End-of-studies internship report
& Master's thesis

Aeroacoustic Optimization of Rotors Using a Vortex Particle Method Solver

Author

Manuel GARCÍA LUCAS

Institution	Supervisor
Capgemini Engineering	*Enrico PICENI*
ISAE-SUPAERO	*Romain GOJON*
Universitat Politècnica de València	*Roberto NAVARRO GARCÍA*

— Toulouse, October 2nd 2024 —

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Abstract

Urban air mobility vehicles, in particular eVTOL, have recently shown their potential to revolutionize the aerospace sector in the near future. However, these concepts currently present noise pollution issues, impacting the densely populated areas in which they are supposed to operate. Propellers must be designed to minimize not only their power consumption, but also their noise emission. The objective of this work is to evaluate the performance of a mid-fidelity aerodynamic solver based on the Vortex Particle Method in the process of aeroacoustic optimization of rotors. This solver is integrated in a multidisciplinary optimization loop using the Ffowcs William-Hawkings analogy for the aeroacoustics. Once the optimization workflow is completed, a surrogate model is created and trained to substitute the aerodynamic solver, allowing for a significantly faster optimization. Parallel to this, the analysis of ducted rotors is included.

Keywords: Aeroacoustics; Ducted Rotors; Multi-Disciplinary Optimization; Rotor Optimization; Surrogate Optimization; Vortex Particle Method

Resumen

Los vehículos de movilidad aérea urbana, en particular los eVTOL, han demostrado recientemente su potencial para revolucionar el sector aeroespacial en el futuro próximo. Sin embargo, estos conceptos presentan actualmente problemas de contaminación acústica, lo que repercute en las zonas densamente pobladas en las que están destinados a operar. Las hélices deben diseñarse para minimizar no solo su consumo de energía, sino también su emisión de ruido. El objetivo de este trabajo es evaluar el rendimiento de un solver aerodinámico de fidelidad media basado en el Vortex Particle Method en el proceso de optimización aeroacústica de rotores. Este solver se integra en un bucle de optimización multidisciplinar utilizando la analogía Ffowcs William-Hawkings para la aeroacústica. Una vez completada la herramienta de optimización, se crea y entrena un modelo subrogado para sustituir al solver aerodinamico, lo cual permite una optimización significativamente más rápida. En paralelo a esto, se incluye el análisis de rotores carenados.

Palabras clave: Aeroacústica; Optimización multidisciplinar; Optimización de rotores; Optimización subrogada; Rotores carenados; Vortex Particle Method

Contents

List of Acronyms

BEM boundary element method

BIEM boundary integral equation method

BPF blade passing frequency

BPM Brooks, Pope and Marcolini

CAD computer aided design

CFD computational fluid dynamics

COBYLA Constrained Optimization BY Linear Approximation

CST Class-Shape function Transformation

DOE design of experiments

DP design points

eVTOL electric vertical take-off and landing

F1A Farassat 1A

FFD full factorial design

FW-H Ffowcs Williams-Hawkings

GEMSEO Generic Engine for Multidisciplinary Scenarios, Exploration and Optimization

GPR Gaussian process regressor

HK hierarchical Kriging

LHS Latin hypercube sampling

MSE mean square error

OASPL overall sound pressure level

R&D Research & Development

RANS Reynolds-averaged Navier-Stokes

RBF radial basis function

RMSE root mean square error

TEC Technology Engineering Center

VPM vortex particle method

VTOL vertical take-off and landing

List of Figures

List of Listings

List of Tables

1. Introduction

Urban air mobility vehicles, in particular eVTOL (electric vertical take-off and landing), have recently shown their potential to revolutionize the aerospace sector in the near future, arising as greener alternatives to traditional transportation systems. However, these concepts currently present noise pollution issues, impacting the densely populated areas in which they are supposed to operate.

The present work evaluates the performance of a mid-fidelity aerodynamic solver based on the vortex particle method (VPM) in the process of aeroacoustic optimization of rotors. This solver is integrated in a multidisciplinary optimization loop using the Ffowcs Williams-Hawkings (FW-H) analogy [1] for the aeroacoustics. The main objective is to develop a workflow for the optimization of propeller design that minimizes not only their power consumption, but also their noise emission, in order to overcome the previously mentioned limitations of eVTOL.

This is intended to be the continuation and improvement of the work of Piceni [2]. There, the author developed a first version of the workflow using Python library GEMSEO (Generic Engine for Multidisciplinary Scenarios, Exploration and Optimization) [3]. The aerodynamic solver used was an internal, low-fidelity CFD solver developed by Capgemini – QuickCFD – which allowed a fast exploration of the design space. The acoustics were solved by inputting the aerodynamic results into PyFfonc, another of Capgemini's internal tools. PyFfonc [4], [5] is a Python library that implements Farassat 1A (F1A) formulation [6] of the FW-H acoustic analogy.

The first part of this work has consisted in resuming the already initiated substitution of the low-fidelity aerodynamic solver for DUST, a mid-fidelity tool that implements the VPM. Then, two tasks have been performed. On the one hand, a surrogate model for the aerodynamics of the rotor has been created, so as to accelerate the optimization process. On the other, the calculation of the sound scattering caused by a duct has been implemented so that shrouded rotors can be analyzed too, something that was initially developed by Saliba P. R. [7] in his internship in this same company. Thus, this work tackles the optimization problem from diverse angles and covers different aspects of the process of aeroacoustic optimization of rotors.

2. Presentation of the company

2.1 Capgemini Engineering

Capgemini Engineering (formerly known as Altran Technologies) is part of the Capgemini group, focused on engineering solutions, disruptive technologies and R&D. By taking advantage of in-depth engineering and digital knowledge, Capgemini Engineering supports their partners bringing together the digital and physical world, accelerating the transition to the sustainable Intelligent Industry. This goal is executed by the company's more than 55000 employees in more than 30 countries, across various sectors, such as aeronautics, aerospace, industrial engineering, rail transportation, communications, health and automotive.

Inside the company, the Technology Engineering Center (TEC), located in Toulouse, has more than 2000 engineers and scientists spread into many different domains, including mechanical, fluid and thermal and electrical engineering, acoustics, flight science, telecommunications and many others. Additionally, it is also home to the TEC R&D department, where more than 100 researchers work on internal projects, ranging from future urban mobility to sustainable energy and industries.

2.2 Pôle Acoustique

Inside the TEC and its Physical & Mechanical Engineering perimeter is the Pôle Acoustique, working to reduce noise in many applications, complying with ever-increasing regulatory demands and improving user comfort. Among the Pôle's capabilities is the know-how in different fields such as aeroacoustics, vibroacoustics, psychoacoustics and underwater noise. This way, it is capable of tackling various acoustic challenges ranging from modeling complex turbomachinery flow and noise to predicting off-shore wind farm noise propagation and its impact on marine fauna.

Noise predictions are an important part of the design process, being seamlessly included in the design loop, by bringing together the team's expertise in different parts of the process, including CAD, meshing, CFD or computational aeroacoustics, post-processing and optimization, providing complete engineering solutions to clients and partners, whether it is by employing advanced modeling tools and methods or by participating in various troubleshooting activities related to acoustics and noise.

2.3 Project OPTIMIND

OPTIMIND is a project with the objectives of analyzing, understanding and modeling multiphysical phenomena such as aerodynamics, vibration, acoustics, structures, etc. The goal is to optimize models and solutions on multi-sector problems, by means of multiphysical approaches. This allows the development of applied solutions on concrete use cases such as R&D projects or customer and partner applications.

Some of the research paths of the project are flutter prediction for aicraft, life-cycle optimization, high-fidelity fluid-structure interaction, cabin noise interior optimization, audio perception and overall aicraft design.

3. Literature review

3.1 DUST

DUST is a mid-fidelity aerodynamic solver that results from a collaboration between the Politecnico di Milano and A^3 by Airbus. Unlike QuickCFD, that only allows for the use of lifting-line elements and prescribed wakes, DUST is based on a combination of different potential elements and vortex particles, thanks to relying on the Helmholtz decomposition of the velocity field into an irrotational and a solenoidal contribution. Quoting the developers [8]:

> Different aerodynamic elements allow for different levels of fidelity in the model, ranging from lifting line elements to zero-thickness lifting surfaces and surface panels. Lifting bodies shed a wake from their trailing edges, that can be represented by vortex lattice sheets, vortex particles or a combination of the two.

This means that apart from the distribution of singularities on the geometry generated for each aerodynamic element – one-dimensional for lifting lines, two-dimensional for vortex lattices and three-dimensional for surface panels – the trailing edges of lifting surfaces produce a number of vortex sheets that then turn into a set of vortex particles, generating a wake that is convected downstream. All of these singularities are taken into consideration by the panel method solver.

The program works using three executables: `dust_pre`, `dust` and `dust_post`, which pre-process the geometry, run the simulation, and post-process the results, respectively [9]. The classical DUST workflow is shown in Figure 3.1.

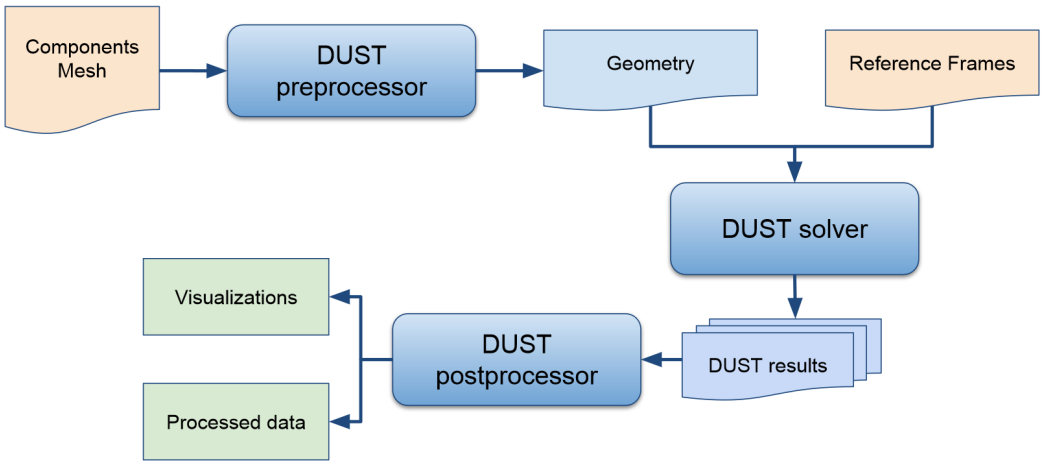

Figure 3.1: Classical DUST workflow, consisting of the pre-process, the simulation and the post-process [9].

The results obtained with DUST are of similar accuracy to those of Finite Volume Method CFD calculations at a fraction of their computational cost, except for strongly detached flow conditions [8], [10]. One of the validations performed was the simulation of

the Vahana vehicle – the eVTOL multi-rotor tiltwing aircraft built by A^3 (see Figure 3.2) [10]. Relative errors in vertical equilibrium of about 6.5%, less than 1% and 2% were found when compared with the results of the flight test campaign in descent, hover, and climb flight, respectively. As for the power consumed by the rotors, a discrepancy below 5% was obtained when comparing with high-fidelity CFD simulations.

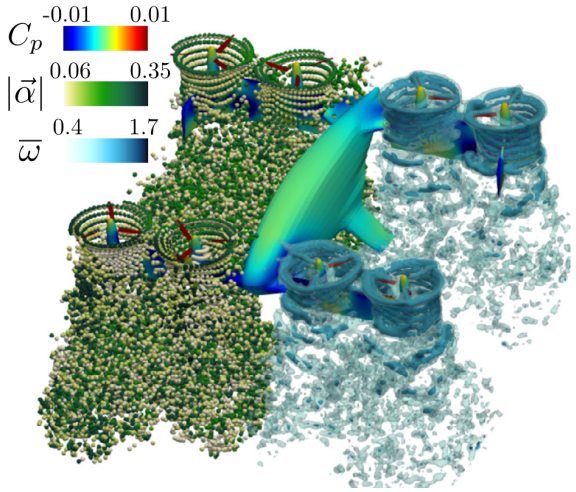

Figure 3.2: DUST simulation of the Vahana Alpha in hover. Surface visualization of pressure coefficient based on rotor tip velocity, in rainbow scale. Right side of the vehicle: visualization of the wake particles with intensity, in green scale. Left side of the vehicle: iso-contours of vorticity, in blue scale [10].

In addition to unconventional VTOL vehicles, the code has been validated for wing-propeller interactions [11], which makes it highly interesting for this application.

The choice of DUST as an aerodynamic solver is not only based on technical reasons, but also practicality: one of the objectives of this work is to increase the fidelity of the workflow designed by Piceni without resorting to costly, higher-fidelity RANS CFD that would render the simulation time prohibitive for the purpose of such an optimization.

3.2 Rotor noise

There are many examples of approaches similar to the one implemented in PyFfonc being used in the literature. Zawodny *et al.* [12] compared two rotors using three different approaches – experimental, unsteady RANS together with F1A formulation and Blade Element Analysis with F1A. They concluded that a higher rotor tip speed is strongly related with an increased overall sound pressure level (OASPL), directivity and broadband noise. McKay and Kingan [13] also compared results from Blade Element Momentum Theory coupled with F1A with experimental ones. Wang *et al.* [14] developed a methodology for the improvement of the aerodynamic and aeroacoustic performance using both experimental and improved delayed detached eddy simulation coupled with FW-H equation. They determined that the vortices shed from the blade tip are stronger than those of the trailing edge.

There are also instances of the use of VPM together with acoustic calculations. Huberson *et al.* [15] employed the Lighthill analogy and VPM to study vortex-solid interactions. Parmentier *et al.* [16] developed a remeshed VPM for compressible flows, using the Möhring

analogy for far-field noise. Alvarez *et al.* [17] studied rotor-on-rotor interactions on multi-rotor vehicles by means of two different approaches – unsteady RANS simulation coupled with FW-H and Brooks, Pope and Marcolini (BPM) method [18] for broadband noise; and a viscous VPM method coupled with FW-H and BPM.

