2016

# Surrogate-based design optimization of dual-rotor wind turbines using steady RANS equations

Andrew Scott Thelen
*Iowa State University*

**Surrogate-based design optimization of dual-rotor wind turbines using steady RANS equations**

by

**Andrew Thelen**

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Aerospace Engineering

Program of Study Committee:

Leifur Leifsson, Major Professor

Anupam Sharma

Christina Bloebaum

Iowa State University

Ames, Iowa

2016

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# ACKNOWLEDGEMENTS

I would like to thank the many people who helped me conduct this research and write this thesis. First and foremost, the knowledge, expertise, and words of encouragement of Dr. Leifur Leifsson were instrumental in completing this work. I would also like to thank my committee members, Dr. Anupam Sharma and Dr. Christina Bloebaum, for their support and helpful insight. Dr. Sharma, I believe, deserves special recognition for lending me his CFD model which serves as the basis of this thesis. I would also like to thank my research group members Jie Ren, Anand Amrit, and Xiaosong Du, whose suggestions were at times instrumental in completing this work. I would also like to thank many other graduate students and faculty who helped me along the way. Finally, I would like to thank my parents and siblings for their unending support.

# ABSTRACT

Dual-rotor wind turbines (DRWT) may offer better energy efficiency over their single-rotor counterparts in isolated and in windfarm operation. The design and analysis of DRWT requires, among other, the use of computational fluid dynamics models. Depending on their formulation, these models can be computationally expensive. Numerous simulations are typically required during the design process, which may render the overall computational cost to be prohibitive. This thesis investigates and compares several optimization techniques for the design of DRWTs. In particular, the DRWT fluid flow is solved using the Reynolds-Averaged Navier-Stokes equations with a two-equation turbulence model on an axisymmetric mesh, while three design approaches are considered: (1) the traditional parametric sweep where the design variables are varied and the responses examined, (2) direct optimization with a derivative-free algorithm, and (3) surrogate-based optimization (SBO) using both data-driven and physics-based surrogates. The approaches are applied to test cases involving two, three, and 11 design variables. Two final cases utilize the physics-based SBO to carry out a parametric study of two of the design variables. The results show that the same optimized designs are obtained with all the approaches. However, going from the two-parameter case to the three-parameter case, the effort of setting up, running, and analyzing the results is significantly higher with the parametric sweep approach. The optimization techniques are more efficient because they require no assumptions of sampling discretization, and are much more likely to find the best design. In addition, they deliver the results with lower computational cost in comparison with the parametric sweep approach, while the SBO algorithms often outperform the direct approach in terms of computational expense.

# NOMENCLATURE

$C_P$ = Power coefficient of single rotor turbine or combined power coefficient for DRWT [-]

$C_{P_m}$ = Power coefficient of main rotor [-]

$C_{P_s}$ = Power coefficient of secondary rotor [-]

$\lambda_1$ = Main rotor tip speed ratio [-]

$\lambda_2$ = Secondary rotor tip speed ratio [-]

$R_1$ = Main rotor tip radius [m]

$R_2$ = Secondary rotor tip radius [m]

$\Delta x$ = Axial separation between the main and secondary rotors [m]

$\mathbf{x}$ = Design variable vector

$\mathbf{f}(\mathbf{x})$ = Model output vector (fine model)

$\mathbf{c}(\mathbf{x})$ = Model output vector (coarse model)

$H$ = Objective function value (function of $\mathbf{f}(\mathbf{x})$ or $\mathbf{c}(\mathbf{x})$)

$\tau$ = Torque [Nm]

$\Omega$ = Rotational rate of rotor [rad/s]

$\rho_\infty$ = Free-stream density [kg/m$^3$]

$v_\infty$ = Free-stream velocity [m/s]

# CHAPTER 1. INTRODUCTION

## 1.1 Motivation

Modern horizontal-axis wind turbines fall short of the theoretical Betz limit for maximum obtainable efficiency because of multiple reasons such as wake loss due to upwind turbines as well as root loss [1, 2]. The root region of a turbine blade has to withstand the gravitational, rotational, and aerodynamic loads, which are substantial due to the large size of the rotor. This structural requirement invariably leads to a root cross-section with high t/c (thickness-to-chord) ratio; such airfoils have poor aerodynamic properties (e.g. lift-to-drag ratio is relatively small). In an effort to increase the energy capture of such wind turbine configurations, the integration of a secondary rotor into the design has been investigated in a number of studies, with many of them yielding promising results [1, 3, 4, 5, 6, 7, 8, 9, 10, 11]. A sample of such a design is provided in Fig. 1.1.

There are a spectrum of numerical tools of varying computational cost and physics fidelity that can be used to model a dual-rotor wind turbine (DRWT) design. Large eddy simulation (LES) [12] and the Reynolds-averaged Navier-Stokes (RANS) [13] simulations have proven to be quite useful for wind turbine applications [1, 2, 6, 8, 14, 15, 16]. With these approaches available, it is important to consider the fidelity necessary, as well as the associated overall computational cost, required to search for an optimized DRWT design. For example, a high-fidelity partial differential equation (PDE) simulation, such as LES, would not be an efficient approach in the preliminary design stage, as sampling throughout the entire design space would be very costly. Similarly, a faster, but lower fidelity model, which may neglect important physics, would not be an effective tool in determining the precise conditions for an optimized DRWT design.

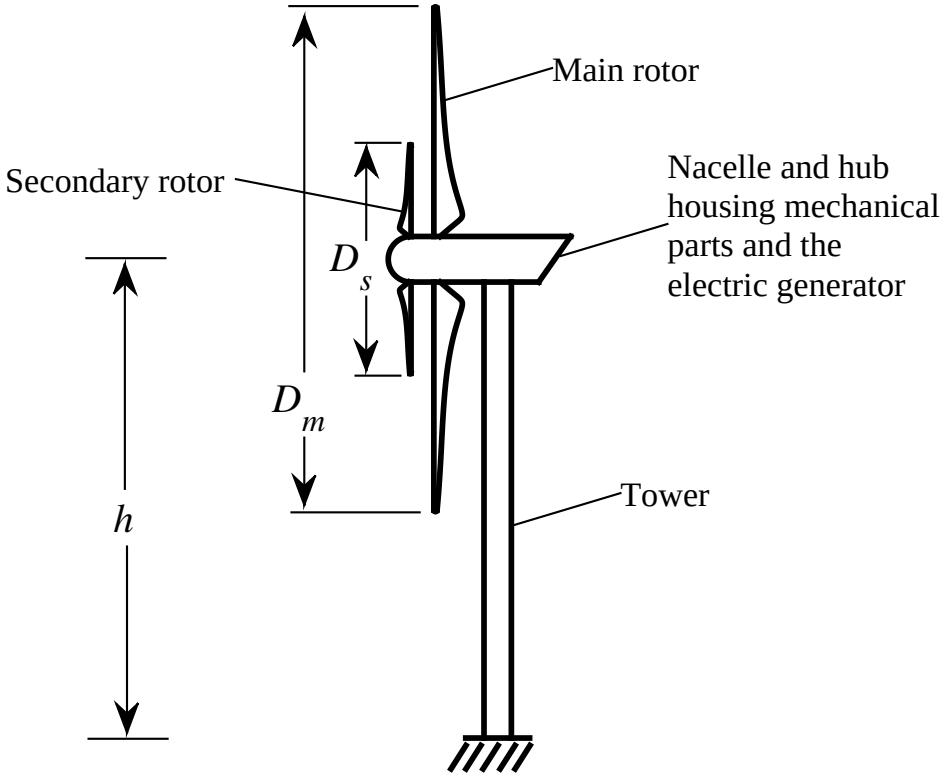



Figure 1.1: A conceptual diagram of the DRWT configuration. $D_m$ and $D_s$ are main and secondary rotor diameters while $h$ is the height of the supporting tower.

Previous studies demonstrate that an actuator disk model, developed using the simpleFoam RANS solver in OpenFoam [17], can be used to model wind turbine performance with reasonable accuracy [1, 2]. Moreover, because certain assumptions allow it to be executed relatively fast (i.e., the unresolved rotor blade surfaces which would introduce boundary layer flow, the steady state assumption, and the lack of hub or tower effects), this type of model can be used to efficiently capture the dominant trends in the DRWT design space. These trends can subsequently be used to dramatically reduce the size of the design space when planning high-fidelity computational fluid dynamics (CFD) simulations or experimental analyses.

While the actuator disk model is faster than other available approaches (such as geometry-resolved RANS [16], or actuator line/disk model LES [1, 8]), the computational resources required to find a global optimum in this design space is nevertheless significant. Because each

design iteration may require multiple model evaluations, the overall computational cost can be prohibitive. For this reason, the work presented in this thesis focuses on developing an efficient approach for optimizing the DRWT using minimal computational resources.

## 1.2 Challenges

There are several challenges to overcome when conducting a CFD-based design optimization process. By in large, these challenges can be categorized based on three aspects of the process: (1) the CFD model, (2) the optimization process, (3) and the integration of the CFD model and optimization algorithm.

### 1.2.1 CFD Model

Depending on their formulation, CFD models can be computationally expensive. Evaluation of a design for one set of flow conditions may require hours, days, or even weeks of computing time. In combination with several other common issues, this high cost makes a CFD-based design optimization process time-intensive and challenging. Moreover, tuning the model for robustness introduces additional challenges to the process.

Closely tied with model cost is convergence of the flow solver. Often, the model must be numerically tuned to more quickly evaluate designs in the design space using the specific grid and fluid flow conditions. For example, the selection of numerical schemes and adjustments of parameters (e.g. CFL number or multi-grid parameters) can significantly affect the amount of time required to evaluate the model. While this task can be challenging on its own, flow solver divergence can sometimes occur later in the optimization process. It is not uncommon for the CFD model to be tuned well for typical designs, while the flow solver diverges for unusual designs within the design space. Thus, tuning the model to reduce the cost of the model can often present unforeseen obstacles later, when the optimization algorithm attempts to evaluate a specific design. Improving the convergence rate and therefore cost of the CFD model thus introduces a unique set of challenges to the design optimization process.

### 1.2.2 Optimization

Invariably, optimization of a design configuration requires multiple model evaluations. Moreover, as dimensionality increases, the number of evaluations required to reach the optimum grows quickly. This is especially true for conventional optimization techniques, which typically require a large number of iterations. Because of this need for numerous evaluations, combined with the high computational cost of each model evaluation, an efficient optimization algorithm is needed. Implementation of such algorithms is not always straightforward, however.

Applying the algorithm can be very time-intensive for two main reasons. Firstly, CFD-based design spaces with large numbers of design variables are often complex and highly nonlinear, and may even have multiple minima. Such scenarios may not be handled well by the optimization algorithm, leading to slow convergence or getting trapped in a local minimum. Secondly, many methods can be implemented in multiple ways, for example by changing one small aspect of the method or by adjusting a fixed parameter. Due to these obstacles, the optimization process may need to be repeated multiple times in order to obtain the best result in terms of design quality or computational expense.

### 1.2.3 Integration of the CFD Model and Optimization Algorithm

Selecting the computational discretization, in other words the mesh, can often be the most time-consuming process in the CFD-based optimization process. This task tends to have a significant effect on two aspects of model output: grid dependence and numerical noise. Through these mesh-based phenomena, optimization using a CFD model can introduce unique challenges that are irrelevant when developing the CFD model or optimization alone.

In order to precisely resolve the flow in a region with nonzero flow gradients, each cell needs to be infinitesimally small. This is impossible because the mesh cell count cannot be infinite, so the flow field must to some extent be approximated. This invariably leads to a finite amount of discretization error in the CFD model, which is often evident in the model output. A common term used to describe this discretization error is grid dependence (which is named as such because the objective function is dependent on not only the governing equations, but the mesh

as well). Typically, the effect of this inherent error must be minimized before beginning a CFD-based optimization task. The basic approach to doing this is to carry out a grid dependence study; for such an analysis, the mesh is refined until the estimated magnitude of discretization error is adequately small. Two consequences may result from not doing so. Most commonly, the magnitude of the model output will differ from that of the theoretical mesh with infinite cell count (which can be estimated by extrapolation). Perhaps more importantly, grid dependence also tends to change where the optima are located in the design space. Therefore, although it can be very time-consuming, selecting a mesh with adequately small grid dependence is an important step in order to obtain meaningful optimization results. Unfortunately, however, a mesh with minimal discretization error also means a more computationally expensive CFD model.

Numerical noise is closely tied to grid dependence, in that it is also caused by discretization error. However, this component of inherent model error refers specifically to the seemingly random oscillations of grid dependence in a minuscule region of the design space. This effect is most significant when using unstructured meshes, particularly those with relatively few cells. For such a mesh, insignificant changes in a design can lead to significant changes in model response (e.g. computed lift or drag) which resemble random noise when generated for a small range of designs. This effect is due to the disorganization of the cells. Two meshes for nearly the same design point typically differ in cell locations and orientations, causing them to have two significantly different values of discretization error. This numerical noise in the model output can be challenging to overcome; often, some methods are completely unable to handle the noisy design space. For example, gradient-based methods that compute gradients by finite difference method often converge immediately on a noise-induced minimum. Fortunately, the use of structured mesh topologies largely minimizes this effect, which is because the cells remain organized in a similar way between adjacent design points.

## 1.3 Research Contribution

The contribution of this work is to investigate efficient strategies for optimizing a dual-rotor wind turbine using CFD simulations. In particular, a multi-fidelity approach is developed for

this specific optimization task, while several additional methods are implemented for comparison. The CFD model used, which was developed by Professor Sharma and his research group at Iowa State University, is relatively cheap but captures the dominant trends of a DRWT design space. Throughout this work, the following optimization approaches will be described and demonstrated:

- Parametric sweep
- Pattern search [18, 19]
- Sequential approximation optimization (SAO) [20, 21, 22, 23, 24]
- Kriging with infill [25, 26, 27]
- Output space mapping (OSM) [28]

## 1.4 Outline

This work is organized as follows. Provided in Chapter 2 is a review of other works on CFD rotor modeling, dual-rotor wind turbines, and optimization using expensive models. In Chapter 3 is a description of the computational model where the specifics of the governing equations, computational fluid dynamics (CFD) model, grid independence study, and numerical validation are given. Then, in Chapter 4 the optimization methodologies are described, including the parametric sweep, direct optimization, and SBO. In Chapter 5, the optimization approaches are applied to several test cases of DRWT, which vary in computational expense and complexity. Conclusions from the investigations are presented in Chapter 6. Supplementary results are provided in Appendix A.

# CHAPTER 2. LITERATURE REVIEW

This chapter provides background information on DRWT optimization. Among these topics, two distinct areas of research are relevant. Firstly, wind turbine analysis is discussed, where two aspects are emphasized: (1) general information on popular wind turbine modeling techniques, and (2) results from DRWT analyses. Secondly, optimization approaches for use on computationally expensive functions are also discussed; here, popular approaches will be described in general terms.

## 2.1 Analysis of Horizontal-Axis Wind Turbines (HAWTs)

### 2.1.1 Computational Modeling

In an atmospheric boundary layer (ABL) flow [29, 15, 8], the rotor blades of a HAWT rotate around a horizontal axis of rotation. The resulting pressure distributions on the blade surfaces yield a net torque around this axis of rotation. As this process continues over time, the kinetic energy of the fluid is converted into mechanical energy [30]. This torque is then transferred through the drivetrain and gearbox into the system's electric generator [31], which converts the mechanical energy into electric energy. The fluid dynamics of such a system can be modeled computationally in many ways; in general, these approaches vary in accuracy, complexity, and computational cost. Among these methods, the two main differences are the governing equations and whether the surface geometry is resolved.

Many sets of governing equations have been used to model HAWTs. Perhaps the most common are the Reynolds-averaged Navier-Stokes (RANS) equations [16, 6, 1, 2, 32] or Large Eddy Simulation (LES) [15, 14, 8]; others instead use Detached Eddy Simulation (DES) [33] or Direct Numerical Simulation (DNS) [34]. Other lower fidelity but faster models even use the

Euler [35] or potential flow equations, for example through the use of the vortex lattice method (VLM) [9]. In addition, many of these models can also vary depending on whether the equations are steady [16, 1, 2, 35] or time-dependent [6, 8, 35, 32]. Among these two alternatives, the use of steady equations tends to be faster and less complex to implement. The disadvantage of this is that the flow structures which are present in realistic HAWT flow may be ignored due to this assumption. For example, when modeling a DRWT, the wake flow of an upstream rotor may periodically affect the performance of the downstream rotor. Consequences of this phenomenon may be completely ignored when steady equations are used.

Regardless of the governing equations, some models resolve the rotor blade geometry [16, 36] while others do not [35, 8, 6, 1, 9, 32]. When the geometry is resolved, the blade surfaces are treated as no-penetration, no-slip surfaces. In such a flow field, there is a boundary layer along rotor blade surfaces. A relatively fine mesh must be used to accurately resolve this boundary layer flow, making this approach costly. When the geometry is not resolved, rotor blade surfaces are discarded from the domain entirely. Instead, the effects of the rotors are modeled by adding or subtracting momentum to the fluid volume elements in the locations of the blades. This is typically done through the use of the actuator line or disk approaches. In general, an actuator line [30, 6, 32] models the rotor's momentum extraction as a section of a disk while an actuator disk does the same for the entire plane of rotation (i.e. the volume swept by the rotors) [30, 37]. These approaches are often used with blade element momentum (BEM) theory (also known as strip theory), which determines how much momentum to add or subtract from the flow based on local flow velocity and direction. Regardless, because boundary layer flow is not present in these models, using an unresolved geometry is faster but potentially less accurate than modeling the rotor surfaces directly.

Among these methods, there is always a trade-off between accuracy and computational cost, while complexity is closely related to cost. For example if cost and complexity were the same for all models, then the use of time-dependent DNS with resolved geometry would always be used (if such models would be available). However, in reality this would lead to a very expensive model. At the same time, alternative methods often yield very similar results, so more complex methods are arguably unnecessary [30]. Moreover, when modeling the interaction of multiple

rotors, the complexity and mesh requirements are even more extensive. This is especially true when the rotors are resolved and rotate in opposite directions, which would necessitate overlapping grids or a moving interface between separate subgrids [6, 36, 38]. As a result, the cheaper and less complex methods tend to be more popular when analyzing DRWTs.

### 2.1.2 Dual-Rotor Wind Turbines (DRWTs)

Many previous works have investigated the aerodynamic performance of DRWTs. Some of these analyses have utilized wind tunnel or field data [3, 4, 7, 10, 11, 5], while others have used computational models [6, 1, 8, 9]. Among these works, the designs being investigated vary by size (i.e. power rating) and whether the rotors are counter-rotating (CRWT) or co-rotating. Overall, the conclusion tends to be the same: DRWTs are in many cases capable of capturing more energy than single-rotor wind turbines (SRWTs). When assessing these works, then, the more valuable considerations are (a) the difference in performance between the DRWT and an equivalent SRWT, and (b) which parameter values lead to the highest overall performance. While parameters tend to vary between studies, the main metric used to assess performance is almost always the power coefficient. This non-dimensional performance metric can theoretically vary between 0 and 1, and describes the power extraction as a fraction of the total flow energy.

In 1985, work by Newman [3] investigated the use of one-dimensional actuator disks for applications to wind turbine analysis. In particular, the author provides a detailed analytical background on the topic and shows that when fluid rotation is neglected, the maximum attainable power coefficient ($C_P$) for a series of $n$ disks is $[8n(n+1)]/[3(2n+1)^2]$. For one rotor this is equivalent to the well-known Betz limit of 16/27, or 59.3%. When a second rotor is introduced, the maximum value is 16/25 (or 64%). Thus, in theory, the best possible DRWT can therefore attain 8% more power than the best possible SRWT. As the number of rotors is increased, however, the improvement per rotor diminishes to an infinitesimal value. For such a case where $n$ approaches a very large value, the maximum $C_P$ approaches 2/3 or 66.67%. This corresponds to a 13% increase in maximum attainable power when compared to a SRWT. This important concept is illustrated in Fig. 2.1.

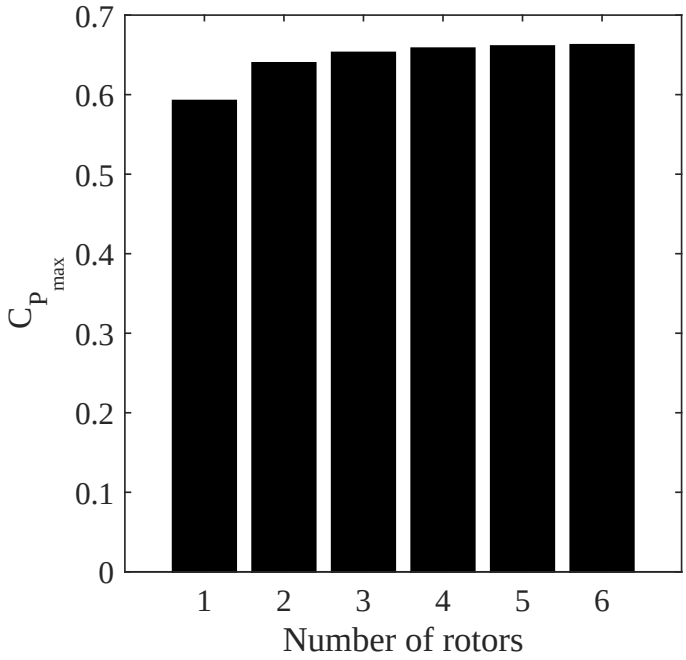



Figure 2.1: Effect of number of rotors on theoretical maximum attainable power coefficient.

In 2002, Appa Technology Initiatives [4] built a prototype CRWT with two 2-bladed rotors with an overall power rating of 6 kW. After 4 months of operation, this study showed that the DRWT could achieve an increase of 30% more power capture than an equivalent SRWT in the same flow. It was also noted that no buffeting of the blades occurred, and that the bending stresses of the tower were reduced for the CRWT. Finally, the authors provide suggestions on how existing SRWTs could be retrofitted to include the secondary rotor.

In 2005, work by Jung *et al.* [5] investigated the performance of a 30 kW CRWT with two 3-bladed rotors. This was done using a combination of quasi-steady strip theory and wind tunnel wake measurements. In particular, the authors study the effect of secondary rotor size and axial separation on power increase relative to an equivalent SRWT. Using an axial separation of zero, the results show that the relative increase in power reaches a maximum of 20% when the secondary rotor is approximately one half the size of the main rotor. It is worth noting, however, that the DRWT system is unique because the main rotor is lengthened using an extension bar. The authors then fix the size of the secondary rotor to this optimal length, and vary the interval length between rotors. An additional increase of 9% (with respect

to the DRWT with zero separation) is observed when rotor separation is one half the main rotor's radius. However, wind tunnel data was not available for greater separation values, so the optimal separation would presumably continue to increase beyond this value of 0.5. Finally, the results were tested using CRWT field measurements. This showed an overall power increase of approximately 21% compared to an equivalent SRWT, while the maximum observed power coefficient was approximately $C_P = 0.5$.

In 2007, work by Shen *et al.* [6] investigated the performance of a CRWT with two three-bladed 500 kW rotors of equal size. This was done through the use of an incompressible time-dependent RANS solver in conjunction with the actuator line technique. BEM theory, along with tabulated airfoil data, is used to compute the amount of body force to apply on the fluid volume elements. In this work, the authors focus on the effect of rotational speeds (at constant inflow velocity) and axial separation between rotors on power extraction. The results show that at high rotational rates, the combined power coefficient can reach as high as 0.55. They also find that rotor separation has very little effect on performance, although there are large fluctuations of power coefficient (in time) when this value is set at 10% of the rotor radius. Finally, they utilize wind speed distribution data (specifically, probability curve of wind speeds) to estimate the annual power improvement of the DRWT compared to an equivalent SRWT. This yields an improvement of 43.5%, and they note that lower wind speeds would see a further improvement.

In 2011, Habash *et al.* [7] investigated the performance of a small CRWT (roughly 0.1-100 kW) with two identical rotors. To do this, they performed experimental testing in a boundary layer wind tunnel. In particular, they studied the total power extraction as a function of axial rotor separation, Reynolds number and wind speed. The results show that rotor separation does not affect performance significantly, although power capture is slightly higher for a distance of approximately one rotor diameter. Both Reynolds and wind speed led to exponential increases in power extraction as these values are increased. In addition, the turbulent intensity of the flow was varied. This was done through the use of different screens upstream of the DRWT, and showed that this flow property can affect power extraction by as much as 12%. Overall, the DRWT was observed to extract up to 20-40% more power than an equivalent SRWT.

