TÉCNICO LISBOA

Aeroacoustical Module Development for the NOVEMOR MDO Software

Miguel Reis Rodrigues Dias

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisor: Prof. Fernando José Parracho Lau

Examination Committee

Chairperson: Prof. Filipe Szolnoky Ramos Pinto Cunha
Supervisor: Prof. Fernando José Parracho Lau
Member of the Committee: Prof. João Manuel Gonçalves de Sousa Oliveira

July 2017

Acknowledgments

For the vast assistance provided in the development of this work I am grateful to Professor Fernando Lau, José Vale and Frederico Reis.

The following people have also, directly or through their work, assisted in this project: Ana Vieira, André Marta, Catarina Dias, Fernando Dias, João Souto, Maria do Céu Reis, Martin Sohst and Mónica Coelho. I thank them all.

This work was developed using several free tools. LaTeXsource was created with TeXstudio and compiled with MiKTeX. Figures were created using paint.net, Inkscape, Blender and LibreOffice. APM's C++ source code was created with Notepad++, Visual Studio Community and Visual Studio Code. The code was managed using the Git version-control system and checked with Cppcheck. The documentation was done with Doxygen. The setup was created using InnoSetup. Finally, a mention to cmder, EditPad Lite, Gource and OBS for helping in complementary tasks.

A sincere thank you to all those who contribute to the software commons or release free property software.

Resumo

O ruído prejudica a saúde humana e é uma das maiores fontes de incómodo para as populações afectadas. Os seus efeitos negativos para o bem-estar conduziram à implementação de limites de emissão para novas aeronaves, cada vez mais restritos, e a novos projectos de investigação, como o Clean Sky, que ambicionam reduzir as emissões de ruído. Uma área ainda num estágio inicial é a criação de ferramentas computacionais para prever emissões acústicas.

Este trabalho foca-se nos fundamentos de uma ferramenta de previsão acústica que foi desenvolvida e integrada num framework de optimização já desenvolvido no IST. Este framework surgiu inicialmente no âmbito do projecto EU FP7 NOVEMOR com o objectivo de estudar e desenvolver configurações originais e soluções de morphing. Até agora este conjunto de ferramentas não tinha a capacidade de fazer previsões acústicas dos modelos projectados. Este novo módulo vai possibilitar fazer previsões acústicas. A formulação 1A de Farassat é usada para determinar o ruído de espessura e de carga da superfície da aeronave. A previsão do ruído proveniente do trem de aterragem e do motor é obtido por modelos semi-empíricos. Adicionalmente o módulo pode comparar as previsões com os limites definidos na legislação de modo a facilitar a avaliação do ruído pelo utilizador. No futuro esta nova ferramenta pode ser usada como base para estudar vários cenários e para optimizar as emissões dentro deste framework.

Palavras-chave: aeroacústica, ruído da fuselagem, emissões de ruído, design preliminar de aeronaves

Abstract

Noise emissions are a human health hazard and one of the top sources of annoyance to the populations affected by aircraft noise. Its negative effects on wellbeing have led to increasingly strict noise standards and new research projects that aim to reduce aircraft noise such as Clean Noise. One field still not fully-fledged is the creation of noise emissions predictors.

This work focuses on the foundations of an acoustic prediction module integrated into an MDO framework. This framework was initially developed in the scope of EU FP7 project NOVEMOR, with the goal of preliminary analysis and design of novel aircraft configurations and morphing solutions. Until now this framework had lacked any way to predict noise emissions in its design. This new module will endow the MDO framework with this capability in order to include noise emissions into its workflow. The tool uses Farassat's Formulation 1A to solve the acoustic equations allowing the computation of loading and thickness noise. Landing gear and engine noise are predicted using semi-empirical models. Additionally, the module can compare noise predictions with noise limit standards, facilitating evaluation of noise legislation accordance. In the future, this module can be used as a basis for a variety of study scenarios and noise emission optimization inside the framework.

Keywords: aeroacoustics, airframe noise, noise emissions, preliminary aircraft design

Contents

Acknowledgments . . . iii
Resumo . . . v
Abstract . . . vii
List of Tables . . . xiii
List of Figures . . . xv
Nomenclature . . . xvii
Glossary . . . xxi

1 Introduction **1**
1.1 Motivation . . . 2
1.2 State of the Art . . . 3
1.3 Objectives . . . 4
1.4 Thesis Outline . . . 5

2 Theoretical Background **7**
2.1 Noise Emissions . . . 7
2.2 Context of Acoustic solver . . . 8
2.3 Airframe noise and Farassat's Formulation 1A . . . 8
2.4 Numerical Integration of the Retarded Time Formulation of the Wave Equation . . . 11
2.5 Landing Gear Noise Modelling . . . 12
2.6 Engine Noise Modelling . . . 14
2.6.1 Fan . . . 15
2.6.2 Turbine . . . 19
2.6.3 Combustion chamber . . . 20
2.6.4 Jet . . . 21
2.7 Signal Processing . . . 25
2.7.1 Frequency Domain, Fourier Transform, and Short Time Fourier Transform . . . 25
2.7.2 Nyquist–Shannon sampling theorem . . . 26
2.7.3 Noise Levels . . . 26
2.7.4 One Third Octave Bands . . . 29
2.8 Noise Limit Certification . . . 29

3 Implementation **31**

3.1 Module Structure . . . 31

3.2 Data Structures . . . 32

3.2.1 Flight Path Data . . . 33

3.2.2 Aircraft and Surface Pressure Data . . . 34

3.2.3 Microphone Data . . . 35

3.2.4 Engine Data . . . 36

3.2.5 Landing Gear Data . . . 38

3.2.6 Noise Level Limits Data . . . 39

3.3 Data Input and Files . . . 40

3.3.1 MDOGUI Input/Output facilities . . . 40

3.3.2 File Formats . . . 40

3.3.3 File location . . . 41

3.3.4 Flight Path Input . . . 41

3.3.5 Microphone Input . . . 43

3.3.6 Aircraft Data Input . . . 43

3.3.7 Engine Input . . . 44

3.3.8 Landing Gear Input . . . 45

3.4 Acoustic Prediction Algorithms . . . 46

3.5 Aircraft and Microphone Movement . . . 47

3.6 Sound Travel Time and Moving Microphones . . . 48

3.7 Formulation 1A Solver Algorithm . . . 49

3.7.1 Derivation of Panel Data . . . 51

3.8 Source Models Algorithm . . . 52

3.9 Signal Interpolation . . . 53

3.10 Short-Time Fourier Transform and Fast Fourier Transform . . . 54

3.11 Data Processing and Limit Certification . . . 55

4 Results and Validation **57**

4.1 Acoustic Code Validations . . . 57

4.2 Fourier Transform Validation . . . 57

4.3 Farassat Formulation Validation . . . 59

4.4 Landing Gear Model Validation . . . 60

4.5 Engine Model Validation . . . 61

4.5.1 Fan model validation . . . 61

4.5.2 Jet model validation . . . 62

4.6 Example Scenario . . . 63

4.7 Run Time . . . 65

4.8 Input impact on Engine model results . . . 66

5 Conclusions and Future Work **69**

5.1 Achievements 69

5.2 Future Work 70

Bibliography **73**

A APM Options file **81**

B Engine input file **83**

C One Third Octave Bands **85**

D Landing Gear Component Verification **87**

E Engine Component Verification **89**

F Input Impact on Engine Components **99**

List of Tables

3.1 Abridged index table for inputs of APM. . . . 40
3.2 Abridged Index table for outputs of APM. . . . 40
3.3 'Simple Format' for the path data input. . . . 42
3.4 Abridged 'Mission Format' for the path data input. . . . 42
3.5 Abridged microphone input file. . . . 43
3.6 Legislation point integer codes. . . . 43
3.7 Aircraft input file. . . . 44
3.8 Parameters for fan model. . . . 45
3.9 Parameters for combustor model. . . . 45
3.10 Parameters for turbine model. . . . 45
3.11 Parameters for jet model. . . . 45
3.12 Landing gear input file. . . . 46

4.1 Test rectangular wing and microphone position. . . . 59
4.2 Boeing 777 main landing gear data for Fink Model. . . . 60
4.3 Landing gear angular positions tested. . . . 60
4.4 Inputs for fan A used in [81]. . . . 61
4.5 Cold circular jet input. . . . 62
4.6 Boeing 737 landing gear used in example scenario. . . . 64
4.7 EPNL results of test scenario. . . . 65
4.8 Computation times in example scenario. . . . 66
4.9 Possible noise output range in dB when one input is determined and the two others vary by 25% of the default values. . . . 67

C.1 Table of One Third Octave Bands. . . . 85

List of Figures

2.1 Noise sources in an aircraft. . . . 7
2.2 Farassat 1A formulation surface representation. . . . 11
2.3 Landing Gear model coordinate system. . . . 13
2.4 Example of engine noise components and their directivity. . . . 15
2.5 Fan diagram and parameters. . . . 17
2.6 Turbine diagram and parameters. . . . 20
2.7 Combustor diagram and parameters. . . . 21
2.8 Jet diagram and parameters. . . . 22
2.9 Microphone/Observer positions in noise certification. . . . 30

3.1 APM's architecture. . . . 31
3.2 APM GUI. . . . 32
3.3 Representation of fight path segment discontinuities. . . . 34
3.4 Aircraft surface panel and its components. . . . 34
3.5 Aircraft mesh from the Aerodynamic module. . . . 35
3.6 Possible microphone study configurations in APM. . . . 36
3.7 Engine model levels. . . . 37
3.8 Class inheritance tree in engine sub-models. . . . 37
3.9 Simplified class inheritance tree in engine noise components. . . . 38
3.10 Requesting Aerodynamic module for aircraft data from APM. . . . 44
3.11 Source Models algorithm flow. . . . 46
3.12 Formulation 1A algorithm flow. . . . 47
3.13 Algorithm flow for integrand computation for all Panel/Mic couple in one time step. . . . 51
3.14 Source modelling algorithm flow in one time step. . . . 52
3.15 Sound propagation path from two generic panels. . . . 53
3.16 Interpolation of noise signal to common set of times. . . . 54
3.17 Window functions used in Short-Time Fourier Transform. . . . 54
3.18 Post processing of noise signal for each microphone. . . . 56

4.1 Comparison of starting step signal and final signal after the inverse FFT. . . . 58
4.2 Output from FFT of step signal in the frequency domain. . . . 58

4.3 APM 1A results versus TU Delf and FAWT. 59
4.4 APM's Fink model results. 61
4.5 APM's Heidman model results. 62
4.6 APM's Stone Jet model results. 63
4.7 PNLT results for flyover position. 64
4.8 PNLT results for lateral positions. 65

D.1 Spectrum Levels for Landing Gear. 87

E.1 Directivity Level for Fan Components. 90
E.2 Power Level for Fan Components. 92
E.3 Spectrum Level for Fan Components. 94
E.4 Directivity Level of Combustion Noise. 94
E.5 Spectral Level of Combustion Noise. 95
E.6 Directivity Level for Turbine Components. 95
E.7 Spectrum Level for Turbine Components. 96
E.8 Jet Mixing Noise Directivity Factor. 96
E.9 Spectral Levels for Jet Components. 97
E.10 Jet Mixing Noise Frequency Shift Parameter. 97

F.1 Fan Inlet Broadband component acoustic power (log Π*). 99
F.2 Fan Discharge Broadband component acoustic power (log Π*). 100
F.5 Fan Inlet Distortion component acoustic power (log Π*). 100
F.3 Fan Discharge Rotor-Stator Interaction component acoustic power (log Π*). 101
F.6 Fan Inlet Combination Tone component acoustic power (log Π*). 101
F.4 Fan Inlet Rotor-Stator Interaction component acoustic power (log Π*). 102
F.7 Jet acoustic power (log Π*). 102
F.8 Combustor acoustic power (log Π*). 103
F.9 Turbine Broadband component acoustic power (log Π*). 103
F.10 Turbine Pure Tone component acoustic power (log Π*). 103

Nomenclature

Greek symbols

α	Angle of attack.
β	Sideslip angle.
β_s	Pressure ratio parameter.
δ	Dirac delta function.
η	Frequency parameter.
ϕ	Azimuthal angle in semi-empirical source models.
Π	Acoustic power function in semi-empirical models.
ρ	Air density.
τ	Emission time.
θ	Polar angle for semi-empirical source models.
θ'	Modified directivity angle.
θ_A	Local angle between the normal to the surface and the radiation direction in Formulation 1A.

Roman symbols

c	Sound speed.
c_p	Pressure coefficient.
D	Directivity function.
d_w	Landing gear's wheel diameter.
F	Spectral distribution function.
f	Frequency.
f_s	Source surface function.
H	Heaviside function.

K Acoustic power constant.

L_s Landing gear strut length

M Mach number.

m Mass.

N_w Number of wheel is one landing gear.

p Pressure.

p' Pressure perturbation

p'_L Pressure perturbation from Loading Noise.

p'_T Pressure perturbation from Thickness Noise.

r Source observer distance.

S Surface area.

S_{lg} Strouhal number for the landing gear.

S_m Strouhal number for the jet mixing noise.

S_s Strouhal number for the jet chock noise.

T Flow temperature.

t Obserser arrival time.

t^* Interpolation time.

V Flow speed.

v Velocity.

x Observer position.

y Source position.

Subscripts

∞ Free-stream condition.

a Ambient condition.

n Normal component.

r Propagation direction component.

ref Reference condition.

ret Retarded time.

Superscripts

*	Non-dimensional value in Semi-empirical model.

Glossary

ACARE	Advisory Council for Aviation Research and Innovation in Europe
ANOPP	Aircraft Noise Prediction Program
APM	Acoustic Prediction Module
ASSPIN	Advanced Subsonic and Supersonic Propeller Induced Noise prediction program
Broad.	Broadband
CSV	Comma-separated values
Combination T.	Fan Combination Tones Component
DLL	Dynamic-link library
Disch. Broad.	Fan Discharge Broadband Component
Disch. R-S. I.	Fan Discharge Rotor-Stator Interaction Tones Component
EASA	European Aviation Safety Agency
EPNL	Effective Perceived Noise Level
F1A	Farassat's Formulation 1A
FAA	Federal Aviation Administration
FAST	Farassat 1A Acoustic Solver Tool
FAWT	Farassat 1A Wind Turbines
FFT	Fast Fourier Transform
FP	Research and Innovation Framework Program
FW-H	Ffowcs Williams-Hawkings
ICAO	International Civil Aviation Organization
INM	Integrated Noise Model
ISF	INI Similar File
Inl. Broad.	Fan Inlet Broadband Component
Inl. Dist.	Fan Inlet Distortion Tones Component
Inl. R-S. I.	Fan Inlet Rotor-Stator Interaction Tones Component

MDOGUI	Multidisciplinary Design Optimization Graphical User Interface
MDO	Multidisciplinary Design Optimization
MUTE	Multidisciplinary Tool for Accurate and Efficient Rotorcraft Noise Prediction
NASA	National Aeronautics and Space Administration
NOVEMOR	Novel Air Vehicle Configurations: From Fluttering Wings to Morphing Flight
OASPL	Overall Sound Pressure Level
OS	Operating System
PANAM	Parametric Aircraft Noise Analysis Module
PNLT	Tone Corrected Perceived Noise Level
PNL	Perceived Noise Level
RAM	Random Access Memory
SAI	Silent Aircraft Initiative
SPL	Sound Pressure Level
STFT	Short Time Fourier Transform
TCCA	Transport Canada Civil Aviation
UTF	Unicode Transformation Format

Chapter 1

Introduction

Aircraft noise is an integral problem in a world increasingly connected by air traffic. The air transportation sector, backed by political decision-makers, is taking on noise emissions to support a sustainable future where the noise hazard to the population is mitigated.

Noise exposure is predicted to rise in the coming years. The air transport business, which historically was characterized by boom to bust cycles, has transformed into a sustainable profit business in the past decade [1, 2]. Worldwide, both scheduled air passenger traffic and air cargo traffic are projected to grow above 4.0 % annually in the next 20 years [3, 4]. This will accrue to a doubling of global air traffic [3, 5].

Parallel to the growth of air travel, the world's population is increasingly urban. Currently, 54 % of the world's population lives in urban centers [6]. By 2030, it is predicted that 60 % will live in cities [6]. The expansion of urban centers engulfed what once were peripherical airports. This trend in human geography and air traffic increases the number of people being exposed to aircraft noise. The situation is aggravated by the fact that a combination of rising demand for air travel and cutbacks in infrastructure expansion plans will result by 2035 in 12 percent of the total demand (1.9 million flights) not being accommodated, in the most likely scenario as forecasted by Eurocontrol [5]. The result will be delays and a congested network, which tend to increase overflight time over urbanized areas.

As noise exposure rises, the public and political decision makers are progressively aware of its effects and associated welfare detriments. The studies conducted so far, most of them epidemiological, still leave gaps in knowledge of how exactly noise affects human welfare [7, 8]. However, the data collected is considered sufficient to adamantly state that noise has significant detrimental health effects [7, 8]. Strong correlations were found between noise and cardiovascular diseases, sleep disturbance, tinnitus and annoyance among others, which take a toll on life expectancy and welfare. In Europe, all environmental noise sources are estimated to toll at least 1 million healthy years of life annually [7]. These correlations have been specifically verified for aircraft noise by several studies [9, 10]. The annoyance and health impacts make public pressure decision makers and the aviation sector to confront this issue.

Action has been taken in two ways to address noise emissions. First, international institutions and lawmakers release industry guidelines and enact strict noise emission limits on new aircraft. The en-

actment of these limits in each country is the responsibility of the aviation authority with the respective jurisdiction. Examples are the Federal Aviation Administration (FAA) in the United States, the European Aviation Safety Agency (EASA) in the majority of European countries or the Transport Canada Civil Aviation (TCCA) in Canada. Initially, legislation was done at national level, but there were efforts to harmonize legislations worldwide to ease the certification process. Nowadays the large majority of aviation authorities adopt the limits set by the International Civil Aviation Organization (ICAO) standard [11], including the authorities mentioned above [12, 13, 14]. These rules have been updated over the years to ever stricter limits [14, 15], accompanying the evolution of noise reduction technologies. The newest standard, ICAO's *Annex 16 Chapter 14* [11] and the respective FAA's *Stage 5*, will start being used at the end of 2017 [15]. Several aircraft already abide by this new standard such as Airbus' A-380, A-350 and Boeing's 747-8 and 787 [15].

Concurrently, incentives were created towards research on aircraft technologies that reduce noise emissions. As examples, we cite the EU Research and Innovation Framework Programs (FP) and the aeronautics focused public-private partnership Clean Sky. The Framework Programs have a broad scope of technological research and innovation. In the instances of funding dedicated to noise emissions under FP5, FP6 and FP7 various successful researches have contributed with new technologies and flight procedures that led to significant noise reductions [16]. Clean Sky was created in 2008 with the main goal of reaching the 2020 Advisory Council for Aviation Research and Innovation in Europe (ACARE) environmental goals for the aviation industry [17]. Regarding noise emissions, the ACARE goal is to reduce external noise in 2020 by 50% compared to 2000's state of the art values [18]. Preliminary assessments can confirm an achieved noise reduction of 37% by 2020 [19]. This gap may be due to a lag between technology development and its implementation in commercial aircraft which is outside the scope of Clean Sky [18]. The Clean Sky venture continues with Clean Sky 2 which was launched in 2014. This second phase will last until 2024 and will continue the work developed to ensure that the goals for 2020 are met [18]. ACARE has since updated their goals and established a long-term commitment to reducing by 2050 the perceived noise emission of flying aircraft by 65% compared to the 2000's baseline [20, 16]. Horizon 2020, FP7's successor, was launched in 2014 and backs up Clean Sky 2's activities in reaching the 2020 goals and progressing towards the 2050 goals [21]. The budgets for these programs have increased which indicates continued political support from the European Commission towards this research [18].

1.1 Motivation

Designers require acoustic prediction tools at several levels, from single component noise to full aircraft emissions up to predictions of noise around airfields in long time scales. Aircraft noise prediction is a field still not mature and lacking in several aspects. One need is to predict noise in early stages of aircraft design. At this early steps, the data is scarce and most predictions are made using empirical or semi-empirical models [22]. There is also a need for high fidelity noise prediction tools integrated into Multidisciplinary Design Optimization (MDO) frameworks that analyse experimental aircraft designs [22].

The EU FP7 project NOVEMOR focused on researching novel air vehicle configurations. In its scope IST developed a software tool, an MDO framework [23, 24]. This framework is extensible and now counts with several modules which allow analysis in different areas such as aerodynamics, structures, and propulsion among many others. So far it lacks the capabilities to predict acoustic emissions. The drive for this work is, then, to endow the MDO framework with the needed acoustic prediction tool.

1.2 State of the Art

Aircraft noise prediction methods are still not mature. Developers, faced with different sets of conditions and goals, have made progress in different directions, which resulted in a myriad of different tools. A state of the art of current noise prediction has been extensively done in Ref [22, 25] and for NASA specific research in [26]. While there were many new tools for rotorcraft noise prediction (such as NASA's Multidisciplinary Tool for Accurate and Efficient Rotorcraft Noise Prediction (MUTE) [27] or FAST [28]), this work focuses on fixed wing aircraft noise prediction and as such this section will address only this subgroup. What follows is a quick summary from the mentioned sources with ancillary information.

Noise prediction methods are subdivided into 2 broad groups: best practice methods and theoretical methods. Best practice methods rely mostly on measurement databases. Theoretical methods, on the other hand, rely on physical models to calculate the noise emissions.

Best practice methods are geared towards providing fast estimates of the noise impact from aircraft operations in an airport and its surroundings over long time frames. These tools are used by national authorities to predict noise contours in their jurisdictions and there are many variations. INM (Integrated Noise Model) is one of the most used and has been continuously improved for decades serving as a base for many other solutions. Because long-term noise impacts are outside the scope of this work we recommend Filippone [22] for more information on several other tools of this kind and their details.

Broadly speaking, the theoretical tools available can be subdivided into 2 major groups: the commercial/industry property; and the research tools developed by universities, other research institutions and governmental institutions such as NASA. There is little information and documentation available on the first group. As an example, Airbus has several prediction programs, developed in the scope of national and international research programs, such as SILENCE(R), SOURDINE, AWIATOR and RAIN [22]. For noise prediction and optimization of single components, such as nacelles, Airbus also uses commercial software based on physical models such as ACTRAN [29].

Tools developed in research focused institutions are better documented and we proceed in discussing the efforts by NASA, the Cambridge-MIT Institute, DLR and The University of Manchester.

NASA's Aircraft Noise Prediction Program (ANOPP), which was the first software tool of its kind, is documented throughout with several papers on individual components and exhaustive user manuals and technical reports [30, 31]. The noise predictions are made relying extensively on empirical models. Despite its first iteration being developed in the 70's, this software tool is still in service because it has been continuously updated. ANOPP has a successor named ANOPP2 [32] which applies some higher fidelity models which are computationally intensive and require a larger number of inputs. Other software tools

have also been developed by NASA such as Advanced Subsonic and Supersonic Propeller Induced Noise prediction program (ASSPIN), focused on propeller noise, which is based on Farassat's formulations [33]. It should also be noted that NASA researchers are engaged in developing new theoretical models which can be applied to new software or to update old tools. One such example is Farassat's Formulation 2B which calculates broadband noise from turbulence data and it could be integrated into ASSPIN and ANOPP [34].

Funded by the Cambridge-MIT Institute, the Silent Aircraft Initiative (SAI) [35] developed a tool which aimed at developing unconventional aircraft. In this work, it was important to consider fuselage shielding and as such, it was required to implement a computationally intensive method of ray-tracing in three-dimensional space.

DLR has also made significant contributions with two pieces of software: Parametric Aircraft Noise Analysis Module (PANAM) [36] and SIMUL [37]. Both rely on source modeling of the major noise components. PANAM focuses on rapid noise evaluations for comparisons between designs. Representative of this is the fact that engine emissions are estimated solely with the fan and jet contributions which are considered predominant.

Finally, we mention FLIGHT [38, 39]. FLIGHT was developed at the University of Manchester and consists of a multi-disciplinary analysis software. It contains an acoustics branch where the acoustics emissions rely mostly on source modeling. However, the tool possesses several extra features such as shielding effects (by jets and, in a simplified way, by the fuselage), noise contours around airports and wind effects per example. This tool has been used for instance to study the impact of modified landing procedures [40].

1.3 Objectives

In this work a software tool shall be developed with the following main goals:

- Develop a low demanding software module that predicts noise emissions of an aircraft and mission profile provided by the framework;
- Include all the major aircraft noise sources;
- Be completely flexible as to the shape of the airframe and to the mission profile due to the exploratory nature of the project it is inserted in;
- Evaluate noise emissions and test if they are inside certification standard's limits;
- Develop the module such that it is ready to be linked with the software framework to create an acoustic emissions optimizer.

This tool will have the particularity of being integrated into a conceptual MDO framework for conceptual preliminary designs, of which there are not many.

1.4 Thesis Outline

This thesis is subdivided into five chapters.

This first chapter introduced the work to be developed and its context.

Chapter 2 will explain the theoretical foundations which support the software tool. The decisions which lead to their selection, in detriment of other possible models, will also be covered.

Chapter 3 will mainly discuss the source code and how the theory in Chapter 2 was implemented into the software.

In Chapter 4, the software tool is validated in its components and results.

For closure, Chapter 5 will state the conclusions and the possible extensions that could expand this work's reach.

Annexes will also be presented after Chapter 5, showing relevant information about the development and models in the software.

Chapter 2

Theoretical Background

This chapter addresses the background theory that served as the basis for the software development and implementation. Alongside the theoretical foundations, it is briefly discussed why this basis was adopted instead of other possible solutions. This chapter is divided into four major parts. Sections 2.1 and 2.2 will make an introduction of noise sources and of the software context of an acoustic solver respectively. Sections 2.3 to 2.6 will cover in detail the foundations of the noise emission predictors that were put into effect. Noise signal processing is discussed in section 2.7. Finally section 2.8 will cover aircraft noise limits validation standards.

2.1 Noise Emissions

The noise emissions from an aircraft have many sources. As we will discuss, in this work we divide the sources into three groups which can be seen in Fig. 2.1. The first group is the aircraft fuselage and lifting surfaces. We group them here because they are addressed in the same fashion using an acoustic analogy method (section 2.3). The second important group is the engines' noise, which is a major contributor to the total emitted noise.

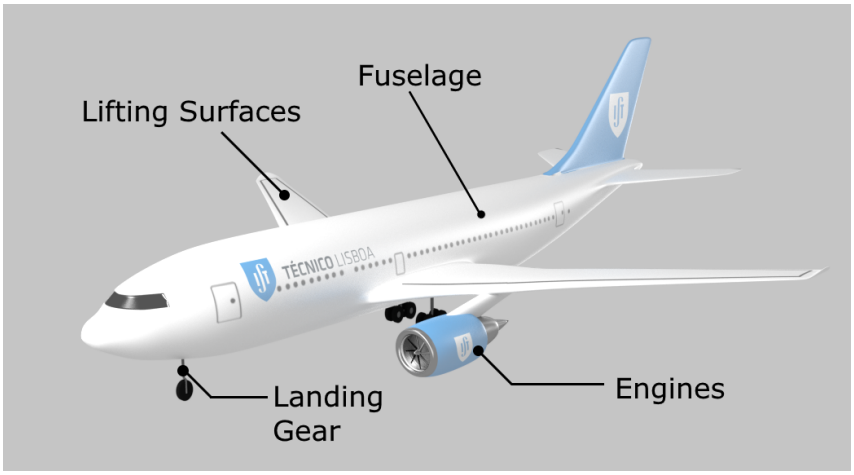


Figure 2.1: Noise sources in an aircraft.

In early development stages, what is usually done is to model the noise of each engine component separately and sum the results. The models used for this approach will be discussed in section 2.6. Finally, we have the landing gear noise. This source is usually grouped with the rest of the airframe noise. In this work, however, they are tackled in a fundamentally different ways and, as such, are addressed separately.

2.2 Context of Acoustic solver

An acoustic prediction tool is generally not fully contained. That is to say, the acoustic code requires a set of companion applications that perform analysis in different fields such as geometry discretizers, aerodynamic solvers, and flight or attitude control modules. Usually, an acoustic solver will require the following basic inputs:

- Flight path data, to locate the aircraft position and attitude in time;
- Observer position;
- Aircraft geometry, which describes the position and other required information of the noise sources in the airplane;
- Aerodynamics data, such as flow information, surface pressures or relevant coefficients.

In this work, the set of tools to work with is mostly restricted to the utilities provided in the software framework. In many occasions, we shall refer to the MDOGUI and its components because its architecture and capabilities will favor certain solutions and hamper others.

2.3 Airframe noise and Farassat's Formulation 1A

Airframe noise prediction is currently achieved by one of four strategies: analytical solutions, semi-empirical models, full numerical methods or acoustic analogy based methods [41].

Analytic solutions are only used for simple geometries and thus require some geometrical simplifications to be made. The objective of the current project is to be flexible regarding the aircraft geometry because the project's nature is to develop innovative design configurations. As such this approach cannot be used in this acoustic module. In addition, the current framework has no symbolic algebra software integration which is required to use an analytic method.

Semi-empirical models have a wide use in software such as ANOPP. Based on vast amounts of acoustic data these methods can predict noise emissions but their use is restricted to conventional flight conditions and airframe configurations. This goes against the explorational nature of the target software framework, where solutions such as morphing wings are studied. Furthermore, there are not semi-empirical models for main wing noise.

In full numerical methods, the fluid flow and noise propagation are integrally computed numerically from the governing partial differential equations for sound propagation and fluid mechanics. These

methods are very computationally intensive. This is compounded by the fact that aircraft emissions can reach very high frequencies which entail very small time steps. Therefore, to simulate the noise emissions of a full aircraft in a broad range of frequencies for the duration of even a small flight segment, the use of this type of methods is not feasible. Moreover, this work is inserted in an early design framework and so quick turnaround times are expected for quick iteration and analyses.

We are left with acoustic analogy based methods. These have the widest implementation. In such a method, there is a first stage where fluid dynamics computations are done. Next, these results are piped to an acoustic code which predicts noise emissions with an algorithm based on the acoustic analogy [42]. This proves to be the best fit to achieve simultaneously fast turnaround of acoustic results and flexibility in the airframe geometries. The governing equation most widely used in this type of acoustic codes is the Ffowcs Williams-Hawkings equation (FW-H) which is based on the Lighthill's acoustic analogy [43]:

$$\Box^2 p' = \frac{\partial}{\partial t}\left[\rho_0 v_n \delta\left(f_s\right)\right] - \frac{\partial}{\partial x_i}\left[p n_i \delta\left(f_s\right)\right] + \frac{\partial^2}{\partial x_i \partial x_j}\left[T_{ij} H\left(f_s\right)\right] \tag{2.1}$$

where p' is the noise pressure perturbation, $\Box$ is the D'Alembert operator $\left(\Box^2 = \frac{1}{c^2}\frac{\partial^2}{\partial t^2} - \nabla^2\right)$, ρ_0 is the ambient air density, v_n the velocity's normal component to the surface, n the surface normal, f_s is the function that describes the source surface, $\delta\left(f_s\right)$ and $H\left(f_s\right)$ are the Dirac delta function and Heaviside function respectively and T_{ij} is Lighthill's stress tensor which is given by:

$$T_{ij} = \rho_0 v_i v_j + p_{ij} - c^2(\rho - \rho_0)\delta_{ij} \tag{2.2}$$

The terms on the right-hand side of the equation (2.1) are, from left to right, know as monopole, dipole, and quadrupole terms. To evaluate the quadruple term, T_{ij}, integration across a volume surrounding the aircraft is required. This implies that the full flow field should be known around the aircraft, including turbulence. Turbulence calculations require fine grids and small time steps and as such they are very costly in computational terms. Moreover, the current MDOGUI framework does not have a turbulence simulation algorithm. However, the quadrupole term is very inefficient at producing sound compared to the other two and it varies with a high power of the fluid flow [42]. Therefore, this term is usually neglected in the subsonic region because the monopole and dipole terms have a much higher impact. The aircraft's noise has the most impact when it is close to the ground in the takeoff and landing stages [41] for which we are in the low subsonic regime.

Equation (2.1) can have several different solutions [44]. Of the several possibilities the solution adopted was Farassat's Formulation 1A (F1A) [44, 45], which is obtained neglecting the quadrupole term in (2.1) [44]. This formulation has several advantages which single it out, namely [44, 45]:

1. The observer position and motion is arbitrary, whilst in some other solutions the observer must move with the source;

2. Formulation 1A is valid for any aircraft motion, insofar as it remains in the subsonic regime, unlike other solutions which limit the source motion to be uniform rectilinear;

3. It is only required to have information on the airframe surface, and the integrations are performed on the surface. This reduces the required inputs for the acoustic code and the computation time.