3.3 Ducted rotors

The code of Saliba P. R. [7] for simulating the noise generation and propagation on ducted rotors was incorporated into the workflow. He devised two modules for the noise scattering, one based on the boundary integral equation method (BIEM) following a cylindrical duct approximation, the other based on the boundary element method (BEM) in which generic axisymmetric ducts can be modeled. Similarly to ours, his work is also a continuation of that of Piceni, so the integration was relatively straightforward.

The results of Saliba P. R. were in accordance to the analytical models – namely Froude's theory for ducted rotors – in that the duct is able to increase the total thrust while reducing or maintaining energy consumption. Furthermore, the fairing partially shields the noise produced by the rotor, diminishing its OASPL. Additionally, Shukla and Komerath [19] had previously concluded that it can improve the figure of merit of the rotor without gain in the OASPL. These advantages are a result of the blade tip vortices interaction with the duct, therefore the effect is enhanced when the vortex core - duct interactions are important.

3.4 Surrogate modeling

A surrogate model is a mathematical model that tries to replicate the behavior of a system without actually simulating it. In this case, a model for the aerodynamics of the propeller is devised with the purpose of recreating the results of DUST at a much lower computational cost.

Regarding the surrogate algorithm, the hierarchical Kriging (HK) algorithm seems to be a widely adopted choice when dealing with the aerodynamics of airfoils, wings and rotors. For instance, Han and Görtz [20] developed a variable-fidelity surrogate model of an airfoil and a full aircraft using both traditional Kriging – also known as Gaussian process regressor (GPR) – and HK algorithms. They found HK to be the most accurate and efficient and confirmed its suitability for aerodynamic analysis and shape optimization.

Xu *et al.* [21] also compared both algorithms in the optimization of high-altitude propellers and obtained similar conclusions: HK required 24.3% less optimization time. Leusink *et al.* [22] carried out a multi-objective design of helicopter rotor blades to optimize both hover and forward flight using a genetic algorithm and HK. Furthermore, Pham *et al.* [23] modeled the aerodynamics of an eVTOL aircraft by means of an extended HK algorithm, in which a combination of non-level low fidelity datasets were used .

Other algorithms apart from Kriging and hierarchical Kriging models have been used too, such as radial basis function (RBF). Jiang *et al.* [24] performed a multi-objective optimization on a helicopter rotor, in this case optimizing both the aerodynamics and the radar cross section using RBF. Sessarego *et al.* [25] employed RBF for wind-turbine optimization as well. They had an interesting surrogate approach divided in two stages – a global search by brute force followed by a infill strategy and a local search using gradient based optimization.

In addition, surrogate models have been used not only to optimize the aerodynamics but also the acoustics, in a similar fashion as the present work aims to do. Kwon *et al.* [26]

performed the aerodynamic and aeroacoustic optimization of a low-noise, coaxial contrarotating open rotor using a dual Kriging algorithm, which was built with RANS CFD data for the aerodynamics and F1A formulation of the FW-H equation for the acoustics. Finally, Bu *et al.* [27] optimized a helicopter rotor blade by means of a HK model fed with RANS simulations and FW-H with penetrable data surface. They also determined that HK saves 35% of the time with respect to the traditional Kriging algorithm. All in all, hierarchical Kriging and its variations are the most used algorithms.

As for the sampling techniques, Latin hypercube sampling (LHS) remains the most common one [21]–[23], [25]–[27]. Other methods include Quasi-Monte Carlo sampling [20] and full factorial design (FFD) [24].

In addition, a parameterization of the rotor blade is required prior to the optimization. Leusink *et al.* [22] and Sessarego *et al.* [25] took advantage of Bézier curves to define the twist and chord laws, while Bu *et al.* [27] used control points on the wing planform and then interpolated between these. Xu *et al.* [21], on the other hand, employed Class-Shape function Transformation (CST) to parameterize the chord and the twist of the blade. Olson [28] extended this method into a generalized three-dimensional CST parametrization scheme to be able to represent non-planar shapes like blended winglets and box wings.

4. Optimization workflow

4.1 Basic optimization concepts

Some elemental terms to the optimization process are hereby defined as per GEMSEO's [3] documentation.

- **Optimization problem**: mathematical problem consisting in finding a set of variables which minimizes or maximizes a mathematical function, while satisfying constraints on these variables or on arbitrary functions.
- **Objective function**: function to be minimized or maximized in the optimization problem.
- **Design variable**: unknown of the optimization problem; i.e., the input with respect to which the objective is minimized or maximized.
- **Constraint**: function of the design variables that must be kept either null or negative in an optimization problem.
- **Optimum**: solution of an optimization problem; i.e., the values of the design variables at the minimum of the function, subject to the constraints.
- **Design space**: mathematical set containing the design variables of the optimization problem. In GEMSEO, it is defined by the variables' names, sizes, types and bounds. It also includes the current value of these variables.
- **Discipline**: program, or arbitrary set of simulation software, that can be viewed as a mathematical multivalued function, taking inputs and producing outputs through its execution. It can be used to compute design variables, constraints or the objective function from other design variables.
- **Scenario**: interface available in GEMSEO that creates an optimization (or sampling) problem. It is based on a set of disciplines, a multidisciplinary formulation, a design space and an objective name. It can then execute said problem using a number of optimization (or sampling) algorithms and post-process it.

4.2 Previous workflow

The contents of this section are taken from the internship report of Piceni [2], serving as an introduction so that the reader can later understand the updates to the workflow implemented in the present work.

The aim of the optimization problem behind this workflow is to change the rotor geometry and rotation rate in order to minimize the objective function – a linear combination of the power consumed by the rotor and its OASPL – while maintaining a certain thrust. The rotor geometry is parameterized by determining the chord and twist at a fixed number of blade sections, as well as the sweep and the dihedral in the span regions between them.

Three main elements are required for this process:

- An **aerodynamic solver**: in this case, QuickCFD, a low-fidelity CFD solver based on the panel method, internally developed by Capgemini Engineering. It takes the rotor geometry and rotation rate together with the flow conditions and computes the thrust and the power, as well as the aerodynamic loads and the blade velocity field.
- An **acoustic solver**: PyFfonc, another internal tool from Capgemini Engineering. Based on Farassat's F1A formulation of the FW-H acoustic analogy, it takes the rotor geometry and rotation rate together with the loads and the velocity field outputted by the aerodynamic solver and computes the OASPL
- An **optimizer**: GEMSEO. It wraps the different solvers using disciplines and manages the passing of the design variables from one discipline to another as well as the optimization process itself.

This workflow is best illustrated in the diagram in Figure 4.1.

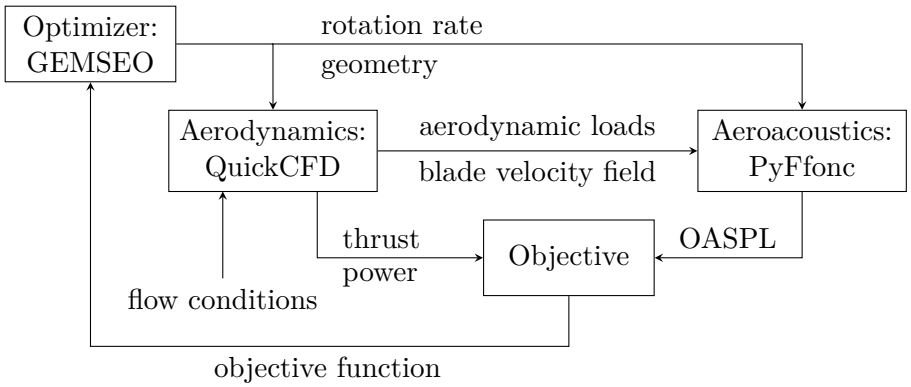

Figure 4.1: Diagram of the aeroacoustic optimization workflow using QuickCFD [2].

As previously stated, the objective function is computed as a weighted sum of the power and the OASPL:

$$\text{objective} = \text{weight} \times \frac{\text{OASPL}}{\text{OASPL}_0} + (1 - \text{weight}) \times \frac{\text{power}}{\text{power}_0}$$

Where power_0 and OASPL_0 are, respectively, the values of the power and the OASPL of the first iteration of the optimization; i.e., the original values, and they are used to normalize the power and OASPL.

The weight parameter can be varied to obtain a Pareto front, but in the case of Piceni's work, a value of 0.9 was been kept. This is, the objective was set to minimize mostly the sound signature of the rotor, while maintaining a reduced power.

4.3 Updated workflow

The migration from QuickCFD to DUST required some changes to the workflow, since the format of the outputs of DUST are different to those of QuickCFD, as they include aerodynamic loads but also pressure and velocity distributions along the blade, geometric parameters like the normal directions of the panels and their area, local angle of incidence, etc. PyFfonc had to be updated accordingly, to accommodate for these new inputs. The resulting new workflow is shown in Figure 4.2.

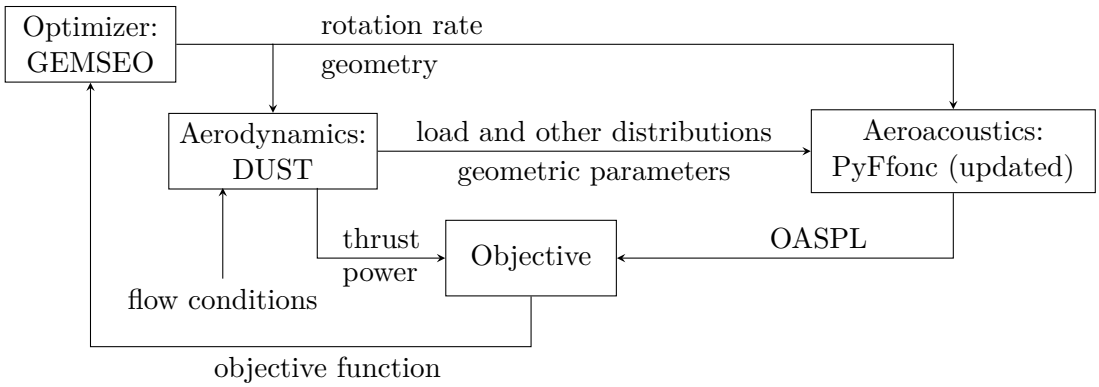

Figure 4.2: Diagram of the aeroacoustic optimization workflow using DUST.

Due to the switch to a new aerodynamic solver, there were features from the previous workflow that now did not work with DUST and had to be fixed. Some new functions were also incorporated.

4.3.1 New features

The new features added in this work include:

- **Creation of a folder for each optimization iteration**, in which all the files pertaining to that iteration are saved. These include DUST files, outputs from the aeroacoustic solver and post-processing results. After the optimal geometry has been found, the corresponding folder is renamed to append the «_OPT» suffix to it.

- **Interpolation of the blade geometry** using cubic splines. The reason behind this is the opposing needs of the optimizer, for which the lesser the design variables, the better in order to have a limited design space; and the aerodynamic discipline, that requires a fine discretization of the blade to achieve a proper panelization of the geometry. The interpolation allows for a finer aerodynamic and aeroacoustic discretization – which in turn permits to achieve more complex blade shapes – without increasing the number of design variables.

  To showcase the effects of the interpolation, a tapered, twisted Caradonna rotor [29] (geometry specified in Table 4.1) has been simulated in DUST with and without interpolation. For both cases, 5 design points (DP) – hub, tip, middle and quarters of the blade – were used, and the geometry was interpolated to 19 points in the corresponding case. The result of the interpolation can be seen in Figure 4.3. In addition, a comparison of the blade planform before and after interpolation is shown in Figure 4.4.

- **Enabling the selection of different airfoils** along the blade span. Once the interpolation is set in place, a dictionary can be used to specify which airfoil is used up to which blade span station. An example of such a dictionary is shown in the code of Listing 4.1. In this example, airfoil 1 is used up to 50% of the blade, airfoil 2 from 50% to 80%, and airfoil 3 from 80% to the end of the blade.

- Allowing to maintain the **original thrust** in the optimization process, instead of manually imposing the desired thrust constraint.

Table 4.1: Geometry and flow conditions of the modified Caradonna rotor [29] used to present the effects of the interpolation.

Parameter	**Value**
Number of blades	2
Radius	1.143 m
Hub radius	0.1905 m
Position of the DP	{0, 25, 50, 75, 100} % of the span
Chord at the DP	{0.2105, 0.2105, 0.1905, 0.1505, 0.1305} m
Twist at the DP	{10, 10, 7, 5, 2} deg
Airfoil	NACA 0012
Rotation rate	1250 rpm
Velocity upstream	0 m/s
Aerodynamic simulation time	20 rev

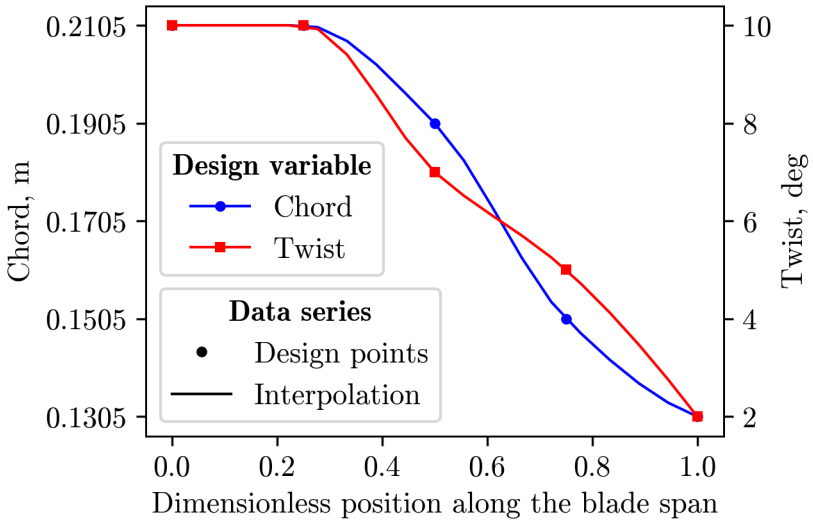

Figure 4.3: Interpolation of the chord and the twist at the 5 design points using splines with 19 points.