Recent work by Sharma *et al.* [9, 1, 8] has specifically investigated the potential mitigation of root loss by positioning a smaller rotor in front of the main rotor. This collection of CFD-based work suggest that the loss of energy capture associated with the root region of a single turbine can be reduced significantly through this approach. For example, in [1], a smaller, upstream rotor (which was aerodynamically tailored based on an inverse design approach) is added to an existing 5 MW SRWT design. The CFD model utilizes the steady incompressible RANS equations in conjunction with the actuator disk technique and BEM theory. The power coefficient is computed as the axial separation between rotors, as well as the tip speed ratio and radius of the secondary rotor, are varied. The authors found that when the main rotor rotates at its optimal tip speed ratio, a maximum power increase of 5-7% is attainable. This corresponds to a secondary rotor tip speed ratio and radius of 6 and 0.25, while the rotor separation is approximately 0.2. A similar design was also tested using the LES technique (with the actuator line approach), which yielded a power increase of 4.6%. Investigations in [8] extend the work in [1] using the same LES CFD model. In particular, the power capture, structural loads, and downstream wake mixing are studied while the inflow velocity distribution and turbulent intensity are varied. The results showed that power increase can vary from 4.9% to 10.8% depending on inflow velocity distribution, with uniform velocity yielding the minimum value. In addition, the root bending moments of the DRWT are reduced slightly compared to a SRWT. The DRWT was also shown to enhance wake mixing downstream of the rotors, with this effect increasing as turbulent intensity is increased. This important result suggests that wake losses in a wind farm would be reduced, leading to further improvements in power capture.

Subsequent investigations done by Hu *et al.* [10, 11] have further confirmed that in a wind tunnel setting, these dual-rotor design configurations can in fact increase the power capture of a turbine operating in isolation. For example, in [11] the authors investigate the performance of DRWTs which use two identical 3-bladed rotors spaced at one half the rotor radius. In particular, this work utilizes a large-scale boundary layer wind tunnel in order to investigate the performance of counter- and co-rotating DRWTs in comparison to an equivalent SRWT. In this work, they observe maximum power gains of 60% and 48% for the counter- and co-rotating DRWTs, respectively. The authors also find that the DRWT systems were subjected

to significantly higher bending moments and axial thrust loading. In addition, [10] specifically investigates the wake losses of a DRWT using the same wind tunnel setup. The secondary, upstream rotor is approximately half the size of the main rotor while axial separation between the two rotors is set to a small value. The work focuses specifically the performance of a SRWT downstream of both a SRWT and a DRWT. The position of the downstream SRWT is varied in the streamwise direction and the resulting power is measured. The results from this work shows that the SRWT captures up to 6% more power (at a distance of 6 rotor diameters) when positioned behind a DRWT as opposed to a SRWT. This is due to the faster recovery of wake flows behind DRWTs when compared to SRWTs. As a result, while DRWTs may capture more power in isolation, further improvements may be seen in a wind farm setting.

## 2.2 Design Using Computationally Expensive Models

There are numerous types of optimization methods capable of optimizing a computationally expensive, computer-based simulation. These methods all belong to one of two categories: (1) direct methods, which include both heuristic and deterministic methods, and (2) surrogate-based methods, which include data-driven and physics-based methods. The following sections will provide an overview of these two classes of optimization methods and their uses in aerodynamic shape optimization (ASO).

### 2.2.1 Direct Optimization

In order to iteratively converge on an optimum in a design space, direct optimization methods utilize the expensive simulation directly rather than indirectly after intermediate processes. The workflow of such a process is illustrated in Fig. 2.2. Methods of this type can differ depending on which information is available from the objective function. For example, 0th-order methods (otherwise known as derivative-free methods), do not utilize derivative information of the function. When using first-order methods, on the other hand, the algorithm is reliant on derivative information of the function. Likewise, second-order methods require the Hessian matrix of the function. In general, the higher the order of optimization method, the more efficient it is in terms of number of function evaluations.

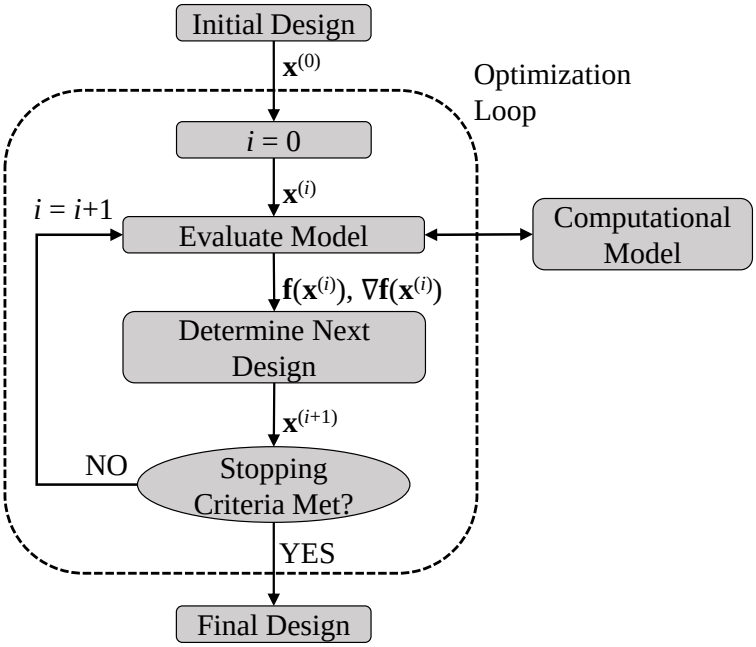



Figure 2.2: Workflow of direct optimization.

#### 2.2.1.1 0th-Order Methods

0th-order direct methods can be categorized as being either heuristic (i.e., dependent on probability in some way) or deterministic (i.e., can be repeated to reach the same solution every time). In general, the deterministic type of direct method is much more popular when optimizing an expensive function. This is because heuristic methods typically require many more function calls; they are instead better suited for global searches of cheap, multi-modal design spaces. For example, Martins *et al.* [39] compared two heuristic methods to several deterministic methods. In particular, the authors solved a wing optimization test case using two of these methods, one genetic algorithm (NSGA2 [40] via pyOpt [41]) and one based on particle swarm optimization (ALPSO [42] via pyOpt [41]). Using eight design variables, all of the methods reached approximately the same solution, with the heuristic methods requiring roughly 1,000 times more function evaluations.

Deterministic 0th-order methods are not among the fastest, but are often needed for ASO when gradient information of the design space is not available. Typically, these methods utilize the estimated gradients in some way. For example, pattern search [18, 19, 43] is a fairly robust

method which utilizes a grid search to estimate which direction to move in. Other methods compute gradients in a less orderly fashion, often using the finite difference method on a very small scale. When applied to ASO, these methods are often ineffective due to the numerical noise caused by the mesh in the simulation.

#### 2.2.1.2 1st- and 2nd-Order Methods

Since the advent of adjoint sensitivity for aerodynamic applications, first-order direct methods have become some of the most popular choices for CFD-based design problems. This valuable technique, which was first applied to aerodynamic design in 1988 by Antony Jameson [44], makes gradient information of a CFD-based objective function available to the optimizer without incurring excessive additional computational cost. To obtain this first-order information of the objective function, an adjoint-solver solution is required, which increases the cost of each function evaluation. In general, the cost of one adjoint solution is comparable to the cost of the flow solution. The adjoint solution is obtained for each cost function or constraint value. Thus, because this cost is independent of the number of design variables, this use of adjoint sensitivity can significantly accelerate the design optimization process. As a result, the most popular direct methods in modern ASO tend to be gradient-based through the use of adjoint information.

Some of the most popular direct methods for ASO are quasi-Newton algorithms. This type of method approximates second-order information of the function using previously obtained 0th- and first-order information. The exact lower-order information along with the approximated Hessian matrix is then applied in a similar manner as Newton's method (with the main difference being constraint handling, in which case Lagrangians are involved). For example, Broyden-Fletcher-Goldfarb-Shanno (BFGS) has proven to be quite popular for ASO applications [45, 46]. In recent years, algorithms based on sequential quadratic programming (SQP) have become very popular as well. This is due especially to the availability of these methods; MATLAB, for example, contains a built-in function *fmincon* which can utilize several quasi-Newton methods, including BFGS and SQP. In addition, many sophisticated optimization codes have in recent decades become widely available. For example, SNOPT [47] is currently

widely used for aerodynamic shape optimization [48, 49, 50], and is just one of many algorithms provided pyOpt [41], a widely available framework for nonlinear constrained optimization.

There are also second-order ASO methods which use the exact Hessian rather than estimating it. In 2001, Taylor *et al.* [51] introduced an approach that uses the automatic differentiation (AD) technique in conjunction with adjoint sensitivities to compute the exact Hessian matrix. Then in 2010, Rumpfkeil *et al.* [46] describe a generalized way to carry out this procedure. Two test cases (ASO of an airfoil in inviscid flow) are solved, one with two variables and another with six. The method is utilized by KNITRO [52], another easily available optimization code which is based on Newton's method. Comparisons to quasi-Newton approaches are made, with the exact Hessian method yielding some of the most efficient outcomes. Overall, these exact-Hessian approaches have shown promising results; however, they may not be among the most popular, which could be due to their complexity and lesser availability.

### 2.2.2 Surrogate-Based Optimization

Surrogate-based optimization (SBO) methods [27, 25, 26, 43] are those which utilize a surrogate model of the objective function. In general, the surrogate model is used in an intermediate step between evaluation of the objective function and selection of the next design to sample. This process is illustrated in Fig. 2.3. Among these methods, there are two general types: data-driven, also called approximation-based, and physics-based methods.

#### 2.2.2.1 Data-Driven Methods

The key difference between data-driven and physics-based surrogate models is that the former utilizes an analytical approximation of the design space. For this reason, data-driven models are sometimes referred to as being approximation-based [25, 26, 43]. These surrogate models can be local or global, and are found using some type of sampling plan. Once the model is formed, it can be evaluated relatively cheaply, therefore making a local or global search of the approximate design space feasible in terms of computational cost. Among these data-driven SBO algorithms, there are three general differences: (1) the sampling technique, (2) the formulation of the surrogate model, and (3) how the algorithm progresses.

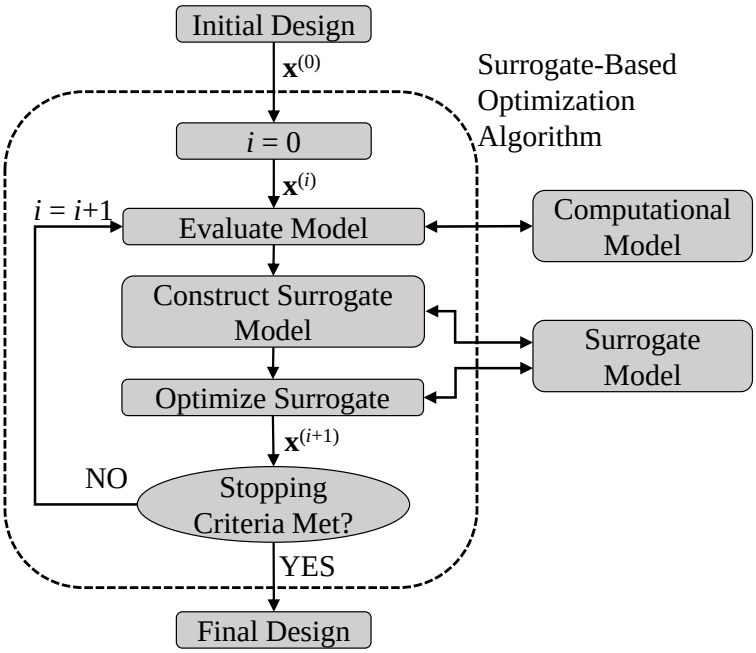
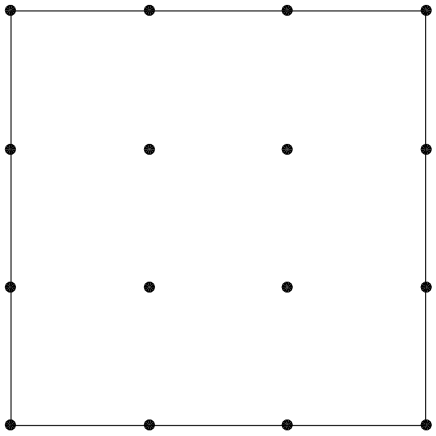
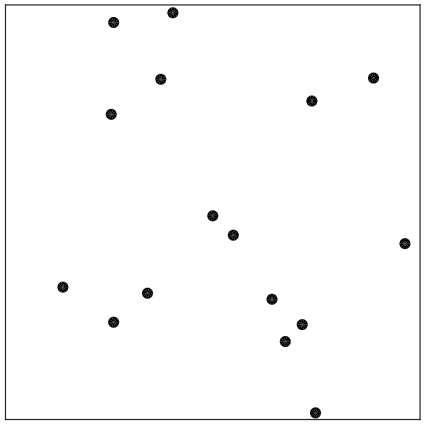



Figure 2.3: Workflow of a surrogate-based optimization (SBO) algorithm.

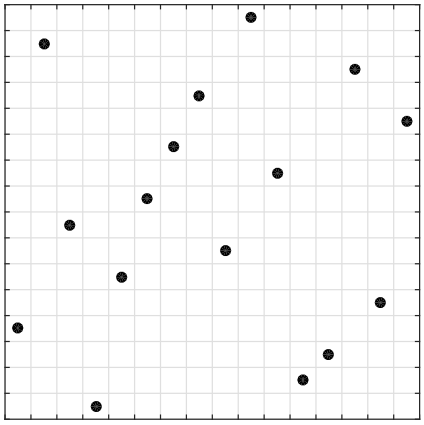
The sampling plan, which is otherwise known as the design of experiments (DOE) [27, 25], used by these methods can have a large effect on computational efficiency. Regardless of whether the method is local or global, the sampling plan must contain enough points to capture dominant trends in the area being searched [53]. At the same time, oversampling in a region of the design space is unnecessarily expensive. Thus, the usual goal when selecting a DOE is to capture the important trends to an extent that allows the algorithm to proceed, while also using minimal computational cost (i.e. number of samples). Provided in Fig. 2.4 are some of the most popular sampling plans. The full factorial approach [54, 27] (Fig. 2.4(a)) is expensive but covers the design space evenly in uniform intervals. Random sampling [54] (Fig. 2.4(b)) is fairly common, but may result in oversampling in some regions and very few samples in others. Latin Hypercube Sampling, or LHS [25, 27, 54] (Fig. 2.4(c)), is one of the most popular because it helps ensures that samples are more widely distributed compared to random sampling. Also, it tends yields better results than full-factorial. Star distribution sampling [55, 24] (Fig. 2.4(d)) is another form of factorial sampling, and is quite popular for local SBO methods. In addition, there are other techniques that can help improve sampling plans. For example, Fig. 2.5 shows an LHS plan that has been optimized using an evolutionary algorithm [27].



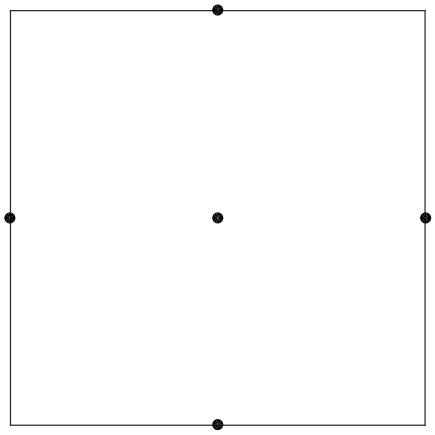
(a) Full factorial



(b) Random



(c) LHS



(d) Star distribution

Figure 2.4: Common sampling plans for use in data-driven SBO methods [55].

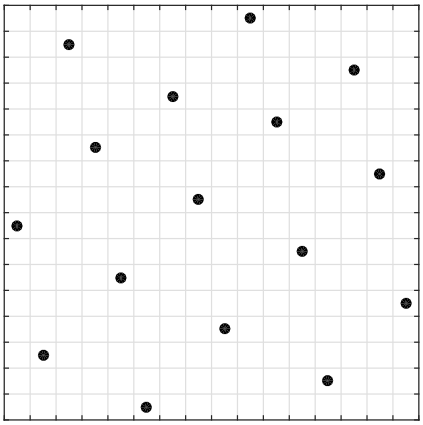


Figure 2.5: LHS sampling plan optimized using an evolutionary algorithm [27].

Regardless of the DOE used, data-driven SBO methods [25, 26, 43] may use many different types of surrogate models. Among the most common for ASO are polynomial response surfaces [25] and kriging models [25, 26, 56]. Response surfaces usually use least squares method to find a polynomial fit of the design space [57]. Kriging also has a polynomial component in addition to a probabilistic component. Another approach widely used for ASO is the Artificial Neural Network (ANN) method [58]. Support Vector Machines (SVM) [59], in addition, have been used in aeronautics but are not commonly applied to aerodynamic shape optimization.

In order for data-driven SBO algorithm to converge, two general approaches may be taken. Firstly, when utilizing a local search, one or more design points are usually added on or near the surrogate's optimum. As the algorithm progresses, the local search region approaches the vicinity of an optimum, while in the process the search space may shrink or grow depending on the surrogate's accuracy (which is known as the trust region technique [21, 60]). This type of method, which utilizes a series of local surrogate-based searches, is known as sequential approximation optimization (SAO) [20, 21, 22, 23, 24]. Secondly, while a similar approach is sometimes used for global search algorithms, the surrogate may instead be updated near points of high surrogate model error. This incrementally improves global accuracy of the surrogate throughout the entire design space; this process is often called the "infill" approach because the surrogate model is incrementally updated with design space data at infill points.

To ensure usable accuracy of these methods, a large number of data samples are required. Moreover, the number of samples grows quickly with the problem dimensionality. Therefore, while data-driven SBO methods can be quite efficient for low-dimensional cases (less than 10 design variables), they are typically less efficient than alternative methods for larger dimensionality (e.g., direct gradient-based and physics-based SBO).

#### 2.2.2.2 Physics-Based Methods

An alternative way of performing SBO is to use physics-based surrogate models [19]. These models are constructed by correcting/enhancing the underlying physics-based low-fidelity models [60]. A low-fidelity model, or a simplified description of the system under consideration, can be obtained by neglecting certain physical or second-order effects, using simplified equations, or by exploiting the high-fidelity model with a coarser computational description. The low-fidelity models are subsequently corrected or enhanced to become a reliable representation of the high-fidelity model by specific methods such as multiplicative or additive corrections [22], space mapping (SM) [61, 28], adaptive response correction (ARC) [62], adaptive response prediction (ARP) [63], multi-level optimization [64, 65], or shape-preserving response prediction (SPRP) [66]. Although the physics-based surrogate models are not as versatile as the data-driven ones, they have the potential to offer significantly better efficiency in terms of the computational cost. Many algorithms require only a single high-fidelity model evaluation per design iteration [60]. Consequently, physics-based surrogates may exhibit better generalization capability than the data-driven ones.

# CHAPTER 3. ROTOR COMPUTATIONAL MODEL

This chapter provides details on the computational model used throughout this work. In particular, the governing equations are described Section 3.1. In Section 3.2 the CFD model is described, where the flow solver, computational domain, and boundary conditions are briefly described. Section 3.3 describes a grid independence study using the CFD model. Finally, numerical validation of the model is provided in Section 3.4, where the CFD model output is compared to experimental measurements and analytical predictions for a single-rotor design.

## 3.1 Governing Equations

The incompressible Reynolds-averaged Navier-Stokes (RANS) equations are taken as the governing flow equations. The continuity equation is

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0, \tag{3.1}$$

and the momentum equation is

$$\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial {x_j}^2} - \frac{\partial \overline{u_i' u_j'}}{\partial x_j} + \frac{f_i}{\rho}, \tag{3.2}$$

In these equations, $u$, $x$, $\rho$, $p$, and $\nu$ represent velocity, position, density, pressure, and kinematic viscosity, respectively. The overbar in these equations denotes a time averaging operation, while the prime symbol represents a perturbation from the mean value. The Reynolds stress tensor, $\overline{u_i' u_j'}$ is modeled using the standard two-equation $k-\epsilon$ turbulence model [67]. The effect of turbine rotors is modeled using the body force term, $f_i$. The amount of body force is determined using the computed local flow velocity and blade element theory with user-provided

lookup tables of rotor blade sectional lift and drag force coefficients. These coefficients can be obtained experimentally or numerically and may be corrected for 3D stall effects. The body force is distributed over a volume and a Gaussian distribution is applied along the flow direction as suggested in Mikkelsen [37]. Prandtl's tip loss correction is applied to account for finite-span induction (tip) loss that is ignored in the blade element theory.

## 3.2 CFD Model

The RANS equations are solved using simpleFoam [17] on an axisymmetric domain with a thickness of one cell in the circumferential direction. The other surfaces of the domain lie at 10 times the main rotor radius in the upwind, downwind, and spanwise directions, while the main rotor radius $R_1$ is nondimensionalized to one. A schematic of the CFD domain is provided in Fig. 3.1. A nondimensional velocity, density, and pressure of one is applied to the inlet surface and a zero-gradient boundary condition is applied on the exit boundary. In the diagram, flow direction is from left to right.

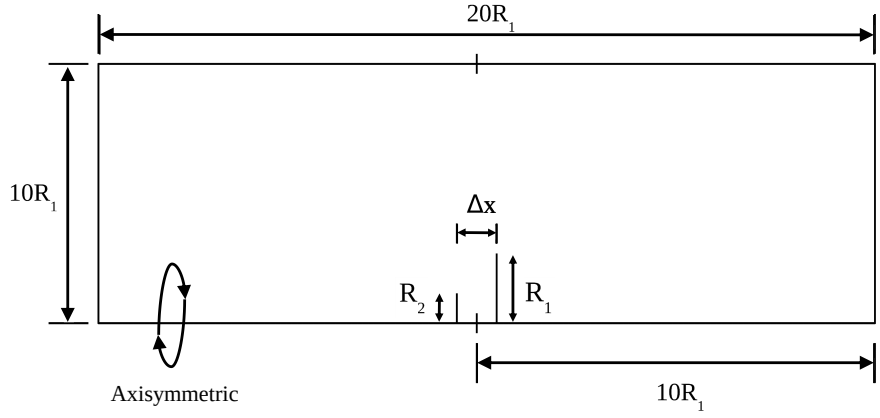



Figure 3.1: Schematic of the solution domain used by the CFD model. $R_1$ and $R_2$ are the radii of the primary and secondary rotors of a DRWT, and $\Delta x$ is the axial separation between the rotors.

## 3.3 Grid Independence Study

A structured mesh, measuring one cell thick in the circumferential direction, is created using the OpenFoam blockMesh utility. This meshing tool generates a structured grid in any number of blocks within the domain. The blockMesh utility, within each block and in each direction, solves for a number of cells as defined by the user. In each direction, the ratio of cell size between the first and last cells in each block can be modified as well. An example of the resulting mesh is provided in Fig. 3.2.

In order to ensure that mesh dimensions do not have an impact on the objective function value, a mesh convergence study was carried out. The design point of $\lambda_1 = 8, \lambda_2 = 6, R_2 = 0.2, \Delta x = 0.5$ was used for this study and the results are provided in Fig. 3.3. The results indicate that the resolution of the numerical discretization does play a role in the predicted value of power coefficient ($C_P$). Moreover, for very small grid dependence, a mesh cell count of hundreds of thousands of cells may be needed. Note that similar trends are provided in Ref. [30], where the authors showed a noticeable grid dependence using both actuator disk and line models; furthermore, they also note that mesh refinement increases the computed power coefficient.

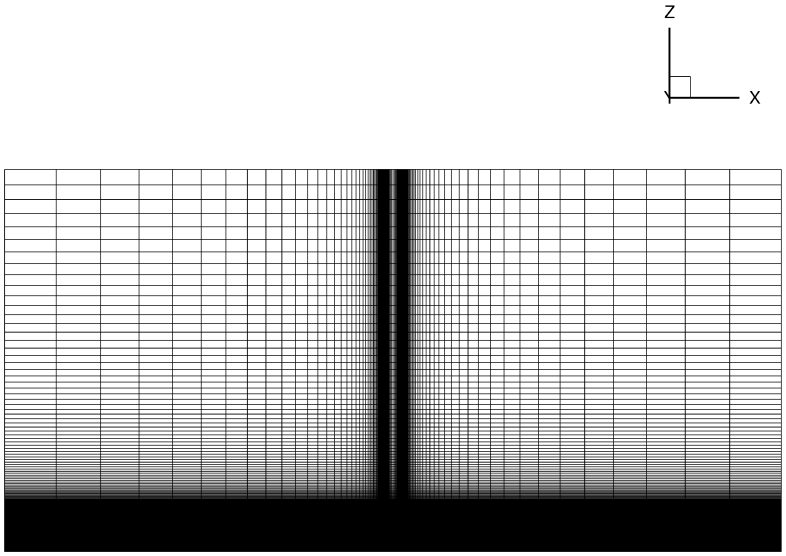
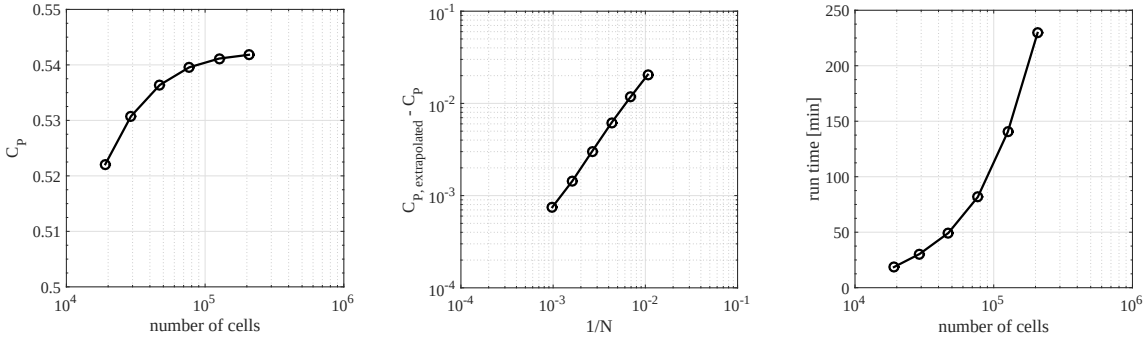



Figure 3.2: Example mesh of the DRWT.

