Direct and Surrogate-Based Optimization of Dual-Rotor Wind Turbines

Andrew Thelen,* Leifur Leifsson,† and Anupam Sharma‡

Iowa State University, Ames, Iowa, 50011, USA

Slawomir Koziel§

Engineering Optimization & Modeling Center, Reykjavik University, Menntavegur 1, 101 Reykjavik, Iceland

Dual-rotor wind turbines (DRWT) may offer better energy efficiency over their single-rotor counterparts. The design and analysis of DRWT requires, among other, the use of computational fluid dynamics models. These models can be, depending on their formulation, computationally heavy. Numerous simulations are then required during the design process, and this may render the overall computational cost to be prohibitive. This paper investigates and compares several optimization techniques for the design of DRWTs. In particular, we solve the DRWT fluid flow using the Reynolds-Averaged Navier-Stokes equations with a two-equation turbulence model on an axisymmetric mesh, and consider three design approaches: the traditional parametric sweep where the design variables are varied and the responses examined, direct optimization with a derivative-free algorithm, and surrogate-based optimization (SBO) using data-driven surrogates. The approaches are applied to test cases involving two and three 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 one, the effort of setting up, running, and analyzing the results increases significantly with the parametric sweep approach. The optimization techniques are much easier to use and deliver the results with lower computational cost, where the SBO algorithm outperforms the direct approach.

Nomenclature

\[ C_P = \text{Power coefficient of single rotor turbine or combined power coefficient for DRWT} \ [\text{-}] \]
\[ C_{P1} = \text{Power coefficient of main rotor} \ [\text{-}] \]
\[ C_{P2} = \text{Power coefficient of secondary rotor} \ [\text{-}] \]
\[ \lambda_1 = \text{Main rotor tip speed ratio} \ [\text{-}] \]
\[ \lambda_2 = \text{Secondary rotor tip speed ratio (assuming counter-rotating rotors)} \ [\text{-}] \]
\[ R_1 = \text{Main rotor tip radius} \ [\text{m}] \]
\[ R_2 = \text{Secondary rotor tip radius} \ [\text{m}] \]
\[ \Delta x = \text{Axial separation between rotor disks} \ [\text{m}] \]
\[ x = \text{Design variable vector} \]
\[ f(x) = \text{Objective function value} \]
\[ \tau = \text{Torque} \ [\text{Nm}] \]
\[ \omega = \text{Rotational rate of rotor} \ [\text{rad/s}] \]
\[ \rho_{\infty} = \text{Free-stream density} \ [\text{kg/m}^3] \]
\[ v_{\infty} = \text{Free-stream velocity} \ [\text{m/s}] \]

*Graduate Student, Department of Aerospace Engineering, AIAA Student Member.
†Assistant Professor, Department of Aerospace Engineering, AIAA Senior Member.
‡Assistant Professor, Department of Aerospace Engineering, AIAA Senior Member.
§Professor, School of Science and Engineering, AIAA Senior Member.
I. Introduction

Modern 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. The root region of the turbine blade must be capable of supporting the weight of the remaining length of the blade. This structural requirement invariably leads to a root cross-section with poor aerodynamic properties. In an effort to mitigate these losses, a number of studies have suggested that the loss of energy capture associated with this root region can be reduced by integrating a secondary rotor into the design (see, for example, Newman1).

There are a spectrum of numerical tools of varying cost and fidelity that can be used to model a dual-rotor wind turbine (DRWT) design. Large eddy simulation (LES) and Reynolds-Averaged Navier-Stokes (RANS) have proven to be quite useful for wind turbine applications.2,3 With these approaches available, it is important to consider the overall computational cost required to search for an optimized DRWT design. For example, a high-fidelity model, 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 lower fidelity model, which may neglect important nonlinear physics, would not be an effective tool in determining the precise conditions for an optimized DRWT design.

Previous studies demonstrate that an actuator disk model, developed using the simpleFoam RANS solver in OpenFoam, can be used to model wind turbine performance with reasonable accuracy.4–6 Moreover, because certain assumptions allow it to be executed relatively fast, 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 CFD or experimental analyses.