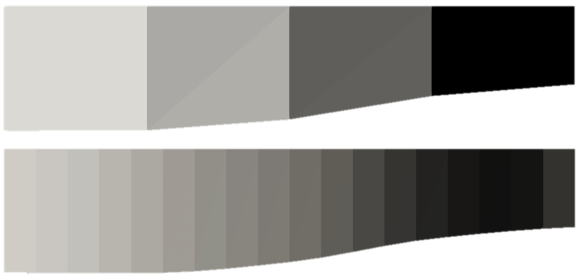

Figure 4.4: Blade planform without and with interpolation. This representation corresponds to the visualization in ParaView of one of DUST post-process files.

```
airfoil_db = {0.5: 'airfoil1.c81', 0.8: 'airfoil2.c81', 1:
    'airfoil3.c81'}
```

Listing 4.1: Example of dictionary used for airfoil specification along the blade span. Keys indicate the fraction of the blade span up to which the airfoil given by the value is used.

4.3.2 Resolution of issues

Two main situations were addressed:

- **Simulations in hover**. In the beginning, the lack of an axial velocity in pure hover cases provoked these simulations to crash due to the recirculation of the vortex particles. For this reason, a slight incoming velocity – of the order of tenths or hundredths of m/s – had to be imposed to convect these particles downstream and avoid this issue. After consulting one of DUST developers through its Git, he suggested that there was a bug related to the `particles_redistribution` feature. Disabling this option in the `dust.in` file allowed for pure hover cases to be run, which is important taking into consideration that all the rotors to be optimized in this work are supposed to be in hovering conditions.

- **Concatenation of the pressure signal**, which consists in replicating the signal of a single period. This allows to simulate only one rotor revolution in PyFfonc that can then be repeated multiple times, providing a longer signal that allows the extraction of a cleaner spectrum, since the higher number of calculated periods corresponds to a finer discretization; all of this with a lower computational cost. This simplification is only possible for situations in which the loading of the rotor blades is steady during their rotation, which is the case for hover in undisturbed flow.

  After the migration to DUST, the concatenation did not work since the pressure signal outputted by the new solver was not perfectly periodic: there were very small differences that made that one period did not start at the same point the previous period had finished. For this reason, the steady aerodynamic solution had to be ran in the acoustic solver for as many rotor revolutions as required – depending on the number of blade passing frequencies (BPFs) to be captured and the desired discretization – with the corresponding negative impact on computation time.

  The concatenation was finally achieved by means of BPF averaging. This procedure consists in averaging the different parts of the signal, each part corresponding to one of the blades. In this way, each period is divided in as many equal portions as the number of blades of the rotor; then, the mean of all these contributions is computed to find the average period. This ensures that the first time step of period i corresponds exactly to the step following the last step of period $i-1$. Once this condition is satisfied, the two periods can be concatenated. A pseudo-code of this process is shown in Listing 4.2.

  The concatenation of the pressure signal of a single rotor revolution 50 times permits a computation three times faster (see Table 4.2) in addition to the expected sharper frequency spectrum, as shown in Figure 4.5.

  Note that the rotor analyzed for this example is the Caradonna rotor [29], with a collective pitch of 10 deg. Its geometry and the characteristics of the simulation are indicated in Table 4.3.

```
pressure = ... # pressure signal during the simulation time
    corresponding to a number of revolutions

# BPF average
for i in range(n_blades):
    pressure_blade_i = ... # slice of the pressure signal
    corresponding to the i-th blade
    pressure_1blade += pressure_blade_i
pressure_1blade /= n_blades

# concatenate the average signal to obtain one full period
for i in range(n_blades):
    pressure_1rev = concatenate(pressure_1rev, pressure_1blade)

# replicate the pressure signal to increase the quality of the
    spectrum (possible only if the signal is exactly one period long)
for i in range(n_repetitions):
    pressure_new = concatenate(pressure_new, pressure_1rev)
```

Listing 4.2: Process of signal concatenation via BPF averaging.

Table 4.2: Elapsed time during the acoustic calculation and resulting OASPL with and without concatenation. With concatenation, only one revolution is computed and then repeated 50 times; without it, 10 revolutions are computed.

	Without concatenation	**With concatenation**
Computation time, s	44	13
OASPL, dB	91.35	91.35

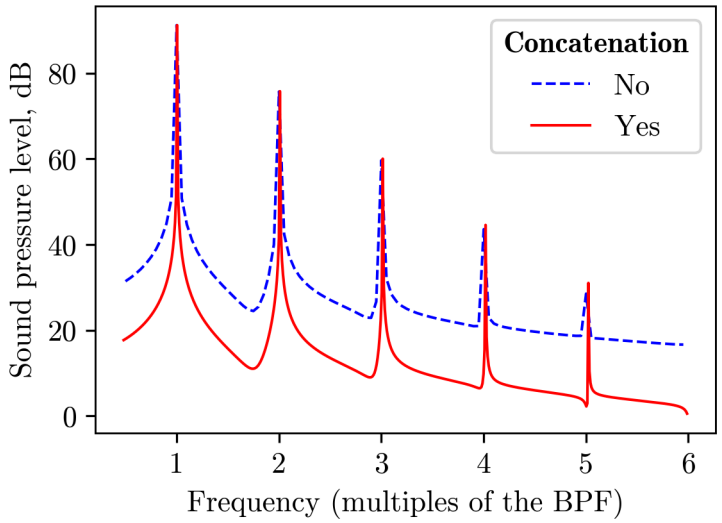

Figure 4.5: Sound pressure level on the frequency spectrum, with and without concatenation. With concatenation, only one revolution is computed and then repeated 50 times; without it, 10 revolutions are computed.

Table 4.3: Geometry and flow conditions of the Caradonna rotor [29] used to present the effects of the pressure signal concatenation.

Parameter	**Value**
Number of blades	2
Radius	1.143 m
Hub radius	0.1905 m
Chord	0.1905 m
Twist	10 deg
Airfoil	NACA 0012
Rotation rate	1250 rpm
Velocity upstream	0 m/s
Aerodynamic simulation time	20 rev

For the acoustic calculation, the observer was placed at 10 m of the center of the rotor at an elevation angle of 45 deg downstream of the rotor, where 0 deg corresponds to the rotor plane. This is represented in Figure 4.6.

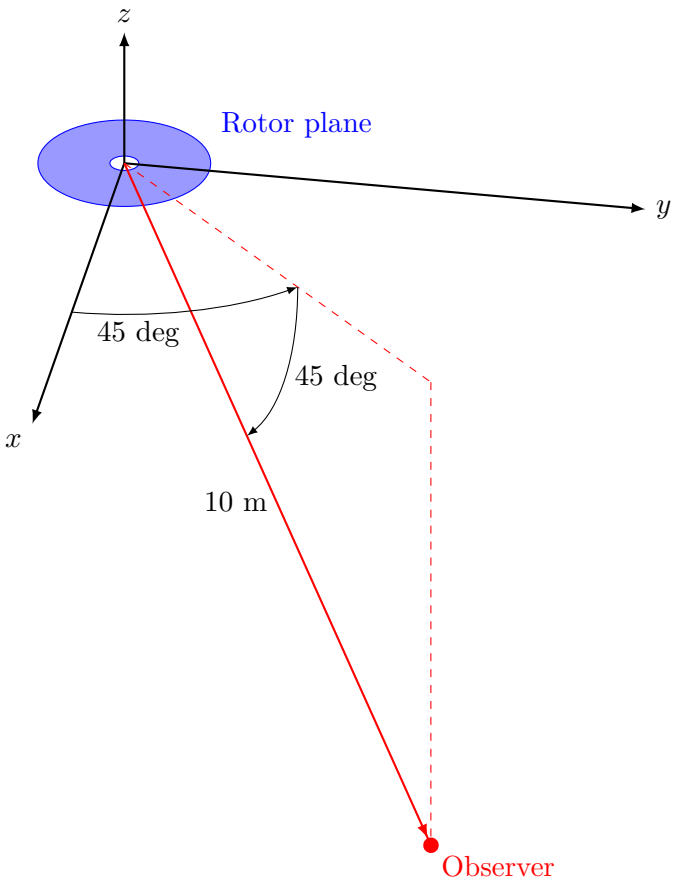

Figure 4.6: Position of the observer for the acoustic calculation, with respect to the rotor plane.

Some other minor fixes like changing some hard-coded parts of the code and updating GEMSEO to the latest version (5.3.1) – with the corresponding changes in syntax – were performed.

4.3.3 Some insight on how the updated workflow works

A configuration file is used to specify the parameters of the optimization. This approach was already employed in the previous workflow, but some of the inputs have changed. There are three main sections:

- One for the **rotor**, with parameters regarding the geometry – position of the DP, original chord, twist, sweep and dihedral, airfoil dictionary, number of blades and tip and hub radius – and the aerodynamics – original rotation rate, incoming flow speed, number of points for the interpolation, number of revolutions to be calculated, etc.
- Another one for the **acoustics**, where the user specifies the arrays of microphones as well as acoustic options such as whether to use the previously discussed signal concatenation.
- One last section for the **optimization**, in which the maximum number of iterations, the bounds for the design variables and the objective thrust (`'baseline'` if the original thrust is to be maintained) are set.

Regarding the aerodynamic discipline, some programs are used to generate the required DUST files (references frames and blade geometry as well as `dust_pre.in`, `dust.in` and `dust_post.in` files). The blade geometry is defined in a parametric manner, where chord, twist and airfoil are defined in multiple sections (corresponding to the number of points for the interpolation) and the blade span, sweep and dihedral are indicated in the regions between these sections. The type of aerodynamic element used are lifting lines because they provide a rapid way of approximating not only the potential but also the viscous effects of the blade at a low computational cost, by means of using a lookup table for the airfoil polar (as opposed to QuickCFD, where lifting lines could only be modeled with the potential theory, their force being proportional to the incidence).

In the `dust.in` file, the time step size is determined by dividing each rotor revolution in 72 steps. Model parameters are also chosen, together with wake parameters and options for the lifting line solver, among others.

In the `dust_post.in` file, different type of analyses are requested: visualization of the pressure and the singularity intensity (see for instance Figure 5.3), integral loads such as the total thrust and moment, sectional loads like the effective incidence or the velocity distribution on the blade (see Figure 5.4) and acoustic data used as inputs to the acoustic solver.

As for the acoustic discipline, only tonal noise is modeled in the current workflow, since broadband noise has yet not been implemented.

Finally, it is worth mentioning that this workflow does not require running the optimization process as a whole, but the user may run the different disciplines separately or substitute one of them for a different solver, an example of which is found in chapter 6, where the DUST discipline is replaced by a surrogate model of the aerodynamics of the rotor.

5. Result of the optimization

In this chapter, the setup and the result of the optimization performed using the workflow described in section 4.3 is presented. Although the workflow allows for the optimization of chord, twist, sweep, dihedral and rotation rate, only the first two were varied in this work. The rotation rate was specifically left out of the optimization process because it was already confirmed by Piceni [2] that the optimizer will try to reduce it in order to bring down the noise. Here it is desired that the optimizer modifies the geometry of the blades, which is more challenging and potentially more interesting for real-world applications, since the rotation rate may be imposed by external constraints such as the engine. As for the sweep and dihedral, they have been fixed at a nil value for the sake of a quicker optimization and a better understanding of the influence of the chord and twist on the output.

The parameters of the optimization are presented in Table 5.1. The original geometry that will later be changed is the Caradonna rotor [29] with a constant chord of 19.5 cm and a constant twist of 10 deg. The bounds have been established at 5 cm above and below the original chord and 10 deg above and below the original twist. The weights of the different parts of the objective function was taken from the original workflow: 10% corresponds to the power consumed by the rotor, 90% to its OASPL, in order to prioritize noise reduction over power economy. The optimization algorithm used is Constrained Optimization BY Linear Approximation (COBYLA) [30], which is an algorithm for derivative-free optimization with nonlinear inequality and equality constraints [31].

The evolution of the thrust, power, OASPL and objective function is shown in Figure 5.1. It can be stated that the solution does not change drastically from iteration 100 onward.

A comparison of these quantities together with the values of the design variables of the original and the optimal geometries are presented in Table 5.2. The blade geometries are compared in Figure 5.2 and the wakes produced can be seen side by side in Figure 5.3. Important distributions such as those of induced velocity, effective incidence and thrust are also compared in Figure 5.4.

The strategy followed by the optimizer consists in loading the blade close to the hub by increasing the twist. This allows for a minimization of the chord and the twist at the extremity of the blade, which in turn unloads the tip and therefore reduces both the induced losses and the sound generation since this region is where the local velocity is highest.

This can be confirmed by the span-wise distributions: the original blade is highly loaded towards its tip, which is to be expected for a rectangular, uniformly twisted rotor blade. The optimal blade has in turn an induced velocity distribution much more centered around zero[1] – which is known to reduce the induced losses – and it is heavily loaded near its center and very lightly loaded towards its tip.

This results in a reduction of 12.9% in the power consumed by the propeller and a decrease of 3.54 dB in its OASPL, all while maintaining the original thrust and therefore providing the same propulsive force. Moreover, the shortening of the chord translates to a smaller, lighter rotor, which will presumably be beneficial in most real-life applications.

Although the approach taken by the optimizer has proven its effectiveness, a problem it presents is that the chord at the extremity of the blade tends to be minimized, which

[1] The tip speed corresponding to a rotation rate of 1250 rpm and a radius of 1.143 m is 149.6 m/s (or Mach 0.44). Therefore, the induced velocity varies approximately from -4% to 2.7% of this reference speed.