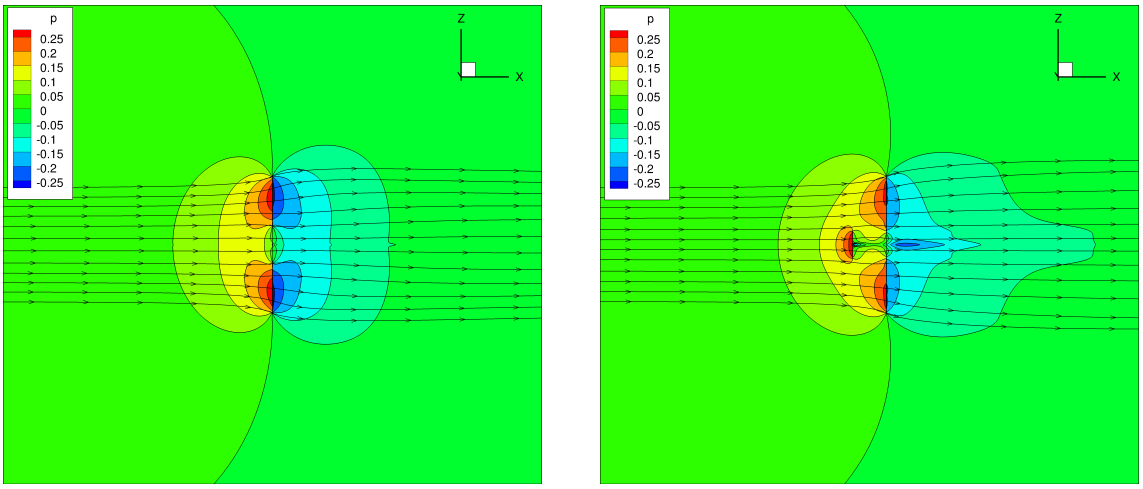



(a) Power coefficient (b) $C_P$ extrapolation (c) Run time

Figure 3.3: Grid independence study results: the effect of number of cells on DRWT power coefficient (left), an approximate error of DRWT power coefficient (center) where $C_{P,extrapolated} = 0.54257$ and $N$ is the number of cells in the stream-wise direction, and simulation run time using a single processor (right).

Initially, this work uses the third coarsest mesh, with approximately 47k cells, for the optimization runs. This mesh was selected because the $C_P$ predicted using this mesh is within 1 percent (0.6%) of the extrapolated value, and further refinement would increase the cost of optimization immensely. In addition, the CFD model will always have some inherent error associated with its assumptions. Thus, further reduction of the error due to more refinement would be insignificant when compared to the error associated with the model's simplification of physics.

However, in Section 5.1.4, this grid convergence study is revisited for several reasons. Firstly, a finer mesh is ultimately decided on because the level of refinement affects where the optimum is located in the design space. Secondly, because this work utilizes a multi-fidelity approach, a wider selection of coarse meshes was ultimately needed. In total, this work utilizes four different meshes, two for low-fidelity models, in addition to the initial and refined meshes. Further details about this process are provided in Section 5.1.4. The example model output are provided in Fig. 3.4 where the axial separation between rotors is 0.5, the tip speed ratio of the main rotor $\lambda_1 = 8$, and the tip speed ratio and radius of the secondary rotor are $\lambda_2 = 6$ and $R_2 = 0.2$, respectively.

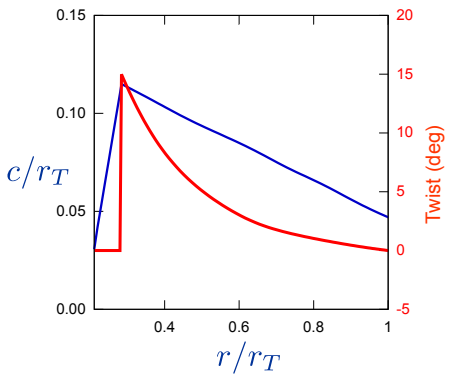



(a) SRWT ($\lambda_1 = 8$) (b) DRWT ($\lambda_1 = 8, \lambda_2 = 6, R_2 = 0.2, \Delta x = 0.5$)

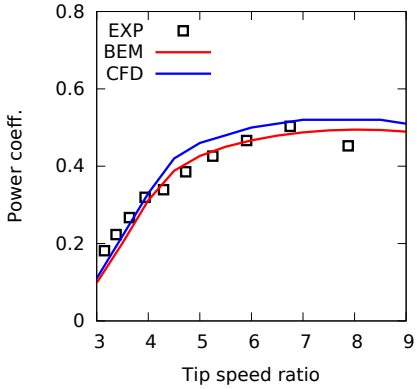
Figure 3.4: Example flow solutions showing gauge pressure contours for single and dual rotors.

## 3.4 Numerical Validation

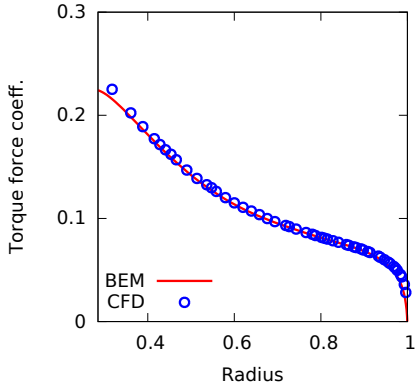
The actuator disk model method is validated against experimental data and blade element momentum (BEM) theory predictions. The three-bladed, stall controlled, 95 kW Tellus T-1995 is used for validation, and the experimental data is obtained from Ref [68]. The turbine rotor diameter is 19 m. The blade chord and twist distributions are shown in Fig. 3.5(a). Figure 3.5(b) compares the turbine characteristic ($C_P - \lambda$) curve as predicted by the actuator disk model against measured data as well as BEM predictions. Variations of torque force coefficient, $C_\tau$ and thrust force coefficient, $C_T$ with blade span, $\lambda$ agree well between the CFD and BEM predictions. Torque force is the component of the aerodynamic force in the plane of the turbine rotor, and the thrust force is normal to this plane. Details about the model implementation and validation can be found in Refs. [1, 8, 2].



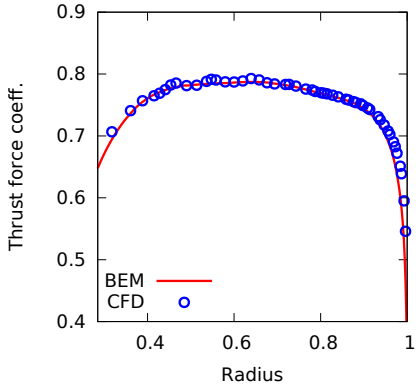

(a) Chord, and twist




(b) $C_P - \lambda$ curve




(c) $C_\tau$




(d) $C_T$

Figure 3.5: Geometry and verification results for the Tellus T-1995 turbine: (a) blade chord and twist distributions, (b) comparison between data and predictions, and radial variations of (c) torque force coefficient, $C_\tau$ and (d) thrust force coefficient, $C_T$.

# CHAPTER 4. OPTIMIZATION METHODOLOGY

Each model evaluation is computationally expensive, so depending on the optimization method used, the amount of time needed to find the optimum can vary dramatically. Therefore, minimizing the number of function evaluations is of primary interest when selecting an optimization method. Having this in mind, this chapter presents three general approaches. These include the traditional parametric sweep, a direct optimization method, and three types of surrogate-based methods, two of which exploit data-driven models while the third exploits a physics-based model. Among the particular methods used, the direct and data-driven methods are essentially off-the-shelf algorithms which could be applied to any design space. The final method, which utilizes a multi-fidelity framework, is formulated specifically for the DRWT computational model.

## 4.1 Optimization Problem Formulation

The purpose of an optimization process is to locate the best set of design variables $\mathbf{x}^*$ within a feasible design space. In its most general form, this task is approached by solving the problem

$$\begin{aligned} \mathbf{x}^* = & \underset{\mathbf{x}}{\operatorname{argmin}} H(\mathbf{f}(\mathbf{x})) \\ \text{s.t. } & \mathbf{g}(\mathbf{x}) \leq 0, \\ & \mathbf{h}(\mathbf{x}) = 0, \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \end{aligned} \tag{4.1}$$

where $\mathbf{x}$ is the design variable vector, $H$ is the objective function, $\mathbf{f}$ is the high-fidelity model response vector, $\mathbf{g}$ are inequality constraints, $\mathbf{h}$ are equality constraints, and $\mathbf{l}$ and $\mathbf{u}$ are the lower and upper bounds on $\mathbf{x}$, respectively.

## 4.2 Search by Parametric Sweeps

The traditional use of parametric sweeps involves sampling the design space at uniformly distributed design points. The maximum value found in this search is then taken to be the optimum. Clearly, this relatively simple method is more costly than direct or surrogate-based optimization methods. Moreover, it does not guarantee that the optimal design is found due to finite size of parameter discretization used in the sweep.

## 4.3 Direct Optimization Methods

A derivative-free method was chosen to perform direct optimization. Gradient-based approaches are not a good choice in this case because adjoint sensitivity information is not available, and the CFD model is not quite smooth due to the use of lookup tables. This aspect of the model leads to very small fluctuations in computed turbine power coefficient throughout the design space. These jumps are virtually invisible in most cases, but can sometimes lead to inaccurate gradient estimates on a small scale.

The particular method chosen for this application is the pattern search algorithm [18, 19]. Methods of this type are typically more costly than their gradient-based counterparts but have the benefit of being more immune to numerical noise, which may be present when using coarse-mesh simulation models. The pattern search algorithm is a stencil-based local optimization method that explores the neighborhood of the current design point. A rectangular grid (i.e., one point in each direction and in each dimension) is used in this implementation. The search process utilizes grid-constrained line search with the search direction determined using the objective function gradient estimated from perturbed designs. In case of a failure the best perturbation (if better than the current design) is selected. Finally, the grid is refined if the poll step does not yield an improved design; this stage is illustrated in Fig. 4.1. For more information about the particular pattern search algorithm used in this work, refer to Ref. [18, 19].

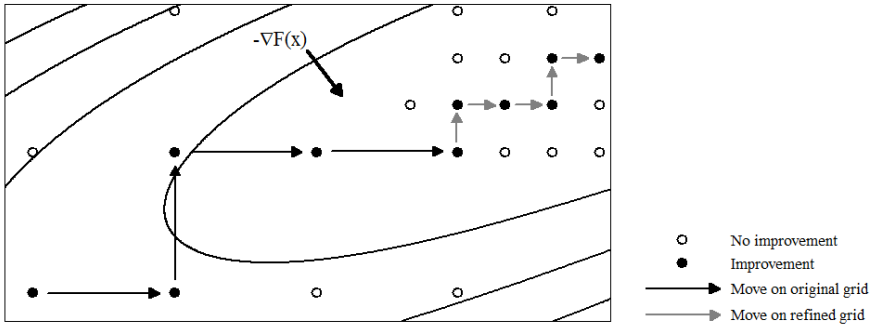



Figure 4.1: A conceptual diagram of pattern-search method using a rectangular grid [18, 19].

## 4.4 Surrogate-Based Optimization with Data-Driven Models

Optimization methods that exploit surrogate models are numerous, and can vary significantly in efficiency and complexity [25, 26, 43]. Moreover, some types of methods are most efficient for low-dimensional problems (less than 10 design variables) while others are efficient for well into the medium-dimensional realm (10-40 design variables). In general, SBO methods can exploit two different types of surrogate models, which include the data-driven and physics-based varieties. Data-driven surrogates include models such as polynomial response surfaces [25] and kriging [26], while physics-based surrogates include methods such as space mapping [61, 28], shape-preserving response prediction [66], adaptive response correction [62], adaptive response prediction [63], and multi-level optimization [64, 65]. The SBO methods used in this work exploit polynomial response surfaces [25], kriging models [26], and space mapping [61, 28] surrogates.

### 4.4.1 Kriging with Infill

In general, a kriging surrogate model is the sum of two components [26]: one term is a polynomial regression model while the other represents a fluctuation around this trend. The fluctuations are modeled as a process with zero mean, and are assumed to be correlated as a function of distance as described by the selected correlation model.

The kriging model is given as [26]

$$\hat{y}(\mathbf{x}) = \beta^{*T} u(\mathbf{x}) + \gamma^{*T} v(S, \mathbf{x}), \tag{4.2}$$

where

$$u(\mathbf{x}) = [u_1(\mathbf{x}), ..., u_m(\mathbf{x})]^T \tag{4.3}$$

and

$$v(S, \mathbf{x}) = [R(\theta, s^1, \mathbf{x}), ..., R(\theta, s^m, \mathbf{x})]^T. \tag{4.4}$$

In these expressions, $m$ is the number of sampled points, $s^i$ is the $i^{th}$ sampled design point (while $S$ represents the set of all $s$), $\mathbf{x}$ is the design point for which a prediction is needed, and $\hat{y}(\mathbf{x})$ is the predicted response at this value $\mathbf{x}$. The $(^*)$ symbol denotes the optimal values of scaling parameters. Given in (4.3) and (4.4) are expressions for $u$ and $v$, which represent unweighted polynomial regression and fluctuation terms, respectively. The model parameters $\beta$, $\gamma$, and $\theta$ are computed within the DACE kriging Matlab toolbox [56].

The parameters $\beta$, $\gamma$, and $\theta$ are functions of not only $S$ and $f(S)$, but also the user-defined regression and correlation models. In this work, a second order polynomial regression model and the Gaussian correlation model [26]

$$R(\theta, s^i, \mathbf{x}) = \prod_{j=1}^{n} R(\theta_j, (\mathbf{x}_j - {s_j}^i)) = \prod_{j=1}^{n} \exp(-\theta_j (\mathbf{x}_j - {s_j}^i)^2) \tag{4.5}$$

are used. In order to generate the surrogate, the 2nd-order regression model requires more samples than the 0th- or 1st-order alternatives. The value for this lower limit is given as [26]

$$m_{min} = 1 + 2n + {}_nC_2, \tag{4.6}$$

where ${}_nC_2$ is the binomial coefficient.

In most cases, a single global kriging model of the design space is either prohibitively expensive or too inaccurate to be used for optimization. For this reason, criteria for updating an initial kriging model, which is generated from sparsely sampled data points, are explored. The initial sampling plan is generated using an algorithm provided in Ref. [27], which generates a Latin hypercube sampling (LHS) plan with an optimized Morris-Mitchell space filling metric by means of an evolutionary algorithm. The minimum number of samples, given by (4.6), is used for each trial.

In terms of infill criteria, two different methods are applied, which include minimizing the predicted objective function (IC1) and maximizing the expected improvement (IC2), which are defined as

$$\mathbf{x}_{IC1} = \underset{\mathbf{x}}{\operatorname{argmin}}(\hat{f}(\mathbf{x})), \tag{4.7}$$

and

$$\mathbf{x}_{IC2} = \underset{\mathbf{x}}{\operatorname{argmin}}(-E[I(\mathbf{x})]). \tag{4.8}$$

Here, following Ref. [27], $E[I(\mathbf{x})]$ for nonzero surrogate error $\hat{s}$ is given as

$$E\left[I(\mathbf{x})\right] = (y_{min} - \hat{y}(\mathbf{x}))\Phi\left(\frac{y_{min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) + \hat{s}\phi\left(\frac{y_{min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right), \tag{4.9}$$

which can implemented using the alternative form

$$E\left[I(\mathbf{x})\right] = (y_{min} - \hat{y}(\mathbf{x}))\left[\frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{y_{min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})\sqrt{2}}\right)\right] + \hat{s}\frac{1}{\sqrt{2\pi}}\exp\left[\frac{-\left(y_{min} - \hat{y}(\mathbf{x})\right)^2}{2\hat{s}(\mathbf{x})^2}\right]. \tag{4.10}$$

In these expressions, $\Phi$ and $\phi$ are the cumulative distribution and probability density functions, respectively, $\hat{s}$ is the predicted error at the input design point $\mathbf{x}$, and $y_{min}$ is the minimum objective function value sampled so far.

When using IC1, the update point is found using two steps:

1. A genetic algorithm (GA) is used to find a point close to the optimum, and
2. The corresponding point is used as an initial point for the pattern search method.

This approach is relatively costly (sometimes on the order of a few seconds), but can typically handle highly multi-modal kriging models. The additional cost associated with it is insignificant in this problem, however, because the surrogate can be evaluated very cheaply when compared to the CFD model.

When using IC2, a similar approach is used to find the global minimum. In this case, however, two different initial points are used by the pattern search method. These two optima are then compared, and the better design is selected to be the next update point. The two initial points used by pattern search are

1. The surrogate's predicted optimum (i.e., the process used for IC1), and
2. The optimum found by a GA.

This process may take 2-3 times longer than IC1, especially if a large number of generations are used by the GA. However, if the objective function is very expensive to evaluate, then the added cost is relatively cheap.

### 4.4.2 Sequential Approximation Optimization

A simple variation of SBO with data-driven surrogates is sequential approximate optimization (SAO) [20, 21, 22, 23]. In SAO, the optimization is limited to a certain (typically, rectangular shaped) subregion of the search space, in which a local surrogate model, such as a low-order polynomial, is set up based on a limited number of high-fidelity model data sampled in the subregion. A new design is found by optimizing the surrogate within the subregion, which is followed by setting up a new subregion according to a chosen relocation strategy. Often the new subregion is in the direction of the last iteration optimum [23], i.e., the last optimum becomes the center of the new region. Adjustment of the subregion size may be based on the quality of the designs obtained by optimizing the surrogate, e.g., a comparison of the actual

versus predicted improvement of the high-fidelity objective function, as in a trust-region-like framework [21, 60]. An illustration of this process is presented in Fig. 4.2.

In this thesis, the work of Leifsson *et al.* [24] is followed, so the local surrogate models are built using a quadratic response surface without mixed terms:

$$\hat{y}(\mathbf{x}) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \beta_{ii} {x_i}^2, \tag{4.11}$$

where $n$ is the space dimensionality, and the coefficients $\beta_i$ are found as

$$\beta = (U^T U)^{-1} U^T y_s \tag{4.12}$$

with

$$U = \begin{bmatrix} 1 & {x_1}^1 & \cdots & {x_n}^1 & (x_1{}^1)^2 & \cdots & (x_n{}^1)^2 \\ \vdots & & & & & & \vdots \\ 1 & {x_1}^m & \cdots & {x_n}^m & (x_1{}^m)^2 & \cdots & (x_n{}^m)^2 \end{bmatrix}. \tag{4.13}$$

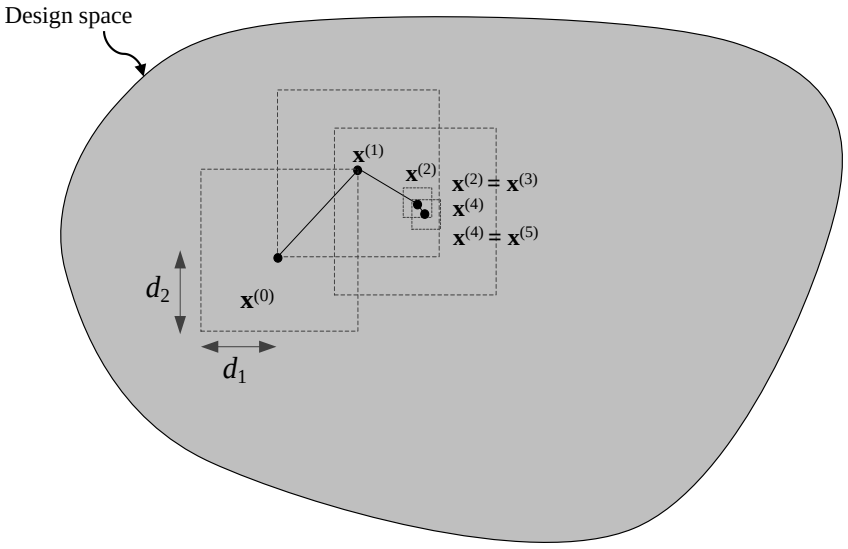



Figure 4.2: SAO algorithm flow. For iteration $i$, $\mathbf{x}^{(i)}$ is the current design point while $\mathbf{d} = [d_1\ d_2]^T$ describes the size of the local search space in each dimension of parameter space.

Here, $m$ is the number of samples, and $y_s = H(\mathbf{f}(\mathbf{x}))$ are the sampled high-fidelity data. In this work, a star distribution of $m = 2n + 1$ grid points is used, where $n$ is the number of design variables, is sampled around the current iteration's design point.

Once the response surface is identified, the optimum on the surface is found. The design space is then sampled at this point and one of two actions are subsequently taken. If $H(\mathbf{f}(\mathbf{x}))$ at this new point is worse than $H(\mathbf{f}(\mathbf{x}))$ at the iteration's center point, then the grid size is scaled down by a user-defined parameter. If, on the other hand, $H(\mathbf{f}(\mathbf{x}))$ at this new point is better than $H(\mathbf{f}(\mathbf{x}))$ at the center point, then the central grid point is relocated to the new design point and the iteration is complete. In the latter case, the grid size is scaled up or down by user-defined parameters depending on the value of the gain ratio

$$r = \frac{H(\mathbf{f}(\mathbf{x}^{tmp})) - H(\mathbf{f}(\mathbf{x}^{(k)}))}{H(\hat{y}^{(k)}(\mathbf{x}^{tmp})) - H(\hat{y}^{(k)}(\mathbf{x}^{(k)}))}, \tag{4.14}$$

where $\mathbf{x}^{tmp}$ is the current surrogate's optimum and $k$ denotes the current iteration. This process is repeated until at least one of the predetermined stopping criteria has been met.

## 4.5 Surrogate-Based Optimization with Variable-Fidelity Models

### 4.5.1 Optimization Algorithm

A generic variable-fidelity optimization algorithm produces a sequence $\mathbf{x}^{(i)}, i = 0, 1, ...,$ of approximate solutions to (4.1) (here, $\mathbf{x}^0$ is the initial design) as

$$\mathbf{x}^{(i+1)} = \underset{\mathbf{x}}{\operatorname{argmin}} H(\mathbf{s}^{(i)}(\mathbf{x})), \tag{4.15}$$

where $\mathbf{s}^{(i)}(\mathbf{x})$ is a surrogate model at iteration $i$. The surrogate model $\mathbf{s}$ is constructed using a suitably corrected physics-based low-fidelity model $\mathbf{c}$. The surrogate model $\mathbf{s}$ is supposedly much cheaper in computational terms than the high-fidelity model $\mathbf{f}$. On the other hand, it is assumed to be sufficiently accurate, particularly in the vicinity of the current design $\mathbf{x}^{(i)}$. If these conditions are satisfied, the algorithm (4.15) is likely to quickly approach the high-fidelity optimum $\mathbf{x}^*$. Typically, the high-fidelity model $\mathbf{f}$ is evaluated only once per iteration (at every

new design $\mathbf{x}^{(i+1)}$). To optimize the surrogate model, the pattern search algorithm [18, 19] discussed previously is used.

