Design optimization of a curved wind turbine blade using neural networks and an aero-elastic vortex method under turbulent inflow

Sessarego, Matias; Feng, Ju; Ramos García, Néstor; Horcas, Sergio González

Published in:
Renewable Energy

Link to article, DOI:
[10.1016/j.renene.2019.07.046](https://doi.org/10.1016/j.renene.2019.07.046)

Publication date:
2020

Document Version
Early version, also known as pre-print

[Link back to DTU Orbit](#)

Citation (APA):
Sessarego, M., Feng, J., Ramos García, N., & Horcas, S. G. (2020). Design optimization of a curved wind turbine blade using neural networks and an aero-elastic vortex method under turbulent inflow. *Renewable Energy*, 146, 1524-1535. <https://doi.org/10.1016/j.renene.2019.07.046>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Design optimization of a curved wind turbine blade using neural networks and an aero-elastic vortex method under turbulent inflow

Matias Sessarego^a, Ju Feng^{a,*}, Néstor Ramos-García^a, Sergio González Horcas^b

^a*Fluid Mechanics Section, Department of Wind Energy, Technical University of Denmark,
Nils Koppels Allé, 2800 Kongens Lyngby, Denmark*

^b*Aerodynamic Design Section, Department of Wind Energy, Technical University of
Denmark,
Frederiksborgvej 399, 4000 Roskilde, Denmark*

Abstract

This article describes the application of neural networks for the design optimization of a curved wind turbine blade using an aero-elastic simulator with synthetic inflow turbulence. A vortex particle method where the wind turbine blades are represented by lifting-line theory is used, while the wind turbine structural dynamics are modeled using a finite-element multi-body based approach. A neural network together with a gradient-based optimizer allows to quickly design a new curved wind turbine blade in a complex aero-elastic wind-turbine simulation scenario. The blade design found from the neural network has increased pre-bend and sweep compared to the straight blade design. It produces approximately 1% more power on average with a slight increase of mean thrust on the rotor of 0.02% compared to the straight one. This study demonstrates that neural networks can be effective for designing wind turbine rotor blades involving complex aero-elastic simulation scenarios with turbulent inflow conditions. Further work may improve the performance of the neural network's predictive capabilities as well as the optimized design.

Keywords: wind turbine blade design, neural network, optimization, vortex particle method, aerodynamics, aero-elasticity

1. Introduction

Wind turbine blade design is a multi-disciplinary problem that involves various potentially conflicting objectives and constraints, which come from considerations on aerodynamic efficiency, structural integrity, cost, reliability, manufacturing convenience, transportation feasibility, etc. [1, 2]. This design problem

*Corresponding author

Email address: jufen@dtu.dk (Ju Feng)

is often split into two parts, i.e., aerodynamic design and structural design. The aerodynamic design determines a blade's outer geometry and affects its aerodynamic efficiency and aerodynamic load. The structural design involves materials, components and the internal structure of the blade, which affect the structural integrity, cost and reliability [1, 3, 4].

Most studies focused on one part of the design problem, either the aerodynamic design [5, 6, 7, 8, 9, 10, 11] or the structural design [4]. Fewer studies tried to solve the two parts together, either separately or sequentially [3, 12], or simultaneously [2, 13, 14]. The aerodynamic design of wind turbine blades covers various aspects of its geometry, e.g., airfoils [5, 8, 9], chord and twist distributions [9, 10, 12, 13], winglets [6], sweep [15, 16], pre-bend [13], and combinations thereof [13, 14]. In this study, the focus is on the design optimization of sweep and pre-bend for a curved blade.

Usually the design of the blade's outer geometry is performed using an aerodynamic or aero-elastic simulator. The majority of blade design studies used simulators based on the blade element momentum (BEM) method, for example, as in [2, 8, 9, 13, 14, 15, 16]. The reason is the low computational cost of BEM-based simulators, which makes the optimization process easier to implement. Simulators of higher fidelity are more accurate, but their higher computational cost makes them difficult to be used in design optimization, especially when optimization methods that require a large number of simulations are used [17]. Sessarego et al. [10] tackled this problem using surrogate modeling and a vortex method as the higher-fidelity aerodynamic simulator. In this study, the design is performed using an aero-elastic vortex particle method and neural networks (NNs).

NNs, also known as artificial neural networks, are computational models inspired by the biological neural networks found in the brain [5, 18, 19]. Furthermore, NNs are the most widely used among the various artificial intelligence (AI) models [20]. The structure of the network is typically comprised of artificial neurons connected by adjustable weights in input, output and hidden layers. NNs learn or are trained through experience by detecting patterns and relationships in data. The result is an intelligent computational architecture capable of learning, prediction and recognition [19, 20]. NNs can be used to solve a wide variety of problems in many fields, e.g., pharmaceutical research [19], speech recognition [21], autonomous vehicle navigation [22], and in the context of the current article, wind energy.

A recent state-of-the-art report [23] found that AI has been applied in various fields of wind energy. Examples include wind power forecasting [24], wind turbine control [25, 26], and fault diagnosis [27]. Catalão et al. [24] used NNs in short-term wind power forecasting. Eskander [25] proposed an NN-based controller for wind turbines with a permanent magnet synchronous generator to achieve maximum power tracking as well as output voltage regulation. Schlechtingen and Santos [27] used NNs for wind turbine fault detection. The current study focuses on the application of NNs for wind turbine blade shape design. Previous works in the literature involving NNs and wind turbine design include [5, 6, 7, 8, 11].

Ribeiro et al. [5] employed computational fluid dynamics (CFD), genetic algorithms (GAs) and NNs for wind turbine airfoil design. Elfarra et al. [6] performed a design optimization of a wind turbine blade winglet using CFD, a GA and an NN as well. Casas et al. [7] used an NN to correct the aerodynamic coefficients of airfoils in an automatic wind turbine blade optimization environment. Similar to [7], Chi et al. [8] also relied on an NN in a blade design optimization framework. Kaya and Elfarra [11] optimized the taper/twist stacking axis location of a wind turbine rotor blade using CFD and NNs.

In summary, the contribution of the current article is the application of an NN to design the curved shape of a wind turbine blade using an advanced aero-elastic simulator with turbulent inflow conditions. Previous works [5, 6, 7, 8, 11] involving wind turbine blade design and NNs have not considered advanced aero-elastic simulations of the entire rotor with turbulent inflow conditions. An advanced aero-elastic simulator refers to a method that is higher in fidelity than the standard BEM based approaches, see e.g., [2, 8, 9, 13, 14, 15, 16, 28, 29, 30]. Advanced methods can be, e.g., CFD or vortex-based methods. Here, a vortex particle method where the wind turbine blades are represented by lifting-line theory is used, while the wind turbine structural dynamics are modeled using a finite-element multi-body based approach. Since the influence of curved blades on trailed vorticity and curved lifting lines are not modeled in BEM-based aerodynamic models [31], the aero-elastic simulator employed here is more suitable for curved wind turbine blade shape design. However, such a simulation scenario is challenging for wind turbine design because of the stochastic wind input and the high computational cost of the simulator.