The mathematical groundwork required to derive Formulation 1A, the generalized function theory and, Green's function solution, are covered extensively in references [46, 47, 48, 49]. The full derivation of Formulation 1A was extensively covered in [45] and [28] which present all the algebraic manipulation steps, and as such, they will not be repeated here. Formulation 1A final result gives the pressure perturbation p' at observer position x and time t by the equation:

$$p'(x,t) = p'_T(x,t) + p'_L(x,t) \tag{2.3}$$

where p'_T is the pressure perturbation from the thickness noise and p'_L the pressure perturbation from the loading noise. The thickness noise originates from the fluid displacement when the body moves, while the loading noise is due to the fluid acceleration induced by the body. These terms should not be labelled as monopole or dipole sources because it has been shown that their radiation fields differ from those of stationary monopoles and dipoles [45]. These values can be computed independently by the following equations:

$$4\pi p'_T(x,t) = \int_{f_s=0} \left[\frac{\rho_0 \dot{v}_n}{r(1-M_r)^2} + \frac{\rho_0 v_n (\hat{r} \cdot \dot{\vec{M}})}{r(1-M_r)^3} \right]_{ret} dS + \int_{f_s=0} \left[\frac{\rho_0 c \dot{v}_n (M_r - M^2)}{r^2(1-M_r)^3} \right]_{ret} dS \ , \tag{2.4}$$

$$4\pi p'_L(x,t) = \int_{f_s=0} \left[\frac{\dot{p}\cos\theta_A}{cr(1-M_r)^3} + \frac{(\hat{r} \cdot \dot{\vec{M}}) p \cos\theta_A}{cr(1-M_r)^3} \right]_{ret} dS + \int_{f_s=0} \left[\frac{p(\cos\theta_A - M_n)}{r^2(1-M_r)^2} + \frac{(M_r - M^2) p \cos\theta_A}{r^2(1-M_r)^3} \right]_{ret} dS \ , \tag{2.5}$$

where ρ_0 is the undisturbed air density, v_n is the normal component of the velocity in relation to the surface ($v_n = \vec{v} \cdot \vec{n}$), r is the distance from the source y to the observer x ($\vec{r} = \vec{x} - \vec{y}$), $\hat{r}$ is the unit vector with the same direction as r, c is the sound speed, M is the Mach number, M_r is the Mach number in the radiation direction $\left(M_r = \vec{M} \cdot \hat{r}\right)$, p is the pressure on the surface, θ_A is the local angle between the normal to the surface and the radiation direction, and S is the surface area. This notation is according to Farassat [45]. The subscript *ret* indicates that the quantities are evaluated at the retarded time τ, including the derivatives $(\partial/\partial\tau)$. Its implications will be discussed later.

As it was mentioned above, this formula has two limitations. First, it does not include turbulence noise generation. Second, it cannot be used in the supersonic range. We have already talked about why turbulence is disregarded in this situation. Some acoustic analogy formulations exist that consider turbulence's impact on the body's emissions, such as Formulation 2B [50]. However, this would require extensive turbulence computations [50] which are currently unavailable in the framework, and thus a new aerodynamic module or an extension to the current one are required before this approach is possible. As for supersonic motion, in this acoustic prediction tool, their use would be limited. Supersonic cases

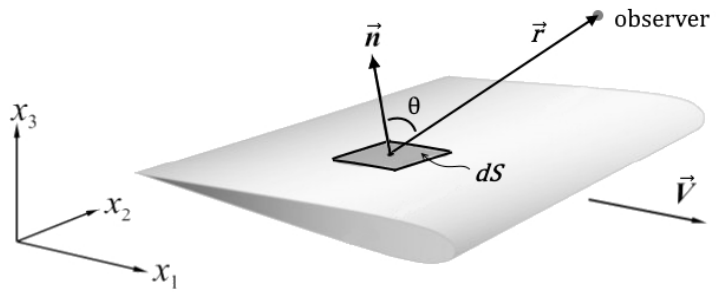


Figure 2.2: Farassat 1A formulation surface representation. Adapted from [45].

are important, for instance, in rotorcraft and propellers. In several acoustic codes, these scenarios are addressed by implementing another solution, Formulation 3, in parallel with Formulation 1A [33, 51]. For fixed wing aircraft, the scenarios in which this is required are more limited. Thus, due to time constraints, focus was directed towards Formulation 1A, because the noise emissions in subsonic conditions, namely landing and taking off, have a much higher interest and wider applicability.

2.4 Numerical Integration of the Retarded Time Formulation of the Wave Equation

The integrals in equations (2.4) and (2.5) have to be calculated using a numerical algorithm. As it has been said these integrals are to be computed at the retarded time. Consider the generic retarded-time formulation for the wave equation:

$$4\pi p'(x,t) = \int_{f_s=0} \left[\frac{Q(y,\tau)}{r|1-M_r|}\right]_{ret} dS \tag{2.6}$$

where r, M_r and S are defined as in the equations (2.4) and (2.5), the condition $f = 0$ describes the integration surface and Q is a known source strength described as a function of the source's position y and the retarded time τ. This integral is solved numerically using one of three methods: Midpanel Quadrature, High-Accuracy Quadrature or Source Time-Dominant Algorithm [52]. In the first two methods the observer time is set and from it, the retarded time is calculated. Since the source function Q is only known at discrete emission times this implies that the input surface data must be interpolated. The Source Time-Dominant Algorithm was chosen as the method to solve the retarded time integral. In this method, it is the retarded time that is known in advance and so interpolations of the input data are avoided. The observer times are calculated as a function of the retarded time, for each source panel. The result is one observer reception time for each panel and for every emission time. In order to obtain the signal in the time domain, all panel pressure contributions must be summed. To do so the pressure values have to be interpolated to the same observer times. This method can be expressed as:

$$4\pi p'(x,t^*) = \sum_{i=1}^{N} I\left[K_i(t), t^*\right], \tag{2.7}$$

where t is the travel time of a sound from one source to the observer, $I[\cdot, t^*]$ is the interpolation operator to one time t^*, K_i is the contribution from panel i, which is calculated by approximating the integral in equation (2.6) by:

$$K_i(t) = \frac{Q(y_i, \tau)}{r_i |1 - M_r|_i} \Delta S_i \ . \tag{2.8}$$

In section 2.7.2 the spacing requirements for the interpolated observer times (Δt^*) will be discussed.

2.5 Landing Gear Noise Modelling

Farassat's Formulation 1A can compute the airframe's loading and thickness noise to the extend its geometry is discretized. The landing gear noise could follow suit and be predicted with F1A. However, two problems arise:

- The current MDOGUI framework provides no landing gear modeling support. That means it can neither provide surface data neither to the aerodynamic nor the acoustics code;
- The correct description of the landing gear will require a finer mesh than the rest of the airframe from its smaller characteristic dimensions and intricate design. This would burden the fluid and the acoustics codes, especially if what is intended is a quick approximate estimation of the noise.

Due to these reasons, it was decided that for the landing gear predictions a source model should be available. The Fink's prediction method [53] was the selected source model. The choice was done based on the fact that this method was proven to give good results for flyover positions [54] and at the same time, it is simple to implement and use, needing only a few inputs which are empirical constants and simple geometrical data like strut length, the number of wheels and the wheel diameter. The Fink method gives the far-field mean-square acoustic pressure of the landing gear by summing the contributions from the strut and from the wheels:

$$\langle p^2 \rangle = \langle p^2 \rangle_{strut} + \langle p^2 \rangle_{wheel} \ . \tag{2.9}$$

The far-field mean-square acoustic pressure for each component is given by

$$\langle p^2 \rangle = \frac{\rho c \Pi}{4\pi r^2} \frac{D(\theta, \phi) F(S_{lg})}{(1 - M \cos\theta)^4} \ , \tag{2.10}$$

where p is the far-field acoustic pressure, D is the directivity function, θ is the polar (flyover) angle, ϕ is the azimuthal (sideline) angle, Π is the acoustic power, r is the distance from source to observer, M is the Mach number and F is the spectrum function, which depends on the landing gear Strouhal number, S_{lg}. All the values are to be computed at emission time. Figure 2.3 shows the coordinate system and the angles used in this model.

The acoustic power for the wheel contribution is calculated with the expression

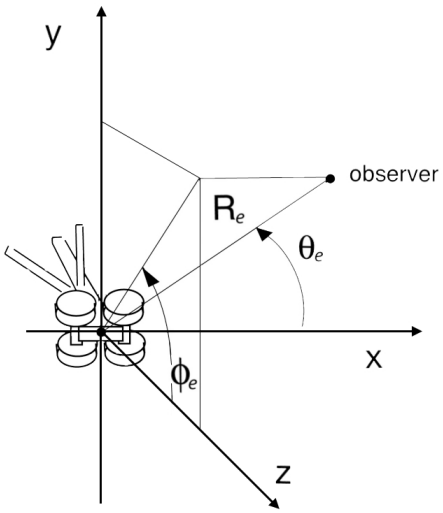


Figure 2.3: Landing Gear model coordinate system.

$$\Pi_{wheel} = K_1 M^6 d_w^2 N_w \ , \tag{2.11}$$

where d_w is the wheel diameter, N_w the number of wheels and K_1 is an empirical constant equal to 4.349×10^{-4} for one and two-wheel assembly and 3.414×10^{-4} for four-wheel assembly [30]. The directivity function for the wheel contribution is given by

$$D_{wheel}\left(\theta, \phi\right) = \frac{3}{2} \sin^2 \theta \ . \tag{2.12}$$

And finally the spectral distribution F

$$F\left(S_{lg}\right) = A \frac{S_{lg}^n}{\left(B + C S_{lg}^m\right)^q} \ , \tag{2.13}$$

where A, B, C, n, m, q are empirical constants which are specific to the model [30]. The Strouhal number is the dimensionless frequency of vortex shedding [55] and in this model it is defined as:

$$S_{lg} = \frac{fd}{Mc} \ . \tag{2.14}$$

As for the strut contribution, the acoustic power is provided by

$$\Pi_{strut} = K_2 M^6 d_w L_s \ , \tag{2.15}$$

where d is the wheel diameter, L the strut length and K_2 is a constant equal to 2.753×10^{-4} [30]. The strut directivity function is the following:

$$D_{strut}\left(\theta_e, \phi_e\right) = 3 \sin^2 \theta_e \sin^2 \phi_e \ . \tag{2.16}$$

The spectral distribution function F for the strut is similar to (2.13), but the empirical constants A, B, C,

n, m, q have values specific to the strut [30], different from those defined for the wheel.

The Fink method has two disadvantages. First, the prediction method is limited to landing gears with a maximum of four wheels. If the number of wheels is above four the method defaults to the 4 wheel case. Secondly, it does not fully capture noise directivity [54]. This was acquiesced to because this light weight model is used to provide a quick estimation of the landing gear noise and its use is optional. In the future, with a lightweight extension to the Geometry module, the landing gear can be readily integrated in F1A's prediction.

2.6 Engine Noise Modelling

The engines are an important contributor to the noise produced by an aircraft. Its prediction with Formulation 1A is even more contestable than in the case of the landing gear due to the following points:

- The engine geometry is very intricate and a very fine mesh would be required to properly discretize it;
- Currently no module in the MDOGUI framework has the utilities to discretize engine geometries;
- It is common for blades inside the rotor, at least in the tip region, to be at Mach numbers close to 1. Therefore, the assumption of a low Mach number that we had made to justify Formulation 1A is not valid;
- An important contribution from the engine noise comes from the exhaust jet. Many times the airflow inside the engine is at or close to sonic conditions. Moreover, turbulence can not be neglected as a noise source in the jet.

For all these reasons Formulation 1A can't be used to calculate the engine noise. The lack of detailed engine data also prevents the application of an acoustic analogy solver. Even if possible the full numerical solver would go against the objective of creating an acoustic predictor with fast turnaround for conceptual designs. As such the alternative available is the use of a semi-empirical source model. The current MDOGUI framework has one Propulsion Module which operates using characteristic values for the engines. This supports the use of Empirical models because they usually receive characteristic geometric dimensions and operation conditions as input. The engine is modeled by joining the contributions from different engine components which act as almost independent sources. As mentioned before the most important contributors to the engine noise are the fan and the exhaust jet. Other components modeled are the combustion chamber, the turbine, and Auxiliary Power Units (APU). Research on these models has progressed at different paces. The fan and the jet have been the most researched among them. For other contributors, like the turbine and APU, there is little data available [22]. All current models predict noise in stationary conditions and so to determine the noise signal in time a series of steady states is used. In this work we modeled the main jet engine contributors to noise:

- Fan and compressors;

- Combustion chamber;
- Turbine;
- Jet.

Fig. 2.4 shows an approximate graphical representation of the directivity and relative importance of each source.

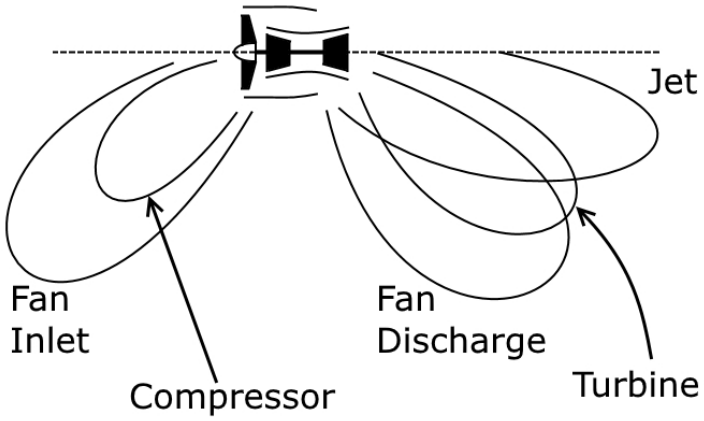


Figure 2.4: Example of engine noise components and their directivity. Based on [56].

We will proceed to present the workings of each empirical model. In the following equations, the parameters indicated with the superscript * are dimensionless. The values used to nondimensionalize the different quantities are (unless otherwise directly stated):

- area - the engine reference area, A_e;
- length - $\sqrt{A_e}$;
- speed - ambient sound speed c_∞;
- density - ambient density ρ_∞;
- temperature - ambient temperature T_∞;

It follows that the nondimensional mean-square pressures , $\langle p^2 \rangle^*$, which will appear in the following sections, are relative to $\rho_\infty^2 c_\infty^4$.

2.6.1 Fan

The fan is one of the most important sources of noise in the engine. It tends to be predominant in the higher frequency range and during landing operations [57]. Its noise comes from two main origins. The periodic motion of the blades generates tonal noise whose spectrum is closely related to the rotation period of the blades. The flow turbulence will also generate noise but usually in a much wider spectrum and so these noise components are called broadband noise.

The empirical method selected to model fan noise was the Heidmann model [58]. This is the most used model [22] and allows the prediction of both fan and compressor stages [30]. The total noise prediction is done summing the contributions from six different components [58]:

1. **Inlet Broadband Noise**, originated by turbulence in the inlet duct which can be caused by boundary layer separation, blade wakes and turbulent inlet flow;

2. **Inlet Rotor-Stator Interaction Tones**, originated by the lift fluctuations on rotor and stator blades due to their aerodynamic interaction. This can include not only the effect of rotor wakes intersecting the stator wakes but also wakes that originated in stator vanes or inlet guide vanes;

3. **Inlet Flow Distortion**, accounts for the noise generation due to lift fluctuations in the rotor and stator blades from inlet turbulence associated with aircraft rotation or ground effects;

4. **Combination Tone Noise**, which occurs when the rotor tip exceeds Mach 1. This creates shock waves that propagate in the inlet duct. Th noise from this origin is also commonly called buzz saw noise. Its spectrum distribution will be along harmonics of the shaft speed;

5. **Discharge Broadband Noise**, also originated by flow turbulence but in the discharge duct;

6. **Discharge Rotor-Stator Interaction Tones**, originated by the same methods as the inlet Rotor-Stator Interaction but in the discharge duct.

The contributions from all these components are summed in mean pressure squared basis to obtain the total noise emitted by the fan. It is not imposed that all components have to be used. If the user so decides, some can be neglected resulting in a less precise prediction. This is required when input arguments are missing. The mean pressure squared $\langle p^2 \rangle^*$ from each component is calculated, as a function of frequency and the observer position relative to the fan:

$$\langle p^2 \rangle^* = \frac{A^* \Pi^*}{4\pi \left(r_s^*\right)^2} \frac{D\left(\theta\right) F_f\left(\eta\right)}{\left(1 - M_\infty \cos\theta\right)^4} \, . \tag{2.17}$$

In this expression A^* is the dimensionless fan inlet cross-sectional area, Π^* is the dimensionless acoustic power, r_s^* is the dimensionless distance from the fan to the observer, D is the directivity function, and F_f is the spectrum function. The Doppler factor $\left(1 - M_\infty \cos\theta\right)^4$ accounts for the forward velocity effects. Figure 2.5 shows a diagram with some of the important parameters in the fan model.

The dimensionless distance r_s^* is calculated using the source to observer distance and the engine reference area A_e:

$$r_s^* = \frac{r_s}{\sqrt{A_e}} \, . \tag{2.18}$$

The directivity functions and spectrum functions are specific for each of the six noise components. The directivity function is a function of the polar angle θ and is independent of the azimuth angle. For the inlet components, noise is primarily propagated forwards, that is to say in polar angles between 0º

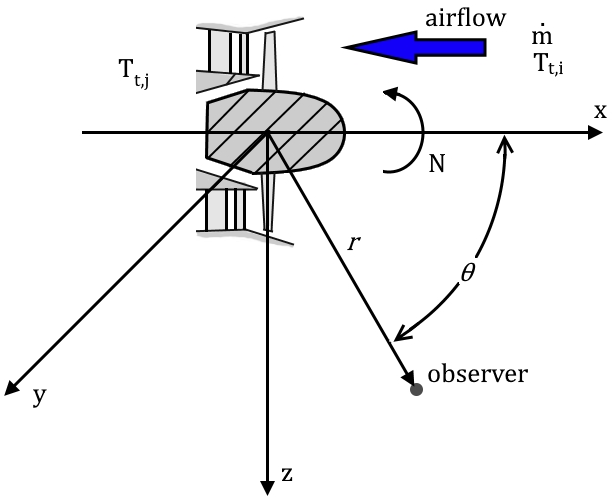


Figure 2.5: Fan diagram and parameters.

and 90º. The opposite happens in the discharge components. The spectrum function depends on the frequency parameter η:

$$\eta = (1 - M_\infty \cos\theta)\,\frac{f}{f_b}\ , \tag{2.19}$$

where f is the frequency, which is an independent parameter, and f_b is the blade passing frequency expressed as a function of the number of blades, B, the rotor diameter, d^*, and the dimensionless rotational speed N^* as:

$$f_b = \frac{N^* B c_\infty}{d^* \sqrt{A_e}}\ . \tag{2.20}$$

For broadband noise components, only one η is required to represent a 1/3 octave band (section 2.7.4). For the tone components, the mean pressure must be evaluated at discrete frequencies; and all discrete frequencies must be summed in the end to obtain the band value. These discrete frequencies are found by splitting the band in regular frequencies knowing that between the highest (η_u) and the lowest harmonic number (η_l) there are $\eta_u - \eta_l + 1$ tones. These harmonic numbers are given by the equations:

$$\eta_l = \lfloor 10^{-1/20}\eta_c \rfloor + 1\ , \tag{2.21}$$

$$\eta_u = \lfloor 10^{1/20}\eta_c \rfloor\ , \tag{2.22}$$

where $\lfloor\rfloor$ is the floor function and η_c is the η for the band center frequency.

The acoustic power for each fan component is calculated by the expression:

$$\Pi^* = KG\,(i,j)\,(s^*)^{-a(k,l)}\,M_m^b\,\frac{\dot{m}^*}{A^*}\,(\Delta T^*)^2\,P\,(M_r, M_m)\ . \tag{2.23}$$

The several parameters in this equation will be covered next. K and b are simple empirical constants and are tabulated for each noise component. G is a constant that reflects the effect of inlet guide vanes

and tone cut-offs, which expresses the importance of the fundamental blade passing frequency. These two effects are interconnected and as such G is provided as a two by two matrix. The correct value to use from this matrix is selected using the indices i and j. Index i reflects the presence of the inlet guide vanes and as such is simply (in a zero-based index system):

$$i = \begin{cases} 0, & \text{Without Inlet guide vanes} \\ 1, & \text{With Inlet guide vanes} \end{cases} . \tag{2.24}$$

Index j, on the other hand, involves a few more steps to compute because the fundamental blade passing frequency depends on the blade tip Mach number. Tip Mach is directly calculated from the dimensionless rotational speed N^*,

$$M_t = \pi N^* . \tag{2.25}$$

From M_t the cut-off factor δ is calculated :

$$\delta = \begin{cases} \dfrac{M_t}{\left|1 - \dfrac{V}{B}\right|} & M_t \leq 1.05 \\ M_t & M_t > 1.05 \end{cases} , \tag{2.26}$$

where V is the number of stator vanes and B the number of rotor blades. Finally, the cut-off index j (in a zero-based index system), required to select G, is obtained from δ,

$$j = \begin{cases} 0 & \delta > 1.05 \\ 1 & \delta \leq 1.05 \end{cases} . \tag{2.27}$$

The next term in equation (2.23), s^*, is the dimensionless rotor-stator spacing using the chord length as reference dimension. Its exponent expresses the acoustic power dependency on the rotor-stator spacing for each component. Like G, it depends on two factors and as such it is selected from a two by two matrix with two indexes, k and l. As i and j, k and l are also presented here in a zero-based index system. Index k relates to the rotor-stator spacing directly,

$$k = \begin{cases} 0 & s^* \leq 1 \\ 1 & s^* > 1 \end{cases} . \tag{2.28}$$

Index l relates to flow distortion. Inlet flow distortion reduces rotor-stator noise components and occurs when the engine is static or during ground roll.

$$l = \begin{cases} 0, & \text{No flow distortion} \\ 1, & \text{With flow distortion} \end{cases} . \tag{2.29}$$

Next, M_m is the design Mach number point, which is defined as:

$$M_m = max\left(1, M_d\right) \ , \tag{2.30}$$

where M_d is the design value for the relative tip Mach number. The mass flow rate, $\dot{m}^*$, and the total temperature rise across the fan, ΔT^*, are input parameters and together with N^* they define the engines running conditions, which may vary in time. Finally, we have the power function P. Each fan noise component has a different power function but all depend on M_m, as previously defined, and the relative Mach number M_r which is given by

$$M_r = \left(M_t^2 + M_x^2\right)^{1/2} \ , \tag{2.31}$$

with M_x being equal to $\dot{m}^*/A^*$. The full set of power functions is not presented here for brevity but it is available in [30].

2.6.2 Turbine

We now present the turbine noise model. This implementation was originally developed by the General Electric Company [59]. Its prediction is made adding only 2 noise components: Broadband noise and a Pure Tone Noise. This simple model is not very accurate but currently, turbine research is still ongoing and accurate models do not exist [22]. The mean square pressure $(\langle p^2 \rangle^*)$ for each turbine component is calculated by the same equation (eq. (2.17)) as for the fan. The directivity function in this equation is given as simple tables for each component as a function of the polar angle θ. Spectrum functions are also tabulated as a function of η. The frequency parameter η and its input, the blade passing frequency are calculated by the same equations as for the fan, respectively equations (2.19) and (2.20). For the turbine, the acoustic power in equation (2.17) is calculated as:

$$\Pi^* = K \left(\frac{h_{t,i}^* - h_{s,j}^*}{h_{t,i}^*} \right)^a \left(M_t\right)^b \ , \tag{2.32}$$

where K, a and b are empirical constants dependent on the component [30], M_t is as defined in equation (2.25), and h^* is the dimensionless specific enthalpy. Subscripts t and s represent total and static conditions, respectively. Subscript i indicates that the value is at the turbine entrance and subscript j at the exit. The main parameters for the turbine model are represented in figure 2.6.

Thermodynamics conversion

There is usually limited engine data available at early design stages. The temperature at different stages is usually known due to their importance in the thermodynamic analysis of a jet engine. For this reason, temperature is used as an input in the turbine source model. However, the acoustic power components in the empirical turbine model are given as a function of specific enthalpy (eq. (2.32)) . Therefore, a conversion is needed from the input temperatures to the required specific enthalpies. This is achieved using [30]:

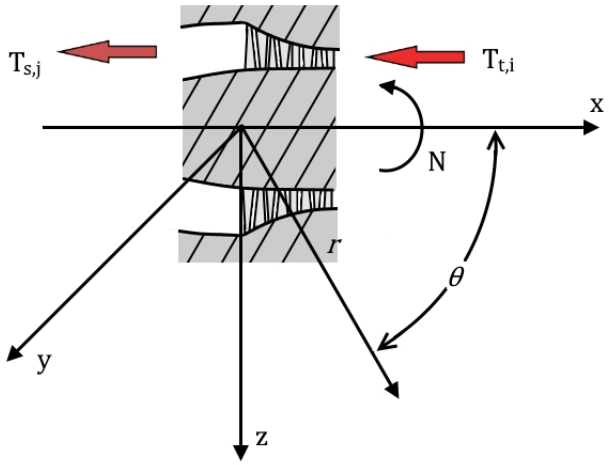


Figure 2.6: Turbine diagram and parameters.

$$h^*\left(T^*\right) = \frac{\left[h_r^*\left(T^* T_\infty^*\right) - h_r^*\left(T_\infty^*\right)\right]}{T_\infty^*} \,, \tag{2.33}$$

where h_r^* is function of the non-dimensional temperatures T^* and given by:

$$h_r^* = \begin{bmatrix} R_{N_2}^* x_{N_2} & R_{O_2}^* x_{O_2} & R_{CO_2}^* x_{CO_2} & R_{H_2O}^* x_{H_2O} & R_{Ar}^* x_{Ar} \end{bmatrix} \begin{bmatrix} h_{rN_2}^* \\ h_{rO_2}^* \\ h_{rCO_2}^* \\ h_{rH_2O}^* \\ h_{rAr}^* \end{bmatrix} \,. \tag{2.34}$$

In this equation h_r^* is provided as a function of temperature from an empirical table [30], x is the mass reaction of each gas and R^* is the dimensionless gas constant given for each gas by:

$$R^* = \frac{1}{R}\frac{\overline{R}}{W} \,, \tag{2.35}$$

where $\overline{R}$ is the universal gas constant, R the dry air gas constant and W the molecular weight for each gas. These five elements are used to calculate the specific enthalpy because together N_2, O_2, CO_2, and Ar constitute 99.03974% (by volume) of the atmosphere (dry air), while water can vary but it is usually in the order of 1 %. [60]. Because they constitute most of the atmosphere, the enthalpy calculated by this way should be a good approximation of the real value. Secondly, O_2, CO_2 and H_2O must be included because of their prevalent role in a combustion reaction [61] and as such, if the combustion's impact to the air composition is to be factored in, these gases must be considered.

2.6.3 Combustion chamber

The core engine noise is based on the investigation by Kazim et al. [62] The core model's mean square pressure is given by:

$$\langle p^2 \rangle^* = \frac{A^* \Pi^*}{4\pi \left(r_s^*\right)^2} \frac{D\left(\theta\right) F_c\left(f\right)}{\left(1 - M_\infty \cos\theta\right)^4} \ . \tag{2.36}$$

The sole difference between equations (2.36) and (2.17) is that, for the combustion chamber, the spectral function F_c is given as a function of frequency f and not the frequency parameter η.

The acoustic power Π^* in equation (2.36) is given by:

$$\Pi^* = \left(8.85 \times 10^{-7}\right) \frac{\dot{m}_i^*}{A^*} \left(\frac{T_j^* - T_i^*}{T_i^*}\right)^2 \left(p_{t,i}^*\right)^2 \left(\Delta T_{des}^*\right)^2 \ . \tag{2.37}$$

Subscripts i and j correspond to the entry and exit conditions respectively, as for the turbine nomenclature. T^* is the total temperature, and p^* the total temperature. $\dot{m}^*$ is the mass flow rate. ΔT_{des}^* is the design temperature drop in the turbine. As such the combustor noise depends on the turbine conditions.

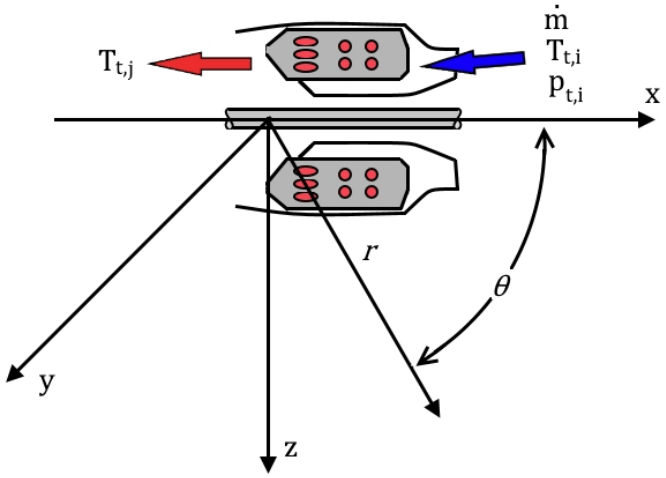


Figure 2.7: Combustor diagram and parameters.

The directivity and the spectrum functions are again given by tabulated functions [30]. The input to the spectrum function is $\log f/f_p$. The peak frequency f_p is defined as:

$$f_p = \frac{400}{1 - M_\infty \cos\theta} \ . \tag{2.38}$$

2.6.4 Jet

Finally, we address the exhaust jet noise. This source is very important in the lower frequency range and is prominent at take-off. There are many jet models available. The one selected was Stone's jet noise model [63, 64] because it has extensive data available and allows for the modeling of both single jet and coaxial jets. As such turbofans can be modeled which is not possible in some other models. Additionally, this model accounts for effects of design details in chevrons and plugs. This model has some limitations namely the primary jet velocity needs to be greater than the secondary jet velocity and only the primary jet can be supersonic. This was considered acceptable because it goes in accordance with the typical behavior of turbofans. The Stone jet model implemented is valid for bypass ratios up to

6. Newer models such as ST2JET are valid up to bypass ratios of 14.9, and they could be implemented based on the same framework. The noise prediction is split into two contributions: Mixing Noise and Shock Noise.

Jet Mixing Noise

Jet mixing noise originates from the mixing of jet streams and the free flow and its shear stresses. The mean square pressure is calculated by:

$$\langle p^2\left(r_s^*,\theta\right)\rangle^* = \frac{\langle p^2\left(\sqrt{A_e},90^o\right)\rangle^*}{\left(r_s^*\right)^2}\left[\frac{1+\left(0.124V_1^*\right)^2}{\left(1+0.62V_1^*\cos\theta\right)^2+\left(0.124V_1^*\right)^2}\right]^{3/2} \times D_m\left(\theta'\right)F_m\left(S_m,\theta'\right)H_m\left(M_\infty,\theta,V_1^*,\rho_1^*,T_1^*\right)G_cG_p\ . \quad (2.39)$$

As in the fan noise model this expression has many factors which we now explain. The dimensionless source to observer distance r_s^* is defined as in equation (2.18). V_1, T_1 and ρ_1 are respectively the velocity temperature and density of one jet. The subscript 1 refers to the primary (hot) jet and 2 to the secondary (cold) jet. θ is the polar angle as it has been previously defined and θ' is the modified directivity angle which corresponds to:

$$\theta' = \theta\left(V_1^*\right)^{0.1}\ . \quad (2.40)$$

Figure 2.8 shows the coordinate system and the stream indexes used.

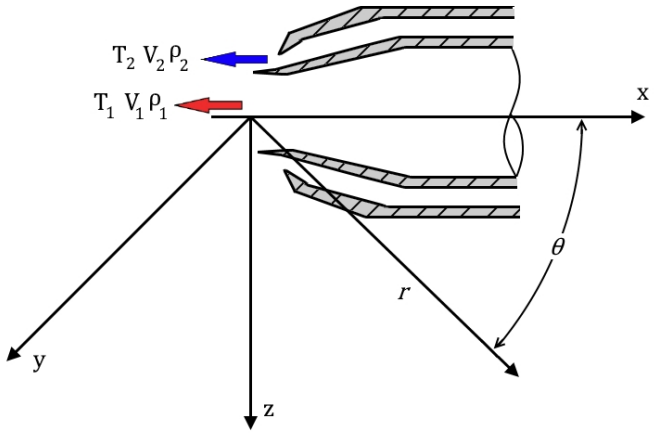


Figure 2.8: Jet diagram and parameters.

$\langle p^2\left(\sqrt{A_e},90^o\right)\rangle^*$ is a reference mean square pressure which is evaluated by:

$$\langle p^2\left(\sqrt{A_e},90^o\right)\rangle^* = \frac{2.502\times10^{-6}A_{j,1}^*\left(\rho_1^*\right)^{\omega_0}\left(V_1^*\right)^{7.5}}{\left[1+\left(0.124V_1^*\right)^2\right]^{3/2}}\ , \quad (2.41)$$

where $A_{j,1}^*$ is the primary jet fully expanded area. The density exponent ω_0 is provided by the expression:

$$\omega_0 = \frac{2\left(V_1^*\right)^{3.5} - 0.6}{\left(V_1^*\right)^{3.5} + 0.6} \ . \tag{2.42}$$

D_m is the directivity function for the mixing noise and is given as a tabulated function of the modified directivity angle [63]. The spectral distribution function F_m is an empirical tabulated function [63], which depends on both the modified directivity angle and the mixing jet noise Strouhal number S_m, which is defined as:

$$S_m = \frac{f^* d_{j,1}^* \left[1 - M_\infty \cos\left(\theta - \delta\right)\right] \left(T_1^*\right)^{0.4\left(1+\cos\theta'\right)}}{V_1^* \left(1 - M_\infty / V_1^*\right)} \times \left\{ \frac{\left[1 + 0.62\left(V_1^* - M_\infty\right)\cos\theta\right]^2 + \left[0.124\left(V_1^* - M_\infty\right)\right]^2}{\left(1 + 0.62 V_1^* \cos\theta\right)^2 + \left(0.124 V_1^*\right)^2} \right\}^{1/2} g_c g_p \ , \tag{2.43}$$

where f^* is the Helmholtz number given by:

$$f^* = \frac{f\sqrt{A_e}}{c_\infty} \ , \tag{2.44}$$

and $d_{j,1}^*$ is the primary jet diameter

$$d_{j,1}^* = \sqrt{\frac{4A_{j,1}^*}{\pi}} \ . \tag{2.45}$$

The configuration factors g_c and g_p depend on the engine geometry and reflect its effects in the mixing Strouhal number. The presence of a plug in the nozzle is accounted for with the g_p factor and the coaxial or single jet options are reflected in g_c as follows:

$$g_p = \begin{cases} \left(R_d\right)^{0.4}, & \text{Nozzle with plug} \\ 1, & \text{Nozzle without plug} \end{cases} \ , \tag{2.46}$$

$$g_c = \begin{cases} \left(1 - T_2^* f_s / T_1^*\right)^{-1}, & \text{Coaxial Nozzle} \\ 1, & \text{Single Nozzle} \end{cases} \ , \tag{2.47}$$

where R_d is the ratio of the jet diameters:

$$R_d = \frac{d_{h,1}^*}{d_{e,1}^*} \ , \tag{2.48}$$

and f_s is the frequency shift and is a tabulated function [63] whose inputs are $1 - A_{j,2}^*/A_{j,1}^*$ and V_2^*/V_1^*.

