0^\circ$  to  $25.2^\circ$  and Mach and Reynolds numbers of up to 0.5 and  $4.6 \times 10^6$ , respectively [19]. The aero-acoustic measures performed allowed to obtain very accurate experimental results, on account of the use of an acoustically controlled wind tunnel as a testing environment. The developed semi-empirical expressions for sound pressure noise prediction, presented in the current chapter, are only dependent on flow characteristics and boundary layer parameters. The BPM model presents two major limitations, namely: the empirical nature of the equations and the fact that the boundary layer parameters are not appropriate for the generalization of airfoils, namely for chambered airfoils.

This model encloses five types of mechanisms of noise generation for airfoils immersed in a flow:

1. Turbulent boundary layer – trailing edge;
2. Separation stall;
3. Laminar boundary layer – vortex shedding;
4. Tip vortex;
5. Trailing-edge bluntness – vortex shedding.

The following deductions for the BPM model equations are based on [19] and [24].

### 2.7.1 Turbulent Boundary Layer - Trailing Edge

When immersed in a flow, airfoils will develop a boundary layer that will be turbulent at a high Reynolds number (typically  $\sqrt{R_c} > 1000$ ). This turbulence will generate noise when it passes the trailing edge.

The Sound Pressure Level (SPL) of the Turbulent Boundary Layer - Trailing Edge (TBL-TE) noise in this model is given by the following contributions:

$$SPL_{TBL-TE} = 10 \log_{10} \left( 10^{\frac{SPL_p}{10}} + 10^{\frac{SPL_s}{10}} + 10^{\frac{SPL_\alpha}{10}} \right), \quad (2.43)$$

where the subscripts  $p$ ,  $s$  and  $\alpha$  refer to the contributions from the pressure side, suction side and angle of attack, respectively. The pressure and suction contributions are given by:

$$SPL_p = 10 \log_{10} \left( \frac{\delta^* M^5 L \bar{D}_h}{r_e^2} \right) + A \left( \frac{St_p}{St_1} \right) + (K_1 - 3) + \Delta K_1 \quad (2.44)$$

and

$$SPL_s = 10 \log_{10} \left( \frac{\delta^* M^5 l \bar{D}_h}{r_e^2} \right) + A \left( \frac{St_s}{St_1} \right) + (K_1 - 3), \quad (2.45)$$

respectively, where  $\delta^*$  is the boundary layer displacement thickness on each side of the airfoil,  $St$  the Strouhal number and  $A$ ,  $\Delta K_1$  and  $K_1$  are empirical functions based on  $St$ ,  $\bar{D}_h$  is the directivity function for TE noise (high-frequency limit),  $L$  is the span and  $l$  is the spanwise extent of tip vortex at TE.

The angle of attack contribution when  $\alpha_* > 12.5^\circ$  or  $\alpha_* > \gamma_0$  is given by:

$$SPL_\alpha = 10 \log_{10} \left( \frac{\delta^* M^5 L \bar{D}_h}{r_e^2} \right) + B \left( \frac{St_s}{St_2} \right) + K_2, \quad (2.46)$$

otherwise,  $SPL_\alpha$  is given by:

$$SPL_\alpha = 10 \log_{10} \left( \frac{\delta^* M^5 L \bar{D}_h}{r_e^2} \right) + A' \left( \frac{St_s}{St_2} \right) + K_2, \quad (2.47)$$

where  $A'$ ,  $B$  and  $K_2$  are empirical functions based on  $St$ . In the latter, since separation will dominate the noise:

$$SPL_p = -\infty \quad (2.48)$$

and

$$SPL_s = -\infty. \quad (2.49)$$

The definitions of the Strouhal numbers are:

$$St_p = \frac{f \delta_p^*}{U}, \quad (2.50)$$

$$St_s = \frac{f \delta_s^*}{U}, \quad (2.51)$$

$$St_1 = 0.02 M^{-0.6}, \quad (2.52)$$

$$\bar{St}_1 = \frac{St_1 + St_2}{2}, \quad (2.53)$$

and

$$St_2 = St_1 \times \begin{cases} 1 & (\alpha_* < 1.33^\circ) \\ 10^{0.0054(\alpha_* - 1.33)^2} & (1.33^\circ \leq \alpha_* \leq 12.5^\circ) , \\ 4.72 & (12.5^\circ < \alpha_*) \end{cases} \quad (2.54)$$

where U is the free-stream velocity.

A and B are the spectral shapes definitions, where

$$A(a) = A_{min}(a) + A_R(a_0)[A_{max}(a) - A_{min}(a)] \quad (2.55)$$

and

$$B(b) = B_{min}(b) + B_R(b_0)[B_{max}(b) - B_{min}(b)]. \quad (2.56)$$

The function A for a particular Reynolds number,  $R_c$ , is obtained by interpolating the curves  $A_{max}$  and  $A_{min}$ , corresponding to  $(R_c)_{max}$  and  $(R_c)_{min}$ , respectively. The mentioned curves are given by:

$$A_{min}(a) = \begin{cases} \sqrt{67.552 - 886.788a^2} - 8.219 & (a < 0.204) \\ -32.665a + 3.981 & (0.204 \leq a \leq 0.244) \\ -142.795a^3 + 103.656a^2 - 57.757a + 6.006 & (0.224 < a) \end{cases} \quad (2.57)$$

and

$$A_{max}(a) = \begin{cases} \sqrt{67.552 - 886.788a^2} - 8.219 & (a < 0.13) \\ -15.901a + 1.098 & (0.13 \leq a \leq 0.321) , \\ -4.669a^3 + 3.491a^2 - 16.699a + 1.149 & (0.321 < a) \end{cases} \quad (2.58)$$

where  $a$  is the absolute value of the logarithm of the ratio of Strouhal number  $St = St_p$  or  $St = St_s$ , and the peak Strouhal number,  $St_{peak} = St_1$ ,  $St_{peak} = \bar{St}_1$  or  $St_{peak} = St_2$ , that is:

$$a = |\log(St/St_{peak})| \quad (2.59)$$

The interpolation factor is determined from:

$$A_R(a_0) = \frac{-20 - A_{min}(a_0)}{A_{max}(a_0) - A_{min}(a_0)}, \quad (2.60)$$

where  $a_0$  is given by:

$$a_0(R_c) = \begin{cases} 0.57 & (R_c < 9.52 \times 10^4) \\ (-9.57 \times 10^{-13})(R_c - 8.57 \times 10^5)^2 + 1.13 & (9.52 \times 10^4 \leq R_c \leq 8.57 \times 10^5) . \\ 1.13 & (8.57 \times 10^5 < R_c) \end{cases} \quad (2.61)$$

The function B is obtained similarly to A. The functions  $B_{min}$  and  $B_{max}$ , through which B is calculated

are:

$$B_{min}(b) = \begin{cases} \sqrt{16.888 - 886.788b^2} - 4.109 & (b < 0.13) \\ -83.670b + 8.138 & (0.13 \leq b \leq 0.145) , \\ -817.8105b^3 + 35.210b^2 - 135.024b + 10.619 & (0.145 < b) \end{cases} \quad (2.62)$$

and

$$B_{max}(b) = \begin{cases} \sqrt{16.888 - 886.788b^2} - 4.109 & (b < 0.13) \\ -31.330b + 1.854 & (0.10 \leq b \leq 0.187) , \\ -80.541b^3 + 44.174b^2 - 39.381b + 2.344 & (0.187 < b) \end{cases} \quad (2.63)$$

where the Strouhal number  $b$  is given by:

$$b = |\log(St_s/St_2)|. \quad (2.64)$$

The interpolation factor  $B_R$  is given by:

$$B_R(b_0) = \frac{-20 - B_{min}(b_0)}{B_{max}(b_0) - B_{min}(b_0)}, \quad (2.65)$$

where  $b_0$  is given by:

$$b_0 = \begin{cases} 0.30 & (R_c < 9.52 \times 10^4) \\ (-4.48 \times 10^{-13})(R_c - 8.57 \times 10^5)^2 + 0.56 & (9.52 \times 10^4 \leq R_c \leq 8.57 \times 10^5) . \\ 0.56 & (8.57 \times 10^5 < R_c) \end{cases} \quad (2.66)$$

The amplitude function  $K_1$ , present in equations 2.44 and 2.45, is given by:

$$K_1 = \begin{cases} -4.31 \log(R_c) + 156.3 & (R_c < 2.47 \times 10^5) \\ -9.0 \log(R_c) + 181.6 & (2.47 \times 10^5 \leq R_c \leq 8.0 \times 10^5) . \\ 128.5 & (8.0 \times 10^5 < R_c) \end{cases} \quad (2.67)$$

The level adjustment,  $\Delta K_1$ , that appears in equation 2.44, is given by:

$$\Delta K_1 = \begin{cases} \alpha_* [1.43 \log(R_{\delta_p^*}) - 5.29] & (R_{\delta_p^*} \leq 5000) , \\ 0 & (5000 < R_{\delta_p^*}), \end{cases} \quad (2.68)$$

where  $R_{\delta_p^*}$  is the Reynolds number based on pressure-side displacement thickness. The amplitude function  $K_2$ , present in equations 2.46 and 2.47, is given as:

$$K_2 = K_1 + \begin{cases} -1000 & (\alpha_* < \gamma_0 - \gamma) \\ \sqrt{\beta^2 - (\beta/\gamma)^2(\alpha_* - \gamma_0)^2} + \beta_0 & (\gamma_0 - \gamma \leq \alpha_* \leq \gamma_0 + \gamma) , \\ -12 & (\gamma_0 + \gamma < \alpha_*) \end{cases} \quad (2.69)$$

where

$$\begin{cases} \gamma = 27.094M + 3.31 \\ \gamma_0 = 23.43M + 4.651 \\ \beta = 72.65M + 10.74 \\ \beta_0 = -34.19M - 13.82 \end{cases} \quad (2.70)$$

## 2.7.2 Laminar Boundary Layer - Vortex Shedding

This type of noise hardly occurs at the Reynolds numbers at which modern rotors commonly operate. It occurs when the boundary layer is laminar and is generated by a feedback loop between vortices being shed at the trailing edge and instability waves in the laminar boundary layer. In a one-third-octave presentation, this noise is estimated on the BPM model by the following equation:

$$SPL_{LBLE-VS} = 10 \log \left( \frac{\delta_p M^5 d \bar{D}_h}{r_c^2} \right) + G_1 \left( \frac{St'}{St'_{peak}} \right) + G_2 \left[ \frac{R_c}{(R_c)_0} \right] + G_3(\alpha_*), \quad (2.71)$$

where  $G$  are empirical functions,  $St'_{peak}$  the peak of the Strouhal number,  $R_c$  constitutes the Reynolds number at the chord  $c$  and  $(R_c)_0$  is a reference value defined as a function of angle to specify the Reynolds number dependency.

The Strouhal definitions are the following:

$$St' = \frac{f \delta_p}{U}, \quad (2.72)$$

$$St'_1 = \begin{cases} 0.18 & (R_C \leq 1.3 \times 10^5) \\ 0.001756 R_C - 0.3931 & (1.3 \times 10^5 < R_C \leq 4.0 \times 10^5), \\ 0.28 & (4.0 \times 10^5 < R_C) \end{cases} \quad (2.73)$$

and

$$St'_{peak} = St'_1 \times 10^{-0.04\alpha_*}. \quad (2.74)$$

$G_1$  defines the spectral shape:

$$G_1(e) = \begin{cases} 39.8 \log(e) - 11.2 & (e \leq 0.5974) \\ 98.409 \log(e) + 2.0 & (0.5974 < e \leq 0.854) \\ -5.076 + \sqrt{2.484 - 506.25[\log(e)]^2} & (0.8545 < e \leq 1.17) \\ -98.409 \log(e) + 2.0 & (1.17 < e \leq 1.674) \\ -39.8 \log(e) - 11.2 & (1.674 < e) \end{cases} \quad (2.75)$$

where  $e = St'/St'_{peak}$ ;  $G_2$  corresponds to the peak scaled level shape curve:

$$G_2(d) = \begin{cases} 77.852 \log(d) + 15.328 & (d \leq 0.3237) \\ 65.188 \log(d) + 9.125 & (0.3237 < d \leq 0.5689) \\ -114.052[\log(d)]^2 & (0.5689 < d \leq 1.17579) \\ -65.188 \log(d) + 9.125 & (1.17579 < d \leq 3.0889) \\ -77.852 \log(d) + 15.328 & (3.0889 < d) \end{cases}, \quad (2.76)$$

where  $d = R_c/(R_c)_0$  and the reference Reynolds number is given by:

$$(R_c)_0 = \begin{cases} 10^{0.215\alpha_* + 4.978} & (\alpha_* \leq 3.0) \\ 10^{0.120\alpha_* + 5.263} & (3.0 < \alpha_*) \end{cases}, \quad (2.77)$$

and  $G_3$  is the angle-dependent level for the shape curve:

$$G_3(\alpha_*) = 171.04 - 3.03\alpha_*. \quad (2.78)$$

### 2.7.3 Tip Vortex

The vortices generated at the tip of the blades are given in the BPM model by:

$$SPL_{tip} = 10 \log \left( \frac{M^2 M_{max}^2 l^3 \bar{D}_h}{r_e^2} \right) - 30.5 (\log_{10} St'' + 0.3)^2 + 126, \quad (2.79)$$

where  $M_{max}$  is the maximum Mach number near the blade tip, within or about the separated flow region, that depends on the angle of attack at the tip of the blade,  $\alpha_{tip}$ , at the region of the TE, given by:

$$M_{max}/M \approx (1 + 0.036\alpha_{tip}), \quad (2.80)$$

where  $l$  is the spanwise extent of the separation zone,  $U_{max}$  is the velocity corresponding to  $M_{max}$ , given by:

$$U_{max} = cM_{max}, \quad (2.81)$$

and  $St''$  is the Strouhal number based on  $l$ , given by:

$$St'' = \frac{fl}{U_{max}}. \quad (2.82)$$

BPM estimates  $l$  differently for round and squared blade tips. For a round tip, the spanwise extent of the separation due to the tip vortex at the TE is given by:

$$l/c \approx 0.008\alpha_{tip}, \quad (2.83)$$

where  $\alpha_{tip}$  is the angle of attack of the tip region in respect to the incoming flow. Equations 2.80 and 2.83 are used for the case of untwisted blades in uniform flow. When the tip loadings differ from the previous case, as occurs in some rotor and propeller blades, the equation must be corrected in the following way:

$$\alpha'_{tip} = \left[ \left( \frac{\left( \frac{\delta L'}{\delta y} \right)}{\left( \frac{\delta L'}{\delta y} \right)_{ref}} \right)_{y \rightarrow tip} \right] \alpha_{tip}, \quad (2.84)$$

where  $L'$  is the lift per unit span along the blade at position  $y$ .

$\alpha_{tip}$  is not a reliable output of standard aeroelastic models and thus the prediction of tip vortex by BPM is not very accurate.

For the case of flat tips, the spanwise extent is given by the following expression:

$$l/c = \begin{cases} 0.0230 + 0.0169\alpha'_{tip} & (0^\circ \leq \alpha'_{tip} \leq 2^\circ) \\ 0.0378 + 0.0095\alpha'_{tip} & (2^\circ < \alpha'_{tip}) \end{cases}. \quad (2.85)$$

## 2.7.4 Trailing-Edge Bluntness - Vortex Shedding

The finite height of the trailing edge generates a vortex shedding, whose sound pressure level is given by:

$$SPL_{BLUNT} = 10 \log \left( \frac{hM^{5.5}L\bar{D}_h}{r_c^2} \right) + G_4 \left( \frac{h}{\delta_{avg}^*}, \psi \right) + G_5 \left( \frac{h}{\delta_{avg}^*}, \psi, \frac{St'''_{peak}}{St'''_{peak}} \right), \quad (2.86)$$

where  $h$  is the blunt thickness of the trailing edge,  $\psi$  is the solid angle between the suction and pressure sides of the airfoil, and  $\delta_{avg}^*$  is the average displacement thickness for both sides of the airfoil.

The Strouhal definitions are the following:

$$St''' = \frac{fh}{U}, \quad (2.87)$$

and

$$St'''_{peak} = \begin{cases} \frac{0.212 - 0.0045\psi}{1 + 0.235(h/\delta_{avg}^*)^{-1} - 0.0132(h/\delta_{avg}^*)^{-2}} & (0.2 \leq h/\delta_{avg}^*) \\ 0.1(h/\delta_{avg}^*) + 0.095 - 0.00243\psi & (h/\delta_{avg}^* < 0.2) \end{cases}, \quad (2.88)$$

where  $h/\delta_{avg}^*$  is the ratio between the TE thickness (degree of bluntness),  $h$ , with the average boundary-layer displacement thickness,  $\delta_{avg}^*$ , where:

$$\delta_{avg}^* = \frac{\delta_p^* + \delta_s^*}{2}. \quad (2.89)$$

The shape functions are given by:

$$G_4(h/\delta_{avg}^*, \psi) = \begin{cases} 17.5 \log(h/\delta_{avg}^*) + 157.5 - 1.114\psi & (h/\delta_{avg}^* \leq 5) \\ 169.7 - 1.114\psi & (5 < h/\delta_{avg}^*) \end{cases}; \quad (2.90)$$

the calculation of  $G_5$  involves an interpolation for spectra between  $\psi = 0^\circ$  and  $\psi = 14^\circ$  in the following way:

$$G_5\left(\frac{h}{\delta_{avg}^*}, \psi, \frac{St'''}{St'''_{peak}}\right) = (G_5)_{\psi=0^\circ} + 0.0714\psi[(G_5)_{\psi=14^\circ} - (G_5)_{\psi=0^\circ}], \quad (2.91)$$

where

$$(G_5)_{(\psi = 14^\circ)} = \begin{cases} m\eta + k & (\eta < \eta_0) \\ 2.5\sqrt{1 - (\eta/\mu^2)} - 2.5 & (\eta_0 < \eta < 0) \\ \sqrt{1.5625 - 1194.99\eta^2} - 1.25 & (0 \leq \eta < 0.03616) \\ -155.543\eta + 4.375 & (0.03616 \leq \eta) \end{cases}, \quad (2.92)$$

$$\eta = \log(St'''/St'''_{peak}), \quad (2.93)$$

$$\mu = \begin{cases} 0.1221 & (h/\delta_{avg}^* < 0.25) \\ -0.2175(h/\delta_{avg}^*) + 0.1755 & (0.25 \leq h/\delta_{avg}^* < 0.62) \\ -0.0308(h/\delta_{avg}^*) + 0.0596 & (0.62 \leq h/\delta_{avg}^* < 1.15) \\ 0.0242 & (1.15 \leq h/\delta_{avg}^*) \end{cases}, \quad (2.94)$$

$$m = \begin{cases} 0 & (h/\delta_{avg}^* \leq 0.02) \\ 68.724(h/\delta_{avg}^*) - 1.35 & (0.02 < h/\delta_{avg}^* \leq 0.5) \\ 308.475(h/\delta_{avg}^*) - 121.23 & (0.5 < h/\delta_{avg}^* \leq 0.62) \\ 224.811(h/\delta_{avg}^*) - 69.35 & (0.62 < h/\delta_{avg}^* \leq 1.15) \\ 1583.28(h/\delta_{avg}^*) - 1631.59 & (1.15 < h/\delta_{avg}^* \leq 1.2) \\ 268.344 & (1.2 < h/\delta_{avg}^*) \end{cases}, \quad (2.95)$$

$$\eta_0 = -\sqrt{\frac{m^2\mu^4}{6.25 + m^2\mu^2}}, \quad (2.96)$$

and

$$k = 2.5\sqrt{1 - \left(\frac{\eta_0}{\mu}\right)^2} - 2.5 - m\eta_0. \quad (2.97)$$

To compute  $(G_5)_{\psi=0^\circ}$ , equation 2.92 is used, replacing the term  $(h/\delta_{avg}^*)$  by  $(h/\delta_{avg}^*)'$ , given by:

$$\left(\frac{h}{\delta_{avg}^*}\right)' = 6.724\left(\frac{h}{\delta_{avg}^*}\right)^2 - 4.019\left(\frac{h}{\delta_{avg}^*}\right) + 1.107. \quad (2.98)$$

## 2.7.5 Directivity

Directivity sound effects lead to a different total sound pressure level perception for different positions of the observer relative to the rotor plane [19]. The directivity correction factors  $\bar{D}_h$  and  $\bar{D}_l$  are based on the position of the observer relative to the noise source. The directivity function for high-frequency

noise sources (these enclose everything except for high-angle separation and turbulent inflow noise) is the following:

$$\bar{D}_h(\Theta_e, \Phi_e) \approx \frac{2\sin^2(\Theta_e/2)\sin^2\Phi_e}{(1 + M\cos\Theta_e)[1 + (M - M_c)\cos\Theta_e]^2}, \quad (2.99)$$

where  $M_c$  is the convective Mach number and  $\Theta_e$  and  $\Phi_e$  are the directivity angles, as presented in figure 2.5. For low-frequency sources, the directivity correction is given by the following expression:

$$\bar{D}_l(\Theta_e, \Phi_e) \approx \frac{\sin^2(\Theta_e)\sin^2\Phi_e}{(1 + M\cos\Theta_e)^4}. \quad (2.100)$$

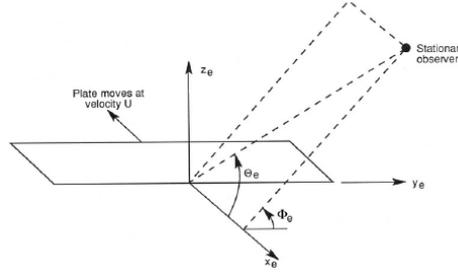


Figure 2.5: Angles in the directivity functions [25].

## 2.8 Brown and Harris model

This section follows the models adopted in [26]. In the Brown and Harris' work [27], the noise emission of a VTOL aircraft considers only the noise generated by the rotors. Rotor noise may be divided into rotational, impulsive and vortex noise. The first normally leads to a higher SPL, and, since the size of the rotors in UAM is smaller when compared to helicopters, its frequency is higher and, thus, it might be above the human threshold. Impulsive noise might happen at some flight conditions when the blade enters stall, when shock waves are formed on its surface, or when blade-vortex interactions occur.

It is important to refer that this model is not compatible with geometric programming.

### 2.8.1 Rotational noise

This type of noise might be divided into loading and thickness noise, which are modeled in Brown and Harris model using the formulation of Gutin [28] and Deming [29], respectively.

The root mean square of the loading and thickness noise pressures are given respectively by:

$$prms_L = \frac{mNB\Omega}{2\sqrt{2}\pi a\Delta S} \left[ T\cos\theta - Q\frac{a}{\Omega r_e^2} \right] J_{mNB} \left( \frac{mNB\Omega}{a} r_e \sin\theta \right), \quad (2.101)$$

and

$$prms_T = \frac{-\rho(mNB\Omega)^2 NB}{3\sqrt{2}\pi\Delta S} ct_{max} r_e J_{mNB} \left( \frac{mNB\Omega}{a} r_e \sin\theta \right), \quad (2.102)$$

where  $m$  is the harmonic number,  $NB$  the number of blades,  $\Omega$  the angular speed of the rotor,  $a$  the speed of sound,  $\Delta S$  the distance between the observer and the rotor,  $T$  the rotor thrust,  $Q$  the rotor torque,  $\theta$  is represented in figure 2.6,  $r_e$  is the effective radius,  $J_{mNB}$  is the first kind Bessel function of  $mNB$  order,  $\rho$  denotes air density,  $c$  the maximum chord and  $t_{max}$  the maximum thickness.

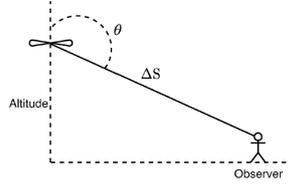


Figure 2.6: Definition of the  $\Theta$  angle [26].

The SPL for a single rotor can be obtained through:

$$SPL = 10 \log_{10} \left[ \frac{prms_L^2 + prms_T^2}{p_{ref}^2} \right], \quad (2.103)$$

where  $p_{ref}$  denotes the reference pressure. Equation 2.104 can be rewritten as:

$$SPL = 10 \log_{10} \left[ NB \frac{prms_L^2 + prms_T^2}{p_{ref}^2} \right]. \quad (2.104)$$

## 2.8.2 Vortex noise

Brown and Harris described the vortex noise as predicted in the semi-empirical model for helicopters of Schlegel et al. [30]. For this model, the SPL in hover condition is given by:

$$SPL = 20 \log_{10} \left[ k_2 \frac{V_{tip}}{\rho \Delta S} \sqrt{\frac{NRT}{\sigma} F_D F_L} \right], \quad (2.105)$$

where  $k_2$  is a set parameter of  $1.206 \times 10^{-2} s^3 / ft^3 (4.259 \times 10^{-1} s^3 / m^3)$  - this value is calibrated for helicopters.

## 2.9 Vortex Lattice Method (VLM)

The deductions present in this section are based in references [31] and [32].

