Journal of Physics: Conference Series

Machine learning to rapidly predict turbine yaw angles for wake steering

Journal of Physics: Conference Series **2767** (2024) 082011 doi:10.1088/1742-6596/2767/8/082011

Machine learning to rapidly predict turbine yaw angles for wake steering

Andrew P. J. Stanley, Tim Mulder, Bart Doekemeijer, and Jasper Kreeft

Shell International Exploration and Production Inc., Houston, TX 77079, USA

E-mail: PJ.Stanley@shell.com

Abstract. Wake steering is an important control strategy to boost power production of a wind farm. Because of computational expense and problem complexity, wind farm layouts are typically optimized assuming they will operate without wake steering. However, performance gains are possible by simultaneously optimizing wind farm layout and the yaw angles for wake steering. In this paper, we present a method to train a machine learning model to predict turbine yaw angles as a function of their position relative to other turbines in the wind farm and the inflow wind speed. This model is able to predict turbine yaw angles with an R^2 value of 0.98. The model also produces turbine yaw angles with wind farm power production that is similar to yaw angles that have been directly optimized. This method to rapidly compute optimal turbine yaw angles for wake steering enables control co-design of wind farms and the associated performance increase.

1. Introduction

An important part of wind farm design and operation is to minimize wake interactions between turbines in order to increase the total farm power production. During the wind farm design stage, this is done with wind farm layout optimization, where the location of turbines in the wind farm is optimized to minimize interactions between turbines and maximize the plant performance [1, 2]. During wind plant operation, control strategies can be used to further decrease wake interactions between turbines. Of the possible control strategies, wake steering through active yaw control is one of the most well know and furthest developed methods [3, 4, 5].

Historic and typical current practice is to design wind farm layouts assuming that they will be operated without wake steering. After fixing the layout, additional gains are achieved through wake steering. Unfortunately, the wind farm layout and yaw control co-design problem is challenging because the computational expense increases disproportionately as the number of wind turbines increases [6]. However, overcoming this "curse of dimensionality" could enable performance improvements by simultaneously optimizing wind farm layout and yaw angles for wake steering. This is a specific instance of control co-design—simultaneously optimizing aspects of system design and control such that the appropriate interactions between the two are accounted for [7].

A recent paper simultaneously solves the wind farm layout and yaw control optimization problem by creating an implicit relationship between wind farm layout and optimal yaw angles [6]. The relationship simplifies and approximates the optimal yaw angle of a turbine as a function of its relative position to other turbines in a wind farm. The geometry-driven optimal

Journal of Physics: Conference Series **2767** (2024) 082011 doi:10.1088/1742-6596/2767/8/082011

yaw angle approximation enables better wind farm layouts to be found because it allows the layout to be optimized while accounting for wake steering through yaw control. This control co-design optimization can be done without prohibitive computational expense because the implicit relationship between optimal yaw angles and turbine locations eliminates any additional computational cost that comes from explicit yaw angle optimizations within each iteration or coupled with the wind farm layout optimization. While this geometry-based relationship enables better wind farm layouts to be found by considering yaw control during the layout optimization, it is a strong simplification. The yaw angles predicted by this relationship underperform quite significantly compared to actual model-based yaw optimizations.

In this paper, we build on the previous work by using machine learning to predict optimal turbine yaw angles as a function of turbine locations. Compared to yaw angles that have been optimized with (computationally intensive) model-based methods, the model that we created for this paper is able to predict optimal turbine yaw angles accurately across a wide range of operating conditions.

2. Methodology

The premise of this research is that optimal turbine yaw angles for wake steering can be determined implicitly from the location of wind turbines in the wind farm and the wind resource. This section presents the methods we used to determine this relationship and how it performs compared to more computationally expensive, traditional yaw-angle optimizations.