The G_p and G_c factors, in equation (2.39), are analogue to g_c and g_p, and reflect the engine's design in the mean acoustic pressure. The plug factor, G_p, is given by

$$G_p = \begin{cases} \left(0.10 + \frac{2R_d^2}{1 + R_d^2}\right)^{0.3}, & \text{Nozzle with plug} \\ 1, & \text{Nozzle without plug} \end{cases} \ , \tag{2.49}$$

while the coaxial/single jet factor, G_c, is given by

$$G_c = \begin{cases} \left(\frac{T_1^*}{T_2^*}\right)^{1/2} \left\{ \left(1 - V_2^*/V_1^*\right)^m + \frac{1.2\left[1 - A_{j,2}^* \left(V_2^*\right)^2 / A_{j,1}^* \left(V_1^*\right)^2\right]^4}{\left(1 - A_{j,2}^*/A_{j,1}^*\right)^3} \right\}, & \text{Coaxial Nozzle} \\ 1, & \text{Single Nozzle} \end{cases} , \quad (2.50)$$

where the m exponent is calculated from the fully expanded jet areas

$$m = \begin{cases} 1.1\sqrt{A_{j,2}^*/A_{j,1}^*}, & A_{j,2}^*/A_{j,1}^* \leq 29.7 \\ 6, & A_{j,2}^*/A_{j,1}^* \geq 29.7 \end{cases} . \quad (2.51)$$

Finally, to evaluate equation (2.39) the factor H_m needs to be calculated. This accounts for the forward flight effects:

$$H_m\left(M_\infty, \theta, V_1^*, \rho_1^*, T_1^*\right) = \left\{ \frac{\left(1 + 0.62 V_1^* \cos\theta\right)^2 + \left(0.124 V_1^*\right)^2}{\left[1 + 0.62\left(V_1^* - M_\infty\right)\cos\theta\right]^2 + \left[0.124\left(V_1^* - M_\infty\right)\right]^2} \right\}^{3/2} \times \frac{\left(1 - M_\infty/V_1^*\right)^5 \left(\rho_1^*\right)^{\omega - \omega_0}}{\left(1 - M_\infty \cos\left(\theta - \delta\right)\right)} . \quad (2.52)$$

The exponent $\omega - \omega_0$ is

$$\omega - \omega_0 = \frac{1.8\left\{\left[V_1^*\left(1 - M_\infty/V_1^*\right)^{2/3}\right]^{3.5} - \left(V_1^*\right)^{3.5}\right\}}{\left\{0.6 + \left[V_1^*\left(1 - M_\infty/V_1^*\right)^{2/3}\right]^{3.5}\right\}\left[0.6 + \left(V_1^*\right)^{3.5}\right]} . \quad (2.53)$$

Because it reflects the forward flight effects, H_m is equal to 1 when M_∞ is zero.

Jet Shock Noise

The Jet Shock noise component predicts the noise generated from the primary jet expansion shocks and its interaction with the flow's turbulence. As in all other models, noise is given in a mean-square pressure square format by:

$$\langle p^2 \rangle^* = \frac{\left(3.15 \times 10^{-4}\right) A_{j,1}^*}{\left(r_s^*\right)^2} \frac{\beta_s^4}{1 - \beta_s^4} \frac{F_s\left(S_s\right) D_s\left(\theta, M_1\right) G_c}{1 - M_\infty \cos\left(\theta - \delta\right)} , \quad (2.54)$$

where β_s is the pressure ratio parameter, F_s is the shock spectral distribution function, S_s the shock Strouhal number, D_s is the shock directivity function and G_c is the coaxial/single jet configuration factor, the same that was obtained for the mixing noise component. The pressure ratio parameter is the determining factor in deciding if shock noise occurs or not. The β_s parameter is computed by:

$$\beta_s = \left(M_1^2 - 1\right)^{1/2} . \quad (2.55)$$

If β_s is not superior to 0, which is to say $M_1 \leq 1$, then no shock noise occurs.

The spectral function F_s is a tabulated function [63] which depends only on the shock Strouhal number given by:

$$S_s = \frac{f^* d^*_{h,1}}{0.70 V_1^*} \beta \left[1 - M_\infty \cos\left(\theta - \delta\right)\right] \left[\left(1 + 0.70 V_1^* \cos\theta\right)^2 + \left(0.14 V_1^*\right)^2\right]^{1/2} , \tag{2.56}$$

where f^* is the Helmholtz number as defined in Eq. (2.44). The directivity function, D_s, is a simple branch function; its branches are separated by the Mach angle θ_m, which is given by:

$$\theta_m = \arcsin\left(1/M_1\right) , \tag{2.57}$$

With the Mach angle, the directivity function is simply evaluated with

$$D_s\left(\theta, M_1\right) = \begin{cases} 1, & \theta \leq \theta_m \\ 1.189, & \theta > \theta_m \end{cases} . \tag{2.58}$$

2.7 Signal Processing

2.7.1 Frequency Domain, Fourier Transform, and Short Time Fourier Transform

Acoustic signals' information is most useful not in the time domain but in the frequency domain. Converting signals from the time domain to the frequency domain is usually done by the use of Fourier's Transform. The Discrete Fourier Transform can be defined as [65]:

$$X(p) = \sum_{n=0}^{N-1} x(n) e^{-i\frac{2\pi}{N}np}, \quad p \in \{0, 1, 2..., N-1\} , \tag{2.59}$$

where N is the number of samples, x is the discrete signal in the time domain and X is the signal in the frequency domain. Using this formulation the number of operations required for the computation of the Fourier transform is in the order of N^2.

While this method can suffice for a small number of points, when the signal has a larger number of points this conventional method for calculating the transform is replaced by the Fast Fourier Transform (FFT) [66]. The FFT method only requires a number of operations in the order of $Nlog_2N$. This algorithm has the extra advantage of being more precise because the number of computations is lower which leads to a smaller round-off error [65]. Because in this project we can calculate the signal emitted by the aircraft along a long flight path (an thus with a large number of points in the time domain), the FFT is the best method to convert the signal to the frequency domain.

Formulation 1A will return pressures in function of time. The semi-empirical models, on the other hand, return pressure for each frequency band in function of time. If these two signals are to be combined then the Fourier Transform cannot be applied to the whole time domain. Instead what must be used is the

Short-Time Fourier Transform. In its discrete form it can be expressed as [65]:

$$X(p) = \sum_{n=0}^{N-1} x(n)w(n-m)e^{-i\frac{2\pi}{N}np}, \quad p \in \{0, 1, 2..., N-1\} \ , \tag{2.60}$$

It differs from equation (2.59) in that the signal x is multiplied by a window function, w . The window function will effectively restrict the signal to a small time frame and make it zero outside the window. The window function can be selected from a wide range of possibilities. The duration of this window will depend on the desired signal resolution in each domain. Short windows give good resolution in the time domain and bad resolution in the frequency domain and vice-versa [67]. If the FFT is applied to the resulting filtered function, the spectrum information across time is obtained.

2.7.2 Nyquist–Shannon sampling theorem

To properly capture a signal's frequencies the sample rate must be adjusted to capture all the frequencies. The Nyquist-Shannon sampling theorem sets the minimum number of samples in the time domain that are required to properly describe the signal. The theorem states the sampling rate required must be at least twice the highest frequency [68].

This answers the problem raised in section 2.4 of determining the time step (Δt^*) we require when computing the interpolation of the noise signal to the common set of times.

2.7.3 Noise Levels

The signal in the frequency domain has the amplitude in Pascal (Pa). This unit is not the most appropriate to analyze the signal, which is usually done on a logarithmic scale. The other scales which will be present here are: Sound Pressure Level (SPL), Overall Sound Pressure Level (OASPL), Perceived Noise Level (PNL), Tone Corrected Perceived Noise Level(PNLT), and Effective Perceived Noise Level (EPNL). PNL and EPNL are especially important because they are essential for technical and legal evaluations of aircraft emissions [56].

Sound Pressure Level

The Sound Pressure Level is simply computed by converting the pressure values in Pa to the logarithmic decibel scale by

$$SPL(dB) = 20 \log \left[\frac{p}{p_{ref}} \right] \ , \tag{2.61}$$

where p is the pressure in Pa and p_{ref} is the reference pressure set equal to 2×10^{-5} Pa [46]. This value is used as reference pressure because it is considered the lower threshold of human hearing. In the engine source models, the pressure values are non-dimensional using $\rho_a c_a^2$ as reference. In this case

equation (2.61) can be rewritten as

$$SPL(dB) = 10\log\left\langle p^2\right\rangle^* + 20\log\left[\frac{\rho_a c_a^2}{p_{ref}}\right] . \tag{2.62}$$

Overall Sound Pressure Level

SPL converted pressure values in Pascal to decibels but these values can correspond to several magnitudes in different frequencies. To aggregate the contributions across the spectrum into a single term the Overall Sound Pressure Level is used

$$OASPL(dB) = 10\log\sum_{i=1}^{n}\left\langle p^2\right\rangle_i^* + 20\log\left[\frac{\rho_a c_a^2}{p_{ref}}\right] , \tag{2.63}$$

where $\sum_{i=1}^{n}$ is the sum of pressures across all frequency bands.

Perceived Noise Level

The Perceived Noise Level takes into account the fact that different frequencies have different impacts to the listener. This is achieved by using a weighting factor, the noisiness factor N, measured in noys. The noisiness factor depends not only on the sound pressure level but also on the frequency it corresponds to.

$$N = \begin{cases} 0, & SPL < A_1 \\ 10^{B_1(SPL-A_1)-1}, & A_1 \leq SPL \leq A_2 \\ 10^{B_2(SPL-A_3)}, & A_2 \leq SPL \leq A_3 \\ 10^{B_3(SPL-A_3)}, & A_3 \leq SPL \leq A_4 \\ 10^{B_5(SPL-A_5)}, & A_4 \leq SPL \leq 150 \end{cases} . \tag{2.64}$$

The SPL value in equation (2.64) is a value obtained from either equation (2.61) or equation (2.62) for a specific frequency band. The values A_j and B_j are tabulated constants defined in function of frequency.

After noisiness values were calculated for all bands the total noys are calculated as

$$N_t = N_{max} + 0.15\left[\left(\sum_{i=1}^{24} N_i\right) - N_{max}\right] , \tag{2.65}$$

where N_{max} is the maximum noy value and $\sum_{i=1}^{24} N_i$ is the summation of the noy values across all 1/3 octave bands (section 2.7.4). Finally, the Perceived Noise Level is given by:

$$PNL = 40 + 33.22\log N_t . \tag{2.66}$$

This value is in a logarithmic scale and the units are PNdB.

Tone Corrected Perceived Noise Level

Tone Corrected Perceived Noise Level builds upon PNL adding a new correction. This correction accounts for the fact that pure tones provoke more irritation than broadband noise. First, the pure tones must be found in the 1/3 octave bands. For each band i the value ΔD_i is computed

$$\Delta D_i = SPL_{i+1} - 2 \times SPL_i + SPL_{i-1} \ . \tag{2.67}$$

If ΔD_i is inferior to -5 then the SPL of band i is tested to be a local maximum against bands $i+1$ and $i-1$. If this is the case then the pure tone impact is considered to be effective in the band. Next, the background SPL is found as an average of the neighbor bands

$$\overline{SPL} = \frac{SPL_{i+1} + SPL_{i-1}}{2} \ . \tag{2.68}$$

From here the correction factor for band i can be determined as a function of the band's frequency f_i and the difference F between the local SPL maximum and the calculated $\overline{SPL}$ is:

$$F = SPL_i - \overline{SPL_i} \ . \tag{2.69}$$

The correction factor for the presence of a tone in band i is then calculated as:

$$C_i(f_i, F) = \begin{cases} if F < 3, & 0 \\ if 3 \leq F < 20 & \begin{cases} F/3, if\ (500 < f_i < 5000) \\ F/6, if\ (f_i \leq 500 \vee f_i \geq 5000) \end{cases} \\ if F \geq 20 & \begin{cases} 6.7, if\ (500 < f_i < 5000) \\ 3.3, if\ (f_i \leq 500 \vee f_i \geq 5000) \end{cases} \end{cases} \ . \tag{2.70}$$

A correction value C_i is computed for each band where a pure tone was found. Of all these, the maximum is selected. Finally, the tone-corrected perceived noise level can be calculated

$$PNLT = PNL + C_{max} \ . \tag{2.71}$$

Effective Perceived Noise Level

Last we address the Effective Perceived Noise Level. This is a parameter calculated by integrating the noise in the time domain [30]:

$$EPNL = 10log \int_{t_i}^{t_f} \langle p_{pnlt}^2 \rangle \, dt \ , \tag{2.72}$$

where $\langle pnlt^2 \rangle$ is the mean pressure equivalent to the PNLT (as given by eq. (2.71)) for each time step, and is computed by:

$$\langle p_{pnlt}^2 \rangle = 10^{PNLT/10} \ . \tag{2.73}$$

2.7.4 One Third Octave Bands

Analysis of signals in the full frequency spectrum continuum is not usually done, instead, the frequencies are divided into a set of bands. For this there are two choices: constant absolute bandwidth bands or constant relative bandwidth bands [69]. The first divides the spectrum continuum in bands whose frequency range is constant for all bands. This is the least used method. Constant relative bandwidth bands allow separating the frequency spectrum in octaves, one of the basic intervals in sound. This is an appropriate way to represent the frequency distribution of the signal because separating the spectrum linearly in the logarithmic scale is usually easier to interpret. For all constant relative bandwidth, it applies that:

$$\log f_c = \frac{\log f_u + \log f_l}{2} \Longrightarrow f_c = \sqrt{f_u \cdot f_l}\ , \tag{2.74}$$

where f_l, f_c, f_u are respectively the lower, centre and upper frequencies in a specific band. The relative bandwidth in percentage, which must remain constant, is given by

$$B(\%) = \frac{f_u - f_l}{f_c} \times 100\ , \tag{2.75}$$

Octave bands are a specific case of constant relative bandwidth bands where $f_u = 2f_l$. In source models, noise limits and noise metrics are given not for octaves bands but for 1/3 octave bands. This result in dividing each octave band into three bands again with equal relative bandwidth. For these bands the following specific relations hold

$$\log \frac{f_u}{f_l} = \log 2 \Longrightarrow f_u = 2^{1/3} f_l\ , \tag{2.76}$$

$$f_c = 2^{1/6} f_l\ . \tag{2.77}$$

The full table of the 1/3 octave bands' frequencies is given in Appendix C.

2.8 Noise Limit Certification

The implemented certification standards are ICAO's and FAA's. The FAA limits are found in section B36.5 of FAA, [14] (henceforth addressed simply as Part 36), while ICAO's limits are found in Chapters 2, 3, 4 and 14 of ICAO [11], (henceforth addressed simply as Annex 16). In both these documents, the noise evaluation measure is the EPNdB [11, 14]. The maximum values for the noise level are set according to the class and weight of the aircraft. The limits are defined for three reference measure points:

- Lateral;
- Flyover;
- Approach.

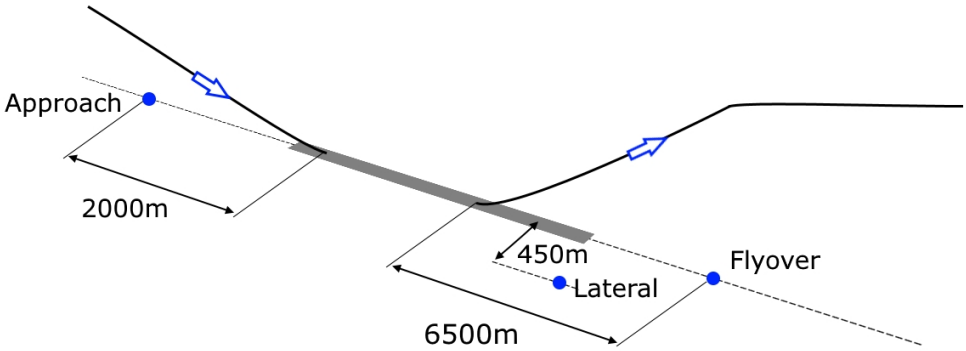


Figure 2.9: Microphone/Observer positions in noise certification. Based on [70]

A schematic with these points is represented in figure 2.9.

The two regulations' limits are very similar. In Part 36 section B36.5 f) Stage 4 limits are expressly stated as the same as in Annex 16's Chapter 4, and according to [14] the same thing applies for Stage 5 and Chapter 14. As for Stages 2 and 3 they are almost identical to Chapters 2 and 3 with only minor differences. There are a few additional details to be taken into account:

- In Chapters 2 and 3 a system of trade-offs is set in case some noise measures surpass the maximum limits set. The correspondent Stages in Part 36 have equally defined trade-offs;
- Chapter 4's maximum noise levels are the ones defined in Chapter 3 but instead of a trade-off system a set of restrictive margins are defined for each individual measure; for the sum of the differences between each measure; and the respective maximum noise level;
- Chapter 14 defines a new set of limits for maximum noise levels and like Chapter 4 it sets restrictive margins for each noise measure; and also for the sum of the gaps between each noise measure and the respective maximum noise level.

These margins have to be taken into account because they are effectively used to set stricter noise emissions limits in the legislation.

Chapter 3

Implementation

This software module consists of a Dynamic-link library (DLL) developed using C++ [71]. Compatibility with the rest of the software framework was the main reason for this choice. The module was developed in the Visual Studio ® [72] Integrated Development Environment and, as such, the compiler used was for Visual C++ [73] which has limited support for C++. The differences are presented in [74].

The module was named Acoustic Prediction Module (APM) and this chapter will focus on describing the data structures, algorithms and procedures in APM.

3.1 Module Structure

A schematic of the program flow, separated in its core constituents, is presented in Fig. 3.1. Groups presented in parallel do not have precedence over each other, but all must have outputs for the next stage, which are tested by APM for their viability.

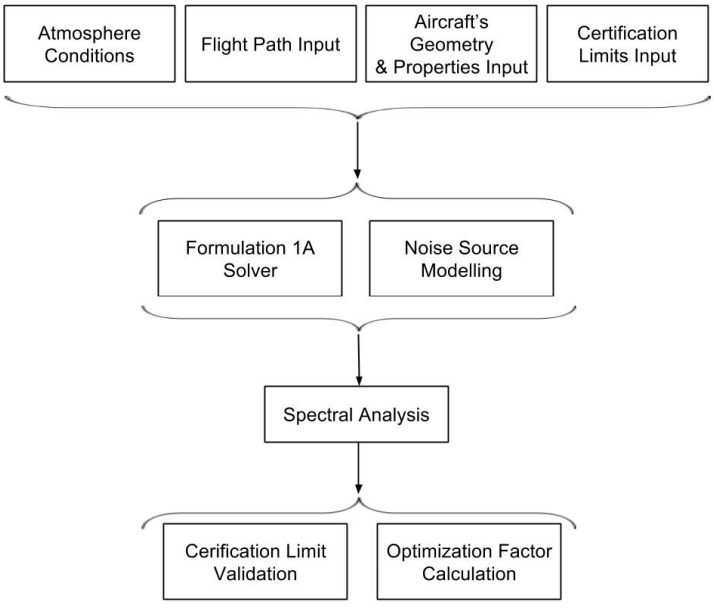


Figure 3.1: APM's architecture.

In this diagram, only the core components for the acoustic calculations are shown, which leaves out the overhead controls, data management and validation, error handlers and the Graphical User Interface (GUI). The APM main window is shown in Fig. 3.2.

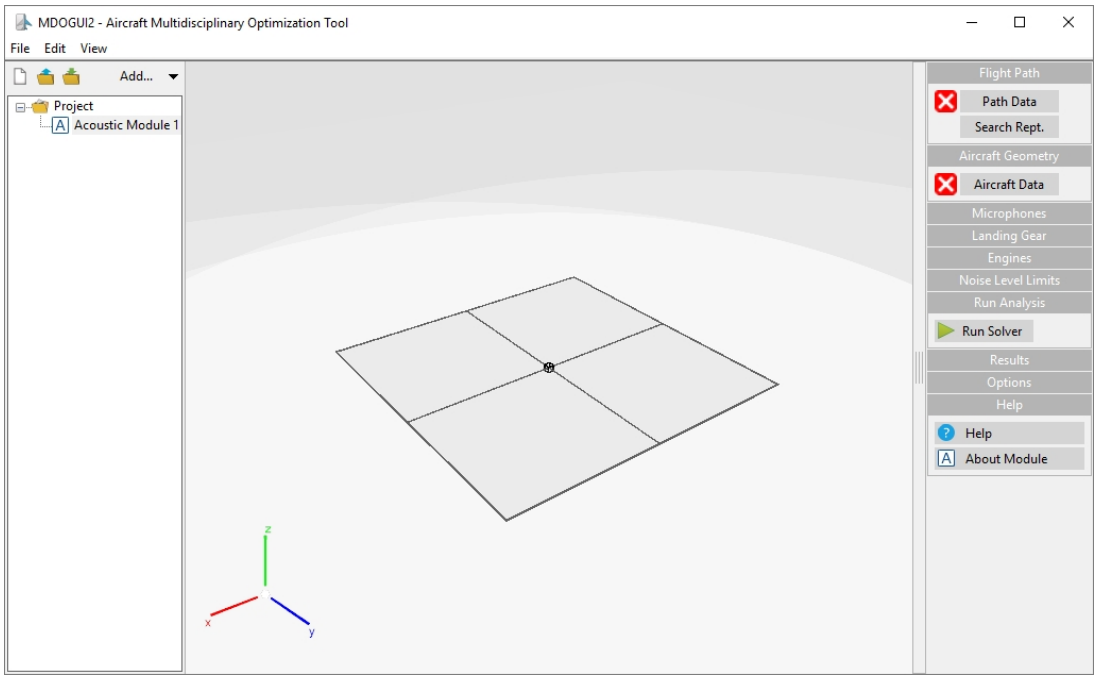


Figure 3.2: APM GUI.

The GUI was designed to be user-friendly and augmented with tool-tips and documentation (with 2 versions, one for users and one for developers). This was considered important because APM will be used by the MDOGUI team and may be revamped or expanded in future projects. Without documentation, maintainability of the module would be unfeasible. That said, APM's front end will not be addressed further in this paper. The sections that follow will focus primarily on the back end implementation. Details on the front end implementation are available in the developer version of APM's documentation.

3.2 Data Structures

In this section, we cover the data structures used in APM. Of course, a software module, integrated into a framework and with GUI, has many classes and structures which only have complementary or support functions to the primary goal of making acoustic predictions. The data structures presented here are a selected few, chosen by one or more of three criteria:

- special relevance to the acoustic computations;
- being critical input data and influencing the input file structure;
- how some data should be stored is not self-evident.

The data required for acoustic computations (sources, path, and observer) is a clear choice because they are the core of the modules' functionality. For the same reason, these are the most important input

files for a user of APM. These selected files will later be addressed in section 3.3. As for data whose representation might not be self-evident, we give the examples of how to represent if the undercarriage is deployed or how the limits in ICAO's Annex 16 are stored.

3.2.1 Flight Path Data

APM calculates the noise generated by an aircraft across a determined flight profile. This flight profile is given by a sequence of discrete states. The aerodynamic solver sees each of these points as an aerodynamic case and thus, henceforth, each of this points in the flight path will be named a *case*.

For each *case* the following information must be known:

- the time it occurs at;
- the aircraft's position and orientation;
- the aircraft's velocities;
- the ambient conditions - temperature, density and sound speed;
- the angle of attack and sideslip angle.

The angle of attack and sideslip angles are kept despite the fact that the aircraft's velocity and orientation are already stored. This is for convenience because they are readily available from a 'Mission Format' file (subsection 3.3.4) and are required to call the Aerodynamic module (Panel Method) (subsection 3.3.6).

For a specific flight path the number of *cases* is fixed and for each *case*, all terms must be known. Thus, the natural way to store the data is in matrices, where it was decided that each column represents a *case*. This matrix representation was implemented, like all other matrix members in APM, using the available matrix class in the MDOGUI software framework `CMatrixLib::FMatrix`.

All quantities have SI units, apart from angles which can be either represented in degrees or radians.

Flight path data requires one additional field. In the current framework, some modules and files handle flight paths that are not continuous. In the transition between segments, there are discontinuities in all the input values. As it was shown in section 2.3, Formulation 1A requires time derivatives. In segment boundaries, the derivatives are meaningless and should not be computed. Thus each *case* in the flight path has an extra number assigned to it. This number labels the segments as it is represented in the next figure.

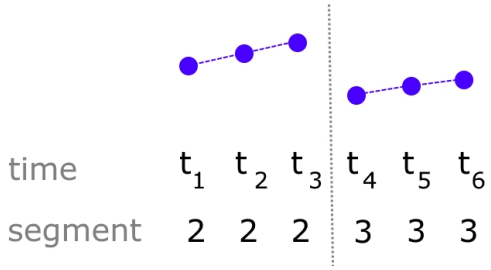


Figure 3.3: Representation of fight path segment discontinuities.

This segment number is tested by the solver to find when a segment ends and another begins. In this manner, the input data can have segments of arbitrary length and the solver will not compute derivatives across segment boundaries.

3.2.2 Aircraft and Surface Pressure Data

Aircraft data pertains to geometry surface information and surface pressure information. The surface of the aircraft is modeled using panels. Each panel can be described by its position (x, y, z), its area and a coordinate system bound to the panel that indicates its orientation in space. Formulation 1A poses no restrictions on the panel shape and therefore it is arbitrary. A generic panel is represented in Fig. 3.4. This is the same discretization as used in other MDOGUI modules, namely the Aerodynamic module.

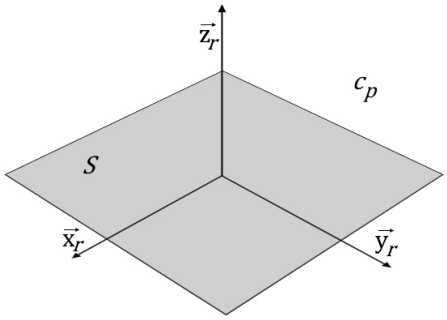


Figure 3.4: Aircraft surface panel and its components.

Surface pressure information is stored as a pressure coefficient (c_p) for each panel in the geometry. This coefficient is given by:

$$c_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty V_\infty^2} \ , \tag{3.1}$$

where p, ρ and V represent pressure, air density and velocity, respectively, while subscript ∞ stands for free stream conditions. These two sets of data, the geometry, and the surface pressure were grouped due to their strong connection. This organization spontaneously arises because:

- c_p's are defined for a panel-*case* combination;
- Formulation 1A requires that all panels must have a c_p associated in each *case*;

- the Aerodynamic module returns them simultaneously when called from APM.

A graphical representation of an aircraft discretized in this way in depicted in Fig. 3.5 which was obtained from the Aerodynamic module (Panel Method).

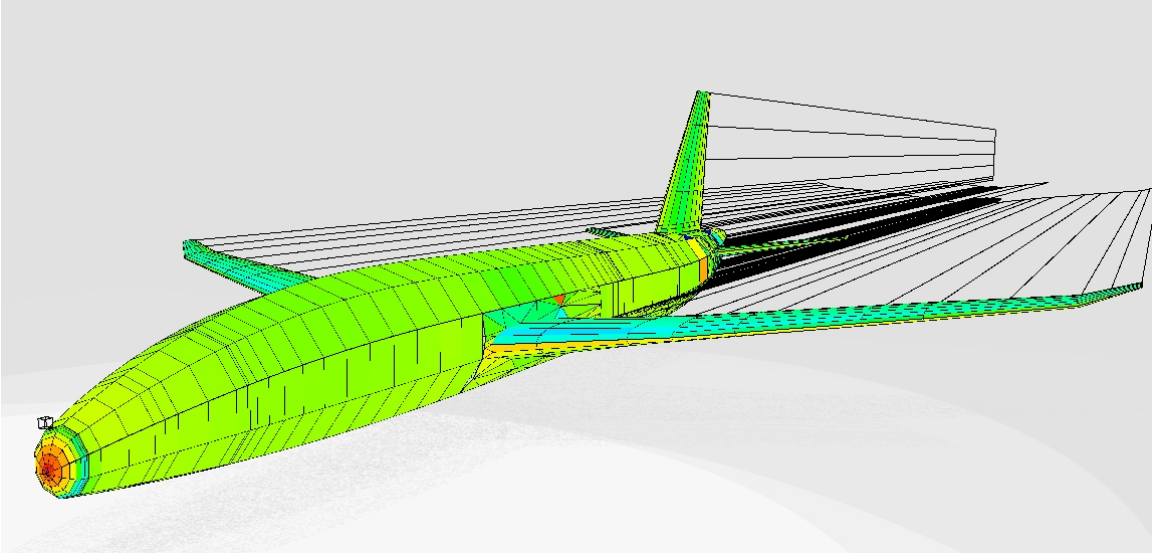


Figure 3.5: Aircraft mesh from the Aerodynamic module. (Color signifies surface pressure).

3.2.3 Microphone Data

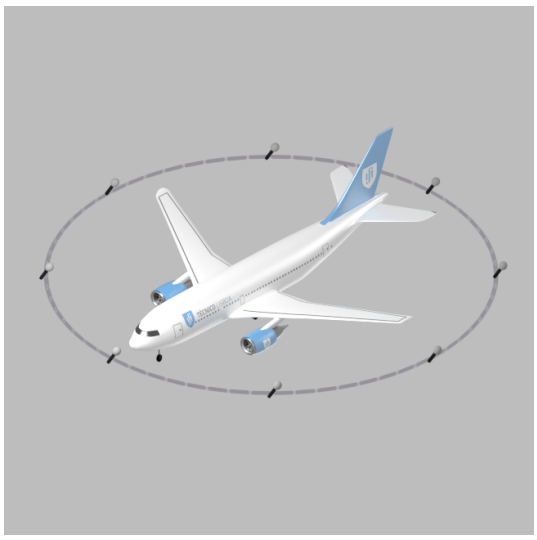
The microphone/observer is essentially a point in space. The point can be defined in a fixed coordinate system or relative to the aircraft's frame of reference. In this last case, one can additionally specify if it should fully track the aircraft's motion or only its translation. This tracking of the aircraft is indicated in the microphone structure as boolean flags and in later sections, it will be addressed how it translates to microphone motion.

The input of the microphone's coordinates can be done in either a Cartesian coordinate system (x, y, z) or spherical coordinate system (r, θ, ϕ). The Cartesian system will typically be more appropriate to input fixed ground observers for instance; while the spherical system is more useful in creating sets of observers placed in a sphere or hemisphere around the source, as is commonly done in acoustics to study noise directivity.

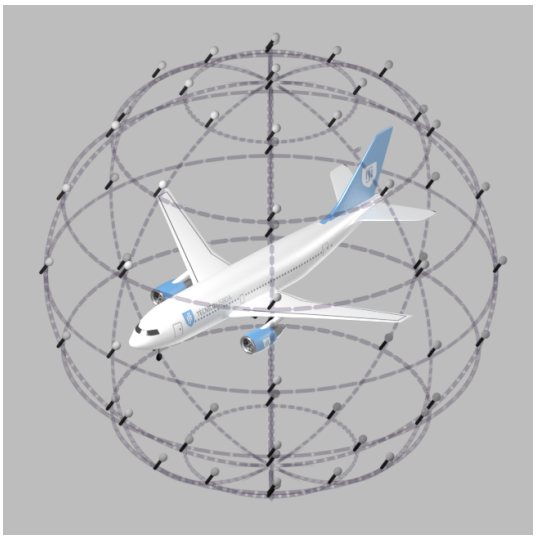
Additionally, each microphone has a tag that indicates if it is one of the points defined in the legislation limits (Approach, Flyover, Lateral). The position of these points will vary with the flight path and, in real world scenarios, it may also be influenced by topography. For this reason, APM can not deduce which observer corresponds to which legislation limit, if the microphones are not tagged. It is the user's responsibility to ensure that the microphones correctly map the legislation points. APM does provide a small utility to assist in the creation of legislation microphones if the user provides a reference point for the runway.