In the calculation of the VLM in this chapter, a finite number of horseshoe vortices approximate the continuous bound vorticity over the wing surface, as shown in figure 2.7. It is assumed that the vortices are parallel to the axis of the vehicle. This assumption is due to the fact that it simplifies the computation of the calculation of influence of the various vortices, without any loss in accuracy when compared to the assumption that the vortices are aligned in parallel to the free stream, since both theories have a similar degree of accuracy within the linearized theory [31].

Through the law of Bio and Savart [31], one can obtain the velocity induced by a vortex filament of

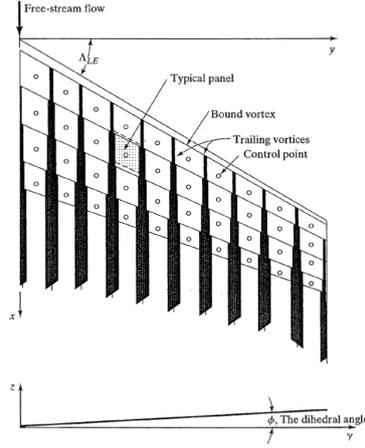


Figure 2.7: Wing in vortice lattice method: coordinate system, elemental panels and horseshoe vortices [31].

strength  $\Gamma_n$ :

$$\vec{dV} = \frac{\Gamma_n (\vec{dl} \times \vec{r})}{4\pi r^3}, \quad (2.106)$$

where  $\vec{dl}$  represents an infinitesimal vector segment along the filament and  $\vec{r}$  the vector of the mentioned segment to a point  $C$  in space.

Figure 2.8 shows the geometric nomenclature applied to the calculation of vortice induced velocity.

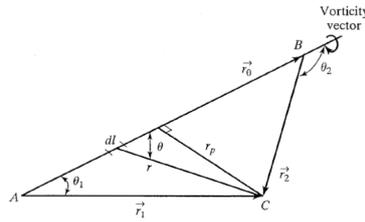


Figure 2.8: Nomenclature relative to the velocity that is induced by a finite-length segment [31].

Through the use of equation 2.106, the velocity induced by each of the three segments of the horseshoe vortex can be obtained. If  $AB$  is one of the segments, with an assumed vorticity going from  $A$  to  $B$ , and  $C$  a point whose normal distance to  $AB$  is given by  $r_p$ , integrating between  $A$  and  $B$ , the magnitude of the induced velocity can be found by:

$$V = \frac{\Gamma_n}{4\pi r_p} = \int_{\theta_1}^{\theta_2} \sin \theta d\theta = \frac{\Gamma_n}{4\pi r_p} (\cos \theta_1 - \cos \theta_2). \quad (2.107)$$

As shown in figure 2.8,  $\vec{r}_0$ ,  $\vec{r}_1$  and  $\vec{r}_2$ , represent the vectors  $\vec{AB}$ ,  $\vec{AC}$  and  $\vec{BC}$ , respectively, then:

$$r_p = \frac{|\vec{r}_1 \times \vec{r}_2|}{r_0}, \quad (2.108a)$$

$$\cos \theta_1 = \frac{\vec{r}_0 \cdot \vec{r}_1}{r_0 r_1}, \quad (2.108b)$$

and

$$\cos \theta_2 = \frac{\vec{r}_0 \cdot \vec{r}_2}{r_0 r_2}. \quad (2.108c)$$

Substituting equations 2.108 in 2.107, and having present that the direction of the unit vector is given by

$$\frac{\vec{r}_1 \times \vec{r}_2}{|\vec{r}_1 \times \vec{r}_2|}, \quad (2.109)$$

one obtains the following basic expression of the VLM induced velocity by the horseshoe vortices:

$$\vec{V} = \frac{\Gamma_n}{4\pi} \frac{\vec{r}_1 \times \vec{r}_2}{|\vec{r}_1 \times \vec{r}_2|^2} \left[ \vec{r}_0 \cdot \left( \frac{\vec{r}_1}{r_1} - \frac{\vec{r}_2}{r_2} \right) \right]. \quad (2.110)$$

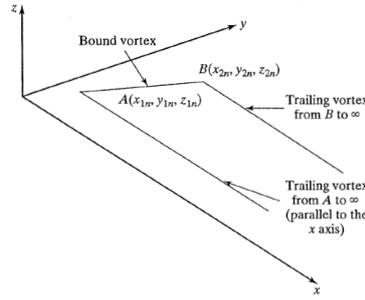


Figure 2.9: Geometrical scheme of a "typical" horseshoe vortex [31].

Through equation 2.110, the expression for the induced velocity by the vortex segment AB in a point C(x, y, z) is obtained:

$$\vec{V}_{AB} = \frac{\Gamma_n}{4\pi} \{Fac1_{AB}\} \{Fac2_{AB}\}, \quad (2.111)$$

where

$$\begin{aligned} \{Fac1_{AB}\} &= \frac{\vec{r}_1 \times \vec{r}_2}{|\vec{r}_1 \times \vec{r}_2|^2} \\ &= \{[(y - y_{1n})(z - z_{2n}) - (y - y_{2n})(z - z_{1n})]\hat{i} - [(x - x_{1n})(z - z_{2n}) - (x - x_{2n})(z - z_{1n})]\hat{j} \\ &\quad + [(x - x_{1n})(y - y_{2n}) - (x - x_{2n})(y - y_{1n})]\hat{k}\} / \\ &\quad \{[(y - y_{1n})(z - z_{2n}) - (y - y_{2n})(z - z_{1n})]^2 + [(x - x_{1n})(z - z_{2n}) - (x - x_{2n})(z - z_{1n})]^2 \\ &\quad + [(x - x_{1n})(y - y_{2n}) - (x - x_{2n})(y - y_{1n})]^2\}, \end{aligned} \quad (2.112)$$

and

$$\begin{aligned} \{Fac2_{AB}\} &= \left( \vec{r}_0 \cdot \frac{\vec{r}_1}{r_1} - \vec{r}_0 \cdot \frac{\vec{r}_2}{r_2} \right) \\ &= \{[(x_{2n} - x_{1n})(x - x_{1n}) + (y_{2n} - y_{1n})(y - y_{1n}) + (z_{2n} - z_{1n})(z - z_{1n})] / \\ &\quad \sqrt{(x - x_{1n})^2 + (y - y_{1n})^2 + (z - z_{1n})^2} \\ &\quad - [(x - x_{1n})(x - x_{2n}) + (y_{2n} - y_{1n})(y - y_{2n}) + (z_{2n} - z_{1n})(z - z_{2n})] / \\ &\quad \sqrt{(x - x_{2n})^2 + (y - y_{2n})^2 + (z - z_{2n})^2}\}. \end{aligned} \quad (2.113)$$

In order to compute the velocity from A to infinity, firstly the velocity from A to D will be calculated. According to figure 2.10:

$$\vec{r}_0 = \overrightarrow{DA} = (x_{1n} - x_{3n})\hat{i}, \quad (2.114a)$$

$$\vec{r}_1 = (x - x_{3n})\hat{i} + (y - y_{1n})\hat{j} + (z - z_{1n})\hat{k}, \quad (2.114b)$$

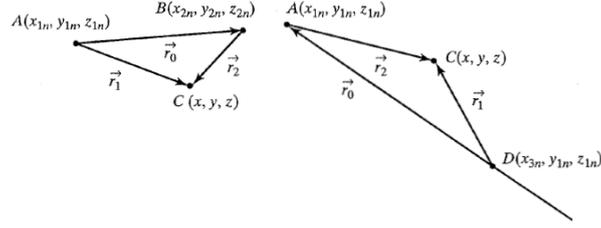


Figure 2.10: Vector elements used for the computation of the induced velocities by the horseshoe in the VLM [31].

and

$$\vec{r}_2 = (x - x_{1n})\hat{i} + (y - y_{1n})\hat{j} + (z - z_{1n})\hat{k}. \quad (2.114c)$$

Thus, the induced velocity will be given by

$$\vec{V}_{AD} = \frac{\Gamma_n}{4\pi} \{Fac1_{AD}\} \{Fac2_{AD}\}, \quad (2.115)$$

where

$$\{Fac1_{AD}\} = \frac{(z - z_{1n})\hat{j} + (y_{1n} - y)\hat{k}}{[(z - z_{1n})^2 + (y_{1n} - y)^2](x_{3n} - x_{1n})}, \quad (2.116)$$

and

$$\{Fac2_{AD}\} = (x_{3n} - x_{1n}) \left\{ \frac{x_{3n} - x}{\sqrt{(x - x_{3n})^2 + (y - y_{1n})^2 + (z - z_{1n})^2}} + \frac{x - x_{1n}}{\sqrt{(x - x_{1n})^2 + (y - y_{1n})^2 + (z - z_{1n})^2}} \right\}. \quad (2.117)$$

When considering  $x_3$  tending to infinity, equation 2.115 becomes

$$\vec{V}_{A\infty} = \frac{\Gamma_n}{4\pi} \left\{ \frac{(z - z_{1n})\hat{j} + (y_{1n} - y)\hat{k}}{[(z - z_{1n})^2 + (y_{1n} - y)^2]} \right\} \left[ 1.0 + \frac{x - x_{1n}}{\sqrt{(x - x_{1n})^2 + (y - y_{1n})^2 + (z - z_{1n})^2}} \right], \quad (2.118)$$

which constitutes the velocity induced by the vortex segment which extends through  $A$  to infinity in a positive direction parallel to the  $x$  axis.

Following the same logic, the velocity induced by the vortex filament that extends from  $B$  to infinity in a positive direction parallel to the  $x$  axis is computed by:

$$\vec{V}_{B\infty} = -\frac{\Gamma_n}{4\pi} \left\{ \frac{(z - z_{2n})\hat{j} + (y_{2n} - y)\hat{k}}{[(z - z_{2n})^2 + (y_{2n} - y)^2]} \right\} \left[ 1.0 + \frac{x - x_{2n}}{\sqrt{(x - x_{2n})^2 + (y - y_{2n})^2 + (z - z_{2n})^2}} \right]. \quad (2.119)$$

The total velocity induced by the horseshoe vortex representing one of the surface elements at a point  $(x,y,z)$  is given by the sum of the equations 2.111, 2.118 and 2.119. The velocity induced by the vortex representing the  $n^{th}$  panel on the control point with coordinates  $(x_m, y_m, z_m)$  is given by:

$$\vec{V}_{m,n} = \vec{C}_{m,n} \Gamma_n, \quad (2.120)$$

and the velocities induced by the 2N vortices added together give the expression for the total induced

velocity at the  $m^{th}$  control point:

$$\vec{V}_{m,n} = \sum_{n=1}^{2N} \vec{C}_{m,n} \Gamma_n. \quad (2.121)$$

In equation 2.121, the vortex strength,  $\Gamma_n$ , represents the lifting field of the wing. Thus, the VLM technique is used for the computation of the lift coefficient and pressure distribution of a wing when there is no significant flow separation over a finite surface. It is a faster technique, which constitutes an alternative to computational fluid dynamic [32].

## 2.10 Blade Element Momentum Theory (BEMT)

BEMT is deduced in detail in references [33] and [34]. In the present section, a brief summary of the theory is presented.

The BEMT equates the thrust and the rate of momentum change given by the blade element theory and by the momentum theory, respectively, in order to obtain the axial induction factor,  $a$ ; and it equates the torque and the rate of change of angular momentum given by the blade element theory and the momentum theory, respectively, in order to obtain the tangential induction factor,  $a'$ .

In blade element theory (BET), an element of a rotor blade at a distance  $r$  from the axis of rotation is considered, as figure 2.11 illustrates.

The relative velocity of the flow seen by the blade,  $W$ , depicted in figure 2.12, is composed by an axial and a tangential components. The tangential component is given by the sum of the tangential velocity of the blade  $\Omega r$  with the tangential velocity of the wake  $a' \Omega r$ , resulting in a total tangential velocity of  $(1 + a') \Omega r$ , where  $\Omega$  is the angular velocity at which the blades are rotating.  $U_\infty$  represents the wind velocity and  $U_\infty a$  the axial velocity of the wake. Thus, the effective velocity of the blade element is given by equation 2.122:

$$W = \sqrt{U_\infty^2 (1 - a)^2 + \Omega^2 r^2 (1 + a')^2}. \quad (2.122)$$

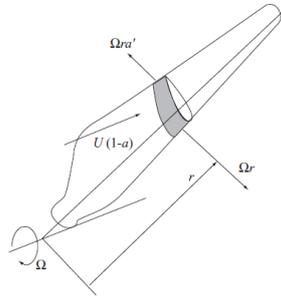


Figure 2.11: Blade element [33].

The relative velocity  $W$  acts at a  $\phi$  angle relative to the plane of rotation, given by:

$$\sin \phi = \frac{U_\infty (1 - a)}{W} \quad (2.123)$$

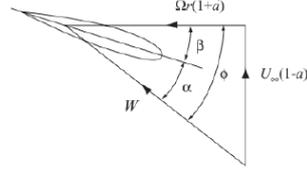


Figure 2.12: Blade element velocities [33].

and

$$\cos \phi = \frac{\Omega r(1+a')}{W}, \quad (2.124)$$

thus,

$$\tan \phi = \frac{(1-a)U_\infty}{(1+a')\Omega r}. \quad (2.125)$$

The lift force,  $F_L$ , (normal to the direction of  $W$ ) and the drag force,  $F_D$ , (parallel to the direction of  $W$ ) are given, respectively, by equations:

$$\delta F_L = \frac{1}{2}\rho W^2 c C_l \delta r, \quad (2.126)$$

and

$$\delta F_D = \frac{1}{2}\rho W^2 c C_d \delta r, \quad (2.127)$$

where  $c$  is the chord of the element,  $C_l$  the lift coefficient,  $C_d$  the drag coefficient, and  $\rho$  the air density.

In BET, the thrust of a blade element is computed by the calculation of the aerodynamic force in the axial direction:

$$\delta T = \delta F_L \cos \phi + \delta F_D \sin \phi = \frac{1}{2}\rho W^2 c (C_l \cos \phi + C_d \sin \phi) \delta r, \quad (2.128)$$

and the torque in one blade element is computed by the calculation of the aerodynamic force in the tangential direction:

$$\delta Q = (\delta F_L \sin \phi - \delta F_D \cos \phi) r = \frac{1}{2}\rho W^2 c (C_l \sin \phi - C_d \cos \phi) r \delta r. \quad (2.129)$$

In momentum theory, the rate of momentum change developed by an annular disk of the rotor is given by:

$$d\dot{M} = 4\pi\rho U_\infty^2 a(1-a)r\delta r, \quad (2.130)$$

and the rate of change of angular momentum is given by:

$$d\dot{L} = 4\pi\rho U_\infty (\Omega r) a'(1-a)r^2 \delta r. \quad (2.131)$$

Equating expressions 2.128 and 2.130, the axial induction factor is obtained:

$$a = \frac{1}{1 + \frac{4 \sin^2 \phi}{\sigma C_a}}, \quad (2.132)$$

where  $C_a$  is the non-dimensional component of the force in the axial direction, given by

$$C_a = C_l \cos \phi + C_d \sin \phi, \quad (2.133)$$

and  $\sigma$  is the solidity, obtained by performing the ratio between the total blade chord length at a given radius and the circumferential length at that radius:

$$\sigma = \frac{NBc(r)}{2\pi r}. \quad (2.134)$$

In equation 2.134,  $NB$  represents the number of rotor blades,  $r$  the radial position of the control volume and  $c(r)$  the size of the local chord at  $r$ .

Equalizing expressions 2.129 and 2.131, the tangential induction factor is obtained:

$$a' = \frac{1}{\frac{4 \sin \phi \cos \phi}{\sigma C_r} - 1}, \quad (2.135)$$

where  $C_r$  is the non-dimensional component of the force on the radial direction, given by

$$C_r = C_l \sin \phi + C_d \cos \phi. \quad (2.136)$$

In the present theory, it is assumed that there is no radial interaction between the flows through continuous annular disks. Such condition is true for axial flow induction factors that don't vary radially and the acceptability of this assumption is supported by experimental results [35].

This theory is vastly used due to its simplicity and its speed, albeit not producing results as accurate as a CFD simulation.

## 2.10.1 Corrections to the BEMT model

### Prandtl's tip-loss Factor

Prandtl's correction factor compensates the assumption of the BEMT of an infinite number of blades, correcting the vortex system in the wake, since this one is different for a finite number of blades. The mentioned factor is given by:

$$F = \frac{2}{\pi} \cos^{-1}(e^{-f}), \quad (2.137)$$

where  $f$  is

$$f = \frac{B}{2} \frac{R - r}{r \sin \phi}. \quad (2.138)$$

In the previous expression,  $R$  represents the total radius of the rotor. Through the application of this correction factor in equations 2.132 and 2.135, one obtains, respectively:

$$a = \frac{1}{1 + \frac{4F \sin^2 \phi}{\sigma C_a}}, \quad (2.139)$$

and

$$a' = \frac{1}{\frac{4F \sin^2 \phi \cos \phi}{\sigma C_r} - 1}, \quad (2.140)$$

which represent the corrected axial and tangential induction factors, respectively.

### Hub-loss model

The hub-loss model corrects the induced velocity from a vortex being shed near the rotor's hub. This model is implemented analogously as the tip-loss model, substituting equation 2.138 by the following:

$$f = \frac{B}{2} \frac{r - R_{hub}}{r \sin \phi}. \quad (2.141)$$

For an element affected by both tip and hub losses, these correction factors must be multiplied, obtaining a total loss factor of

$$F = F_{tip} F_{hub}. \quad (2.142)$$

### Glauert's Thrust Correction

The classic BEMT model does not provide a solution for cases where the induction factor is greater or equal to 0.5, as it enters in what the theory mentions as a *turbulent wake state*. Although momentum theory describes it as a propagation of some of the far wake upstream, the physically correct explanation would be that, at high tip speeds, flow from outside the wake enters this region, leading to an increase in turbulence and in the thrust in the rotor and to a decrease of the velocity of the flow in the wake.

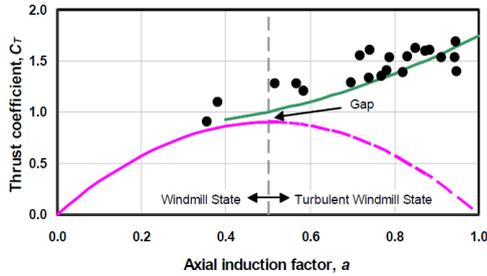
Glauert [36] fitted a parabola to experimentally obtained data from helicopters with a large induced velocity that proposed to be used as a correction for the rotor thrust coefficient, as represented in figure 2.13(a). Glauert's parabola presented some limitations since when plotting this curve for a correction factor of  $F=0.9$ , a gap between the correction proposed by Glauert's and the  $C_T$  curves occurs. This gap can lead to numerical instability whenever a computer is used for the iteration to calculate the induction factor. Buhl [37] deduced a curve, shown in figure 2.13(b), that eliminated the gap previously mentioned, allowing the use of the model for different correction factors and, thus, removing the previously existing discontinuity and its consequent dynamic instability. The correction proposed by Buhl is given by equation 2.143:

$$C_T = \frac{8}{9} + \left(4F - \frac{40}{9}\right)a + \left(\frac{50}{9} - 4F\right)a^2. \quad (2.143)$$

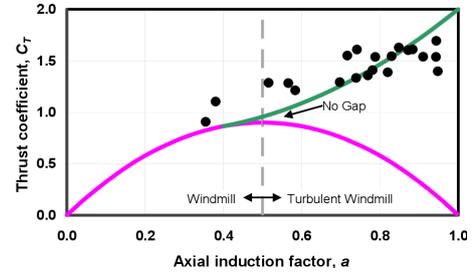
Solving equation 2.143 for the induction factor gives:

$$a = \frac{18F - 20 - 3\sqrt{C_T(50 - 36F)(+12F(3F - 4))}}{36F - 50}. \quad (2.144)$$

The presented correction should be applied for values of  $C_T$  greater or equal to  $0.96F$ , substituting the induction factor previously calculated through equation 2.139 [38]:



(a) Classic  $C_T$  curve and Glauert's correction for loss coefficient  $F=0.9$ .



(b) Classic  $C_T$  curve and Buhl's correction for loss coefficient  $F=0.9$ .

Figure 2.13: Classic  $C_T$  curve and two proposed corrections for loss coefficient  $F=0.9$  [37].

## 2.10.2 Iteration Procedure

The following algorithm is based on references [38] and [34]. It uses the equations previously presented in this section and can be applicable to each strip of the blade, as the different control volumes are assumed independent. Thus, the following procedure is applied for a certain radius position and constitutes a summary of the BEM algorithm. It is given by the subsequent steps:

1. Initialize  $a$  and  $a'$ , typically used values are  $a = a' = 0$ .
2. Compute the flow angle ( $\phi$ ) using equation 2.125.
3. Compute the local angle of attack ( $\alpha$ ) in the following manner:

$$\alpha = \phi - \theta. \quad (2.145)$$

4. Read  $C_l(\alpha)$  and  $C_d(\alpha)$  from an aerodynamic table.
5. Compute  $C_r$  and  $C_a$  through equations 2.136 and 2.133, respectively.
6. Compute the tip and the hub losses combining equation 2.137 with 2.138 and 2.141, respectively, and use them in order to obtain the total loss, given by equation 2.142.
7. Compute the thrust coefficient using the following expression:

$$C_T = \frac{\sigma(1-a)^2 C_a}{\sin^2 \phi}. \quad (2.146)$$

8. In the case of the obtained  $C_T$  being greater than  $0.96$ , compute the induction factor  $a$  using equation 2.144, otherwise, obtain the mentioned factor through equation 2.139.

9. Compute the tangential induction factor  $a'$  through the use of equation 2.140.
10. In the case where  $a$  and  $a'$  have changed more than a certain tolerance, go back to step (2), otherwise the algorithm has reached its end.

## 2.11 Rotor on Rotor Interactions

This section is based on the analysis done by Lee and Lee [39] and by Shukla et al. [40].

Shukla et al. [40] performed a study of the effect of the separation between the rotors axis and the Reynolds number of the flow on the aerodynamic interaction of multirotor Unmanned Aircraft Vehicle (UAV) in hover using high-speed Stereo Particle Image Velocimetry (SPIV) and performance measurements. Lee and Lee [39] conducted numerical simulations of a quadcopter in hover to investigate the mutual interaction effects between rotors on the aerodynamic performance, wake structures, and sound pressure level, through the application of the nonlinear VLM with vortex particle method (VPM) and acoustic analogy based on Farassat's formulation 1A.

In the study made by Shukla et al. [40], the instantaneous vorticity of one rotor interacts with the one from another rotor for small axis shift values, resulting on a deviation of the tip vortexes from their expected typical wake trajectories, getting split into small spots of high velocity between the wakes, as represented in figure 2.14. The mentioned wake interactions are relatively higher for  $R_c$  of 40k when compared to 80k. This is probably a product of the weaker tip vortexes (in terms of circulation) and lower downwash of the vortexes for low  $R_c$ , making them more vulnerable to velocities induced by flow features in close proximity. The increase of the axis shift tends to lead to a decrease of rotor interaction and the region between the wakes becomes free of sporadic flow.

An important aspect to notice from figure 2.15 is that for an axis shift of  $2.2R$ ,  $2.3R$  and  $2.4R$  -where  $R$  represents the rotor's tip radius-, the velocities, non-dimensionalized using the rotor's tip speed, are basically the same. This is probably derived from the fact that the  $R_c$  is not significant for the inflow and outflow velocities, at least for rotors with axis further then  $2.2R$  apart.

In the study by Lee and Lee [39], a comparison of wake interaction for a quadcopter vehicle with different values of distance between rotors' axes is made. The geometry of the multirotor used is represented in figure 2.16, where  $d$  represents the distance between neighbouring rotor tips and  $D$  represents the rotor's diameter. This study corroborates with the previous one in the sense that its results also demonstrate that an increase in rotor spacing leads to a decrease of wake interaction.

For a multirotor system, the unsteady loading noise due to rotor interaction is the main acoustic contributor, increasing significantly the pressure amplitude. For this particular experiment, observers were placed above and below the plane of the rotor, with an azimuth angle ( $\theta$ ) varying between  $-180^\circ$  and  $180^\circ$ , at a radial position distanced  $16R$  from the center of the vehicle. The unsteady aerodynamic loading introduced by severe rotor interaction causes an increase in the sound pressure level, especially for observers placed at  $\theta = 45^\circ$  and  $\theta = 90^\circ$ . In contrast, for  $\theta = 0^\circ$  the noise amplitude is the least affected by the interaction between the rotors, since for the perpendicular direction of the rotating axis, the thickness noise is the main contributor. These facts are corroborated by the numerical simulations

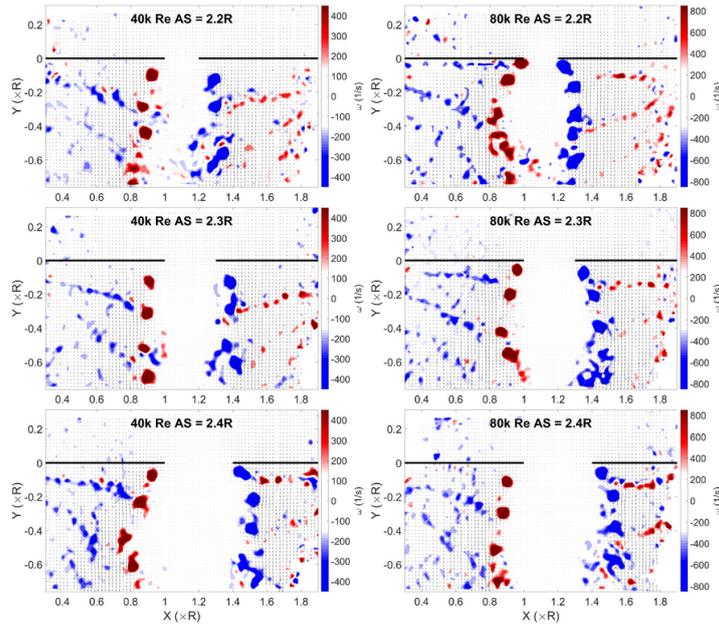


Figure 2.14: Plots of the instantaneous vorticity contours [40].