NNs are particularly suitable for the problem presented here, since they are able to map any kind of input-to-output problem [11] by iterative adjustment of the weights [19]. In the context of the present work, NNs are also advantageous because they are effective for reducing the computational cost of an optimization problem, as shown by previous work [6]. The current article is a continuation of the work from [32], where arbitrary blade sweep and winglet shapes were chosen and their aerodynamic performance were compared to their straight blade counterpart. In the current work, an NN is constructed using twenty sample blade shapes and is then used to find an improved curved blade shape based on aero-elastic simulations with turbulent wind inflow.

The article is organized into four sections: Introduction, methodology, results and conclusions. The methodology section briefly describes the wind turbine aero-elastic simulator and the setup of the simulation cases in two subsections: Simulation tool and simulation setup. The NN and optimization setup is also described in the methodology section. The results from the aero-elastic simulations involving turbulent wind inflow as well as the improved blade design shapes found from the NNs are provided in the results section. Conclusions are given in the last section.

2. Methodology

95 The methodology contains four subsections: Simulation tool, simulation
setup, NN setup and optimization setup. In the first subsection, the wind
turbine aero-elastic simulator will be briefly described. The second subsection
will describe the setup of the simulation cases and blade geometry. The NN
setup is described in the third subsection. Last, how the NN is used to find
100 optimum blade shapes is described in the fourth subsection.

2.1. Simulation tool

The in-house aero-elastic vortex code used in the current work is called MIRAS [33, 34], i.e., Method for Interactive Rotor Aero-elastic Simulations. MIRAS is a multi-fidelity aero-elastic tool for wind turbine performance analyses.
105 In the present design study, the lifting line aerodynamic module is used, which
relies on two-dimensional (or 3D-corrected) polar data for lift and drag. Wind
turbine elasticity, including blades, tower, and shaft, is taken into account by
coupling the aerodynamic module of MIRAS with the multi-body finite-element
structural model of HAWC2 [29, 35]. The HAWC2 structural model supplies the
110 wind turbine structural deformations and dynamics. A loose-coupling approach
is used, where the aerodynamic and structural information is exchanged once
per time step for every time step.

MIRAS is capable of simulating long time series (>10 min real time) involving complex three-dimensional (3D) aerodynamic inflow scenarios in a few
115 hours of computational time. The high computational efficiency of MIRAS is
due to its particle mesh and high performance computing parallelization imple-
mentations. In the traditional approach found in free-wake vortex codes, the
wake is represented by constructing vortex filaments where the trajectory of
its end-markers is governed by the Biot-Savart interaction law. In the particle
120 mesh, the wind turbine wake is represented by particles that are projected onto
a uniform mesh. The particle interactions within the mesh are computed using
a Fast Fourier Transform algorithm. Parallelization in MIRAS is based on the
message passing interface (MPI) software. Each simulation in this study uses
one-hundred and sixty 2.8 GHz processors on a Linux cluster and requires ap-
proximately seventeen hours of computational time. Further details regarding
125 MIRAS and the particle mesh implementation are available in [33, 34, 36].

Synthetic inflow turbulence in MIRAS is simulated by 1) using the Mann
model [37, 38] to generate a velocity field (i.e., a turbulence box), 2) trans-
forming the velocity field into a set of vortex particles (i.e., a particle cloud),
130 and 3) releasing gradually the particle cloud in a plane upstream of the wind
turbine rotor. The advantage of using a particle cloud is the ability to simulate
free turbulence, where the vorticity due to turbulent wind can interact with
the vorticity generated by the interaction of the wind with the turbine rotor
blades. The methodology represents more accurately the physics occurring on
wind turbines operating in turbulent wind than the traditional frozen-/velocity-
135 based approaches typically used in BEM-based codes such as HAWC2 [29] and
FAST [30]. The `windsimu` [39] program is used to generate velocity-field boxes

according to the Mann model. The inputs to the `windsimu` program used in the current study are summarized in Table 1. Details on the inflow turbulence are described in detail in [40, 41].

Table 1: Inputs to the `windsimu` program for generating the Mann turbulence boxes with a turbulence intensity of approximately 6.3% at hub height.

Input	Description
3	Three-dimensional field
3	All three components to be simulated
8192	Number of longitudinal points
128	Number of horizontal points
128	Number of vertical points
9120	Longitudinal length in meters
180	Horizontal length in meters
180	Vertical length in meters
sea	Open sea configuration
11.4	Mean wind speed
90	Height above the surface
0	Spectrum type
-5	Random Seed

2.2. Simulation setup

The simulation setup is comprised of two parts: MIRAS setup and blade geometry setup. The two parts are described in subsections 2.2.1 and 2.2.2, respectively.

2.2.1. MIRAS setup

The NREL 5 MW wind turbine [42] is used as the baseline turbine for the design study presented here. Polar data from [42], which includes three-dimensional (3D) corrections for rotational effects near the blade root region, is employed in the lifting line procedure. Be aware that the blades in MIRAS are simulated as lifting lines and not as 3D surfaces. Therefore, all 3D surface geometries shown in the current article are for visualization purposes. Only backward-swept blades are studied, since backward-sweep have been reported to be beneficial for performance [16, 43]. Similarly, only pre-bended blades where the blade tip points upwind from the rotor plane are considered, since modern wind turbines are designed this way to increase blade-to-tower clearance [44] and reduce blade weight [13].

The MIRAS simulations are aero-elastic and the wind inflow contains synthetic turbulence with mean wind speed of 11.4 m/s. There is no controller modeled, therefore the rotor rotational speed is fixed at 12.1 RPM. The chosen wind speed and rotor rotational speed correspond to the rated operating condition occurring between the variable-speed and constant-power control regions taken from [42]. The real-time simulated is 600 seconds, i.e., 10 minutes,

165 excluding transients in accordance with international standards [45]. Figure 1 depicts the instantaneous velocity magnitude on a two-dimensional plane for a 11.4 m/s turbulent wind inflow MIRAS simulation of the NREL 5 MW wind turbine with straight blades.

Figure 1: Instantaneous velocity magnitude depicted on a two-dimensional plane for a 11.4 m/s turbulent wind inflow MIRAS simulation of the NREL 5 MW wind turbine with straight blades.