Microphone tracking and the possible input schemes allow creating microphone sets to study many scenarios as is shown in Fig 3.6. If the user wishes to study the directivity of noise emissions configura-

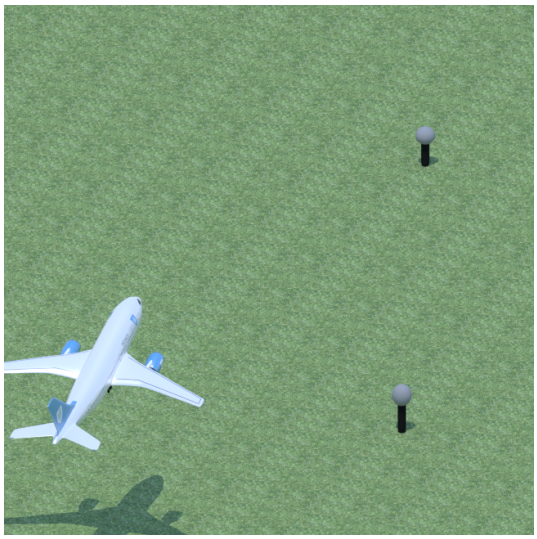
tions such as those represented in *a* or *b* can be used depending on the angular dependencies. If the user is focused on noise at ground level he can, for instance, create only legislation observers as in *c* or meshes of microphones, *d*, which allow creating noise profiles in a certain area, such as around an airport.



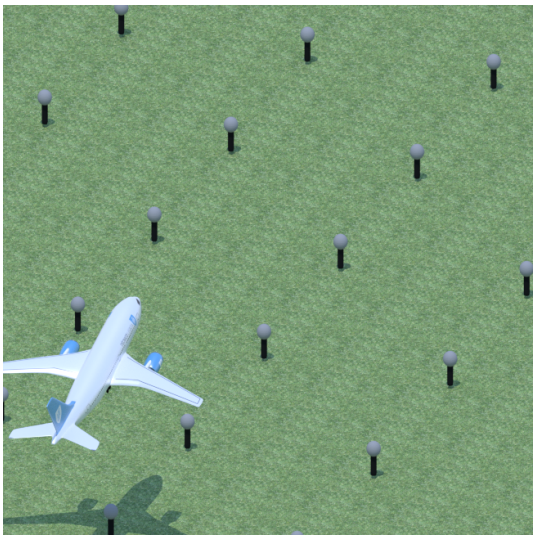
(a) Polar directivity



(b) Polar and azimuthal directivity



(c) Legislation points



(d) Area distribution

Figure 3.6: Possible microphone study configurations in APM.

Finally, each microphone has an ID number. These are automatically generated for new microphones and are unique to each microphone. Their sole purpose is to ease identification of different observers and label the output files appropriately.

3.2.4 Engine Data

As addressed in section 2.6 the engine model is subdivided in many sub-models (*e.g.* fan, turbine, *etc.*). In each of these sub-models, the total noise is computed by adding the different noise components. Each component captures a fraction of the noise generated (*e.g.* the jet mixing noise, or the fan

broadband noise). This leads to an intricate class to fully describe the engine. A natural hierarchy arises from the engine structure and similarities between the models for each component.

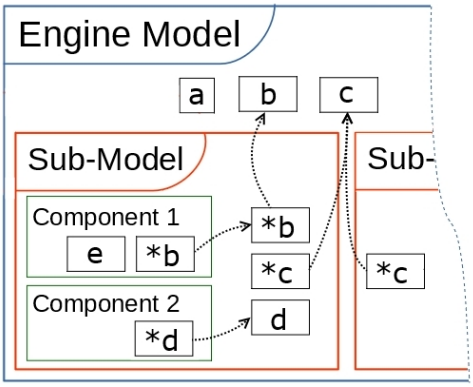


Figure 3.7: Engine model levels scheme.

As is represented in Fig 3.7, not only each level may have its own data, but all must share information with the nested structures. Furthermore, many procedures are equal or similar between different sub-models and components. In order to avoid repetition of implementation, which would lead to redundancy and potential mistakes, the sub-models and the component classes are built with an inheritance system (subtype polymorphism), where shared methods and data are defined in abstract classes (Fig 3.8).

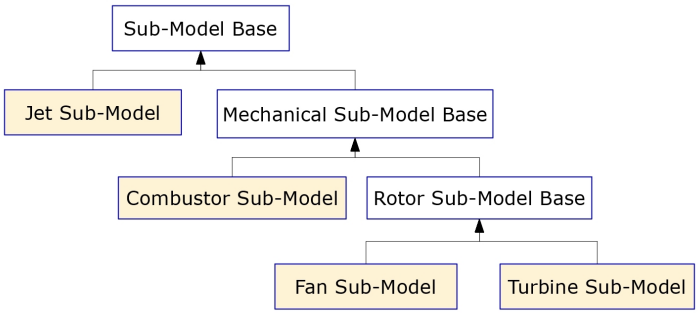


Figure 3.8: Class inheritance tree in engine sub-models.

Due to their higher number of ramifications, components have an even larger inheritance tree. A simplified version is presented in Fig 3.9, where the inheritance of directivity and spectrum modeling is omitted.

The list of all the data stored in these classes is too long to enumerate. However, this data can be subdivided into four main groups:

- **frequency data**, band frequencies from which noise emissions will be calculated
- **operational data**, data, defined by the user, portraying the engine operation during flight (*e.g.* fuel flows, shaft rotational speeds, flow temperatures, *etc*);
- **geometry data**, user defined constants, that define the engine geometry (*e.g.* engine position relative to the aircraft, reference area, number of stator vanes, *etc*);

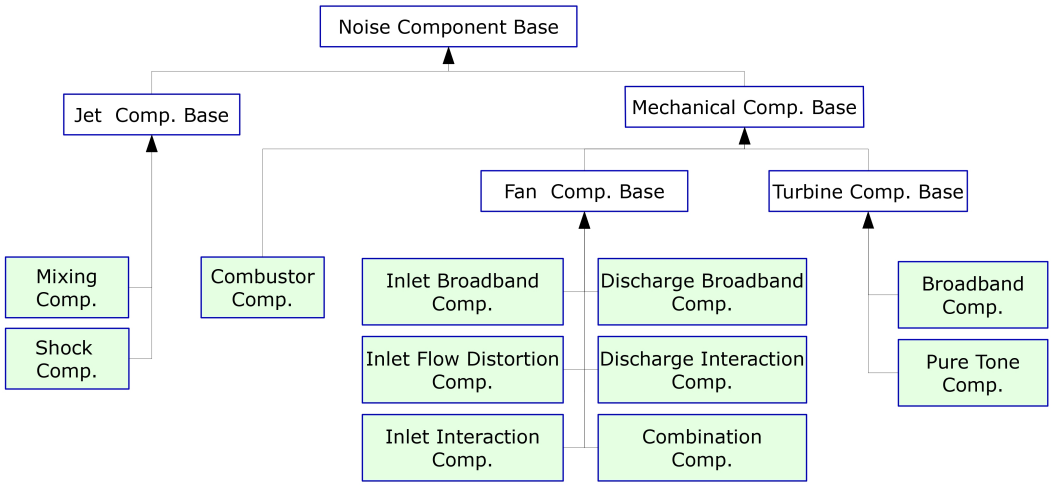


Figure 3.9: Simplified class inheritance tree in engine noise components.

- **noise model constants**, fixed constants which are characteristic of the models and should not be changed by the user.

Generally, the user inputs concern the engine and the sub-model levels, while noise model constants are used at the component level and load from a fixed set of files.

Complementary, each engine, like each microphone, has a unique ID number.

3.2.5 Landing Gear Data

Landing gear noise is also obtained from a semi-empirical model. This model, however, is much simpler than that used for the engine. Moreover, the landing gear model is divided into only two components, the wheel and the strut, whose spectral and power functions are similar. The values required by the model are:

- position relative to the aircraft (x, y, z);
- gear geometry constants (wheel diameter, wheel number, and strut length);
- noise model constants for the computation of the power and spectrum functions.
- information on the deployment status of the landing gear, which is achieved by a flag to indicate the initial condition and a time when this condition toggles.

The deployment status of the landing gear is simply given by an initial deployment flag and a time when this condition is reversed. This set of data is simple but sufficient to simulate flight paths with take off, landing or both take off and landing in the same flight profile.

Landing gears also have an extra number for identification just like the microphones and engines.

3.2.6 Noise Level Limits Data

The APM module software holds the values of all Annex 16's maximum noise limits and the corresponding trade-offs/margins. This includes the most recent Chapter 14. It also allows the user to input new custom limits and as such, it is important to understand how the noise limits are stored. The maximum noise levels for all chapters are stored as a list of items. Each item consists of a set of six values:

- Chapter number of the item;
- Flight Condition (Lateral, Approach, Flyover) associated with the represented limit;
- Mass Low Bound (MLB) in tons;
- Number of engines to which this limit is associated (set to 0 if no discrimination is made based on the number of engines);
- Two real constants: A and B used to calculate the EPNdB limit.

A list of these items can fully represent all maximum noise level in Annex 16 and Part 36. A set of all items with the same chapter number and flight condition holds all the information required to determine the maximum noise level for that chapter. The procedure to extract this number is as follows. The Maximum Take Off Weight (MTOW) of the aircraft, which we will represent as M, is compared to the highest MLB of the set. If M is above this bound, then this item holds the values required to calculate the maximum. If that is not the case then the aircraft's mass is compared to the next higher MLB until the required item is found. When found, the item's A and B parameters are used to calculate the maximum noise level according to the expression:

$$L_{lp}(M) = A + B \log M \ , \tag{3.2}$$

where L_{lp} is the EPNdB legislation limit for a single certification point. This representation is inspired in how the data is presented in Attachment A to Annex 16 [11]. If a limit is constant for a certain mass range then B is simply set to 0. After the maximum limits are calculated it is required to test for the extra conditions. These are simple to implement for generically they require only the storage of three values per stage, each of them used as a limiting value in the inequalities that are presented in Attachment A to Annex 16.

This structure is more complicated than a possible alternative implementation with a single set of tabulated constants for the latest Chapter. That alternative was deprecated because the current one has two big advantages over it. First, by being flexible enough to store all the limits from previous Chapters, this structure will allow the addition of new standards by the users in the future. Secondly, the implemented solution can handle multiple Chapters at once, which might be useful in the transition periods between different noise limit levels.

3.3 Data Input and Files

In this section, we briefly cover how all the main data structures, used in the acoustic predictions, are stored and loaded into the module. There are 3 different ways to transfer data in and out of APM:

- module to module inside MDOGUI using the `SetInput()/SetOutput()` functions from `ModuleBase`, the base class from which all modules in MDOGUI are derived from;
- loading/saving the data from/to a file;
- specifying the values through the GUI.

As mentioned before, this work focuses on APM's back end and as such the third alternative will not be addressed here. APM's documentation gives a thorough explanation on how to use its input GUI elements.

3.3.1 MDOGUI Input/Output facilities

The first method to transfer data between modules is using the functions provided in `ModuleBase`. This is uniform across the modules and therefore of trivial implementation. Nonetheless, this functionality is vital for APM to properly integrate with the rest of the framework, which is an expressed goal of this work. Here we give a reference to the indices for input and output of the most important data types in APM.

Table 3.1: Abridged index table for inputs of APM.

Microphones	Flight Path	Aircraft	Landing Gear	Engine
0	1	2	3	4
Fan Parameters	Combustor Parameters	Turbine Parameters	Jet Parameters	Options
5	6	7	8	9

Table 3.2: Abridged Index table for outputs of APM.

Total Noise	EPNdB	Fuselage Noise	Engine Noise	Gear Noise
0	1	2	3	4

3.3.2 File Formats

The second method for data transfer is through files. This is not standardized in MDOGUI and requires further explanation. All data related to APM is saved in one of two possible file formats:

CSV File

Comma-separated values (CSV) files load most of the data for the APM module and, broadly speaking, for the MDOGUI framework. This format is easy to use in MDOGUI due to the already existing methods in `FMatrix` to load and save CSV files from an `FMatrix` object.

This format was preferred over an INI Similar File (see below) when:

- there is a clear matrix nature to the data stored (*e.g.* panel positions or a model's directivity tables);
- the data stored can be easily aggregated in a matrix;
- the data to be stored is simple and does not involve many unrelated values.

INI Similar File

Some types of data would be difficult to represent and edit in tabular form. The best option would be to use the standard INI file format. INI files are amply used to store values and they would be implemented in APM if it wasn't for one hindrance. There are many open libraries to handle INI files but currently, there is no defined parser in MDOGUI. Because this should be made homogeneous across the framework, APM abstained from integrating a third party INI library. A small variation to the INI format was implemented so that the resulting files could be read using the C++ Standard Template Library `ifstream`. This change is the addition of a single blank space after the equal sign, making it easy to revert to INI when a parser is implemented. Reference [75] was used as the guideline for the INI format.

For convenience, this type of file will, henceforth, be called ISF (INI Similar File). Two examples of such files are in the appendixes A and B.

3.3.3 File location

APM allows the user to load/save files from any directory in the system by using the Windows application programming interface (API) to create *Load* and *Save* dialogues. An exception is made for the constant data files for source models and PNL calculations. The data contained in these files should in a safe directory and not altered by the user inadvertently, and as such APM provides no facilities to change them. It was decided that they should be stored in *Program Data* as is recommended for the Windows OS [76]. An experienced user which intends to change the models' constants can still do it because all files are encoded in UTF-8.

3.3.4 Flight Path Input

Since each case corresponds to a column in the path data matrices, the natural way to store the data was CSV. Flight path data can exceptionally be read from two different file formats which were named the 'Mission Format' and the 'Simple Format'. The 'Simple Format' is the default file for APM path data, while 'Mission Format' files are to ensure compatibly with inherited files from other modules in APM.

The 'Simple Format' condenses all the data APM uses to describe the flight path. It is easily obtained by vertical concatenation of all matrix data members, and as a result, each column in the file refers to a *case*. The file will have the structure presented in table 3.3.

Table 3.3: 'Simple Format' for the path data input.

	case 1	*case* 2	...	*case* N
time (s)	0.0	0.077	...	143.46
segment	1	1	...	12
position x (m)	0.0	10.211	...	20442.0
position y(m)	0.0	0.0	...	0.0
position z(m)	457.2	458.72	...	3163.2
yaw (deg)	0.0	0.0	...	0.0
pitch (deg)	12.087	12.087	...	8.48
roll (deg)	0.0	0.0	...	0.0
vel. vec. yaw (deg)	0.0	0.0	...	0.0
vel. vec. pitch (deg)	8.502	8.502	...	5.315
vel. vec. roll (deg)	0.0	0.0	...	0.0
vel. magnitude (m/s)	133.69	133.70	...	157.89
temperature (K)	285.18	285.17	...	267.59
density (Kg/m^3)	1.225	1.225	...	1.225
sound speed (m/s)	338.52	338.51	...	327.91
AOA (deg)	3.585	3.585	...	3.165
sideslip angle (deg)	0.0	0.0	...	0.0

The values shown in the previous table are taken from real files (like they will be for all similar tables that follow); their purpose is to exemplify the typical values that would be in their respective cells.

The 'Mission Format' ensures compatibility with pre-existing CSV files in the MDOGUI project which describe a mission's flight path (hence the name). In this format each *case* corresponds to a row. Files in this format will contain much more information than what is required by APM. Table 3.4 shows the specific data columns that APM extracts from these files.

Table 3.4: Abridged 'Mission Format' for the path data input.

	segment	time (s)	altitude (m)	ground distance (m)	speed (m/s)	AOC (deg)	AOA (deg)
	col. 2	col. 10	col. 15	col. 16	col. 17	col. 19	col. 23
case 1	1.0	0.0	457.2	0.0	133.69	8.502	3.585
case 2	1.0	0.077	458.72	10.211	133.7	8.502	3.85
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
case N	12.0	143.46	3163.2	20442	157.89	5.315	3.165

In 'Mission Format', *cases* at the segment boundaries may have equal emission times. This would imply the aircraft is simultaneously in two positions. For this reason, APM is also capable of filtering the values after reading the file and removing *cases* with conflicting times.

When reading one of these files, much data is not stored simply because APM does not require it.

Because the data that APM ends with is incomplete to recreate a full 'Mission Format' file, these can only be read and never be created by APM.

3.3.5 Microphone Input

Multiple microphones can be stored in a CSV file. Such a file is exemplified in table 3.5, where the boolean flags for each motion track are omitted. These would simply be a 0/1 value for each degree of freedom concatenated to the end of each line.

Table 3.5: Abridged microphone input file.

	ID number	x(m)	y(m)	z(m)	Limit Flag
mic 1	1	0.0	5000	0.0	0
mic 2	2	1000	1000	0.0	2
⋮	⋮	⋮	⋮	⋮	⋮
mic N	3	-500	0.0	200	3

The Limit Flag is what indicates the legislation measure points. They are coded as integers as follows:

Table 3.6: Legislation point integer codes.

Lateral	0
Approach	1
Flyover	2
None	3

3.3.6 Aircraft Data Input

Aircraft geometry and pressure data can be obtained in two ways, either it is loaded from a CSV file or is requested from the Aerodynamic module.

Similarly to the 'Simple Format' for the flight path, the aircraft file structure is obtained from a vertical concatenation of all `FMatrix` as shown in table 3.7. This can be done because the number of columns is constant and equal to the number of panels that discretize the aircraft.

From this file, the number of expected *cases* in the flight path can also be deduced from the number of rows referent to c_p's. Hence, mismatches in the number of *cases* between flight path and aircraft data can be detected and avoided.

Requesting the data from the Aerodynamic module (Panel Method) is also an alternative if no aircraft data file exists at run-time. Calling another module implies using the same shared utilities, that were referenced in 3.3.1, in the target module. APM can provide the Aerodynamic module with the aerodynamics *cases* in the flight path data. Aerodynamic module will, in turn, create a panel discretization of

Table 3.7: Aircraft input file.

	panel 1	panel 2	panel 3		panel N
center position - x (m)	34.613	33.937	34.355	...	36.229
center position - y (m)	0.025564	0.0056305	0.03138	...	-4.7484
center position - z (m)	-0.3305	-0.59067	-0.43125	...	1.094
area (m^2)	0.022746	0.00018749	0.011262	...	0.11988
X axis - x	-0.92524	0.93715	-0.93083	...	0.981
X axis - y	-0.090275	0.050079	-0.057995	...	0.023636
X axis - z	-0.36847	0.34531	-0.36087	...	0.19254
Y axis - x	-0.090349	0.049953	-0.07257	...	0.012265
Y axis - y	0.99576	-0.99711	0.99697	...	-0.9981
Y axis - z	-0.017091	0.0092696	0.028071	...	0.06033
Z axis - x	0.36846	0.34533	0.35816	...	0.19361
Z axis - y	0.017468	0.008562	0.052318	...	-0.05684
Z axis - z	-0.92948	-0.93844	-0.93219	...	-0.97943
c_p *case* 1	-0.016351	-0.29848	-0.0054317	...	0.14043
c_p *case* 2	-0.016351	-0.29848	-0.0054318	...	0.14043
⋮	⋮	⋮	⋮		⋮
c_p *case* K	-0.0028496	-0.31587	-9.2049e-05	...	0.14746

the aircraft from the Geometry module and proceed to calculate the pressure in all panels for all the *cases* that were set. When it finishes its computations, Aerodynamic module returns control to APM which receives all geometry and pressure data. This flow is represented in Fig 3.10.

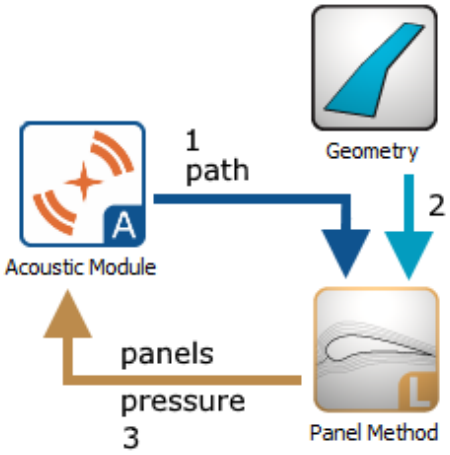


Figure 3.10: Requesting Aerodynamic module (Panel Method) for aircraft data from APM.

3.3.7 Engine Input

In section 3.2.4, we saw that the engine data can be subdivided into four groups. Of these, the frequency bands and the noise model constants should not be changed by the user. These data are automatically loaded and the user does not need to select them. As for the engine geometric constants and operational data, they may vary from operation to operation and must be provided by the user.

The engine geometric constants are stored in an ISF. An example is shown in Appendix B. This file controls not only the geometric constants for the engine but also which noise components should be used in the prediction since the models allow predictions with only a subset of the components. APM is also capable of reading engine inputs from the currently implemented Propulsion Module in MDOGUI. This speeds the input process but the Propulsion Module does not have the full set of constants the engine class in APM requires and therefore the user must always confirm and complete this data.

The operational data define the conditions at which the engine is running at a specific time. These conditions will vary with the mission and even for the same mission different operation profiles may be selected. APM thus requires the user to specify tabulated values for several engine parameters as a function of emission time. Because all this data are tabulated functions, the natural way to store this information is in CSV format. Each row in these files will correspond to a time. The table can be as small as having a single row, in which case it is considered that the operating conditions remain constant in time. In the following four tables (3.8 to 3.11) the structure of the files for each engine sub-model is shown. The values in theses tables are the ones considered by default in APM:

Table 3.8: Parameters for fan model.

time (s)	$\dot{m}^*$	N^*	ΔT^*
1.0	0.2	0.3	0.2

Table 3.9: Parameters for combustor model.

time (s)	$\dot{m}_i^*$	$p_{t,i}^*$	T_i^*	T_j^*	ΔT_{des}^*
1.0	0.2	1.0	1.0	2.0	0.5

Table 3.10: Parameters for turbine model.

time (s)	f	N^*	T_i^*	T_j^*
1.0	0.0	0.3	3.0	2.0

Table 3.11: Parameters for jet model.

time (s)	$A_{j,1}^*$	$d_{e,1}^*$	$d_{h,1}^*$	M_1	T_1^*	V_1^*	ρ_1^*	$A_{j,2}^*$	M_2	T_2^*	V_2^*	ρ_2^*
1	1.0	$2/\sqrt{\pi}$	$2/\sqrt{\pi}$	1.0	1.0	1.0	1.0	0.0	0.0	1.0	0.0	1.0

3.3.8 Landing Gear Input

Multiple landing gears can be stored in a single CSV file like it was done for the microphones.

'Initial state' codes what is the initial position of the gear, deployed (1) or retracted (0). 'Toggle time' will be the time when this deployment state inverts. In the example above all gears start deployed and

Table 3.12: Landing gear input file.

	ID number	x (m)	y (m)	z (m)	wheel number	wheel diameter (m)	strut length (m)	Initial state	Toggle time (s)
landing gear 1	1	-6.23	0.0	-3.81	2	1.0668	1.905	1	20
landing gear 2	2	-32.13	5.5	-3.81	2	1.27	3.8862	1	20
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
landing gear N	3	-32.13	-5.5	-3.81	6	1.27	3.8862	1	20

at emission time $20s$ they will retract, meaning their noise contribution will no longer be considered.

3.4 Acoustic Prediction Algorithms

In the next few sections, the algorithms used to make the noise predictions will be covered. Abstractly, to avoid overcomplicating the diagrams, we can represent the acoustic prediction scheme as two separate flows, one for source models (see Figure 3.11) and one for Formulation 1A (see Figure 3.12).

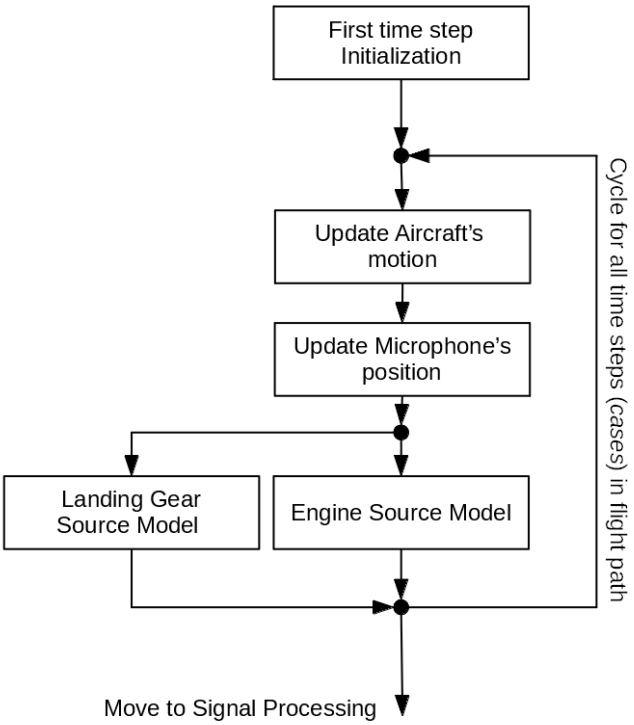


Figure 3.11: Source Models algorithm flow.

In practice, these two algorithms occur simultaneously in order to share the overhead of moving the aircraft and computing the microphone positions. From the perspective of the source model algorithm, the loop in path windows done in Formulation 1A is invisible. Details of these algorithms will be the focus of the next few sections.

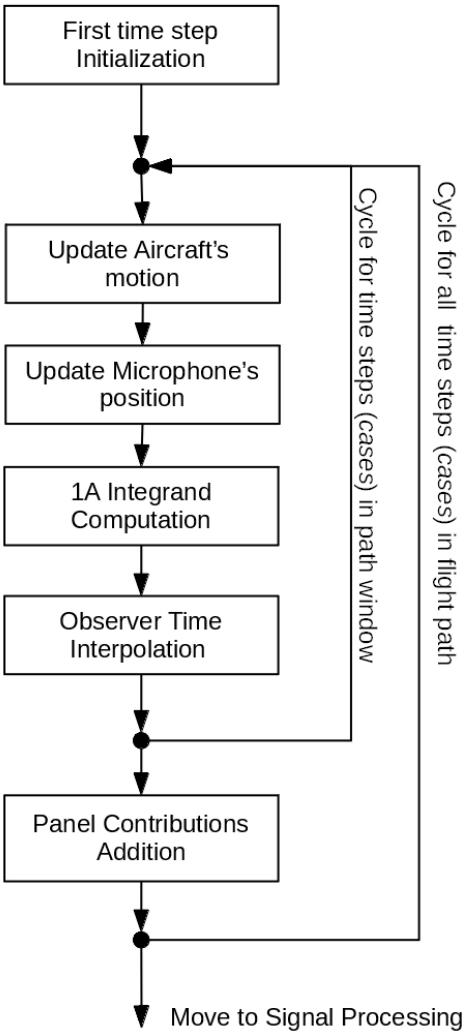


Figure 3.12: Formulation 1A algorithm flow.

3.5 Aircraft and Microphone Movement

For each case in the flight path, the aircraft must be put in its correct position and orientation in space. If there are microphones whose position is defined in the aircraft's frame of reference then these microphones must be moved as well.

To correctly place the aircraft in its position first it is rotated by the yaw, pitch and roll angles in the flight path structure. The rotation is done using a rotation matrix. Rotation about an arbitrary unit vector (a, b, c) by an angle θ is given by the rotation matrix:

$$R = \begin{bmatrix} a^2(1-\cos\theta)+\cos\theta & ab(1-\cos\theta)-c\sin\theta & ac(1-\cos\theta)+b\sin\theta \\ ab(1-\cos\theta)+c\sin\theta & b^2(1-\cos\theta)+\cos\theta & bc(1-\cos\theta)-a\sin\theta \\ ac(1-\cos\theta)-b\sin\theta & bc(1-\cos\theta)+a\sin\theta & c^2(1-\cos\theta)+\cos\theta \end{bmatrix}. \tag{3.3}$$

The derivation of this result is presented in [77].

After the rotation, the aircraft is translated to the position (x, y, z) defined in flight path structure for the current *case*.

The same procedure is then applied to all the landing gear and engines so they stay in the same relative position to the aircraft.

This same transformation is also applied to all microphones which were defined in the aircraft frame of reference. This will not be their final position, which will be addressed in the next section.

3.6 Sound Travel Time and Moving Microphones

When calculating the noise at an observer, it is required to compute when the signal arrives. For a fixed observer this is easily done by finding the root of

$$t - \tau - |\vec{x} - \vec{y}(\tau)|/c = 0 \ , \tag{3.4}$$

where τ is the emission time, t the observer time and $\vec{x}$ and $\vec{y}$ the microphone's and the source's positions, respectively. The emission time, τ, is known and is constant for a specific *case*. Thus, for a fixed microphone, all quantities are known except for reception time and (3.4) is easy to solve algebraically.

When the microphones are in the aircraft's frame this will not be the case. Both t and $\vec{x}$ are unknown at the start, thus (3.4) takes the form:

$$t - \tau - |\vec{x} - (t) - \vec{y}(\tau)|/c = 0 \ . \tag{3.5}$$

The root of this function must be found numerically. From the available approaches, the bisection method was selected. The bisection method's algorithm is simple to implement and it was based on the one presented in [78]. For this method, an interval must be selected that contains the zero. The emission time is the obvious low bound for this interval since the noise reception must happen after the emission. The initial position in which the microphone was placed (see the previous section) is used to estimate the width of the interval required. From this position and the source's emission position the interval is estimated as:

$$\Delta t = \frac{r}{c - v_{max}} \ , \tag{3.6}$$

where r is the initial distance between source and observer and v_{max} is the maximum velocity in the mission and c the sound speed. This value provides a wide enough interval for microphones that move directly away from the source, and thus it is wide enough for all other cases.

The bisection method has guaranteed convergence for the majority of functions, but its convergence is slow [78]. To minimize this, and keep the number of iterations small, the initial width of the search interval should be as tight as possible. This is achieved when r and/or v_{max} are small. The condition of v_{max} being small is expected to occur and having small values of $c - v_{max}$ should never happen because this is an invariant to the APM code (recall that the acoustic predictors used are only valid in the subsonic range). Small r is left to user discretion. This should not pose a problem to the user because tracking microphones are most useful when describing spheres or hemispheres that surround the aircraft. In this scenario distances in the order of 1 or 10^1 m are to be expected between the sources and the

observer. Microphones at distances in the order of 10^3 m or higher are expected only when defining fixed microphones in the ground. Microphones that track the source distanced several kilometers from it have little use, especially if one considers that in reality there will always be atmosphere distortions.

Due to the principle of relativity, a workaround can be used to fasten computations. The aircraft's position in the flight path is fixed and microphones' positions are set fixed in this frame. This will not be a cost-free operation because the overhead of ensuring that all input quantities are defined in the aircraft's frame of reference is delegated to the user.

3.7 Formulation 1A Solver Algorithm

The computation of Formulation 1A consists of evaluating the integrands of the thickness and loading noise equations and solving for the pressure perturbation.

The flow of the solver requires two time loops, as can be seen in Fig. 3.12. The outer loop moves through emission time, effectively advancing the motion of the aircraft in its flight path. The need for the inside time loop, with a shorter time window, is due to the RAM intensive nature of the Source Time-Dominant algorithm. This will be further discussed in section 3.9.

The step representing the evaluation of the loading and thickness noise integrands, shown in Fig. 3.12, requires another algorithm which can be broken down. First, we must specify which integrands need to be computed. For the thickness noise, applying (2.8) to (2.4), we get:

$$4\pi p'_T(x,t) = \sum_{i=0}^{N-1} \left\{ \left[\frac{\rho_0 \dot{v}_n}{r(1-M_r)^2} + \frac{\rho_0 v_n (\hat{r} \cdot \dot{\vec{M}})}{r(1-M_r)^3} \right]_{(y_i,\tau)} \Delta S_i + \left[\frac{\rho_0 c \dot{v}_n (M_r - M^2)}{r^2(1-M_r)^3} \right]_{(y_i,\tau)} \Delta S_i \right\} . \quad (3.7)$$

As for the loading noise, by applying the same procedure to (2.5), we get:

$$4\pi p'_L(x,t) = \sum_{i=0}^{N-1} \left\{ \left[\frac{\dot{p}\cos\theta}{cr(1-M_r)^3} + \frac{(\hat{r} \cdot \dot{\vec{M}}) p \cos\theta}{cr(1-M_r)^3} \right]_{(y_i,\tau)} \Delta S_i \right.$$
$$\left. + \left[\frac{p(\cos\theta - M_n)}{r^2(1-M_r)^2} + \frac{(M_r - M^2) p \cos\theta}{r^2(1-M_r)^3} \right]_{(y_i,\tau)} \Delta S_i \right\} , \quad (3.8)$$

where i is the panel index, y_i is the panel position and τ is the emission time. The parameters in these two equations can be subdivided into two groups: the ones that depend solely on the panel position, and those that also depend on the observer too. In the first group there are the following terms: ρ_0, c, ΔS_i, M, $\dot{\vec{M}}$, v_n, M_n, $\dot{v}_n$ and $\dot{p}$. While in the second we have: $\vec{r}$, $\hat{r}$, M_r and $\cos\theta$.

We focus now on how these values are computed. The values for ρ_0, c and ΔS_i are already known at this stage because they are directly stored in the structure for the flight path and the aircraft. The Mach vector M_j, is simply computed by dividing the velocity vector v_j by the sound speed c. The magnitude of

this vector gives the panel's Mach number M. The next term, v_n, is the velocity vector component in the direction normal to the panel. Because the outer normal of the panel is a unitary vector, v_n is obtained by the inner product between the speed and the outer normal:

$$v_n = \vec{v} \cdot \vec{n} \ . \tag{3.9}$$

M_n follows as

$$M_n = \frac{v_n}{c} \ . \tag{3.10}$$

To compute the derivative of v_n the product rule for derivation is used:

$$\dot{v}_n = \frac{\partial(\vec{v} \cdot \vec{n})}{\partial \tau} = \frac{\partial \vec{v}}{\partial \tau}\vec{n} + \vec{v}\frac{\partial \vec{n}}{\partial \tau} \ . \tag{3.11}$$

How the derivatives are computed is presented in the next section. The same can be said for $\dot{p}$ and $\dot{\vec{M}}$ which are obtained by derivation of the already stored values without further steps. It should be noted that the auxiliary term $\partial\vec{n}/\partial\tau$ is also independent of the observer.