2.1. Training Data Set

The first step of this work was to create the training data set from which we would create our machine learning model. To create this data set, we generated several grid wind farm layouts with five rows and five columns each. The rows were aligned east to west and the columns north to south, meaning there was no offset between subsequent rows or columns. We varied the spacing between rows and columns of each of these turbine grids between five and ten rotor diameters at one rotor diameter intervals. For each one of these wind farms, we then optimized each turbine yaw angle to maximize wind farm power production for wind directions between zero and ninety degrees at one degree intervals, for freestream wind speeds between five and fifteen meters per second at 1 meter per second intervals. The total result was over 900,000 individually optimized turbine yaw angles with which we trained our model.

To generate this training data, we used the Scipy's SLSQP gradient-based optimizer with finite-difference gradients within the Pyoptsparse framework [8, 9]. We modeled the wind farm power production with FLORIS version 3.2 using the cumulative-curl wake model, square root of the sum-of-squares for wake combinations, and the NREL 5-MW reference turbine [10, 11, 12]. Any engineering wake model should be able to be used to train a model with the method demonstrated in this paper, given that it can be used to optimize yaw misalignment angles for wake steering. Even higher fidelity wake models should work as well given that the optimal yaw angle of a turbine as a function of its location does not have significant stochasticity.

2.2. Data Preprocessing

For this paper, we follow similar assumptions as the simplified geometric yaw relationship previously mentioned [6]; that the yaw angle of a turbine of interest can be determined from that streamwise and cross-stream distance to any downstream turbines that are waked by the turbine of interest. There are a number of studies that use a similar approach to predict power outputs and wind speed deficits [13, 14, 15]. These distances are shown in Fig. 1A and B for one turbine to the nearest waked turbine. Figure 1A shows a group of five turbines with the wind coming from the left. To train a model to determine the yaw angle of the orange turbine, we used the streamwise distance (dx) and the cross-stream distance (dy) to the N nearest waked turbines, one of which is shown in blue.

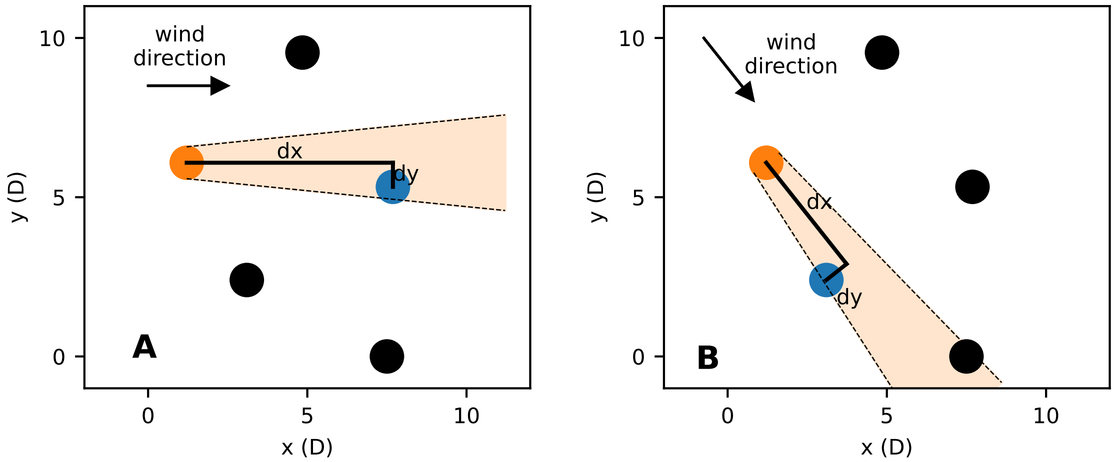

Figure 1. A representation of the geometric preprocessing used to develop our machine learning model. The panels show how the streamwise (dx) and cross-stream (dy) distances to downstream waked turbines is defined for two different wind directions.

The black circles represent the other wind turbines in this cluster. In Fig. 1A, notice that there are two turbines closer to the orange turbine, but these are not waked and therefore do not affect the yaw angle of the orange turbine. To determine if a turbine was waked, we assumed that a wake spreads behind a turbine with a wake radius of $r_{\text{wake}} = \alpha x + r_{\text{turbine}}$, where r_{wake} is the radius of the wake, α is the wake spread constant, x is the streamwise distance downstream of the waking turbine, and r_{turbine} is the radius of the waking turbine [16]. By testing a few different wake spread constants, we found best results with a value of $\alpha = 0.08$, although better performance may be achieved with a more exhaustive search. We defined a turbine as "waked" if any part of the rotor overlaps with the wake. We recognize that this definition of a waked turbine is fairly arbitrary and has the potential to affect the quality of our model; this is an area for further development. Note that for our purposes, the wake radius given in this paragraph was only used to determine if a downstream turbine was waked and was not used to calculate the wind farm power production.