To calculate the power (C_P) and thrust (C_T) coefficients, the mean values of aerodynamic power (P_{mean}) and thrust (T_{mean}) in the 600 second period are used, see Equations 1 and 2:

$$C_P = \frac{P_{\text{mean}}}{\frac{1}{2}\rho U_o^3 \pi R^2} \quad (1)$$

$$C_T = \frac{T_{\text{mean}}}{\frac{1}{2}\rho U_o^2 \pi R^2} \quad (2)$$

170 where $U_o = 11.4$ m/s is the wind speed and $\rho = 1.225$ kg/m³ is the density of air at sea level and 15°C. In addition, the rotor radius of the straight blade, $R = 63$ m, is used in the calculation of C_P and C_T for all blade shapes, despite curved shapes having a shorter rotor radius, see subsection 2.2.2. In this study, the C_P and C_T calculations are used only to make the results non-dimensional. What is important here is to measure the change in P_{mean} and T_{mean} in dimensional terms relative to the straight blade.

2.2.2. Blade geometry setup

The following equation is used to define the sweep and pre-bend [46, 47]:

$$y = d_{\text{tip}} \left(\frac{x - x_{\text{start}}}{L_{\text{blade}} - x_{\text{start}}} \right)^\gamma \quad (3)$$

175 where y is the local distance from the pitch axis to the sweep/pre-bend curve, d_{tip} is the distance from the pitch axis to the sweep/pre-bend curve at the

blade tip, x is the local distance along the straight blade measured from the blade root, x_{start} is the position of the beginning of the blade sweep/pre-bend, L_{blade} is the length of the straight blade, and γ is the sweep/pre-bend exponent. Figure 2 depicts the parameters used to define the wind turbine blade sweep and pre-bend. A higher γ places more curvature outboard of the blade [47]. In the studies that follow, d_{tip} [m], x_{start} [m], and γ [-] are design variables. Since both sweep and pre-bend are considered, the total number of variables is then six. For the NREL 5 MW rotor, the blade length is $L_{\text{blade}} = 61.5$ m.

Figure 2: Parameters that define the wind turbine blade sweep and pre-bend.

For all curved blade shapes, the arc length of the blade is equal to the length of the straight blade. In other words, a curved blade will have the same arc length as the straight blade but with a slightly smaller rotor radius. The arc length of the blade should be kept the same to avoid blade extension, which will bias results towards longer blades that produce more power. Figure 2 shows that the axial displacement due to blade bending can be small relative to d_{tip} .

The local radius of the curved blade, $r_c(x)$, is obtained by subtracting the axial displacement from the local radius of the straight blade, $r_s(x)$, as shown in Equation 4:

$$r_c(x) = r_s(x) - \int_0^x 1 - \sqrt{1 - \left[\left(\frac{dy_1(r')}{dr'} \right)^2 + \left(\frac{dy_2(r')}{dr'} \right)^2 \right]} dr' \quad (4)$$

where r' is a dummy variable representing the span along the blade being curved and x represents the original span of the straight blade along the pitch axis. The blade root is represented as $x = 0$ in Equation 4. The variables y_1 and y_2 represent the local distance from the pitch axis to the sweep and pre-bend curve, respectively. Refer to Figure 2 for a depiction of x , $y = y_1$ and $y = y_2$ and references [48, 49] for a derivation of Equation 4.

A sanity check is performed on all curved blades to check that the arc length is equal to the length of the straight blade using Equation 5:

$$\text{arc length} = L_{\text{blade}} = \int_0^{R_c} \sqrt{\left[\left(\frac{dy_1(r_c)}{dr_c} \right)^2 + \left(\frac{dy_2(r_c)}{dr_c} \right)^2 \right]} dr_c \quad (5)$$

where R_c is the tip radius of the curved blade. A script is used to automatically calculate r_c , perform the arc length sanity check, and prepare the blade-geometry input for each MIRAS simulation.

200 *2.3. Neural network setup*

The neural network setup is comprised of two steps: data set preparation and neural network selection. The two steps are described in subsections 2.3.1 and 2.3.2, respectively. A cross-validation exercise is described in subsection 2.3.3.

205 *2.3.1. Data set preparation*

In the present work, neural networks are used to approximate the functions representing the power and thrust coefficients of a wind turbine rotor. A sample data set containing twenty blade shapes is generated using Latin hypercube sampling (LHS) from the pyDOE [50] package in Python to create the inputs P and T required to build the neural networks. The number twenty was the maximum allowed based on computing cost and time limits. The input P contains the parameter values that define the blade geometry in terms of sweep and prebend for the sample data set, while the input T contains the target power and thrust coefficients. Details regarding input variables and coefficient calculations were provided in subsection 2.2.

215 Table 2 contains the parameter values that define the sweep and pre-bend for twenty sample wind turbine blade shapes as well as their performance in terms of C_P and C_T . In addition, parameter values and performance for the straight blade are included for comparison in Table 2. Normalized (Norm.) and dimensional (Dim.) values for the upper and lower limits are included. All parameter values are normalized using the upper and lower dimensional values to prevent bias towards a particular variable due to differences in magnitude. The twenty blade shapes from Table 2 are shown on top of each other in Figures 3 and 4. In Figure 3, the upper limits $d_{\text{tip}} = 3.0\text{ m}$ and $x_{\text{start}} = 0.8L_{\text{blade}}$ are depicted and the circles on the x -axis represent x_{start} for the twenty configurations. In Figure 4, the twenty shapes are used to construct the neural network for MIRAS simulations with synthetic inflow turbulence.

Table 2: Parameter values that define the sweep and pre-bend for twenty sample wind turbine blade shapes and their performance in terms of C_P and C_T compared to the straight blade.

	Sweep			Pre-bend			Performance	
Sample, i [-]	d_{tip}	x_{start}	γ	d_{tip}	x_{start}	γ	C_P	C_T
1	0.675	0.625	0.275	0.925	0.725	0.675	0.472	0.682
2	0.225	0.675	0.675	0.325	0.925	0.175	0.492	0.734
3	0.375	0.425	0.425	0.575	0.375	0.075	0.489	0.718
4	0.475	0.975	0.125	0.225	0.675	0.975	0.480	0.702
5	0.825	0.275	0.525	0.725	0.225	0.125	0.474	0.675
6	0.525	0.475	0.625	0.375	0.275	0.875	0.482	0.700
7	0.625	0.025	0.775	0.125	0.825	0.775	0.484	0.697
8	0.575	0.375	0.175	0.275	0.175	0.575	0.492	0.713
9	0.025	0.825	0.225	0.975	0.425	0.425	0.499	0.764
10	0.175	0.875	0.975	0.825	0.975	0.525	0.490	0.740
11	0.425	0.725	0.575	0.425	0.775	0.625	0.484	0.711
12	0.775	0.775	0.725	0.025	0.075	0.325	0.469	0.680
13	0.975	0.075	0.825	0.625	0.875	0.475	0.466	0.664
14	0.125	0.125	0.075	0.475	0.525	0.825	0.499	0.758
15	0.275	0.925	0.875	0.675	0.125	0.725	0.491	0.741
16	0.725	0.525	0.475	0.075	0.575	0.375	0.475	0.678
17	0.075	0.325	0.925	0.775	0.475	0.025	0.495	0.752
18	0.925	0.175	0.375	0.875	0.025	0.925	0.477	0.684
19	0.875	0.575	0.025	0.175	0.325	0.225	0.486	0.688
20	0.325	0.225	0.325	0.525	0.625	0.275	0.495	0.735
Straight blade	0.000	-	-	0.000	-	-	0.494	0.753
Upper (Norm.)	1	1	1	1	1	1		
Lower (Norm.)	0	0	0	0	0	0		
Upper (Dim.)	3.00	$0.8L_{\text{blade}}$	5.00	3.00	$0.8L_{\text{blade}}$	5.00		
Lower (Dim.)	0.00	0.00	1.00	0.00	0.00	1.00