Now for the computations that require not only panel data but also microphone data. Vector $\vec{r}$, is computed as explained in section 2.3. From the $\vec{r}$, its magnitude r is readily calculated. The propagation direction vector $\hat{r}$ is then given by:

$$\hat{r} = \frac{\vec{r}}{r} \ . \tag{3.12}$$

With $\hat{r}$, the Mach vector component along the propagation direction M_r is computed as :

$$M_r = \vec{M} \cdot \hat{r} \ . \tag{3.13}$$

And finally, the cosine of the angle between the outer normal and the propagation direction is provided by:

$$\cos\theta = \vec{n} \cdot \vec{r} \ . \tag{3.14}$$

Each panel-microphone pairing is unique. As mentioned previously, there will be computations which are independent of the microphone. It follows naturally that between the two cycles, the one for microphones and the one for all panels, the cycle for panels must be the outer cycle, as represented in Fig. 3.13. In this way, the number of computations is minimized. Otherwise, in each microphone loop, all panel specific computations would be repeated. This poses no problems when there is only one microphone. With more than one microphone, the total number of useless repeated operations would be

$$(number\ of\ microphones - 1) \times (number\ of\ panels) \times (number\ of\ panel\ specific\ operations) \ . \tag{3.15}$$

The number of panels will usually vary from 10^3 to 10^6. The number of operations (arithmetic, assignment, allocation, *etc*) is estimated to be in the order of magnitude of 10^2. Thus, if the loop order was

reversed, a large number of useless computations would be performed for each additional microphone.

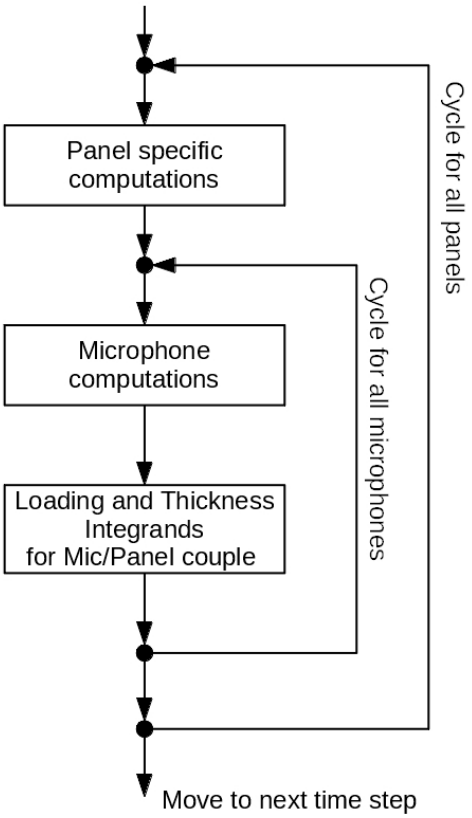


Figure 3.13: Algorithm flow for integrand computation for all Panel/Mic couple in one time step.

3.7.1 Derivation of Panel Data

Farassat's Formulation requires the derivation of panel pressure, Mach vectors, velocity vectors and panel outer normal vectors. The segmentation of the flight path (section 3.2.1) complicates the derivation process because derivatives cannot be taken across segment borders. By default, if there are no discontinuities in the flight path, the derivatives of a generic function $\Phi(t)$ are calculated by the second order central differences scheme [79]:

$$\left(\frac{\partial \Phi}{\partial t}\right)_i = \frac{\Phi_{i+1}\left(\Delta t_i\right)^2 - \Phi_{i-1}\left(\Delta t_{i+1}\right)^2 + \Phi_i\left[\left(\Delta t_{i+1}\right)^2 - \left(\Delta t_i\right)^2\right]}{\Delta t_{i+1}\Delta t_i\left(\Delta t_i + \Delta t_{i+1}\right)} . \tag{3.16}$$

In Farassat's integrands, variable t in the equation (3.16) will correspond to the emission time, τ.

If a segment discontinuity is found, and depending on where it lies, then the derivation reverts to first order forward or backward schemes. In this fashion, all cases in the flight path, including those bordering a segment transition or the flight path boundaries will have proper derivatives for the values required by Farassat.

3.8 Source Models Algorithm

Each source model procedure (engine and landing gear) in Fig. 3.11 can be represented by the algorithm shown in Fig. 3.14. Source models, unlike Formulation 1A, already compute the noise pressure data in the frequency domain, and therefore a loop for all 1/3 octave bands' frequencies is required.

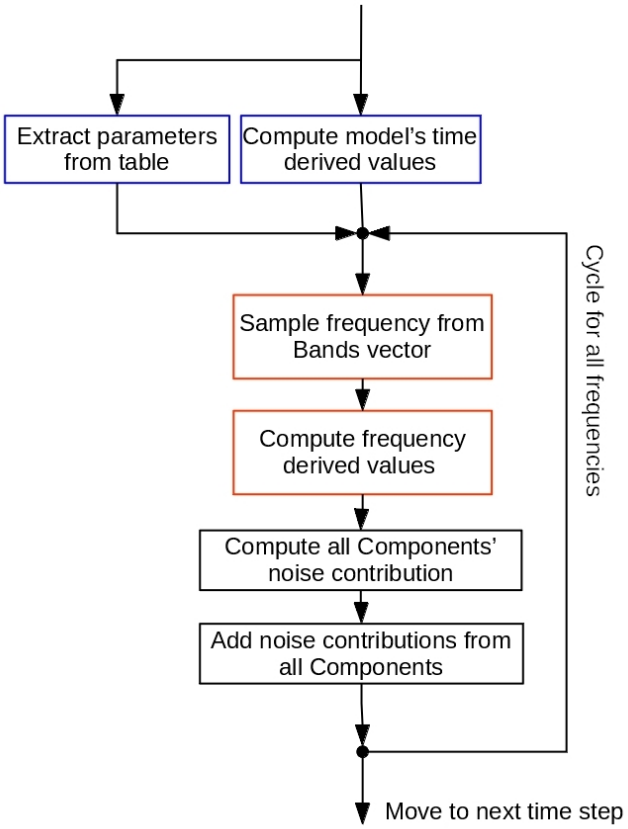


Figure 3.14: Source modelling algorithm flow in one time step.

In each time step, prior to starting the loop in frequencies, the values that solely depend on the time step are computed, and the operational parameters are retrieved from the respective table for the current emission time. At the end of this procedure, the mean squared pressures for all frequency bands will be known.

These computations could be done for every time step, however, this is neither required nor particularly useful. Formulation 1A computed the noise signal in time, and therefore it requires a small time step to provide a good resolution of the frequency spectrum. The source models already provide information across their frequency spectrum, and because the results are given for bands no improved resolution would be achieved from smaller time steps.

On the other hand, the Short-time Fourier Transform must be applied to the Formulation 1A results in order to obtain their spectrum in time. Inevitably this transformation will reduce the resolution in the time domain. Thence, the higher resolution for the source model predictions will be lost when the noise signals are summed. The intended resolution in time for the source models is left as a decision to the

user for two reasons. First, the time resolution may not be uniform in a path and vary significantly with the flight path inputted by the user. Second, the user may want to compute scenarios were Formulation 1A is ignored and only the noise from the models is computed, in which case the user should be free to set the resolution he/she desires.

This said, the application of the semi-empirical source models does not involve other complicated implementation details. Its complexity lies in the implementation of the large amount of model constants and expressions that were shown in sections 2.5 and 2.6.

3.9 Signal Interpolation

The core algorithm to compute the interpolation of pressure results was based on that used in FAST [28]. Major differences were introduced to also allow the interpolation of source model data and replace the C style arrays with MDOGUI's standard matrix class, `FMatrix`. If we consider any two panels, A and B (see Fig. 3.15), their distances to a microphone will, more often than not, be different. Therefore, the propagation times until the signal reaches the observer will differ.

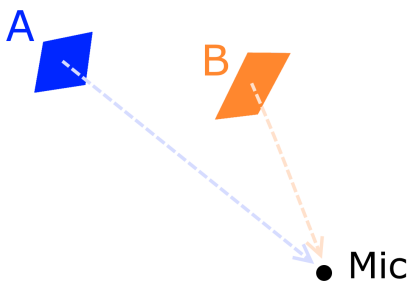


Figure 3.15: Sound propagation path from two generic panels.

Because arrival times will differ, the noise signal cannot be added directly. As it was addressed in section 2.4 an interpolation must be performed to a shared set of interpolation times, t^*. As shown in Fig. 3.16 there is no guaranteed order in the arrival times. As such the values obtained at each retarded time loop cannot be immediately aggregated. To ensure that the correct interpolated values are added together, the sum is only performed at periodic intervals when there are enough points to allow a match between correspondent interpolated observer time contributions for all panels. The length of this time segment must be balanced due to conflicting interests. A longer period allows reducing computational overhead. However, all previous panel contributions in time must be kept in memory to allow matching at the end of the inner cycle. This can be demanding on the computer's RAM resources and a reduction in the inner loop time frame alleviates them.

The results from source models must also be interpolated to a set of shared times. This interpolation must be done, not for a single pressure value as before, but for the pressure in each 1/3 octave band. The total number of noise models sources will never be very large considering the large majority of aircraft has no more than three landing gears and four engines. This is at least two or three orders of

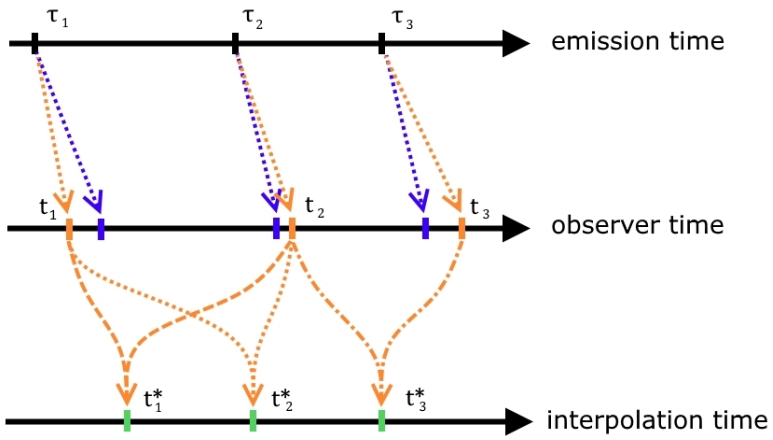


Figure 3.16: Interpolation of noise signal to common set of times.

magnitude smaller than the number of panels that discretize the geometry. The number of time steps is also smaller as it was explained in section 3.8. As a result the source models do not pose problems of excessive memory usage and, for this reason, there is no need for the inner time loop with a smaller time window, as is used for Formulation 1A.

3.10 Short-Time Fourier Transform and Fast Fourier Transform

The application of the Short-Time Fourier Transform (STFT) depends upon the selection of one or more window functions. The two windows available in APM are the rectangular and the triangular windows (see Fig. 3.17). The selection of the window is done by the user.

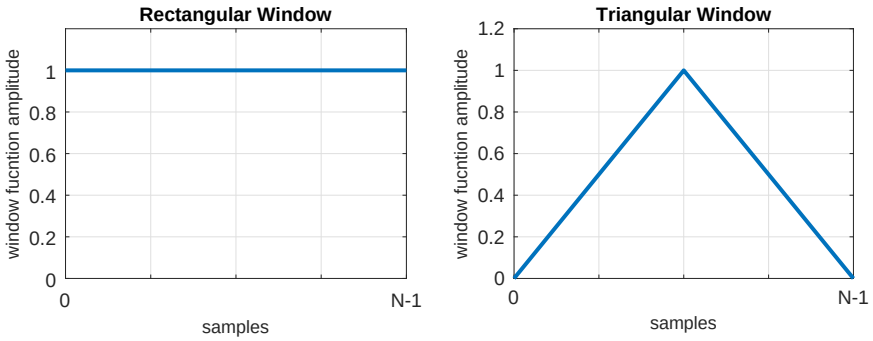


Figure 3.17: Window functions used in Short-Time Fourier Transform.

It is known that there are many more complex window functions which have better behavior when it comes to spectral leakage. The restriction to these two functions was due to time constraints. In the Fig. 3.17, N represents the number of samples which will be used in the Short-Time Fourier Transform. The higher the number of samples the better the frequency resolution but worse is the temporal resolution of the resulting signal. The number of samples N is set such that the temporal resolutions for the F1A signal and the source model signals will be the same. Thus, the two signals can be directly added. In figure 3.18 how the STFT is used might be better understood in relation to the other components.

The Fast Fourier Transform algorithm is of complex implementation, even though it can be implemented with relatively few lines of code. In this work, the FFT algorithm was modeled after the published BASIC FFT routine in [65]. In the algorithm, the time domain data is decomposed by bit reversal. Then the frequency domain data is synthesized by a three loop routine. The butterfly calculation, the basic computation in the FFT, is executed in the innermost loop which repeats for each point in the frequency spectra. The middle and the outer loops cycle through each frequency and through the $log2N$ stages, respectively. The inverse FFT routine was also modeled as described in [65]. This allows confirming the validity of code by running a signal through the FFT and the inverse FFT and comparing the input and the output of this process.

After the FFT and the inverse FFT codes were implemented, they were isolated in a separate DLL. This was done so that all other modules in MDOGUI, which until now had no Fourier Transform implementation, can have access to this commonly used transformation.

3.11 Data Processing and Limit Certification

This final section addresses how the signals provided by the two branches of the noise prediction, are processed to deliver values for noise certification and possible optimization factors. Figure 3.18 shows the flow of how both signals are joined and the noise levels computed. The connecting lines in this diagram are color-coded to show on what depends on the noise signal. For the signals to be added, the STFT must be applied to the Formulation 1A signal and the result is then separated among the 1/3 octave frequency bands. This procedure is repeated for all microphones.

The quantity which was chosen to use as optimization parameter was the EPNL. This value ultimately reflects noise annoyance to a human observer, containing both frequency and duration information. Moreover, EPNL is the most used parameter in legal and technical evaluations of aircraft noise [56]. The question that follows is to decide which observers should be considered. For this there are many options for the user to chose:

- the EPNL value for a single microphone;
- the sum or average of EPNL values for a set of microphones, no matter if they correspond to certification points or not;
- the inverse of the sum/average of the differences between the predicted EPNL for the certification points and the corresponding noise limits.

These group of goals when combined with the possible microphone configurations, some presented in Fig 3.6, allow the user to create many different study scenarios. To minimize the impact from the perspective of the observer, one can, for instance, concern solely with the noise at legislation points. Another options is placing microphones over an area, which can represent the area surrounding an airport, and obtain noise contours. From the perspective of emissions, the user may work to reduce noise

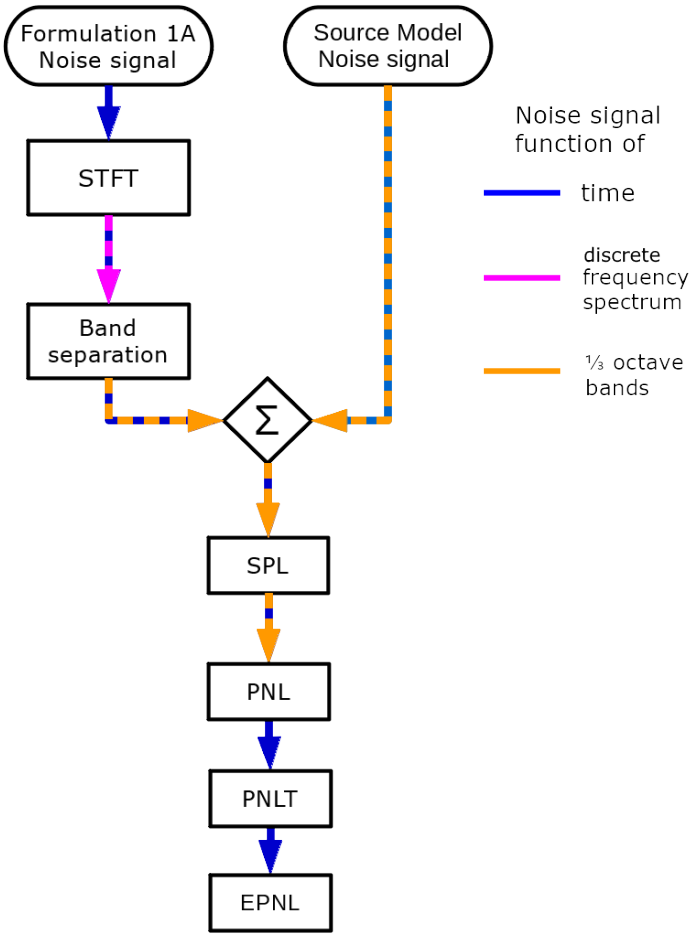


Figure 3.18: Post processing of noise signal for each microphone.

emissions overall or in specific directions. However, if the user does not desire to perform optimization, but simply confirm that the noise limits are within the legislation bounds, APM can return a simple boolean check of these limits.

Chapter 4

Results and Validation

4.1 Acoustic Code Validations

Currently, the validation of aircraft acoustic codes is difficult. Comparison between codes is usually deficient and non-conclusive [22]. There are several reasons for this [22, 57]:

- many different noise models are applied across different codes which make comparison difficult,
- the models used for noise emissions can not model all scenarios,
- there are no defined standards to which new codes should compare their results.

This may be due to the fact that even today much acoustics research is being undertaken and both the acoustic models and the software are still morphing and continuously being improved [57]. Thence, it is difficult to set a standard for comparison. Developers of acoustic prediction software will usually compare their tools to experimental data if possible, but often codes are confirmed for each model separately.

Results from APM will be shown in the next sections. To start the FFT will be confirmed, because not only is it important in APM but also because it will be shared across the MDOGUI framework independently. The tests for the acoustic predictors are then confirmed separately, starting with F1A followed by the landing gear and the engines models. Then, the results for an example scenario with predictions from all models are shown. Lastly, the computation time of APM and the Aerodynamic Module are compared and a preliminary analysis of the input impact on engine emissions is done.

4.2 Fourier Transform Validation

The MDOGUI framework had not yet a Fourier transform implemented. In the scope of developing APM, an FFT algorithm was implemented in a separated DLL to share with all modules in MDOGUI. The FFT was validated using the procedure recommended in [65], which is to run a signal through the FFT and then its inverse. The signal chosen was the same as presented in [65], a step signal with 64 samples in the time domain. The resulting signals are confirmed to be identical (see Figure 4.1).

Furthermore, the signal in the frequency domain as calculated by APM's FFT was compared to that computed by MATLAB®. The signals are identical and moreover, we can verify the expected symmetry [65] in both the real and imaginary parts (See Figures 4.2).

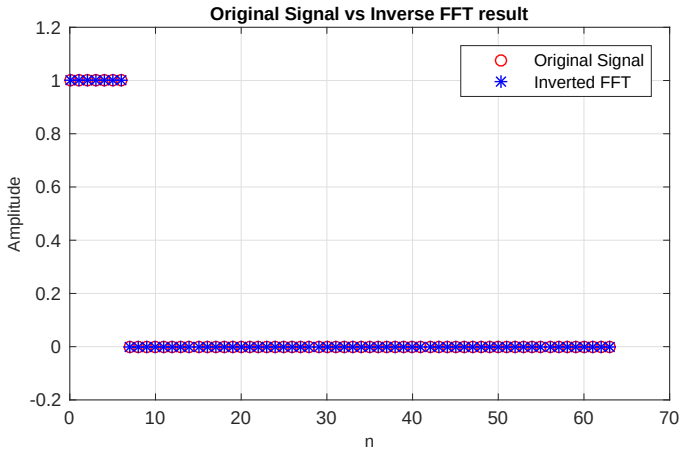
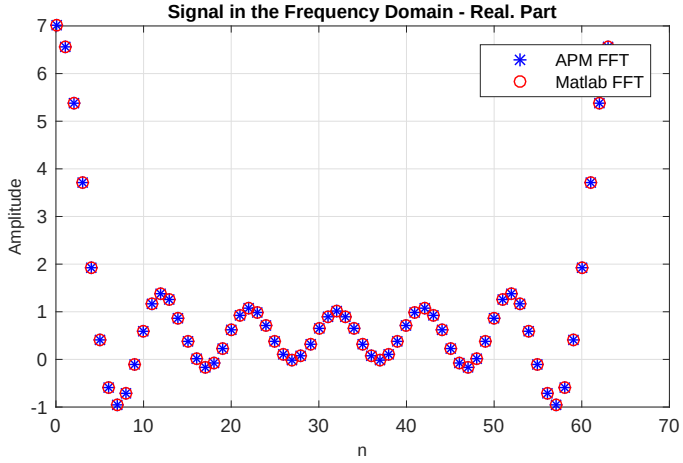
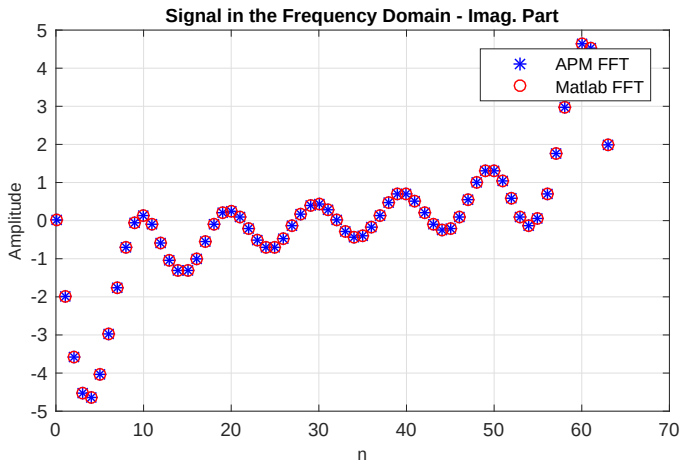


Figure 4.1: Comparison of starting step signal and final signal after the inverse FFT.



(a) Real part



(b) Imaginary part

Figure 4.2: Output from FFT of step signal in the frequency domain.

4.3 Farassat Formulation Validation

For the acoustic implementation, we start with the validation of Formulation 1A for the airframe noise. Other applications of the Ffowcs Williams-Hawkings equation or one of its derivatives to compute full airframe noise were not found (semi-empirical models are usually used). Thus, the validation of Formulation 1A could not be done *versus* another airframe noise predictor.

The strategy of validation was to confirm the F1A solver for a test case *versus* another tool that also uses Formulation 1A. This validated prediction software for comparison was from TU Delft and is based on the same principles. The test considered is a simple rectangular wing with the leading edge aligned with the z-axis and the chord is along the x-axis. The wing's dimensions and the observer position for this test are shown in table 4.1. The geometry was discretized using 33 280 panels. Velocities and pressure fluctuations were applied to the surface, matching the inputs exactly between the tools.

Table 4.1: Test rectangular wing and microphone position.

profile	chord (m)	span (m)	microphone x (m)	microphone y (m)	microphone z (m)
NACA 0018	0.2	0.8	0.2	2.0	0.0

All the input data was modified such that it is compatible with APM's inputs. The results for APM and TU Delft are shown in Fig. 4.3. In this figure, it is also presented a complementary third set of data calculated by the acoustic prediction software Farassat 1A Wind Turbines (FAWT) [80]. FAWT was developed in IST for the purposes of studying turbine noise emissions. It was validated from the same set of data provided by TU Delf and thus it is only shown here as a complement.

It is verified that all the noise computations give close noise pressure results.

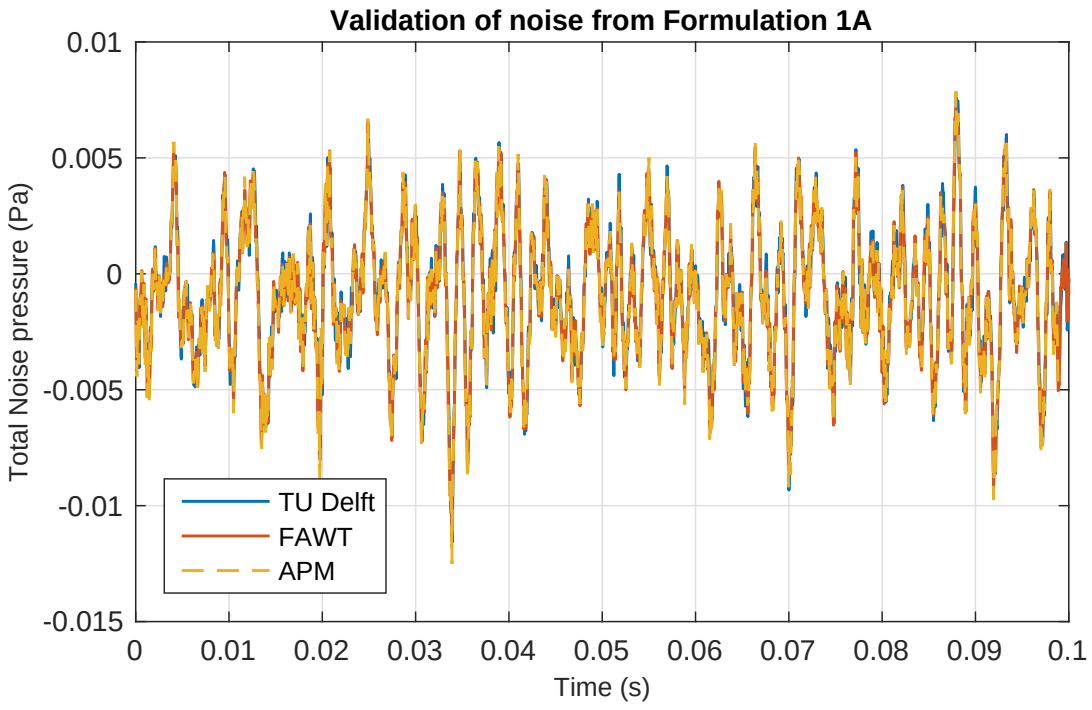


Figure 4.3: APM F1A results *versus* TU Delf and FAWT.

The root mean square error between APM's Farassat signal and TU Delft's is $5.885 \times 10^{-5} Pa$, which is two orders of magnitude smaller than the amplitude of the noise signal.

4.4 Landing Gear Model Validation

In this section, we compare the Fink model as implemented in APM to experimental data found in [54], a work in which one of the goals was to study the behaviour of the Fink model *versus* experimental data. The gear used in this case is a Boeing 777's main landing gear. Its relevant data is presented in table 4.2.

Table 4.2: Boeing 777 main landing gear data for Fink Model.

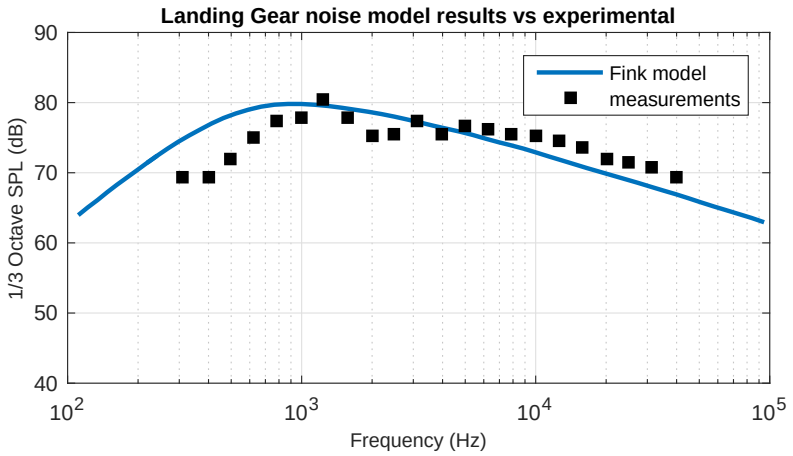
Number of wheels	Wheel diameter (m)	Strut Length (m)
6	1.27	3.89

The experimental data in [54] was obtained in a wind tunnel from a 6.3% scale model of the main landing gear. Because the landing gear noise depends on both polar (θ) and azimuthal (ϕ) angles, two conditions were tested. The angular positions in these two conditions are presented in table 4.3.

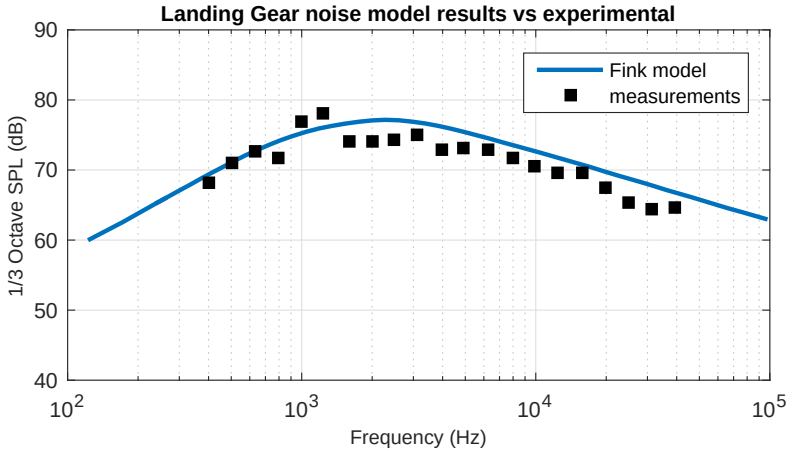
Table 4.3: Landing gear angular positions tested.

	θ (deg)	ϕ (deg)
condition 1	87.1	-1.0
condition 2	59.3	51.7

The observer is set at $24,19$ m from the landing gear for both conditions. In figure and 4.4 the results from the Fink model implemented in APM and the measurements from the scaled model are presented.



(a) $\theta = 87.1^{\text{o}}$, $\phi = -1.0^{\text{o}}$.



(b) $\theta = 59.3^{o}$, $\phi = 51.7^{o}$.

Figure 4.4: APM's Fink model results.

Additionally, the spectrum functions were computed across the frequency spectrum. These are shown in Appendix D and can be compared to the correspondent plots in [30].

4.5 Engine Model Validation

A process similar to that applied to the landing gear was also used for the engine. For the two most important noise sub-models, the fan and the jet, we checked the amplitude, spectral distribution, and directivity by comparing the models implemented in APM to the available model and experimental results. This was not possible for the combustor and the turbine. These have been, as mentioned before, the least studied models, and thus no references were found which had results and sufficient input data for APM. For these models, the component functions were verified *versus* the ones available in [30]. These component verifications were also performed for the fan and the jet. All these results are presented in Appendix E.

4.5.1 Fan model validation

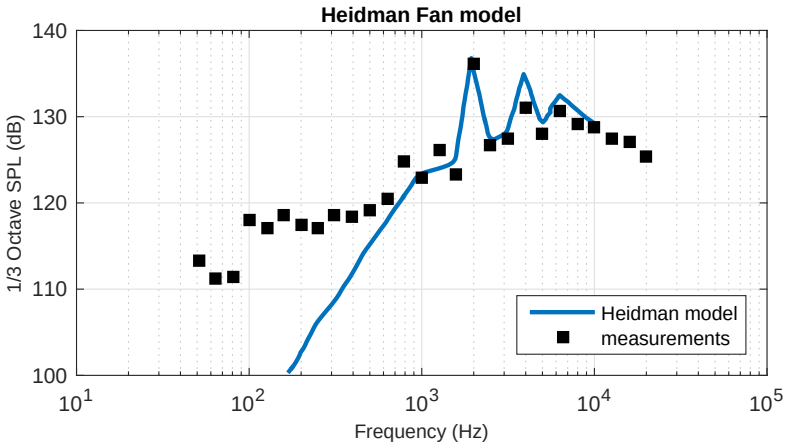
The Heidmann fan model implemented in APM was compared to the results presented in [58], where several fans are tested experimentally and their results compared to the Heidmann fan model for many polar angles. In here we shall focus on fan A, whose data is presented in table 4.4:

Table 4.4: Inputs for fan A used in [81].

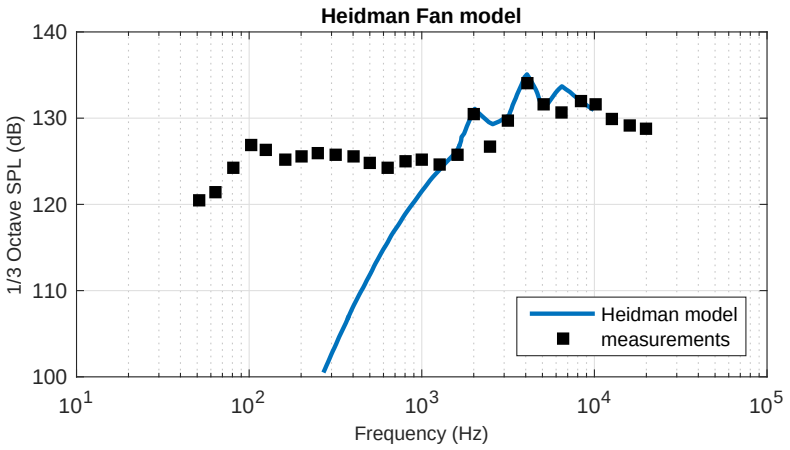
$r_s\ (m)$	$d\ (m)$	B	V	s^*	M_t	M_r	$f_b\ (Hz)$	$\dot{m}\ (Kg/s)$	Fan Pressure Ratio
1.0	1.8288	50	90	2.0	1.04	1.20	2420	430	1.5

From this set of input data all other required inputs could be computed. The pressure ratio is used to calculate the temperature rise across the fan as is indicated in [82].

The microphones were placed 1 m from the fan in two polar angles, 40º and 130º, whose results are shown in figure 4.5. These angles were chosen to allow checking directivity and both the inlet and the discharge fan components.



(a) $\theta = 40^{o}$



(b) $\theta = 130^{o}$

Figure 4.5: APM's Heidman model results.

There is a clear discrepancy in the lower frequency region ($< 10^3$ Hz) but this was confirmed to be noise from other sources that could not be isolated in the experiments [57, 81].