### 4.5.2 Multi-Point Output Space Mapping

In this work, multi-point output space mapping (OSM) [28] is used. For this problem, and for this variation of the space mapping technique, the surrogate model is constructed as

$$\begin{aligned}\mathbf{s}^{(i)}(\mathbf{x}) &= \mathbf{A}^{(i)} \circ \mathbf{c}(\mathbf{x}) + \mathbf{D}^{(i)} + \mathbf{q}^{(i)} \\ &= \left[a_m{}^{(i)} c_m(\mathbf{x}) + d_m{}^{(i)} + q_m{}^{(i)} \quad a_s{}^{(i)} c_s(\mathbf{x}) + d_s{}^{(i)} + q_s{}^{(i)}\right]^T,\end{aligned} \tag{4.16}$$

where $\circ$ denotes component-wise multiplication, and the subscripts $m$ and $s$ indicate main and secondary rotor components of the model response vector (which are $\mathbf{c}(\mathbf{x}) = [c_m \; c_s]^T$ and $\mathbf{f}(\mathbf{x}) = [f_m \; f_s]^T$ for the coarse and fine models, respectively). Note that when using alternative space mapping approaches, this surrogate model formulation may differ. For example, input space mapping applies additive and multiplicative terms to the input argument $\mathbf{x}$, rather than doing a similar process on the output arguments $\mathbf{c}(\mathbf{x})$. For OSM, the response correction parameters $\mathbf{A}^{(i)}$ and $\mathbf{D}^{(i)}$ are obtained by solving

$$[\mathbf{A}^{(i)}, \mathbf{D}^{(i)}] = \underset{[\mathbf{A},\mathbf{D}]}{\operatorname{argmin}} \sum_{k=0}^{i} ||\mathbf{f}(\mathbf{x}^{(k)}) - (\mathbf{A} \circ \mathbf{c}(\mathbf{x}^{(k)}) + \mathbf{D})||^2, \tag{4.17}$$

i.e., the response scaling is supposed to (globally) improve the matching for all previous iteration points. The additive response correction term $\mathbf{q}^{(i)}$ is defined as [28]

$$\mathbf{q}^{(i)} = \mathbf{f}(\mathbf{x}^{(i)}) - \left[\mathbf{A}^{(i)} \circ \mathbf{c}(\mathbf{x}^{(i)}) + \mathbf{D}^{(i)}\right], \tag{4.18}$$

i.e., it ensures a perfect match between the surrogate and the high-fidelity model at the current design $\mathbf{x}^{(i)}$, i.e. $\mathbf{s}^{(i)}(\mathbf{x}^{(i)}) = \mathbf{f}(\mathbf{x}^{(i)})$ (zero-order consistency).

The term $\mathbf{q}^{(i)}$ can be calculated using (4.18). In place of solving (4.17), $\mathbf{A}^{(i)}$ and $\mathbf{D}^{(i)}$ can be obtained analytically as [28]

$$\begin{bmatrix} {a_m}^{(i)} \\ {d_m}^{(i)} \end{bmatrix} = (\mathbf{C}_m^T \mathbf{C}_m)^{-1} \mathbf{C}_m^T \mathbf{F}_m, \quad \begin{bmatrix} {a_s}^{(i)} \\ {d_s}^{(i)} \end{bmatrix} = (\mathbf{C}_s^T \mathbf{C}_s)^{-1} \mathbf{C}_s^T \mathbf{F}_s, \tag{4.19}$$

where ($f$ refers to the high-fidelity model $\mathbf{f}$)

$$\begin{aligned} \mathbf{C}_m &= \begin{bmatrix} c_m(\mathbf{x}^{(0)}) & c_m(\mathbf{x}^{(1)}) & ... & c_m(\mathbf{x}^{(i)}) \\ 1 & 1 & ... & 1 \end{bmatrix}^T, \\ \mathbf{F}_m &= \begin{bmatrix} f_m(\mathbf{x}^{(0)}) & f_m(\mathbf{x}^{(1)}) & ... & f_m(\mathbf{x}^{(i)}) \end{bmatrix}^T, \end{aligned} \tag{4.20}$$

$$\begin{aligned} \mathbf{C}_s &= \begin{bmatrix} c_s(\mathbf{x}^{(0)}) & c_s(\mathbf{x}^{(1)}) & ... & c_s(\mathbf{x}^{(i)}) \\ 1 & 1 & ... & 1 \end{bmatrix}^T, \\ \mathbf{F}_s &= \begin{bmatrix} f_s(\mathbf{x}^{(0)}) & f_s(\mathbf{x}^{(1)}) & ... & f_s(\mathbf{x}^{(i)}) \end{bmatrix}^T, \end{aligned} \tag{4.21}$$

which is a least-square optimal solution to the linear regression problem $\mathbf{C}_m a_m^{(i)} + d_m^{(i)} = \mathbf{F}_m$ and $\mathbf{C}_s a_s^{(i)} + d_s^{(i)} = \mathbf{F}_s$, equivalent to (4.17). Note that the matrices $\mathbf{C}_m^T \mathbf{C}_m$ and $\mathbf{C}_s^T \mathbf{C}_s$ are non-singular for $i > 1$. For $i = 1$ only the multiplicative correction with $\mathbf{A}^{(i)}$ components are used, which can be calculated in a similar way.

### 4.5.3 Low-Fidelity Modeling

The low-fidelity model $\mathbf{c}$ needs to be a fast representation of the high-fidelity one $\mathbf{f}$. Most commonly, this is done in two different ways. Firstly, the high- and low-fidelity models can use separate sets of governing equations. For example, the high-fidelity CFD model could use RANS equations while the low-fidelity model uses Euler, VLM, or even analytical equations, all of which are faster yet less accurate than RANS equations. Secondly, the two models can use the same governing equations while using different meshes and flow solver convergence criteria. In this work, the latter approach is taken. Section 5 provides further details about the specific low-fidelity models utilized in this work.

# CHAPTER 5. NUMERICAL APPLICATIONS

This chapter presents results of applying the optimization methods described in Chapter 4 to the design of dual-rotor wind turbines (DRWTs). In particular, three test cases of varying cost and complexity are examined; after this, the most complex case is extended to obtain a range of optimal designs rather than single designs. Then, for comparison, the results using low-fidelity models are compared to those using the high- and multi-fidelity approaches. Lastly, the advantages and disadvantages are discussed in the final section.

In the process of these investigations, two similar but distinct approaches are used. These approaches differ in the objective function used as well as by the computational discretization used by the CFD model. In the first approach, a moderately fine mesh (converged to within 1% of the extrapolated power coefficient) is selected and the sum of power coefficients of the two rotors is maximized. The second, more recent approach makes use of a finer mesh (converged to approximately 0.001 in terms of power coefficient absolute value) and the increase in power coefficient with respect to the main rotor operating in isolation is maximized.

## 5.1 Problem Description

This section provides the objective functions for the different approaches, and also describes which optimization methods utilize these objective functions. Additionally, constraint functions and design variable bounds for each test case are provided.

### 5.1.1 Objective Function 1: Combined Power Coefficient

For DRWTs, we want to maximize the combined power capture from both rotors, therefore the objective function used for the first approach to this problem is $H(\mathbf{f}(\mathbf{x})) = -(P_m + P_s)/\left(\frac{1}{2}\rho_\infty {v_\infty}^3 A\right)$, where $P_m = \tau_m \Omega_m$ and $P_s = \tau_s \Omega_s$ are power captured by the rotors, $\tau_{m,s}$

and $\Omega_{m,s}$ are rotor torque and angular speed, respectively, and $A$ is the area swept by the main rotor. This is implemented by simply summing the computed power coefficients for each rotor ($C_{P_m}$ and $C_{P_s}$). When using this objective function, the stopping criterion for all optimization methods is simply when the number of function calls exceeds 50.

When using this objective function, all methods except the output space mapping (OSM) approach are used (i.e. parametric sweep, pattern search, SAO, and kriging with infill). The reason for this is based on the mesh used for this approach. For this approach, only a moderately fine mesh is used. This means that unless an extremely coarse mesh is used as a low-fidelity model, the simulation time ratio between the high- and low-fidelity models would most likely be too small to achieve efficient results. The responses of the models would likely be very similar as well, so corrections to the low-fidelity model would be insignificant. Thus, a multi-fidelity approach would not be a logical choice for this particular scenario.

### 5.1.2 Objective Function 2: Relative Power Increase

In this revised approach, the focus is shifted to maximizing the increase in power capture when compared to a single-rotor wind turbine. The objective function is then $H(\mathbf{f}(\mathbf{x})) = -100(C_{P_m} + C_{P_s} - C_{P_0})/C_{P_0}$, where $C_{P_0}$ is dependent on the mesh used. The magnitude of this objective function value is therefore equivalent to the percent increase in power capture compared to the main rotor operating in isolation. When using this objective function, the stopping criteria for all of the methods are based on the number of function calls as well as the change in $H(\mathbf{f}(\mathbf{x}))$ between consecutive iterations.

While all direct and surrogate-based optimization methods are tested using this objective function, the parametric sweep approach is not. This is because the grid used for this newer approach is more costly than the previous mesh, so a full-factorial sampling of the design space would, in the author's opinion, be too computationally expensive for what there is to gain from it. Additionally, when using kriging, only IC1 is used. This is due to two reasons. Firstly, when using the LHS sampling plan in this work, the expected improvement function is often greatest at the boundaries of the domain. These additional samples are costly and do not contribute

model accuracy in the vicinity of the optimum. More importantly, the objective function stopping criterion does not work well for this infill criteria; often, it would stop prematurely.

When using OSM, the choice of model response vectors ($\mathbf{c}$ and $\mathbf{f}$ in Sec. 4.5.2) can affect the final outcome. In this work, these low- and high-fidelity CFD model response vectors are

$$\begin{aligned} \mathbf{c}(\mathbf{x}) &= [c_m \quad c_s]^T = \left[100\left(\frac{C_{P_m,c} - C_{P_0,c}}{C_{P_0,c}}\right) \quad 100\left(\frac{C_{P_s,c}}{C_{P_0,c}}\right)\right]^T \\ \mathbf{f}(\mathbf{x}) &= [f_m \quad f_s]^T = \left[100\left(\frac{C_{P_m,f} - C_{P_0,f}}{C_{P_0,f}}\right) \quad 100\left(\frac{C_{P_s,f}}{C_{P_0,f}}\right)\right]^T \end{aligned} \tag{5.1}$$

so the objective function for OSM is then $H = -(c_m + c_s)$ and $H = -100(f_m + f_s)$ for the low- and high-fidelity model responses, respectively. In these expressions, the subscripts $m$ and $s$ indicate a value for the main and secondary rotors. Similarly, $c$ and $f$ indicate values from the coarse and fine (low- and high-fidelity) models.

### 5.1.3 Design Variables and Constraints

Numerous properties of the model could be treated as design variables. For example, airfoil selection, chord length, and blade twist at different spanwise stations all impact wind turbine power capture. However, these properties will be kept fixed for the first two cases. For the main rotor, these geometric properties are based on the NREL 5MW offshore wind turbine [31], while the secondary rotor geometry was determined using an inverse design analysis described by Rosenberg *et al* [1]. In Case I, the design variables will be restricted to the tip speed ratio ($\lambda_2$) and radius ($R_2$) of the secondary rotor, while the axial separation between the rotors ($\Delta x$) is fixed at 0.5 (see Fig. 3.1). The design variable vector for Case I is written as $\mathbf{x} = [\lambda_2 \; R_2]^T$. In Case II, $\Delta x$ is added as a design variable so the design variable vector becomes $\mathbf{x} = [\lambda_2 \; R_2 \; \Delta x]^T$. For both cases, there are no inequality or equality constraints, only design variable bounds.

Table 5.1: Design variables bounds and constraints used for each case.

| | $\lambda_2$ | $R_2$ | $\Delta x$ | $c$ | $g(\mathbf{x})$ |
|---|---|---|---|---|---|
| Case I | [0,12] | [0,0.8] | 0.5 | fixed | none |
| Case II | [0,12] | [0,0.8] | [0.05,0.8] | fixed | none |
| Case III | [0,12] | [0,0.8] | [0.05,0.8] | [0.001,0.2] | (5.2) |

In the third test case, the design space is further expanded to include the geometric shape of the secondary rotor. This is done by fitting a b-spline curve to the initial chord distribution of the secondary rotor and subsequently optimizing the vertical positions of the control points. Further details about this parameterization are provided in Sec. 5.4. For all three cases, the upper and lower bounds of the design variables are give in Table 5.1. Note that in this table, $c$ represents b-spline control point coordinates which control the spanwise chord distribution of the secondary rotor. Thus, these design variables $c$ should not be confused with the low-fidelity model response $\mathbf{c}$ seen previously.

Unlike Cases I and II, which are solved using no constraints, the design space in Case III includes non-physical designs. Such a scenario can occur when $c$ becomes large enough, and $\Delta x$ small enough, for the rotor blades (of the main and secondary rotors) to overlap. Therefore, a nonlinear constraint must be implemented when solving the problem. The expression used for this constraint is

$$g_1(\mathbf{x}) = \frac{1}{2}\underset{r}{\text{argmax}}\left(c_m \cos\theta_m + c_s \cos\theta_s\right) + \Delta_{min} - \Delta x \quad \text{for} \quad r \in [0, R_2], \tag{5.2}$$

where $c$ and $\theta$ are chord and twist values for the spanwise station $r$, and $\Delta_{min}$ is a predetermined constant which describes the minimum desired spacing between the trailing and leading edges of the secondary and main rotors, respectively. In this work, this value is arbitrarily set to a small but nonzero value of 0.001, simply to ensure that the rotors are not flush with one another.

### 5.1.4 Model Setup

Four different meshes are used throughout this work. The first approach utilizes a moderately fine mesh to maximize the total power coefficient. This mesh was selected because it yields a power coefficient within an estimated 1% of the actual value. However, when setting up the multi-fidelity optimization cases, it became clear that a more diverse range of meshes may be needed to efficiently obtain a high-quality design. Thus, the grid dependence study was ultimately revisited in search of a more effective low-fidelity model for this application.

While the results presented in Fig. 3.3 varied only the most influential parameter, the cell growth rate in the stream-wise direction, the newer analysis varied an additional parameter. This added parameter is the number of cells in the stream-wise direction within each rotor. Because the mesh blocks containing the rotors are of fixed width, this parameter is inversely proportional to the minimum cell width in the stream-wise direction. Exponential refinement of this parameter is done as axial cell growth rate is reduced incrementally, giving a 4x5 grid of power coefficients for the same design point used in the initial grid study. In the radial direction, cell growth rates are kept fixed while the smallest cell width, which is located at the main rotor tip, is kept fixed at five times the minimum cell width in the axial direction.

The results of this more detailed mesh convergence study are provided in Fig. 5.1. As a result of the analysis, a high-fidelity mesh (HF), which is converged in terms of combined power coefficient to approximately $10^{-3}$, is selected. For the low-dimensional cases, HF is used as the high-fidelity model and LF 1 as the low-fidelity model when using space mapping. For the medium dimensional case, this work investigates two separate cases which utilize LF 1 and LF 2 respectively as the low-fidelity model.

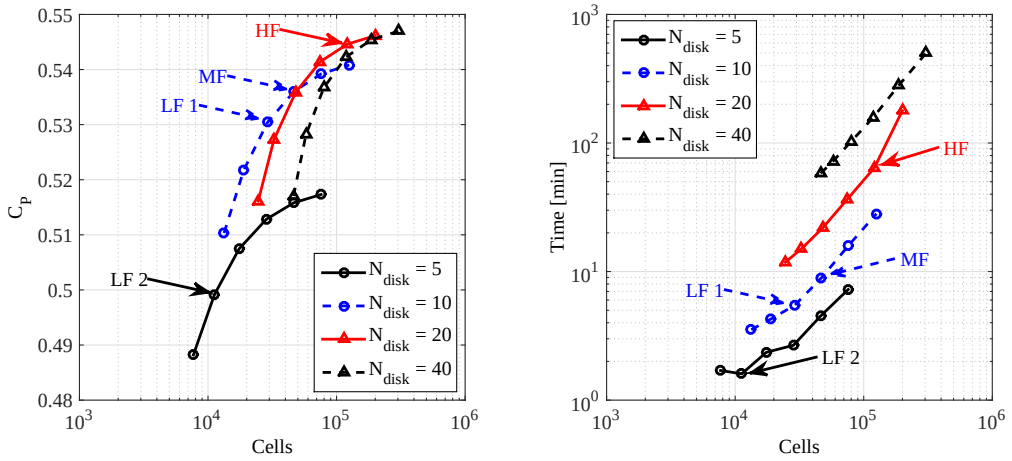



Figure 5.1: Results of the secondary mesh independence study: power coefficients (left), simulation run time using three processors (right). MF is the medium-fidelity mesh used in first approach. HF is the high-fidelity model used in the second approach. LF 1 and LF 2 are the low-fidelity models used by multi-point OSM in the second approach.

The purpose of using two separate low-fidelity models is to investigate the effects of different coarse models on the quality of the solution, as well as computational cost of the optimization process. In addition to using separate meshes, residual stopping criteria are adjusted as well. Based on typical convergence rates of power coefficient, residual stopping criteria are relaxed for the coarse models by two orders of magnitude, which approximately halves the number of flow iterations required. When one processor is used, the simulation time ratios of LF 1 and LF 2 with respect to HF are around 25 and 60, respectively. The number of cells, computed power coefficient values, and simulation time for these models are provided in Table 5.2.

The dimensions of the mesh topology change with rotor separation, which may affect the baseline power coefficient value $C_{P_0}$. Thus, it is worthwhile to investigate the effect of rotor separation on this baseline value, which is found by setting radius and tip speed ratio of the secondary rotor equal to zero. Values of $C_{P_0}$ for a range of rotor separation values were found for each of the four meshes shown in the grid convergence results (Fig. 5.1). Based on these results, which are shown in Fig. 5.2(a), grid dependence due to rotor separation clearly reduces as the grid is refined (i.e., $C_{P_0}$ for HF is approximately constant). Therefore, optimizing the high-fidelity model using both absolute power coefficient value as well as the value relative to the baseline turbine may yield the same solution. However, when considering the low-fidelity models, grid dependence of the baseline value may have a large effect on where the optimum would lie in the design space. Regardless, when computing the objective function for each high- and low-fidelity model, $C_{P_0}$ is taken to be the mean value of $C_{P_0}(\Delta x)$ in this figure. For reference, the computed pressure contours and streamlines are provided in Fig. 5.2(b). In this figure, the flow solution was obtained using HF.

Table 5.2: Combined power coefficient, number of cells, and run time using the meshes HF, MF, LF 1, and LF 2. $C_P$ is the power coefficient for the grid study design point, $N_{cells}$ is the number of mesh cells, and $t$ is the simulation run time using 16 processors.

| | HF | MF | LF 1 | LF 2 |
|---|---|---|---|---|
| $C_P$ | 0.54462 | 0.53608 | 0.53062 | 0.49925 |
| $N_{cells}$ | 121,218 | 46,864 | 29,088 | 11,248 |
| $t$ (min) | 12.681 | 2.411 | 1.340 | 0.744 |

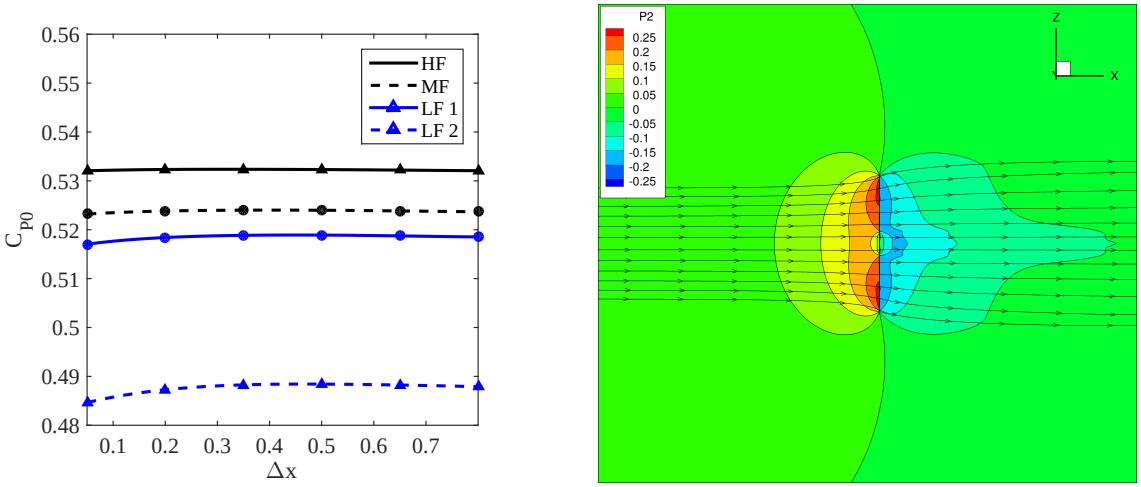



Figure 5.2: Performance and flow field of the baseline design: grid dependence of the baseline power coefficient value for each mesh (left), pressure contours and streamlines (right).

## 5.2 Case I: Optimizing $R_2$ and $\lambda_2$

In this section, the size and rotational rate of the secondary rotor are optimized. In other words, the design variables will be restricted to the tip speed ratio ($\lambda_2$) and radius ($R_2$) of the secondary rotor, while the axial separation between the rotors ($\Delta x$) is fixed at 0.5 (see Fig. 3.1). The design variable vector for this two-parameter case is written as $\mathbf{x} = [\lambda_2 \; R_2]^T$. As given in Table 5.1, the design variable bounds are chosen to be $0 \leq \lambda_2 \leq 12$ and $0 \leq R_2 \leq 0.8$. In the following sections, this two-parameter case is solved using both approaches described in Sec. 5.1.

### 5.2.1 Maximization of Combined Power Coefficient (Case I)

For Case I, the parametric sweep was done by uniformly sampling a $12 \times 12$ array of grid points in the design space. The axial separation selected for this case is 0.5. A main rotor tip speed ratio of 8 is used for this study because it corresponds approximately to the maximum energy capture of this turbine if the secondary rotor is removed. The results of this study are provided in Fig. 5.3.

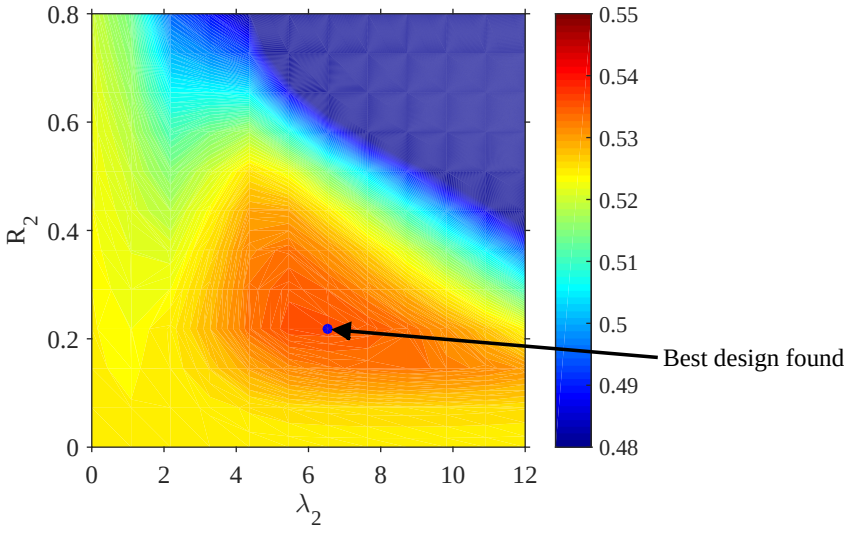



Figure 5.3: Parametric sweep result for Case I: contours of the combined power coefficient of both rotors.

This approach, requiring a total of 144 high-fidelity CFD evaluations, yielded an optimum at $\lambda_2 = 6.55, R_2 = 0.218$ and a power coefficient of 0.536. This value is approximately 99.97% of the maximum value obtained by optimization techniques. However, results from all optimization algorithms obtain higher $C_P$ values in much less time, which is to be expected.

Three initial design points, which are given in Table 5.3, were utilized when using pattern search [18, 19, 43]. One of these points led to zero secondary rotor radius and therefore a single rotor design (Fig. 5.5). The other two initial design points led to the global maximum, which is near $R_2 = 0.2$ and $\lambda_2 = 6$. The results of these optimization studies are presented below in Figs. 5.4, 5.5, and 5.6. The optimum found in the first and third design searches is reproducible by using multiple initial points. Therefore, it is clear that this is the global optimum, and the performance of surrogate-based methods can now be compared to these ideal direct cases.