While the actuator disk model is faster than other available approaches, 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 present study focuses on developing an efficient approach for optimizing the DRWT using minimal computational resources. In particular, we investigate several approaches to search the design space. First, the problem is approached by carrying out parametric sweeps. Then, a direct search for the optimized design will be carried out using the pattern-search algorithm7 (which is a derivative-free approach). Finally, we use surrogate-based optimization8,9 (SBO) with data-driven surrogate models. The approaches are compared based on the design quality as well as optimization cost metrics, such as the computing time and the number of function evaluations.

The paper is organized as follows. We begin by describing the rotor computational model where the specifics of the governing equations, computational fluid dynamics (CFD) model, grid independence study, and numerical validation are given. Then the optimization methodology is described, including the parametric sweep, direct optimization, and SBO. The optimization approaches are applied to two cases of DRWT, one with two parameters and the other with three. The paper ends with conclusions.

II. Rotor Computational Model

A. Governing Equations

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

\[
\frac{\partial \bar{u}_i}{\partial x_i} = 0,
\]

and the momentum equation is

\[
\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j^2} - \frac{\partial \bar{u}'_i \bar{u}'_j}{\partial x_j} + \frac{f_i}{\rho},
\]

where the Reynolds stress tensor, \(\bar{u}_i' \bar{u}_j'\), is modeled using the standard two-equation \(k - \epsilon\) turbulence model.10 In these equations, \(u, x, \rho, p\), and \(\nu\) represent velocity, position, density, pressure, and kinematic viscosity, respectively. The bar in these equations represents the mean value over time, while the prime symbol
represents a perturbation from the mean value. 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. The body force is distributed over a volume and a Gaussian distribution is applied along the flow direction as suggested in Mikkelsen.\textsuperscript{11} Prandtl’s tip loss correction is applied to account for finite-span induction (tip) loss that is ignored in the blade element theory.

B. CFD Model

The RANS equations are solved on an axisymmetric domain with a wedge angle of one degree using simpleFoam. 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. 1. A nondimensional velocity, density, and pressure of one is applied to the inlet surface. In the diagram, flow direction is from left to right.

C. 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. For reference, the blocks used for the model are provided in Fig. 1. 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, viewed from two different perspectives, is provided in Fig. 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. The results indicate that the resolution of the numerical discretization does play a role in the predicted value of power capture. Moreover, for very high accuracy, a mesh cell count of hundreds of thousands of cells may be needed.

The present study will use the third coarsest mesh, with approximately 47k cells, for the optimization runs. This mesh was selected because it 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 in error due to more refinement would be insignificant when compared to the error associated with the model’s simplification of physics. Example model output are provided in Fig. 4 where the axial separation between rotor disks is 0.5, the tip speed ratio of the main rotor is 8, and the tip speed ratio and radius of the secondary rotor are 6 and 0.2, respectively.

![Figure 1: The openFoam CFD model: a view of the blocks (left), and a schematic of the solution domain (right).](image_url)
Figure 2: Example mesh of the DRWT.

Figure 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).

Figure 4: Example flow solutions showing pressure coefficient contours for single and dual rotors.

D. Numerical Validation

The actuator disk model method is validated against experimental data and blade element momentum theory (BEMT) predictions. The three-bladed, stall controlled, 95 kW Tellus T-1995 is used for validation. The
Figure 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, $C_T$ and (d) thrust force coefficient, $C_T$.

Each model evaluation is costly, so depending on the optimization method used, the amount of time needed to find the optimum can vary dramatically. Therefore, when selecting a method, minimizing function evaluations is of primary interest. With this motivation, this section presents three general approaches. These include the traditional parametric sweep, a direct optimization method, and two types of data-driven surrogate-based methods.