Figure 3: Twenty distinct sweep and pre-bend configurations shown on top of each other where the circles on the x -axis represent x_{start} for the twenty configurations.

Figure 4: NREL 5 MW wind turbine blade with twenty distinct sweep and pre-bend configurations shown on top of each other in two views (a) and (b).

The upper and lower limits for d_{tip} are 3.0 and 0.0 m, respectively. The upper limit for d_{tip} is affected by transportation requirements and is chosen to be 3.0 m based on the maximum chord of the blade [47]. The upper limit of 3.0 m is also approximately 5% of the blade length, which is within the range used in previous wind turbine blade optimization studies involving sweep and pre-bend [14, 16]. A value of $d_{\text{tip}} = 0.0$ m corresponds to a straight blade and thus represents the lower limit. Upper and lower limits for x_{start} and γ are chosen arbitrarily to capture a wide range without including blade shapes that may be too curved to be manufacturable. Wind turbine blade manufacturing materials, e.g., fiberglass sheets, have limits to their curvature [47].

2.3.2. Neural network selection

The `feedforwardnet` (hereafter `ffwd`), `fitnet`, `cascadeforwardnet`, `newrb`, `newrbe`, and `newgrnn` neural networks from the Neural Network Toolbox of MATLAB [51] were tested. Feed-forward, function-fitting and cascade-forward networks include the process of training, where the neural network's weights and biases are tuned to optimize network performance based on a metric, e.g., mean square error. There is an input parameter specifying the number of neurons in each hidden layer and the training function can also be selected. The `newrb`, `newrbe`, and `newgrnn` are radial basis networks used for function approximation. They create two-layer networks and have an additional input parameter, `spread`, which adjusts the smoothness of the function approximation. The radial basis networks do not involve a training process. Details for each neural

network will not be described here but can be found in the documentation [51].

Since there are many neural networks to choose from with each having a variety of input parameters, only two approaches will be discussed here for brevity. The first approach is the radial basis network, **newrb**. The second is the feed-forward network, **ffwd**, with one hidden layer containing two neurons. The training function is the Levenberg-Marquardt backpropagation [51]. Defaults were used for all other parameters.

The two approaches were selected in particular because they performed reasonably well on the data set in terms of mean squared error as well as for the optimization problem described in subsection 2.4 in comparison with the other methods. In addition, one approach is taken from the group of radial basis networks and the other is taken from the group of networks that involve training. The two approaches are not necessarily the ideal ones for the current study. Detailed comparison studies is outside the scope of this work, but may give better insight into improving the neural network's performance and the blade design results. Subsection 2.3.3 provides insight into the accuracy of the chosen two approaches.

Figure 5 depicts the actual (Act.) vs predicted (Pred.) (a) power coefficient, C_P , and (b) thrust coefficient, C_T , for the twenty blade shapes (sample number 1 to 20) generated from Latin hypercube sampling. The actual values of C_P and C_T are listed in Table 2. The predicted values are obtained from the radial basis (**newrb**) and feed-forward (**ffwd**) neural networks. In Figure 5, **newrb** nearly reproduces the actual values while **ffwd** does not. The consequences of the impreciseness of **ffwd** will be discussed in section 3.

Figure 5: Actual (Act.) vs predicted (a) power coefficient, C_P , and (b) thrust coefficient, C_T , for the twenty blade shapes (sample number 1 to 20) generated from Latin hypercube sampling.

2.3.3. Cross validation

In subsection 2.3.2, the **newrb** and **ffwd** neural networks were selected based on their performance, e.g., smallest mean squared error, on the training data shown in Table 2. However, it is important to test the neural networks to

predict values outside the data set which the neural networks were built upon. The cross-validation test checks whether or not the neural networks has the tendency of overfitting the training data.

A leave-one-out cross-validation (LOOCV) [52] is employed here, where the **newrb** and **ffwd** neural networks are built based on the data shown in Table 2 with one sample left-out. The neural network is then used to predict the one sample that was left-out of the training data. The process is repeated in a loop until each sample in Table 2 has been removed from the training data and tested. The root mean squared error gives an estimate of the prediction accuracy of the **newrb** and **ffwd** neural networks.

The root mean squared error (RMSE) is calculated using Equation 6:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (\text{Act.}^{(i)} - \text{Pred.}^{(i)})^2}{N}} \quad (6)$$

where $\text{Act.}^{(i)}$ and $\text{Pred.}^{(i)}$ are the actual and predicted values of the left-out sample, i , and N is the total number of samples. Referring to Table 2, $N = 20$, $\text{Act.}^{(i)}$ is the C_P or C_T of the left-out sample, and Pred. is the value obtained from the **newrb** or **ffwd** neural network.

Table 3 summarizes the RMSE values from the LOOCV procedure. The RMSE values for the C_P from the **newrb** and **ffwd** neural networks are 0.006 and 0.005, respectively. Similarly, the RMSE C_T values from the **newrb** and **ffwd** are 0.012 and 0.016, respectively. For a neural network with good predictive capabilities, the RMSE should be as close to zero as possible. However, all RMSE values in Table 3 are quite high, considering they are approximately 15% of the maximum and minimum range of C_P and C_T in Table 2. Nevertheless, the **newrb** and **ffwd** neural networks will be used within in an optimization context as a means to hopefully find improved blade designs outside the data set in Table 2. Further work may improve the performance of the neural network's predictive capabilities and in turn the optimized design.

Table 3: Root mean squared error (RMSE) from the leave-one-out cross-validation procedure

	newrb	ffwd
RMSE C_P	0.006	0.005
RMSE C_T	0.012	0.016

Note that the NNs used in the optimization are still the ones built on all the available samples, as there are only $N = 20$ of them. The LOOCV procedure described here is used to demonstrate the validity of the modeling methodology based on NNs.