Table 5.3: Initial points used for pattern search for Case I using objective function 1 (Sec. 5.1.1).

| | $\lambda_2$ | $R_2$ | $\Delta x$ |
|---|---|---|---|
| Trial 1 | 10 | 0.5 | 0.5 |
| Trial 2 | 2 | 0.65 | 0.5 |
| Trial 3 | 6 | 0.6 | 0.5 |

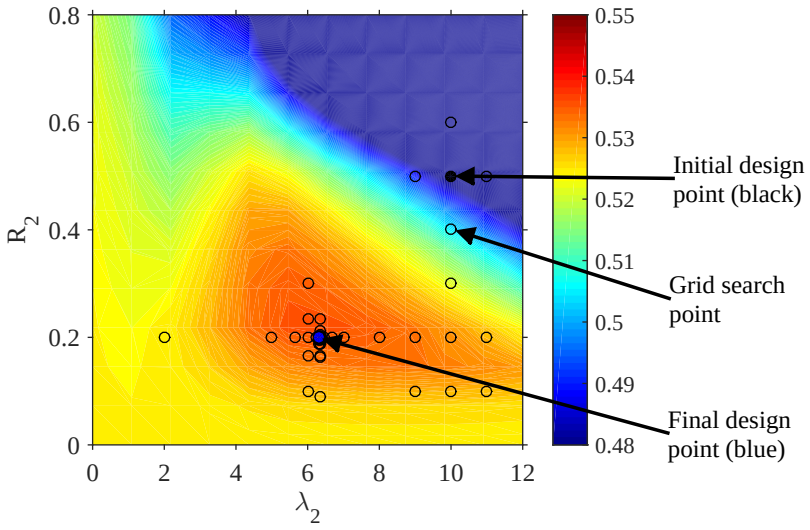



Figure 5.4: Pattern search trial 1: sampled designs with power contours as a backdrop.

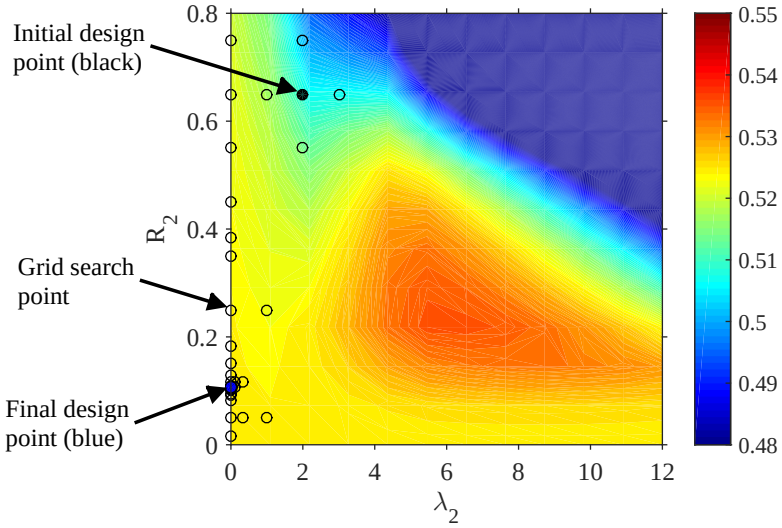



Figure 5.5: Pattern search trial 2: sampled designs with power contours as a backdrop.

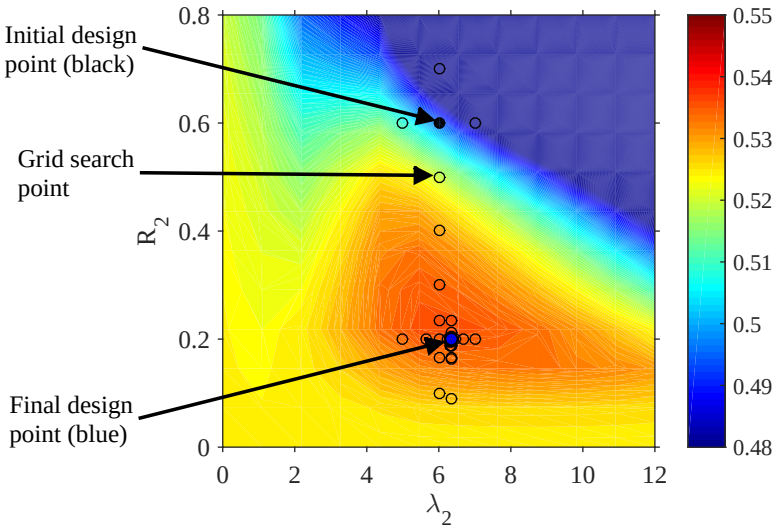



Figure 5.6: Pattern search trial 3: sampled designs with power contours as a backdrop.

Optimization using SAO (Sec. 4.4.2) yielded results comparable to pattern search (Sec. 4.3). However, this method immediately appears to have one advantage; it tends to move diagonally in the design space more readily, leading to a faster rate of convergence. The resulting design points sampled by this method are presented in Fig. 5.7. The convergence metrics are given in Fig. 5.8.

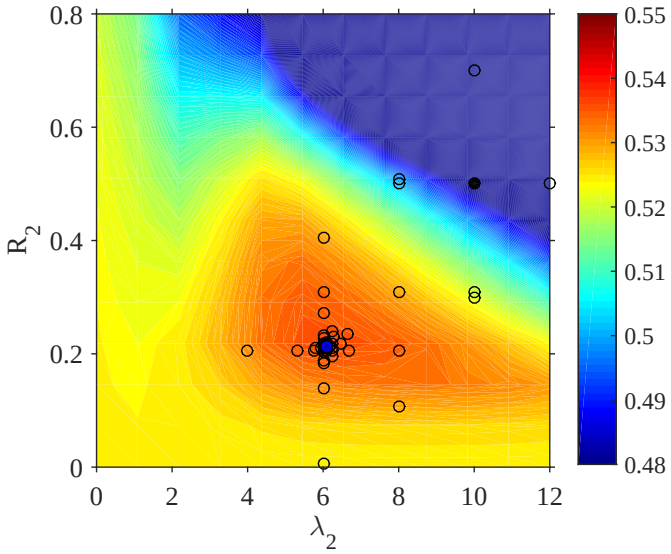
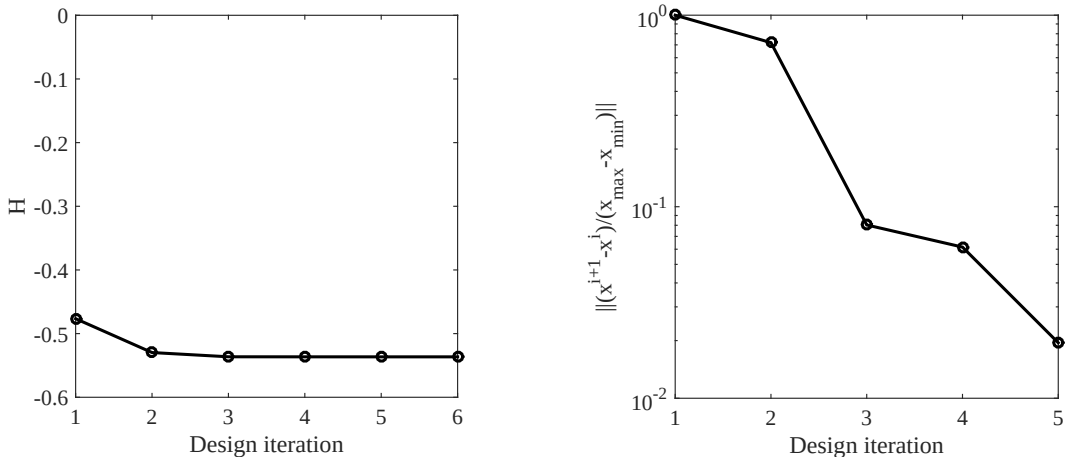



Figure 5.7: Contours of $C_P$ and the sampled design points using SAO for Case I.

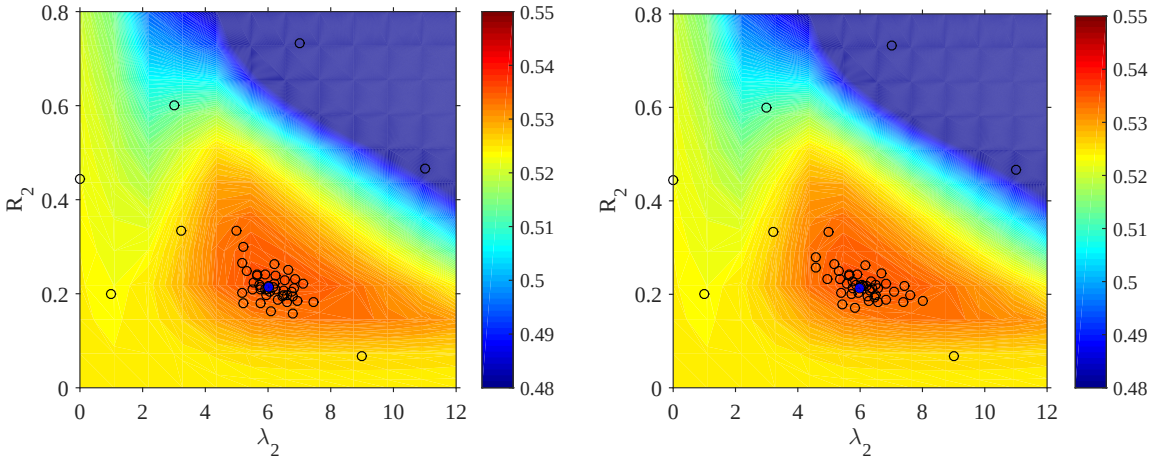



(a) Objective function (b) Norm of the argument

Figure 5.8: Convergence history of the SAO method for Case I using objective function 1.

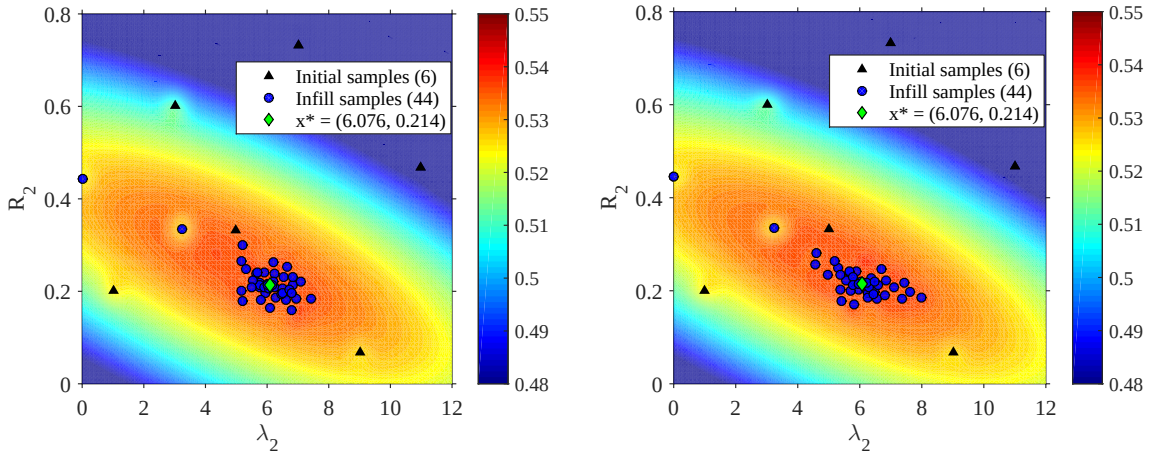
When using kriging-based optimization (Sec. 4.4.1), the final result can depend heavily on the initial sampling plan. Therefore, in order to better compare the two infill criteria IC1 and IC2, which in general represent design exploitation and exploration, respectively, the same initial sampling will be used for both criteria. Note that the formulation of the criteria are given in Sec. 4.4.1. Figure 5.9 shows the results of the kriging-based algorithm using the two infill criteria. In these figures, the initial sampling plan and subsequent update points (50 function evaluations in total) are shown with the parametric sweep results as a backdrop.

To some extent, the two criteria can be qualitatively assessed using the final surrogate models as well as all sampled design points. These final models and their corresponding sampled points are presented in Fig. 5.10, which indicate that both infill criteria are able to quickly find the region corresponding to maximum power. Additionally, they result in very similar updates for many iterations until small differences accumulate. However, it is clear that IC2 results in a more spread out distribution of samples near the optimum, while IC1 is more likely to add updates very close to existing samples. This result is somewhat intuitive, as EI can often increase as a function of the predicted model error of the surrogate, which is higher in gaps between sampled points.




(a) IC1 (b) IC2

Figure 5.9: Kriging with updates for Case I using objective function 1 (Sec. 5.1.1): contours of $C_P$ and the sampled points.




(a) IC1 (b) IC2

Figure 5.10: Final kriging models for Case I using objective function 1 (Sec. 5.1.1): contours of the kriging-approximated $C_P$ and the sampled points.

Table 5.4: Summary of results for Case I using objective function 1 (Sec. 5.1.1).

| | Pattern search | SAO | Kriging (IC1) | Kriging (IC2) |
|---|---|---|---|---|
| $\mathbf{x}_0$ | $[10\ 0.5]^T$ | $[10\ 0.5]^T$ | - | - |
| $x_1*$ | 6.07407 | 6.08308 | 6.01553 | 5.97475 |
| $x_2*$ | 0.21481 | 0.21341 | 0.21320 | 0.21392 |
| $H$ | -0.536504 | -0.536507 | -0.536505 | -0.536503 |
| Cost[a] | 14 | 11 | 8 | 8 |

[a] Function calls needed to reach 99.9% of optimum.

Unfortunately, convergence seems to be slow for both kriging update criteria. This may be due to factors such as the flatness of the design space near the optimum, as well as the unavoidable numerical noise in the CFD model. To counteract these factors, the model may benefit from sampling a separate set of design points, which could be used to find better model constants in the kriging formulation, particularly in the vicinity of the optimum. Another approach that may improve convergence is to implement a variation of the trust region concept [21, 60].

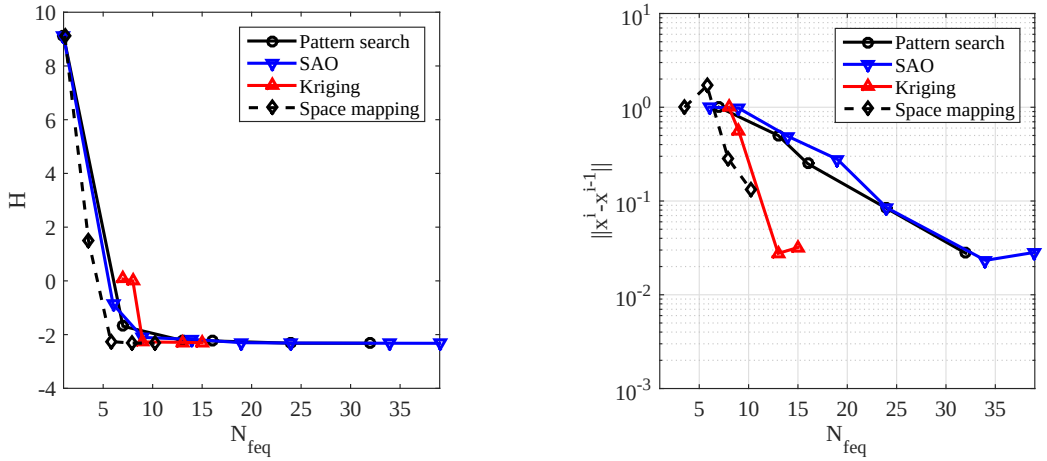
The final results of the different methods for Case I are listed in Table 5.4. While they all give approximately the same final design after 50 function evaluations, some methods are better than others. Specifically, kriging appears to be the most efficient while SAO is slightly more costly but converges more reliably.

### 5.2.2 Maximization of Relative Power Gain (Case I)

This second approach uses a finer grid, a more meaningful objective function, and more realistic stopping criteria for the optimization methods. In addition, the variable-fidelity method is also used and compared to the methods used in the initial approach.

Figure 5.11 shows convergence metrics of all methods used in this study. The stopping criteria used for this case are $\left|H^{(i)} - H^{(i-1)}\right| < 0.01$ or $N_f > 40$, where $H^{(i)}$ is the objective function at design iteration $i$, and $N_f$ is the number of high-fidelity function calls. When using kriging, the objective function stopping criterion is applied to the best design found so far, rather than on the objective function at each sampled design. Among these results, the best optimized design $\mathbf{x}^* = [6.13\ 0.213]^T$ was obtained after 40 function calls using SAO. At this

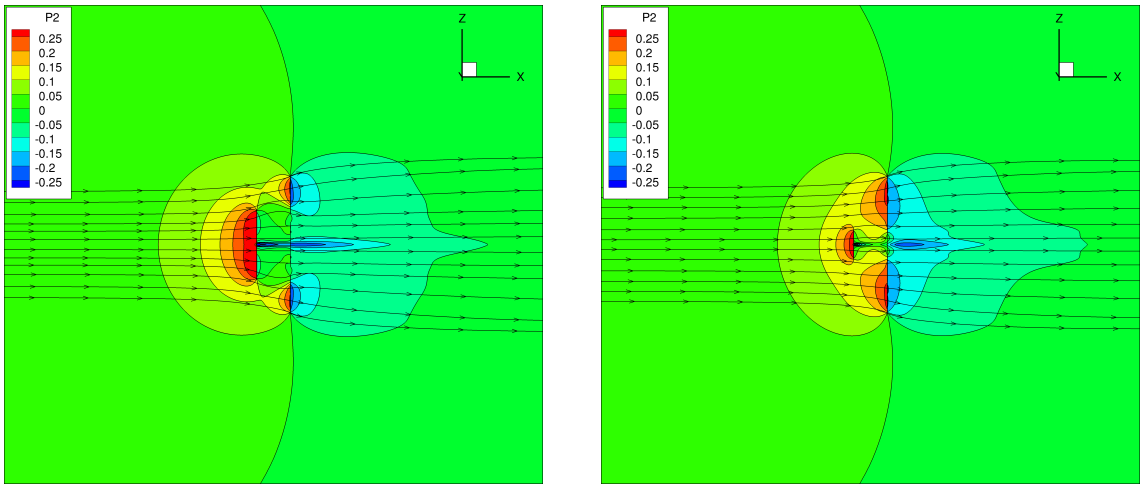
design point, the computed relative power increase with respect to the baseline value is 2.3%. For reference, the computed pressure contours and streamlines for the initial and final designs are provided in Fig. 5.12.




(a) Objective function

(b) Norm of the argument

Figure 5.11: Convergence histories for Case I using objective function 2 (Sec. 5.1.2), where the solutions for both models were computed using 16 processors. $N_{feq}$ is the equivalent high-fidelity function calls in terms of computation time. Norm of the argument is normalized by the initial magnitude, and each of its dimensional components are normalized by bounds size.




(a) Initial

(b) Optimized

Figure 5.12: Pressure contours and streamlines of the initial and optimized designs for Case I using objective function 2 (Sec. 5.1.2).

Optimized design variable and objective function values, as well as the number of low- and high-fidelity function calls are provided in Table 5.5. Among the direct and data-driven SBO methods, kriging is by far the fastest to converge, although the final design point is slightly worse in comparison to the rest. However, based on convergence of objective function $H(\mathbf{f}(\mathbf{x}))$ with respect to equivalent high-fidelity function calls $N_{feq}$, this method appears to approach the optimum only slightly faster than SAO, with pattern search not far behind. Meanwhile, space mapping is much faster than all other methods, reaching near the optimum in approximately 6 equivalent high-fidelity function calls. Note, however, that because the simulation time ratio between high- and low-fidelity models is affected significantly by the number of processors used (which in this case is 16), $N_{feq}$ can vary greatly. Thus, because the time ratio tends to increase as the number of processors is reduced, using fewer processors would lead to a larger speedup in terms of equivalent high-fidelity function calls. For example, this case was solved with space mapping using both 16 and 6 processors, which converged to the same optimum with final $N_{feq}$ values of 10.2 and 7.3, respectively.

Table 5.5: Summary of results for Case I using objective function 2 (Sec. 5.1.2) (stopping criteria: $\left|H^{(i)} - H^{(i-1)}\right| < 0.01$ or $N_f > 40$). $N_c$ and $N_f$ are the number of low- and high-fidelity function calls, while $N_{feq}$ is the number of equivalent high-fidelity function calls (in terms of computation time).

| | PS | SAO | Kriging | OSM (LF 1) |
|---|---|---|---|---|
| $\mathbf{x}_0$ | $[10\ 0.5]^T$ | $[10\ 0.5]^T$ | - | $[10\ 0.5]^T$ |
| $x_1^*$ | 6.400 | 6.133 | 5.925 | 6.267 |
| $x_2^*$ | 0.1978 | 0.2125 | 0.2268 | 0.2067 |
| $H$ | -2.322 | -2.324 | -2.302 | -2.322 |
| $C_P$ | 0.54458 | 0.54459 | 0.54447 | 0.54458 |
| $N_c$ | - | - | - | 47 |
| $N_f$ | 32 | 40 | 15 | 5 |
| $N_{feq}$ | 32 | 40 | 15 | 10.2 |
| $t_c$ (min) | - | - | - | 69.6 |
| $t_f$ (min) | 435.8 | 535.6 | 202.2 | 66.9 |
| $t_{tot}$ (min) | 435.8 | 535.6 | 202.2 | 136.5 |

## 5.3 Case II: Optimizing $R_2$, $\lambda_2$, and $\Delta x$

In Case II, the axial distance between the rotors $\Delta x$ is added as a design variable with the following bounds: $0.05 \leq \Delta x \leq 0.8$. The design variable vector is now $\mathbf{x} = [\lambda_2 \ R_2 \ \Delta x]^T$. Other parameters and variables are the same as in Case I. In the following sections, this three-parameter case is solved using both approaches described in Sec. 5.1.

### 5.3.1 Maximization of Combined Power Coefficient (Case II)

The parametric sweep approach for Case II was carried out by sampling an $8 \times 8 \times 8$ array of grid points throughout the design space. Given in Fig. 5.13 are power coefficient isocontours in the three-dimensional design space. This particularly expensive analysis (512 samples) resulted in an optimum of $\lambda_2 = 6.86, R_2 = 0.229, \Delta x = 0.157$, and a $C_P$ value of 0.539. This value of power coefficient is 99.2% of the maximum value obtained by optimization techniques. Clearly, this traditional parametric approach is much less efficient than the other methods.

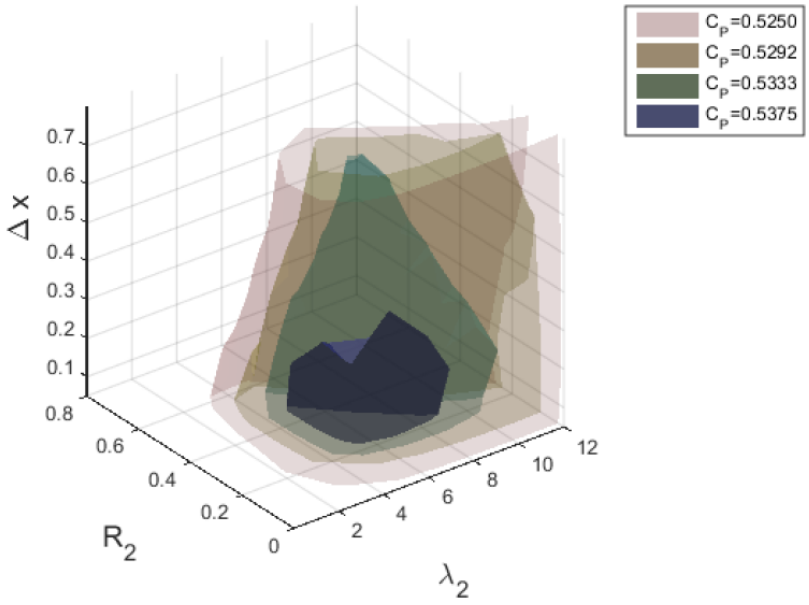
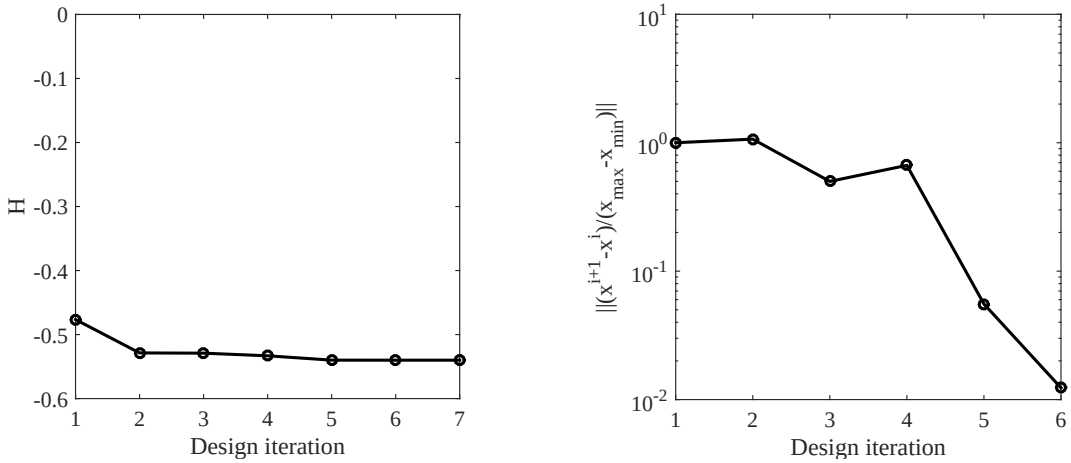



Figure 5.13: Parametric sweep results for Case II using objective function 1 (Sec. 5.1.1): contours of $C_P$.