### A. Problem Formulation

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 have a significant impact on wind turbine power.
capture. However, these properties will be kept fixed for the present study, and are selected based on an inverse design analysis described in Rosenberg et al.\textsuperscript{4}

In general, an optimization task is approached by finding the optimal design vector $\mathbf{x}^*$ by solving the following problem

$$
\mathbf{x}^* = \arg\min_{\mathbf{x}} H(\mathbf{f}(\mathbf{x})) \text{ s.t. } \mathbf{g}(\mathbf{x}) \leq 0, \mathbf{h}(\mathbf{x}) = 0, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u},
$$

(3)

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 $\mathbf{u}$ and the lower and upper bounds on $\mathbf{x}$, respectively.

For DRWTs, we want to maximize the power capture from both rotors, therefore the objective function used for this problem is $H(\mathbf{f}(\mathbf{x})) = -C_{P1} - C_{P2}$, where $C_P = \omega \tau / (\frac{1}{2} \rho_{\text{inf}} v_{\text{inf}}^3 A)$. In this expression, $\omega$ is the rate of rotation, $\tau$ is the rotor's torque, $\rho_{\text{inf}}$ and $v_{\text{inf}}$ are free stream density and velocity of one, and $A$ is the area of the main rotor disk. The design variables used in this case are the tip speed ratio ($\lambda_2$) and radius ($R_2$) of the secondary rotor as well as the axial separation between disks ($\Delta x$) (see Fig. 1). The design variable vector is written as $\mathbf{x} = [\lambda_2, R_2, \Delta x]^T$. There are no inequality or equality constraints in this case, only design variable vector bounds (specified in Sec. IV).

B. 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.

C. Direct Optimization

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 jumps in 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.\textsuperscript{7,12} 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

![Diagram of pattern-search method using a rectangular grid.](image)

Figure 6: Conceptual diagram of pattern-search method using a rectangular grid.\textsuperscript{7,12}
A rectangular grid (i.e., one point in each direction and in each dimension) is used in our 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 in case the above poll step does not lead to an improved design. The poll stage of the pattern search process is illustrated in Fig. 6. For more information about the particular pattern search algorithm used for this study, refer to Ref. 7, 12

D. Surrogate-Based Optimization

There are many different surrogate modeling approaches that could successfully model the physics of DRWT power generation.8,9,13 There are two general types: data-driven and physics-based surrogates. Data-driven models include methods such as polynomial response surfaces8 and kriging,9 while physics-based models include methods such as space mapping,14,15 shape-preserving response prediction,16 adaptive response correction,17 adaptive response prediction,18 or multi-level optimization.19,20 For this application, the use of polynomial response surfaces,8 as well as kriging,9 will be explored.

1. Sequential Approximation Optimization (SAO)

A simple variation of SBO with data-driven surrogates is sequential approximate optimization (SAO).21–24 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 move strategy. Often the new subregion is in the direction of the last iteration optimum,24 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.22, 25

In this paper, we follow the work of Leifsson et al.26 and build the local surrogate models using a quadratic response surface without mixed terms:

\[
\hat{y}(\mathbf{x}) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \beta_{ii} x_i^2
\]

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

\[
\beta = (U^T U)^{-1} U^T y_s
\]

with

\[
U = \begin{bmatrix}
1 & x_1 & \cdots & x_1^1 & \cdots & x_1^2 & \cdots & x_m^2 & \cdots & x_m^n & \cdots & x_m^1 & \cdots & 1 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
1 & x_1^n & \cdots & x_m^n & \cdots & (x_1^n)^2 & \cdots & (x_m^n)^2 & \cdots & (x_m^1)^2 & \cdots & (x_m^1)^2 & \cdots & (x_m^n)^2
\end{bmatrix}
\]

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

Once an expression for the response surface is found, the optimum on the surface is found using the pattern search method discussed previously. The design space is then sampled at this point and one of two actions are subsequently taken. If \(H(f(\mathbf{x}))\) at this new point is worse than \(H(f(\mathbf{x}))\) at the iteration’s center point, then the grid size is scaled down by a user-defined parameter. If, on the other hand, \(H(f(\mathbf{x}))\) at this new point is better than \(H(f(\mathbf{x}))\) 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 may be scaled up or down by user-defined parameters depending on the value of the gain ratio.
\[
\delta = \frac{H(f(\mathbf{x}^{imp})) - H(f(\mathbf{x}^{(k)})))}{H(g^{(k)}(\mathbf{x}^{imp})) - H(g^{(k)}(\mathbf{x}^{(k)})))}
\]

where \(\mathbf{x}^{imp}\) is the current surrogate’s optimum and \(k\) denotes the current iteration). This process is repeated until 50 function evaluations have been carried out. In this study, no other convergence criteria are used so that the numerical behavior can be better compared to other methods.