4.5.2 Jet model validation

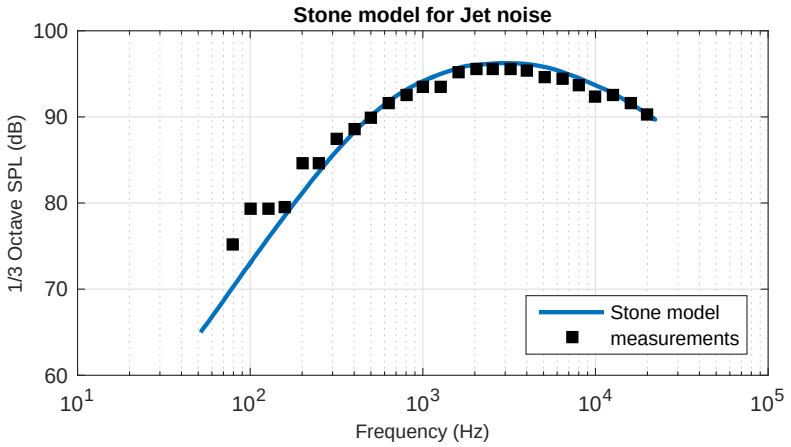
A similar procedure was done to confirm Stone's jet model. In this case, the noise from a circular cold jet was predicted. For this jet the values presented in table 4.5 were used:

Table 4.5: Cold circular jet input.

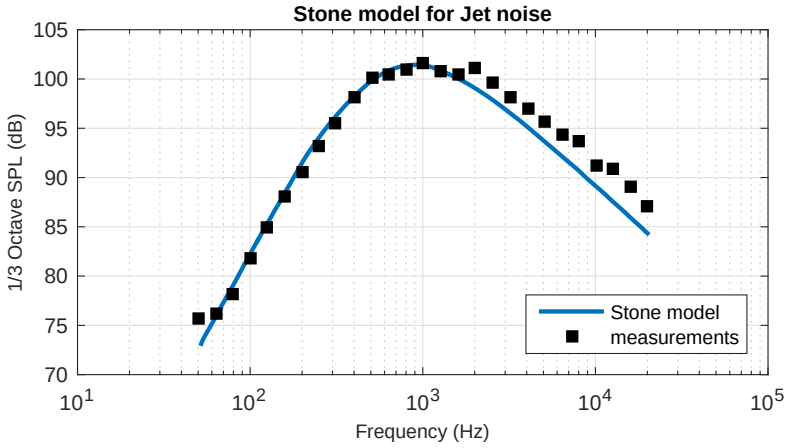
$r_s\ (m)$	$d_{j,1}\ (m)$	$V_1\ (m/s)$
4.57	0.102	302

The other fluid properties, density, and temperature were taken to be equal to the ambient value as

done in [63]. There is no plug in the considered design and as such the stream's hydraulic and equivalent diameters are considered identical. In the two conditions for the jet the microphone was placed at 4.57 m from the origin and the two polar angles considered were: 120º and 145º. The implemented model results and the experimental values from [63] are shown in Fig. 4.6.



(a) $\theta = 120^{o}$



(b) $\theta = 145^{o}$

Figure 4.6: APM's Stone Jet model results.

4.6 Example Scenario

Having confirmed the noise models, the results for a test scenario are presented. In this scenario, a take-off condition is considered and the noise is measured in the Flyover and Lateral points defined in the legislation. The noise was broken down into its several sources. The results are shown in Figures 4.7 and 4.8. Not enough engine operation data was found on landing flight to run that simulation.

The geometry used for the aircraft was the one given by the Aerodynamic module, which has been previously shown (Figure 3.5). The resulting noise emissions are identified in Figures 4.7 and 4.8 with the label '1A' since they were computed using Formulation 1A.

The source models were selected from the B-737. The B-737, probably due to its wide popularity, is commonly used in many acoustic studies. Therefore, it has the higher amount of reliable data for the acoustic models.

The B-737 landing gear data was extracted from [83]. Both main landing gear and the nose gear were used. Their data for the Fink model is represented in the table 4.6:

Table 4.6: Boeing 737 landing gear used in example scenario.

	Number of wheels	Wheel diameter (m)	Strut Length (m)
Nose Gear	2	0.6604	1.1704
Main Gear	4	1.0668	1.9304

In this scenario, it was considered that the landing gear was deployed for 45 seconds.

The engine modeled was the CFM56, which is used in many B737. Series seven of this engine is the most recent and the one for which most data was found available to the public; therefore it was the one used. The engine input file is given in Appendix B. For the purposes of this simulation, when engine operational data was lacking the values were estimated based on the data available and typical values presented in [84]. Due to lack of geometric and operational data of this engine's turbine these models were not included, and only the fan, jet and combustion chamber contributions were computed.

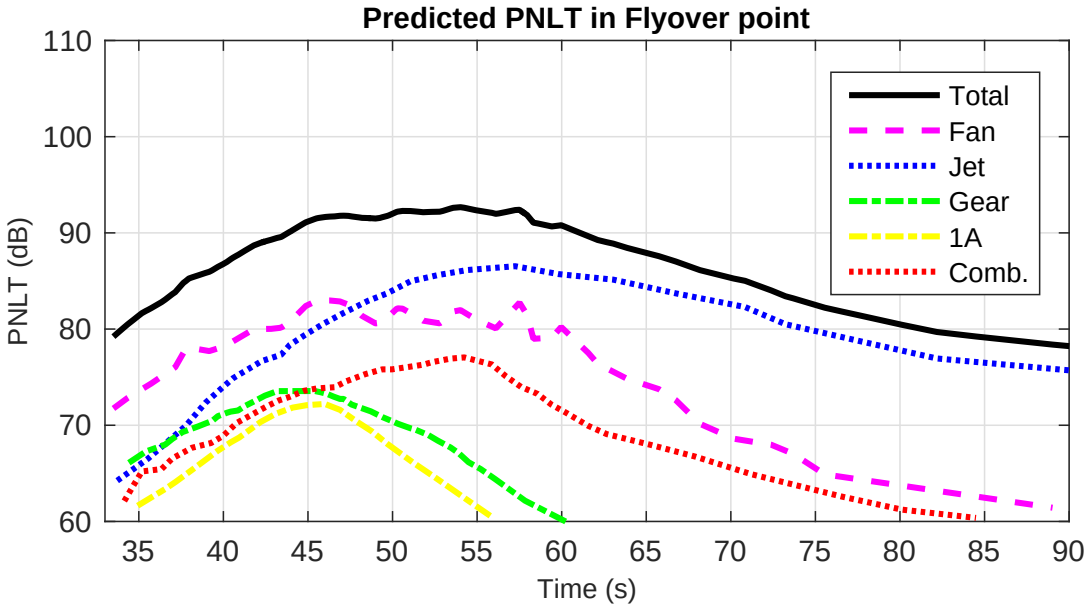


Figure 4.7: PNLT results for flyover position.

From these results, one can notice that the jet and the fan are the dominant sources. Airframe noise is significantly lower than in the examples found in the literature. This is most likely due to the fact that slats and flaps were not modeled. These provide a significant noise contribution and sometimes only their contribution is account for in the airframe [38, 30]. However, there was no support in MDOGUI for their correct modeling at the present time.

From the PNLT values, the EPNL values for each microphone were calculated (see table 4.7):

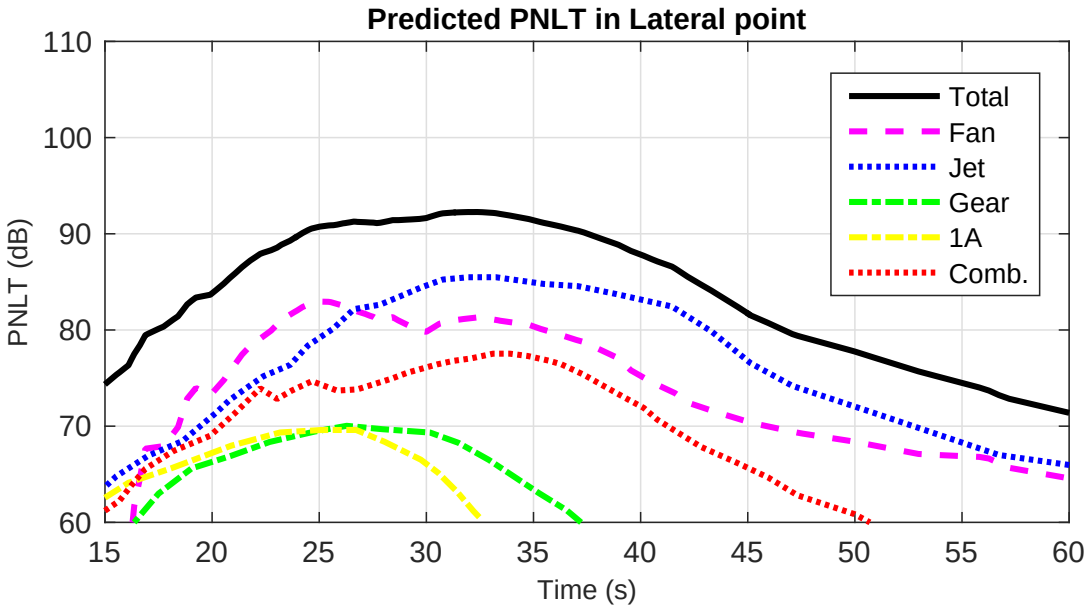


Figure 4.8: PNLT results for lateral position.

Table 4.7: EPNL results of test scenario.

	EPNL (EPNdB)
Lateral	104.18
Flyover	106.04

These results are above the allowable values for a single-aisle commercial jet's emissions. If we take the MTOW of the B-737-900 (79 016Kg), APM computes the noise limits, according to Annex 16, to be 91.87 EPNdB for the flyover condition and 97.02 EPNdB for the lateral condition. The predicted values are similar to those obtained for large aircraft. Current single-aisle commercial jets have lower noise values. If we consider a B-737, values usually range from 80 to 90 for the flyover position and 90 to 95 for the lateral position, depending on the model [85]. The values that were obtained are still within the range of common EPNL values [22, 85], but usually for aircraft larger than single-aisle.

The largest contributor to this discrepancy is suspected to be the lack of implementation of atmosphere absorption effects. Because absorption effects are proportional to the distance covered by the noise signal, this coheres with the flyover EPNL being higher than the lateral EPNL. This behavior is contrary to real measured noise profiles. In our prediction, the Flyover point, being further away than the Lateral point would congruently have a larger decrease in the received noise from atmosphere absorption.

4.7 Run Time

APM is to be integrated into MDOGUI, and as mentioned before, its turnout must be fast in order to be practically incorporated into an optimization pipeline. The acoustic computations require the aerody-

namic data which, inside this framework, will be provided by the Aerodynamic module (Panel Method) as mentioned in section 3.3.6. As such this is a good candidate to compare its runtime with APM. For the example scenario presented in the previous section, the following run times were obtained:

Table 4.8: Computation times in example scenario.

	Time (s)
APM	682,26
Aerodynamic module	37 263,88

These computations were made with the release version of MDOGUI in a 64 bits system with 32,00 GB of RAM and a Intel® Xeon® E5620 processor.

From the results is can be seen that APM's computation, even with Farasat's Formulation, is far less demanding than other critical modules. In the example scenario, APM did all required computations in 682,26 seconds which corresponds to eleven minutes, whilst the Aerodynamic module module required around ten hours for all the corresponding geometry discretization and aerodynamics computations. As such it will not be a bottleneck in the optimization pipeline.

4.8 Input impact on Engine model results

As mentioned before, the components in the engine noise models are optional and can be toggled on and off. Since the user may not have all the required inputs for all components, it would be beneficial to understand how the variation of the inputs can affect the noise results. With this information the user can decide what components to use and, if not all inputs are available, which are the most important to consider. This preliminary analysis has the intention of acessing the influence of the several parameters. Of the several functions required for the noise prediction the most important to analyse is the acoustic power function (Π^*) of each model since it is this function that determines the overall amplitude of the noise signal. These functions were exhibited in section 2.6.

The results presented in Appendix F and table 4.9. Appendix F shows the graphs relating the acoustic power of each component with its most relevant inputs. The full spectrum of possible pressure results was computed using 125 000 points equally spaced across the input domain for each component. On the other hand table 4.9 shows the maximum range of output noise in dB if the user sets one of the respective inputs and the others are allowed to vary 25% off the default values set by the models (exception made to jet's M_1 since shock noise power depends solely on M_1 and therefore the value shown is the range due to the variation of M_1 itself). For all other cells, the following reasoning applies:

- if all margins are high then the noise output is highly dependent on all the inputs and all must be computed for a correct prediction;
- if all margins are low then the noise output depends little on the inputs and a good estimation can be obtained with any of the inputs;

- if a margin is significantly lower than another, then the respective input for the noise computation is more important than the rest;

The default values set in the models represent typical operating values for most jet engines and as such flight conditions will tend to be in its neighbourhood. It should be noted, therefore, that table 4.9 serves only as reference for most common operations and does not encompass the whole domain. The values below each input are the default values considered for each model.

Table 4.9: Possible noise output range in dB when one input is determined and the two others vary by 25% of the default values.

Fan			
	$\dot{m}^*/A^*$	M_m	N^*
	(0.2)	(1.0)	(0.3)
Inl. Broad.	4.8	4.0	4.1
Disch. Broad.	3.9	3.1	4.2
Disch. R-S. I.	3.9	3.1	4.2
Inl. R-S. I.	20.3	11.9	11.0
Inl. Dist.	20.3	11.9	11.0
Combination T.	32.0	36.3	0.0

Turbine			
	T^*_t	T^*_s	U^*_t
	(3.0)	(2.0)	(0.94)
Broad.	6.2	6.7	8.2
Pure Tone	13.4	13.3	11.2
	Jet		
	ρ^*_1	V^*_1	M_1
	(1.0)	(1.0)	(1.0)
Mixing	14.2	1.6	-
Shock	-	-	37.0

Combustor		
T^*_j	T^*_i	p^*_t
(2.0)	(1.0)	(5.0)
12.3	11.6	14.4

Fan

The behaviour of the fan components can be separated into three groups.

Inlet Broadband, Discharge Broadband and Discharge Rotor-Stator Interaction Tone components depend little on their inputs. This can be confirmed in the general flatness of the corresponding graphs in Fig. F.1, F.2 and F.3. Of the three inputs, $\dot{m}^*/A^*$ is the most important since significant drops in noise occur when $\dot{m}^*/A^*$ approaches 0. If the user verifies that the operation is not in this outlining case, then the noise can be predicted with no further input. During take-off this should be no concern since the hight thrust requirements will imply a high $\dot{m}^*/A^*$. The user would only need to care about this aspect during landings.

The same cannot be said for the Inlet Rotor-Stator Interaction Tone and Inlet Distortion Tone components. These depends highly on both M_m and N^* alike, as can be confirmed in Fig. F.4 and F.5. Such would be expected since they originate in the rotor blade rotations and its interaction with the flow. No good approximation can be done without the correct computation of these values.

Lastly, Combination tone noise varies little with M_m but is highly correlated with N^*, and as can be seen in F.6, the most common operating condition sits in a plateau where there is no significant variation with $\dot{m}^*/A^*$. As such, if the computation of N^* verifies that it lies in the region of common operation, this contribution can be closely approximated with no further input.

In conclusion, when there is little data available, both broadband components can be easily estimated without much error. The other four components are responsible for tone information. Of these, the

Distortion and Inlet Rotor-Stator are the most difficult to estimate without the exact knowledge of the rotor rotation speed.

Jet

As mentioned in Chapter 2, the Jet is, along with the Fan, the most important noise source in the engine. As it can be seen from Fig. F.7 and the results in table 4.9, the velocity of the inner jet (V_1^*, M_1) is the most important input to be considered. A preliminary assessment of these values should often be at hand since they are commonly obtained from a balances to evaluate its thrust. Without them no reliable prediction can be made, since the noise in the output can vary in the order of 14 dB for the Mixing component and 37 dB in the Shock component.

Combustor and Turbine

The two remaining components have the least impact in the total engine noise. In both, the dependence on temperature relates most with their difference than with any of the two individually as can be seen in Fig. F.8, F.9 and F.10. The valey that can be seen in graph (a) of these figures corresponds to line where the temperatures are equal. The Turbine and the Combustor differ in the sense that, for the combustor the influence of the inlet pressure is of the same relevance as the temperature gradient, while for the turbine the blade rotation speed is more important than the temperature difference. This is more pronounced in the Turbine Pure Tone component. In line with the results for the Fan, the Turbine Broadband component also presents smaller margins than the Pure Tone component.

Closing this topic, we can conclude that a first rough approximation can be obtained from only the Fan and the Jet components. The Turbine and the Combustor, requiring a detailed appreciation of the inputs, may be disregarded in a first trial as is done in other tools. For most flight conditions the Fan contributions from the Discharge Rotor-Stator Interaction and both Broadband components can be easily estimated. The rest of the Fan's tonal components will require a more detailed knowledge of the rotation velocities in the rotor. For the estimation of the Jet's noise it is paramount to know the exhaust jet velocity. If this is computed, then both Jet components can be used to give an approximation of its emission.

Chapter 5

Conclusions and Future Work

5.1 Achievements

In this work, APM, an entire new module to the MDOGUI framework, was created. This new module will serve as the foundations for the acoustic predictions in the framework, providing from the very onset the following acoustic related features:

1. airframe noise prediction with Farassat Formulation 1A;
2. landing gear noise source modelling;
3. engine noise source modelling;
4. arbitrary flight path trajectories;
5. moving observers;
6. SPL, PNL, PNLT, EPNL calculation;
7. FFT and STFT;
8. certification limits benchmarking;

All the noise predictors were validated in the scope of this work. Farassat's Formulation 1A will allow studying noise emissions for the novel airframe configurations designed in MDOGUI. The use of Formulation 1A is an unconventional approach since most acoustic codes use semi-empirical relations. Therefore, APM is uniquely prepared to conduct studies in airframe noise. The additional features for microphone placement and motion, flight path handling and certification limits calculation will allow the use of the module in a variety of study cases.

Special care was placed in code design to allow reusability and code sharing, as recommended in sources such as [71]. Examples of this are the implementation of the FFT algorithm as a separate DLL making it accessible to the entire framework, and the node class hierarchy used for the engine and landing gear noise source predictors. Such an architecture will ease the implementation of new alternative source predictors in the future.

Moreover, APM is accompanied by an extensive documentation. This will not only facilitate its use but assist in future development of features and expansions to the base module, since the source code is thoroughly documented. Such a detail was paramount in a work with such a strong collaborative nature. As discussed before, there are several features of relevance (like air absorption effects) which can be added to the module and this will be the topic of the next section.

5.2 Future Work

APM provides the foundations for acoustic predictions in MDOGUI. However, in the scope of MDOGUI, this work is open-ended and many more features could be added for further improvements and new functionalities.

There are several ways in which this work can be expanded and improved upon, such as:

- atmospheric and ground effects;
- creation of an acoustic emissions optimizer;
- aircraft shielding effects in wings, tail, and fuselage;
- Formulation 1A for propellers integration;
- expansion of available window functions for STFT;
- new positioning algorithm for moving microphones;
- implementation of supersonic noise predictor.

Next, these points are quickly discussed as a guide for future development.

Currently, APM does not account for atmospheric nor ground effects. These can significantly alter the noise that reaches the observer. The most commonly modeled are air and ground absorption, which attenuate the noise; refraction, which changes the propagation direction due to wind or temperature gradients; and ground reflection [57, 30]. Atmosphere absorption especially should be modeled because it has a big impact if the noise travels a large distance to the observer. Support of these features varies with acoustic codes but there are ample studies and implementations to support this work.

MDOGUI has an inherent optimization focus. With the addition of APM, noise emissions optimization can now be performed. Several optimization scenarios could be studied such as flight path trajectories for minimal noise impact and airframe geometries for noise minimization among others.

One of the initial ambitions for APM was to confirm noise reduction in a joined-wing aircraft. In joined-wing and blended-wing-body aircraft, the effects of reflection and shielding cannot be neglected. Usually, the wings and fuselage in these aircraft shield the engine noise. Thus ray tracing methods must be implemented to properly study the noise profiles. If this is not done then a simple gross estimation of the noise can be obtained and little correlations can be made between the aircraft's geometry and its noise characteristics. APM already supports arbitrary fuselage and wing geometries and as such only a shielding method is missing to study these designs. This should also increase the predictions' precision

for conventional aircraft, but its impact should be smaller since in these the engines are usually mounted below the wing. The best alternative is to implement a full three-dimensional ray cast method as it was done for SAI [86].

In this work, several models were introduced to describe the engine components' noise. However, it was not possible to include them all. One such component is the propeller. In APM a workaround was inserted to allow to sum the propeller noise which is calculated outside the module. This was done to allow users to use another software for propeller noise and insert it in APM to be accounted for along the other sources. An example of such a software is FAST. FAST applies Formulation 1A to predict the noise for rotating blades. This is aligned with the modern codes being used to predict noise emissions [22, 33]. Building upon FAST, a robust interface could be developed to allow easy data transfer between APM and FAST. Alternately a source model could be implemented as a low computational cost solution to predict noise. The addition of the propeller solver would certainly find applications in MDOGUI because often small experimental aircraft have propellers.

On the STFT code, better windows could be implemented. In APM only triangular and rectangular windows were implemented. This should provide a small improvement in the spectral behavior of the STFT when applied to the airframe noise.

The bisection method is used to compute the microphone position at reception time. This was selected because it is easy to implement and guarantees convergence. It has the disadvantage of being slow to converge. It has been shown that the time interval where the algorithm has to search for the root will tend to be a small one. However, if there are many moving observers this algorithm will have poor performance. The bisection method can be used as a starting point to estimate the root and then be supplemented by a faster converging method [78].

Finally, if MDOGUI is ever used to study supersonic aircraft considerable additions would need to be made. In this case, the current implementation for airframe noise, Formulation 1A, would no longer be valid. As is done in other programs [33] another formulation could be implemented in parallel to handle supersonic cases. This implies improving the moving observer support which was built with the assumption of subsonic flight velocities.

Bibliography

[1] Federal Aviation Administration. FAA Aerospace Forecast Fiscal Years 2015-2035. URL https://www.faa.gov/data_research/aviation/aerospace_forecasts/media/2015_National_Forecast_Report.pdf.

[2] B. Pearse. The state of air transport markets and the airline industry after the great recession. *Journal of Air Transport Management*, 21:3–9, 2012.

[3] Airbus S.A.S. Global Market Forecast - Mapping Demand 2016-2035. Blagnac Cedex, France, September 2016. ISBN ISBN: 978-2-9554382-1-6.

[4] Boeing. Current market outlook 2015-2034, 2015. Boeing Commercial Airplanes Market Analysis. Seattle, WA 98124-2207.

[5] EuroControl. Challenges of growth 2013 summary report. Technical report, European Organisation for the Safety of Air Navigation, Brussels, Belgium, June 2013. URL www.eurocontrol.int/articles/challenges-growth.

[6] United Nations. World urbanization prospects - the 2014 revision. Technical Report (ST/ESA/SER.A/366), Department of Economic and Social Affairs, Population Division, 2015.

[7] WHO. Burden of Disease from Environmental Noise - Quantification of Healthy Life Years Lost in Europe. Copenhagen, Denmark, 2011. ISBN 978-92-890-0229-5.

[8] C. Clark and S. A. Stansfeld. The effect of nocturnal aircraft noise on health: a review of recent evidence. Technical report, London Borough of Hounslow, London, UK, September 2011.

[9] R. N. Lawton and D. Fujiwara. Living with aircraft noise: Airport proximity, aviation noise and subjective wellbeing in england. *Transportation Research Part D: Transport and Environment*, 42: 104–118, January 2016.

[10] C. Clark. Aircraft noise effects on health. Technical report, Queen Mary University of London, May 2015. Prepared for the Airports Commission.

[11] ICAO. *Annex 16 Environmental Protection, Volume I Aircraft Noise, Amendment 11*. 9^{th} edition, July 2014.

[12] Transport Canada. Managing noise from aircraft, 2017. URL https://tc.gc.ca/eng/civilaviation/standards/aerodromeairnav-standards-noise-menu-923.htm. Online; accessed 13-May-2017.

[13] EASA. EASA Approved Noise Levels, 2017. URL https://www.easa.europa.eu/document-library/noise-type-certificates-approved-noise-levels#group-easa-related-content. Online; accessed 13-May-2017.

[14] FAA. Noise Standards: Aircraft Type and Airworthiness Certification, 14 C.F.R. Part 36 , March 2016.

[15] FAA. Stage 5 airplane noise standards. *Federal Register, Proposed Rules*, 81(part 36-number 9), January 2016.

[16] E. Lecomte and E. Kors. Progress on aviation noise research in europe. *Skyline, Special Edition Noise reduction*, page 10, November 2014.

[17] Clean sky at a glance. Avenue de La Toison d'Or 56-60, Brussels, Belgium, March 2016.

[18] G. Maoui. *Innovation Takes Off, CleanSky European research for aeronautics*. Cherche Midi, Rue du Cherche Midi, Paris, France, 2016. ISBN 2-7491-5232-5.

[19] V. Guenon. The past 15 years and the next 15 years of european aviation research, an industry perspective, 20 October 2016. Key-Note at the EASN International Conference. Porto, Portugal.

[20] European Union. Flightpath 2050, Europe's Vision for Aviation. Report of the High Level Group on Aviation Research, 2011. ISBN 92-79-19724-6. doi: 10.2777/50266.

[21] European Commission. Proposal for a Council Regulation on the Clean Sky 2 Joint Undertaking, July 2013. COM(2013) 505 final, 2013/0244 (NLE), Brussels.

[22] A. Filippone. Aircraft noise prediction. *Progress in Aerospace Sciences*, 68:27–63, July 2014.

[23] F. Afonso, J. Vale, F. Lau, and A. Suleman. Performance based multidisciplinary design optimization of morphing aircraft. *Aerospace Science and Technology*, 67:1 – 12, 2017. doi: https://doi.org/10.1016/j.ast.2017.03.029. URL http://www.sciencedirect.com/science/article/pii/S1270963817304613.

[24] J. Vale, F. Afonso, F. Lau, and A. Suleman. Performance based mdo of a joined-wing regional transport aircraft. In *56th AIAA/ASCE/ AHS/ASC Structures, Structural Dynamic, and Material Conference, AIAA Science and Technology Forum and Exposition, (SciTech 2015)*, number 0696 in 2015, Kissimmee, Florida, USA, January 2015. doi: 10.2514/6.2015-0696.

[25] ESDU. Aircraft noise prediction, November 2011. URL https://www.esdu.com/cgi-bin/ps.pl?sess=unlicensed_1170504180957kgw&t=doc&p=esdu_09009b.

[26] M. D. Dahl. Assessment of NASA's Aircraft Noise Prediction Capability. Technical Report NASA/TP-2012-215653, NASA, Langley Research Center, Hampton, VA United States, July 2012. Doc ID:20120012957.

[27] B. D. Yi Liu, Phuriwat Anusonti-Inthra. Development and Validation of a Multidisciplinary Tool for Accurate and Efficient Rotorcraft Noise Prediction (MUTE). Technical Report 2011-217057, NASA National Institute of Aerospace, Hampton, VA United States, February 2011. Doc ID:20110007358.

[28] A. E. A. Vieira. Helicopter rotor noise, development of an acoustic software tool. Thesis to obtain master of science degree, Instituto Superior Técnico, Lisbon, November 2013.

[29] MSC. Simulating reality. MSC Software Magazine, Volume IV. Summer 2014 Issue. Based on an interview with Jean-Yves Suratteau, Airbus.

[30] W. E. Zorumski. *Aircraft Noise Prediction Program, Theoretical Manual*. NASA Langley Research Center, 1982.

[31] L. C. Nguygen and J. J. Kelly. A users guide for the nasa anopp propeller analysis system. Technical Report NASA-CR-4768, NAS 1.26:4768, NASA Langley Research Center, Hampton, VA United States, February . Doc ID: 19970012644.

[32] L. V. Lopes and C. L. Burley. Design of the Next Generation Aircraft Noise Prediction Program:ANOPP2. Technical Report NF1676L-11643, NASA, Langley Research Center, Hampton, VA United States, June 2011. Doc ID:20110012482.

[33] M. H. Dunn and G. M. Tarkenton. *Computational Methods in the Prediction of Advanced Subsonic and Supersonic Propeller Induced Noise - ASSPIN User's Manual*. Hampton, Virginia, April 1992. Contractor Report 4434, Prepared for Langley Research Center under Contract NAS1-19000.

[34] F. Farassat and J. H. Casper. Broadband Noise Prediction When Turbulence Simulation Is Available - Derivation of Formulation 2B and Its Statistical Analysis. *Journal of Sound and Vibration*, 331(10): 2203–2208, May 2012.

[35] D. T. Z. S. Adam Diedrich, James Hileman. Multidisciplinary design and optimization of the silent aircraft. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, number AIAA 2006-1323, Reno, NV., January 2006.

[36] S. G. Lothar Bertsch, Werner Dobrzynski. Tool Development for Low-Noise Aircraft Design. *Journal of Aircraft*, 47(2), March 2010.

[37] L. Bertsch and U. Isermann. Noise prediction toolbox used by the DLR aircraft noise working group. In *INTER-NOISE, International Congress and Exposition on Noise Control Engineering, At*, Innsbruck, Austria, September 2013.

[38] A. Filippone. *Flight Performance Software FLIGHT*. School of Mechanical, Aerospace and Civil Engineering, The University of Manchester, United Kingdom, January 2016. AF-AERO-UNIMAN-2014-10.

[39] A. Filippone. Theoretical Framework for the Simulation of Transport Aircraft Flight. *Journal of Aircraft*, 47(5):1669–1696, September 2010. doi: 10.2514/1.C000252.

[40] A. Filippone. Options for aircraft noise production on arrival and landing. *Aerospace Science and Technology*, 60:31–38, January 2017.

[41] F. Farassat and J. H. Casper. Towards an airframe noise prediction methodology: Survey of current approaches. In *Presented at AIAA 44th Aerospace Sciences Meeting and Exhibit, Reno, Nevada*, Hampton, Virginia, January 2006. NASA Langley Research Center.

[42] M. J. Lighthill. On Sound Generated Aerodynamically. I. General Theory. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 211(1107): 564–587, March 1952.

[43] 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. Series A, Mathematical and Physical Sciences*, 264(1151):321–342, May 1969.

[44] F.Farassat. Theory of noise generation from moving bodies with an application to helicopter rotors. Technical report, NASA Langley Research Center, 1975.

[45] F. Farassat. Derivation of Formulations 1 and 1A of Farassat. Technical Report NASA/TM-2007-214853, NASA, Langley Research Center, Hampton, VA United States, March 2007. Doc ID:20070010579.

[46] M. S. Howe. *Theory of Vortex Sound*. Cambridge University Press, 2003. ISBN 0-521-01223-2.

[47] F. Farassat. Introduction to generalized functions with applications in aerodynamics and aeroacoustics. Technical Report 3428, NASA, Langley Research Center, Hampton, VA United States, May 1994. Doc ID:19940029887.

[48] F. Farassat and M. K. Myers. Multidimensional generalized functions in aeroacoustics and fluid mechanics. part 1; basic concepts and operations. Pre-Print, January 2011. Doc ID:20110011404.

[49] F. Farassat, K. S. Brentner, and M. H. Dunn. Working with the wave equation in aeroacoustics: The pleasures of generalized functions. In *13th AIAA/CEAS Aeroacoustics Conference; 21-23 May 2007*, number AIAA 2007-3562, Hampton, VA, United States, May 2007. NASA Langley Research Center.

[50] F. Farassat and J. H. Casper. Broadband Noise Prediction When Turbulence Simulation Is Available - Derivation of Formulation 2B and Its Statistical Analysis. Technical report, NASA, Langley Research Center, Hampton, VA United States, January 2012. Doc ID:20120008645.

[51] M. D. F. Farassat, S. L. Padula. Advanced turboprop noise prediction based on recent theoretical results. *Journal of Sound and Vibration*, 119(1):53–79, December 1987. doi: 10.1016/0022-460X(87)90189-1.

[52] K. S. Bretner. Numerical algorithms for acoustic integrals with examples for rotor noise prediction. *AIAA Journal*, 35(4):625–630, April 1997.

[53] M. R. Fink. Noise component method for airframe noise. *Journal of Aircraft*, 16(10):659–665, 1979.

[54] C. L. Burley, T. F. Brooks, W. H. Jr., and J. Rawls. ANOPP Landing Gear Noise Prediction Comparisons to Model-scale Data. In *13th AIAA/CEAS Aeroacoustics Conference*, number AIAA-2007-3459, Hampton, VA, United States, 21-23 May 2007. NASA Langley Research Center. Doc ID: 20070021696.

[55] Brederode, V. de. *Fundamentos de Aerodinâmica Incompressível*. Lisboa, Portugal, 1^{st} edition, 1997. ISBN 972-974-02-0-8.

[56] M. J. T. Smith. *Aircraft Noise*. Cambridge Aerospace Series. Cambridge University Press, 1^{st} edition, 1989. ISBN 978-0-521-33186-2.

[57] K. Attenborough,O. Zaporozhets, V. Tokarev. *Aircraft noise assessment, prediction and control*. Spon Press, 711 Third Avenue, New York, NY, 1^{st} edition, 2011. ISBN 978-0-415-24066-6.

[58] M. F. Heidmann. Interim prediction method for fan and compressor source noise. Memorandum NASA TM X-71763, Lewis Research Center, Cleveland, OH, United States, June 1979.

[59] R. K. Matta, G. T. Sandusky, V. L. Doyle. Ge core engine noise investigation - low emission engines, February 1977. FAA-RD-77-4.