Table 5.1: Parameters of the optimization performed with the new workflow. Geometric, aerodynamic, acoustic and optimizer parameters are specified.

Parameter	**Value**
Number of blades	2
Radius	1.143 m
Hub radius	0.1905 m
Position of the DP	{0, 25, 50, 75, 100} % of the span
Original chord	0.1905 m
Original twist	10 deg
Airfoil	NACA 0012
Interpolation	Yes
Rotation rate	1250 rpm
Velocity upstream	0 m/s
Aerodynamic simulation time	20 rev
Microphone (distance, elevation)	(10, −45) (m, deg)
Acoustic simulation time	1 rev
Signal concatenation	Yes
Optimization algorithm	COBYLA
Max. number of iterations	400 iterations
Objective thrust	Original value
Chord bounds	$[0.1405, 0.2405]$ m
Twist bounds	$[0, 20]$ deg
Weight of the acoustics in the objective	0.9

can lead to non-physical blade shapes if this is not taken into consideration. To prevent this, a reasonable lower bound for the chord must be chosen, taking into account structural limitations, aspect ratio, etc.

Furthermore, one must also be thoughtful when selecting the upper bound for the twist, since the optimizer tries to reach a high effective incidence closer to the hub, which can lead to non-linear aerodynamic phenomena that cannot be predicted by DUST, such as the blade stalling. In this case, the maximum effective angle of attack is about 9 deg, which is already moderately high, and had the twist not been limited to 20 deg, it would have arguably kept increasing according to Figure 5.2.

Table 5.2: Comparison of the values of the design variables and objectives for the original and the optimal geometry. Note that in the first case the chord and twist are indicated as a single value since they are constant throughout the DP.

Parameter	**Original**	**Optimal**	**Difference**
Iteration	1	221	–
Chord at the DP	0.1905 m	{0.1659, 0.1502, 0.1406, 0.1405, 0.1427} m	–
Twist at the DP	10 deg	{13.89, 19.98, 19.98, 6.937, 0.1622} deg	–
Thrust	737.7 N	737.8 N	–
Power	12.25 kW	10.67 kW	−12.9%
OASPL	91.35 dB	87.81 dB	−3.54 dB
Objective	1	0.9523	−4.77%

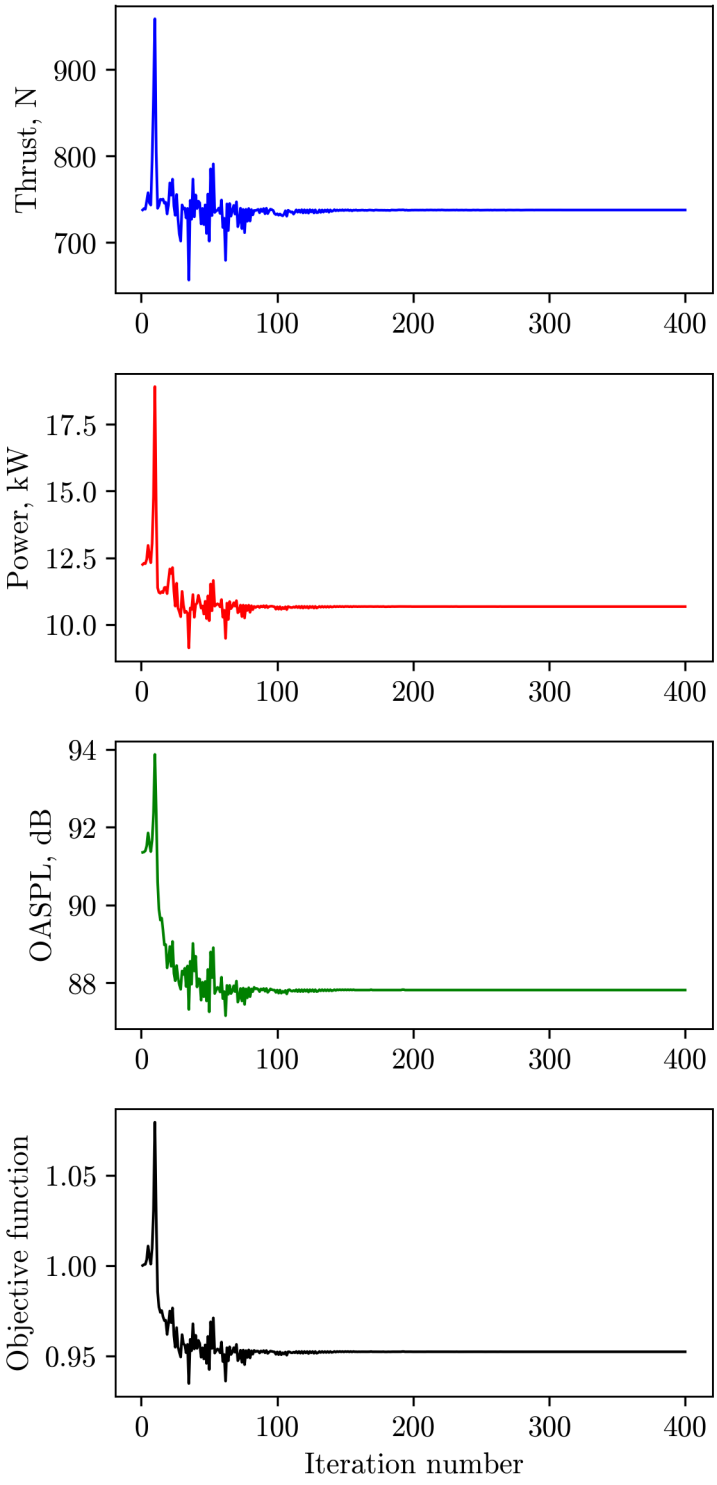

Figure 5.1: Evolution of the thrust, power, OASPL and objective function throughout the optimization process.

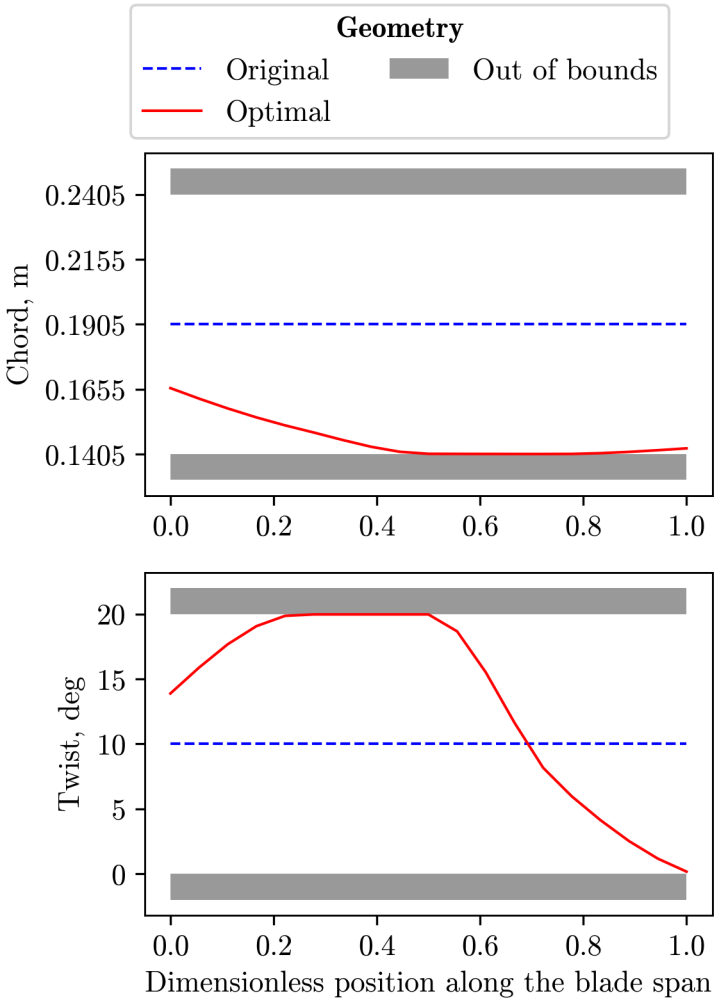

Figure 5.2: Comparison of the original and optimal twist and chord.

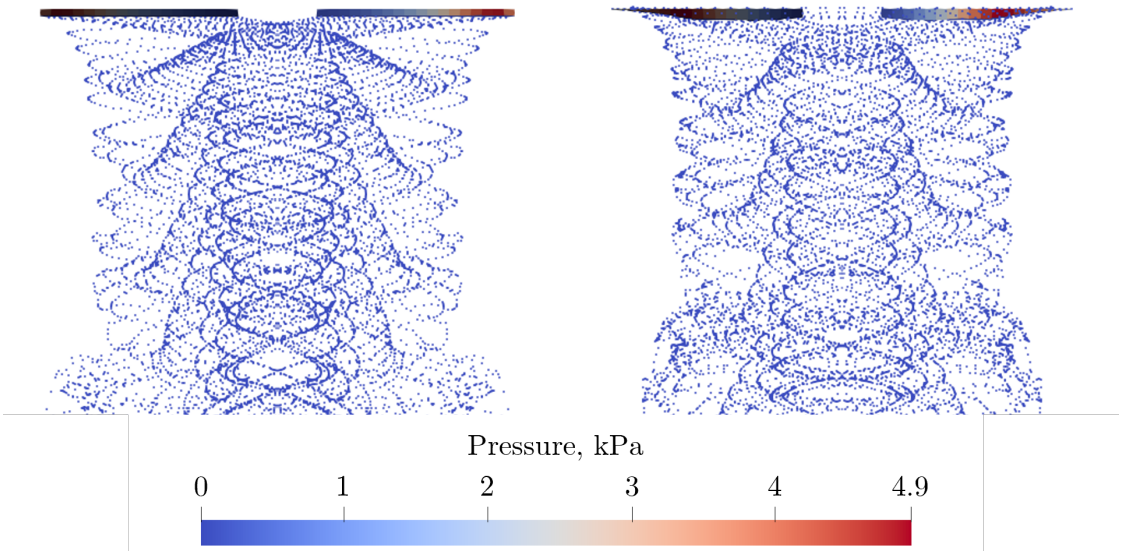

Figure 5.3: Comparison of the wakes generated by the original (left) and the optimal (right) rotor. This representation corresponds to the visualization in ParaView of the pressure distribution on the vortex particles outputted by DUST.

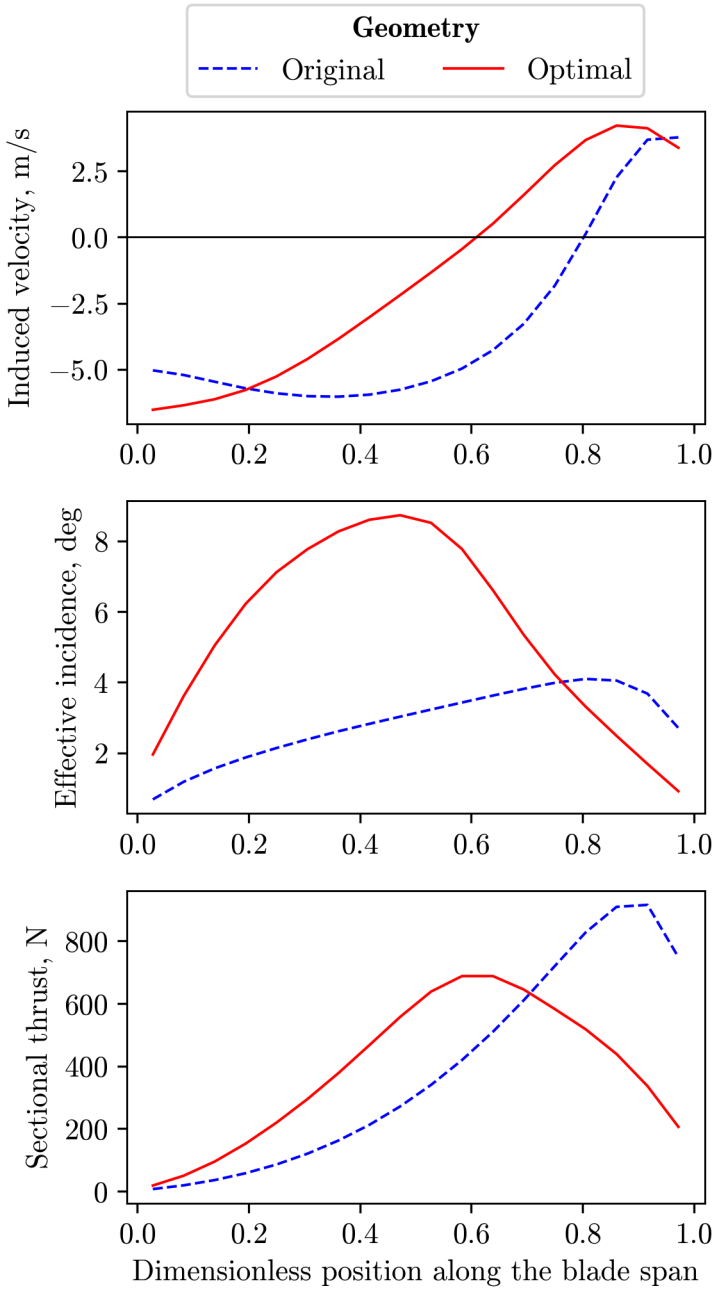

Figure 5.4: Comparison of the original and optimal induced velocity, effective incidence and thrust distributions.

6. Optimization using surrogate models for the aerodynamics of the rotor

In this chapter, a surrogate model is used to substitute the aerodynamic solver. In order to do this, a sampling plan is established to generate the samples and a surrogate algorithm is chosen. Then, these samples are used to train the model and finally the workflow is modified to accommodate for this new way of calculating the aerodynamics.