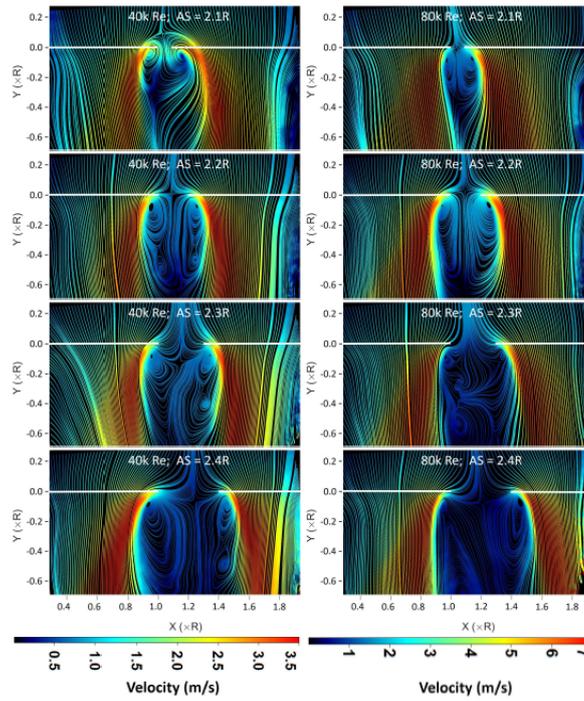


Figure 2.15: Plots of the average flow field streamline (the white line represents the rotors) [40].

presented in figure 2.17, where, for a separation distance between rotors of  $d/D=0.2$ , the noise is increased the most for  $\theta = 90^\circ$  and the least for  $\theta = 0^\circ$ , when compared to a separation distance of  $d/D=1$ .

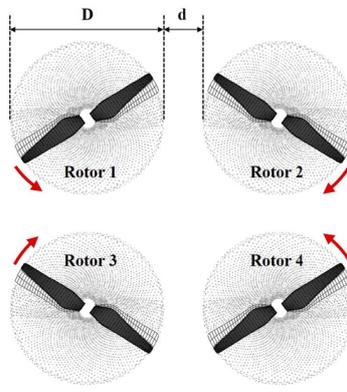


Figure 2.16: Multirotor configuration [39].

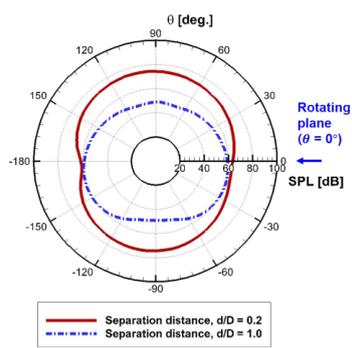


Figure 2.17: Overall Sound Pressure Level (OASPL) directivity of the multirotor depending on separation distances [39].

# Chapter 3

## Methodology

### 3.1 FLOWUnsteady Software Description

This section will provide an insight of FLOWUnsteady [41], an open source code that comprises “a simulation engine of mixed-fidelity unsteady aerodynamics and aeroacoustics” [42] that was used as a base code for the noise prediction presented in the current thesis. The code is implemented on the *julia* programming language, which is considered by its developers as “a flexible dynamic language, appropriate for scientific and numerical computing, with performance comparable to traditional statically-typed languages” [43].

FLOWUnsteady includes several github packages as its dependencies, using its results as inputs for its simulations. Regarding noise predictions, it integrates a binary of the PSU-WOPWOP code [44]. This code consists on a numerical implementation of Formulation 1A of FW-H for the computation of the tonal aeroacoustic noise [45]. The signal is given in time-domain, which is posteriorly provided in frequency domain through a Fourier Transformation. This signal is integrated in the frequency domain, in order to obtain a single OASPL value at each observer. For the computation of the broadband noise [46], it uses a BPM [47] code. This last package is integrated in the aeroacoustic solver and consists on a set of functions that will implement the BPM semi-empirical equations in order to predict the acoustic propagation of a wind turbine, with the proviso that turbulent inflow is not assumed in the code. In the case of FLOWUnsteady, the BPM code is also used applied to rotors. The theory behind Formulation 1A of FW-H and BPM is presented in sections 2.5 and 2.7, respectively.

On its dependencies FLOWUnsteady also includes FLOWVLM [48] and CCBlade [49, 50]. FLOWVLM solves the vortex lattice on lifting surfaces, calculating the circulation for each horseshoe of the lifting surface through the implementation of a VLM code. CCBlade, in its turn, implements the BEM method for the aerodynamic analysis of propellers and turbines. The theory behind VLM and the BEM method is described in sections 2.9 and 2.10, respectively.

The FLOWUnsteady code may also provide a VPM package. This package is not open-source and is used for the computation of the vorticity from lifting surfaces and bodies in the case of unsteady dynamics, solving the vorticity form of the Navier-Stokes equations in a meshless Lagrangian scheme. It was not

possible to access this package during the execution of the present thesis. Thus, FLOWUnsteady was implemented without VPM and the assumption of quasi-steady dynamics was made. This presupposition is plausible since the observers used for the computation of the noise were not far from the rotors (at a distance of approximately 2 meters from its hub) and the basis study case consists on the optimization of the noise produced by a single rotor. Taking the close position of the observers in regard to the hub of the rotor into consideration, and knowing that quasi-steady aerodynamic solvers will assume that the perturbation at the wake-shedding surface will immediately affect all the wake, the assumption of a quasi-steady physical model is plausible in this experiment. In a quasi-steady mode, FLOWUnsteady uses VLM and BEMT to solve the wing and the rotor aerodynamics, respectively. Regarding the rotors' aerodynamics, the code does not attempt to compute the wake interactions and, consequently, it is not advisable for the simulation of rotors operating in close proximity [51]. In the ambit of the present thesis that does not pose an issue, since the main objective will be to decrease the noise produced by a single rotor in hover. Taking further into consideration that the main purpose of this thesis is the optimization of the noise produced by the blades of the rotor, it is also worth mentioning that the use of a low to medium fidelity aerodynamic model will decrease the computational cost of the optimization, making this quasi-steady model a good choice to execute the reduction of the design space, until the reach of the optimal design. Furthermore, the unsteady FLOWUnsteady code (using the VPM package) takes up to 6 hours in an average laptop in order to serve a typical maneuver [51], which makes it not advisable in optimization programming.

The flowchart of this quasi-steady solver is present in figure 3.1 [9, 51]. The following explanation of the mentioned flowchart refers to its application to the simulations done in the current thesis. In the flowchart, it is observable that the outer layer of computation consists on the aerodynamic solver (represented by the blue block on the chart). In step (1) of the aerodynamic block, the difference in the kinematic translational and rotational velocities ( $\Delta V$  and  $\Delta \Omega$ ) is computed, updating  $V$  and  $\Omega$ , and the vehicle is translated accordingly - the letter  $G$  in this step represents the vehicle's state variables. In step (2), the rotors are rotated accordingly to the rotations per minute (RPM) inputted and the kinematic velocity ( $u^{kin}$ ) of each control point ( $CP$ ) is computed. Relatively to step (3), since a quasi-steady solver was used, the perturbations at the source of a wake-shedding surface happen simultaneously in all wake. Step (4) contains the implementation of the quasi-steady solver. In step (5), the wake is updated accordingly to the change in circulation computed in step (4), whilst in step (6) the vehicle is passed to a user defined function where it can be connected to another solver. This function returns a break flag. Finally, step (7) consists on the saving of the output files for further post-processing. The described scheme is embedded in a cycle that runs until either a break is returned in step (6) or the solving time reaches its maximum allowed. Would the vehicle also comprise a wing and the aerodynamic solution would also resort to FLOWVLM for the computation of its results. In the case of the availability of the VPM code, the aerodynamic solution would be embedded in the latest, and the reasoning previously presented would maintain itself, now being called within the VPM solver.

As the FLOWUnsteady software at the date of the performing of the current thesis only provided a simulation based on kinematic movement, a dynamic code was implemented, as further described in

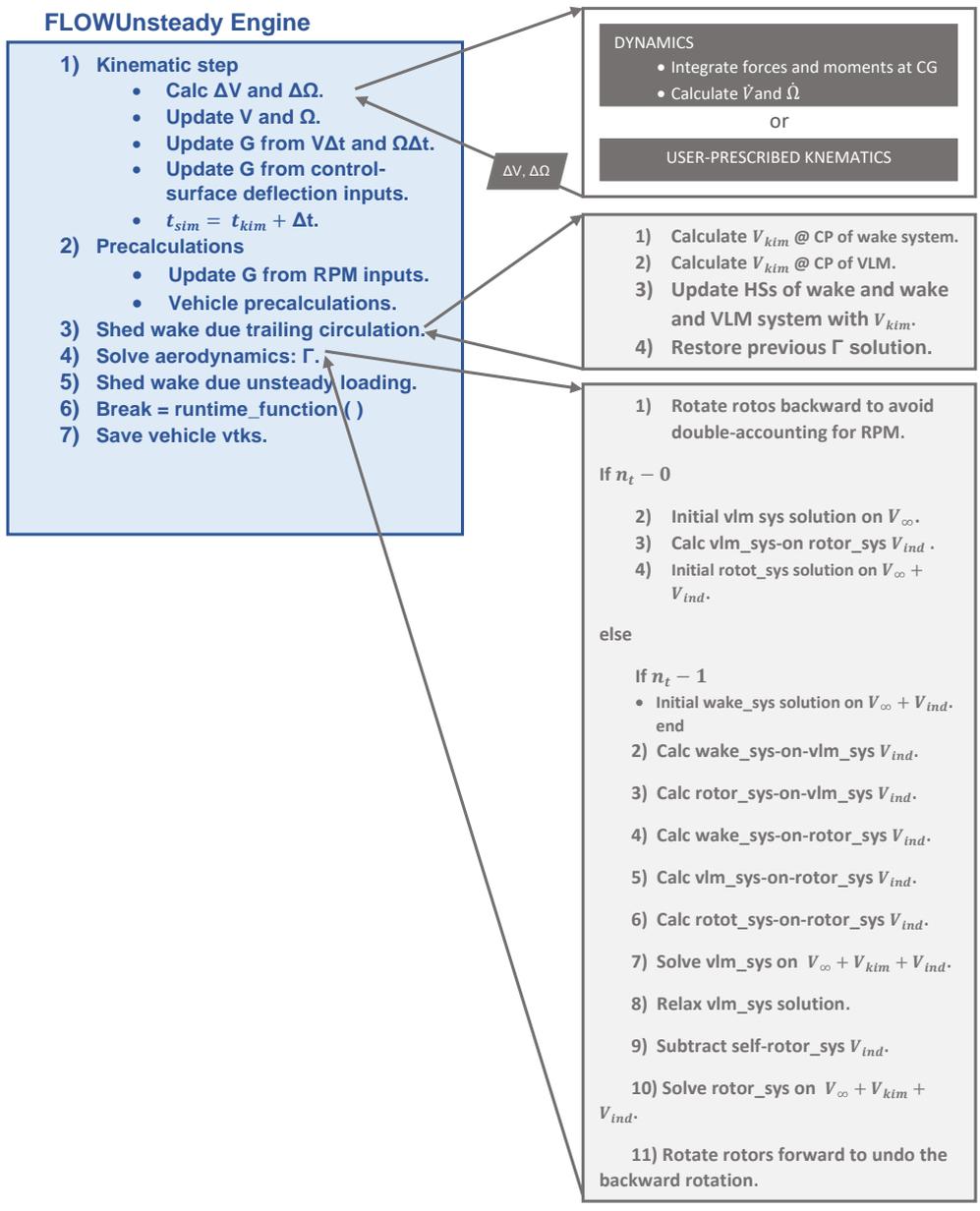


Figure 3.1: Flowchart of the FLOWUnsteady code for the quasi-steady solver, adapted from [51].

section 3.1.2.

### 3.1.1 Single-rotor Noise Example

The FLOWUnsteady github [52] comprises an example of the prediction of the noise produced by rotor DJI9443 in hover. This example was modified for the purpose of the study of the optimization of the rotor’s geometric parameters in order to decrease the noise produced. The optimization code created was integrated within a modified FLOWUnsteady code (in order to make both codes compatible), as described in section 3.1.3.

A succinct scheme of the example code presented by FLOWUnsteady [53] and the underlying theories is presented in figure 3.2. The code starts by calculating the aerodynamic solution, based on the BEMT. With the aerodynamic results, the code resorts to PSU-WOPWOP and to BPM to compute the tonal and the broadband noises, respectively. As a third and final step, the data is processed providing the following results: pressure waveform, SPL Spectrum, A-weighted SPL spectrum, tonal directivity for the first Blade Passing Frequency (BPF), tonal directivity for the second BPF, overall SPL directivity and A-weighted overall SPL directivity.

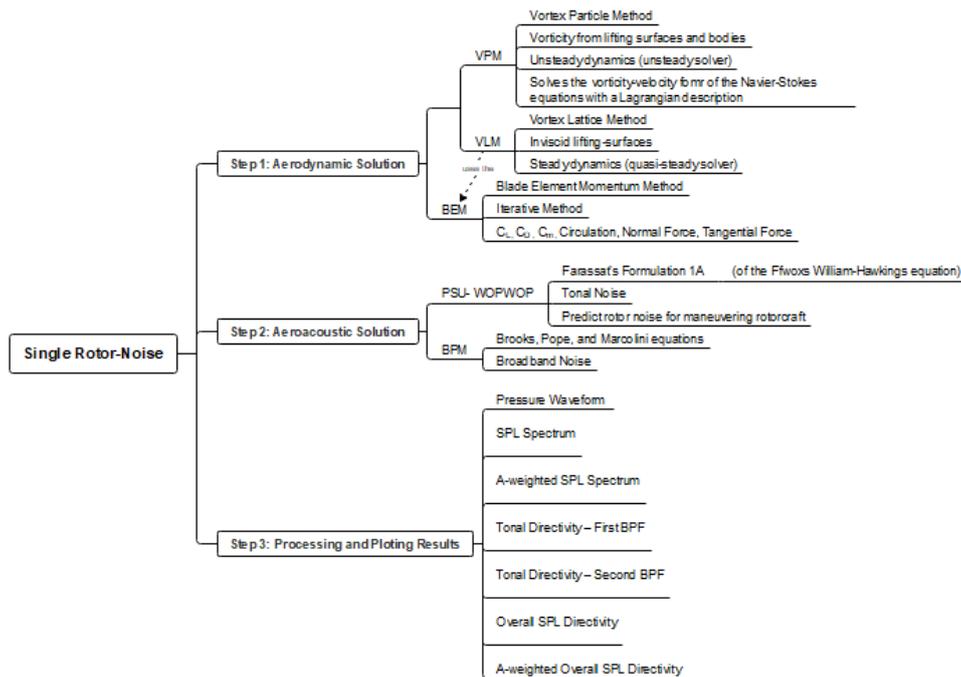


Figure 3.2: Scheme of the Single-rotor Noise example of FLOWUnsteady.

### 3.1.2 Dynamic Code

Upon the date of the making of the current thesis, the FLOWUnsteady software only had implemented a kinematic movement simulation, with no dynamic movement. A *julia* module for the implementation of the dynamics for a quadcopter was created and connected to the FLOWUnsteady code in a partnership with Gabriele Bossotto [54]. The referred module is present in annex B.

The current section serves as a description of the created module and its connection with FLOWUnsteady. For a more profound knowledge of the physics behind the dynamics of a quadcopter, the following bibliographic references are advised [55, 56].

The code revolves about a function named *equations* that is called upon the simulation cycle of FLOWUnsteady in functions *calc\_dV* and *calc\_dΩ*, where the present values of the linear velocity vector  $\dot{x}$ , of the angular velocity vector  $\Omega$  and of the vector of the roll, pitch and yaw angles in the body frame  $\theta = (\phi, \theta, \psi)^T$  are saved within a created mutable struct of type *AbstractManeuver* (a type implemented in the FLOWUnsteady package) as the values in the previous time step, and the values of the mentioned variables in the present time step are computed by function *equations*, being also saved in the same struct. In this manner, function *calc\_dV* will return the change in velocity  $dV = [dV_x, dV_y, dV_z]$  and *calc\_dΩ* will return the change in angular velocity  $d\Omega = [d\Omega_x, d\Omega_y, d\Omega_z]$  at time  $t$ , with the help of function *equations* for the computation of the velocities in the current time. *calc\_dV* and *calc\_dΩ* are constantly being called at every time step within the FLOWUnsteady simulation and, consequently, the new struct created of the type *AbstractManeuver* is constantly being actualized with the current values of the variables previously mentioned.

In function *equations*, the computation of the linear acceleration ( $\ddot{x}$ ) in the inertial frame is made using Newton's law:

$$m\ddot{x} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + RT_B + F_D, \quad (3.1)$$

where  $g$  is the gravitational acceleration;  $T_B$  is the thrust in the body frame (computed through a function created named *thrusttorque*) and converted to the body frame by the rotation matrix ( $R$ ) that goes from the body to the inertial frame and is computed by:

$$R = \begin{bmatrix} c_\phi c_\psi - c_\theta s_\phi s_\psi & -c_\psi s_\phi - c_\phi c_\theta s_\psi & s_\theta s_\phi \\ c_\theta c_\psi s_\psi & c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\psi s_\theta \\ s_\phi s_\theta & c_\phi s_\theta & c_\theta \end{bmatrix}; \quad (3.2)$$

$F_D$  represents the global drag forces and is computed by:

$$F_D = \begin{bmatrix} -k_d \dot{x} \\ -k_d \dot{y} \\ -k_d \dot{z} \end{bmatrix}, \quad (3.3)$$

in equation 3.2,  $c$  and  $s$  represent the functions cosine and sine, respectively, and in equation 3.3,  $k_d$  is the friction constant and  $\dot{x} = (\dot{x}, \dot{y}, \dot{z})^T$  the velocity of the quadrotor in the inertial frame.

The previously mentioned function *thrusttorque* computes the thrust ( $T_B$ ) and external torque ( $\tau_B$ )

vectors in the body frame, in the following manner:

$$T_B = \sum_{i=1}^4 T_{Bi} = k \begin{bmatrix} 0 \\ 0 \\ \sum \Omega_i^2 \end{bmatrix}, \quad (3.4)$$

and

$$\tau_B = \begin{bmatrix} Lk(\Omega_1^2 - \Omega_3^2) \\ Lk(\Omega_2^2 - \Omega_4^2) \\ b(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix}, \quad (3.5)$$

respectively. In equation 3.4,  $k$  represents an adequately proportional constant. In equation 3.5,  $L$  is the distance from the quadcopter's centre to any of its propellers geometric centre, and  $\omega$  the angular velocity vector (pointing along the axis of rotation). This is an overly simplified model, made in order to facilitate the simulation that does not take into consideration several nonlinear effects, such as blade flapping, rotational drag forces and the effects of the wind, amongst others. In both the equations, the index  $i$  and the numeration replacing it represent each one of the four rotors, as depicted in figure 3.3.

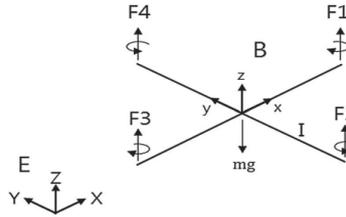


Figure 3.3: Quadrotor body (B) and inertial (E) frames [56].

The angular acceleration vector  $\dot{\Omega}$  is given by:

$$\dot{\Omega} = \begin{bmatrix} \dot{\Omega}_x \\ \dot{\Omega}_y \\ \dot{\Omega}_z \end{bmatrix} = I^{-1}(\tau - \Omega \times (I\Omega)), \quad (3.6)$$

where the quadcopter's inertia matrix  $I$  is given by

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}, \quad (3.7)$$

taking into consideration the symmetry of the quadcopter in figure 3.3, its inertia matrix must be diagonal, as represented in equation 3.7.

The angular velocity vector  $\dot{\theta}$  is computed through the function  $omega2thetadot$ , using:

$$\dot{\theta} = \begin{bmatrix} 1 & 0 & -s_{\theta} \\ 0 & c_{\phi} & c_{\theta}s_{\phi} \\ 0 & -s_{\phi} & c_{\theta}c_{\phi} \end{bmatrix}^{-1} \Omega, \quad (3.8)$$

denoting the difference between  $\Omega$  and  $\dot{\theta}$ , since the first constitutes the angular velocity with the previously mentioned direction, whilst the second is the derivative of the yaw, pitch and roll.

Finally, the angular velocity vector  $\Omega$  is updated by adding the angular acceleration multiplied by the time step. Similarly, for the angles in the body frame ( $\theta$ ) and their correspondent angular velocities ( $\dot{\theta}$ ), linear velocity ( $\dot{x}$ ) and linear acceleration ( $\ddot{x}$ ), and for the position ( $x$ ) and linear velocity ( $\dot{x}$ ). Finalizing, in this manner, the implementation of the dynamic movement for the present time step.

### 3.1.3 Optimization

#### Problem

The optimization problem for all continuous problems with a single-objective may be summarized in the following manner:

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && h_p(x) = 0, p = 1, 2, \dots, N_h \\ & && g_m(x) \geq 0, m = 1, 2, \dots, N_g \\ &\text{by varying} && x_k^L < x_k < x_k^U, k = 1, \dots, N_p \end{aligned} \quad (3.9)$$

where  $f$  is the objective function,  $x$  the vector of design variables,  $h$  the vector of equality constraints and  $g$  the vector of inequality constraints [57]. The objective function constitutes a scalar, computed for a given design variable vector  $x$ , that might be minimized or maximized, accordingly to the proposed problem. The computation of the objective function can range from a simple explicit equation to a system of coupled implicit methods, with the choice of this function being critical for the quality of the results that will be provided by the optimization. Each variable  $x_k$  is constrained between a lower value  $x_k^L$  and an upper value  $x_k^U$ , which constitute its side constraints [58]. The region within the design space where the variables satisfy all equality, inequality and side constraints is named as *feasible search region*. The example presented consists on a minimization problem and, consequently, the objective would be to minimize the objective function within the feasible search region. In the case of the problem consisting on the maximization of the objective function, one should assume:

$$\max[f(x)] = -\min[-f(x)]. \quad (3.10)$$

The described function evaluation is classified as being *direct*, since it is performed by solving numerical models of the system; it is also possible to implement *surrogate models*, using them in the optimization process.

## Numerical optimization techniques

There are several categories one can use to classify an optimization process.

Relatively to the order of information, the algorithm might be gradient-free, using only use the values of the computations of the objective and the constraint values for a given set of allowed variables, or gradient-based algorithms, using gradients of both the objective and of the constraint functions with respect to the design variables. By using gradient-based methods, the number of function evaluations required is reduced by many orders of magnitude, when compared with when the same evaluation is provided by gradient-free methods and, in addition, the optimization criteria used by these methods are more rigorous than the ones used by gradient-free methods [59]. Within the gradient-based methods are included, for example, linear and nonlinear programming. On its turn, gradient-free methods do not assume the function's continuity, characteristic that is essential for gradient-based algorithms. That is, albeit gradient-based algorithms allow the presence of a discontinuity away from an optimum point, gradient-free algorithms probably consist on the only adoptable option for functions where excessive numerical noise and discontinuities exist [59].

The algorithm used within the optimization process can be subdivided into mathematical and heuristic categories, accordingly to the degree on which it is grounded on provable mathematical principles. This definition is not exclusive, since most algorithms mix both mathematical and heuristic arguments to some degree. Whereas heuristic optimal criteria cannot prove the existence of a local optimum, mathematical criteria are unambiguous, converging to the optimum (within the working precision). Gradient-based algorithms are usually covered in the mathematical classification both for the optimal criteria and the iterative process, while gradient-free are split in both categories for both the optimal criteria and the iterative process. In a heuristic optimization, concepts found in nature are used to reach the global optimal solution [60]. These methods include, for example, Genetic Algorithms, Ant Colonies, Differential Evolution and Particle Swarm.

Heuristic methods are generally more flexible than mathematical models, not requiring an analytic definition of the objective function. On the down side, the quality of the solution is not guaranteed, they are computationally heavier and the increase of the number of variables decreases the probability of finding the optimal solution.