2.4. Optimization setup

A gradient-based solver, **fmincon**, from the Optimization Toolbox of MATLAB [51] is used to find an improved design close to the vicinity of an initial

starting point taken from the sample data set. The objective and constraint functions are defined in Equations 7 and 8, respectively:

$$\text{maximize } C_{P,\text{Pred.}}(\mathbf{x}) \quad (7)$$

$$C_{T,\text{Pred.}}(\mathbf{x}) - C_{T,\text{straight}} \leq 0 \quad (8)$$

where $\mathbf{x}$ is the vector of design variables and $C_{P,\text{Pred.}}$ and $C_{T,\text{Pred.}}$ are the neural network predictions of the power and thrust coefficients, respectively. The vector $\mathbf{x}$ contains the six design variables discussed in subsection 2.2.2. To optimize the performance of wind turbines, it is desirable to increase C_P as much as possible to within an acceptable limit of the loading on the rotor. The wind turbine rotor loading is modeled using the C_T and is constrained to be less than or equal to the C_T corresponding to the straight blade, $C_{T,\text{straight}}$, in Equation 8. The sample shape from Table 2 with the highest C_P value and that also satisfies the constraint in Equation 8, i.e., sample $i = 20$, is chosen as the initial starting point in `fmincon`. All parameters used in `fmincon` were set to the defaults.

3. Results

The wind turbine blade design results are shown in this section. The results section consists of three subsections: optimization results, blade and rotor loading, and blade and rotor profiles.

3.1. Optimization results

Table 4 summarizes the actual (Act.) and predicted (Pred.) power (C_P) and thrust (C_T) coefficient results and parameter values (d_{tip} , x_{start} , γ) for the blade designs obtained from the optimization. Predicted values are those provided by the radial basis (`newrb`) and feed-forward (`ffwd`) neural networks. Actual values are obtained from running MIRAS simulations using the same parameter values associated with the predicted values. In addition, percent difference with the straight design, $C_{P,\text{diff}}$ and $C_{T,\text{diff}}$, as well as the number of function evaluations (Func. eval.) involved in each optimization are included in the table.

The number of function evaluations involved in each optimization to obtain `newrb`_{Pred.} and `ffwd`_{Pred.} and their associated parameter values are 273 and 359, respectively. The computational time required for each optimization is approximately three seconds. Although more accurate, using a traditional optimizer, e.g., `fmincon`, with the aero-elastic simulator directly and with the same number of function evaluations would have required approximately 200 to 250 days for completion. As mentioned in subsection 2.1, each simulation (or function evaluation) requires approximately seventeen hours of computational time. Using neural networks as a substitute for the numerical model clearly has advantages in terms of computational time for optimization problems.

Table 4: Actual (Act.) and predicted (Pred.) power (C_P) and thrust (C_T) coefficients and parameter values (d_{tip} , x_{start} , γ) for the blade designs obtained from the optimization using the radial basis (**newrb**) and feed-forward (**ffwd**) neural networks.

		Straight	newrb _{Act.}	newrb _{Pred.}	ffwd _{Act.}	ffwd _{Pred.}
Performance	C_P	0.494	0.498	0.502	0.500	0.499
	C_T	0.753	0.753	0.753	0.767	0.753
	$C_{P,\text{diff}}$	0.000%	0.959%	1.603%	1.194%	1.149%
	$C_{T,\text{diff}}$	0.000%	0.017%	0.000%	1.891%	0.000%
Sweep	d_{tip}	0.000	0.169		0.000	
	x_{start}	-	0.390		1.000	
	γ	-	0.001		0.879	
Pre-bend	d_{tip}	0.000	0.398		0.998	
	x_{start}	-	0.413		0.001	
	γ	-	0.424		1.000	
Func. eval.			273		359	

Table 4 shows that the neural networks together with gradient-based optimization produces curved blade shapes with increased C_P between 0.9% and 1.2%, as desired. However, the C_T for the new designs do not necessarily meet the thrust constraint as defined in Equation 8, particularly for **ffwd**_{Act.}. The **newrb**_{Act.} design produces 0.96% more C_P but with 0.02% more C_T than the straight design, which violates Equation 8 by 0.02%. The **ffwd**_{Act.} design produces 1.2% more C_P but with 1.9% more C_T . Although **newrb**_{Pred.} and **ffwd**_{Pred.} satisfy the thrust constraint with $C_{T,\text{diff}} = 0.0\%$, the impreciseness of the neural networks causes **newrb**_{Act.} and **ffwd**_{Act.} to violate the constraint in Equation 8. The impreciseness is also observed for the C_P , where **newrb**_{Pred.} and **ffwd**_{Pred.} deviate from **newrb**_{Act.} and **ffwd**_{Act.}. Nevertheless, the neural networks, at least **newrb**, captures the trend and can be used with an optimizer to produce blade design shapes with improved C_P and a reasonable level of C_T .

Subsection 3.2 describes in detail the blade and rotor loading, which can only be obtained from MIRAS simulations and not from the neural networks. Recall that **newrb**_{Act.} and **ffwd**_{Act.} represent MIRAS simulations using the same parameter values as **newrb**_{Pred.} and **ffwd**_{Pred.}, respectively.

3.2. Blade and rotor loading

Figure 6 compares the aerodynamic (aero.) (a) power, P , and (b) thrust, T , versus time between the straight, **newrb**_{Act.} and **ffwd**_{Act.} blade designs in the 600 second time period. The P and T around time = 200 seconds is shown more closely in Figure 6(c) and (d), respectively. Figure 6 shows that **newrb**_{Act.} and **ffwd**_{Act.} produces a higher P than the straight design. The T for **newrb**_{Act.} is almost the same but higher for **ffwd**_{Act.} compared to the straight design. Note that the rapid variation in P and T is due to the turbulent wind inflow and setting a fixed rotor speed and pitch in the aero-elastic simulator.

(a)

(b)

(c)

(d)

Figure 6: Comparison of aerodynamic (aero.) (a,c) power, P , and (b,d) thrust, T , versus time between the straight, newrb_{Act.} and fwd_{Act.} blade designs in the (a,b) 600 second time period and (c,d) at around time = 200 seconds.

Figure 7 depicts the ten-minute maximum (top), mean (middle) and mini-

mum (bottom) aerodynamic load normal to the rotor plane for the straight (solid
 line), **newrb** (dashed line), and **ffwd** (dotted line) blades in two different y-axis
 scales (a) and (b). Similarly, Figure 8 depicts the same as Figure 7, but with
 the aerodynamic load tangential (tang.) to the rotor plane instead. Figure 9
 depicts the percent difference relative to the straight design for the ten-minute
 maximum, mean and minimum aerodynamic (aero.) load (a) normal and (b)
 tangential (tang.) to the rotor plane for the straight (solid), **newrb**_{Act.} (dashed)
 and **ffwd**_{Act.} (dotted) blade designs.