The reason for implementing a surrogate optimization is that its lower computational cost allows to perform more optimizations, which in this case has been used to draw the Pareto front of optimal solutions by varying the weight of the acoustics in the objective function. Furthermore, it opens the door for different optimization strategies, such as using gradient-based optimization algorithms, which would provide a more efficient descent towards the optimum; or taking advantage of a multi-start approach, that consists in starting the optimization in different places of the search space, with the aim of avoiding local optima.

6.1 Sampling and design of experiments

Before building the surrogate models, both a sampling technique and a method for the parameterization of the rotor blade must be chosen. In Table 6.1, a summary of the advantages and the disadvantages of some of the sampling techniques available in GEMSEO and used in the literature is presented.

Table 6.1: Advantages and disadvantages of a number of sampling techniques available in GEMSEO. The last column indicates the references in which said techniques are used.

Sampling	**Advantages**	**Disadvantages**	**Ref.**
Random or Monte Carlo sampling	Simple to implement. Unbiased representation of the design space	May miss important regions. Can require a large number of samples [32]	–
Quasi-Monte Carlo sampling	Better coverage of the design space compared to Monte Carlo [33]	More complex than Monte Carlo to implement. Requires a large number of samples [33]	[20]
LHS	Ensures all portions of the input space are sampled [34]	Complex to ensure uniformity in high-dimensional spaces	[21]–[23] [25]–[27]
FFD	Explores all combinations of inputs [32]	Computationally expensive. May not be feasible for large design spaces [32]	[24]

Finally, the LHS method has been selected due to its easy implementation through GEMSEO and its exhaustive exploration of the design space.

Regarding the parametrization method, the choice has been to maintain the previous

process of imposing the design variables at given DP, then interpolate this geometry on a greater number of points. To be able to reproduce the previous optimization using DUST, the chord and twist of the blade are specified by the optimizer at 5 DP, then they are interpolated to 19 points equally spaced along the blade span. For the same reasons explained in chapter 5, nor the rotation rate nor the dihedral or the sweep are varied.

Using these, a design of experiments (DOE) can be created and launched to obtain the required number of samples to train the surrogate models. 500 samples have been generated, 400 of which will be used for training the models and 100 for testing them, i.e., evaluating the error obtained in samples unknown to the model. In Table 6.2, the main parameters of this DOE are specified. Note that the bounds set for the design variables of the DOE do not only contain those for the optimization, but they are wider by a margin of 2 cm for the chord and 5 deg for the twist. Otherwise, the interpolation would have a smoothing effect that would prevent the extremes to be reached, and even if they were, the prediction would not be accurate due to the low number of samples in this region.

Table 6.2: Parameters of the DOE to generate the samples for the surrogate models. Geometric, aerodynamic and sampling parameters are specified.

Parameter	**Value**
Number of blades	2
Radius	1.143 m
Hub radius	0.1905 m
Position of the DP	{0, 25, 50, 75, 100} % of the span
Airfoil	NACA 0012
Interpolation	Yes
Rotation rate	1250 rpm
Velocity upstream	0 m/s
Aerodynamic simulation time	20 rev
Number of samples	500 samples
Sampling algorithm	LHS
Chord bounds	$[0.1205, 0.2605]$ m
Twist bounds	$[-5, 25]$ deg

Since the objective of the surrogate models is to predict the aerodynamics, the only outputs considered are the thrust, the power and the pressure distribution on the blade.

It goes without saying that this approach would greatly benefit from parallelization. In this case, the different samples of the DOE were run sequentially; however, making use of parallel computation could have significantly accelerated the generation of the DOE.

6.2 Surrogate algorithm and training of the models

In Table 6.3, a summary of the advantages and the disadvantages of some of the surrogate algorithms used in the literature is presented.

The algorithm chosen to create the surrogate models is the Gaussian process regressor because of its performance and its simple implementation in GEMSEO, given that parameter tuning can be relatively automatized using the class `MLAlgoSelection` from the machine learning module.

Table 6.3: Advantages and disadvantages of a number of surrogate algorithms used in the literature. The last column indicates the references in which said techniques are used.

Algorithm	**Advantages**	**Disadvantages**	**Ref.**
Response surface modeling	Simple to implement. Provides an explicit mathematical form [35]	May not capture complex, nonlinear relationships [36]	–
RBF	Flexible in approximating functions. Can handle complex, nonlinear functions [37]	Choice of basis function affects performance. High computational cost for large datasets [37]	[24], [25]
GPR or Kriging	Provides best linear unbiased prediction. Incorporates spatial correlation [38]	High computational cost for large datasets. Requires extensive parameter tuning [39]	[22]
HK	More accurate and cost effective than traditional Kriging [20]	More complex to implement	[20]–[23] [27]
Neural networks	Highly flexible and can model complex, non-linear relationships [40]	Requires large datasets for training. Expensive and time consuming to train [40]	–

The inputs to the models are the chord and twist at the DP of the blade, which are scaled using a standard scaler; i.e., they have zero mean and unit standard deviation, which is one of the hypotheses behind the regression model and, in general, is considered to be good practice. As for the outputs, a different model has been used to predict each quantity: one for the prediction of the thrust produced by the rotor, another one for the prediction of the power consumed by it, and a last one for the prediction of the pressure distribution on the blade.

The latter poses the problem of having to predict a distribution rather than a single value. One must keep in mind that with the interpolation enabled, 19 blade sections are used to simulate the aerodynamics and therefore, the pressure distribution is computed on 18 regions of the blade. The larger the dimension of the output, the more computationally expensive becomes the training of the model. For this reason, it was decided to use only half of these 18 pressure values, following a sine-spaced distribution so that the discretization is finer towards the blade tip.

In order to train the model, 80% of the 500 generated samples are used, while the rest serve to compute the mean square error (MSE); i.e., the average value of the square of the difference between the solution predicted by the model on the test samples and the observed value. After testing several combinations of alpha-parameter (nugget effect for regularization), number of restarts of the optimizer and kernels, the best predictions and the lowest MSE were found using no regularization, 10 restarts of the optimizer and an anisotropic Matern kernel with a nu-parameter of 0.5 – equivalent to an absolute exponential kernel [41] – and a length scale with size equal to the number of inputs: in this case 10, as there are 5 values for the chord and 5 for the twist.

This choice has been made using the `select()` method of the class `MLAlgoSelection`.

This method also optimizes the parameters in the calibration space [3] and returns the MSE. In table Table 6.4, the optimal length obtained by this method is presented, together with the root mean square error (RMSE) computed when compared with the test data and the time it took to train the model with the learning data.

Table 6.4: Optimal length scale, RMSE and training time of each surrogate algorithm.

Parameter	**Optimal length scale**	**RMSE**	**Training time**
Thrust	{36.2, 138, 16.3, 11.9, 24.7, 17.9, 8.48, 4.16, 2.77, 4.58}	13.7 N	4.86 s
Power	$\{44.9, 2.96 \cdot 10^4, 15.5, 8.45, 14.8, 63.9, 10.5, 4.64, 2.36, 4.74\}$	0.337 kW	6.33 s
Pressure distribution	{31.3, 33.6, 15.4, 9.49, 7.52, 18.6, 10.4, 4.91, 2.56, 3.53}	69.8 Pa	14.1 s

In order to check the quality of the trained algorithms, the error on the 100 test samples is calculated and normalized with the mean value of each predicted quantity (for the pressure distribution, the mean of the average pressure). The distribution of this error is represented in Figure 6.1. The RMSE is also shown as a percentage of the mean. It can be observed that in most samples – 80% approximately – the error is below 2.5%, and that only in around 5% of the cases the error is higher than 5%.

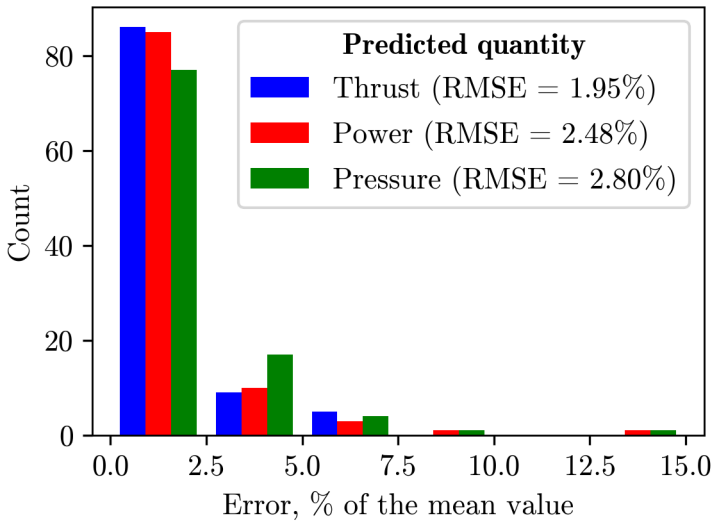

Figure 6.1: Histogram of the error of the surrogate algorithms on the test data.

In the context of this work, it can be concluded that this error is satisfying and that the surrogate models perform correctly if the point to be predicted falls inside the bounds of the DOE. If a higher accuracy was required, the error could be further reduced in theory by using more training samples.

6.3 Setup and result of the optimization

The workflow using the surrogate models to predict the aerodynamics can be better understood thanks to the diagram in Figure 6.2.

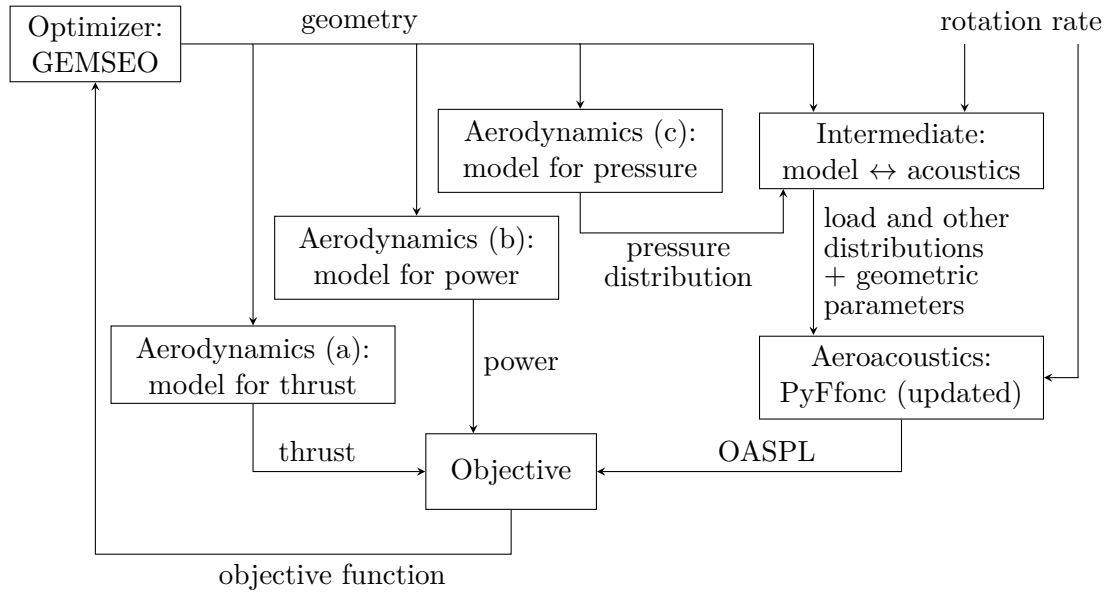

Figure 6.2: Diagram of the aeroacoustic optimization workflow using surrogate models for the aerodynamics.

Since now PyFfonc does not have the data from DUST as an input, an intermediate discipline is required to transform the outputs of the surrogate models in order to match the format of the inputs of the acoustic solver. This intermediate discipline uses the geometry to build the arrays of cell areas, normal vectors, velocities, etc. from scratch. It also takes the 9-point, sine-spaced pressure distribution given by the corresponding surrogate model and interpolates it again into the 18 regions of the blade.

Once the workflow is set up, the optimization can be performed. The parameters of the optimization are the same as those used for the optimization with DUST (see Table 5.1). The evolution of the OASPL and the objective function are shown in Figure 6.3. In this case, the solution does not vary significantly after the 150th iteration.

The optimal geometry found by the optimizer has been simulated in DUST and then PyFfonc in order to validate the results of the surrogate optimization. The comparison is shown in Table 6.5.

The optimal geometry and objectives are also compared to the original ones and to those found by the optimization workflow using DUST (from now onward, also referred to as traditional optimization). This comparison is presented in Table 6.6. Finally, the blade geometries are compared in Figure 6.4 and the distributions of induced velocity, effective incidence and sectional thrust along the blade in Figure 6.5. The results presented for the surrogate optimization are those corresponding to the validation with DUST, since they allow to obtain the aforementioned distributions.

It is observed that when validating the solution of the surrogate optimization in DUST, there is a minor deviation of the results of the surrogate models with respect to the higher-fidelity simulation, of the order of the error shown in Figure 6.1. However, as presented in

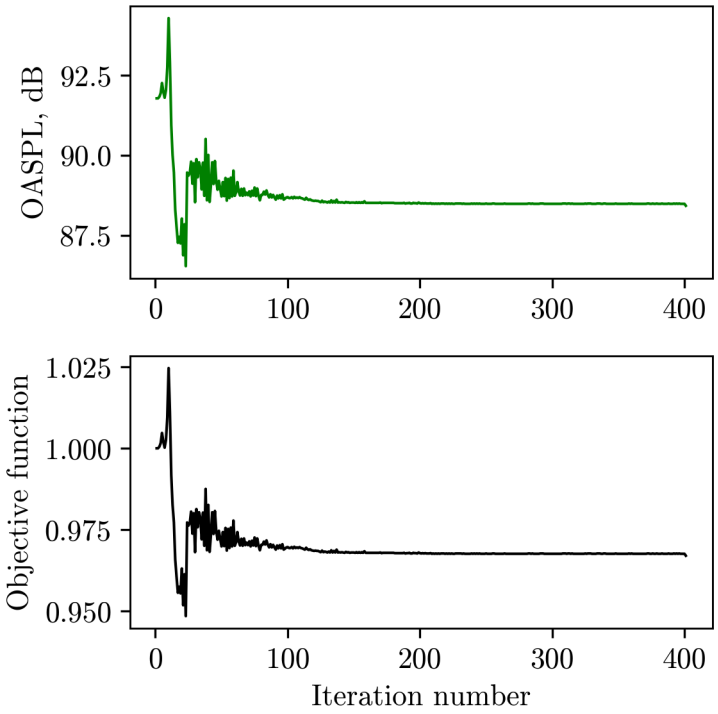

Figure 6.3: Evolution of the OASPL and the objective function throughout the surrogate optimization process.