## Evolutionary Algorithms

Evolutionary algorithms (EA) are based on processes that occur in nature or in society. Their vast majority is population-based and derivative-free, enclosing several optimization algorithms, such as differential evolution (DE), evolution strategies (ES), genetic algorithms (GA), and evolutionary programming (EP) [61]. All of the algorithms mentioned are inspired by Darwin's evolution theory, namely by the principle of natural selection and the survival of the fittest. Through performing mutation and/or recombining current (*parent*) solutions, an EA generates its new solutions (*offspring*), at each of the algorithm's iteration.

EA are global (opposite to local) in terms of search classification, meaning that they try to span all of

the design space in the search of the global optimum; this comes from the fact that the vast majority of these algorithms are population based, involving a set of points that are spread within the design space at each iteration; heuristic in terms of their algorithm; classified as direct, in terms of function evaluation; and are stochastic (opposite to deterministic), in terms of stochasticity, meaning that they evaluate a different set of points if ran multiple times from the same initial conditions, contributing to not remaining at a local minimum. Although the presented features increase the likelihood of the global minimum being found, they do not guarantee it, especially taking into consideration the poor convergence rate of heuristic algorithms, made worse when the problems have a higher number of design variables. A general rule of thumb is that gradient-free methods are useful at finding a solution for small-size problems (usually  $\leq 30$  variables) [62].

GA are the most widely used type of EA. They constitute gradient-free, parallel optimization algorithms, vastly applied to complex systems [63]. In a GA, the initial set of design points is updated at each iteration. This iteration is named *generation* and each set of design points possesses a *population* with  $n_p$  points, each one of them represented by a *chromosome*, and containing values for all the design variables. Each one of the design variables is represented by a *gene*. The arrival to the optimal solution through the application of a genetic algorithm is achieved by the manipulation of chromosomes of previous solutions by genetic operators, in order to achieve new solutions that will insert the new generation [63]. This loop will be executed until the obtaining of a satisfactory fitness value or the satisfaction of other imposed criterion to the algorithm, such as the arrival to a maximum iteration time or the reach of a maximum number of generations, and includes the operations of crossover (or recombination), mutation and selection. The crossover or recombination operation exchanges a random portion of genes from a chromosome with an also random string from another. Mutation is the process of locally randomly altering a gene of children individuals, within the allowed range, increasing the variability of the population. After undergoing the previously mentioned operations, the selection process leads towards the optimum, preferring fitter individuals over the rest. This fitness value might only reflect the objective function and/or also reflect the level of constraint satisfaction. The chosen children are then inserted into the new population in a process named insertion. Most of the GA follow the presented procedure, existing many variations within them derived from the flexibility of the steps presented. The representation of the design variables divide the GA into two major categories: binary-encoded (this constitutes the original genetic algorithms) and real-encoded GA. The first group uses bits for the representation of the design variables, whereas in the second group the design variables are depicted as real numbers. The binary string representation is convenient for combinatorial problems, however, there are other type of problems that are jeopardized by it since binary mapping may require a rather complex encoding and decoding, being possible to lead to an introduction of multimodality [64].

DE, on its turn, consists on an improvement of the GA that leads to a faster and more robust optimization [65]. Unlike simple GA, DE usually uses floating point numbers as decision variables, making it a derivative-free, continuous function optimizer. This algorithm has been proven to be a simple effective evolutionary algorithm, when it comes to the optimization of several real-world problems [66]. The initial population is randomly generated accordingly to a normal or uniform distribution. After this process

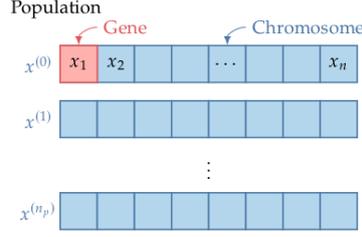


Figure 3.4: Schematic representation of a population, a chromosome and a gene [59].

of initialization, the DE algorithm enters a loop consisting on the evolutionary operations of mutation, crossover and selection. Similarly to GA, this loop occurs until either the algorithm converges to the true optimum solution or a certain termination criterion is fulfilled.

There are different possible strategies of DE based on the vector to be perturbed, the number of different vectors considered for perturbation and the type of crossover used. The ten different strategies proposed by Price & Storn [67] follow the nomenclature  $DE/x/y/z$ , where  $x$  represents the string of the vector to be perturbed (which can either occur only in the *best* vector of the previous generation, or in any random, *rand*, vector),  $y$  the number of vectors of  $x$  considered for perturbation, and  $z$  the type of crossover used. Although, in DE, the adopted strategy and parameters should be determined by trial and error, the method  $DE/rand/1/bin$  is considered to be the most successful and thus it is the most broadly used one [65] (where *bin* stands for binomial). In order to achieve a further comprehension of the evolutionary process in DE, the evolutionary operations will be subsequently described.

The subsequent mutation strategies are frequently used in literature, among others:

- ‘DE/rand/1’:  $v_{i,g} = x_{r0,g} + F_i(x_{r1,g} - x_{r2,g})$ ,
- ‘DE/current-to-best/1’:  $v_{i,g} = x_{i,g} + F_i(x_{best,g} - x_{i,g}) + F_i(x_{r1,g} - x_{r2,g})$ ,
- ‘DE/best/1’:  $v_{i,g} = x_{best,g} + F_i(x_{r1,g} - x_{r2,g})$ ,

where  $v_{i,g}$  is the created mutant vector for individual  $i$  of generation  $g$ ;  $r0$ , and  $r1$  and  $r2$  are distinct integers uniformly chosen from a set  $\{1, 2, \dots, NP\} \setminus \{i\}$  (where  $NP$  represents the population size;  $x_{best,g}$  is the best vector from the current generation  $g$  and  $F_i$  is the mutation factor. The optimal value for  $F$ , accordingly to Price & Storn, lies within 0.4 and 1.0, for most functions [67]. Contrarily to what is done in classic DE algorithms, where  $F_i = F$  normally consists on a single parameter for the entire generation, in many adaptive DE algorithms, the value of  $F_i$  is associated with an individual  $i$ .

After the vector undergoes a mutation, a binomial or an exponential crossover (these are the most common types present in literature) is performed between the mutant vector and its correspondent  $i$ -th population vector, originating a trial vector  $u_{j,i,g}$ . For a binomial crossover, the procedure is the following:

$$u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{if } rand_j(0, 1) \leq CR_i \text{ or } j = j_{rand} \\ x_{j,i,g} & \text{otherwise,} \end{cases} \quad (3.11)$$

where  $v_{j,i,g}$  and  $x_{j,i,g}$  are the  $j$ -th components of the  $i$ -th individual of the  $g$  generation mutation and

parent vectors, respectively;  $CR_i \in [0, 1]$  is the crossover probability, representing the probability of a parameter from a trial vector being inherited in individual  $i$  from the mutation vector;  $rand_j(0, 1)$  is a uniform random number within the interval  $[0, 1]$  generated for each  $j$ ; index  $j_{rand}$  is a randomly chosen integer within 1 and  $D$  for each  $i$ . When using classic *DE*,  $CR_i = CR$  constitutes a single parameter, whilst in adaptive *DE* algorithms,  $CR_i$  is a probability associated with each individual  $i$  and  $D$  is the dimension of each individual vector). In *DE*, the crossover is non-uniform so that the parameter values of the child vector are inherited in different proportions from the parent vectors.

Lastly, the trial vector  $u_{i,g}$  and its correspondent parent vector  $x_{i,g}$  undergo a selection process where the one that gets a greater fitness value  $f(\cdot)$  survives, entering the next generation. This process occurs for every parent and correspondent trial vector, with the survivors becoming the parents of the next generation.

As a title of example, when performing a minimization problem, the selection process is represented as:

$$x_{j,g+1} = \begin{cases} u_{i,g} & \text{if } f(u_{i,g}) < f(x_{i,g}), \\ x_{i,g} & \text{otherwise.} \end{cases} \quad (3.12)$$

A *DE* algorithm's performance depends intensively on the control parameters chosen (for example  $F$  and  $CR$ ) [61]. Several adaptive or self-adaptive mechanisms were proposed in order to execute a parameter control without the user's knowledge or interaction during the searching process. Accordingly to the classification for parameter control techniques proposed by Eiben et al. [68] regarding how the change of a parameter is made, three categories can be considered:

- Deterministic Parameter Control: occurs when a value of a parameter is changed due to a deterministic rule, not taking into consideration any feedback from the search;
- Adaptive Parameter Control: occurs when feedback from the evolutionary search is taken into consideration in order to dynamically change the control parameters;
- Self-Adaptive Parameter Control: consistently mentioned as an 'evolution of evolution'. In this category, the parameters to be adapted are encoded into the chromosomes of individuals, undergoing mutation and recombination. The underlying idea being that better parameters will origin fitter individuals, thus, with a greater survival probability and, consequently, this parameters will be propagated to a greater number of offspring.

From the characteristics above mentioned, one can be led to the conclusion that both adaptive and self-adaptive parameter control mechanisms, when well-designed, can lead to more robust optimization algorithms, eliminating the trial-and-error method and adapting the parameters dynamically during the optimization process accordingly to the presented problem, leading to an improvement in the algorithm's convergence.

Regarding the problematic of the application of constraints in EAs - that is, parents might satisfy certain constraints that offspring do not (which is understandable when taking into consideration that both the mutation and recombination operations do not take constraints into consideration) -, this difficulty

can be handled directly, through the adaptation of the EA, or indirectly, through the incorporation of the constraints in the fitness function [63].

As the topic of design optimization is broad and extensive, for a more acute comprehension, a detailed background is recommended [59]; regarding the topic of differential evolution, for a more extensive approach the bibliography by Zhang and Sanderson is proposed [61].

## B-spline theory

The theory presented in the current subchapter is based in references [69–71].

Bézier splines constitute parametric curves used to draw smooth lines, being straight if, and only if, it is possible to draw a straight line through all of the control points.

The general form of a  $n$  degree Bézier curve is defined using  $n+1$  control points and its mathematical expression is given by:

$$B(t) = \sum_{i=0}^n b_{i,n}(t)P_i \text{ with } t \in [0, 1], \quad (3.13)$$

where  $C$  is the obtained curve,  $t$  is the parameter used,  $P_i$  denotes the  $i$  –  $th$  control point, and  $b_{i,n}$  is the Bernstein basis polynomial of order  $n$ , given by:

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i} \text{ with } i = 0, \dots, n, \quad (3.14)$$

where  $\binom{n}{i}$  denotes the binomial coefficient.

The Bézier curves respect the following properties [72]:

- the basis polynomials are real;
- the number of control points is equal to the degree of the Bézier curve plus one;
- generally the Bézier curve follows the shape of the defining polygon;
- the first and last points on the curve and on the defining polygon are coincident, respectively;
- the tangent vectors at the ends of the curve and the first and last polygon spans have the same direction;
- the curve is contained within the largest polygon obtainable with the defining vertices (this property is often referred to as the convex hull property);
- the curve does not oscillate about any straight line more often than the polygon itself (this is referred to as the variation diminishing property);
- the curve maintains its shape under an affine (linear) transformation.

B-splines consist on several Bézier curves joint together. A B-spline of degree  $k$  is defined by the following equation:

$$S(t) = \sum_{i=0}^n N_{i,k}(t)P_i, \quad (3.15)$$

where  $S$  stands for 'spline', and  $N_{i,k}$  are known as the basis functions, defined by the Cox-de Boor recursion formula:

$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t) \quad t \in [t_0, t_m], \quad (3.16)$$

where  $m = k + n + 1$  represents the relation between the number of knots,  $m + 1$ , and the number of control points,  $n + 1$ , with  $k$  being a natural number belonging to  $[0, k]$ . In this manner, a B-spline of order  $k$ , defined by  $n + 1$  control points, will consist of  $n - k + 1$  connected Bézier curves. Therefore, the advantage of B-splines over Bézier curves consists on the fact that, in the first, the number of control points are only restringed inferiorly by the degree of the interpolation (the number of control points must be higher than  $k$ ), whilst in Bézier curves the number of control points is also restringed superiorly, being equal to  $k + 1$ .

$N_{i,0}$ , by its turn, is given by:

$$N_{i,0} = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \quad (3.17)$$

There are three assumptions that B-splines must obey:

- $C^0$  continuity, meaning that the curve is continuous in the control points.
- $C^1$  continuity, that is to say that the first derivative is continuous in the control points.
- $C^2$  continuity, which is analogous to  $C^1$ , but for the second derivative.

B-splines are classified as uniform when their points are equally distant. When B-splines are uniform they are not defined for the entire range of the knot vector  $T$ . This can be corrected by setting the first undefined  $t$  to  $t_k$  and the last to  $t_n$ , that is, the knots going from  $t_0$  until  $t_{k-1}$  and from  $t_{n+1}$  until  $t_m$  in the uniform B-splines will all be replaced by  $t_k$  and  $t_m$ , respectively. This correction originates the so called open uniform B-splines.

When performing a computational polynomial interpolation, a smoothing factor  $s$  might be provided. The value of  $s$  allows the number of knots in the interpolation to be increased while  $s$  still satisfies equation 3.18:

$$\sum ((w[i] \times (y[i] - spline(x[i]))) \times 2) \leq s, \quad (3.18)$$

where  $w$  is the weight applied to each point  $i$ , and  $x$  and  $y$  are the coordinates of point  $i$ .

### Optimization Framework Description

An optimization code, in order to decrease the noise emissions, was developed to obtain the optimal blade geometry. In the present optimization, a plane-bladed rotor was studied when performing a hover

flight at 4000 RPM. The design parameters chosen were the tip radius, the chord and the twist distribution along the rotor's span, and the distributions of the height (which describes the height of the leading edge from the top face of the rotor's hub) and the sweep of leading edge (which describes the distance of the the leading edge from a line that comes from the center of the rotor's hub) along the span of the rotor. The original design point is present in annex C, where the distribution of the mentioned variables along the span and the radius at the tip of the rotor are depicted. The code makes use of an adaptation of FLOWUnsteady and its dependencies created specifically for the purpose. Figure 3.5 presents a flowchart of the optimization code developed that will be thoroughly described in the present section.

Firstly, the basis geometry of the rotor is created within the simulation. After, the user can optimize the rotor's configuration through a b-spline interpolation, of a chosen degree  $k$  (with 5 being the maximum degree possible), using the values of the design variables at a chosen number of  $n$  points along the span of the rotor. This manner of designing blades presented is already portrayed in literature, for example, applied to propeller blades, being a representation of b-splines surfaces for blades widely used [73]. The number of points used is likely to be inversionally proportional to the smoothness of the control surface, being a small number of control points desired [73]. The geometry values are passed to the optimization loop as the original design variables. The computation of the initial energy averaged overall sound pressure level (EAOASPL) value is made directly using the rotor created from the FLOWUnsteady simulation with the original geometry. The points of the design vector created on each iteration are posteriorly interpolated via the use of a b-spline in order to obtain the blade's new geometry along its span. This interpolation was done using the julia Dierckx package [74, 75], already implemented in FLOWUnsteady. The smoothing factor  $s$  was extrapolated accordingly to each case, within reasonable limits, taking the fact that the interpolation did not necessarily have to pass through the exact points suggested by the optimization algorithm, and thus  $s$  did not have to be zero, but also it is needed for this parameter to be small enough so that not only the limits imposed to the search range are respected, but also the geometry makes physical sense. The package Dierckx was modified so that the code provided with an approximate polynomial for the cases where the performed interpolation could not satisfy equation 3.18 within 20 steps, returning an approximate value for the design variables, without stopping the optimization process. In this manner, a b-spline of the curve for the design parameters in function of the adimensionalized radius (radius of the section divided by the radius at the tip of the hub) is made and saved on the CSV file of the optimized blade.

Since the main goal of a rotor is to provide a certain amount of thrust ( $T$ ), this component was introduced as a constraint to the design space, so that the thrust of the obtained optimized blades performing under the same flight conditions would not be inferior to the one obtained using the basis geometry. In this manner, the component of performance was introduced in the optimization code. Regarding the implementation of the mentioned constraint, an indirect method was chosen where a penalty on the objective function was imposed in case of the violation of the thrust constraint. This method will guarantee that the fitness value of the current iteration obtained for geometries that violate the thrust constraint would be outside of the scope of the solution and, consequently, be greater than the fitness values obtained when no violation occurs. As follows, the thrust of the original blade when

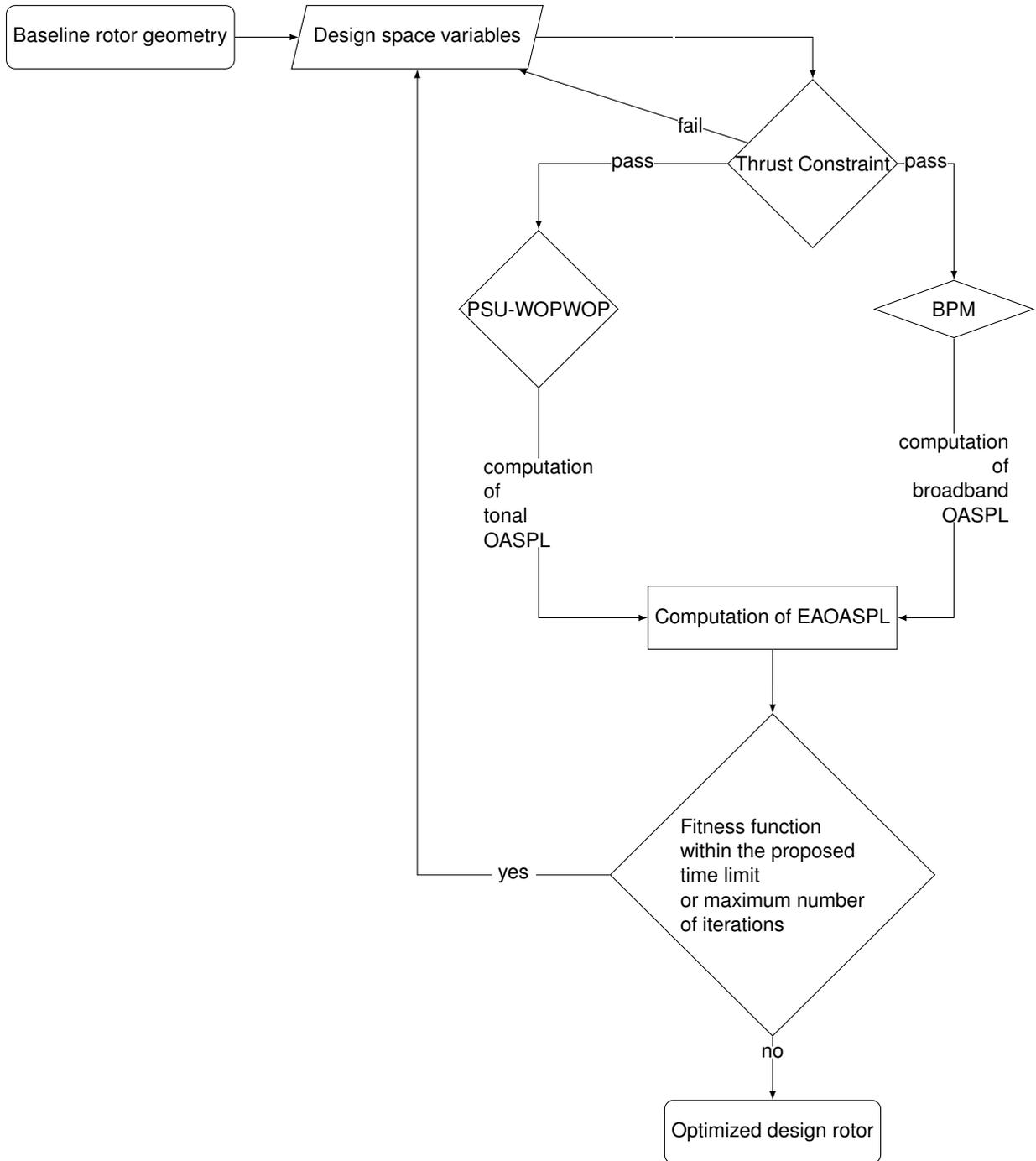


Figure 3.5: Flowchart of the optimization code.

performing a simulation at the evaluated flight conditions was introduced as a state variable (functioning as a constraint) with its bounds being equal to itself, securing the retaining of the value of the basis thrust throughout the optimization process. In doing so, the initial thrust enters the loop as a design variable. Logically, when within the optimization loop, the first step implemented consists on the creation of a new rotor and on its testing for the thrust constraint under the given flight conditions. In the case of its thrust being lower than the one of the basis rotor, the current iteration is stopped and the penalty value previously mentioned will constitute the fitness value of the current iteration. A new iteration will after begin, where a new rotor is to be implemented.

In case the algorithm successfully passes the thrust constraint, the tonal and the broadband OASPL in decibels are computed respectively via the use of the PSU-WOPWOP binary [44] (from Pennsylvania State University) and the FLOWBPM [47] packages, and added together. This procedure is done by converting each tonal and broadband OASPL at a given frequency to its corresponding root mean square pressure, adding them together, and converting the result back into decibels. The output CSV files containing the OASPL at the grid of microphones simulated are read, but these cannot be directly optimized, since the method available within BlackBoxOptim only allows for the optimization of a scalar (i.e., the value returned by the objective function must be a scalar). Consequently, the EAOASPL was chosen as a metrics for the optimization process, as it consists on a logarithmic average of the OASPL measured in each microphone, therefore taking into account both tonal and broadband components of the noise. The EAOASPL is given by the following equation:

$$EAOASPL = 10 \log \sum_{i=1}^N 10^{\frac{OASPL(i)}{10}}, \quad (3.19)$$

where  $OASPL(i)$  is the value of OASPL registered in the  $i$ -th microphone of the sphere, and  $N$  is the number of data points across the surface of the aeroacoustic sphere. In the case implemented,  $N$  is represented by 360 microphones placed around the rotor, equally spaced at a  $90^\circ$  degree rotation plane relatively to the rotor's plane. After the EAOASPL is computed, it is returned as the fitness value of the present iteration. It is worth mentioning that the use of the EAOASPL in rotor optimization is documented in previous literature [76].

The mathematical formulation of the described problem is present in equation 3.20.

$$\begin{aligned} \text{Minimize} \quad & f(x) = 10 \log \sum_{i=1}^N 10^{\frac{OASPL(i)}{10}} \\ \text{with respect to} \quad & x = R_{tip}, \quad c, \quad \theta, \quad LE_z, \quad LE_x \\ \text{subject to} \quad & c = T_{baseline} - T_{optimized} \leq 0 \end{aligned} \quad (3.20)$$

The maximum number of function evaluations and/or the maximum running time can be defined by the user, in this case, if the computational time is within its chosen limit and/or the number of function evaluations is lower than the maximum number of steps defined, the loop continues. Otherwise, it ends and the optimized design variables corresponding to the lowest EAOASPL can be saved into

their correspondent CSV files through implemented functions. The files created contain information in a way that is properly supported as an input for the FLOWUnsteady software, so that they can be postprocessed using this same software.

Taking into consideration the absence of a definition of a single expression for the objective function (bearing in mind that the EAOASPL uses the OASPL values that rely on intricate expressions, the subsequent theory of which is explained in chapter 2), a gradient-free method must be used, as explained in section 3.1.3. Within gradient-free methods, an adaptive differential evolution was chosen and applied in the optimization process given that, when there might exist multiple local optima, as in the present case, evolutionary algorithms become a preferable solution as they allow the exploration of all parts of the feasible design space [63]. The method chosen was the adaptive differential evolution algorithm *DE/rand/1/bin*, as it is considered to be the most successful method comprised within adaptive differential evolution [65]. The *DE/rand/1/bin* method was implemented through the use of the Black-BoxOptim julia package [77]. Considering a function as a “black box” means that the observer only sees the inputs (including the design variables and, in the present case, also the thrust constraint) and the outputs (i.e., the objective EAOASPL and the optimal geometry distributions), with limited or no understanding of the modeling and of the numerical solution process that is used for the purpose of obtaining the fitness value.