Figure 7: Ten-minute maximum (top), mean (middle) and minimum (bottom) aerodynamic
 (aero.) load normal to the rotor plane for straight (solid), **newrb**_{Act.} (dashed) and **ffwd**_{Act.}
 (dotted) blade designs in two different y-axis scales (a) and (b).

Figure 8: Ten-minute maximum (top), mean (middle) and minimum (bottom) aerodynamic
 (aero.) load tangential (tang.) to the rotor plane for straight (solid), **newrb**_{Act.} (dashed) and
ffwd_{Act.} (dotted) blade designs in two different y-axis scales (a) and (b).

Figure 9: Percent difference relative to straight design for ten-minute maximum, mean and minimum aerodynamic (aero.) load (a) normal and (b) tangential (tang.) to the rotor plane for straight (solid), **newrb**_{Act.} (dashed) and **ffwd**_{Act.} (dotted) blade designs.

The **newrb** design produces a larger tangential load in Figure 8 and slightly higher normal load in Figure 7 towards the tip, i.e., radius > 50 m, compared to the straight design. The **ffwd** design, though, produces larger tangential and normal loads in the same region compared to both the straight and **newrb** designs. In Figure 8, the slight decrease in tangential load for the **newrb** and **ffwd** designs between radius 0 m and 45 m does not affect the P_{mean} as much as the tangential load for radius above 45 m, since most of the rotor power stems from the outer part of the blade. All trends are present for the maximum, minimum and mean values, and support the findings in Table 4 and Figure 6.

3.3. Blade and rotor profiles

Figure 10 depicts the sweep and pre-bend for the blade designs obtained from the (a) **newrb** and (b) **ffwd** neural networks. The upper limits $d_{\text{tip}} = 3.0$ m and $x_{\text{start}} = 0.8L_{\text{blade}}$ are also depicted and the circles on the x -axis represent x_{start} . Similarly, Figure 11 depicts the (a) **newrb** and (b) **ffwd** designs (solid) compared to the straight blade (mesh). The **newrb** design has increased pre-bend and sweep, while the **ffwd** design has increased pre-bend and no sweep. The pre-bend on the **ffwd** design is also more pronounced than the **newrb** design. The sweep and pre-bend parameter values (d_{tip} , x_{start} , γ) associated with each design are provided in Table 4.

(a)

(b)

Figure 10: Sweep and pre-bend for the (a) **newrb** and (b) **ffwd** designs where the circles on the x -axis represent x_{start} .

(a)

(b)

Figure 11: Blade designs obtained from the (a) **newrb** and (b) **ffwd** neural networks (solid) compared to the straight blade (mesh).

Figure 12 depicts the undeflected rotor state at time zero ($t = 0$) (mesh) and deflected rotor state at $t = 200$ seconds (solid) for the (a) straight, (b) **newrb** and (c) **ffwd** designs. Time zero here means prior to the simulation starting with no wind, while $t = 200$ seconds can be referred in Figure 6. Power and thrust increase for the pre-bended **newrb** and **ffwd** designs due to blade straightening under loading, which exposes the rotor more to the wind inflow compared to the straight design. The blade straightening effect is pronounced towards the outer part of the blade, which leads to a larger normal and tangential aerodynamic load in that region.

Figure 12: Undeflected rotor state at time zero ($t = 0$) (mesh) and deflected rotor state at $t = 200$ seconds (solid) for the (a) straight, (b) **newrb** and (c) **ffwd** designs.

Figure 13 is the same as Figure 12 but depicts the designs towards the outer part of the blade more closely and on top of each other. The **newrb** design appears slightly shorter than the straight and **ffwd** designs because it has a slight sweep. The **ffwd** design has a straighter profile in the deflected state than the **newrb** and straight designs, therefore the **ffwd** design produces the most power and thrust. The effect of sweep in the **newrb** design is not as evident. Figure 9(a) shows that the **newrb** design has a reduction in normal load for radius < 50 m compared to the other designs. The sweep possibly reduces the normal load and the rotor radius. Shortening of the rotor radius reduces the rotor swept area, making the rotor less exposed to the incoming wind. Furthermore, the calculation of thrust is not as influenced by the loading on the outer part of the blade as the power. The combination of the pre-bend of the **newrb** design being not as pronounced and its sweep might explain why the **newrb** violates

the thrust constraint, i.e., Equation 8, by only 0.02% as opposed to the 1.9% of the **ffwd** design. Whether sweep increases or decreases the power is unclear, thus more analysis is required. The trends seen in Figures 12 and 13 support the findings in Table 4 and Figures 6, 7, 8 and 9.

Figure 13: Undeformed rotor state at time zero ($t = 0$) (mesh) and deformed rotor state at $t = 200$ seconds (solid) for the straight, **newrb** and **ffwd** designs.

4. Conclusions

Wind turbine blades with pre-bend and sweep can potentially increase the wind turbine rotor's aerodynamic efficiency. Conventional BEM methods do not model the aerodynamics of curved blades as accurately as higher-fidelity methods, such as CFD and vortex-based methods. However, the computational cost of higher-fidelity methods is larger than methods based on BEM.

This article attempts to design a curved wind turbine blade by using neural networks together with a higher-fidelity aero-elastic simulator under synthetic inflow turbulence. The aero-elastic simulator is based on a vortex particle method where the wind turbine blades are represented by lifting-line theory, while the turbine structural dynamics are modeled by a finite-element multi-body based approach. A neural network together with gradient-based optimization produces a curved design with approximately 1% more power on average with a slight increase in mean thrust of 0.02% compared to the straight blade design. This study demonstrates that using neural networks for wind turbine blade design is effective when considering complex aero-elastic simulation scenarios with turbulent inflow conditions. Further work may improve the performance of the neural network's predictive capabilities and the optimization method, which in turn will provide improved blade designs.

5. Acknowledgements

This research work was funded by the Department of Wind Energy, Technical University of Denmark.