Table 6.5: Validation of the optimal thrust, power and OASPL found by the surrogate optimization process. The results are validated by simulating the optimal geometry in DUST and PyFfonc.

Parameter	**Surrogate optimization**	**Validation with DUST**	**Difference**
Thrust	738.5 N	746.7 N	−1.10%
Power	10.86 kW	10.99 kW	−1.20%
OASPL	88.48 dB	88.42 dB	0.06 dB

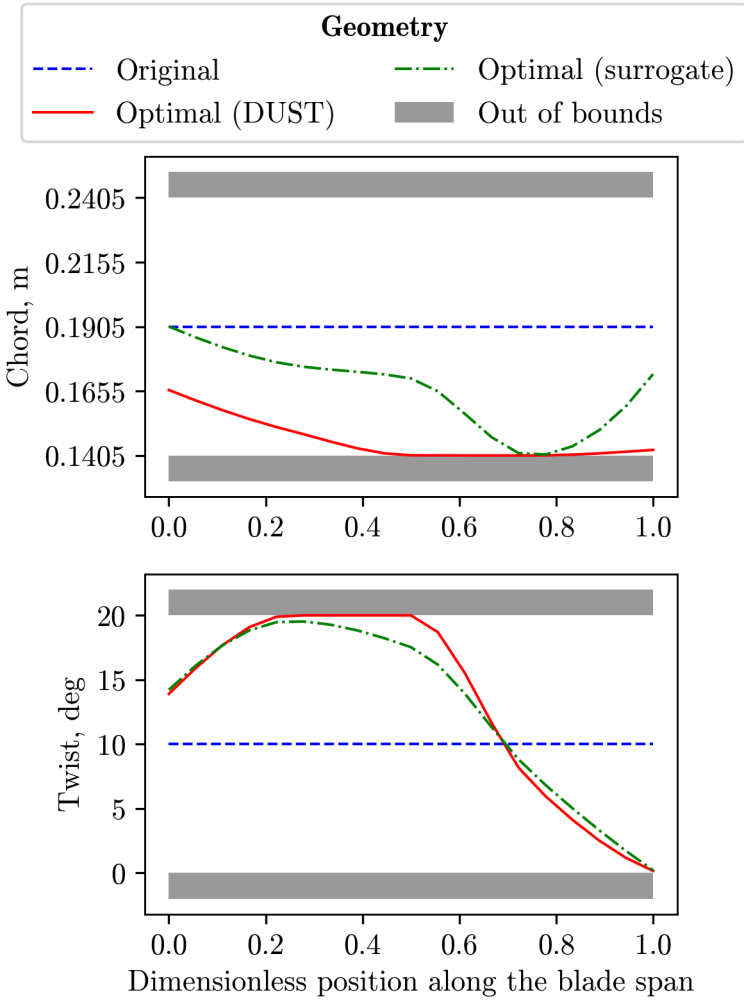

Figure 6.4: Comparison of the original twist and chord and the optimal ones found by the optimization with DUST and the surrogate optimization.

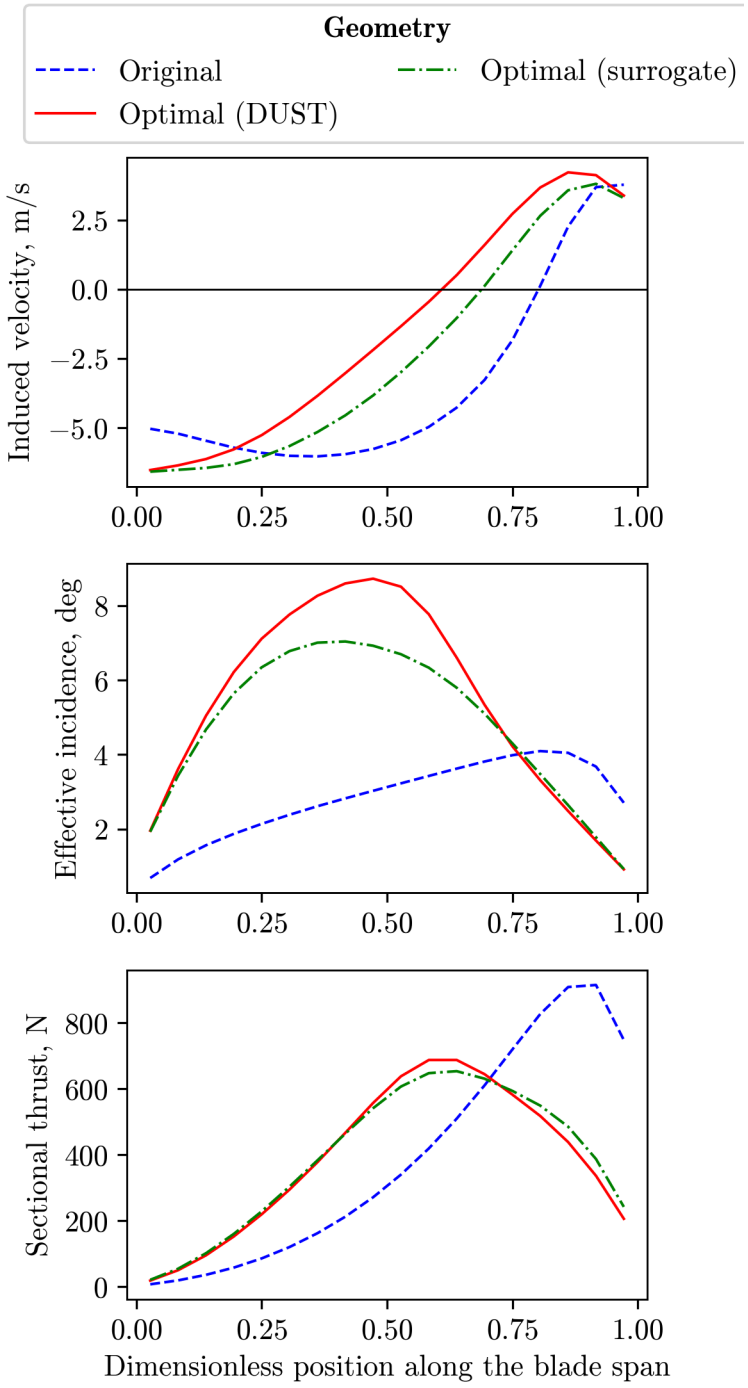

Figure 6.5: Comparison of the original induced velocity, effective incidence and thrust distributions and the optimal ones found by the optimization with DUST and the surrogate optimization.

Table 6.6: Comparison of the values of the design variables and objectives for the original and the optimal geometry found by the surrogate optimization (validated with DUST) and by the optimization with DUST. The differences in thrust, power and OASPL with respect to the original ones are indicated in parentheses.

Parameter	**Original**	**Optimal (DUST)**	**Optimal (surrogate)**
Iteration	1	221	338
Chord at the DP	0.1905 m	{0.1659, 0.1502, 0.1406, 0.1405, 0.1427} m	{0.1905, 0.1757, 0.1704, 0.1405, 0.1721} m
Twist at the DP	10 deg	{13.89, 19.98, 19.98, 6.937, 0.1622} deg	{14.21, 19.54, 17.51, 7.764, 0.2059} deg
Thrust	737.7 N	737.8 N	746.7 N
Power	12.25 kW	10.67 kW (−12.9%)	10.99 kW (−10.0%)
OASPL	91.35 dB	87.81 dB (−3.54 dB)	88.42 dB (−2.93 dB)
Objective	1	0.9523 (−4.77%)	0.9608 (−3.92%)
Elapsed time	–	7 days 20 h on the company's HPC[2]	39 min on a laptop

[2]This time can largely vary depending on the how busy the high-performance computer (HPC) is. In this particular instance, approximately 2 iterations were performed each hour; however, computation speeds of 10 iterations per hour or more have been observed. Therefore, this time could be reduced down to 1 day 14 hours in the best-case scenario.

Table 6.6 and Figure 6.4, the actual optimal geometries obtained with the different optimization workflows are not the same.

In fact, the traditional optimization achieves a higher reduction in both power and OASPL. This can be attributed to the, although small, not negligible error introduced by the surrogate models: since the surrogate model underestimates the thrust and the power but overestimates the OASPL, the optimizer reduces the twist in its search of noise minimization, having to compensate with a larger blade in order to attain the desired thrust. The effect of the aforementioned decrease in twist can be observed in the effective incidence distribution of Figure 6.5. Overall, it can be seen that the distribution are not as advantageous as in the case of the optimization with DUST.

Being aware of this deviation, one could characterize its magnitude and correct it in subsequent surrogate optimizations, so that results are more similar to the ones achieved with the traditional workflow.

Furthermore, despite being different to the solution found with DUST, it still performs significantly better than the initial blade, and it has been obtained in a fraction of the time and at a much more affordable computational cost compared to the traditional optimization, which were the main objectives of performing the surrogate optimization.

6.4 Generation of a Pareto front

One of the main advantages of implementing the surrogate optimization workflow is that several optimizations with different configurations can be launched provided the parameters that vary are considered as inputs in the DOE. This includes experimenting with different initial geometries, varying the objective thrust or changing the bounds of the optimization

as long as they are contained within the bounds of the DOE.

Since the optimization is multi-objective, this also permits the generation a Pareto front by varying the weight of the acoustics in the objective function; something whose cost would be prohibitive using the optimization workflow with DUST. This is precisely what has been done, choosing a list of weights that range from 0, minimizing only the power at the expense of the noise, to 1, which is the opposite case. The resulting plot is represented in Figure 6.6, where the points are labeled with the weight for the acoustics.

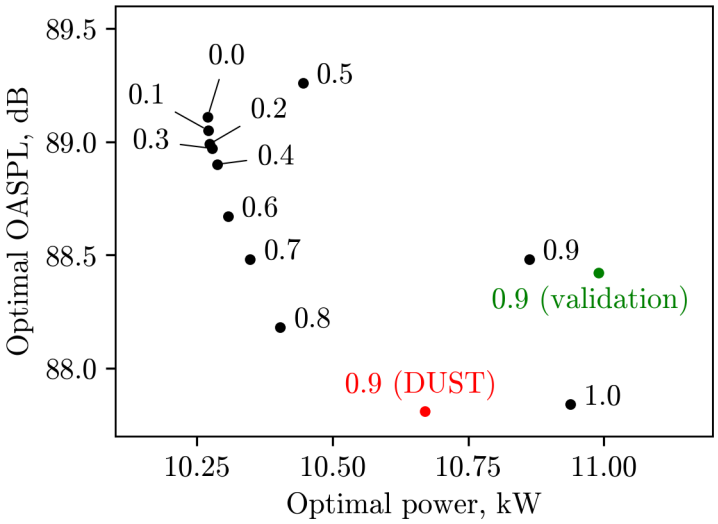

Figure 6.6: Pareto front in the power-OASPL space, obtained by varying the weight of the acoustics (the number next to the points) in the objective. The results of the surrogate optimization validated in DUST and the traditional optimization are included.

Generally, the trend is that of a convex Pareto front, where the solutions of the surrogate optimization (in black) decrease in OASPL and increase in power as the weight of the acoustics in the optimization grows. However, there are some outliers in the front of dominating solutions, in particular those for weights 0.5 and 0.9. This may be due to the optimization not performing with the same accuracy in all cases, although it should be further investigated. When performing the optimization with weight 0.9 with DUST (in red), the point lies much closer to the Pareto front.

7. Aeroacoustic simulation of ducted rotors

As previously mentioned in section 3.3, the addition of a duct may improve the performance of a propeller either in terms of power consumption or noise emission.

In this chapter, the simulation of a shrouded rotor in DUST is presented, and its aerodynamics are compared to those of an open propeller. Afterwards, Saliba P. R.'s code on the noise scattering due to the duct has been integrated in order to analyze the effect of the duct on the acoustics.

7.1 Aerodynamics of a ducted rotor

Up to this point in the internship, only open rotors could be simulated in DUST using the present workflow. In order to simulate the shrouded rotor, a method of specifying the duct parameters and afterwards generating the duct geometry needs to be devised. In the end, it has been chosen to create the duct geometry file in a similar fashion as the rotor blades themselves, but distributing the wing regions in a circumference about the rotation axis. Surface panels are used to model the duct elements, since they allow to recreate the three-dimensional effects brought by such a low-aspect ratio lifting surface.

A section for the duct is included in the configuration file, where the user must specify:

- The **radius** of the duct.
- Its **length**.
- The **airfoil** to be used as the duct section. Its chord line is placed at a distance equal to the duct radius from the rotation axis, and its chord corresponds to the duct length.

The user may also specify:

- The **offset** of the duct lip with respect to the rotor plane. The default value is 0, and negative values indicate that the lip is upstream of the rotor.
- Whether to **invert** the airfoil or not; i.e., if the inner wall of the duct is shaped by the airfoil suction side or its pressure side. The default option is to not invert the airfoil.
- The number of wing regions in the tangential direction. The default value is 36.
- The number of wing regions in the longitudinal direction, refined in a cosine distribution at both edges. The default value is 10.

For the use case, the modified Caradonna rotor used as the default geometry to start the previous optimization processes is simulated. This rotor is surrounded by a duct designed by Yilmaz *et al.* [42], and later used by Piceni and Saliba P. R. in their work. The characteristics of the duct are shown in Table 7.1; first, the dimensionless parameters defined by Yilmaz *et al.*; then, the corresponding items to be specified in the configuration file. A schematic view of the duct and the propeller is also depicted in Figure 7.1.