[60] Dr. David R. Williams. Earth fact sheet, February 2017. URL `https://nssdc.gsfc.nasa.gov/planetary/factsheet/earthfact.html`. Last Updated : 23 December 2016.

[61] M. A. B. Yunus A. Çengel. *Thermodynamics: and Engineering Approach*, chapter 15. McGraw-Hill, 7^{th} edition, 2011. ISBN 978-007-131111-3.

[62] S. B. Kazin , R. K. Matta , K. R. Bilwakesh , J. J. Emmerling , D. Latham. Supplement 1 - prediction methods. In *Core Engine Noise Control Program*, volume III. March 1976.

[63] J. R. Stone. Interim prediction method for jet noise. Memorandum NASA-TM-X-71618, Lewis Research Center, Cleveland, OH, United States, January 1974.

[64] J. Stone and F. Montegani. An improved prediction method for the noise generated in flight by circular jets. Memorandum NASA-TM-81470, Lewis Research Center, Cleveland, OH, United States, January 1980.

[65] S. W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publisher, 1st edition edition, 1997. ISBN ISBN-0-9660176-3-3.

[66] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, April 1965.

[67] P. J. Wolfe. Short-time fourier transform, es 157/257: Speech and audio processing, Spring 2009. URL http://isites.harvard.edu/fs/docs/icb.topic541812.files/lec12_spr09.pdf. Online; accessed 11-Jan-2017.

[68] G. F. Franklin, J. D. Powell, A. Emami-Naeini. *Feedback Control of Dynamics Systems*, chapter 8, pages 596–597. Pearson, 6^{th} edition, 2010. ISBN 0-13-50050-9.

[69] J. Hammond and K. Shin. *Fundamentals of Signal Processing for Sound and Vibration Engineers*. Wiley, February 2008. ISBN 978-0-470-51188-6.

[70] J. Boettcher. Aircraft noise re-certification - assessment criteria, November 2006. URL http://www.icao.int/Meetings/EnvironmentalWorkshops/Documents/Forms/AllItems.aspx. ICAO Noise Certification Workshop, Asia/Pacific Region.

[71] B. Stroustrup. *The C++ programming language*. Addison-Wesley Professional, 4^{th} edition, May 2013. ISBN 978-0-321-56384-2.

[72] Microsoft Corporation (2015). Visual Studio Community 2013 (Version 12.0.40629.00 Update 5 - v.120) [Computer Software]. Retrieved from. URL https://www.visualstudio.com/pt-br/vs/older-downloads/.

[73] Microsoft Corporation. Visual C++ in Visual Studio 2013, 2017. URL https://msdn.microsoft.com/en-us/library/60k1461a(v=vs.120).aspx. Online; accessed 03-Jan-2017.

[74] Microsoft Corporation. Support For C++11/14/17 Features (Modern C++), 2014. URL https://msdn.microsoft.com/en-us/library/hh567368(v=vs.140).aspx. Online; accessed 03-Jan-2017.

[75] Apache Software Foundation. INIConfiguration, 2017. URL http://commons.apache.org/proper/commons-configuration/apidocs/org/apache/commons/configuration2/INIConfiguration.html. Online; accessed 07-Dec-2016.

[76] Microsoft Corporation. CSIDL, 2017. URL https://msdn.microsoft.com/en-us/library/windows/desktop/bb762494.aspx. Online; accessed 22-Oct-2016.

[77] W. M. Newman and R. F. Sproull. *Principles of Interactive Computer Graphics*. Computer Science Series. McGraw-Hill, New York, 2^{nd} edition, 1979. ISBN 9780070463387.

[78] H. Pina. *Métodos Numéricos*. Escolar Editora, 2010. ISBN 978-972-592-284-2.

[79] J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. 3^{rd} edition, 2002. ISBN 3-540-42074-3. doi: 10.1007/978-3-642-56026-2.

[80] J. H. dos Santos Rodrigues Nogueira Santos Souto. Development of an aeroacoustic prediction tool for wind turbine noise. Thesis to obtain master of science degree, Instituto Superior Técnico, Lisbon, 2017. To appear.

[81] D. S. W. Joe W. Hough. Aircraft Noise Prediction Program (ANOPP) Fan Noise Prediction for Small Engines. Technical Report NASA-CR-198300, NAS 1.26-198300, NASA Langley Research Center, Marietta, GA United States, April 1996. Doc ID: 19960042711.

[82] C. F. Marcus F. Heidmann. Noise comparisons from full-scale fan tests at nasa lewis research center. Memorandum NASA TM X-68289, Lewis Research Center, Cleveland, OH, United States, October 1973.

[83] R. A. Golub and Y.-P. Guo. Empirical prediction of aircraft landing gear noise. Technical Report NASA/CR-2005-213780, Hampton, VA, United States, July . Doc ID: 20050209966.

[84] H. Saravanamuttoo, G. Rogers, H. Cohen and P. Straznicky. *Gas Turbine Theory*. Pearson Education, Essex, England, 6^{th} edition, 2009. ISBN 978-0-13-222437-6.

[85] Federal Aviation Administration. Noise Levels for U.S.Certificated and Foreign Aircraft: Advisory Circular, May 2012. AEE-100.

[86] A. Agarwal, A. P. Dowling, Ho-Chul Shin and W. Graham. A ray tracing approach to calculate acoustic shielding by the silent aircraft airframe. In *12th AIAA/CEAS Aeroacoustics Conference*, number AIAA 2006-2618, Cambridge, Massachusetts, May 2006.

Appendix A

APM Options file

This appendix serves to explain the structure of the APM options file and at the same time be a sample of an ISF. Each member's function is explained.

```
;Options_Custom_File_for_APM_Aeroacustics_Module
;Date_of_creation:  mmm dd yyyy
;Time_of_creation:  hh:mm:ss
;-----------------------------------------------------------------
[data_strictness]                    If the flag is set to 1 then APM checks the validity
pathDataStrict= 1                    of the corresponding data set before starting
aircraftDataStrict= 1                to compute acoustic predictions
engineDataStrict= 1
;-----------------------------------------------------------------
[path_settings]
pathInputAnglesAssumeNed= 1          If attitude angles are according to the RPY convention
pathInputAnglesInDegrees= 1          If angles in flight path are in degrees
searchPathforTimeRepeat= 1           If APM should search for time repetitions in path data
;-----------------------------------------------------------------
[aircraft_settings]
preRotateAircraft180= 1              If the aircraft is to be rotated by 180 deg to align with x axis
;-----------------------------------------------------------------
[save_settings]                      Flags which data will be auto-saved after running the solver
savePath= 1
saveAero= 1
saveGear= 1
saveMic= 1
saveNoiseLimit= 1
```

```
saveResults= 1
saveDestination= 2
;-------------------------------------------------------------
[check_settings]                                  Check file existence on start-up
checkHelpFiles= 1                                 and lock GUI for missing elements
checkModelFiles= 1
;-------------------------------------------------------------
[help_settings]                                   Chose Help documentation version
developerHelp= 1
helpTarget= 0
```

Appendix B

Engine input file

This appendix serves to explain the structure of the APM engine geometric constants file and component selection. This was the engine used for the example case of section 4.6.

```
;Engine_File_for_APM
;Date_of_creation:  mmm dd yyyy
;Time_of_creation:  hh:mm:ss
;-----------------------------------------------------------------
[engine_general]
Engine_ID= 1
Position_x= -14.32
Position_y= 4.67
Position_z= -1.22
Engine_Reference_Area_m2= 1.8869
;-----------------------------------------------------------------
[fan_model]
To_Be_Used_Flag= 1
Guide_Vane_Index= 0
Flow_Distortion_Index= 0
Rotor_Stator_Spacing_Index= 1
Fan_Cross_Sectional_Area= 1.0
Fan_Rotor_Diameter= 1.128
Rotor_Stator_Spacing= 2.33
Number_Stator_Vanes= 68
Number_Rotor_Blades= 36
Desing_Mach_Number= 1
Fan_Rotor_Relative_Tip_Mach_Design= 1
```

```
;-------------------------------------------------------------
[fan_components]
Inlet_Broadband_To_Be_Used_Flag= 1
Inlet_Interaction_To_Be_Used_Flag= 1
Inlet_Distortion_To_Be_Used_Flag= 1
Inlet_Combination_To_Be_Used_Flag= 1
Discharge_Broadband_To_Be_Used_Flag= 1
Discharge_Interaction_To_Be_Used_Flag= 1
;-------------------------------------------------------------
[combustor_model]
To_Be_Used_Flag= 1
Combustor_Cross_Sectional_Area= 0.17
;-------------------------------------------------------------
[turbine_model]
To_Be_Used_Flag= 0
Turbine_Cross_Sectional_Area= 1
Turbine_Rotor_Diameter= 0
Number_Rotor_Blades= 0
;-------------------------------------------------------------
[turbine_components]
Broadband_To_Be_Used_Flag= 0
Pure_Tone_To_Be_Used_Flag= 0
;-------------------------------------------------------------
[jet_model]
To_Be_Used_Flag= 1
;-------------------------------------------------------------
[jet_components]
Mixing_To_Be_Used_Flag= 1
Shock_Tone_To_Be_Used_Flag= 1
```

Appendix C

One Third Octave Bands

In this Appendix the used ISO 1/3 Octave Band frequencies are presented for reference. These are applied in all source modelling and noise level calculations. They were retrieved from references [30, 56].

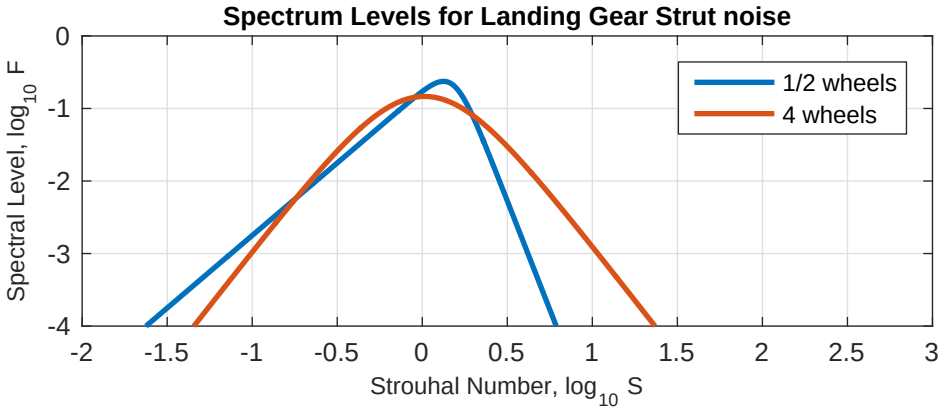
Table C.1: Table of One Third Octave Bands.

Band Number	f_l	f_c	f_u
1	44.7	50	56.2
2	56.2	63	70.8
3	70.8	80	89.1
4	89.1	100	112.2
5	112.2	125	141.3
6	141.3	160	177.8
7	177.8	200	223.9
8	223.9	250	281.8
9	281.8	315	354.8
10	354.8	400	446.7
11	446.7	500	562.3
12	562.3	630	707.9
13	707.9	800	891.3
14	891.3	1000	1122
15	1122	1250	1413
16	1413	1600	1778
17	1778	2000	2239
18	2239	2500	2818
19	2818	3150	3548
20	3548	4000	4467
21	4467	5000	5623
22	5623	6300	7079
23	7079	8000	8913
24	8913	10000	11220

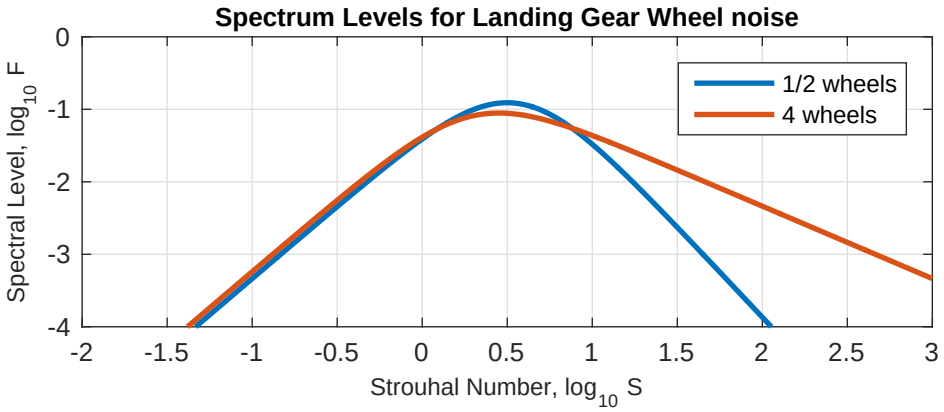
Appendix D

Landing Gear Component Verification

In this appendix the landing gear spectral functions are shown (Figure D.1). They inform the user of what spectrum to respect from the landing gear and can be readily compared with the reference graphs for validation [30].



(a) Strut noise.



(b) Wheel noise.

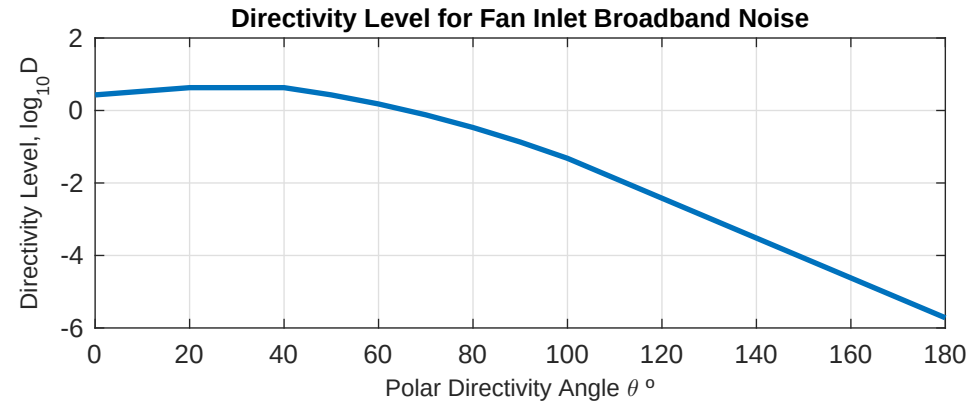
Figure D.1: Spectrum Levels for Landing Gear.

Appendix E

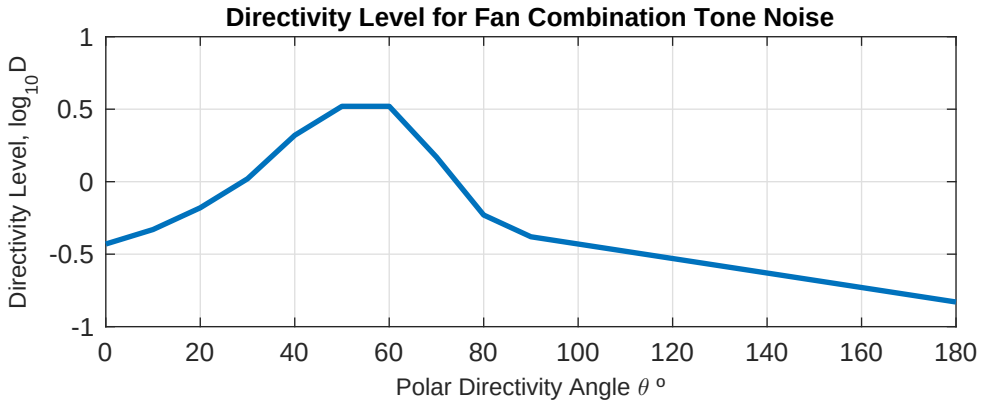
Engine Component Verification

This appendix shows the directivity, spectral and power functions as implemented in APM. The tables shown (Figures E.1 to E.10) can be readily compared with the reference graphs for validation [30].

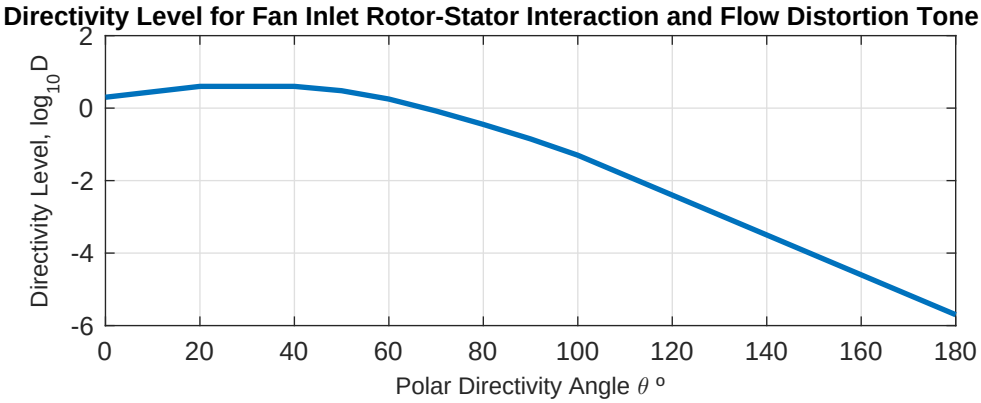
Fan Tables



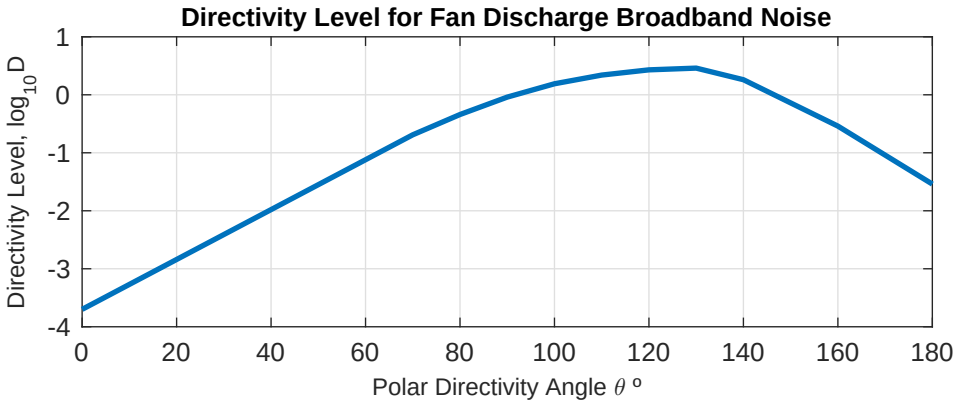
(a) Inlet Broadband Noise.



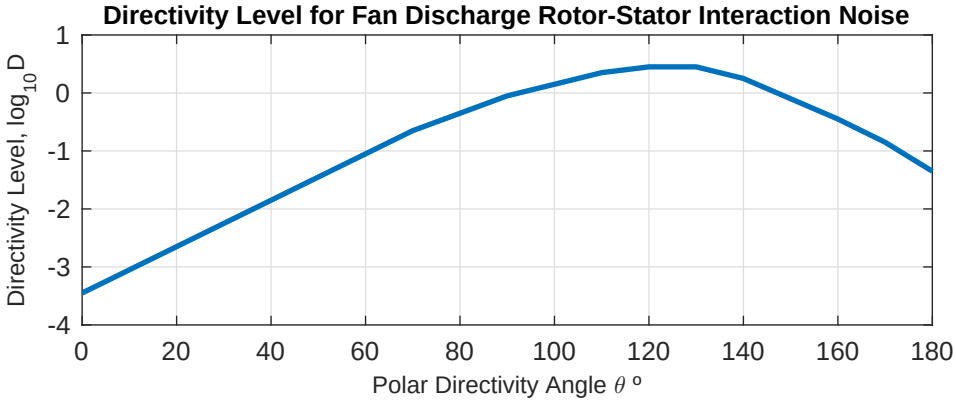
(b) Combination Tone Noise.



(c) Rotor-stator Interaction and Flow Distortion Tone Noise.

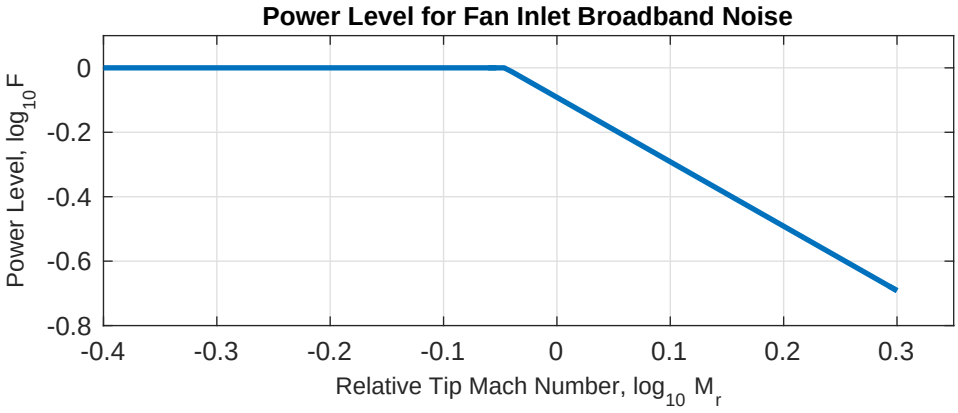


(d) Discharge Broadband Noise.

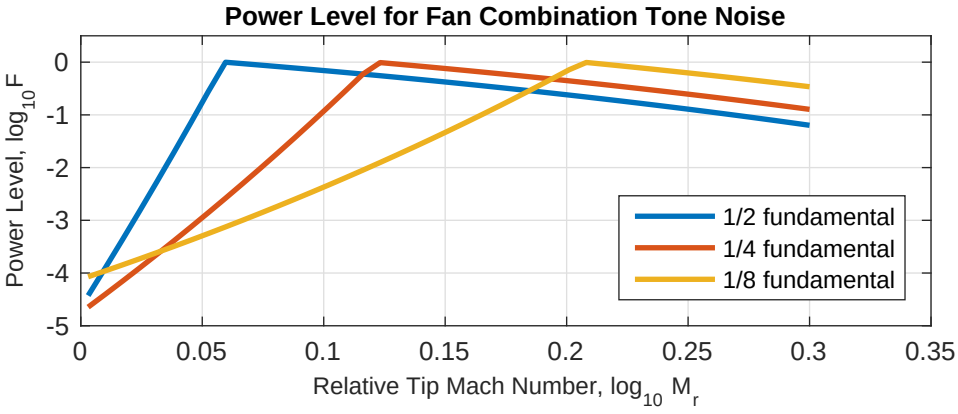


(e) Discharge Rotor-stator Interaction Noise.

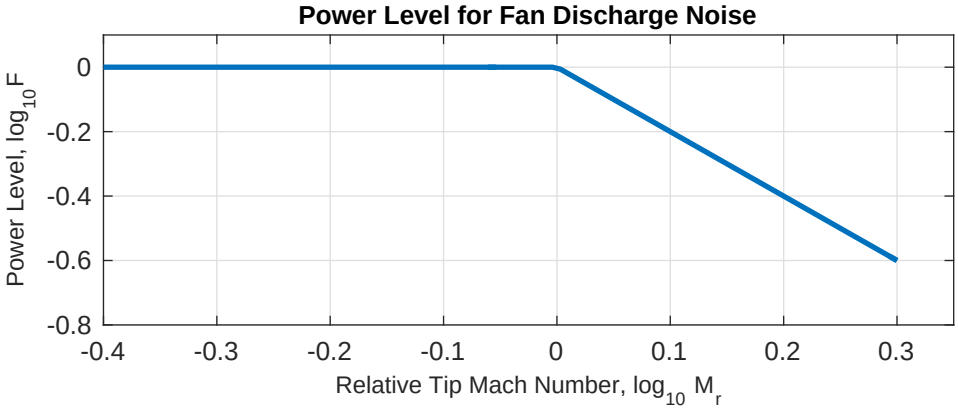
Figure E.1: Directivity Level for Fan Components.



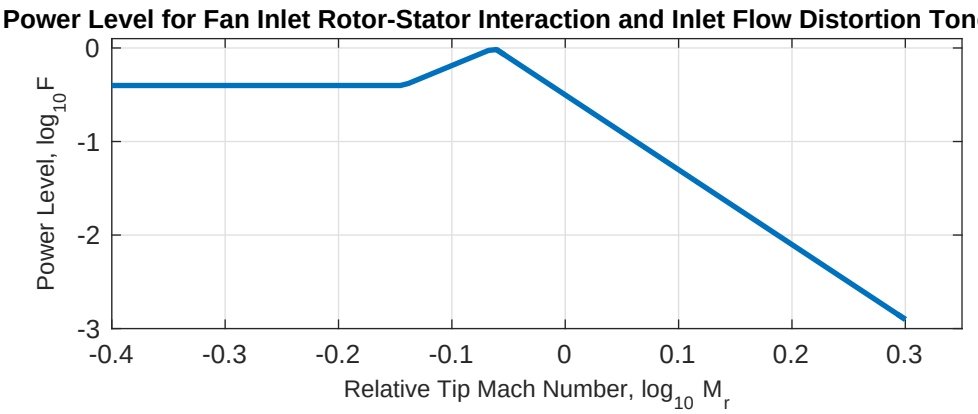
(a) Inlet Broadband Noise.



(b) Combination Tone Noise.

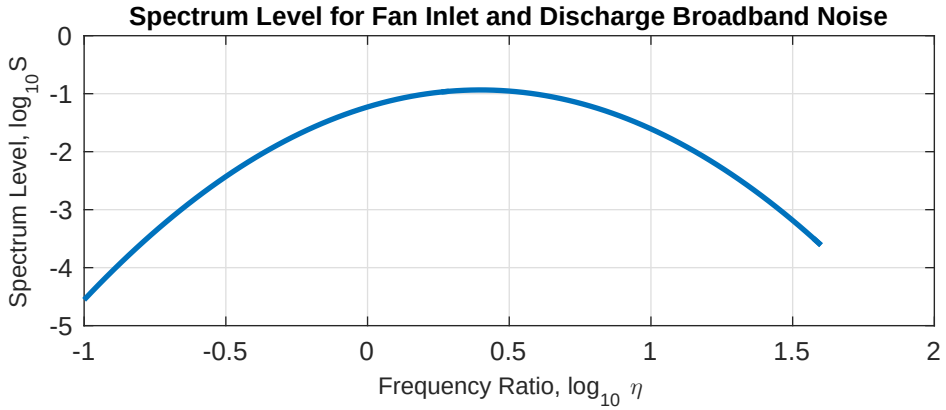


(c) Discharge noise.

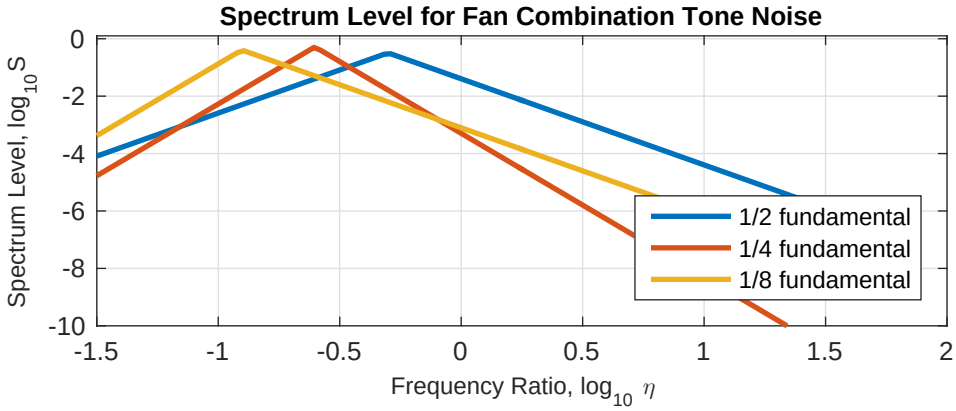


(d) Inlet Rotor-stator Interaction and Inlet Flow Distortion Tone Noise.

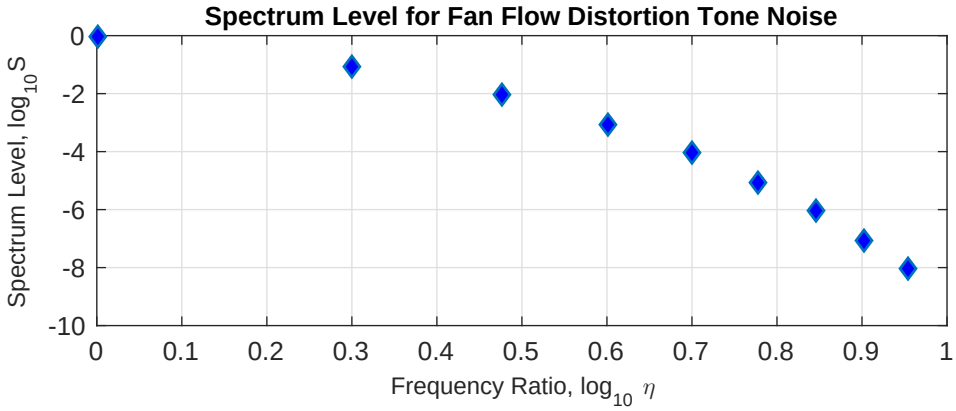
Figure E.2: Power Level for Fan Components.



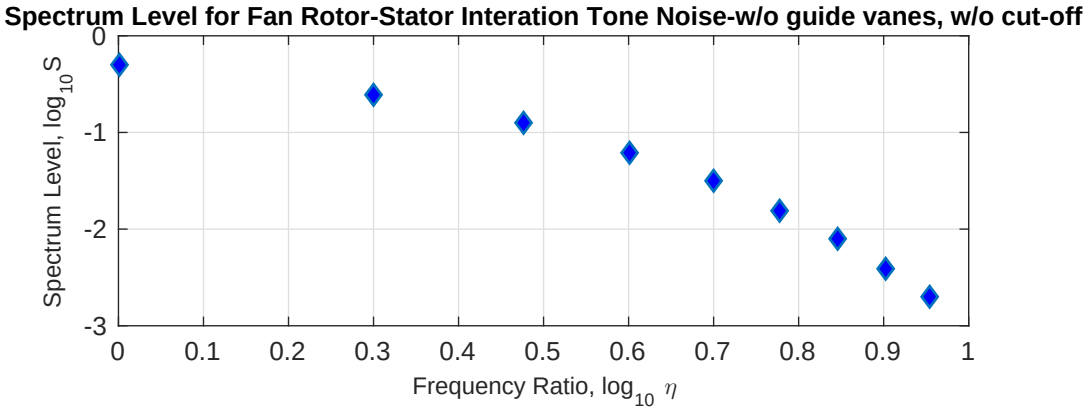
(a) Inlet and Discharge Broadband Noise.



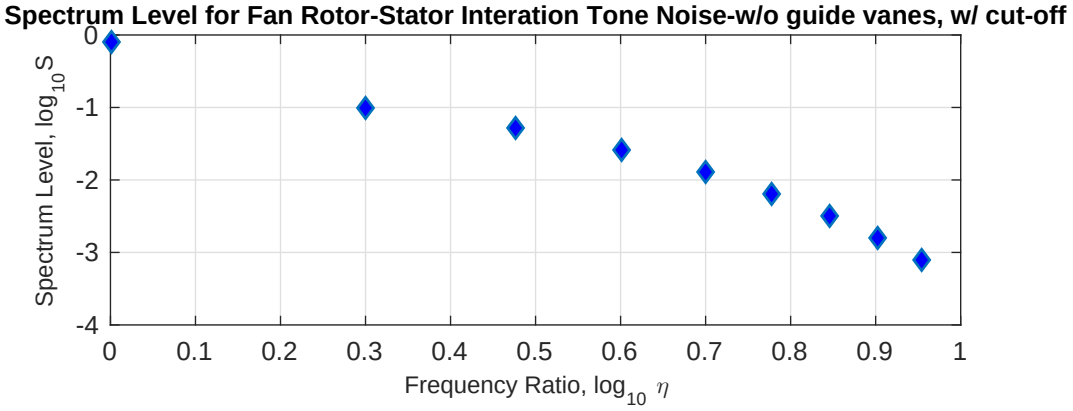
(b) Combination Tone Noise.



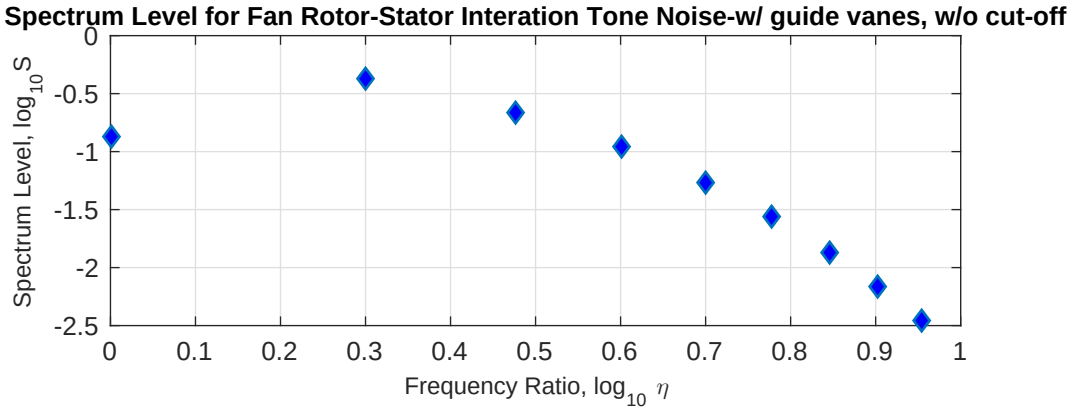
(c) Flow Distortion Tone Noise.