Regarding the DE algorithm, the  $F$  and  $CR$  parameters were chosen each through the use of a Cauchy bimodal distribution, the parameters of which were based on the parameters proposed by Wang et al. [78] and Zhang and Sanderson [79]. In this manner, if the trial vector enters the next population  $F_{i,G+1} = F_{i,G}$  and  $CR_{i,G+1} = CR_{i,G}$ , whereas if the contrary occurs, then  $F_{i,G}$  and  $CR_{i,G}$  are generated accordingly to equations 3.21 and 3.22, respectively (where  $F_{i,G}$ ,  $CR_{i,G}$ ,  $F_{i,G+1}$ ,  $CR_{i,G+1}$  represent the mutation factors and crossover rates of each individual within the population of generations  $G$  and  $G + 1$ , respectively):

$$F_{i,G} = \begin{cases} randc_i(0.65, 0.1) & \text{if } rand(0, 1) \leq 0.5 \\ randc_i(1.5, 0.1) & \text{otherwise} \end{cases}, \quad (3.21)$$

where  $randc_i(a, b)$  is a random number that obeys the Cauchy distribution given by location parameter  $a$  and scale parameter  $b$ , and

$$CR_i = \begin{cases} randc_i(0.1, 0.1) & \text{if } rand(0, 1) \leq 0.5 \\ randc_i(0.95, 0.1) & \text{otherwise} \end{cases}. \quad (3.22)$$

Concerning the mutation factor  $F_{i,G}$  the use of a higher location parameter (1.5) provides a global exploration, whereas the lower parameter (0.65) tends to focus on local exploitation. Whereas concerning the crossover rate  $CR_i$ , the use of a higher location parameter (0.95) promotes the exploration by allowing the trial vector to inherit more information from the mutant vector, whereas the lower parameter (0.1) denotes that the trial vector is likely to be more similar to the target vector, accelerating the convergence by focusing on the neighbor of the parent. Regarding both equations 3.21 and 3.22, the scale factor of 0.1 symbolizes that both  $CR$  and  $F$  are located in a small neighbourhood of the location

parameters.

### Design Constraints

The design variables were altered within the loop of the designed optimization code, respecting the bounds present in table 3.1. The bounds were chosen accordingly to a market study of the geometric data of rotors DJI9441 and DJI-II [41, 80], since these rotors have a tip radius within 0.12 and 0.13 meters, being thus of similar dimensions and being used for alike applications, namely in drones [80].

When performing a B-spline optimization, one has to consider that the limits present in table 3.1 are the constraints for the control points along the blade's span, not being directly the values of the design variables, but majoring them. These bounds enclose the control polygons of the B-splines, as commented in 3.1.3 (the CSV files of the obtained optimized geometries where this phenomenon occurs can be observed in annex C).

Table 3.1: Design constraints of the optimization code.

Design variable	Lower bound	Upper bound
Tip radius (m)	0.1	0.4
Chord/tip radius	0.01	0.03
Theta (degrees)	1.00	20.00
Leading edge height/tip radius	-0.007	0.03
Leading edge twist/tip radius	-0.03	0.05
Thrust	1.0	1.0

## 3.2 Verification and Validation

Regarding the accuracy of the code, a comparison between the results obtained running the FLOWUnsteady software for the DJI9443 (the geometry of which is further described in [41]) rotor and the experimental data obtained by Zawdny et al. [81] when studying that same rotor under the same flight conditions is presented within FLOWUnsteady's validation [53]. As the OASPL was the metrics used as a mean to compute the result of the objective function of the optimization code created (described in section 3.1.3), figure 3.6 is presented to validate the use of FLOWUnsteady as a basis for the noise computation integrated in the optimization code developed. Figure 3.6 shows a good accordance between the experimental and the simulational OASPL, with a greater deviation for an observer at 0° (of approximately 5dB, possibly related to an underestimation of the thickness noise, which is the major component on this direction), and a perfect match for an observer at 45°. The simulation was performed using a total of 360 observers equally spaced in a circular array with a 1.905 metres of radius from the

hub of the rotor, being studied at a  $90^\circ$  plane with the rotor's plane. Frequency-wise, the comparison between the experimental data by Zawodny et al. [81] across the frequency range with the data provided by the FLOWUnsteady simulation shows there is a good prediction of the first BPF and the general trend of the SPL until a frequency of approximately 40 Hz, as presented in figure 3.7.

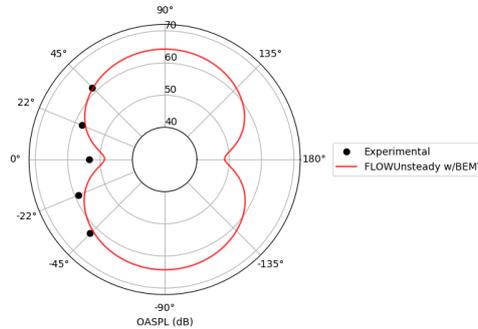


Figure 3.6: OASPL (dB) obtained for observers in a circular array of microphones located at a  $90^\circ$  plane with the rotor's plane at 1.905 meters distance from the rotor's hub, both experimentally by Zawodny et al. [81] and by the FLOWUnsteady software with BEMT simulation [53] for a rotor performing in hover at 5400 RPM.

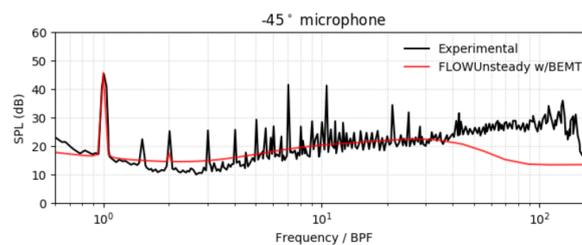
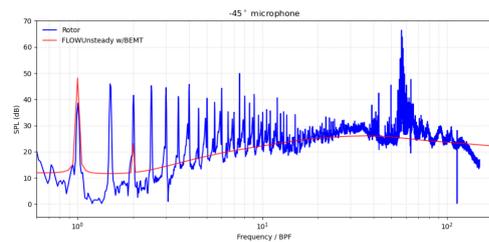


Figure 3.7: SPL (dB) obtained for an observer position at an angle of  $-45^\circ$  of an  $90^\circ$  plane with the rotor's plane at 1.905 meters distance from the rotor's hub, both experimentally by Zawodny et al. [81] and by the FLOWUnsteady software with BEMT simulation [53], for a rotor performing in hover at 5400 RPM.

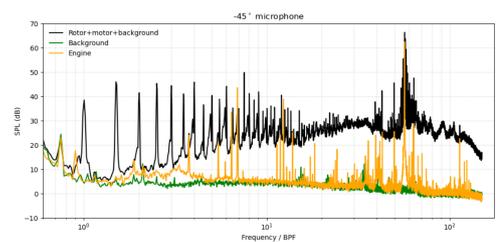
For a further study of the validation of the FLOWUnsteady code, a study performed by Gabriele Bossotto [54] can be consulted by the reader, where the validation of the FLOWUnsteady software is tested against results obtained in the aeroacoustic wind tunnel of Instituto Superior Técnico. These results are of particular importance regarding the validation of the optimization results (presented in chapter 4), since the rotor utilized for the validation process is of similar dimensions to one used in the optimization within the current thesis. The complete description of the geometry of which is present in [54]. Bossotto's portrays a good agreement between the experimental and computational values of SPL for the mentioned rotor in hover with rotational values above 2500 RPM for microphones at  $45^\circ$  and  $-45^\circ$  at a  $90^\circ$  plane with the rotor's plane, whereas for microphones at a  $0^\circ$  angle it presented a slightly lower value of SPL for simulation values [54] (see figures 3.8(a) and 3.8(c)). Particularly for a RPM equal to 4000 (the same RPM as the one used in the optimizations performed within the current thesis), FLOWUnsteady managed to follow the SPL trend, with the first BPF being well captured at both angles of the microphones studied ( $0^\circ$  and  $-45^\circ$ ) - as presented in figures 3.8(a) and 3.8(c). Although a peak between 7560 and 7570 Hz that is not predicted by the FLOWUnsteady software is also noticeable in figures 3.8(a) and 3.8(c), this is not correlated with a misprediction of the software since it should

mainly be due to the noise produced by the electric motor utilized in the experimental procedure that was not manageable to subtract when obtaining the rotor's noise (acquired through the subtraction of the background's and motor's noises to the total noise measured in the microphones). This phenomenon can be observed in figures 3.8(b) and 3.8(d), where a peak is present for the noise produced by the motor in this same range of frequencies. Once more, for an observer at  $0^\circ$ , a greater deviation between experimental and computational values is presented, accordingly to figure 3.8(c), for the previously explained reasons.

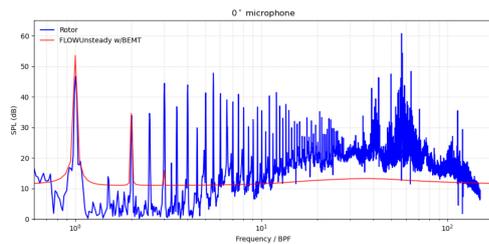
For a wider knowledge of the process behind the obtaining of the experimental data for the validation of the FLOWUnsteady software, references [81] and [54] are recommended.



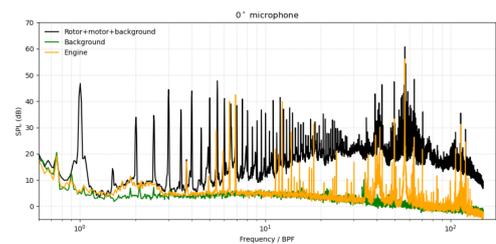
(a) Experimental and computational (FLOWUnsteady with BEMT) SPL comparison for an observer at  $-45^\circ$ .



(b) Experimental SPL for the whole structure, engine and background at  $-45^\circ$ .



(c) Experimental and computational (FLOWUnsteady with BEMT) SPL comparison for an observer at  $0^\circ$ .



(d) Experimental SPL for the whole structure, engine and background at  $0^\circ$ .

Figure 3.8: SPL (dB) obtained for an observer at a  $90^\circ$  plane with the rotor's plane, positioned at a 2.3 meters distance from the rotor's hub, for a rotor in hover at 4000 RPM [54].

### 3.3 Code Approximations and Limitations

Apart from the assumptions made by the FLOWUnsteady software, described as a part of the theoretical background behind the software in section 3.1, the limitations directly derived from the optimization code created should also be referred.

There were several alterations implemented in the introduced packages in order to facilitate the user's experience, such as the changing of the original smoothing parameter  $s$  to 0.1 chosen, bearing into consideration that the interpolation did not necessarily have to pass through the exact points suggested by the optimization algorithm, and thus the smoothing factor did not have to be zero, but also it should be small enough so that the obtained geometries made physical sense. The 20 limit number of iterations that the program had originally implemented was deleted in order to allow the return of an approximation in the case of the parameter  $s$  not being satisfied, without stopping the optimization cycle.

Albeit the previously mentioned, the user should take into consideration the limitations of generating a rotor in this software, being aware of these concerns when choosing the search domain, which can sporadically lead to a trial or an error situation during the selection of the design variables. Particularly, close attention must be taken to the domain provided for the range of the chord, since its value cannot be negative when performing the spline interpolation and also that it will influence the Reynolds number computed by *xfoil*. Moreover, since *xfoil* computes the aerodynamic coefficients for a 2D blade, when provided with less conventional blades, very different from NACA profiles, it might present difficulties when performing the boundary layer analysis and airfoils polar definition. These might lead to a premature ending of the program. It is left to the user to provide a feasible search range for its physical implementation. It is important to underline that the restriction of the domain will not compromise in any way the efficiency of the procedure, being that the code will only provide an optimized blade when a feasible physical solution is possible, presenting an error to the user otherwise, given that the simulation software will impose limits for a feasible blade.

A number of thirty blade elements was chosen in order to decrease the computational effort. By doing so, since the same number of blade elements was used for the basis and the optimized blades, the optimization results were not compromised. This radial-wise division was used for the elements within the BEM theory, when computing the blade loading.

Albeit noise regulations using the A-OASPL as their metrics, as described in section 2.2, the optimization code took the EAOASPL (which, by its turn utilizes the OASPL in its computation) as its metric, since the software utilized produces more accurate results for the second. This can be seen when comparing figure 3.9 with figure 3.6, where the difference between the experimental A-OASPL results and the ones obtained when using the FLOWUnsteady software is clearly higher than the the difference between the experimental results and the simulation results for the OASPL. This is due to humans being most sensitive to the frequency range between 1000 and 10000 Hz, and the accuracy of the FLOWUnsteady software being greater at lower frequencies, as shown in figure 3.7.

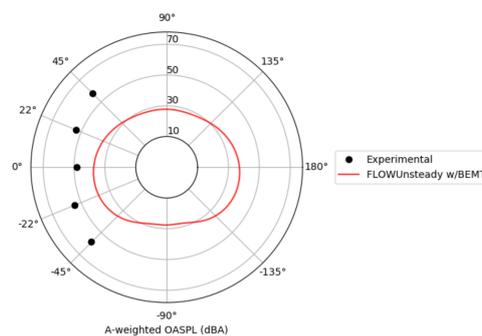


Figure 3.9: A-OASPL(dB) obtained for observers in a circular array of microphones located at a 90° plane with the rotor's plane and at 1.905 meters distance from the rotor's hub both experimentally by Zawodny et al. [81] and by the FLOWUnsteady software with BEMT simulation [53].

# Chapter 4

## Results

### 4.1 Introduction

An optimization using splines along a rotor blade's span was implemented in the present work. This constitutes a commonly used technique with optimizations involving shapes that very often leads to a decrease on computational times with little loss in the model parametrization fidelity [59, 82].

The conditions/assumptions made in the performed optimizations are depicted in table 4.1, where  $J$  represents the advance ratio,  $R$  is the radius at the tip of the blade of the rotor, and  $RPM$  the revolutions per minute of the simulated rotor. The properties of the air used during the optimization process are presented in table 4.2. A circular array of 360 microphones, separated 1 degree from one another, at a radial distance of 1.905m from the rotor hub, located at a perpendicular plane to the plane of rotation were simulated as the points where to measure the acoustic results.

Table 4.1: Conditions for the optimization scenario.

Number of blade elements	$\Omega$ [RPM]	Advanced ratio	Freestream velocity[m/s]
30	4000	0.0001	$J \cdot RPM / 60 \cdot (2 \cdot R)$

Table 4.2: Properties of the air used during the optimizations.

Density[kg/m <sup>3</sup> ]	Dynamic viscosity [kg/ms]	Speed of sound [m/s]
1.071778	1.85508e-5	342.35

The influence of the number of control points and the degree of the spline was also studied in the current optimization process. Several interpolation studies were performed before reaching the chosen optimized blade, namely: a first order 2 point spline ( $n2k1$ ); a first, and a second order 3 point spline

( $n3k1$ , and  $n3k2$ , respectively); a first, a second, and a third order 4 point splines ( $n4k1$ ,  $n4k2$ , and  $n4k3$ , respectively); a first, a second, a third, and a fourth order 5 point splines ( $n5k1$ ,  $n5k2$ ,  $n5k3$ , and  $n5k4$ , respectively); a first, a second, a third, a fourth, and a fifth order 6 point splines ( $n6k1$ ,  $n6k2$ ,  $n6k3$ ,  $n6k4$ , and  $n6k5$ , respectively).

## 4.2 Computational

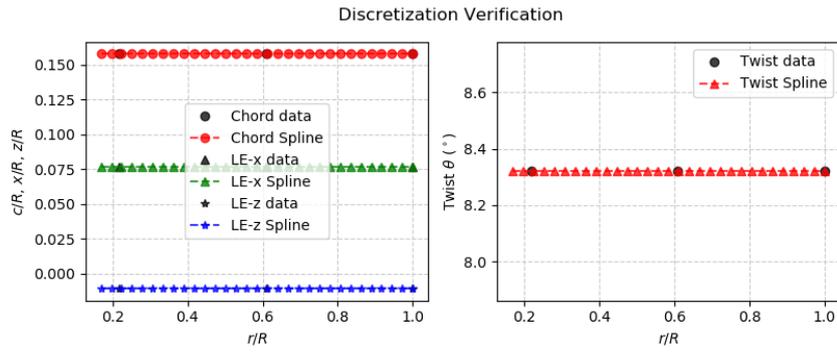
The computational optimization module created was run incorporating an adapted FLOWUnsteady software for the performed acoustic simulations. Both OASPL and SPL graphs were plotted in order to be able to register not only the directions of the noise reduction, but also the differences suffered by the SPL on the frequency spectra, respectively.

These results were explained based on the physical characteristics of the optimized blades obtained during the process. The thickness, loading and broadband OASPL graphs were also plotted in order to further understand which nature of the rotor noise was more affected by the optimization performed, describing it based on the physical impacts of the optimized geometries obtained. Needless to say that the influence of the parameters would be easier to inspect if a gradient based algorithm was used. Nonetheless, due to the intricate functions used within the optimization, this type of method was not possible to perform, and the characteristics of the obtained blades are well mirrored on the changes obtained in the acoustical graphs, when comparing these with their analogous from the basis blade geometry.

In order to overcome the limitations presented by the *xfoil* software, an EAOASPL optimization was performed with the constraint of the thrust obtained being greater than 1.0. This imposed a looser restriction on the loading forces, thus decreasing the loading noise. Nevertheless, this constraint adopted during the optimizations did not impact the goal of the present study of the physical impact of the geometry of the rotor on the noise levels obtained during its flight performance. In this way, the alteration of the *xfoil* software, or the junction of the optimization package with a commercial software that already supplanted these restrictions, leading to the possibility of imposing more restrictive constraints is left as a suggestion for future work. In this manner, the values of the optimized design variables for the obtained geometries through the use of the optimization module created were compared amongst them, in order to study their physical impact on noise production.

### 4.2.1 Basis Blade

A plane blade was used as the basis for the optimization process in order to measure the magnitude of the possible EAOASPL optimization that could be obtained with the created framework, providing the optimizer module with a simple non-optimized geometry as its starting point. The geometry of the original rotor is present in figure 4.1, with its discretization presented in figure 4.1(a), and for a broader description of the input geometry, the corresponding CSV files are present in annex C. The initial rotor had a 0.17775m radius at its blade tip and a 0.03m distance from its center axis until the end of its hub.



(a) Geometric discretization of the basis rotor.



(b) 3D representation of the basis rotor.



(c) Contour of the original rotor.

Figure 4.1: Geometry of the original rotor.

### Numerical Study

In this section, the results of the polynomial order ( $k$ ) and the number of interpolation points ( $n$ ) studies are presented, measuring their impact on the results obtained. The function *Spline1D* provided by the *dierckx julia* package was used for the interpolation of the design variables along the span of the blade.

In the current study, the termination criteria for running the optimization was the code getting to an unfeasible solution, in order to see the adaptation of the splines to the points generated by the module. For this reason this criteria was chosen in detriment of the more frequently imposed in literature (such as the maximum number of steps or the maximum time allowed for the optimization process).

The results of the optimization are presented in table 4.3. In this same table, one can observe that, in a general manner, the number of total iterations performed within the optimization frame increases with the decrease of both the number of points and of the spline order, of course with the exception of certain orders that adequate more to a certain numbers of points - meaning that a certain  $k$  may not function for a particular number of interpolation points  $n$ , due to the interpolation itself. This mainly derives from the creation of untypical blade formats, that might lead to unfeasible pressure distributions computed within *xfoil*.

The geometry of the most optimized blade present in table 4.3, that is a two point first order optimization ( $n2k1$ ), is going to be discussed in more detail to evaluate the pressure directivity distributions of the different noise components in section 4.2.2. In section 4.2.3, a comparison between different

Table 4.3: Optimization results.

Case	Total Number of Iterations	Best Iteration	EAOASPL (dB)	Thrust (N)
Basis blade	-	-	66.08	3.62
$n2k1$	188	17	55.37	1.03
$n3k1$	0	-	-	-
$n3k2$	27	22	63.4	2.24
$n4k1$	47	17	65.00	2.22
$n4k2$	6	2	57.02	1.24
$n4k3$	35	34	60.00	1.49
$n5k1$	109	39	55.50	1.05
$n5k2$	90	9	57.86	1.12
$n5k3$	17	9	58.84	1.63
$n5k4$	25	22	65.39	2.24
$n6k1$	11	8	58.47	1.50
$n6k2$	105	63	56.67	1.14
$n6k3$	11	-	-	-
$n6k4$	1	-	-	-
$n6k5$	22	2	62.06	1.38

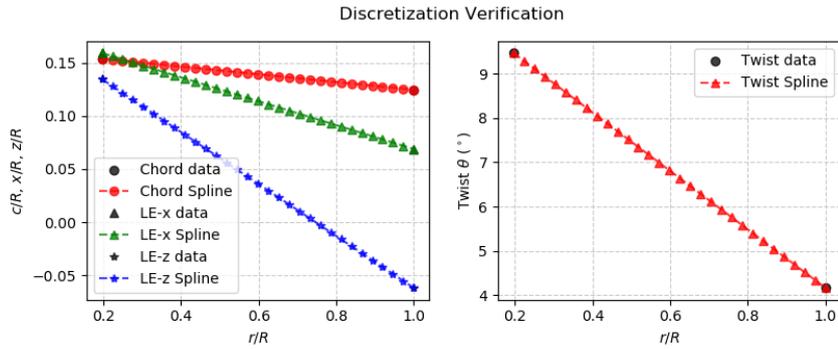
cases namely  $n2k1$ ,  $n5k1$ , and  $n6k2$  spline optimization is performed - due to the EAOASPL and thrust proximity obtained within these cases, as presented in table 4.3 - as basis for the study of the impact of the physical characteristics of the blades obtained during hover. From further inspection of table 4.3, the cases of  $n2k1$ ,  $n5k1$ , and  $n6k2$  appear to be close to an optima, having reached the value of the minimum thrust imposed by the ruling optimization constraint, that is, having reached thrust values close to the  $1N$  thrust constraint imposed. For further knowledge of the optimization results obtained for the remaining cases, the reader is advised to consult annex D.

#### 4.2.2 Optimized Blade

The blade that suffered the greatest reduction in EAOASPL was obtained when performing a two point second order spline ( $n2k1$ ), as represented in table 4.3. This blade obtained a 10.71dB decrease of the EAOASPL, when comparing with the non-optimized blade, which represents a 16.21% reduction. This is mainly due to the reduction of the loading noise achieved by the loosening of the thrust constraint on the optimization code. The mentioned EAOASPL reduction is further represented in figure 4.3(a), where a comparison between the original and the optimized OASPL is showed.

Regarding the geometry of the obtained optimized blade, its discretization along its span is depicted

in figure 4.2(a), its CSV files further describing the geometry obtained are present in annex C, and the modeling of the obtained geometry and its contour are represented in figures 4.2(b) and 4.2(c), respectively. As aforementioned, the physical interpretation of the obtained geometry will be performed in section 4.2.3, along with the comparison with other optimized blades that performed similarly, whereas in the current section, the overall directivities of the noise components and their importance will be reflected upon.



(a) Geometric discretization of the optimized rotor.

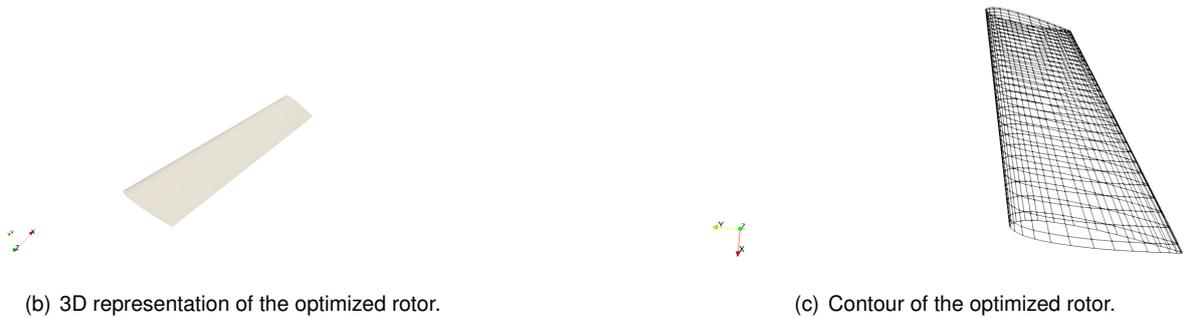


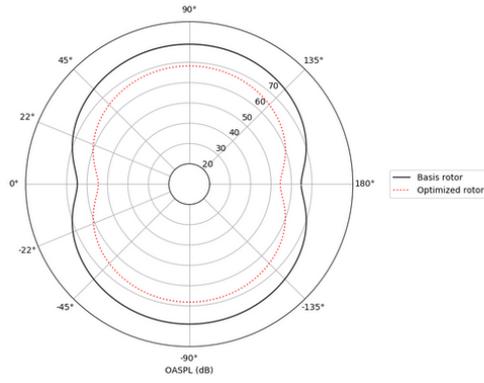
Figure 4.2: Geometry of the optimized two-bladed rotor with a  $n2k1$  spline optimization.

The OASPL of the basis and of the optimized rotor were decomposed into their frequency and broadband components, so that their importance is depicted.

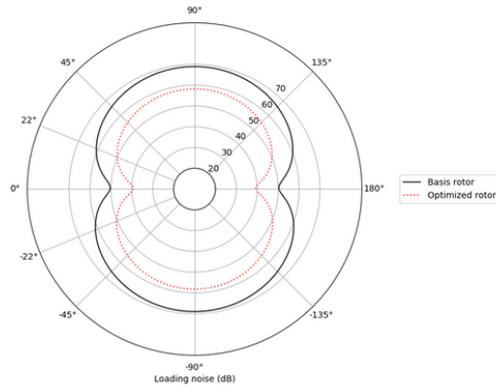
Firstly, the tonal noise components of the OASPL, namely the loading and the thickness noises were studied, as present in figures 4.3(b) and 4.3(c), respectively. The first constitutes the largest component of the OASPL, being mainly propagated in the perpendicular direction, whilst the second is mainly propagated in an oblique direction. In this manner, for the in-plane microphone positions, thickness noise will dominate the pressure-time history, whereas for observers above or below the rotor tip-plane, steady thrust becomes the dominant contributor to the measured noise (that is, the loading noise will be higher at these positions), whilst the thickness noise is lessened.