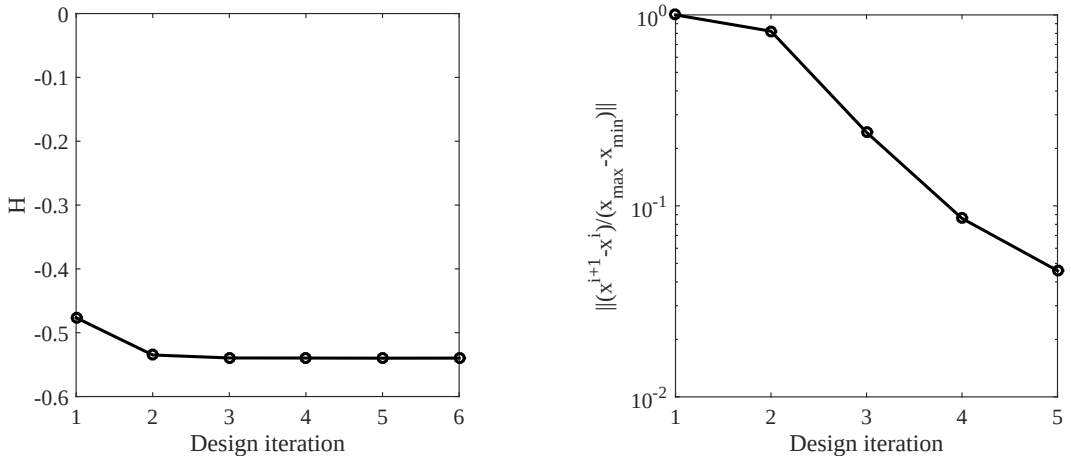
The pattern search method was able to hone in on the optimum point fairly quickly, but converged slowly. Convergence of this method for Case II is provided in Fig. 5.14.

SAO worked fairly well for Case II as well. If a convergence criterion based on $H$ or $\mathbf{x}$ had been used, the method would have converged after just 3 iterations. The results for this case are provided in Fig. 5.15.




(a) Objective function (b) Norm of the argument

Figure 5.14: Convergence of pattern search for Case II using objective function 1 (Sec. 5.1.1).




(a) Objective function (b) Norm of the argument

Figure 5.15: Convergence of SAO for Case II using objective function 1 (Sec. 5.1.1).

Kriging with infill worked fairly well for Case II, locating the region of the optimum immediately after the sampling plan ended. This implies that the initial kriging model, which is found using only ten samples, approximates this design space quite well. Unfortunately, convergence on the optimum is slow when compared to pattern search and SAO. Note, however, that the sampling plan can influence the accuracy of the kriging model. Thus, the initial model's accuracy near the optimum, as well as the convergence of the algorithm, may change significantly depending on the chosen sampling plan. Convergence of this method is provided in Fig. 5.16.

Table 5.6 shows a comparison of results for Case II. In this case, SAO achieved the fastest convergence while both kriging infill criteria resulted in the slowest. However, it is worth noting that the performance of this method depends heavily on the initial sampling plan. This may explain why kriging yielded the best results in Case I but the worst results in Case II.

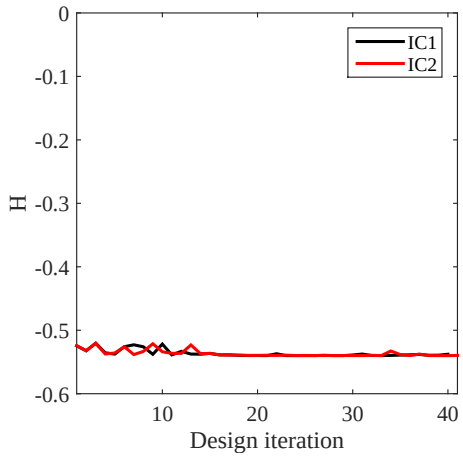



Figure 5.16: Convergence of kriging for Case II using objective function 1 (Sec. 5.1.1).

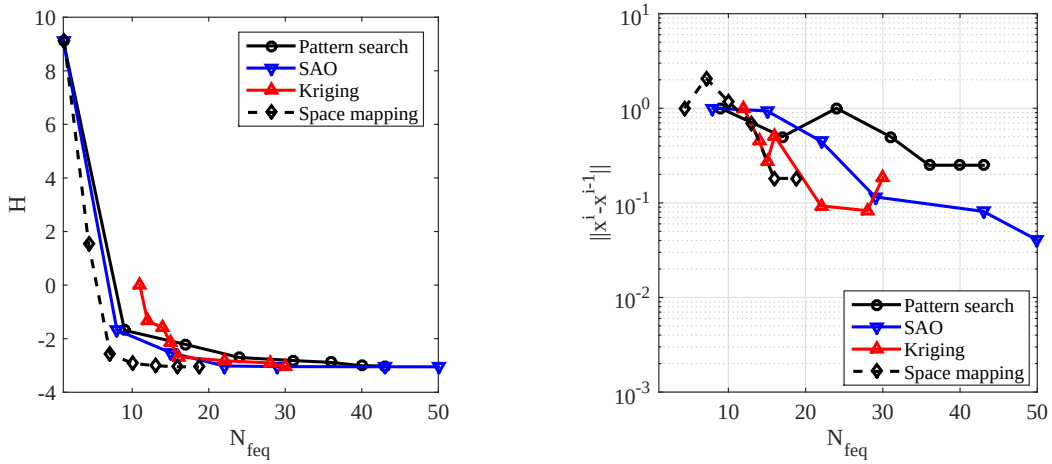
Table 5.6: Summary of results for Case II using objective function 2 (Sec. 5.1.2).

| | Pattern search | SAO | Kriging (IC1) | Kriging (IC2) |
|---|---|---|---|---|
| $\mathbf{x}_0$ | $[10\ 0.5\ 0.5]^T$ | $[10\ 0.5\ 0.5]^T$ | - | - |
| $x_1*$ | 5.92593 | 6.17608 | 5.99375 | 5.92742 |
| $x_2*$ | 0.32222 | 0.30576 | 0.33094 | 0.32509 |
| $x_3*$ | 0.10000 | 0.10912 | 0.11421 | 0.09995 |
| $H$ | -0.540009 | -0.539949 | -0.539962 | -0.540006 |
| Cost[a] | 21 | 15 | 28 | 26 |

[a] Function calls needed to reach 99.9% of optimum.

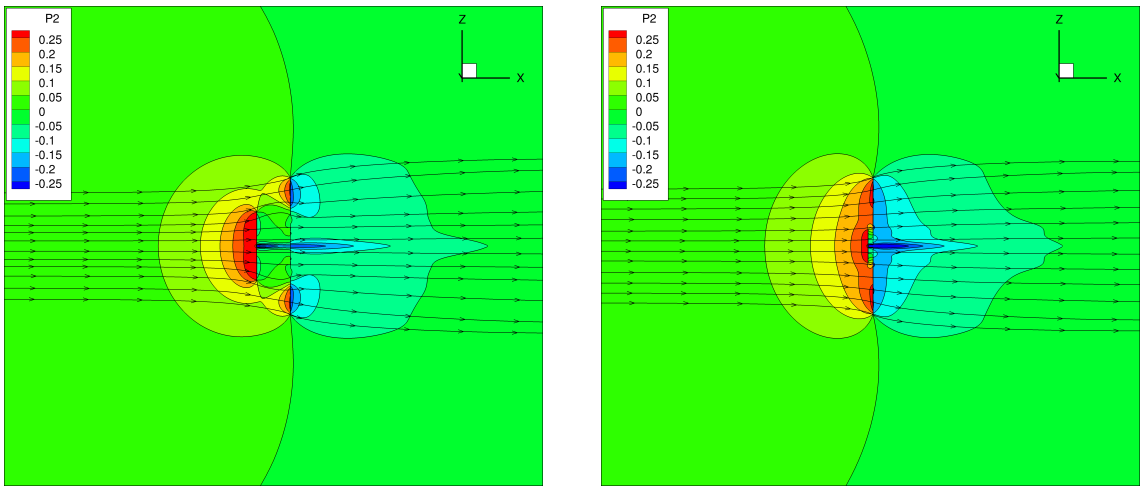
### 5.3.2 Maximization of Relative Power Gain (Case II)

Figure 5.17 shows the convergence histories for all the optimization algorithms, where the stopping criteria are $\left|H^{(i)} - H^{(i-1)}\right| < 0.01$ or $N_f > 50$. The algorithms converge to approximately the same optimized design $\mathbf{x}^* \approx [7.07\ 0.242\ 0.075]^T$. In this case, there are only 3 design variables so the algorithms exhibit similar convergence rate. For reference, the pressure contours and streamlines for the initial and final designs are provided in Fig. 5.18.




(a) Objective function (b) Norm of the argument

Figure 5.17: Convergence histories for Case II using objective function 2 (Sec. 5.1.2).




(a) Initial (b) Optimized

Figure 5.18: Pressure contours and streamlines of the initial and optimized designs for Case II using objective function 2 (Sec. 5.1.2).

Provided in Table 5.7 are the optimized design variable and objective function values, as well as overall costs of the optimization methods. Space mapping again reached the optimum in much less time in comparison to the other methods, and found a better solution as well. For reference, the final value of $N_{feq}$ with space mapping using 6 processors is 12.1, while Table 5.7 shows a value of 18.7, which was obtained using 16 processors. Among the other algorithms, kriging and SAO converge slightly faster than the pattern search method and resulted in approximately the same optimum. Pattern search stopped at a slightly worse solution than the others, and SAO failed to reach the objective function stopping criterion before the maximum function calls was reached, in spite of its quick convergence in the beginning. Convergence of kriging is somewhat unpredictable and depends on the initial sampling plan; however, this disadvantage did not affect performance in this particular case.

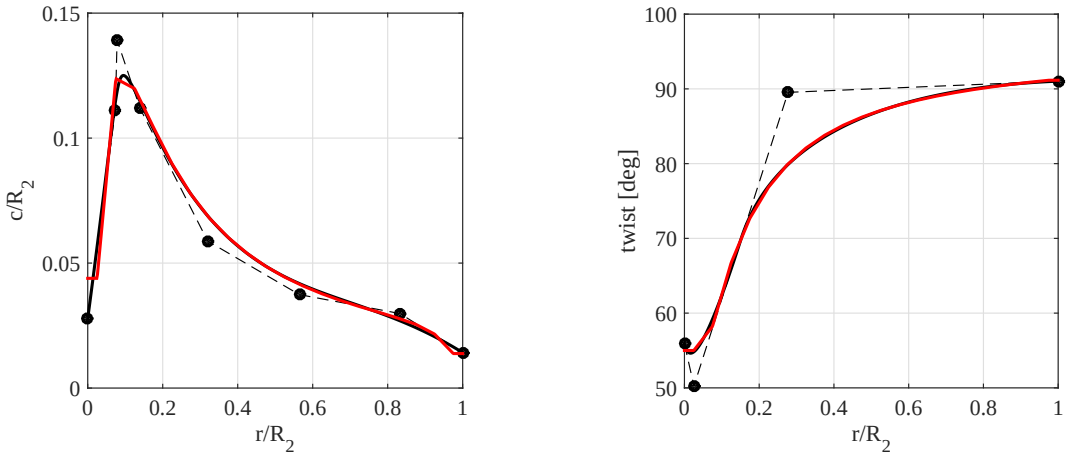
Table 5.7: Summary of results for Case II using objective function 2 (Sec. 5.1.2) (stopping criteria: $\left|H^{(i)} - H^{(i-1)}\right| < 0.01$ or $N_f > 50$). $N_c$ and $N_f$ are the number of low- and high-fidelity function calls, while $N_{feq}$ is the number of equivalent high-fidelity function calls (in terms of computation time).

| | PS | SAO | Kriging | OSM (LF 1) |
|---|---|---|---|---|
| $\mathbf{x}_0$ | $[10\ 0.5\ 0.5]^T$ | $[10\ 0.5\ 0.5]^T$ | - | $[10\ 0.5\ 0.5]^T$ |
| $x_1^*$ | 6.400 | 6.446 | 6.844 | 7.067 |
| $x_2^*$ | 0.2600 | 0.2575 | 0.2481 | 0.2422 |
| $x_3^*$ | 0.1250 | 0.0913 | 0.0841 | 0.0750 |
| $H$ | -3.019 | -3.048 | -3.052 | -3.053 |
| $C_P$ | 0.54828 | 0.54844 | 0.54846 | 0.54847 |
| $N_c$ | - | - | - | 94 |
| $N_f$ | 43 | 50 | 30 | 7 |
| $N_{feq}$ | 43 | 50 | 30 | 18.7 |
| $t_c$ (min) | - | - | - | 142.5 |
| $t_f$ (min) | 546.4 | 628.6 | 387.7 | 85.0 |
| $t_{tot}$ (min) | 546.4 | 628.6 | 387.7 | 227.5 |

It is worth noting the discrepancy between this result (roughly 3% additional power) and the results found by Sharma *et al.* [1]. The authors essentially used the same CFD model as this thesis, but observed a power increase of 5-7% when varying the same design variables used in Case II ($R_2$ and $\Delta x$). By in large, this is most likely due to meshing differences. In this thesis, mesh dependence on computed power was studied extensively (which is necessary when using variable-fidelity approaches), and in an effort to reduce this dependence, the block topology was modified to allow for continuous cell width distributions across the block interfaces (i.e., the cell sizes on the boundaries of adjacent blocks were set equal to one another). Perhaps more important is the overall refinement of the mesh which was done in this thesis; in the process of mesh refinement, the computed value of the relative power increase tends to reduce in value. Based on these results, then, this grid dependence (which is also observed in Ref. [30]) appears to be a potential disadvantage of using the actuator disk model.

## 5.4 Case III: Optimizing $R_2$, $\lambda_2$, and $\Delta x$, as well as the Blade Chord Distribution

Cases I and II used predetermined radial distributions of the blade chord length and twist angle. These distributions were found in a previous study using an inverse design approach which tailored the shape to produce a desired pressure distribution. In Case III, the blade chord distribution of the secondary rotor is added as a designable shape. To do this, the chord distribution of the rotor blade is parameterized using b-spline curves [69]. Figure 5.19 shows the baseline chord and twist distributions used in Cases I and II. In this figure, the red lines show the baseline values used in the CFD model, while the solid black lines show b-spline curves and are controlled using the control points shown. For this case, the vertical positions of these b-spline control points are adjusted by the optimizer in order to obtain the best rotor shape. The initial control point positions, including both horizontal and vertical coordinates, were found by minimizing the L2 norm between both curves. Note, however, that only the chord distribution (Fig. 5.19(a)) is optimized while the twist remains fixed.




(a) Chord (b) Twist

Figure 5.19: Baseline blade chord and twist distributions of the secondary rotor. The red curves are the initial distributions while b-spline curves and control points are shown in black. The chord distribution will be optimized, whereas the twist distribution will be kept fixed. The control points of the chord distribution are allowed to move in the vertical direction, i.e., their radial location is held fixed during the optimization.

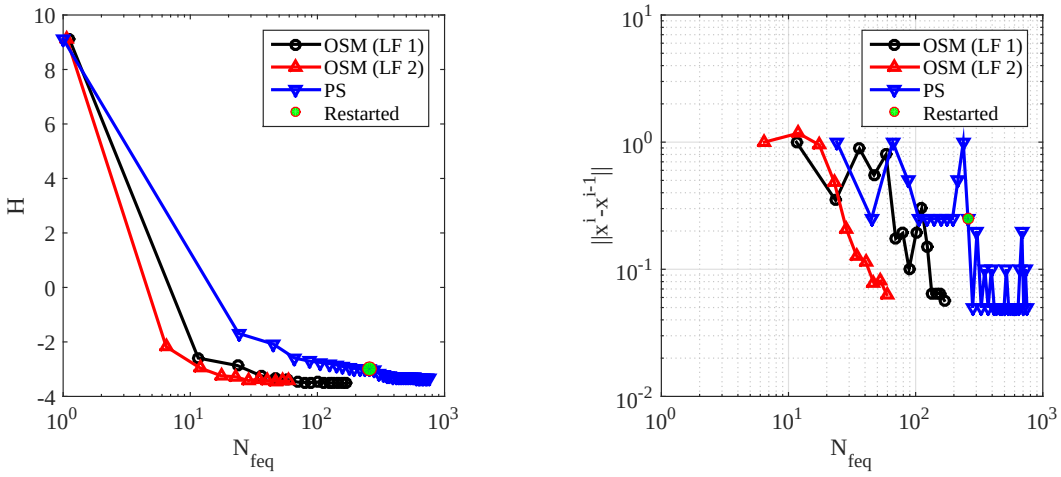
Preliminary sensitivity analyses and optimization trials, which utilized the b-spline parameterization in Fig. 5.19, showed that in this case rotor chord distribution is much more influential than twist angle. Therefore, the twist distribution in this design problem is disregarded, so the design variable vector is then $\mathbf{x} = [\lambda_2 \ R_2 \ \Delta x \ c_1 \ ... \ c_8]^T$, where $c_i$ are the vertical locations of chord control points in chord-radial distance space. The lower and upper bounds for these new design variables are 0.001 and 0.2, respectively.

In this 11-parameter case, the design problem is solved using OSM as well as pattern search for comparison. This decision to not use all methods was made due to the high expected computational cost of the other methods. Also, in each iteration of OSM, the optimization of the surrogate is driven by pattern search using the same initial grid sampling size. Thus, comparison of OSM with pattern search rather than the other methods allows for more meaningful results, as it shows how OSM (or in general, a multi-fidelity optimization framework) can affect convergence, computational expense, and solution quality of a particular optimization algorithm (in this case, pattern search). Regardless, for this case, only the newer approach to the problem which utilizes a finer mesh and a relative objective function (Sec. 5.1.2) is used.

When applying OSM to this case, two different low-fidelity models are utilized in order to identify advantages and disadvantages of using a quicker but less accurate low-fidelity model. Among these two models LF 1 and LF 2 (see Figs. 5.2 and 5.1), LF 1 is slower but models the high-fidelity design space better. In theory, then, using LF 1 should produce a better quality result while also being more computationally expensive.

### 5.4.1 Maximization of Relative Power Gain (Case III)

The convergence metrics for this case are presented in Fig. 5.20. When using OSM, the stopping criteria are $\left|H^{(i)} - H^{(i-1)}\right| < 0.001$ or $N_f > 20$. Clearly, the use of LF 2 results in a much faster solution, but also a slightly poorer quality of solution. This problem may be multi-modal which could cause such a result. However, this discrepancy is most likely due to the failure of LF 2 to adequately resolve the flow through the secondary rotor. It is worth noting, however, that the use of LF 2 in Cases I and II led to very similar results as the use of LF 1.



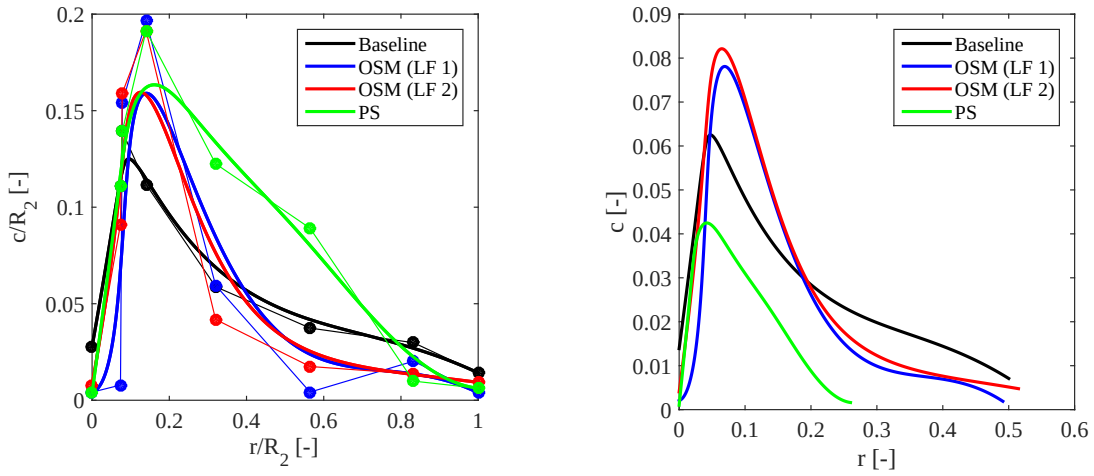

(a) Objective function

(b) Norm of the argument

Figure 5.20: Case III convergence histories using 16 processors for all models.

Compared to OSM, the convergence of pattern search is very costly. When using this direct method, a stopping criterion of $\left|H^{(i)} - H^{(i-1)}\right| < 0.001$ was first used, leading to an optimum of $H = -3.001$ at $N_{feq} = 258$. Because OSM led to a significantly better solution, and because the convergence metric based on $\mathbf{x}$ indicated that the method may not be converged, this method is restarted from the first optimum. For this second trial, the initial grid size is reduced by a factor of 5, and the stopping criterion is changed to $N_{feq} > 500$. In addition to OSM convergence, Fig. 5.20 shows this entire optimization process using pattern search, with the restart point shown as well. Clearly, restarting this algorithm led to a significantly better design; unfortunately, however, it is still not clear whether the method has fully converged.

The final rotor chord distributions are presented in Figure 5.21. While using OSM with LF 1 resulted in a better design than OSM with LF 2, the two chord designs appear to be changing in similar ways. With regard to the baseline shape, in both trials the optimized chord is thicker at the root while the chord near the tip is thinner. Also, both trials resulted in an optimized rotor radius close to the initial value of $R_2 = 0.5$. Meanwhile, pattern search led to a shorter rotor with a significantly different chord distribution. While the inner region of this shorter rotor ($r < 0.2$) appears to be similar to that from OSM, the rotor shape lacks the relatively flat region near the tip of the rotor. Ultimately, this leads to slightly worse performance as computed by the CFD model.




(a) Normalized by tip radius

(b) Actual blade geometry

Figure 5.21: B-spline curves and control points of baseline and optimized chord distributions.

Table 5.8 provides final objective function values, as well as final constraint function values, number of function calls, and run time. Table 5.9 provides the final values of design variables. OSM yields similar tip speed ratios found in Cases I and II, while rotor radii ($R_2 \approx 0.5$) differ significantly from those found in Case II ($R_2 \approx 0.25$). Meanwhile, pattern search converged to a significantly different design. Overall, the best objective function reached is -3.5%, which corresponds to a computed power increase of 3.5% in comparison to the baseline design. The pressure contours and streamlines of the initial and best designs are provided in Fig. 5.22.

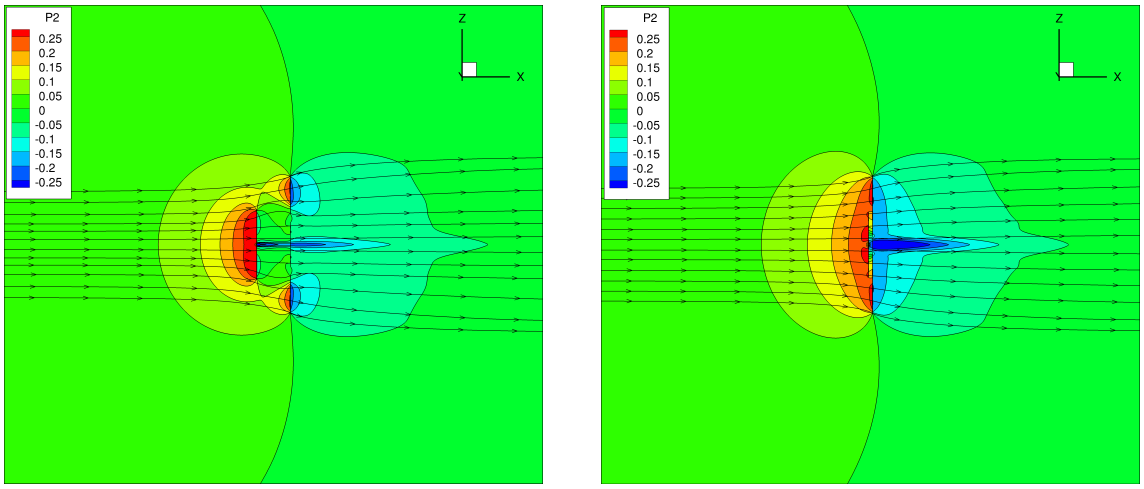
Table 5.8: Summary of results for Case III (stopping criterion: $\left|H^{(i)} - H^{(i-1)}\right| < 0.001$): final attributes.

| | OSM (LF 1) | OSM (LF 2) | PS |
|---|---|---|---|
| $H$ | -3.521 | -3.441 | -3.345 |
| $C_P$ | 0.55096 | 0.55054 | 0.55002 |
| $g$ | -0.00120 | -0.00466 | -0.07602 |
| $N_c$ | 1126 | 731 | - |
| $N_f$ | 16 | 11 | 751 |
| $N_{feq}$ | 168.1 | 59.3 | 751.0 |
| $t_c$ (min) | 1,683.0 | 544.1 | - |
| $t_f$ (min) | 177.0 | 124.0 | 8,445.1 |
| $t_{tot}$ (min) | 1,860.0 | 668.2 | 8,445.1 |

Table 5.9: Summary of results for Case III (stopping criterion: $\left|H^{(i)} - H^{(i-1)}\right| < 0.001$): final design variables. Here, $x_1 = \lambda_2$, $x_2 = R_2$, and $x_3 = \Delta x$, while $x_i (i = 4, ..., 11) = c_j (j = 1, ..., 8)$ are the chord design variables.