Just as the optimal yaw angle for a turbine changes depending on the incoming wind direction, the relative distance to downstream waked turbines from a turbine of interest also changes. Figure 1B shows the same group of five turbines as Fig. 1A, but with the wind coming from the upper left corner. For this wind direction, the nearest waked turbine is different than in Fig. 1A. Additionally, there is one more turbine in the wake of the orange turbine for this wind direction. We preprocessed the training data to calculate the streamwise distance and the cross-stream distance from each turbine to the five nearest downstream waked turbines. We believe that five would be sufficient for any wind farm layout, but this should be confirmed for large wind farms. As stated in the previous section, these distances were associated with the optimal yaw angle from the continuous gradient-based optimizer.

The preprocessed data was organized into the training matrix shown in Eq. 1. We used an N value of five, meaning that we used the first five downstream waked turbines to determine the yaw angle of a turbine of interest. Because our training data used only 5-by-5 turbine grids, this means that all downstream waked turbines were accounted for. In Eq. 1, the first N columns shown in **red** represent the streamwise distance from the turbine of interest to the downstream waked turbines. These columns are sorted from the nearest to furthest waked turbine, with the nearest waked turbine in the first column. The next N columns shown in **teal** represent the cross-stream distance from the turbine of interest to the downstream waked turbines. These columns are sorted like the first N with the nearest waked turbines on the left and the furthest on the right. Up to this point in the matrix, empty values are filled with 0. That is, in the case that fewer than N turbines are waked by the upstream turbine of interest, the streamwise and cross-stream distances are filled with a zero value. Because zero is meaningful in this context (a zero cross-stream distance would mean the downstream turbine is fully waked by the upstream turbine), the next N columns shown in **violet** are Boolean values that indicate if the indicated downstream turbine is present. If a waked turbine of the appropriate index exists, the value of the corresponding W entry is 1; else the value is -1. Finally, we expect the yaw control angle for an upstream turbine to be partially dependent on the wind speed. The final two columns shown in **cyan** represent the inflow wind speed to the turbine of interest and the freestream wind speed.

$$\begin{bmatrix} dx_{1,1} & dx_{1,2} & \cdots & dx_{1,N} & dy_{1,1} & dy_{1,2} & \cdots & dy_{1,N} & W_{1,1} & W_{1,2} & \cdots & W_{1,N} & u_1 & u_{1,\text{ free}} \\ dx_{2,1} & dx_{2,2} & \cdots & dx_{2,N} & dy_{2,1} & dy_{2,2} & \cdots & dy_{2,N} & W_{1,1} & W_{1,2} & \cdots & W_{1,N} & u_2 & u_{2,\text{ free}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ dx_{T,1} & dx_{T,2} & \cdots & dx_{T,N} & dy_{T,1} & dy_{T,2} & \cdots & dy_{T,N} & W_{T,1} & W_{T,2} & \cdots & W_{T,N} & u_T & u_{T,\text{ free}} \end{bmatrix} \tag{1}$$

2.3. Machine Learning Model

Using the matrix given in Eq. 1 and the array of optimized yaw angles from the training data, we trained a machine learning model to find optimal turbine yaw angles for wake steering as a function of the wind resource and the turbine locations relative to each other. We used the multilayer perceptron neural-network, a feed-forward network with fully-connected layers, in the Scikit-learn python package to create our model [17]. This model was single-output, meaning we predicted the yaw angle of every turbine independently. In the neural network, we normalized all of the inputs and outputs between -1 and 1. We used five hidden layers with 100 neurons each and 2000 as our maximum number of iterations. For all other hyper-parameters we used the default values, which include a constant learning rate of 0.001 and the stochastic gradient-based Adam optimizer. We tested up to ten hidden layers with different numbers of neurons, and found the best results with this combination of hyper-parameters. We trained the model with 80% of our data and reserved the final 20% for testing.