Regarding the results obtained of the loading noise, due to the thrust constraint being inferior to the thrust of the basis blade, it is expected that figure 4.3(b) presents a significant reduction for the loading noise for the optimized blade, as it does.

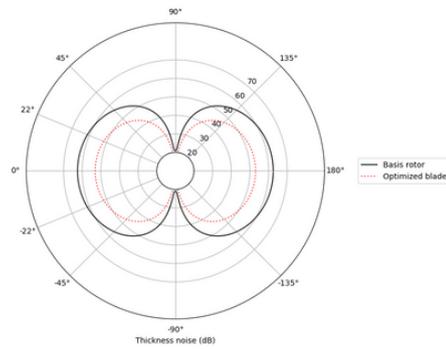
In terms of broadband noise, this component presents the lowest pressure levels, as depicted in figure 4.3(d). In the current simulation conditions, this type of noise is mainly composed by laminar boundary layer vortex shedding and tip vortex formation noise (see section 2.3.2).



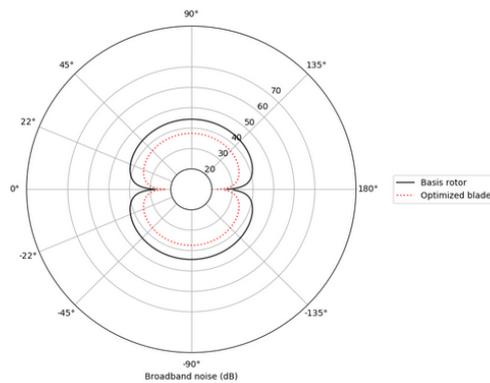
(a) OASPL (dB) of the optimized blade.



(b) Loading noise (dB) of the optimized blade.



(c) Thickness noise (dB) of the optimized blade.



(d) Broadband noise (dB) of the optimized blade.

Figure 4.3: Noise results of the optimized blade.

### 4.2.3 Comparison between cases: $n2k1$ , $n5k1$ , and $n6k2$

The results of  $n2k1$ ,  $n5k1$ , and  $n6k2$  geometries optimization (which are further described in annex C) are proposed to be interpreted in the current section, as the more general noise interpretations have already been covered in section 4.2.2. These cases were chosen due to their relatively close EAOASPL and thrust values obtained within the optimization process, as shown in table 4.3. Another main factor that allowed the choice of the mentioned geometries is that they present a value close to the minimum thrust imposed, leading to believe that even if not fully optimized, they present geometries close to a noise minima within the design space. The discretizations along the blade span of the geometries obtained for the cases of the  $n2k1$ ,  $n5k1$ , and  $n6k1$  are depicted in figures 4.2(a), 4.5(a), and 4.6(a), respectively.

Table 4.4 presents the values obtained for the radius at the tip of the blades for the cases studied. In all cases, a reduction of the radius at the tip of the blade occurred. This was expected, as the reduction of the thrust imposed during the optimization allows the geometry to present new configurations that conduce to the reduction of the loading noise. In this manner, the reduction of the tip radius uses the loosen thrust constraint as a manner to decrease the EAOASPL through the reduction of the loading noise, as presented in figure 4.7(b) where the loading noises for each case are presented. For the same thrust (imposed by the inferior limit made by the thrust constraint), a reduction of the radius at the tip of the blade leads to an increase in the loading noise, as the disk loading increases, as present in figure 4.4. The fact that the optimized blade with the largest radius at the tip of the blade obtained the lower loading noise responds accordingly to the previously stated hypothesis, as depicted in figure 4.7(b). The loading noise reduction was further influenced by other parameters, namely the increase of the chord, which also conduces to a decrease of the disk loading. Taking the previous reasoning into consideration, the variable that influenced this type of noise the most is the radius at the tip of the blade, with the optimizations with the greatest radius ( $n2k1$  and  $n5k2$ ) having obtained close values of loading noise. On the other hand, the blade with the lowest radius at the tip obtained the worst optimization in terms of this particular noise component. The loading noise is further influenced by the pitch distribution, as it influences spanwise inflow distribution, consequently affecting the aerodynamic forces [83]. Regarding this component of the optimization, an increase of its value is correlated to an increase of the loading noise, as depicted in figure 4.7(b), where the overall loading noise is proportional to the pitch values along the optimized blades [84], with the blade containing the highest pitch distribution ( $n6k2$ ) having the highest loading noise. Conversely, the blade with the lowest pitch distribution ( $n2k1$ ) possesses the overall lowest loading noise.

Thickness noise derives from the transverse periodic displacement of the air by the volume of a passing blade element, being its amplitude thus proportional to the blade volume. From figure 4.7(c), although every optimized blade suffers a reduction in its thickness noise, the range of such reductions appear to be correlated to the rank of the blade span reduction, with the blade with the greatest radius ( $n2k1$ ) at its tip being associated with the highest thickness noise value. Another important factor to take into consideration when examining the thickness noise is the chord since the boundary thickness parameters and the chord of the element are proportional amongst them, as demonstrated previously

Table 4.4: Optimization results for the radius at the tip of the blade.

Case	Rtip (m)
Basis blade	0.177775
<i>n2k1</i>	0.153152
<i>n5k1</i>	0.121984
<i>n6k2</i>	0.104620

in chapter 2. Thus, the thickness noise of the five point interpolation, that possesses the lower chord values across the blade span and the intermediate value for the tip radius, achieved the lower thickness noise values of the three. The addition of sweep to the blades also conduces to a decrease of the thickness noise since it decreases its effective velocity, leading to a decrease in the critical Mach number. Consequently, the difference between the thickness noise of *n5k1* and *n6k2* is further incremented by the fact that the sweep of the *n5k1* optimization is higher than the one of the *n6k2*. From the stated above, thin blade sections and planform sweep are used to control this type of noise [85]. It is also fundamental to mention that the sweep of the blade will be more important at higher Mach numbers, not being of significant influence for the noise production at low Mach number [85] (here the reader may consider low to moderate speeds the ones with a Mach number lower than 0.5), as in the current optimization process. The thickness and loading sources are linear and act on the blade surfaces; when flow over the blade sections reaches transonic speeds, nonlinear effects can become significant and might be modeled with quadrupole sources distributed in the volume surrounding the blades that can identify the viscous and propagation effects not encompassed by the thickness and loading sources, thus causing an increase of the noise for unswept, high-tip-speed rotors [85].

The broadband component of the noise for the studied blades is presented in figure 4.7(d). For its mitigation, the increase of twist (that is maximum for the 6 point discretization) leads to a shift of the aerodynamic loading inwards, implying the reduction of tip vortex strength [85], and consequently of the broadband noise generated from vortex roll-up. Thus, blade twist leads to a decrease of broadband noise.

An increase in blade incidence allows the rotor to generate the same level of thrust at a lower angular velocity and thus leads to a reduction of the thickness noise as there is less air displaced by a volume blade element per a certain period of time established. This is depicted in figure 4.6(a), as the blade with the lowest radius (*n6k2*) presents the highest twist in order for the rotor to respect the thrust constraint, since a larger radius leads to a higher rotational velocity near the blade tip and vice-versa.

The SPL spectra of the performed optimizations for a  $45^\circ$  and a  $90^\circ$  microphones at a  $90^\circ$  plane with the rotor's vertical axis and at 1.905 meters of distance from the rotor's hub is reached, being presented in figures 4.8(a) and 4.8(b), respectively. The values of SPL registered across the frequency spectrum for both a microphone located at  $-45^\circ$  and  $-90^\circ$  for the best optimized blade (*n2k1*) register a decrease of 10dB when comparing with the values obtained for the same conditions for the basis blade until a frequency of approximately 10Hz, where this décalage starts to decrease until a frequency of

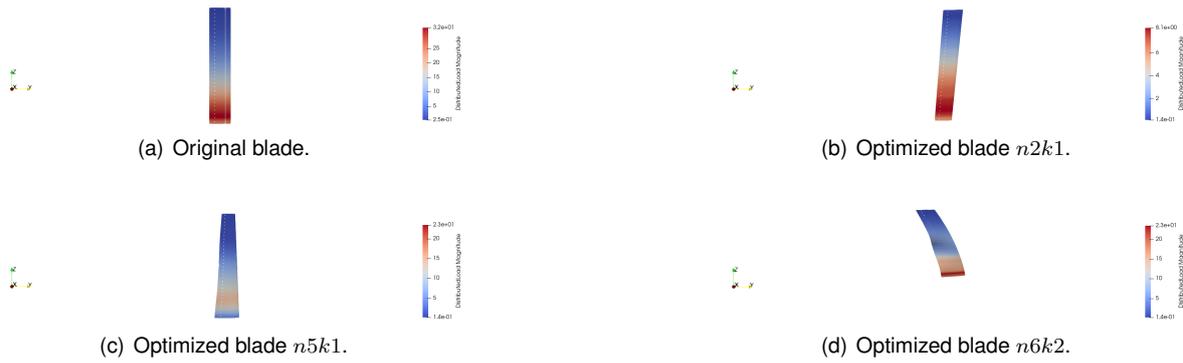


Figure 4.4: Distributed loads along the blade spans.

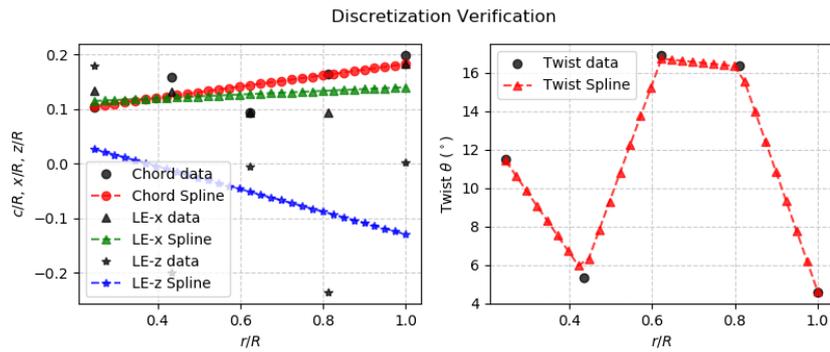
approximately 3000 Hz, where the SPL obtained for the optimized blade equals the SPL of the basis rotor. From this frequency until over 10000 Hz, an inversion of the curves occurs in both graphs, where the SPL of the optimized blade is superior to the one of the basis rotor. As of this last frequency forward, the SPL of the optimized blade is less than 5dB smaller than the original. Regarding the first and second BPFs, through the analysis of the results obtained for the best acoustically performing blade, a decrease of approximately 10dB can be observed in both. In this manner, there is only a small region of the spectra where the optimized blade performs worse than the basis blade, with a decrease of the SPL values being obtained across the frequency spectra. The spectra of the optimized blade seems as if translated to the right, which might be related to the fact that the optimized blade has a smaller diameter, when compared to the basis blade, making its wavelength of vibration also smaller and, consequently, its frequency higher.

When comparing the SPL captured for all the three optimizations, for a  $-45^\circ$  microphone the first BPF is well represented for all the optimizations, for the same frequency, as  $BPF = NB \times RPM/60$ . The second BPF is only represented within the optimizations for  $n2k1$ , as this optimization presents the greatest thickness noise, which is the greatest noise component at this microphone location. The higher frequency peaks are not captured by the software, capturing only the general trend of the SPL, as mentioned in the performed validation of the software in section 3.3.

At the  $-90^\circ$  microphone, the peaks corresponding to the BPFs are not captured because the broadband noise is higher than the frequency noise components in this direction, as previously stated in section 4.2.2.

For both microphones considered and for lower frequencies,  $n5k1$  and  $n6k2$  have higher SPL than the basis blade, which leads to the conclusion that  $n2k1$  is the best optimization. Despite of this outcome, all the three optimizations performed reduced the OASPL in all of the microphones simulated, as represented in figure 4.7(a), thus accomplishing the proposed optimization objective. Furthermore, bearing in mind the A-weighting process, the frequencies more annoying to humans (and, consequently, that are more important for the approval of an UAM vehicle, since its regulation measures noise using A-weighting, as explained in section 2.2) are indeed optimized for all the cases, as represented in figure 4.8. It is also noticeable that, amongst the three studied optimizations, the SPL curve of  $n2k1$  is the one that most closely follows the most the shape of the original curve (because they are geometrically more

similar in terms of the blade and of the blade's shape, as they both concern linear interpolations).



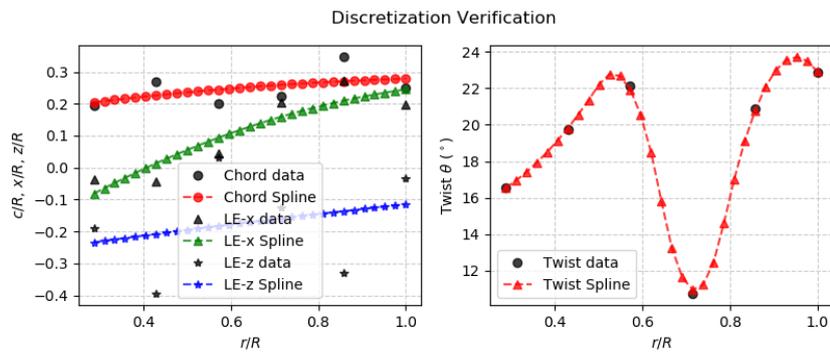
(a) Geometric discretization of the optimized rotor.



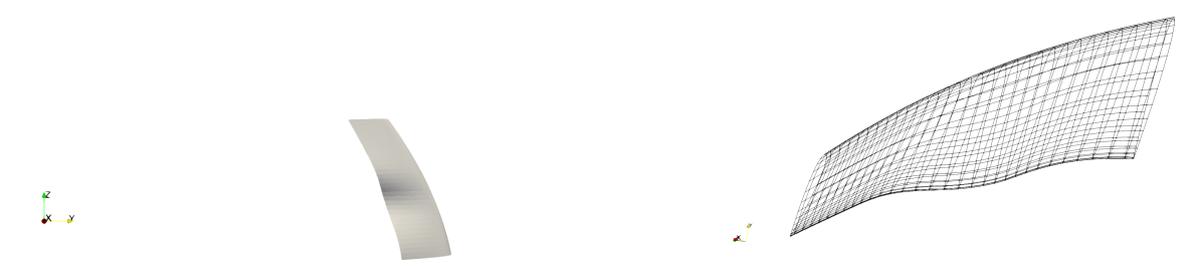
(b) 3D representation of the optimized rotor.

(c) Contour of the optimized rotor.

Figure 4.5: Geometry of the optimized two-rotor with an  $n5k1$  spline optimization.



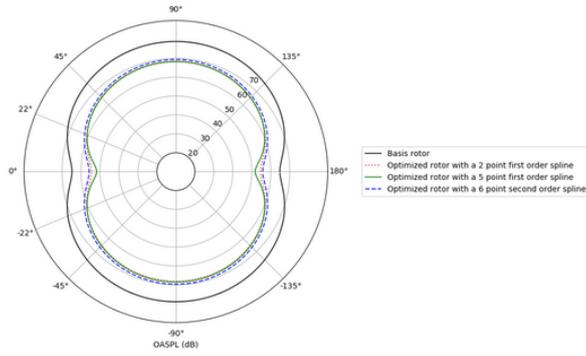
(a) Geometric discretization of the optimized two-bladed rotor.



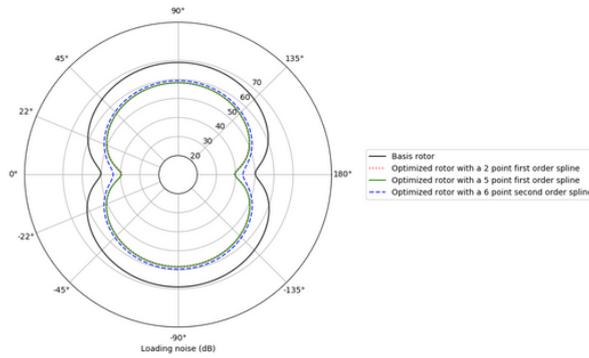
(b) 3D representation of the optimized rotor.

(c) Contour of the optimized rotor.

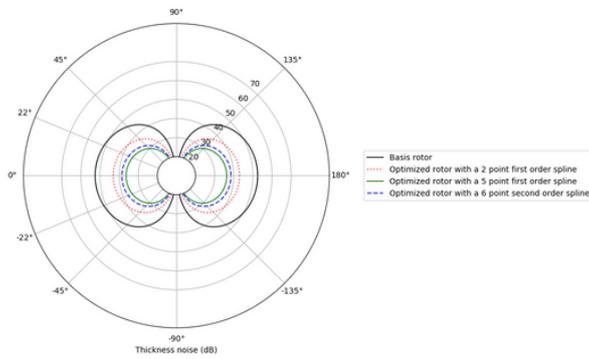
Figure 4.6: Geometry of the optimized two-bladed rotor with an  $n6k2$  spline optimization.



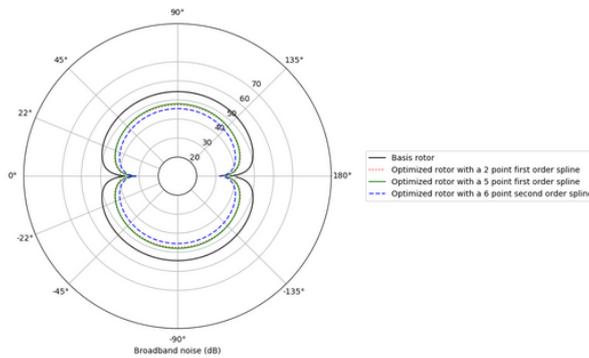
(a) OASPL noise (dB) of the optimized blades.



(b) Loading noise (dB) of the optimized blades.

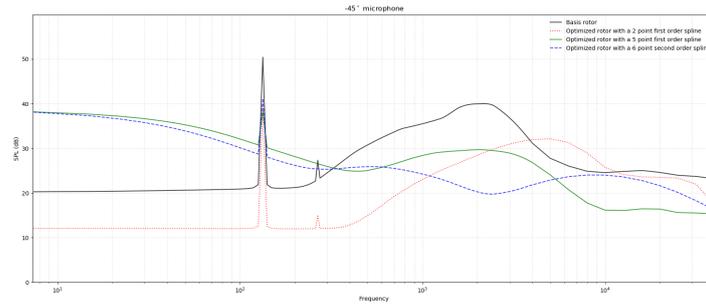


(c) Thickness noise (dB) of the optimized blades.

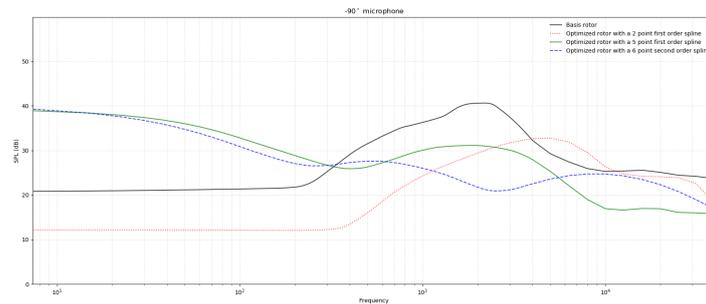


(d) Broadband noise (dB) of the optimized blades.

Figure 4.7: Noise results of the optimized blades.



(a) SPL Spectrum registered at a  $-45^\circ$  microphone for the optimized rotor.



(b) SPL Spectrum registered at a  $-90^\circ$  microphone for the optimized rotor.

Figure 4.8: SPL Spectrum.

### Effect of the number of individuals

The higher the number of individuals, the higher the variation within the population and the higher the number of new vectors introduced per iteration, providing more variability. Albeit of the positive consequences enumerated derived from a high number of individuals in the population, it also leads to an increase of computational effort. In order to study both of the enumerated effects, the optimization of the basis blade with a two point second order polynomial was studied for a population of 50, 100 and 150 individuals.

By observation of the results in table 4.5 and in figure 4.9, one can see that the variation of the population has no major impact on the case studied, which implies that the parameters chosen for the bimodal distributions for the setting of the next generation's  $F$  and  $CR$  values were well adjusted (see section 3.1.3), introducing enough variation into a 50-individual population for it to achieve results as good as the ones obtained with a 100 or 150 population. This leads to a more efficient optimization, saving computational effort, since smaller populations lead to shorter optimization times.

### Parametric study on the effect of the number of blades

A study on the effect of the number of blades on noise optimization was obtained through the optimization of two-blade, three-blade, and four-blade rotors for a  $n2k1$  spline along their span and the result

Table 4.5: Optimization results: study of the impact of the number of individuals on a two-bladed rotor optimization.

Case	Population	Total number of iterations	Best iteration	EAOASPL (dB)	Thrust (N)
Basis blade	-	-	-	66,08	3,62
$n2k1$	50	188	17	55.37	1.02
$n2k1$	100	163	19	55.48	2.82
$n2k1$	150	97	6	56.13	1.06

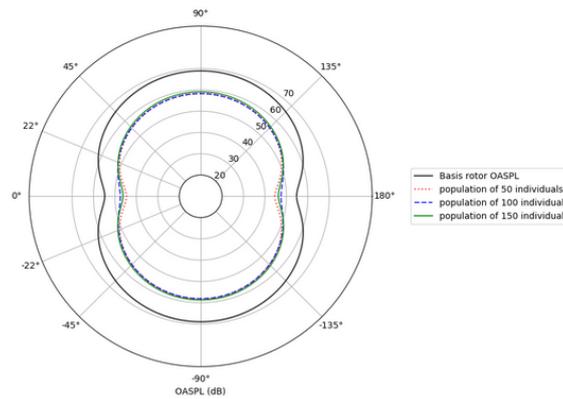


Figure 4.9: Comparison of the optimized OASPL for two-bladed rotor optimizations with a different number of individuals.

comparison amongst them.

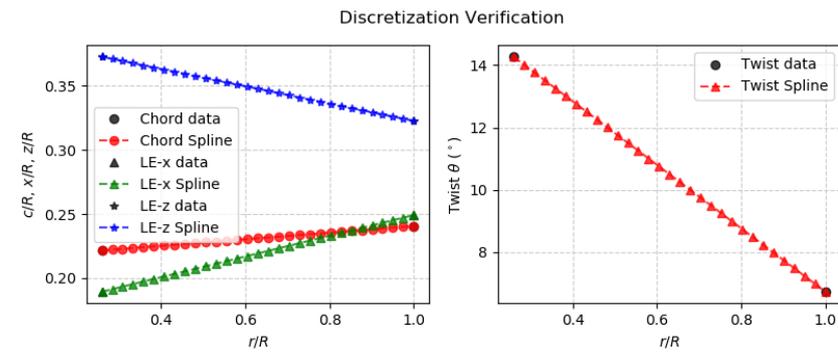
Similar EAOASPL's were obtained for the three studied cases, as presented in table 4.6, with a small reduction of OASPL for observers between  $20^\circ$  and  $-20^\circ$ , and between  $160^\circ$  and  $200^\circ$  for  $NB = 3$  and  $NB = 4$ , when comparing with  $NB = 2$ , as presented in figure 4.11(a). This is mainly due to a decrease in the thickness noise for the optimizations performed for higher number of blades, as depicted in figure 4.11(c). A secondary cause is the decrease in the loading noise in this direction, as presented in figure 4.11(b). This occurs despite the fact that the increase of the number of blades tends to lead to an increase of the thickness noise. In the presented case, this was compensated through the reduction of the chord, with the chord values being lower for higher numbers of blades, as shown in the comparison between figures 4.2(a), 4.10(a) and 4.10(b) (with the tip radius of the mentioned cases being present in table 4.6). The introduced chord reduction also leads to an increase of the loading noise; on the other hand, the increase of the number of blades leads to a decrease in disk loading and to an increase in the facility to comply with the thrust constraint, thus leading to a decrease in the loading noise levels that compensates the aforementioned increase. In this manner, the chord and the number of blades are highly correlated in the matters of noise mitigation and an optimal solution is found when a balance between these two variables is reached, with an optimal point for the chord being critical for obtaining the best acoustic solution, i.e., a chord geometry that manages to perfectly balance its opposite effects in terms of the loading and thickness noises.

Regarding the broadband component of the noise, increasing the number of blades reduces the

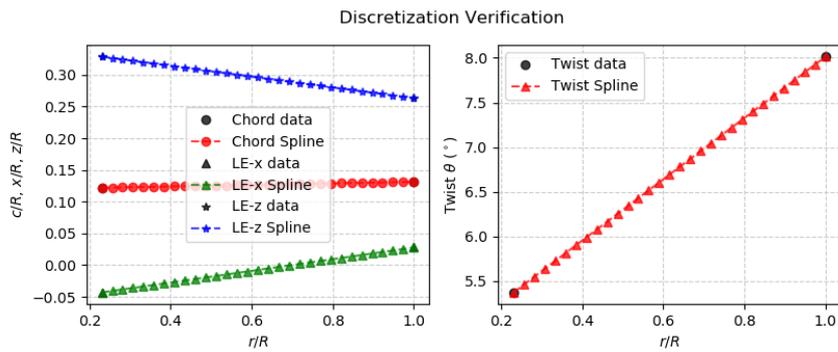
strength of each blade-tip vortex (for the same thrust). In the present case, regarding this noise component, presented in figure 4.11(d), the increase of the twist of the optimized rotors (presented in figures 4.2(a), 4.10(a), 4.10(b), with the twist of the basis rotor geometry being present in figure 4.1(a) as a term of comparison) lead to the reduction of the broadband noise generated from vortex roll-up, as explained previously in section 4.2.3.