In Table 7.2, the aerodynamic performances of the open and the ducted rotor are compared. In Figure 7.2, the induced velocity, effective angle of attack and thrust distributions of both rotors are plotted.

Table 7.1: Characteristics of the duct for the simulation of the shrouded propeller in DUST.

Parameter	Value
Airfoil	NACA 4312
Clearance[3]	2.4%
Length-to-diameter ratio[4]	0.5
Forebody-to-length ratio[5]	0.3
Radius	1.2289 m
Length	0.58522 m
Offset	−0.17556 m
Inverted	Yes
Tangential discretization	72 regions
Longitudinal discretization	20 regions

[3]Ratio of the gap between the inner surface of the duct at the rotor plane and the blade tip to the rotor radius.
[4]Ratio of the length to the diameter of the inner wall of the duct at the rotor plane.
[5]Ratio of the distance between the lip and the rotor plane (what was previously defined as the offset) to the duct length.

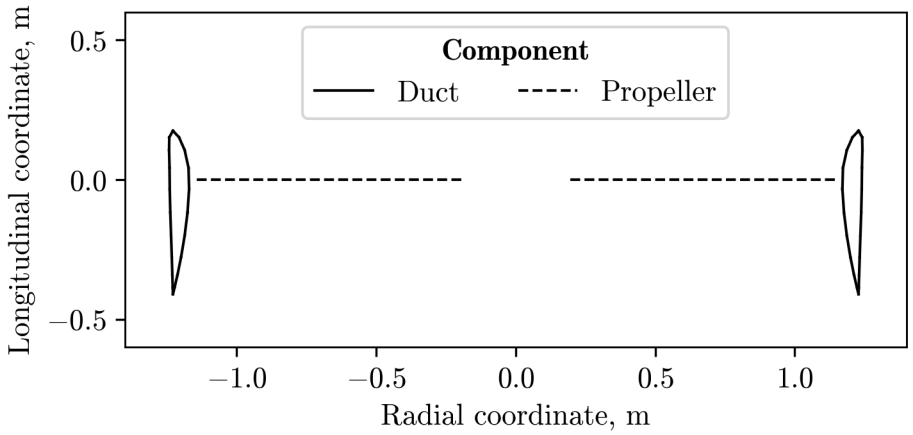

Figure 7.1: Schematic sectional view of the shrouded rotor.

Table 7.2: Thrust and power comparison of the open and the shrouded rotor.

Parameter	Open rotor	Ducted rotor			Difference
		Rotor	Duct	Total	
Thrust	737.7 N	670.3 N	145.9 N	816.1 N	+10.6%
Power	12.25 kW	–	–	12.03 kW	−1.80%

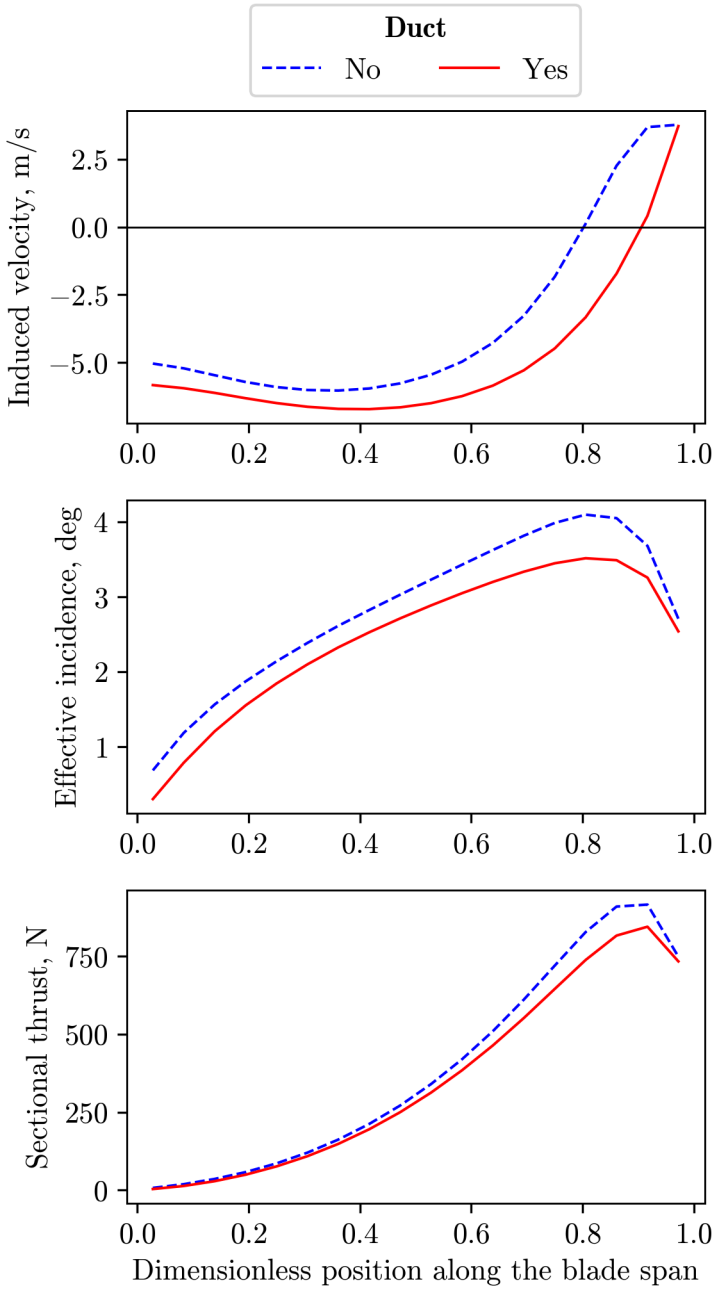

Figure 7.2: Comparison of the induced velocity, effective incidence and thrust distributions of the open and the ducted rotor.

The shroud induces an additional velocity on the rotor, which reduces the effective incidence of the blades and, as a result, both the thrust and power are decreased by 9.1% and 1.8%, respectively. However, the pressure distribution on the duct produces an additional force that not only compensates for this reduction, but increases the total thrust by more than 10%. This is in accordance with the solution of Froude's theory for shrouded propellers and the findings of Saliba P. R. regarding the aerodynamic performance of the ducted rotor.

Nevertheless, there are some limitations in this aerodynamic simulation to be taken into account. Since the type of elements used to model the duct in DUST are surface panels, only potential effects are considered and therefore the contribution of the duct to the thrust is probably overestimated. This can be mitigated by activating the `airfoil_table_correction` option, which applies a viscous correction to the resultant force given by the airfoil polar.

Additionally, the weight of the duct is not accounted for, which could result in structural and stability issues of the vehicle.

7.2 Acoustics of a ducted rotor

As for the acoustics, the code developed by Saliba P. R. [7] was modified in order to integrate it with the present workflow. Two use cases were available: on the one hand, a cylindrical duct modeled using the BIEM and using as a source the pressure distribution on the blade given by QuickCFD, the previous aerodynamic solver. On the other hand, an axisymmetric duct whose section is shaped by an airfoil, where the sources are modeled using dipoles placed at an arbitrary position, and the acoustics are solved using the BEM.

For this work, both approaches are combined in order to perform the acoustic simulation of the shrouded rotor described in Table 7.1. The noise scattering due to the duct is computed by means of the BEM and the incident noise is the result of the pressure distribution on the blade outputted by the aerodynamic simulation in DUST.

As a summary, the code works in the following manner:

1. Run the aerodynamic simulation to obtain the pressure distribution on the rotor blade.

2. Compute the derivative of the pressure on the duct surface, by placing microphones at both sides of the duct walls, then run the BEM solver. In this case, the extrados and the intrados of the duct section are discretized in 20 elements each, and the microphones are situated on the normal to these elements, one on the inside and the other on the outside.

3. Generate the array of microphones at the positions in which the pressure is to be calculated.

4. Compute the incident pressure (due to the rotor) and the scattered pressure (due to the duct) at the previous microphones.

5. Sum the incident and the scattered pressures to obtain the resultant pressure.

The reader must keep in mind that since the duct is axisymmetric, all calculations are performed only in one half of the space.

In a first instance, an array of 41×31 microphones are used to discretize the space within 3 m in the radial direction and 2 m in the longitudinal direction to the center of the rotor. The acoustic pressure field as well as the sound pressure level field are shown in Figure 7.3.

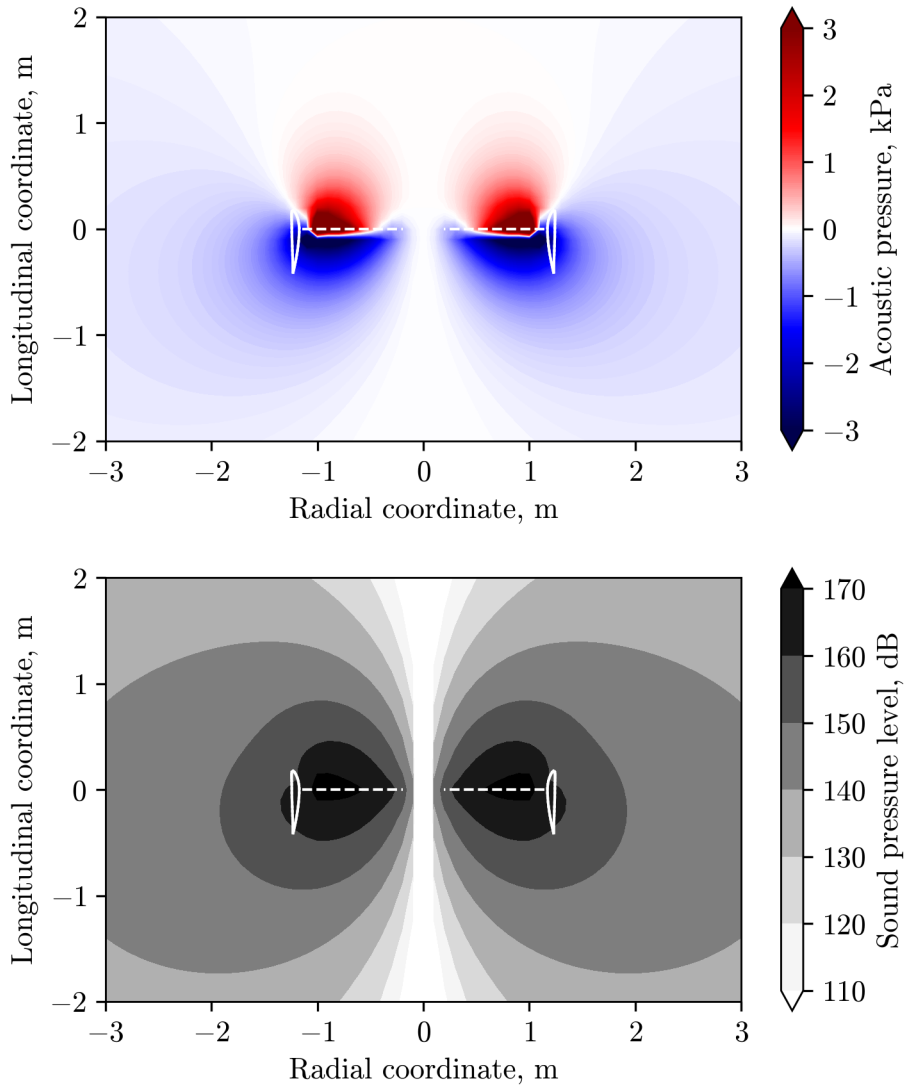

Figure 7.3: Fields of acoustic pressure and sound pressure level in the vicinity of the ducted rotor.

It is shown that the noise is slightly projected downstream, which is to be expected in a hovering rotor.

Secondly, the directivity is plotted in Figure 7.4 by placing a semi-circular array of 60 microphones placed at 10 m of the center of the rotor. The sound pressure level of the open and the ducted rotor are compared. The shielding of the noise by the duct is displayed, but its effect is not very significant. This is again verified by the work of Saliba P. R. [7].

If ducted rotors were to be incorporated to the optimization workflow, it is to be considered that the incident and the scattered pressure would be computed only in one microphone, as it is done in the traditional optimization workflow.

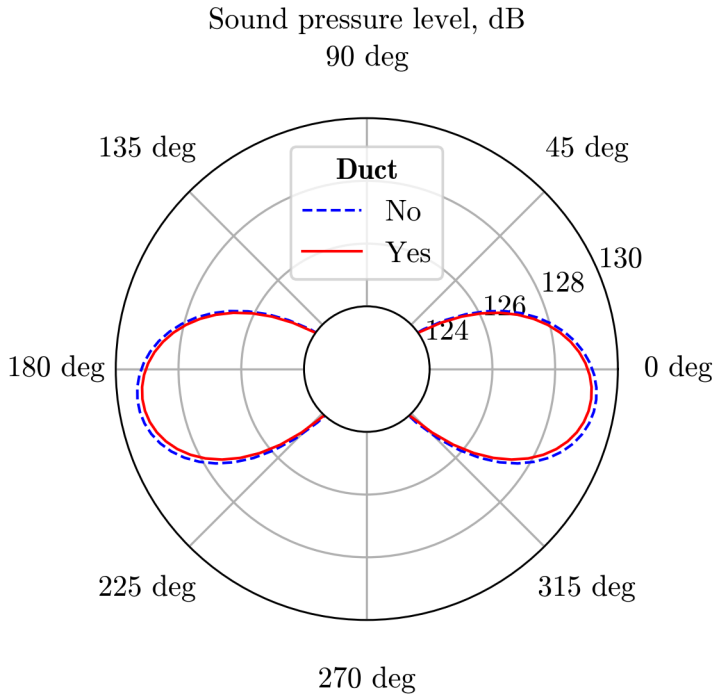

Figure 7.4: Comparison of the directivity of the open and the ducted rotor at 10 m of the center of the rotor.