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | $x_{10}$ | $x_{11}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{x}_0$ | 10.00 | 0.500 | 0.500 | 0.0278 | 0.1111 | 0.1392 | 0.1118 | 0.0588 | 0.0374 | 0.0298 | 0.0142 |
| SM[a] | 5.20 | 0.491 | 0.057 | 0.0042 | 0.0079 | 0.1540 | 0.1966 | 0.0596 | 0.0042 | 0.0202 | 0.0039 |
| SM[b] | 5.20 | 0.515 | 0.061 | 0.0079 | 0.0912 | 0.1591 | 0.1914 | 0.0419 | 0.0175 | 0.0136 | 0.0093 |
| PS | 4.96 | 0.260 | 0.110 | 0.0039 | 0.1111 | 0.1392 | 0.1914 | 0.1225 | 0.0891 | 0.0099 | 0.0062 |

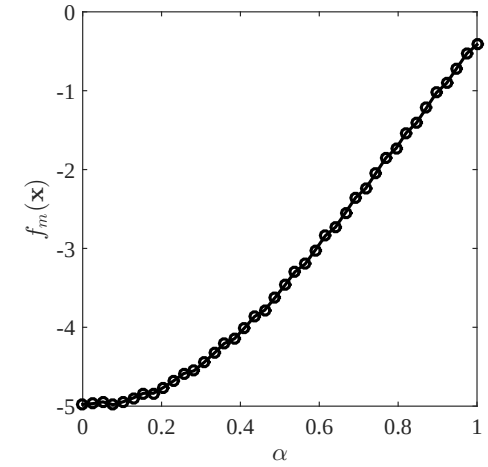
[a] LF 1
[b] LF 2



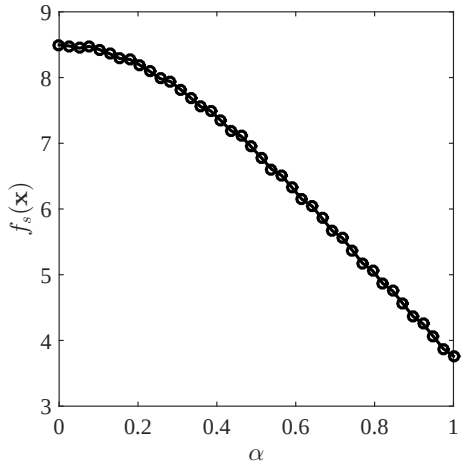

(a) Initial (b) Optimized

Figure 5.22: Pressure contours and streamlines of the initial and final designs for Case III.

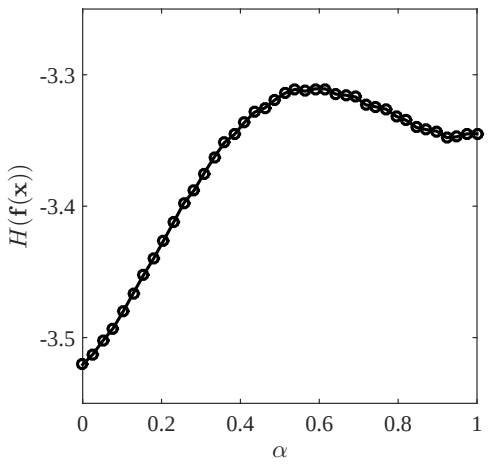
While the two OSM solutions are fairly similar, pattern search led to a completely different design (smaller radius, different chord distribution). To investigate this, the design space is sampled in a linear path between these two optima (which comprises a so-called alpha-plot). Specifically, this additional analysis computes the objective function $H(\mathbf{f}(\mathbf{x}))$ for $\mathbf{x} = \alpha\mathbf{x}_{PS} + (1-\alpha)\mathbf{x}_{OSM}$, where $\alpha$ is varied from 0 to 1 in 40 evenly spaced intervals while $\mathbf{x}_{OSM}$ and $\mathbf{x}_{PS}$ are the two optimized design variable vectors. The results of this analysis are provided in Fig. 5.23. In addition, the values of the nonlinear constraint function (defined in (5.2)) are provided in Fig. 5.24.



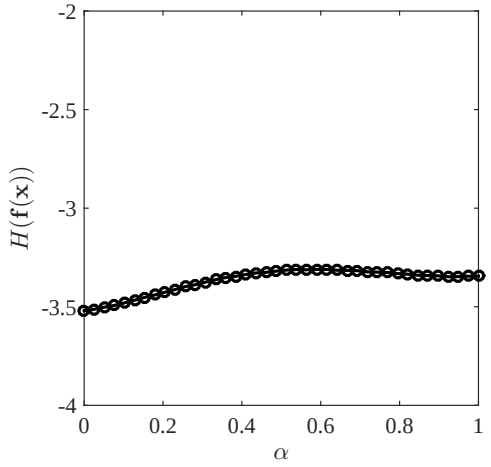

(a) Main rotor component of high-fidelity model response $\mathbf{f}(\mathbf{x})$ $(100(C_{P_m} - C_{P_0})/C_{P_0})$




(b) Secondary rotor component of high-fidelity model response $\mathbf{f}(\mathbf{x})$ $(100C_{P_s}/C_{P_0})$




(c) Objective function $(-f_m - f_s)$




(d) Objective function, wider view

Figure 5.23: Results of design space sampling between the OSM and pattern search optima. Note that $\alpha = 0$ and $\alpha = 1$ correspond to OSM and pattern search optima, respectively.

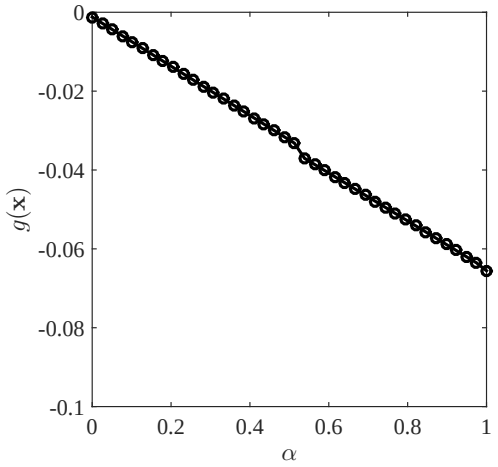



Figure 5.24: Nonlinear constraint value between the OSM and pattern search optima.

When assessing only this single direction through the design space, there appears to be multiple minima. This may explain why pattern search reached a significantly different design. However, the existence of multiple minima is hard to confirm, and would require further investigations. An alternative explanation for this result is that pattern search is simply not converged yet; due to design space nonlinearity, convergence may not be achievable without many more function evaluations. Unfortunately, if this is the case, then the pattern search method would require upwards of 10,000 minutes to reach full convergence.

In addition, this analysis introduces two concerns. Firstly, the model response is not smooth; the cause of this is not well understood, but it is likely due to the CFD model's use of airfoil data lookup tables. Another concern is that pattern search failed to stop at the minimum at approximately $\alpha = 0.9$. Presumably, this level of convergence would be met if the algorithm had been ran for a longer period of time. Regardless, this analysis provides some insight into why pattern search struggles with this case. Also worth noting is the nonlinear constraint values, which shows that the constraint is nearly active at the OSM optimum; this is due to the large chord length at its root, small rotor separation, and relatively large secondary rotor radius.

## 5.5 Additional Investigations

Using the design space from Case III, two additional cases are investigated using the variable-fidelity approach, with each investigation requiring multiple uses of OSM optimization. For the first of these secondary analyses, the 11-parameter case is converted to eight 10-parameter cases, each at a specific secondary rotor radius $R_2$ (i.e., $\mathbf{x} = [\lambda_2 \ \Delta x \ c_1 \ ... \ c_8]^T$). Similarly, in the second of these analyses, eight 10-parameter designs are optimized, each at a specific rotor separation $\Delta x$ (i.e., $\mathbf{x} = [\lambda_2 \ R_2 \ c_1 \ ... \ c_8]^T$). All of these cases use the same initial design as the previous cases. This, in theory, yields the best possible scenario for each value of these parameters. Doing so provides some insight into how influential these parameters are, while the robustness of the optimization algorithm can be qualitatively assessed as well.

### 5.5.1 Optimal Designs for a Range of $R_2$

For this case, eight evenly distributed values of secondary rotor radius are used as fixed parameters. Provided in Fig. 5.25 are the final objective function values for this analysis. The shape of this curve is quite flat between $R_2 = 0.2$ and $R_2 = 0.6$, and has a minimum value near $R_2 = 0.45$. While this figure does not show optimized blade geometries, the chord shapes become progressively thinner at the blade tips as the radius is increased. As a result, for values greater than $R_2 = 0.6$, there is a sharp increase because the chord design variables reached their lower bounds. For these designs, the objective function becomes approximately zero, thus extracting power at the same rate as the main rotor in isolation.

Provided in Fig. 5.26 are the final tip speed ratios and rotor separations from this analysis. While there is some numerical noise in the solution, the optimal tip speed ratio is approximately 5 for radius values of less than 0.6. For larger rotor radii, the tip speed ratio goes approximately to zero. This indicates that the optimizer led to a single rotor design for these large blade lengths. Also worth noting is the rotor separation. For radii of 0.6 or less, the separation remains fixed at around 0.1 (i.e. one tenth of the main rotor radius). For greater values of radius, the rotor separation tends towards 0.3, which is presumably a meaningless numerical artifact.

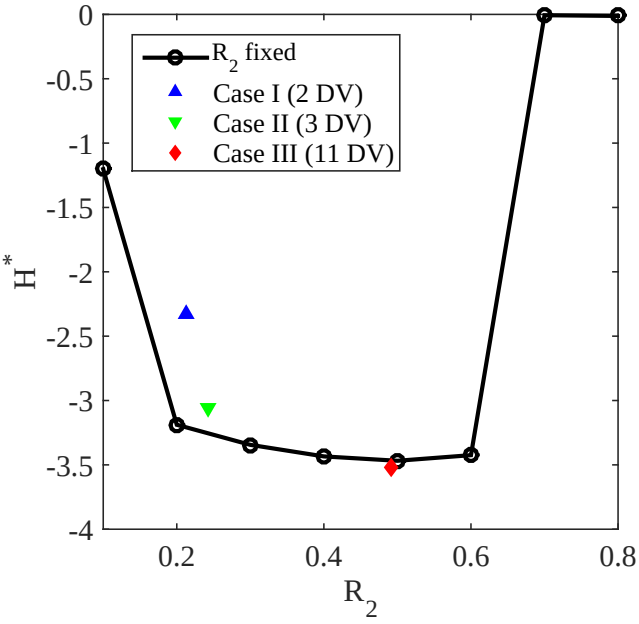
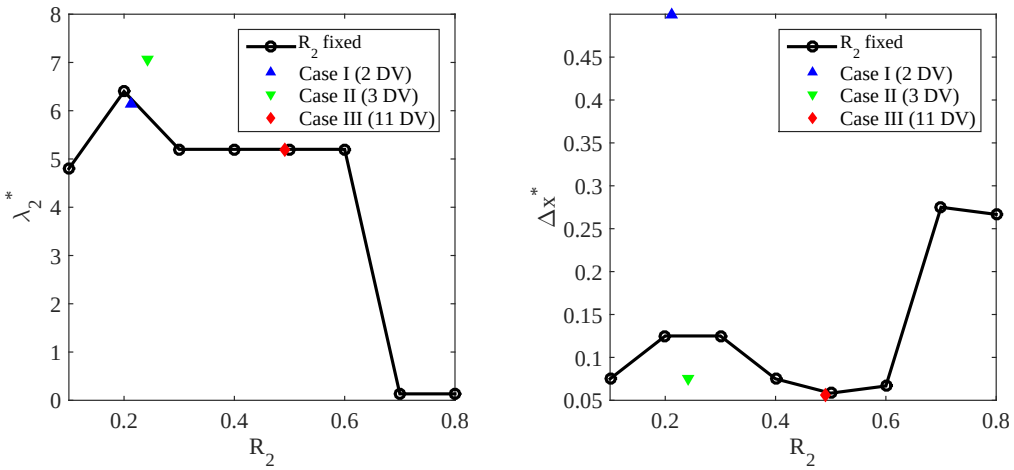



Figure 5.25: Final objective function for fixed radii.




(a) Secondary rotor tip speed ratio

(b) Rotor separation

Figure 5.26: Final design variable values (excluding chord control points) for the fixed $R_2$ investigation.

### 5.5.2 Optimal Designs for a Range of $\Delta x$

This case also uses eight values of the parameter, which is in this case rotor separation. However, the points were not chosen to be evenly distributed because results in previous cases indicate that small $\Delta x$ values tend to yield the highest performance (whereas optimal $R_2$ values have ranged from approximately 0.2 to 0.5). The resulting objective function values from this analysis are provided in Fig. 5.27.

Provided in Fig. 5.28 are the final values of tip speed ratio and radius of the secondary rotor. Compared to the previous investigation, the tip speed ratio remains approximately the same. Meanwhile, the final radius values of the secondary rotor are highly discontinuous. Presumably, this is due purely to numerical reasons and not due to the shape of the design space. However, the complexity of the design space may also play a role.

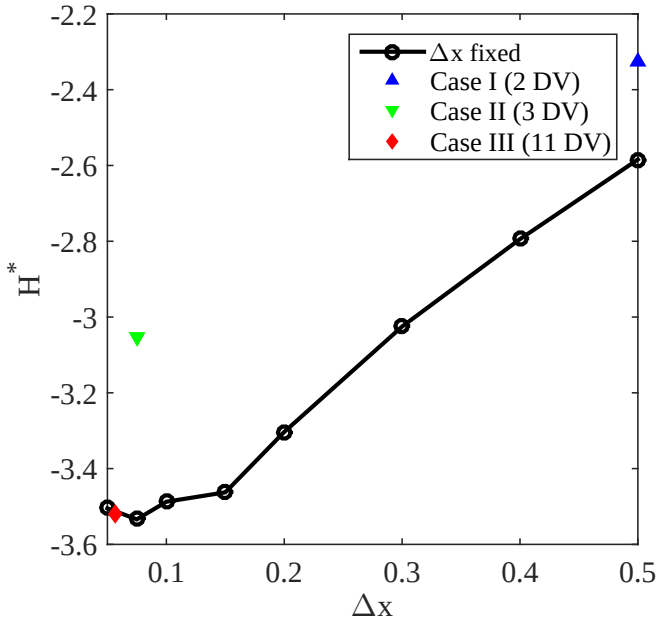
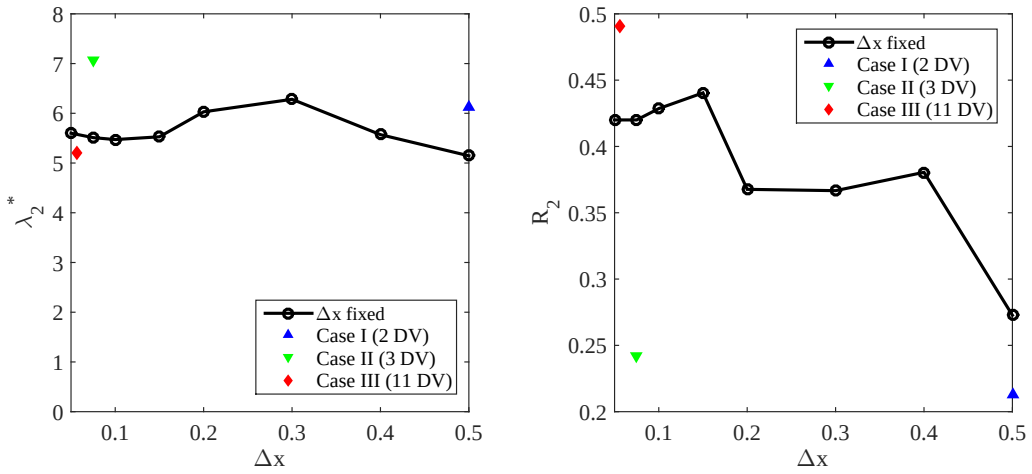



Figure 5.27: Final objective function for fixed rotor separation.




(a) Secondary rotor tip speed ratio

(b) Secondary rotor radius

Figure 5.28: Final design variable values (excluding chord control points) for the fixed $\Delta x$ investigation.

### 5.5.3 Coarse Model Optimization

In this section, the low-fidelity models are used to solve Cases I, II, and III described previously. The results are then compared to the results from the multi-fidelity approach (OSM) as well as the other algorithms (pattern search, SAO, kriging with infill) which utilize the high-fidelity mesh HF. The purpose of this comparison is to demonstrate that the simpler low-fidelity approach is inherently cheaper, but can often result in a worse design when compared to OSM (particularly in problems of high dimensionality).

#### 5.5.3.1 Case I: Optimizing $R_2$ and $\lambda_2$

For Case I, all approaches except for the parametric sweep approach are compared to pattern search using both low-fidelity models (LF 1 and LF 2). The results of this are provided in Table 5.10. Note that when the table lists $H$ for LF 1 and LF 2, the value is obtained by running HF at the low-fidelity optimum; this is how the value $N_f = 1$ rather than $N_f = 0$ is introduced. For this simple two-parameter case, using the low-fidelity models in a direct optimization algorithm led to approximately the same optimum as the other methods. In addition, the use of these low-fidelity models led to much cheaper results.

Table 5.10: Summary of results for Case I using objective function 2 (Sec. 5.1.2). In this case, pattern search with low-fidelity models LF 1 and LF 2 are utilized.

| | PS (HF) | SAO | Kriging | OSM (LF 1) | PS (LF 1) | PS (LF 2) |
|---|---|---|---|---|---|---|
| $\mathbf{x}_0$ | $[10\ 0.5]^T$ | $[10\ 0.5]^T$ | - | $[10\ 0.5]^T$ | $[10\ 0.5]^T$ | $[10\ 0.5]^T$ |
| $x_1^*$ | 6.400 | 6.133 | 5.925 | 6.267 | 6.000 | 6.000 |
| $x_2^*$ | 0.1978 | 0.2125 | 0.2268 | 0.2067 | 0.2067 | 0.2067 |
| $H$ | -2.322 | -2.324 | -2.302 | -2.322 | -2.316 | -2.316 |
| $C_P$ | 0.54458 | 0.54459 | 0.54447 | 0.54458 | 0.54455 | 0.54455 |
| $N_c$ | - | - | - | 47 | 27 | 27 |
| $N_f$ | 32 | 40 | 15 | 5 | 1 | 1 |
| $N_{feq}$ | 32 | 40 | 15 | 10.2 | 4.6 | 3.3 |
| $t_c$ (min) | - | - | - | 69.6 | 39.4 | 20.1 |
| $t_f$ (min) | 435.8 | 535.6 | 202.2 | 66.9 | 12.8 | 12.8 |
| $t_{tot}$ (min) | 435.8 | 535.6 | 202.2 | 136.5 | 52.1 | 32.8 |

While OSM and pattern search led to very similar optima, the methods take different paths. This may partly explain why pattern search leads to a different optimum in Case III (Sec. 5.4). This result is illustrated in Fig. 5.29. These different paths are particularly interesting because in each OSM iteration, the same pattern search algorithm is used to optimize the surrogate model. One would expect, then, that both algorithms would take the same path but this does not occur.

#### 5.5.3.2 Case II: Optimizing $R_2$, $\lambda_2$, and $\Delta x$

Case II using the low-fidelity models led to similar results as Case I. However, due to the added dimension in the design space, results using LF 1 and LF 2 are no longer the same. While the finer low-fidelity model LF 1 led to almost the same solution as the other methods, LF 2 led to a significantly poorer solution. These results are provided in Table 5.11.

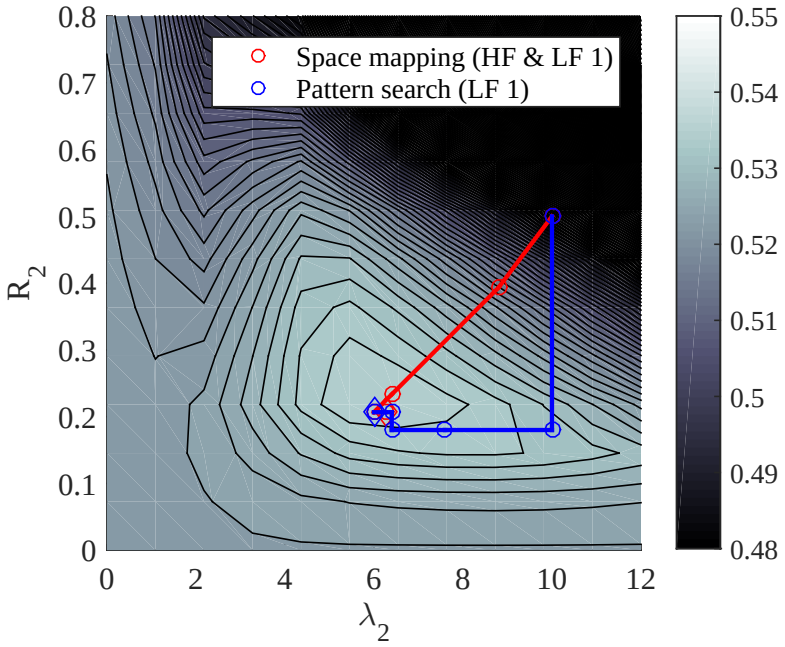



Figure 5.29: Case I optimization paths using OSM (HF & LF 1) and pattern search (LF 1) Note that the diamonds mark the optima found by the two methods.

Table 5.11: Summary of results for Case II using objective function 2 (Sec. 5.1.2). In this case, pattern search with low-fidelity models LF 1 and LF 2 are utilized.

| | PS (HF) | SAO | Kriging | OSM (LF 1) | PS (LF 1) | PS (LF 2) |
|---|---|---|---|---|---|---|
| $\mathbf{x}_0$ | $[10\ 0.5\ 0.5]^T$ | $[10\ 0.5\ 0.5]^T$ | - | $[10\ 0.5\ 0.5]^T$ | $[10\ 0.5\ 0.5]^T$ | $[10\ 0.5\ 0.5]^T$ |
| $x_1^*$ | 6.400 | 6.446 | 6.844 | 7.067 | 6.400 | 6.400 |
| $x_2^*$ | 0.2600 | 0.2575 | 0.2481 | 0.2422 | 0.2867 | 0.3400 |
| $x_3^*$ | 0.1250 | 0.0913 | 0.0841 | 0.0750 | 0.0750 | 0.0500 |
| $H$ | -3.019 | -3.048 | -3.052 | -3.053 | -3.035 | -2.864 |
| $C_P$ | 0.54829 | 0.54844 | 0.54846 | 0.54847 | 0.54837 | 0.54746 |
| $N_c$ | - | - | - | 94 | 54 | 43 |
| $N_f$ | 43 | 50 | 30 | 7 | 1 | 1 |
| $N_{feq}$ | 43 | 50 | 30 | 18.7 | 8.3 | 4.7 |
| $t_c$ (min) | - | - | - | 142.5 | 75.9 | 33.8 |
| $t_f$ (min) | 546.4 | 628.6 | 387.7 | 85.0 | 10.4 | 10.3 |
| $t_{tot}$ (min) | 546.4 | 628.6 | 387.7 | 227.5 | 86.2 | 44.1 |

#### 5.5.3.3 Case III: Optimizing $R_2$, $\lambda_2$, and $\Delta x$, as well as the Blade Chord Distribution

For Case III, the low- and high-fidelity pattern search optima are compared to the OSM optima. Fig. 5.30 provides the final chord distributions using these models and optimization methods. Provided in Table 5.12 is solution data from these methods. Surprisingly, pattern search using the coarsest low-fidelity model yielded a better design than the high-fidelity pattern search approach. On the other hand, pattern search using both HF and LF 1 led to similar, lower-quality designs. Overall, while results using pattern search led to significantly different designs depending on the mesh used, OSM in both cases located a similar type of design. Thus, for this case, and when compared to pattern search, OSM appears to have the advantage of consistency regardless of the low-fidelity model used. Note, however, that the stopping criterion for all of these results is simply when $\left|H^{(i)} - H^{(i-1)}\right| < 0.001$. Results shown in Sec. 5.4.1 showed that the high-fidelity pattern search approach can be restarted without this criterion, leading to further convergence. Presumably, this would also occur for the pattern search results using LF 1 or even LF 2.