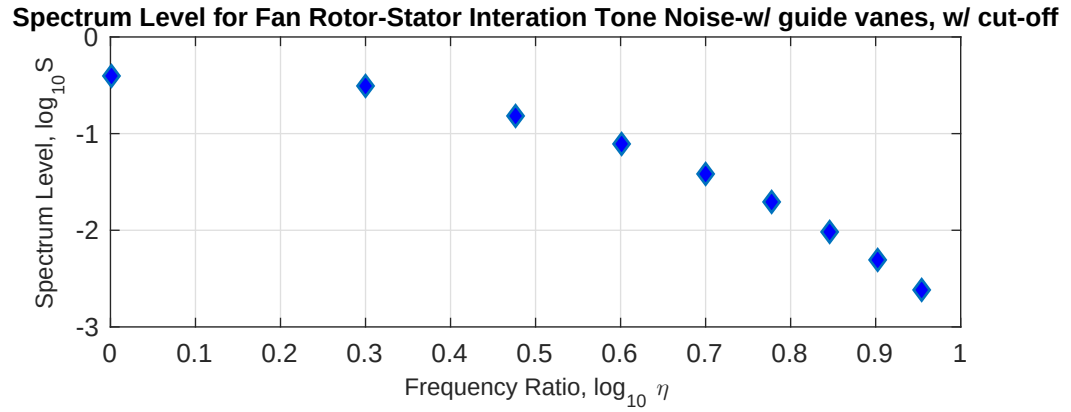
(d) Rotor-Stator Interaction Tone Noise-w/0 guide vanes, w/o cut-off.



(e) Rotor-Stator Interaction Tone-w/o guide vanes, w/ cut-off.



(f) Rotor-Stator Interaction Tone Noise-w/ guide vanes, w/o cutt-off.



(g) Rotor-Stator Interaction Tone Noise-w/ guide vanes, w/ cut-off.

Figure E.3: Spectrum Level for Fan Components.

Combustor Tables

The combustor noise prediction is given by a single noise component.

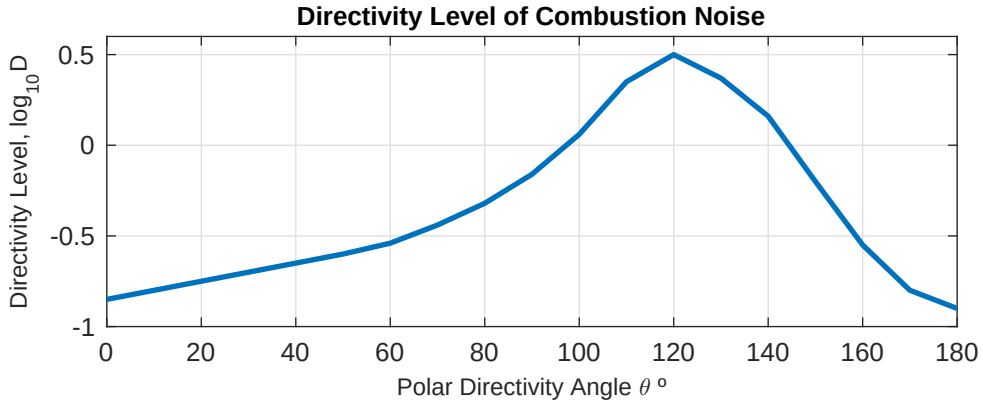


Figure E.4: Directivity Level of Combustion Noise.

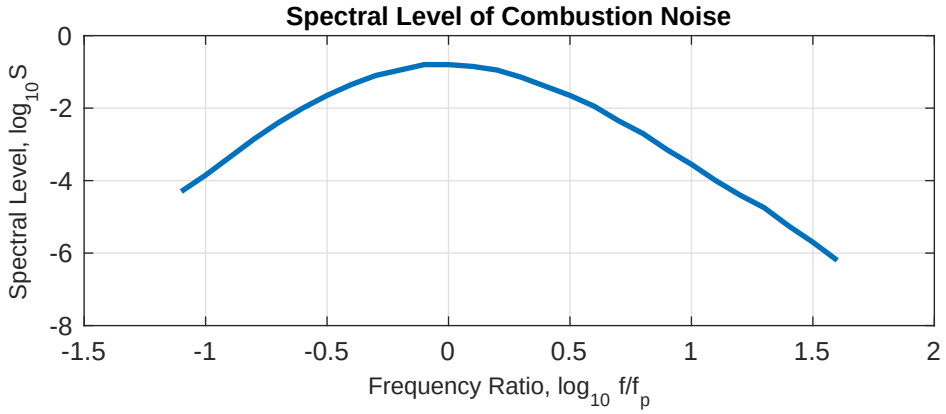
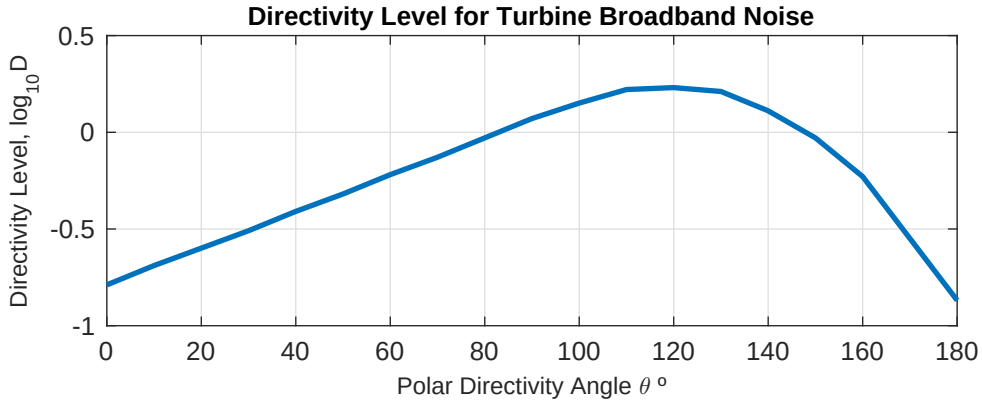
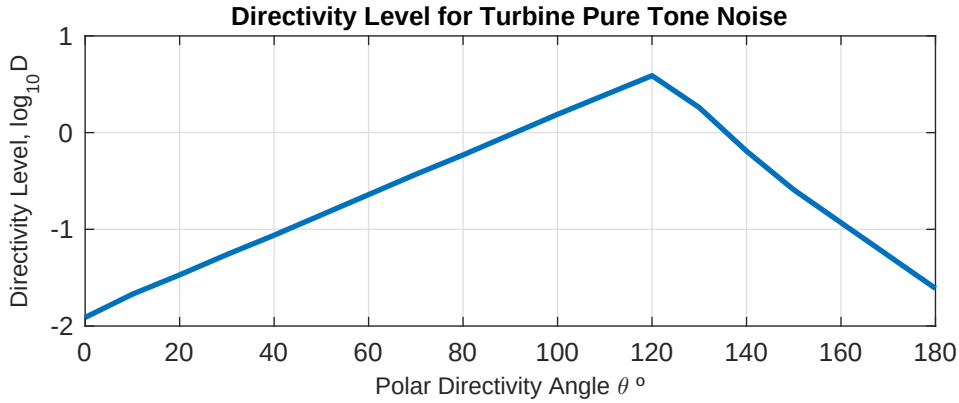


Figure E.5: Spectral Level of Combustion Noise.

Turbine Tables

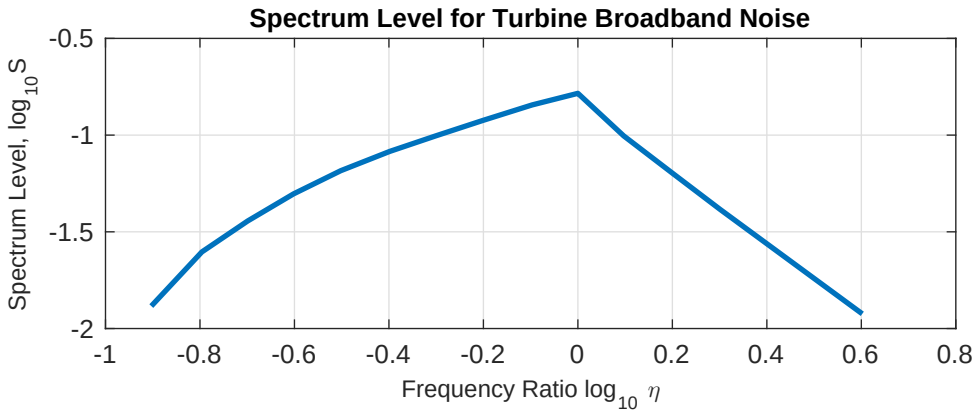


(a) Broadband Noise.

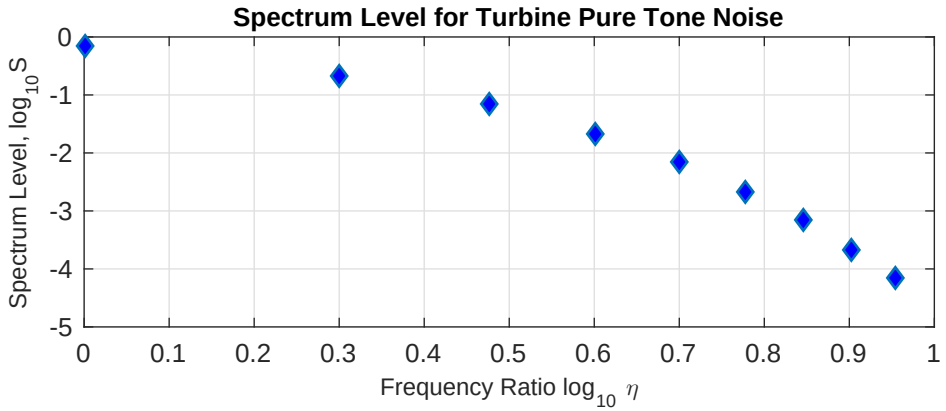


(b) Pure Tone Noise.

Figure E.6: Directivity Level for Turbine Components.



(a) Broadband Noise.



(b) Pure Tone Noise.

Figure E.7: Spectrum Level for Turbine Components.

Jet Tables

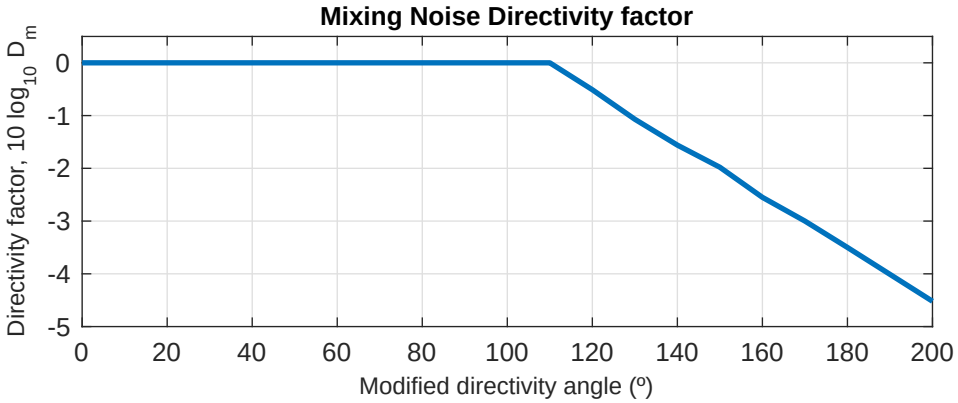
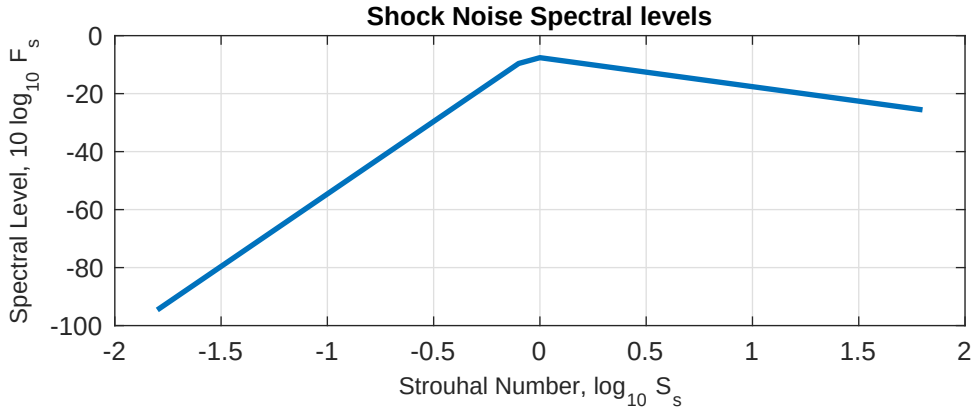
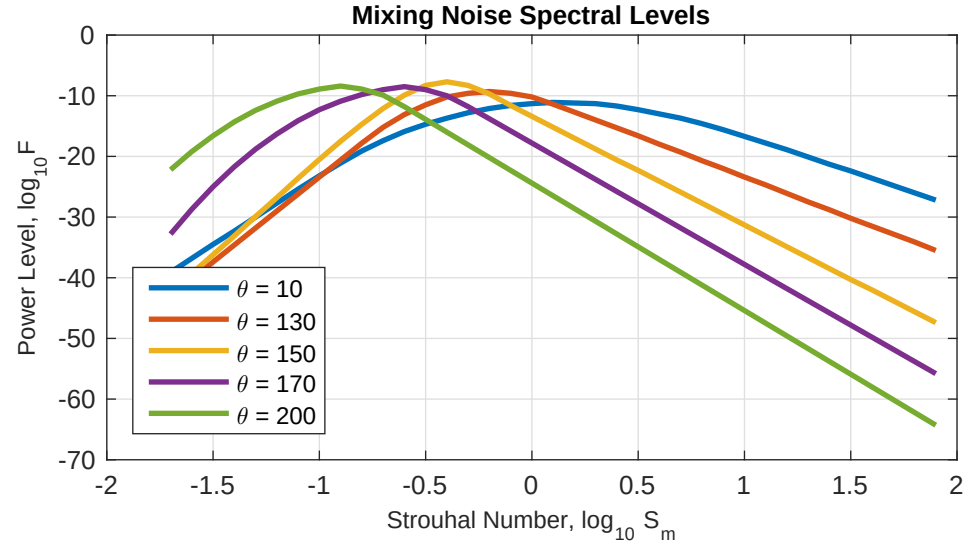


Figure E.8: Jet Mixing Noise Directivity Factor.



(a) Shock Noise.



(b) Mixing Noise.

Figure E.9: Spectral Levels for Jet Components.

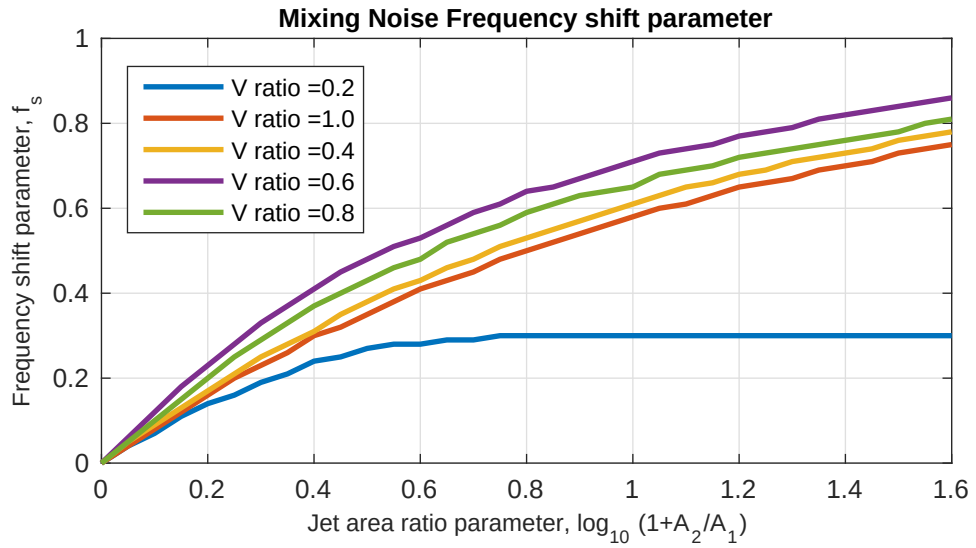


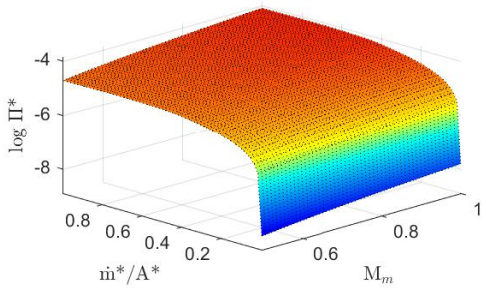
Figure E.10: Jet Mixing Noise Frequency Shift Parameter.

Appendix F

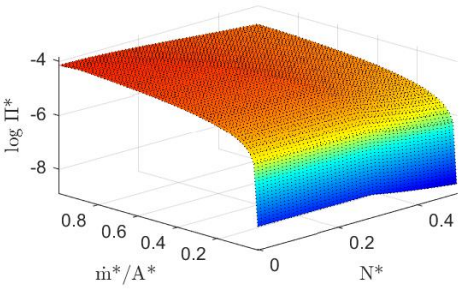
Input Impact on Engine Components

This appendix shows the impacts of the most important input parameters on the acoustic power functions (Π*) of all engine noise models.

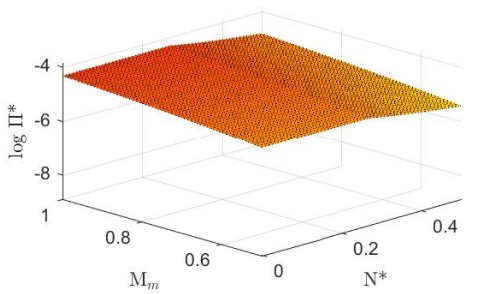
Fan power



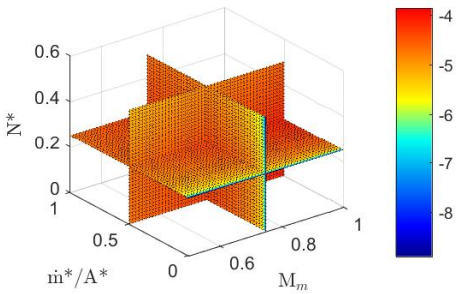
(a) M_m *vs* ṁ*/A*



(b) N* *vs* ṁ*/A*

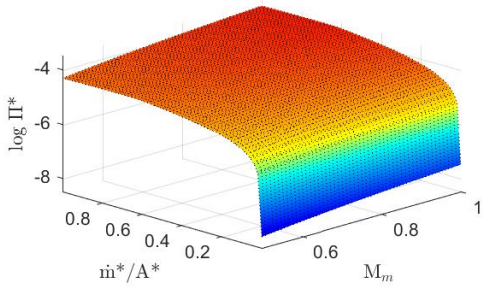


(c) M_m *vs* N*

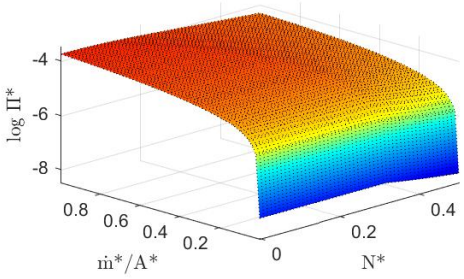


(d) log Π* in function of M_m, N* and ṁ*/A*

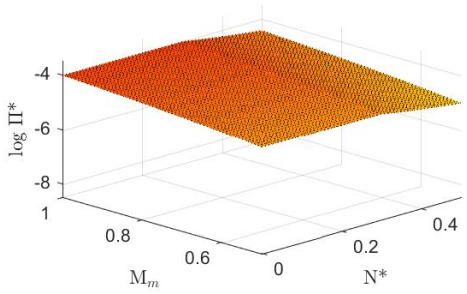
Figure F.1: Fan Inlet Broadband component acoustic power (log Π*).



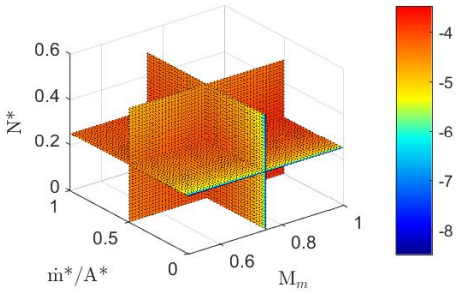
(a) M_m *vs* ṁ/A*



(b) N* *vs* ṁ/A*

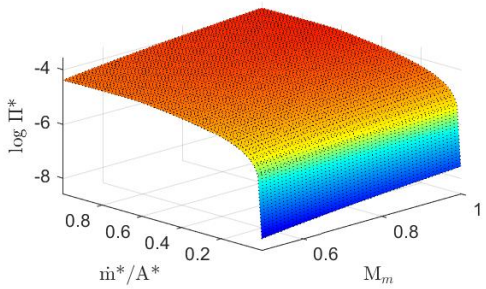


(c) M_m *vs* N*

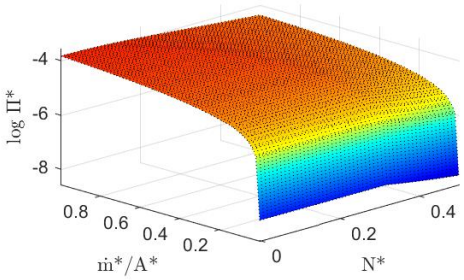


(d) log Π* in function of M_m, N* and ṁ*/A*

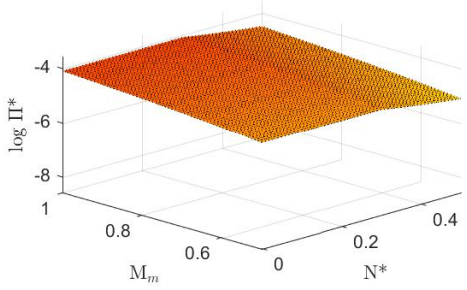
Figure F.2: Fan Discharge Broadband component acoustic power (log Π*).



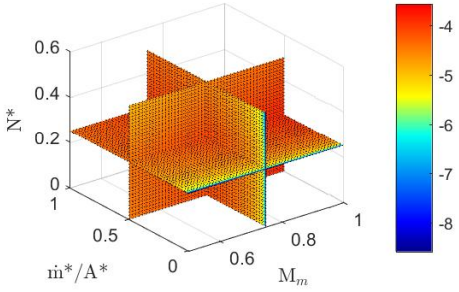
(a) M_m *vs* ṁ/A*



(b) N* *vs* ṁ/A*

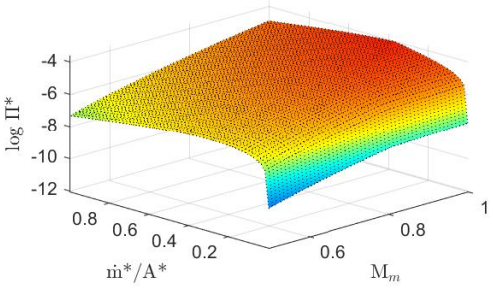


(c) M_m *vs* N*

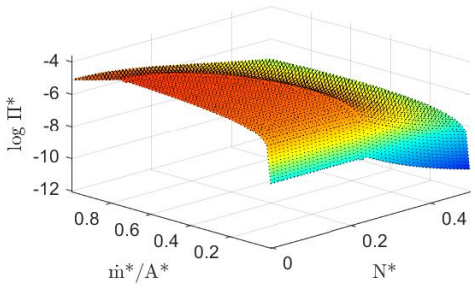


(d) log Π* in function of M_m, N* and ṁ*/A*

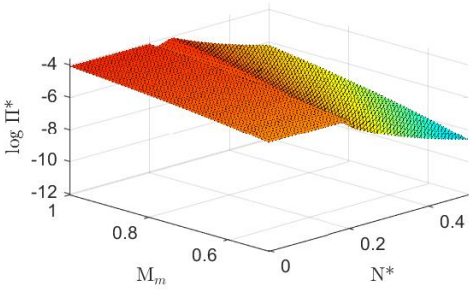
Figure F.3: Fan Discharge Rotor-Stator Interaction component acoustic power (log Π*).



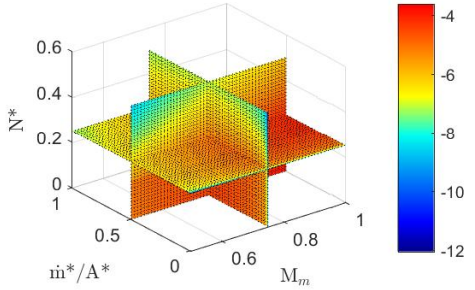
(a) M_m *vs* ṁ*/A*



(b) N* *vs* ṁ*/A*

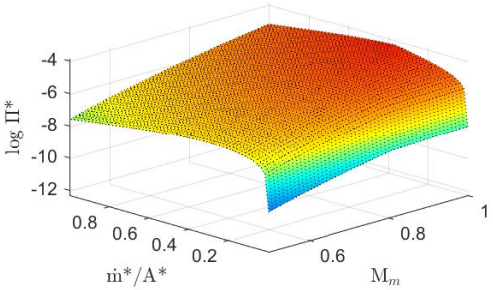


(c) M_m *vs* N*

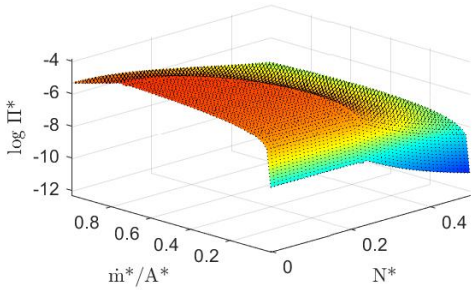


(d) log Π* in function of M_m, N* and ṁ*/A*

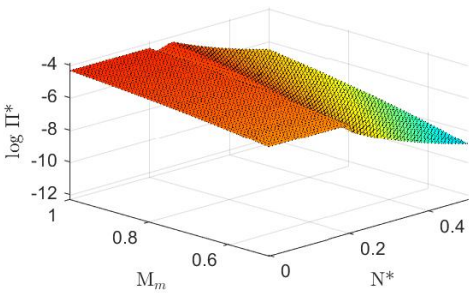
Figure F.4: Fan Inlet Rotor-Stator Interaction component acoustic power (log Π*).



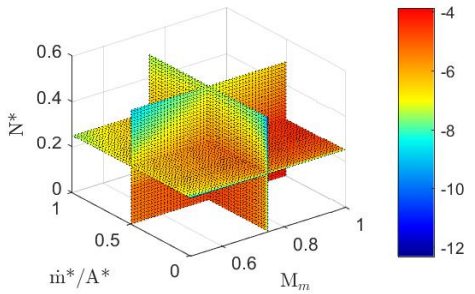
(a) M_m *vs* ṁ*/A*



(b) N* *vs* ṁ*/A*

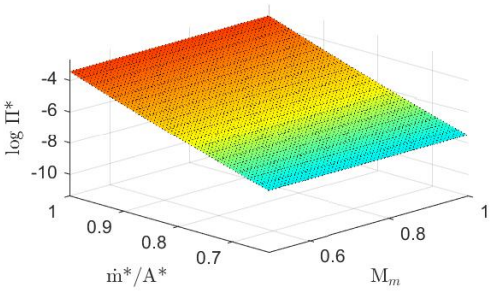


(c) M_m *vs* N*

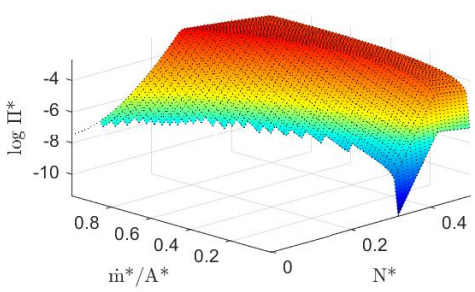


(d) log Π* in function of M_m, N* and ṁ*/A*

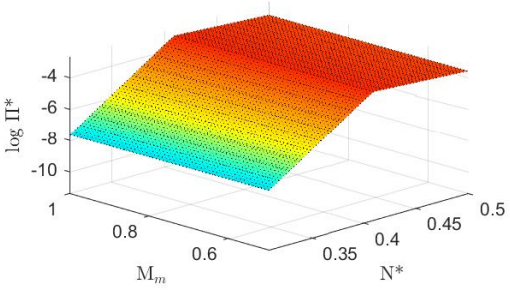
Figure F.5: Fan Inlet Distortion component acoustic power (log Π*).



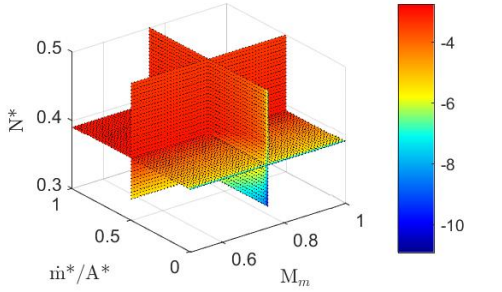
(a) M_m *vs* ṁ*/A*



(b) N* *vs* ṁ*/A*



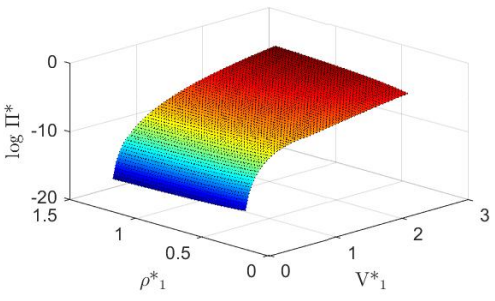
(c) M_m *vs* N*



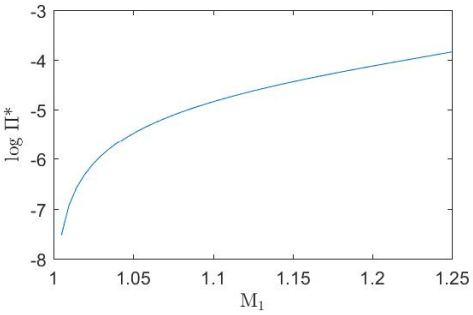
(d) log Π* in function of M_m, N* and ṁ*/A*

Figure F.6: Fan Inlet Combination Tone component acoustic power (log Π*).

Jet power



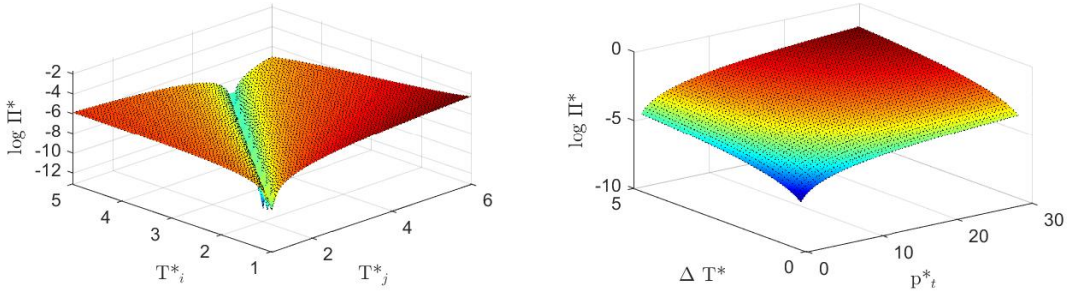
(a) V_1* *vs* ρ_1* in Mixing component



(b) M_1 impact in Shock component

Figure F.7: Jet acoustic power (log Π*).

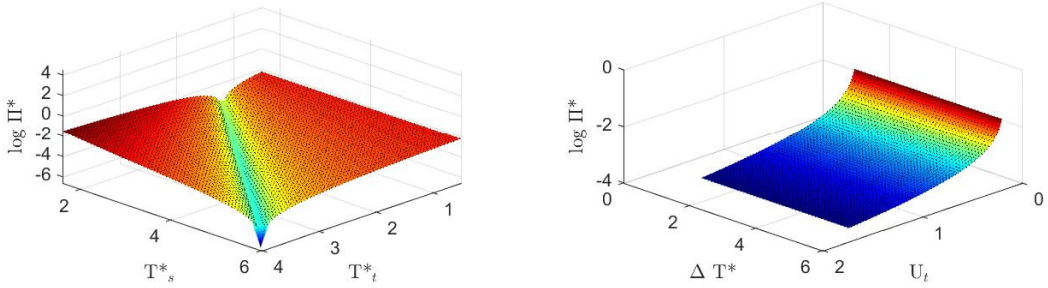
Combustor power



(a) T_i* *vs* T_j* (b) ΔT* *vs* p_t*

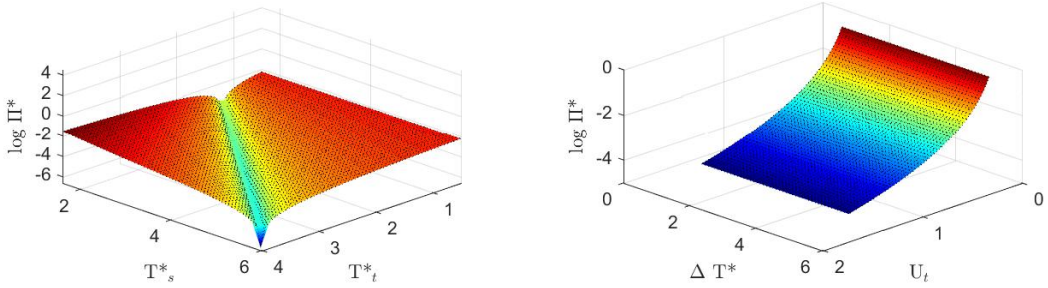
Figure F.8: Combustor acoustic power (log Π*).

Turbine power



(a) T_t* *vs* T_s* (b) ΔT* *vs* U_t*

Figure F.9: Turbine Broadband component acoustic power (log Π*).



(a) T_t* *vs* T_s* (b) ΔT* *vs* U_t*

Figure F.10: Turbine Pure Tone component acoustic power (log Π*).