8. Conclusions

8.1 Results and objectives

Regarding the optimization workflow itself, a successful migration from QuickCFD to DUST was carried out. Furthermore, its performance has been verified, since the optimized propeller significantly outperforms the original one, with a reduction of 12.9% of the power and 3.54 dB less of OASPL.

As for the surrogate model, a good prediction is obtained provided that the inputs are contained within the DOE bounds, although some margin is recommended. The large reduction in optimization time allows for several optimizations to be performed, which in this work has been used to generate a Pareto front of the optimal solution. Furthermore, it saves computational power and therefore energy, being a more environmentally friendly approach than the traditional optimization. However, there are some limitations compared to the optimization with DUST: first, the solution found is not as performing; second, it is not possible to vary the rotation rate or the flight condition without building a whole new DOE. Finally, a balance should be found between the DOE size and the number of iterations to be computed, since these benefits are only applicable if the latter is higher than the former, either in term of number of iterations per optimization or the quantity of optimizations performed. Parallelization of the DOE would increase the reasons for using the surrogate optimization, thanks to the lower cost of gathering the samples.

In addition, the simulation of ducted rotors has been incorporated into the workflow. It has proven ducted rotors to be aerodynamically efficient and even have a slightly positive effect on the acoustics. Generally, the results are in agreement with previous research. Some limitations to the process followed in this work is the neglect of viscous forces on the duct and its weight not being accounted for.

8.2 Future work

The workflow could be improved by incorporating the calculation of broadband noise into the acoustics. Moreover, it would be advisable to study the limits up to which the bounds for the chord and twist can be broadened without resulting in a non-realistic blade or large flow separation that cannot be accurately modeled with DUST. Besides, future users of the workflow could validate it by comparing the energetic and acoustic performance of the optimal rotor to rotors of similar size and thrust.

With respect to the surrogate optimization, the reason for the outliers in the Pareto front could be further investigated. More interestingly, a gradient-based optimization can be set up now that the derivatives of the model are easily accessible thanks to the surrogate; also, the surrogate enables the use of a multi-start. These strategies could make the optimization process even faster but their cost using the traditional workflow was prohibitive. In addition, it is generally recommended after performing a surrogate optimization to check the optimality conditions; i.e., verify that the optimum found is also at least a local minimum of the true function – in this case, DUST.

Regarding ducted rotors, it would be advisable to compute the contribution of the viscous

effects to the force of the duct in order not to overestimate the thrust produced by it. It would also be beneficial to quantify in some way its weight. Moreover, the results of the simulation should be validated either with a higher-fidelity simulation or with experimental data. Finally, performing an optimization of a ducted rotor would be interesting, either optimizing only the blade shape, which would be straightforward since everything is already included in the workflow, or including the duct geometry within the design variables too.

8.3 Final remarks

To conclude this document, the global approach taken in this work is highlighted. For one thing, the main objective has been achieved, having developed an optimization workflow for the aeroacoustic optimization of rotors using a vortex particle method solver; and using it to improve the performance of a propeller by a considerable factor. For another, different strategies have been employed to improve this workflow and to broaden its capabilities, via the generation of a surrogate model and enabling surrogate optimizations to be performed; as well as allowing the simulation and the subsequent optimization of ducted rotors.

References

[1] J. E. Ffowcs 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*, vol. 264, no. 1151, pp. 321–342, 1969.

[2] E. Piceni, "Aeroacoustic Multidisciplinary Design Optimisation of an open and ducted rotor," 2022. [Online]. Available: https://hdl.handle.net/10589/196689.

[3] F. Gallard, C. Vanaret, D. Guenot, *et al.*, "GEMS: A Python Library for Automation of Multidisciplinary Design Optimization Process Generation," in *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. DOI: 10.2514/6.2018-0657. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2018-0657. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2018-0657.

[4] F. Boyer, A. Drapier, Y. Mérillac, C. Nana, and R. Serre, "Multidisciplinary optimization of a MAV propeller for noise reduction," 2017.

[5] C. Nana, M. Yann, and R. Serré, "Fast Multidisciplinary Optimization of a MAV propeller for noise reduction: from simulation to experimentation," in *53rd 3AF International Conference On Applied Aerodynamics*, 2018.

[6] F. Farassat, "Derivation of Formulations 1 and 1A of Farassat," *Technical Report NASA/TM-2007-214853, NASA Langley Research Center Hampton, VA, United States*, Mar. 2007.

[7] P. Saliba P. R., "Aeroacoustic modeling of ducted rotors," 2023.

[8] D. Montagnani, M. Tugnoli, F. Fonte, A. Zanotti, M. Syal, and G. Droandi, "Mid-fidelity analysis of unsteady interactional aerodynamics of complex VTOL configurations," 2019.

[9] D. Montagnani, M. Tugnoli, F. Fonte, A. Savino, A. Cocco, and A. Colli, *DUST User Manual*. Aug. 2016.

[10] M. Tugnoli, D. Montagnani, M. Syal, G. Droandi, and A. Zanotti, "Mid-fidelity approach to aerodynamic simulations of unconventional VTOL aircraft configurations," *Aerospace Science and Technology*, vol. 115, p. 106 804, Aug. 2021, ISSN: 12709638. DOI: 10.1016/j.ast.2021.106804. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S127096382100314X.

[11] A. Savino, A. Cocco, A. Zanotti, and V. Muscarello, "Numerical investigation of wing-propeller aerodynamic interaction through a vortex particle-based aerodynamic solver," Sep. 2022.

[12] N. S. Zawodny, D. D. Boyd Jr., and C. L. Burley, "Acoustic Characterization and Prediction of Representative, Small-Scale Rotary-Wing Unmanned Aircraft System Components," presented at the American Helicopter Society (AHS) Annual Forum, May 17, 2016.

[13] R. McKay and M. J. Kingan, "Multirotor Unmanned Aerial System Propeller Noise Caused by Unsteady Blade Motion," in *25th AIAA/CEAS Aeroacoustics Conference*, Delft, The Netherlands: American Institute of Aeronautics and Astronautics, May 20, 2019, ISBN: 978-1-62410-588-3. DOI: 10.2514/6.2019-2499. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2019-2499.

[14] Z. Wang, A. Pandey, M. Sutkowy, *et al.*, "A Comprehensive Approach to Study Aerodynamic and Aeroacoustic Performances of Small Multicopter Unmanned Aerial Systems," in *2018 AIAA Aerospace Sciences Meeting*, Kissimmee, Florida: American Institute of Aeronautics and Astronautics, Jan. 8, 2018, ISBN: 978-1-62410-524-1. DOI: 10.2514/6.2018-0268. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2018-0268.

[15] S. Huberson, E. Rivoalen, and S. Voutsinas, "Vortex particle methods in aeroacoustic calculations," *Journal of Computational Physics*, vol. 227, no. 21, pp. 9216–9240, Nov. 2008, ISSN: 00219991. DOI: 10.1016/j.jcp.2008.06.011. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0021999108003124.

[16] P. Parmentier, G. Winckelmans, and P. Chatelain, "A Vortex Particle-Mesh method for subsonic compressible flows," *Journal of Computational Physics*, vol. 354, pp. 692–716, Feb. 2018, ISSN: 00219991. DOI: 10.1016/j.jcp.2017.10.040. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0021999117308070.

[17] E. Alvarez, T. Critchfield, A. Ning, and A. Schenk, "Rotor-on-Rotor Aeroacoustic Interactions of Multirotor in Hover," in *Proceedings of the Vertical Flight Society 76th Annual Forum*, Virtual: The Vertical Flight Society, Oct. 5, 2020, pp. 1–12. DOI: 10.4050/F-0076-2020-16489. [Online]. Available: https://vtol.org/store/product/rotoronrotor-aeroacoustic-interactions-of-multirotor-in-hover-16489.cfm.

[18] T. F. Brooks, D. S. Pope, and M. A. Marcolini, "Airfoil self-noise and prediction," Tech. Rep., 1989. [Online]. Available: https://ntrs.nasa.gov/api/citations/19890016302/downloads/19890016302.pdf.

[19] D. Shukla and N. Komerath, "Rotorduct aerodynamic and acoustic interactions at low Reynolds number," *Experiments in Fluids*, vol. 60, no. 1, p. 20, Jan. 2019, ISSN: 0723-4864, 1432-1114. DOI: 10.1007/s00348-018-2668-z. [Online]. Available: http://link.springer.com/10.1007/s00348-018-2668-z.

[20] Z.-H. Han and S. Görtz, "Hierarchical Kriging Model for Variable-Fidelity Surrogate Modeling," *AIAA Journal*, vol. 50, no. 9, pp. 1885–1896, Sep. 2012, ISSN: 0001-1452, 1533-385X. DOI: 10.2514/1.J051354. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.J051354.

[21] J. Xu, Z. Han, W. Song, and K. Li, "Efficient Aerodynamic Optimization of Propeller using Hierarchical Kriging Models," *Journal of Physics: Conference Series*, vol. 1519, no. 1, p. 012 019, Apr. 1, 2020, ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/1519/1/012019. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1519/1/012019.

[22] D. Leusink, D. Alfano, and P. Cinnella, "Multi-fidelity optimization strategy for the industrial aerodynamic design of helicopter rotor blades," *Aerospace Science and Technology*, vol. 42, pp. 136–147, Apr. 2015, ISSN: 12709638. DOI: 10.1016/j.ast.2015.

01.005. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1270963815000206.

[23] V. Pham, M. Tyan, T. A. Nguyen, and J.-W. Lee, "Extended Hierarchical Kriging Method for Aerodynamic Model Generation Incorporating Multiple Low-Fidelity Datasets," *Aerospace*, vol. 11, no. 1, p. 6, Dec. 20, 2023, ISSN: 2226-4310. DOI: 10.3390/aerospace11010006. [Online]. Available: https://www.mdpi.com/2226-4310/11/1/6.

[24] X. Jiang, Q. Zhao, G. Zhao, and P. Li, "Integrated optimization analyses of aerodynamic/stealth characteristics of helicopter rotor based on surrogate model," *Chinese Journal of Aeronautics*, vol. 28, no. 3, pp. 737–748, Jun. 2015, ISSN: 10009361. DOI: 10.1016/j.cja.2015.03.011. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1000936115000564.

[25] M. Sessarego, N. Ramos-García, H. Yang, and W. Z. Shen, "Aerodynamic wind-turbine rotor design using surrogate modeling and three-dimensional viscousinviscid interaction technique," *Renewable Energy*, vol. 93, pp. 620–635, Aug. 2016, ISSN: 09601481. DOI: 10.1016/j.renene.2016.03.027. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0960148116302117.

[26] H. Kwon, S. Choi, J.-H. Kwon, and D. Lee, "Surrogate-Based Robust Optimization and Design to Unsteady Low-Noise Open Rotors," *Journal of Aircraft*, vol. 53, no. 5, pp. 1448–1467, Sep. 2016, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.C033109. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.C033109.

[27] Y. Bu, W. Song, Z. Han, Y. Zhang, and L. Zhang, "Aerodynamic/aeroacoustic variable-fidelity optimization of helicopter rotor based on hierarchical kriging model," *Chinese Journal of Aeronautics*, vol. 33, no. 2, pp. 476–492, Feb. 2020, ISSN: 10009361. DOI: 10.1016/j.cja.2019.09.019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1000936119303498.

[28] E. D. Olson, "Three-Dimensional Piecewise-Continuous Class-Shape Transformation of Wings," in *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Dallas, TX: American Institute of Aeronautics and Astronautics, Jun. 22, 2015, ISBN: 978-1-62410-368-1. DOI: 10.2514/6.2015-3238. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2015-3238.

[29] F. X. Caradonna and C. Tung, "Experimental and analytical studies of a model helicopter rotor in hover," in *European rotorcraft and powered lift aircraft forum*, 1981.

[30] M. J. D. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," in *Advances in Optimization and Numerical Analysis*, ser. Mathematics and Its Applications, S. Gomez and J.-P. Hennart, Eds., vol. 275, Springer, 1994, pp. 51–67. DOI: 10.1007/978-94-015-8330-5_4.

[31] S. G. Johnson, *The NLopt nonlinear-optimization package*, https://github.com/stevengj/nlopt, 2007.

[32] S. Das and S. Tesfamariam, "State-of-the-Art Review of Design of Experiments for Physics-Informed Deep Learning," in Feb. 2022. [Online]. Available: https://www.researchgate.net/publication/358576317_State-of-the-Art_Review_of_Design_of_Experiments_for_Physics-Informed_Deep_Learning.

[33] Y.-C. Hung, "A review of Monte Carlo and quasi-Monte Carlo sampling techniques," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 16, no. 1, e1637, 2024.

[34] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.

[35] D. C. Montgomery, *Design and analysis of experiments*. John wiley & sons, 2017.

[36] R. H. Myers, D. C. Montgomery, and C. M. Anderson-Cook, *Response surface methodology: process and product optimization using designed experiments*. John Wiley & Sons, 2016.

[37] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB*. World Scientific Pub Co Inc, 2007.

[38] Columbia University Mailman School of Public Health, *Kriging interpolation explanation*, Mar. 2023. [Online]. Available: https://www.publichealth.columbia.edu/research/population-health-methods/kriging-interpolation#:~:text=The%5C%20value%5C%20that%5C%20is%5C%20generated,Linear%5C%20Unbiased%5C%20Predictors%5C%20(BLUPs)..

[39] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2.

[40] I. Goodfellow, *Deep learning*, 2016.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[42] S. Yilmaz, D. Erdem, and M. Kavsaoglu, "Performance of a ducted propeller designed for UAV applications at zero angle of attack flight: An experimental study," *Aerospace Science and Technology*, vol. 45, pp. 376–386, 2015, ISSN: 1270-9638. DOI: https://doi.org/10.1016/j.ast.2015.06.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1270963815001820.