Nomenclature

3D	three-dimensional
Act.	actual
aero.	aerodynamic
AI	artificial intelligence
BEM	blade element momentum
CFD	computational fluid dynamics
Dim.	dimensional
GA	genetic algorithm
HAWC2	Horizontal Axis Wind turbine simulation Code 2 nd generation
LHS	Latin hypercube sampling
LOOCV	leave-one-out cross validation
max/min	maximum/minimum
MIRAS	Method for Interactive Rotor Aero-elastic Simulations
MPI	Message Passing Interface
NN	neural network
Norm.	normalized
NREL	National Renewable Energy Laboratory
Pred.	neural network prediction
RMSE	root mean squared error
RPM	revolutions per minute
tang.	tangential
C_P	power coefficient
$C_{P,\text{diff}}$	difference in C_P compared to the straight design
C_T	thrust coefficient
$C_{T,\text{diff}}$	difference in C_T compared to the straight design
$C_{T,\text{straight}}$	thrust coefficient of the straight design
d_{tip}	distance from pitch axis to sweep/pre-bend curve at blade tip
i	sample number
γ	sweep/pre-bend exponent
L_{blade}	blade length
P	aerodynamic power at a time instant
P_{mean}	mean aerodynamic power in ten-minute time period
ρ	density of air
N	number of samples
R	rotor radius
$r_c(x)$	local radius of curved blade
$r_s(x)$	local radius of straight blade
r'	dummy variable representing the span along blade being curved
R_c	tip radius of curved blade
t	time
T	aerodynamic thrust at a time instant
T_{mean}	mean aerodynamic thrust in ten-minute time period
U_o	wind speed
x	local distance along blade pitch axis measured from blade root

x_{start}	starting position of blade sweep/pre-bend
$\mathbf{x}$	design variable vector
y	local distance from pitch axis to sweep/pre-bend curve
y_1	local distance from pitch axis to sweep curve
y_2	local distance from pitch axis to pre-bend curve
<code>cascadeforwardnet</code>	cascade-forward neural network
445 <code>feedforward, fwd</code>	feed-forward neural network
<code>fitnet</code>	function fitting neural network
<code>fmincon</code>	gradient-based nonlinear programming optimization solver
<code>newgrnn</code>	generalized regression neural network
<code>newrb</code>	radial basis neural network
<code>windsimu</code>	a program for simulation of turbulence in complex terrain

References

- [1] P. Schubel, R. Crossley, Wind turbine blade design, *Energies* 5 (9) (2012) 3425–3449. doi:10.3390/en5093425.
- [2] G. R. Fischer, T. Kipouros, A. M. Savill, Multi-objective optimisation of
450 horizontal axis wind turbine structure and energy production using aerofoil
and blade properties as design variables, *Renewable Energy* 62 (2014) 506–
515. doi:10.1016/j.renene.2013.08.009.
- [3] A. Maheri, S. Noroozi, J. Vinney, Decoupled aerodynamic and structural
design of wind turbine adaptive blades, *Renewable Energy* 32 (10) (2007)
455 1753–1767. doi:10.1016/j.renene.2006.11.004.
- [4] E. M. Fagan, M. Flanagan, S. B. Leen, T. Flanagan, A. Doyle, J. Goggins,
Physical experimental static testing and structural design optimisation for
a composite wind turbine blade, *Composite Structures* 164 (2017) 90–103.
doi:10.1016/j.compstruct.2016.12.037.
- 460 [5] A. Ribeiro, A. Awruch, H. Gomes, An airfoil optimization technique for
wind turbines, *Applied Mathematical Modelling* 36 (10) (2012) 4898–4907.
doi:10.1016/j.apm.2011.12.026.
- [6] M. A. Elfarrar, N. Sezer-Uzol, I. S. Akmandor, NREL VI rotor blade: nu-
merical investigation and winglet design and optimization using CFD, *Wind*
465 *Energy* 17 (4) (2014) 605–626. doi:10.1002/we.1593.
- [7] V. D. Casas, F. L. Pena, R. J. Duro, Automatic design and optimization
of wind turbine blades, in: 2006 International Conference on Computa-
tional Intelligence for Modelling Control and Automation and International
Conference on Intelligent Agents Web Technologies and International Com-
470 merce (CIMCA'06), 2006, pp. 205–205. doi:10.1109/CIMCA.2006.54.
- [8] H.-W. Chi, P.-S. Wu, K. Chen, Y.-H. Jhuo, H.-Y. Wu, Improved aero-
dynamic optimization for the design of wind turbine blades, *Journal of*
Vibroengineering 14 (3) (2012) 1132–1140.

- [9] W. J. Zhu, W. Z. Shen, J. N. Sørensen, Integrated airfoil and blade design method for large wind turbines, *Renewable Energy* 70 (2014) 172–183. doi:10.1016/j.renene.2014.02.057.
- [10] M. Sessarego, N. Ramos-García, H. Yang, W. Z. Shen, Aerodynamic wind-turbine rotor design using surrogate modeling and three-dimensional viscous-inviscid interaction technique, *Renewable Energy* 93 (2016) 620–635. doi:10.1016/j.renene.2016.03.027.
- [11] M. Kaya, M. Elfarra, Optimization of the taper/twist stacking axis location of NREL VI wind turbine rotor blade using neural networks based on computational fluid dynamics analyses, *Journal of Solar Energy Engineering* 141 (1) (2018) 011011. doi:10.1115/1.4041102.
- [12] Z. Sun, M. Sessarego, J. Chen, W. Z. Shen, Design of the OffWindChina 5 MW wind turbine rotor, *Energies* 10 (6) (2017) 777. doi:10.3390/en10060777.
- [13] X. Guo, X. Fu, H. Shang, J. Chen, Integrated aero-structural optimization design of pre-bend wind turbine blades, *Journal of Mechanical Science and Technology* 30 (11) (2016) 5103–5113. doi:10.1007/s12206-016-1028-2.
- [14] M. K. McWilliam, F. Zahle, A. Dicholkar, D. Verelst, T. Kim, Optimal aero-elastic design of a rotor with bend-twist coupling, *Journal of Physics: Conference Series* 1037 (4) (2018) 042009.
- [15] Y. Ding, X. Zhang, An optimal design method of swept blades for HAWTs, *Journal of Renewable and Sustainable Energy* 8 (4) (2016) 043303. doi:10.1063/1.4961588.
- [16] C. Pavese, T. Kim, J. P. Murcia, Design of a wind turbine swept blade through extensive load analysis, *Renewable Energy* 102 (Part A) (2017) 21–34. doi:10.1016/j.renene.2016.10.039.
- [17] G. Venter, Review of Optimization Techniques, *Encyclopedia of Aerospace Engineering*, 2010. doi:10.1002/9780470686652.eae495.
- [18] E. Alpaydin, Introduction to Machine Learning, 2nd Edition, The MIT Press, Cambridge, Massachusetts, USA.
- [19] S. Agatonovic-Kustrin, R. Beresford, Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research, *Journal of Pharmaceutical and Biomedical Analysis* 22 (5) (2000) 717–727. doi:10.1016/S0731-7085(99)00272-1.
- [20] M. Flaszynski, Introduction to Artificial Intelligence, Springer, 2016.
- [21] G. Dede, M. H. Sazli, Speech recognition with artificial neural networks, *Digital Signal Processing* 20 (3) (2010) 763–768. doi:10.1016/j.dsp.2009.10.004.