2. **Kriging with Updates**

In general, a kriging surrogate model is the sum of two components: 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 by

\[
\hat{y}(\mathbf{x}) = \beta^T u(\mathbf{x}) + \gamma^T v(S, \mathbf{x}),
\]

where

\[
u(\mathbf{x}) = [u_1(\mathbf{x}), ..., u_p(\mathbf{x})]^T
\]

and

\[
v(S, \mathbf{x}) = [R(\theta_1, s_1, \mathbf{x}), ..., R(\theta_m, s_m, \mathbf{x})]^T.
\]

In these expressions, \(s\) is a sampled design point (while \(S\) represents the set of all \(s\)), \(x\) is the design vector for which a prediction is needed, and \(\hat{y}(x)\) is the predicted response at this value \(x\). Given in (9) and (10) are expressions for \(u\) and \(v\), which represent unweighted polynomial regression and fluctuation terms, respectively. Reasonable values for the model parameters \(\beta\), \(\gamma\), and \(\theta\) are computed within the DACE kriging Matlab toolbox.

The parameters \(\beta\), \(\gamma\), and \(\theta\) are functions of not only \(S\) and \(f(S)\), but also the user-defined regression and correlation models. For this study, we use a second order polynomial regression model and the Gaussian correlation model

\[
R(\theta, \mathbf{x}, s) = \prod_{j=1}^n R_j(\theta_j, (\mathbf{x}_j - s_j)) = \prod_{j=1}^n \exp(-\theta_j (\mathbf{x}_j - s_j)^2).
\]

In order to generate the surrogate, the 2nd order regression model requires more samples than the 0th or 1st order alternatives. The value which describes this lower limit is given by

\[
m_{min} = 1 + 2n + \left(\begin{array}{c} n \\ 2 \end{array}\right).
\]

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., 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 (12), 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):
\[
x_{IC1} = \arg\min_x (\hat{f}(x)) \tag{13}
\]

\[
x_{IC2} = \arg\min_x (-E[I(x)]) \tag{14}
\]

Here, \(E[I(x)]\) is found by

\[
E[I(x)] = (s > 0) * \left[ (y_{min} - \hat{y}(x)) \Phi \left( \frac{y_{min} - \hat{y}(x)}{s(x)} \right) + s \phi \left( \frac{y_{min} - \hat{y}(x)}{s(x)} \right) \right], \tag{15}
\]

which can be reduced to

\[
E[I(x)] = (y_{min} - \hat{y}(x)) \left[ \frac{1}{2} + \frac{1}{2} \text{erf} \left( \frac{y_{min} - \hat{y}(x)}{\hat{s}\sqrt{2}} \right) \right] + \hat{s} \frac{1}{\sqrt{2\pi}} \exp \left[ -\left( \frac{(y_{min} - \hat{y}(x))^2}{2\hat{s}^2} \right) \right]. \tag{16}
\]

In these expressions, \(\Phi\) and \(\phi\) are the cumulative distribution and probability density functions, respectively, \(s\) is the predicted error at the input design point \(x\), 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 in this work insignificant, 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 whichever design is better 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.

\section*{IV. Numerical Results}

The results of two separate studies are presented. In the first case, we limit the number of design variables to two. In particular, we let the secondary rotor radius \(R_2\) and tip speed \(\lambda_2\) be design variable while keeping a constant axial separation \(\Delta x\) between the rotor disks, i.e., the design variable vector is \(x = [\lambda_2 \ R_2]^T\) with \(\Delta x = \text{constant}\). The second case investigates the effect of all three design variables on power generation, i.e., we have \(x = [\lambda_2 \ R_2 \ \Delta x]^T\). The upper and lower bounds of the design variables are give in Table 1.