Table 4.6: Optimization results: study of the effect of the number of blades on the rotor optimization process.

Case	Number of blades	Total number of iterations	Best iteration	Rtip (m)	EAOASPL (dB)	Thrust (N)
Basis blade	2	-	-	0.177775	66.08	3.62
$n2k1$	2	188	17	0.153152	55.37	1.02
$n2k1$	3	334	389	0.115822	55.06	1.02
$n2k1$	4	497	425	0.130287	55.32	1.05

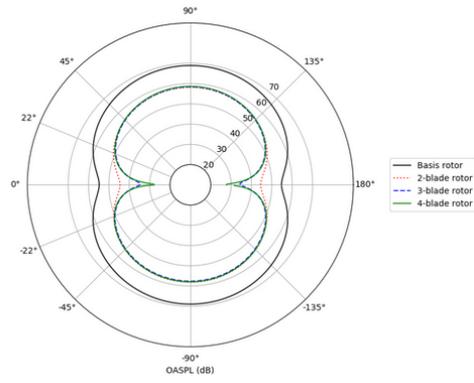


(a) Geometric discretization of the optimized two point blade first order spline three-blade rotor.

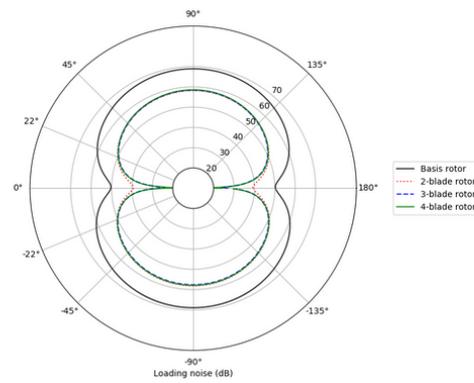


(b) Geometric discretization of the optimized two point blade first order spline four-blade rotor.

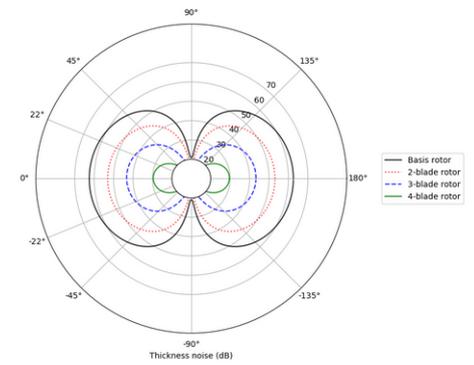
Figure 4.10: Geometry of a rotor with an  $n2k1$  spline optimization for a 3 and a 4-blade rotor.



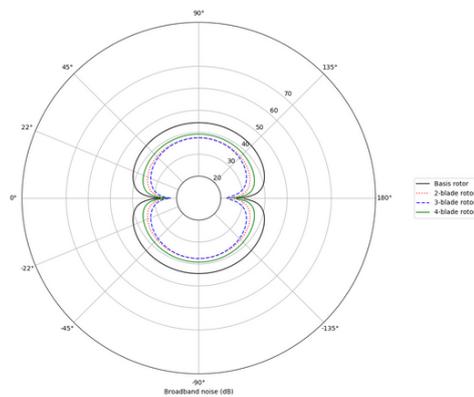
(a) OASPL (dB) of the optimized blades.



(b) Loading noise (dB) of the optimized blades.



(c) Thickness noise (dB) of the optimized blades.



(d) Broadband noise (dB) of the optimized blades.

Figure 4.11: Noise results for the parametric study of the effect of the number of blades.

# Chapter 5

## Conclusions

The current chapter presents a summary of the achievements accomplished during the elaboration of this thesis as well as suggestions that would further complement what has already been achieved.

### 5.1 Achievements

The main objective of this thesis was the creation of an optimization tool capable of reducing the noise produced by a rotor under hover conditions that responded to the growing of the UAM segment and to the increasingly demanding noise constraints, which consist in one of its main impediments. Thus, the major accomplishment obtained in the current thesis is the creation of an optimization framework with the aim of reducing the aeroacoustic noise produced by UAM rotors. Similar tools to the one developed must be used during the design process of the rotors for eVTOL in order to decrease the noise induced by the previous components whilst flying.

The FLOWUnsteady simulation software was adapted and successfully introduced into an optimization framework created for the current thesis, that allowed a study on the influence imposed by several geometric parameters of the blade on the output noise of a rotor during the hover segment of the flight envelope. The optimization study performed made use of a differential evolutionary algorithm and focused on the optimization of five design variables: radius at the tip of the blade, chord along the span of the blade, twist along the span of the blade, height of the LE from the top face of the hub along the span of the blade and sweep of the LE from the middle point of the hub along the span of the blade. These design variables were used for the creation of Bézier curves for the interpolation of the shape along the span of the blade. Several trends were observed within the diverse interpolation cases studied upon the optimization process. Within the mentioned trends, it should be noted that the increase of the tip radius (for the same thrust value), the increase of the chord, and the decrease of the pitch distributions lead to a decrease of the loading noise; regarding the broadband noise, increasing the twist leads to the decrease of the noise generated from vortex roll-up; finally, decreasing the blade volume leads to a decrease of the thickness noise. The optimization code was also incorporated in a parametric study of the number of blades regarding the importance of this parameter on noise efficacy and a parametric

study of the number of individuals in the population and its effect on the final EAOSPL obtained within the optimization scheme.

In such a manner, the optimization tool created successfully integrates an aeroacoustic simulator, providing a possible way to optimize a rotor geometry in an initial design phase in terms of the OASPL it generates. It consists in a versatile tool where several inputs can be changed by the observer in the *jupyter notebook* file created for the running of the optimization tool. The outputs of the optimization take the form of *CSV* formatted files (containing the geometric results of the performed optimization) that are able to be used as an input to the FLOWUnsteady software for further post-processing if wanted.

It should further be taken into consideration that the dimensions of the optimized and basis blade are smaller than the ones utilized in UAM both due to the fact that the verification performed for the simulator incorporated within the optimization tool supported blades of the same dimensions as the ones optimized during the current thesis, but also since the obtained blades have dimensions capable of being produced in a standard 3D printer, for a further development of the current work. Albeit of the previous, the geometrical conclusions of the present work are extendable to larger dimensions [85].

## 5.2 Future Work

The current section serves as a mean to provide suggestions that can be posteriorly implemented in order to create a more robust tool and to reach further developments.

In order to further test the optimization module implemented, the results obtained should be compared with experimental results, by comparing the OASPL of the optimized rotor in hover in an aeroacoustic wind tunnel setting with the predicted by the optimization module. As a means to see the scope of the different types of geometry optimization, the implementation of an airfoil optimization within the created module and the comparison of both the results obtained would consist in an interesting complement to the optimization already done. The integration of the optimized blade in an entire aircraft simulation within FLOWUnsteady is further encouraged to increase the applicability of the model in a real UAM vehicle. The creation of a pareto front model to observe the aerodynamic and aeroelastic consequences of the acoustic optimization would be of great accomplishment. One issue faced during the optimization process was the limitations of *xfoil* when provided with less conventional blades, such as NACA profiles, having difficulties with the boundary layer analysis and airfoils polar definition. A future adaptation of the code so that it does not give an error in the stated case, but instead penalizes the solutions that are not possible to compute would therefore increase the robustness of the code. In terms of computational efficiency, an improvement of the algorithm by using parallel computation would decrease the computational time of the optimization process. As the evaluation of every individual is done in an independent manner within the differential evolutionary algorithm, this would not be a difficult implementation. Lastly, the application of the current optimization module to blades of the dimension of the UAM market is encouraged, as both the simulation and the optimization code are capable to perform the simulation of larger blades, being recommended to firstly perform a validation study for these new scale in order to support the results obtained.

# Bibliography

- [1] P. D. Vascik. Systems analysis of urban air mobility operational scaling. Technical Report ICAT-2020-02, MIT International Center for Air Transportation, February 2020.
- [2] P. D. Vascik and R. J. Hansman. Scaling Constraints for Urban Air Mobility Operations: Air Traffic Control, Ground Infrastructure, and Noise. In *2018 Aviation Technology, Integration, and Operations Conference*, Atlanta, Georgia, June 2018. American Institute of Aeronautics and Astronautics. ISBN 9781624105562. doi: 10.2514/6.2018-3849. URL <https://arc.aiaa.org/doi/10.2514/6.2018-3849>. (Accessed on 30/06/2021).
- [3] The future of urban air mobility in europe. <https://siliconcanals.com/news/future-of-urban-air-mobility-in-europe/>, 2014-2020. (Accessed on 25/04/2021).
- [4] What is horizon 2020? <https://ec.europa.eu/programmes/horizon2020/en/what-horizon-2020,2020>. (Accessed on 25/04/2021).
- [5] Eu funds new wave of horizon 2020 urban air mobility research. <https://www.unmannedairspace.info/uncategorized/eu-funds-new-wave-horizon-2020-urban-air-mobility-research/>, 2020. (Accessed: 25/04/2021).
- [6] Flying forward 2020. <https://ec.europa.eu/programmes/horizon2020/en/what-horizon-2020,2020>. (Accessed on 25/04/2021).
- [7] SESAR Joint Undertaking | CORUS-XUAM - Concept of Operations for European U-space Services - Extension for Urban Air Mobility, 2021. URL <https://www.sesarju.eu/projects/CORUSXUAM>. (Accessed on 26/04/2021).
- [8] H. Eißfeldt. Sustainable Urban Air Mobility Supported with Participatory Noise Sensing. *Sustainability*, 12(8):3320, Apr. 2020. ISSN 2071-1050. doi: 10.3390/su12083320. URL <https://www.mdpi.com/2071-1050/12/8/3320>.
- [9] E. J. Alvarez and A. Ning. Unsteady mixed-fidelity aerodynamics solver for maneuvering multirotor aircraft. *SciTech*, 2021.
- [10] International Civil Aviation Organization. Unmanned aircraft systems Circular328AN/190, 2011.
- [11] E. Commission. Horizon 2020—work programme 2018–2020 part 11: Smart, green and integrated transport, September 2020.

- [12] H. Eißfeldt and V. Vogelpohl. Drone acceptance and noise concerns - some findings. In *International Symposium on Aviation Psychology*, pages 199–204, July 2019.
- [13] J. Wothge, C. Belke, U. Möhler, R. Guski, and D. Schreckenber. The combined effects of aircraft and road traffic noise and aircraft and railway noise on noise annoyance—an analysis in the context of the joint research initiative norah. *International Journal of Environmental Research and Public Health*, 14(1539), August 2017.
- [14] A. T. Martinez, R. Self, and J. Lawrence. Psycho acoustic characterisation of a small fixed-pitch quadcopter. In *Internoise*, 2019.
- [15] D. A. Senzig, M. Marsan, R. S. Downs, A. L. Hastings, C. J. Cutler, and R. W. Samiljan. UAS noise certification and measurements status report. Technical report, United States Department of Transportation, 2017.
- [16] International Civil Aviation Organization. A40-wp/292, urban air mobility, June 2019. URL [https://www.icao.int/Meetings/a40/Documents/WP/wp\\_292\\_en.pdf](https://www.icao.int/Meetings/a40/Documents/WP/wp_292_en.pdf). Agenda Item 26. (Accessed on 24/03/2021).
- [17] *Annex 16 to the Convention on International Civil Aviation — Environmental Protection*, volume 1 - Aircraft Noise, July 2017. International Civil Aviation Organization.
- [18] K. S. Brentner and F. Farassat. Modeling aerodynamically generated sound of helicopter rotors. *Progress in Aerospace Sciences*, 39(2-3):83–120, Feb. 2003. ISSN 03760421. doi: 10.1016/S0376-0421(02)00068-4. URL <https://linkinghub.elsevier.com/retrieve/pii/S0376042102000684>. (Accessed on 30/06/2021).
- [19] T. F. Brookes, D. S. Pope, and M. A. Marcolini. Airfoil self-noise and prediction. *NASA Reference Publication*, (1218), July 1989.
- [20] J. E. F. Williams and D. L. Hawkings. Sound generation by turbulence and surfaces in arbitrary motion. *Philosophical Transactions of the Royal Society of London*, 264(1151):321–342, May 1969.
- [21] F. Farassat. Derivation of formulations 1 and 1a of farassat. Technical Memorandum 214853, NASA, March 2007.
- [22] K. S. Brentner, C. L. Burley, and M. A. Marcolini. Sensitivity of Acoustic Predictions to Variation of Input Parameters. *Journal of the American Helicopter Society*, 39(3):43–52, July 1994. ISSN 2161-6027. doi: 10.4050/JAHS.39.43. URL <https://www.ingentaconnect.com/content/10.4050/JAHS.39.43>. (Accessed on 11/06/2021).
- [23] K. S. Brentner, G. A. Brès, G. Perez, and H. E. Jones. Maneuvering Rotorcraft Noise Prediction: A New Code For a New Problem. In *AHS Aerodynamics, Acoustics, and Test Evaluation Specialist Meeting*, San Francisco, CA, January 2002.
- [24] P. Bortolotti, E. Branlard, A. Platt, P. J. Moriarty, C. Sucameli, and C. L. Bottasso. Aeroacoustics noise model of openfast. Technical report, National Renewable Energy Laboratory, August 2020.

- [25] P. Moriarty and P. Migliore. Semi-empirical aeroacoustic noise prediction code for wind turbines. Technical Report TP-500-34478, National Renewable Energy Laboratory, December 2003.
- [26] F. Afonso, A. Ferreira, I. Ribeiro, F. Lau, and A. Suleman. On the design of environmentally sustainable aircraft for urban air mobility. *Transportation Research Part D: Transport and Environment*, 91, Feb. 2021. ISSN 13619209. doi: 10.1016/j.trd.2020.102688. URL <https://linkinghub.elsevier.com/retrieve/pii/S1361920920308725>. 2021-06-30.
- [27] A. Brown and W. Harris. A Vehicle Design and Optimization Model for On-Demand Aviation. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Kissimmee, Florida, Jan. 2018. American Institute of Aeronautics and Astronautics. ISBN 9781624105326. doi: 10.2514/6.2018-0105. URL <https://arc.aiaa.org/doi/10.2514/6.2018-0105>. (Accessed on 26/06/2021).
- [28] L. Gutin. On a sound field of a rotating propeller. Tech. rep. tm 1195, NACA, 1948.
- [29] A. F. Deming. Propeller Rotation Noise Due to Torque and Thrust. *The Journal of the Acoustical Society of America*, 12(1), jul 1940. ISSN 0001-4966. doi: 10.1121/1.1916089. URL <https://asa.scitation.org/doi/10.1121/1.1916089>. (Accessed on 2021-06-26).
- [30] R. Schlegel, R. King, and H. Mull. HELICOPTER ROTOR NOISE GENERATION AND PROPAGATION. Technical report, UNITED TECHNOLOGIES CORP STRATFORD CT SIKORSKY AIRCRAFT DIV, oct 1966. URL <https://apps.dtic.mil/sti/citations/AD0645884>. (Accessed on 2021-06-26).
- [31] J. J. Bertin and R. M. Cummings. *Aerodynamics for engineers*. Pearson, Boston, sixth edition edition, 2014. ISBN 9780132832885.
- [32] S. Pinzón. Introduction to vortex lattice theory. *Ciencia y Poder Aéreo*, 10(1):39–48, 2015. doi: <http://dx.doi.org/10.18667/cienciaypoderaereo.433>.
- [33] T. Burton, N. Jenkins, D. Sharpe, and E. Bossanyi. *Wind Energy Handbook: Burton/Wind Energy Handbook*. John Wiley & Sons, Ltd, Chichester, UK, May 2011. ISBN 9781119992714 9780470699751. doi: 10.1002/9781119992714. URL <http://doi.wiley.com/10.1002/9781119992714>.
- [34] M. O. L. Hansen. *Aerodynamics of wind turbines*. Earthscan, London ; Sterling, VA, 2nd ed edition, 2008. ISBN 9781844074389. OCLC: ocm86172940.
- [35] C. N. H. Lock, H. Bateman, and H. C. H. Townend. *Experiments to verify the independence of the elements of an airscrew blade*. HMSO, 1925.
- [36] H. Glauert. *The analysis of experimental results in the windmill brake and vortex ring states of an airscrew*. HMSO, London, 1926.
- [37] M. Buhl. New Empirical Relationship between Thrust Coefficient and Induction Factor for the Turbulent Windmill State. 2005.

- [38] P. J. Moriarty and A. C. Hansen. AeroDyn theory manual. 1 2005. doi: 10.2172/15014831. URL <https://www.osti.gov/biblio/15014831>.
- [39] H. Lee and D.-J. Lee. Rotor interactional effects on aerodynamic and noise characteristics of a small multirotor unmanned aerial vehicle. *Physics of Fluids*, 32(4):047107, Apr. 2020. ISSN 1070-6631, 1089-7666. doi: 10.1063/5.0003992. URL <http://aip.scitation.org/doi/10.1063/5.0003992>. (Accessed on 2021-06-11).
- [40] D. Shukla and N. Komerath. Multirotor Drone Aerodynamic Interaction Investigation. *Drones*, 2(4): 43, Dec. 2018. ISSN 2504-446X. doi: 10.3390/drones2040043. URL <http://www.mdpi.com/2504-446X/2/4/43>. (Accessed on 14/05/2021).
- [41] byuflowlab FLOWUnsteady, 2021. URL <https://github.com/byuflowlab/FLOWUnsteady>. (Accessed on 20/09/2021).
- [42] Flowunsteady. <https://flow.byu.edu/FLOWUnsteady/>. (Accessed on 25/03/2021).
- [43] Julia Documentation · The Julia Language. <https://docs.julialang.org/en/v1/>. (Accessed on 25/03/2021).
- [44] K. Brentner. Psu-wopwop computational aeroacoustic analysis, 2021. URL <https://sites.psu.edu/wopwop/>. (Accessed on 02/10/2021).
- [45] U. Saetti. Rotorcraft simulations with coupled flight dynamics, free wake, and acoustics. Master's thesis, The Pennsylvania State University The Graduate School College of Engineering, 2016.
- [46] E. Alvarez, A. Schenk, T. Critchfield, and A. Ning. Rotor-on-rotor aeroacoustic interactions of multirotor in hover. 2020.
- [47] byuflowlab BPM, 2021. URL <https://github.com/byuflowlab/BPM.jl>. (Accessed on 20/09/2021).
- [48] byuflowlab FLOWVLM, 2021. URL <https://github.com/byuflowlab/FLOWVLM>. (Accessed on 20/09/2021).
- [49] A. Ning. Using blade element momentum methods with gradient-based design optimization. *Structural and Multidisciplinary Optimization*, 64(2):991–1014, Aug. 2021. ISSN 1615-147X, 1615-1488. doi: 10.1007/s00158-021-02883-6. URL <https://link.springer.com/10.1007/s00158-021-02883-6>. (Accessed on 2021-09-20).
- [50] byuflowlab CCBlade, 2021. URL <https://github.com/byuflowlab/CCBlade.jl>. (Accessed on 20/09/2021).
- [51] E. J. Alvarez. Quasi-steady aerodynamics solver for a high-fidelity controls framework. *FLOWUnsteady Documentations*, 2020.
- [52] Flowunsteady github, 2021. URL <https://github.com/byuflowlab/FLOWUnsteady>. (Accessed on 8/04/2021).

- [53] byuflowlab FLOWUnsteady. URL <https://github.com/byuflowlab/FLOWUnsteady/blob/d302a74ec6b8015485fc24701126461d4357278f/examples/rotonoise/singlerotor.ipynb>. (Accessed on 25/03/2021).
- [54] G. Bossotto. Urban air vehicle implementation and limitations - development of a new acoustic tool for noise prediction. Thesis to obtain the master of science degree in aerospace engineering, Instituto Superior Técnico de Lisboa, October 2021.
- [55] J. C. Lopez. Quadcopter simulation. URL <https://engineering.usu.edu/students/clubs/shpe/quadcopter-simulation>. (Accessed on 25/04/2021).
- [56] J. Yoon and J. Lee. Altitude and roll control of a hovering quad-rotor air vehicle using the multi-objective approximate optimization of proportional–integral–differential control. *Engineering Optimization*, 49(10):1704–1718, Oct. 2017. ISSN 0305-215X, 1029-0273. doi: 10.1080/0305215X.2016.1265201. URL <https://www.tandfonline.com/doi/full/10.1080/0305215X.2016.1265201>. (Accessed on 2021-11-07).
- [57] A. Marta. Multidisciplinary design optimization of aircrafts. Class notes, Instituto Superior Técnico, 2021.
- [58] S. S. Rodrigues. Aeroacoustic optimization of wind turbine blades, November 2012.
- [59] J. Martins and A. Ning. *Engineering Design Optimization*. 10 2021. ISBN 9781108833417. doi: 10.1017/9781108980647.
- [60] J. M. R. Branco. Aeroacoustic Optimization of a Low-Speed Fan, June 2015.
- [61] J. Zhang and A. C. Sanderson. *Adaptive Differential Evolution A Robust Approach to Multimodal Problem Optimization*, volume 1. 2009. doi: 10.1007/9783642015274.
- [62] L. Rios and N. Sahinidis. Derivative-free optimization: A review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56, 11 2009. doi: 10.1007/s10898-012-9951-y.
- [63] L. Chambers, editor. *The practical handbook of genetic algorithms: applications*. Chapman & Hall/CRC, Boca Raton, Fla, 2nd ed edition, 2001. ISBN 9781584882404.
- [64] T. Back and H.-P. Schwefel. Evolutionary computation: an overview. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 20–29, Nagoya, Japan, 1996. IEEE. ISBN 9780780329027. doi: 10.1109/ICEC.1996.542329. URL <http://ieeexplore.ieee.org/document/542329/>. (Accessed on 2021-09-16).
- [65] B. Babu and M. Jehan. Differential evolution for multi-objective optimization. In *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, volume 4, pages 2696–2703 Vol.4, 2003. doi: 10.1109/CEC.2003.1299429.

- [66] K. V. Price, R. M. Storn, and J. A. Lampinen. *Differential evolution: a practical approach to global optimization*. Natural computing series. Springer, Berlin ; New York, 2005. ISBN 9783540209508.
- [67] R. Storn and K. Price. Differential evolution: A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces. *Journal of Global Optimization*, 23, 01 1995.
- [68] A. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124–141, July 1999. ISSN 1089778X. doi: 10.1109/4235.771166. URL <http://ieeexplore.ieee.org/document/771166/>. (Accessed on 2021-09-17).
- [69] N. M. Patrikalakis and T. Maekawa. *Shape interrogation for computer aided design and manufacturing*. Springer, Heidelberg ; New York, 2010. ISBN 9783642040733. OCLC: ocn462918509.
- [70] J. Schiach. Lecture notes in bézier curves, March 2015. URL <https://www.youtube.com/watch?v=2HvH9cmHbG4>. (Accessed on 30/09/2021).
- [71] J. Schiach. Lecture notes in b-splices, March 2015. URL <https://www.youtube.com/watch?v=qhQrRCJ-mVg>. (Accessed on 30/09/2021).
- [72] W. T. Scarbrough. Naca four-digit airfoil section generation using cubic parametric curve segments and the golden section curve segments and the golden section. Thesis to obtain the master of science degree in mechanical engineering, Department of Mechanical Engineering, Rochester Institute of Technology, April 1992.
- [73] F. Pérez-Arribas and R. Pérez-Fernández. A B-spline design model for propeller blades. *Advances in Engineering Software*, 118:35–44, Apr. 2018. ISSN 09659978. doi: 10.1016/j.advengsoft.2018.01.005. URL <https://linkinghub.elsevier.com/retrieve/pii/S0965997817311134>. (Accessed on 01/10/2021).
- [74] P. Dierckx. *Curve and surface fitting with splines*. Monographs on Numerical Analysis. Clarendon, Oxford, repr edition, 1996. ISBN 9780198534402.
- [75] package Dierckx, 2021. URL <https://github.com/kbarbary/Dierckx.jl>. (Accessed on 29/09/2021).
- [76] N. D. Polyzos, S. Vouros, I. Goulos, and V. Pachidis. Multi-disciplinary optimization of variable rotor speed and active blade twist rotorcraft: Trade-off between noise and emissions. *Aerospace Science and Technology*, 107:106356, Dec. 2020. ISSN 12709638. doi: 10.1016/j.ast.2020.106356. URL <https://linkinghub.elsevier.com/retrieve/pii/S1270963820310385>. (Accessed on 02/10/2021).
- [77] Blackboxoptim, 2021. URL <https://github.com/robertfeldt/BlackBoxOptim.jl>. (Accessed on 02/10/2021).