- [22] D. A. Pomerleau, Efficient training of artificial neural networks for autonomous navigation, *Neural Computation* 3 (1) (1991) 88–97. doi:10.1162/neco.1991.3.1.88.
- 515 [23] J. Feng, Artificial intelligence for wind energy (AI4Wind): A state of the art report, Tech. Rep. DTU Wind Energy E-01806, DTU Wind Energy (February 2019).
- [24] J. P. S. Catalão, H. M. I. Pousinho, V. M. F. Mendes, Short-term wind power forecasting in Portugal by neural networks and wavelet transform, *Renewable Energy* 36 (4) (2011) 1245–1251. doi:10.1016/j.renene.2010.09.016.
- 520 [25] M. N. Eskander, Neural network controller for a permanent magnet generator applied in a wind energy conversion system, *Renewable Energy* 26 (3) (2002) 463–477. doi:10.1016/S0960-1481(01)00140-9.
- 525 [26] A. S. Yilmaz, Z. zer, Pitch angle control in wind turbines above the rated wind speed by multi-layer perceptron and radial basis function neural networks, *Expert Systems with Applications* 36 (6) (2009) 9767–9775. doi:10.1016/j.eswa.2009.02.014.
- [27] M. Schlechtingen, I. F. Santos, Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection, *Mechanical Systems and Signal Processing* 25 (5) (2011) 1849–1875. doi:10.1016/j.ymssp.2010.12.007.
- 530 [28] M. Hansen, *Aerodynamics of wind turbines*, 3rd Edition, Earthscan, 2015.
- [29] T. J. Larsen, A. M. Hansen, *HAWC2 - User manual*, DTU-Risø-R-1597 (July 2015).
- 535 [30] J. M. Jonkman, M. L. Buhl, *FAST User’s Guide*, National Renewable Energy Laboratory, Golden, Colorado., NREL/EL-500-29798 (August 2005).
- [31] A. Li, G. Pirrung, H. A. Madsen, M. Gaunaa, F. Zahle, Fast trailed and bound vorticity modeling of swept wind turbine blades, *Journal of Physics: Conference Series* 1037 (6) (2018) 062012.
- 540 [32] M. Sessarego, N. Ramos-García, W. Z. Shen, Analysis of winglets and sweep on wind turbine blades using a lifting line vortex particle method in complex inflow conditions, *Journal of Physics: Conference Series* 1037 (2) (2018) 022021.
- 545 [33] N. Ramos-García, J. N. Sørensen, W. Z. Shen, Three-dimensional viscous-inviscid coupling method for wind turbine computations, *Wind Energy* 19 (1) (2016) 67–93. doi:10.1002/we.1821.

- [34] M. Sessarego, N. Ramos-García, J. N. Sørensen, W. Z. Shen, Development of an aeroelastic code based on three-dimensional viscous-inviscid method for wind turbine computations, *Wind Energy* 20 (7) (2017) 1145–1170. doi:10.1002/we.2085.
- [35] T. Kim, A. M. Hansen, K. Branner, Development of an anisotropic beam finite element for composite wind turbine blades in multibody system, *Renewable Energy* 59 (2013) 172–183. doi:10.1016/j.renene.2013.03.033.
- [36] N. Ramos-García, M. M. Hejlesen, J. N. Sørensen, J. H. Walther, Hybrid vortex simulations of wind turbines using a three-dimensional viscous-inviscid panel method, *Wind Energy* 20 (11) (2017) 1871–1889, we.2126. doi:10.1002/we.2126.
- [37] J. Mann, The spatial structure of neutral atmospheric surface-layer turbulence, *Journal of Fluid Mechanics* 273 (1994) 141–168. doi:10.1017/S0022112094001886.
- [38] J. Mann, Wind field simulation, *Probabilistic Engineering Mechanics* 13 (4) (1998) 269–282. doi:10.1016/S0266-8920(97)00036-2.
- [39] J. Mann, windsimu: a program for simulation of turbulence in complex terrain, Technical University of Denmark (November 19, 1999).
- [40] J. Rasmussen, J. Walther, Particle methods in bluff body aerodynamics, Ph.D. thesis (2011).
- [41] N. Ramos-García, H. J. Spietz, J. N. Sørensen, J. H. Walther, Vortex simulations of wind turbines operating in atmospheric conditions using a prescribed velocity-vorticity boundary layer model, *Wind Energy* 21 (11) (2019) 1216–1231. doi:10.1002/we.2225.
- [42] J. Jonkman, S. Butterfield, W. Musial, G. Scott, Definition of a 5-MW reference wind turbine for offshore system development, Tech. rep. (Feb 2009). doi:10.2172/947422.
- [43] D. Verelst, T. Larsen, Load consequences when sweeping blades - a case study of a 5 MW pitch controlled wind turbine, Tech. Rep. Risø-R-1724(EN), Risø National Laboratory for Sustainable Energy, DTU Wind Energy (August 2010).
- [44] C. Bak, F. Zahle, R. Bitsche, T. Kim, A. Yde, L. Henriksen, M. Hansen, J. Blasques, M. Gaunaa, A. Natarajan, The DTU 10-MW reference wind turbine (2013).
- [45] International Standard IEC 61400-1: Wind turbine - Part 1: Design requirements, 3rd Edition, no. IEC 61400-1:2005(E), International Electrotechnical Commission, 2005.

- 585 [46] M. Zuteck, Adaptive blade concept assessment: Curved planform induced twist investigation, Tech. Rep. SAND2002-2996, Sandia National Laboratories (October 2002).
- [47] S. Larwood, C. P. van Dam, D. Schow, Design studies of swept wind turbine blades, *Renewable Energy* 71 (2014) 563–571. doi:10.1016/j.renene.2014.05.050.
- 590 [48] J. Jonkman, Modeling of the UAE wind turbine for refinement of FAST_AD, Tech. Rep. NREL/TP-500-34755, National Renewable Energy Laboratory, Golden, Colorado 80401-3393 (December 2003).
- [49] S. W. R.E. Wilson, P. Heh, Technical and User’s Manual for the FAST_AD Advanced Dynamics Code, Oregon State University, Corvallis, Oregon 97331-6001, OSU/NREL report 99-01 Edition (May 1999).
- 595 [50] pyDOE: The experimental design package for Python (accessed Oct. 24, 2018).
URL <https://pythonhosted.org/pyDOE/>
- 600 [51] The MathWorks Inc., Natick, MA, MATLAB version 9.4.0 (R2018a) (2018).
URL <https://www.mathworks.com/>
- [52] C. Sammut, G. I. Webb (Eds.), *Leave-One-Out Cross-Validation*, Springer US, Boston, MA, 2010, pp. 600–601. doi:10.1007/978-0-387-30164-8_469.
- 605