\subsection*{A. Case 1 Results}

\subsubsection*{1. Parametric Sweep}

For this two-dimensional case, 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
Table 1: Bounds on the design variables for each case.

<table>
<thead>
<tr>
<th></th>
<th>(\lambda_2)</th>
<th>(R_2)</th>
<th>(\Delta x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>[0,12]</td>
<td>[0,0.8]</td>
<td>0.5</td>
</tr>
<tr>
<td>Case 2</td>
<td>[0,12]</td>
<td>[0,0.8]</td>
<td>[0.05,0.8]</td>
</tr>
</tbody>
</table>

Table 2: Initial points used for pattern search for Case 1.

<table>
<thead>
<tr>
<th></th>
<th>(\lambda_2)</th>
<th>(R_2)</th>
<th>(\Delta x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trial 1</td>
<td>10</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Trial 2</td>
<td>2</td>
<td>0.65</td>
<td>0.5</td>
</tr>
<tr>
<td>Trial 3</td>
<td>6</td>
<td>0.6</td>
<td>0.5</td>
</tr>
</tbody>
</table>

of 8 is used for this study because it corresponds approximately to the maximum energy capture of a single rotor wind turbine. The results of this study are provided in Fig. 7 (top left).

This approach, requiring a total of 144 high-fidelity CFD evaluations, yielded an optimum at \(\lambda_2 = 6.54591, R_2 = 0.21818\) and a power coefficient of 0.536361. This value is approximately 99.97% of the maximum possible value. However, results from all optimization algorithms obtain higher \(C_P\) values in much less time, which is to be expected.

2. Direct Optimization: Pattern Search

Three initial design points, which are given in Table 2, were used in the direct search for the optimal design. One of these points led to zero secondary rotor radius and therefore a single rotor design (Fig. 7, bottom left). 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 Figures 7 (top and bottom right). 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.

3. Surrogate-based Optimization: SAO

Optimization using SAO yielded results comparable to pattern search and kriging. However, this method appears to have two advantages. Firstly, it can move diagonally in the design space, unlike the pattern search algorithm. Secondly, it appears to converge more reliably when compared to kriging with updates. The resulting design points sampled by this method are presented in Fig. 8. Convergence metrics are given in Fig. 9.

4. Surrogate-based Optimization: Kriging with Updates

When using kriging-based optimization, the final result can depend heavily on the initial sampling plan. Therefore, in order to better compare the two infill criteria, the same initial sampling will be used for both criteria. Figure 10 below shows the result of 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 below in Fig. 11. The results shown in Fig. 11 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
Figure 7: Parametric sweep and pattern search results for Case 1: increase of power coefficient compared to single-rotor design (upper left), trials 1, 2, and 3 of pattern search (with upper left values as a backdrop) (upper right, lower left, lower right, respectively).

Figure 8: Sampled design points using SAO for Case 1.

Figure 9: Convergence history of the SAO method for Case 1.
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.

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. Another approach that may improve convergence is to implement a variation of the trust region concept.

5. Discussion

Given in Table 3 are the final results of the different methods for Case 1. While they all give approximately the same final design after 50 function evaluations, some methods are better than others. Specifically, kriging appears to be best suited while SAO is slightly more costly but converges better.

B. Case 2 Results

1. Parametric Sweep

The parametric sweep carried out for Case 2 were obtained by sampling an $8 \times 8 \times 8$ array of grid points throughout the design space. Given in Fig. 12 are power coefficient isocontours in the three-dimensional design space. This particularly expensive analysis resulted in an optimum of $\lambda_2 = 6.85757$, $R_2 = 0.22857$, $\Delta x = 0.15714$, and a $C_P$ value of 0.339588. This value of power coefficient is 99.2% of the maximum possible value. Clearly, this traditional parametric approach is much less efficient than the other methods.
Table 3: Summary of results for Case 1.