- [78] Y. Wang, H.-X. Li, T. Huang, and L. Li. Differential evolution based on covariance matrix learning and bimodal distribution parameter setting. *Applied Soft Computing*, 18:232–247, May 2014. ISSN 15684946. doi: 10.1016/j.asoc.2014.01.038. URL <https://linkinghub.elsevier.com/retrieve/pii/S1568494614000581>. (Accessed on 26/12/2021).
- [79] Jingqiao Zhang and A. Sanderson. JADE: Adaptive Differential Evolution With Optional External Archive. *IEEE Transactions on Evolutionary Computation*, 13(5):945–958, Oct. 2009. ISSN 1941-0026, 1089-778X. doi: 10.1109/TEVC.2009.2014613. URL <http://ieeexplore.ieee.org/document/5208221/>. (Accessed on 26/12/2021).
- [80] Dji - official website, 2021. URL <https://www.dji.com/pt>. (Accessed on 04/10/2021).
- [81] C. Thurman and N. Zawodny. Aeroacoustic characterization of optimum hovering rotors using artificial neural networks. 05 2021.
- [82] J. Brest, V. Zumer, and M. Maucec. Self-Adaptive Differential Evolution Algorithm in Constrained Real-Parameter Optimization. In *2006 IEEE International Conference on Evolutionary Computation*, pages 215–222, Vancouver, BC, Canada, 2006. IEEE. ISBN 9780780394872. doi: 10.1109/CEC.2006.1688311. URL <http://ieeexplore.ieee.org/document/1688311/>. (Accessed on 21/09/2021).
- [83] D. Weitsman and E. Greenwood. Parametric Study of eVTOL Rotor Acoustic Design Trades. In *AIAA Scitech 2021 Forum*, VIRTUAL EVENT, Jan. 2021. American Institute of Aeronautics and Astronautics. ISBN 9781624106095. doi: 10.2514/6.2021-1987. URL <https://arc.aiaa.org/doi/10.2514/6.2021-1987>. (Accessed on 06/01/2022).
- [84] Thickness and Loading Noise from Helicopter Rotor at various Pitch Angles. *Journal of the Korean Society for Aeronautical & Space Sciences*, 35(10):868–874, Oct. 2007. ISSN 1225-1348. doi: 10.5139/JKSAS.2007.35.10.868. URL <http://koreascience.or.kr/journal/view.jsp?kj=HGJHBY&py=2007&vnc=v35n10&sp=868>. (Accessed on 06/01/2022).
- [85] H. H. Hubbard and Acoustical Society of America, editors. *Aeroacoustics of flight vehicles: theory and practice*. Published for the Acoustical Society of America through the American Institute of Physics, Woodbury, NY, 1995. ISBN 9781563964077 9781563964046 9781563964060.
- [86] Transport up. <https://transportup.com/>, 2020. (Accessed on 12/04/2021).

# Appendix A

## Market Study of UAM aircraft

Table A.1: Market Study of UAM aircraft, adapted from [86].

Model	Aircraft Type	Powerplant	Range	Top Speed	Propeller Configuration	Passenger Capacity	Autonomy Level	Stage of Development
Airbus City Airbus	Wingless eVTOL	All-electric	60 miles	75 mph	8 lift/thrust rotors (4 dual rotors)	4 passengers	Autonomous	Flight Testing
Boeing PAV	Winged VTOL	All-electric	50 miles	Undisclosed	8 lifting rotors, 1 pusher propeller	2 passengers	Autonomous	Flight Testing
Bell Nexus Air Taxi	Winged VTOL	6HX - Hybrid-electric; 4EX - All-electric	6HX - 150 miles; 4EX - 60 miles	150 mph	6HX - 6 tilt-rotors; 4EX - 4 tilt-rotors	1 pilot, 4 passengers	Semi-Autonomous	Prototype Build
WISK Cora	Winged VTOL	All-electric	25 miles	100 mph	6 lift propellers under each wing (non-tilting), 1 rear pusher-propeller for forward thrust.	2 passengers	Autonomous	Flight Testing
EHang 2016	Wingless VTOL	All-Electric	22 miles	80 mph	16 lift/thrust rotors (8 dual rotors)	2 Passengers, or 485 lbs total payload	Autonomous	Certification
Joby Aviation S4	Winged VTOL	Distributed Electric Propulsion (DEP) System powered by lithium-nickel-cobalt-manganese-oxide batteries	150 miles	200 mph	4 propellers tilt vertically including its entire motor nacelle, and 2 of the propellers tilt vertically with a linkage mechanism	1 pilot and 4 passengers	Piloted	Flight Testing
Vertical Aerospace4 VA-1X	Winged VTOL	All-electric	100 miles	150 mph	8 lift rotors, 4 tilting rotors (front)	1 pilot, 4 passengers, 1000 lbs	Piloted with autonomy, eventual full-autonomy	Prototype Build
Karem Butterfly	Winged VTOL	All-electric	100 miles	150-200 mph	Quad tiltrotor with Optimum Speed Tiltrotor (OSTR) technology; large rotor, slow turning propeller.	1 pilot, 4 passengers	Piloted	Preliminary Design
Beta Technologies Alta	Winged VTOL	All-electric	2288 miles	167 mph	4 vertical lift propellers, one rear pusher propeller	6 seats	Semi-autonomous with eventual full-autonomy	Flight Testing

## Appendix B

# Dynamic Movement Code

```
module Movement
import Dierckx
import CSV
import DataFrames
import JLD
import Dates
using PyPlot
using LinearAlgebra: cross, I

# ----- FLOW CODES -----
# FLOWVLM https://github.com/byuflowlab/FLOWVLM
import FLOWVLM
const vlm = FLOWVLM

# FLOWVPM https://github.com/byuflowlab/FLOWVPM.jl
try # Load FLOWVPM if available
    import FLOWVPM
catch e # Otherwise load a dummy version of FLOWVPM
    @warn("FLOWVPM module not found. Using dummy module instead.")
    include("FLOWUnsteady_dummy_FLOWVPM.jl")
end
const vpm = FLOWVPM

# GeometricTools https://github.com/byuflowlab/GeometricTools.jl
import GeometricTools
const gt = GeometricTools

import FLOWNoise
const noise = FLOWNoise

# BPM https://github.com/byuflowlab/BPM.jl
import BPM

# ----- GLOBAL VARIABLES -----
const module_path = splitdir(@__FILE__)[1] # Path to this module
const def_data_path = joinpath(module_path, "../data/") # Default path
# to data folder
```

```

# ----- HEADERS -----
include("FLOWUnsteady_vehicle_vlm.jl")
include("FLOWUnsteady_maneuver.jl")
include("FLOWUnsteady_rotor.jl")
include("FLOWUnsteady_simulation_types.jl")
include("FLOWUnsteady_simulation.jl")
include("FLOWUnsteady_utils.jl")
include("FLOWUnsteady_processing.jl")
include("FLOWUnsteady_monitors.jl")
include("FLOWUnsteady_noise_wopwop.jl")
include("FLOWUnsteady_noise_bpm.jl")

using LinearAlgebra: norm, cross
using Plots

"""
    initializationMovementMatrix(nsteps)

Initiates the Movement Matrix at 0
"""

function initializationMovementMatrix(nsteps)
    row = 22
    MovementMatrix = zeros((row,nsteps))
    return MovementMatrix
end

"""
    equations(dt, self, t, nt)

Called upon the simulation cycle
Compute the positions, velocities and accelerations
return F, M
"""
function equations(dt, self, t, nt)
    self.a = acceleration(self)
    I = [self.mp.Ixx  0.0  0.0;
         0.0  self.mp.Iyy  0.0;
         0.0  0.0  self.mp.Izz]
    T, self.tau = thrusttorque(self)
    self.omegadot = inv(I) * (self.tau - cross(self.omegaprev, I *
        self.omegaprev))
    self.omega = self.omegaprev + dt * self.omegadot
    self.thetadot = omega2thetadot(self)
    self.theta = self.thetaprev + dt * self.thetadot
    self.xdot = self.xdotprev + dt * self.a
    self.x = self.xprev + dt * self.xdot

    global self.MovementMatrix[1, nt+1] = self.x[1]
    global self.MovementMatrix[2, nt+1] = self.x[2]
    global self.MovementMatrix[3, nt+1] = self.x[3]
    global self.MovementMatrix[4, nt+1] = self.xdot[1]
    global self.MovementMatrix[5, nt+1] = self.xdot[2]

```

```

global self.MovementMatrix[6, nt+1] = self.xdot[3]
global self.MovementMatrix[7, nt+1] = self.theta[1]
global self.MovementMatrix[8, nt+1] = self.theta[2]
global self.MovementMatrix[9, nt+1] = self.theta[3]
global self.MovementMatrix[10, nt+1] = self.thetadot[1]
global self.MovementMatrix[11, nt+1] = self.thetadot[2]
global self.MovementMatrix[12, nt+1] = self.thetadot[3]
global self.MovementMatrix[13, nt+1] = self.thetadot[3]
global self.MovementMatrix[14, nt+1] = self.omega[1]
global self.MovementMatrix[15, nt+1] = self.omega[2]
global self.MovementMatrix[16, nt+1] = self.omega[3]
global self.MovementMatrix[17, nt+1] = self.omegadot[1]
global self.MovementMatrix[18, nt+1] = self.omegadot[2]
global self.MovementMatrix[19, nt+1] = self.omegadot[3]
global self.MovementMatrix[20, nt+1] = self.a[1]
global self.MovementMatrix[21, nt+1] = self.a[2]
global self.MovementMatrix[22, nt+1] = self.a[3]

return self.a , self.omegadot, self.omega,
self.thetadot, self.theta, self.xdot, self.x, self.tau,
self.MovementMatrix

end

"""
    inertialtobody(theta)

Construct a rotation matrix from inertial frame to body frame

The assumed order of rotation is
1) psi radians about the z axis,
2) theta radians about the y axis,
3) phi radians about the x axis.

This is an orthogonal transformation so its inverse is its transpose.
"""

function inertialtobody(theta)

    R = Array{eltype(theta)}(undef,3, 3)

    cphi, ctht, cpsi = cos.([theta[1], theta[2], theta[3]])
    sphi, stht, spsi = sin.([theta[1], theta[2], theta[3]])

    R[1, 1] = ctht*cpsi
    R[1, 2] = ctht*spsi
    R[1, 3] = -stht

    R[2, 1] = sphi*stht*cpsi - cphi*spsi
    R[2, 2] = sphi*stht*spsi + cphi*cpsi
    R[2, 3] = sphi*ctht

    R[3, 1] = cphi*stht*cpsi + sphi*spsi
    R[3, 2] = cphi*stht*spsi - sphi*cpsi
    R[3, 3] = cphi*ctht

```

```

    return R

end

"""
    bodytoinertial(theta)

Construct a rotation matrix from body frame to inertial frame
"""
function bodytoinertial(theta)

    Ri = Array{eltype(theta)}(undef,3, 3)

    cphi, ctht, cpsi = cos.([theta[1], theta[2], theta[3]])
    sphi, stht, spsi = sin.([theta[1], theta[2], theta[3]])

    Ri[1, 1] = cphi*cpsi - ctht*sphi*spsi
    Ri[1, 2] = -cpsi*sphi - cphi*ctht*spsi
    Ri[1, 3] = stht*spsi
    Ri[2, 1] = ctht*sphi+cpsi + cphi*spsi
    Ri[2, 2] = cphi*ctht+cpsi - sphi*spsi
    Ri[2, 3] = -cpsi*stht

    Ri[3, 1] = sphi*stht
    Ri[3, 2] = cphi*stht
    Ri[3, 3] = ctht

    return Ri

end

"""
    thrusttorque(self)

Computes a vector of Thrust and Torque for the rotors
"""
function thrusttorque(self)
    T = Real
    Q = Real

    sum=0
    i=1
    for i in 1:4
        sum = sum + self.rpm[i]
        i=1+i
    end
    T = [0.0; 0.0; self.k*sum]
    Q = [self.l*self.k*(self.rpm[1]-self.rpm[3]);
        self.l*self.k*(self.rpm[2]-self.rpm[4]);
        self.b*(self.rpm[1]-self.rpm[2]+self.rpm[3]-self.rpm[4])]

    return T, Q
end

```

```

"""
    acceleration(self)

Computation of the linear acceleration
"""
function acceleration(self)
    gravity= [0.0; 0.0; -self.atm.g]
    R = bodytoinertial(self.theta)
    Tb, q = thrusttorque(self)
    Ti = R * Tb
    Fd = -self.kd * self.xdot
    a = gravity + 1/self.mp.m * Ti + 1/self.mp.m * Fd
    return a
end

"""
    omega2thetadot(self)

Computation of thetadot
"""
function omega2thetadot(self)
    ct, cp = cos.([self.theta[2], self.theta[1]])
    st, sp = sin.([self.theta[2], self.theta[1]])

    C = [1.0  0.0  -st;
         0.0  cp  ct*sp;
         0.0  -sp  ct*cp]

    thetadot = inv(C)*self.omega
    return thetadot
end

end # module

```

# Appendix C

## Geometry CSV files

### C.1 Basis Rotor

Table C.1: Radius of the tip, radius of the hub and number of blades.

<b>property</b>	<b>file</b>	<b>description</b>
Rtip	0.17775	(m) Radius of blade tip
Rhub	0.03	(m) Radius of hub
B	2	Number of blades
blade	blade.csv	Blade file

Table C.2: Airfoil geometry CSV file.

<b>r/R</b>	<b>Contour file</b>	<b>Aero file</b>
0.0	naca0012.csv	naca0012polar.csv
1.0	naca0012.csv	naca0012polar.csv

Table C.3: Blade file.

<b>property</b>	<b>file</b>	<b>description</b>
chorddist	chorddist.csv	Chord distribution
pitchdist	pitchdist.csv	Pitch distribution
sweepdist	sweepdist.csv	LE sweep distribution
heightdist	heightdist.csv	LE height distribution
airfoil_files	airfoils.csv	Airfoil distribution
spl_k	2	Spline order
spl_s	1.0e-2	Spline smoothing

Table C.4: Spanwise CSV of the chord distribution.

<b>r/R</b>	<b>c/R</b>
0.0843881856540084	0.157637130801688
0.225035161744022	0.157637130801688
0.421940928270042	0.157637130801688
0.70323488045007	0.157637130801688
0.945147679324895	0.157637130801688
1	0.157637130801688

Table C.5: Spanwise CSV of the height distribution.

<b>r/R</b>	<b>z/R (height of leading edge from top face of hub)</b>
0.0843881856540084	-0.0109704641350211
0.225035161744022	-0.0109704641350211
0.421940928270042	-0.0109704641350211
0.70323488045007	-0.0109704641350211
0.945147679324895	-0.0109704641350211
1	-0.0109704641350211

Table C.6: Spanwise CSV of the pitch distribution.

<b>r/R</b>	<b>twist (deg)</b>
0.0843881856540084	8.32
0.225035161744022	8.32
0.421940928270042	8.32
0.70323488045007	8.32
0.945147679324895	8.32
1	8.32

Table C.7: Spanwise CSV of the sweep distribution.

<b>r/R</b>	<b>y/R (y-distance of LE from the middle point of hub)</b>
0.0843881856540084	0.0767369901547117
0.225035161744022	0.0767369901547117
0.421940928270042	0.0767369901547117
0.70323488045007	0.0767369901547117
0.945147679324895	0.0767369901547117
1	0.0767369901547117

## C.2 First order two point interpolation

Table C.8: Radius of the tip, radius of the hub and number of blades.

property	file	description
Rtip	0.15315245114409806	(m) Radius of blade tip
Rhub	0.03	(m) Radius of hub
B	2	Number of blades
blade	pedro_current_blade.csv	Blade file

Table C.9: Blade file.

property	file	description
chorddist	current_chorddist.csv	Chord distribution
pitchdist	current_pitchdist.csv	Pitch distribution
sweepdist	current_sweepdist.csv	LE sweep distribution
heightdist	current_heightdist.csv	LE height distribution
airfoil_files	airfoils.csv	Airfoil distribution
spl_k	1	Spline order
spl_s	1.0e-1	Spline smoothing

Table C.10: Spanwise CSV of the chord distribution.

r/R	c/R
0.1958832508124444	0.15403024667812548
1.0	0.12463441464272901

Table C.11: Spanwise CSV of the height distribution.

r/R	z/R (height of leading edge from top face of hub)
0.1958832508124444	0.13465687874982904
1.0	-0.0619113042631163

Table C.12: Spanwise CSV of the pitch distribution.

r/R	twist (deg)
0.1958832508124444	9.470954215178935
1.0	4.162809594708207

Table C.13: Spanwise CSV of the sweep distribution.

<b>r/R</b>	<b>y/R (y-distance of LE from the middle point of hub)</b>
0.1958832508124444	0.15925740611943182
1.0	0.06889361962198068

### C.3 First order five point interpolation

Table C.14: Radius of the tip, radius of the hub and number of blades.

<b>property</b>	<b>file</b>	<b>description</b>
Rtip	0.12198389161518143	(m) Radius of blade tip
Rhub	0.03	(m) Radius of hub
B	2	Number of blades
blade	pedro_current_blade.csv	Blade file

Table C.15: Blade file.

<b>property</b>	<b>file</b>	<b>description</b>
chorddist	current_chorddist.csv	Chord distribution
pitchdist	current_pitchdist.csv	Pitch distribution
sweepdist	current_sweepdist.csv	LE sweep distribution
heightdist	current_heightdist.csv	LE height distribution
airfoil_files	airfoils.csv	Airfoil distribution
spl_k	1	Spline order
spl_s	1.0e-1	Spline smoothing

Table C.16: Spanwise CSV of the chord distribution.

<b>r/R</b>	<b>c/R</b>
0.24593411148612976	0.10355597570087384
0.4344505836145973	0.15848040401606145
0.622967055743065	0.0936915751607888
0.8114835278715324	0.16460169393869217
1.0	0.19793172889613356

Table C.17: Spanwise CSV of the height distribution.

<b>r/R</b>	<b>z/R (height of leading edge from top face of hub)</b>
0.24593411148612976	0.1799481526228537
0.4344505836145973	-0.19905081573477654
0.622967055743065	-0.0053923377143831296
0.8114835278715324	-0.23554853459324818
1.0	0.0027798044005781174

Table C.18: Spanwise CSV of the pitch distribution.

<b>r/R</b>	<b>twist (deg)</b>
0.24593411148612976	11.508767490414412
0.4344505836145973	5.322664609747395
0.622967055743065	16.879127846138765
0.8114835278715324	16.368717589755846
1.0	4.580035792791076

Table C.19: Spanwise CSV of the sweep distribution.

<b>r/R</b>	<b>y/R (y-distance of LE from the middle point of hub)</b>
0.24593411148612976	0.13330047426583205
0.4344505836145973	0.1321428556415045
0.622967055743065	0.09380391743014332
0.8114835278715324	0.09421043859248138
1.0	0.18368269222274938

## C.4 Second order six point interpolation

Table C.20: Radius of the tip, radius of the hub and number of blades.

<b>property</b>	<b>file</b>	<b>description</b>
Rtip	0.1046200130549139	(m) Radius of blade tip
Rhub	0.03	(m) Radius of hub
B	2	Number of blades
blade	current_blade.csv	Blade file

Table C.21: Blade file.

<b>property</b>	<b>file</b>	<b>description</b>
chorddist	current_chorddist.csv	Chord distribution
pitchdist	current_pitchdist.csv	Pitch distribution
sweepdist	current_sweepdist.csv	LE sweep distribution
heightdist	current_heightdist.csv	LE height distribution
airfoil_files	airfoils.csv	Airfoil distribution
spl_k	2	Spline order
spl_s	1.0e-1	Spline smoothing

Table C.22: Spanwise CSV of the chord distribution.

<b>r/R</b>	<b>c/R</b>
0.28675201927429816	0.19512408224962813
0.42940161541943855	0.2692094947923524
0.5720512115645789	0.20220077280912996
0.7147008077097192	0.22229196499826967
0.8573504038548596	0.3461468324379855
1.0	0.2484594009897129

Table C.23: Spanwise CSV of the height distribution.

<b>r/R</b>	<b>z/R (height of leading edge from top face of hub)</b>
0.28675201927429816	-0.19076884038748965
0.42940161541943855	-0.3942632888608881
0.5720512115645789	0.03301133450014103
0.7147008077097192	-0.12311327035304864
0.8573504038548596	-0.3284166315635257
1.0	-0.03265832533971684

Table C.24: Spanwise CSV of the pitch distribution.

<b>r/R</b>	<b>twist (deg)</b>
0.28675201927429816	16.542543810692948
0.42940161541943855	19.73227607435355
0.5720512115645789	22.12070683987077
0.7147008077097192	10.738189643586477
0.8573504038548596	20.889406229595227
1.0	22.89033733206168

Table C.25: Spanwise CSV of the sweep distribution.

<b>r/R</b>	<b>y/R (y-distance of LE from the middle point of hub)</b>
0.28675201927429816	-0.03791268978584183
0.42940161541943855	-0.04187914765425317
0.5720512115645789	0.04480969799346985
0.7147008077097192	0.20280597298249314
0.8573504038548596	0.27278609935678466
1.0	0.1986213768117942

# Appendix D

## Acoustic Optimization Results

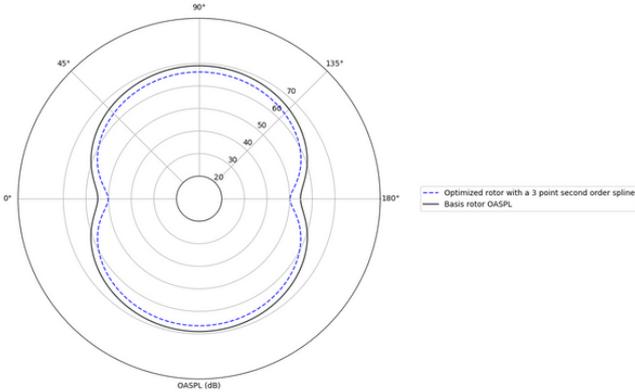


Figure D.1: Optimization OASPL results for a 3 point spline two-bladed rotor.

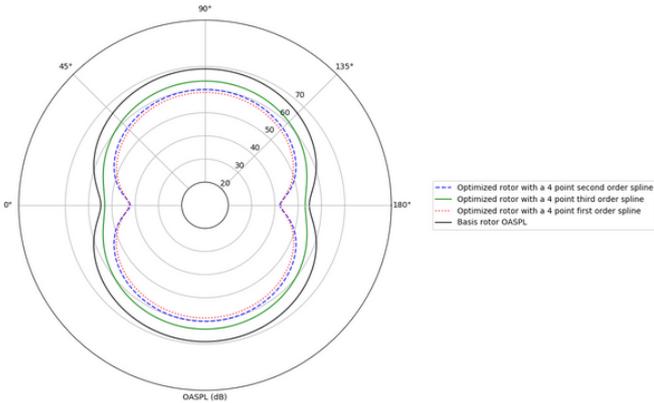


Figure D.2: Optimization OASPL results for a 4 point spline two-bladed rotor.

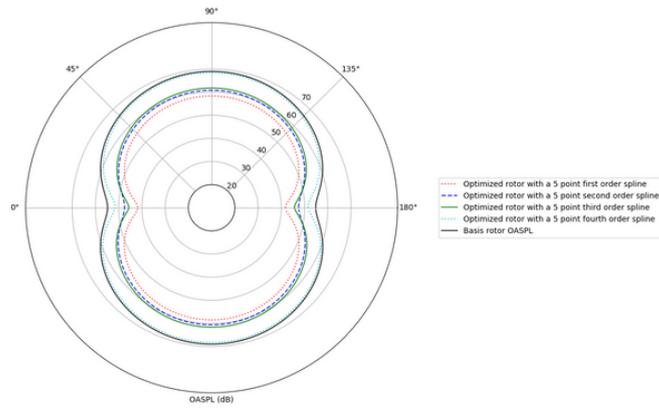


Figure D.3: Optimization OASPL results for a 5 point spline two-bladed rotor.

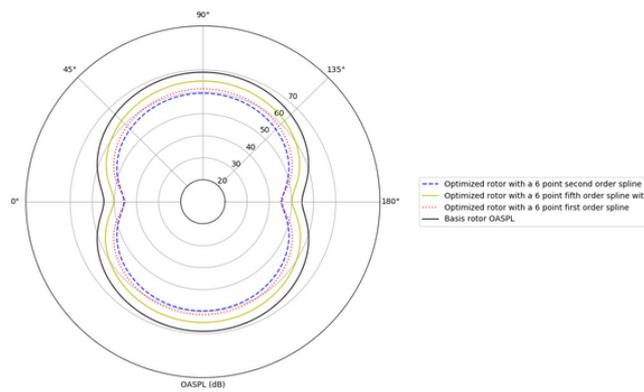


Figure D.4: Optimization OASPL results for a 6 point spline two-bladed rotor.