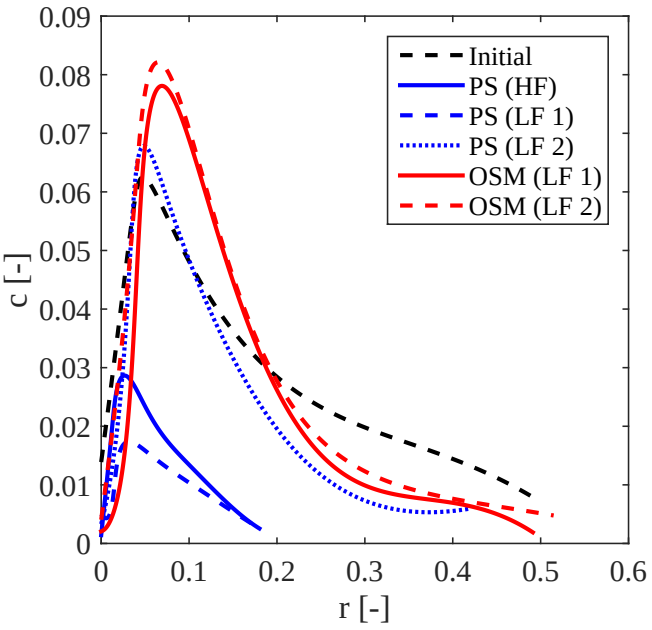



Figure 5.30: Baseline chord distribution versus optimized distributions using of high-, low-, and multi-fidelity PS and OSM.

Table 5.12: Summary of results for Case III using objective function 2 (Sec. 5.1.2). In this case, pattern search with low-fidelity models LF 1 and LF 2 are utilized.

| | OSM (LF 1) | OSM (LF 2) | PS (HF) | PS (LF 1) | PS (LF 2) |
|---|---|---|---|---|---|
| $\mathbf{x}_0$ | $[10\ 0.5\ 0.5\ c_0]^T$ | $[10\ 0.5\ 0.5\ c_0]^T$ | $[10\ 0.5\ 0.5\ c_0]^T$ | $[10\ 0.5\ 0.5\ c_0]^T$ | $[10\ 0.5\ 0.5\ c_0]^T$ |
| $x_1^*$ | 5.200 | 5.200 | 7.600 | 7.600 | 5.200 |
| $x_2^*$ | 0.4911 | 0.5148 | 0.1800 | 0.1800 | 0.4200 |
| $x_3^*$ | 0.0565 | 0.0611 | 0.1250 | 0.0500 | 0.1250 |
| $H$ | -3.521 | -3.441 | -3.001 | -2.921 | -3.348 |
| $C_P$ | 0.55096 | 0.55054 | 0.54819 | 0.54777 | 0.55004 |
| $g$ | -0.0012 | -0.0047 | -0.0760 | -0.0035 | -0.0680 |
| $N_c$ | 1126 | 731 | - | 254 | 405 |
| $N_f$ | 16 | 11 | 258 | 1 | 1 |
| $N_{feq}$ | 168.1 | 59.3 | 258.0 | 35.3 | 35.5 |
| $t_c$ (min) | 1,683.0 | 544.1 | - | 438.2 | 441.3 |
| $t_f$ (min) | 177.0 | 124.0 | 2,929.9 | 11.1 | 11.7 |
| $t_{tot}$ (min) | 1,860.0 | 668.2 | 2,929.9 | 449.3 | 453.0 |

## 5.6 Comparison of Methods

Throughout these optimization cases, the results indicate advantages and disadvantages of each optimization method. A summary of these pros and cons are given in Table 5.13. Depending on the optimization problem (i.e., number of design variables or design space complexity) and the desired outcome (e.g., exploration or quick optimization), any of these methods may be logical to use. For example, the parametric sweep approach may be the best choice for investigating design trends in two design variables, while kriging could be used for this when more design variables are considered (e.g. by using kriging model error as the infill criterion in order to obtain a globally accurate approximation). Overall, for fast design optimization, the use of OSM is recommend when possible. If no low-fidelity model is available, then SAO or kriging may both be logical choices depending on factors such as the magnitude of numerical noise in the CFD model or whether there are multiple minima in the design space. Meanwhile, while pattern search is slower than the rest, in this work it is able to drive the OSM algorithm with reasonable efficiency (which it does by optimizing the space mapping surrogate each iteration).

Table 5.13: Advantages and disadvantages of the optimization techniques.

| **Optimization approach** | **Pros** | **Cons** |
|---|---|---|
| Parametric sweep | • Shows design trends | • Expensive<br>• May not find optimum |
| Pattern search | • Handles numerical noise well | • Converges slowly |
| Kriging | • Global search<br>• May quickly approach optimum | • Unreliable convergence due to numerical noise and design space nonlinearity |
| SAO | • Handles numerical noise well<br>• May quickly approach optimum | • Slowly reaches objective function stopping criterion |
| OSM | • Fast convergence<br>• Can handle higher number of design variables | • Less versatile than others (requires low-fidelity model) |

# CHAPTER 6. CONCLUSION

Design approaches using computationally expensive computational fluid dynamics (CFD) simulations for dual-rotor wind turbines (DRWTs) have been demonstrated and investigated. In particular, the traditional parametric sweep, direct optimization, and surrogate-based optimization (SBO) were considered. The design space of the DRWTs is multi-modal but still fairly smooth. A data-driven surrogate model captures the main features of the design space and can be constructed using fewer evaluations of the CFD model when compared to the parametric sweep and the direct optimization approaches. Both of these data-driven methods, kriging and sequential approximation optimization (SAO), seem well suited for the problem although they both have advantages and disadvantages.

In addition, variable-fidelity design optimization of DRWTs using multi-point output space mapping (OSM) has been demonstrated. The approach constructs a fast yet reliable surrogate model of the DRWT using a coarse mesh CFD model and a correction based on high-fidelity model data. The algorithm uses only one high-fidelity model evaluation per design iteration. The approach is applied to design cases with low-dimensional and medium-dimensional sized design spaces. In the low dimensional cases, the space mapping approach proved to be by far the fastest, and resulted in similar or better solutions when compared to the other methods. The method also worked well in the medium-dimensional case, where it vastly outperformed pattern search, with the total cost being much less than the predicted costs of Kriging with infill and SAO.

In the medium-dimensional case, chord distribution of the secondary rotor led to further improvement of the DRWT design compared to the low-dimensional cases. The final increase in power capture computed by the model is modest, but because the model neglects important physics, the magnitude is not expected to be an exact replica of realistic wind turbine perfor-

mance. Instead, the optimization results are intended to potentially highlight the dominant trends in the design space that may be present when using high-fidelity CFD simulations or experimental testing. Furthermore, the geometry of only the secondary rotor was modified in this study, while the main rotor operated approximately at its optimal tip speed ratio. When computed for lower wind speeds, different inflow velocity distributions, or higher turbulent intensity, the DRWT would see significantly higher performance gains compared to the main rotor operating in isolation. In addition, previous works suggest that such a design would also have a lower cut-in wind speed. When taking into account these other factors, the improvement in power capture would likely coincide better with works described in Chapter 2.

Overall, this thesis observed a 3.5% increase in power relative to the main rotor operating in isolation. This improvement is less than the theoretical value found by Newman [3], who showed that the maximum attainable power extraction by a DRWT is 8% greater than that of a SRWT. This analysis is not directly comparable to the current study, however. This is because the present work takes fluid rotation into account through the use of BEM theory, while Newman's analysis is based solely on actuator disk theory, therefore neglecting this additional component of fluid velocity. Furthermore, the author noted that when validating the theory using a wind tunnel, this analytical result begins to break down for actuator disk separation values of less than one disk diameter.

In the future work, several aspects will be considered. Firstly, Case III will be repeated using different initial design points. Doing so may help pattern search lead to the global optimum rather than a local minimum; this will also help assess the robustness of the OSM algorithm. Secondly, Case III will be repeated using an alternative physics-based SBO method. In particular, input space mapping (ISM) or multi-level optimization [64, 65] will be investigated.

Additional future work will investigate a different objective function as well as a further expanded design space. Previous work indicates that DRWTs yield better wake mixing downstream of the rotors, which leads to improved performance of wind turbines directly downstream. Thus, the alternative objective function will quantify downstream mixing; optimizing such a metric may lead to significantly different design compared to the cases in this work. Finally, the design space will be expanded to include the geometry of both rotors, rather than

only the smaller, upstream one. Although this case would be significantly more expensive, OSM would handle it in a reasonable amount of time. This case is expected to yield large improvements in performance because the parameterization would provide much greater flexibility.

# APPENDIX A. SECONDARY INVESTIGATIONS

The purpose of this chapter is to provide information or results which supplement the main findings in Chapter 5. In particular, the power coefficient contours will be provided for the initial and optimal rotor geometries.

## A.1 Parametric Sweep of the Optimized Rotor

Following the parametric sweep approach shown in Sec. 5.2.1, this brief section provides the power coefficient contours for a rotor separation of 0.5 using the fine mesh (HF). However, this time two rotor geometries are used: the initial chord distribution used in Cases I and II, and the best chord distribution found in Case III (see Sec. 5.4). The results of this analysis are provided in Figs. A.1 and A.2 for the baseline and optimized geometries, respectively.

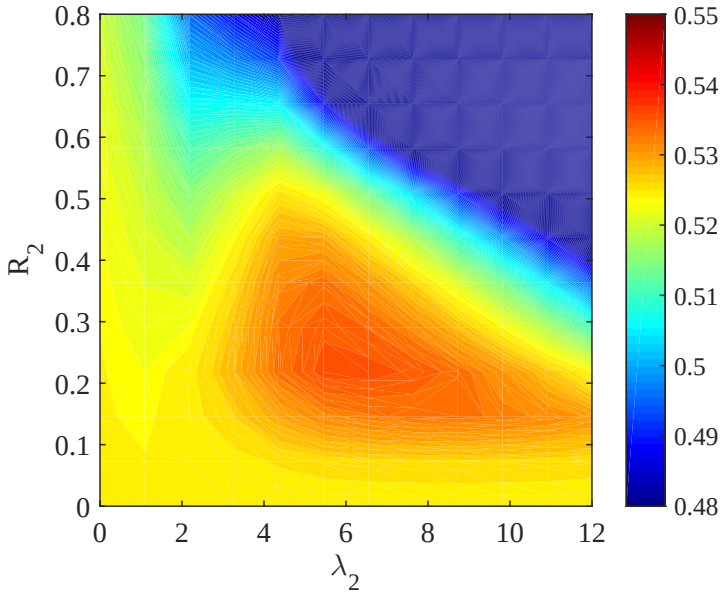



Figure A.1: Power contours of the initial rotor geometry for $\Delta x = 0.5$.

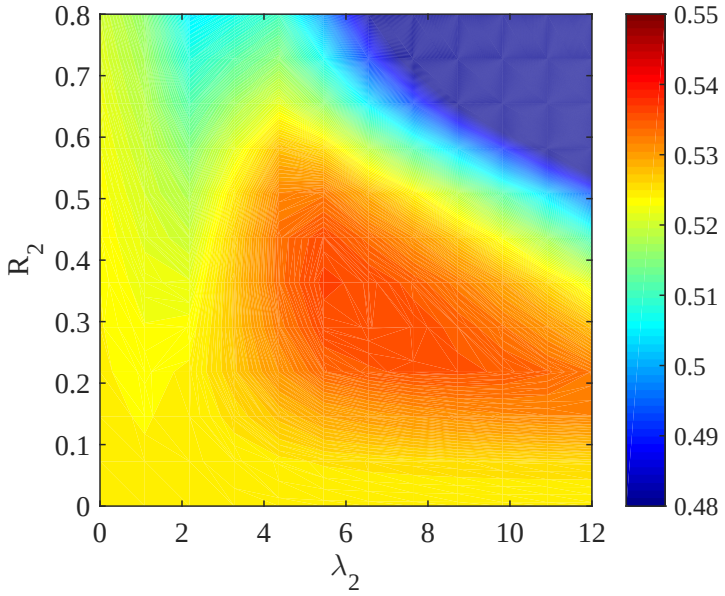



Figure A.2: Power contours of the optimized rotor geometry for $\Delta x = 0.5$.

# BIBLIOGRAPHY

[1] A. Rosenberg, S Selvaraj, and A. Sharma. A novel dual-rotor turbine for increased wind energy capture. *Journal of Physics: Conference Series*, 524(1), 2014.

[2] Suganthi Selvaraj. Numerical investigation of wind turbine and wind farm aerodynamics. Master's thesis, Iowa State University, Ames, IA, USA, 2014.

[3] B.G. Newman. Multiple actuator-disk theory for wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics*, 24:215–225, 1985.

[4] K. Appa. Counter rotating wind turbine systems. Technical report, Energy Innovations Small Grant (EISG) Program Technical Report, 1997.

[5] S.N. Jung, T.S. No, and K.W. Ryu. Aerodynamic performance prediction of a 30 kw counter-rotating wind turbine system. *Renewable Energy*, 30:631–644, 2005.

[6] W. Shen, Z. Zakkam, A. K. Sorensen, and K. Appa. Analysis of counter-rotating wind turbines. *Journal of Physics: Conference Series*, 75:1–9, 2007.

[7] R. W. Y. Habash, V. Groza, Y. Yang, C. Blouin, and P. Guillemette. Performance of a contra rotating small wind energy converter. In *IEEE 6th International Workshop on Electronic Design, Test and Application*, 2011.

[8] B. Moghadassian, A. Rosenberg, H. Hu, and A. Sharma. Numerical investigation of aerodynamic performance and loads of a novel dual rotor wind turbine. In *AIAA Science and Technology Forum and Exposition*, Kissimmee, FL, Jan. 5-9, 2015.

[9] A. Rosenberg and A. Sharma. A prescribed-wake vortex line method for aerodynamic analysis and optimization of multi-rotor wind turbines. In *North American Wind Energy Academy (NAWEA)*, 2015.

[10] Z. Wang, A. Ozbay, W. Tian, A. Sharma, and H. Hu. An experimental investigation on the wake characteristics behind a novel twin-rotor wind turbine. In *AIAA Science and Technology Forum and Exposition*, Kissimmee, FL, Jan. 5-9, 2015.

[11] A. Ozbay, W. Tian, and H. Hu. An experimental investigation on the aeromechanics and near wake characteristics of dual-rotor wind turbines (drwts). In *AIAA Science and Technology Forum and Exposition*, National Harbor, MD, Jan. 13-17, 2014.

[12] J. Smagorinsky. General circulation experiments with the primitive equations: I. the basic experiment. *Monthly Weather Review*, 91(3):99–164, 1963.

[13] O. Reynolds. On the dynamical theory of incompressible viscous fluids and the determination of the criterion. *Philosophical Transactions of the Royal Society of London. A*, 186:123–164, 1895.

[14] M. Churchfield, S. Lee, and P. Moriarty. A large-eddy simulation of wind-plant aerodynamics. In *50th AIAA Aerospace Sciences Meeting*, Nashville, TN, January 9-12, 2012.

[15] F. Porte-Agel, C. Meneveau, and M. B. Parlange. A scale-dependent dynamic model for large-eddy simulation: application to a neutral atmospheric boundary layer. *Journal of Fluid Mechanics*, 415:261–284, 2000.

[16] A. AbdelSalam and V. Ramalingam. Wake prediction of horizontal-axis wind turbine using full-rotor modeling. *Journal of Wind Engineering and Industrial Aerodynamics*, 124:7–19, 2014.

[17] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 12(6), Nov/Dec 1998.

[18] S Koziel. Multi-fidelity multi-grid design optimization of planar microwave structures with sonnet. In *26th Annual Review of Progress in Applied Computational Electromagnetics*, Tampere, Finland, April 26-29, 2010.

[19] L. Leifsson and S. Koziel. *Simulation-Driven Aerodynamic Design Using Variable-Fidelity Models*. Imperial College Press, London, UK, 2015.

[20] V.V. Toropov, A.A. Filatov, and A.A Polynkin. Multiparameter structural optimization using fem and multipoint explicit approximations. *Structural Optimization*, 6:7–14, 1993.

[21] A.A. Giunta and M.S. Eldred. Implementation of a trust region model management strategy in the dakota optimization toolkit. In *Proceedings of the AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA, September 6-8, 2000.

[22] S. Hosder, L.T. Watson, B. Grossman, W.H. Mason, and H. Kim. Polynomial response surface approximations for the multidisciplinary design optimization of a high-speed civil transport. *Optimization and Engineering*, 2:431–452, 2001.

[23] J.H. Jacobs, L.F.P Etman, F. van Keulen, and J.E Rooda. Framework for sequential approximate optimization. *Structural and Multidisciplinary Optimization*, 27:384–400, 2004.

[24] L. Leifsson, E. Hermannsson, and S. Koziel. Optimal shape design of multi-element trawldoors. *Journal of Computational Science*, 10:55–62, September, 2015.

[25] N.V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P.K. Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1):1–28, 2005.

[26] A.I.J. Forrester and A.J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1–3):50–79, 2009.

[27] A. Forrester, A. Sobester, and A. Keane. *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley, 2008.

[28] S. Koziel and L. Leifsson. Knowledge-based airfoil shape optimization using space mapping. In *30th AIAA Applied Aerodynamics Conference*, AIAA Paper 2012-3016, New Orleans, Louisiana, June 25-28, 2012.

[29] P. E. Hancock and T. D. Farr. Wind-tunnel simulations of wind-turbine arrays in neutral and non-neutral winds. *Journal of Physics: Conference Series*, 524(1):012166–012178, 2014.

[30] L. A. M. Tossas and S. Leonardi. Wind turbine modeling for computational fluid dynamics. Technical report, National Renewable Energy Laboratory (NREL), December 2010 – December 2012.

[31] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-mw reference wind turbine for offshore system development. Technical report, National Renewable Energy Laboratory (NREL), February 2009.

[32] J. N. Sorensen and W. Z. Shen. Numerical modeling of wind turbine wakes. *Journal of Fluids Engineering*, 124(2):393–399, 2002.

[33] K. Meister, T. Lutz, and E. Kramer. Simulation of a 5mw wind turbine in an atmospheric boundary layer. *Journal of Physics: Conference Series*, 555(1):012071–012081, 2014.

[34] S. S. A. Ivanell. *Numerical Computations of Wind Turbine Wakes*. PhD thesis, Gotland University, Visby, Sweden, 2009.

[35] J. N. Sorensen and A. Myken. Unsteady actuator disc model for horizontal axis wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics*, 39:139–149, 1992.

[36] E. C. Lynch. *Advanced CFD Methods for Wind Turbine Analysis*. PhD thesis, Georgia Institute of Technology, Georgia, USA, 2011.

[37] R. Mikkelsen. *Actuator Disk Models Applied to Wind Turbines*. PhD thesis, Technical University of Denmark, Copenhagen, Denmark, 2003.

[38] S. W. McIntyre. Space-time computations of wind-turbine aerodynamics with higher-order functions in time. Master's thesis, Rice University, Texas, USA, 2013.

[39] Z. Lyu, Z. Xu, and J. R. R. A. Martins. Benchmarking optimization algorithms for wind aerodynamic design optimization. In *The Eighth International Conference on Computational Fluid Dynamics (ICCFDS)*, Chengdu, Sichuan, China, July 14-18, 2014.

[40] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions*, 6(2):181–197, 2002.

[41] R. E. Perez, P. W. Jansen, and J. R. R. A. Martins. pyOpt: A Python-based object-oriented framework for nonlinear constrained optimization. *Structures and Multidisciplinary Optimization*, 45(1):101–118, 2012.

[42] P. Jansen and R. Perez. Constrained structural design optimization via a parallel augmented lagrangian particle swarm optimization approach. *International Journal of Computational Structures*, 89(13–14):1352–1366, 2011.

[43] S. Koziel, D. Echeverria-Ciaurri, and L. Leifsson. Surrogate-based methods. pages 33–60, In Studies in Computational Intelligence, Editors Koziel, S., and Yang, X.S., Springer-Verlag, Berlin, Germany, 2011.

[44] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computation*, 3:233–260, 1988.

[45] D. I. Papadimitriou and K. C. Giannakoglou. Aerodynamic shape optimization using first and second order adjoint anddirect approaches. *Archives of Computational Methods in Engineering*, 15, 2008.

[46] M. P. Rumpfkeil and D. J. Mavriplis. Efficient hessian calculations using automatic differentiation and the adjoint method with applications. *AIAA Journal*, 48(10):2416–2417, October 2010.

[47] P. E. Gill, W. Murray, and M. A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005.

[48] G. R. Anderson, M. Nemec, and M. J. Aftosmis. Aerodynamic shape optimization benchmarks with error control and automatic parameterization. In *AIAA Science and Technology Forum and Exposition*, Kissimmee, FL, Jan. 5-9, 2015.

[49] F. Bisson and S. Nadarajah. Adjoint-based aerodynamic optimization of benchmark problems. In *AIAA Science and Technology Forum and Exposition*, National Harbor, MD, Jan. 13-17, 2014.

[50] K. Telidetzki, L. Osusky, and D. W. Zingg. Application of jetstream to a suite of aerodynamic shape optimization problems. In *AIAA Science and Technology Forum and Exposition*, National Harbor, MD, Jan. 13-17, 2014.

[51] A. Taylor, L. Green, P. Newman, and M. Putko. Some advanced concepts in discrete aerodynamic sensitivity analysis. In *AIAA 15th Computational Fluid Dynamics Conference*, Anaheim, CA, June 11-14, 2001.

[52] R. H. Byrd, J. Nocedal, and R. A. Waltz. Knitro: An integrated package for nonlinear optimization. In *Large Scale Nonlinear Optimization, 35-59, 2006*, pages 35–59. Springer-Verlag, 2006.

[53] I. Salle and M. Yildizoglu. Efficient sampling and meta-modeling for computational economic models. *Computational Economics*, 44(4):507–536, 2014.

[54] F. M. Alam, K. R. McNaught, and T. J. Ringrose. A comparison of experimental designs in the development of a neural network simulation metamodel. *Simulation Modelling Practice and Theory*, 12(7):559–578, 2004.

[55] S. Koziel and S. Ogurtsov. *Antenna Design by Simulation-Driven Optimization.* Springer International Publishing, New York, USA, 2014.

[56] S. Lophaven, H. Nielsen, and J. Sondergaard. Aspects of the matlab toolbox dace. In *Technical Report, IMM-TR-2002-13, Informatics and Mathematical Modelling, Danish Technical University, Denmark*, 2002.

[57] A. I. Khuri and S. Mukhopadhyay. Response surface methodology. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(2):128–149, 2010.

[58] M. Khurana, H. Winarto, and A. Sinha. Application of swarm approach and artificial neural networks for airfoil shape optimization. In *12th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Victoria, British Columbia Canada, September 10-12, 2008.

[59] A. Basudhar, S. Lacaze, and S. Missoum. Constrained efficient global optimization with probabilistic support vector machines. In *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, TX, USA, September 13-15, 2010.

[60] S. Koziel and S. Ogurtsov. Cpu-budget-driven automated microwave design optimization using variable-fidelity electromagnetic simulations. In *IEEE MTT-S International Microwave Symposium Digest*, June 17-22, 2012.

[61] S. Koziel, Q.S. Cheng, and J.W. Bandler. Space mapping. *IEEE Microwave Magazine*, 9(6):105–122, Dec. 2008.

[62] S. Koziel and L. Leifsson. Adaptive response correction for surrogate-based airfoil shape optimization. In *30th AIAA Applied Aerodynamics Conference*, New Orleans, Louisiana, June 25-28, 2012.

[63] S. Koziel and L. Leifsson. Multi-fidelity airfoil optimization with adaptive response prediction. In *14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Indianapolis, Indiana, Sept. 17–19, 2012.

[64] S. Koziel and L. Leifsson. Multi-level surrogate-based airfoil shape optimization. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Grapevine, Texas, January 7–10, 2013.

[65] Y.A. Tesfahunegn, S. Koziel, and L. Leifsson. Multi-level airfoil optimization with adjoint sensitivity. In *AIAA Aviation and Aeronautics Forum and Exposition (AVIATION)*, Dallas, Texas, June 22–26, 2015.

[66] S. Koziel and L. Leifsson. Surrogate-based aerodynamic shape optimization by variable-resolution models. *AIAA Journal*, 51(1):94–106, 2013.

[67] BE Launder and BI Sharma. Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. *Letters in heat and mass transfer*, 1(2):131–137, 1974.

[68] J.G. Schepers, A.J. Brand, A. Bruining, J.M.R. Graham, M.M. Hand, D.G. Infield, H.A. Madsen, R.J.H. Paynter, and D.A. Simms. Final report of iea annex xiv: Field rotor aerodynamics. Technical Report ECN-C-97-027, Energy Research Center of the Netherlands, 1997.

[69] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, Boston, MA, 1993.