<table>
<thead>
<tr>
<th></th>
<th>Pattern search</th>
<th>SAO</th>
<th>Kriging (IC1)</th>
<th>Kriging (IC2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \bar{x}_0 )</td>
<td>[10, 0.5]</td>
<td>[10, 0.5]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>( x_{1*} )</td>
<td>6.07407</td>
<td>6.08308</td>
<td>6.01553</td>
<td>5.97475</td>
</tr>
<tr>
<td>( x_{2*} )</td>
<td>0.21481</td>
<td>0.21341</td>
<td>0.21320</td>
<td>0.21392</td>
</tr>
<tr>
<td>( H )</td>
<td>-0.536504</td>
<td>-0.536507</td>
<td>-0.536505</td>
<td>-0.536503</td>
</tr>
<tr>
<td>Cost(^a)</td>
<td>14</td>
<td>11</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

\(^a\) Function calls needed to reach 99.9% of optimum.

2. Direct Optimization: Pattern Search

The pattern search method was able to hone in on the optimum point fairly quickly, but converged slowly. Convergence of this method for Case 2 is provided in Fig. 13.

3. Surrogate-based Optimization: SAO

SAO again worked fairly well for this application. If a convergence criterion based on \( H \) or \( x \) had been used, the method would have converged after just 3 iterations. The results for this case are provided in Fig. 14.

4. Surrogate-based Optimization: Kriging with Updates

The use of kriging with updates worked fairly well for Case 2, locating the region of the optimum immediately after the sampling plan ended. Unfortunately, convergence on the optimum design point is slow when compared to pattern search and SAO. Convergence of this method is provided in Fig. 15.

5. Discussion

Table 4 shows a comparison of results for Case 2. In this case, SAO resulted in the best performance while both kriging infill criteria resulted in the slowest convergence. However, it is important to note that the performance of this method depends heavily on the initial sampling plan. This may explain why kriging yielded the best results in Case 1 but the worst results in Case 2.

Figure 12: Parametric sweep results for Case 2.
V. Conclusion

Design approaches using a computationally expensive computational fluid dynamics (CFD) model 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. In particular, the sequential approximation optimization (SAO) approach seems particularly well suited for the problem. It is important to note, however, that results using kriging are heavily dependent on the initial sampling plan.
Table 4: Summary of results for Case 2.

<table>
<thead>
<tr>
<th>Pattern search</th>
<th>SAO</th>
<th>Kriging (IC1)</th>
<th>Kriging (IC2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bar{x}_0$</td>
<td>[10, 0.5, 0.5]</td>
<td>[10, 0.5, 0.5]</td>
<td>-</td>
</tr>
<tr>
<td>$x_1^*$</td>
<td>5.92593</td>
<td>6.17608</td>
<td>5.99375</td>
</tr>
<tr>
<td>$x_2^*$</td>
<td>0.32222</td>
<td>0.30576</td>
<td>0.33094</td>
</tr>
<tr>
<td>$x_3^*$</td>
<td>0.10000</td>
<td>0.10912</td>
<td>0.11421</td>
</tr>
<tr>
<td>$H$</td>
<td>-0.540009</td>
<td>-0.539949</td>
<td>-0.539962</td>
</tr>
<tr>
<td>Cost*</td>
<td>21</td>
<td>15</td>
<td>28</td>
</tr>
</tbody>
</table>

* Function calls needed to reach 99.9% of optimum.

In future work, we will investigate the application of alternative optimization methods to cases of increased dimensionality. In particular, we will consider a case with 10 or more design variables, which, among other parameters, would control chord and twist distributions of the secondary rotor. In such a case, the methods described in the present study would be prohibitively expensive. As a result, we will use linear response surfaces in place of the quadratic variety used in this study. We will also focus on the use of physics-based surrogate methods. Specifically, we will investigate the use of space-mapping as well as a combination of space-mapping and linear response surfaces. Additionally, the effects of initial sampling plan will be investigated with regards to the method using kriging with updates. As a result, we expect to reach an improved design using increased dimensionality of design parameters as well as a collection of more efficient optimization methods.

References