The output of the neural network was a trained model with an out-of-sample R^2 value of 0.98 across the entire data set. This value may be skewed by the large number of turbines with zero yaw and will be investigated further in the continuation of this research. The out-of-sample model performance is shown in Fig. 2. In this figure, each point represents the yaw angle from the training data versus the yaw angle predicted by our trained model for a single turbine. Visually, we can see that the model well predicts the turbine yaw angles. However, there is a streak where the training data has an optimized yaw angle of zero but the model predicts some yaw misalignment. This could be a shortcoming of the model, a shortcoming of the optimization used to generate the training data, or simply an artifact of the multimodal nature of this design space where multiple different yaw angles can result in almost identical wind farm power production. One thing to remember when evaluating the performance of this model is overfitting, as large neural networks tend to overfit. Our high out-of-sample R^2 value lends confidence that this predictive model doesn't contain many local extremes. Even so, any

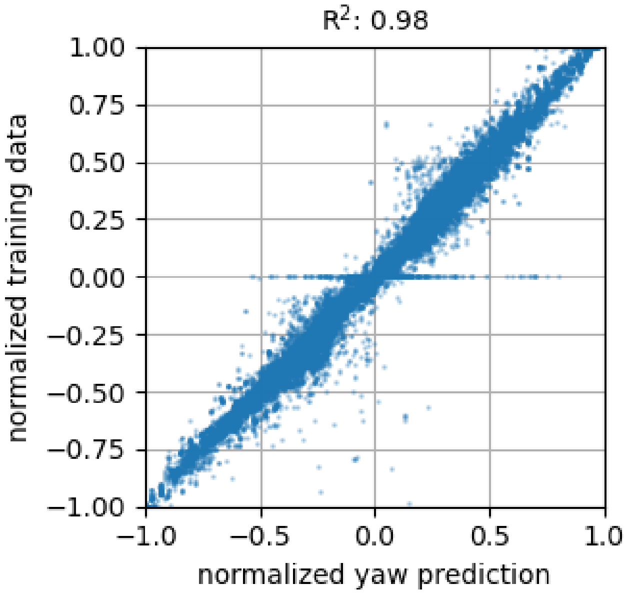

Figure 2. Performance of our trained model to predict optimal yaw angles compared to the training data. Each small dot represents the yaw angle of a single turbine.

overfitting will be important to better understand before implementing this model in a layout optimization problem.

3. Model Performance

While it may be important for some applications for the trained model to be able to predict the actual yaw angles achieved from a direct yaw optimization, our end goal is not the yaw angles themselves but to produce a full wind farm with favorable power production. Figure 3 shows how the wind farm power production with wake steering from our model compares to the baseline power and to the farm power with direct yaw angle optimization. The results in this figure are for a grid wind farm with a column spacing of 7.5 rotor diameters and row spacing of 5.5 rotor diameters. These grid spacings are within the upper and lower bounds of the training data, but has a unique layout that had not yet been tested with the model.

The columns of Fig. 3 show the baseline power production of the wind farm without wake steering, the percent power improvement from wake steering from our model and directly optimized yaw angles, and the percent difference between the power production with our model and from the directly optimized yaw angles, from left to right respectively. As labeled, each row shows a different freestream wind speed, 13, 10, and 7 meters per second, from top to bottom respectively. Indicated by the plateau in power production for the 13 meter per second baseline power, this is above the rated wind speed of the turbines.

The most important takeaway from Fig. 3 can be seen in the middle column. The power gain of wake steering over the baseline power is very similar between the yaw angles from our model and directly optimized yaw angles. Notice that the highest gains from wake steering in the middle column occur at the wind directions with lower baseline power shown in the left column. At these wind directions, the turbines in the grid are aligned with the incoming wind, meaning wake losses are high and more benefit is possible by reducing the wake interactions

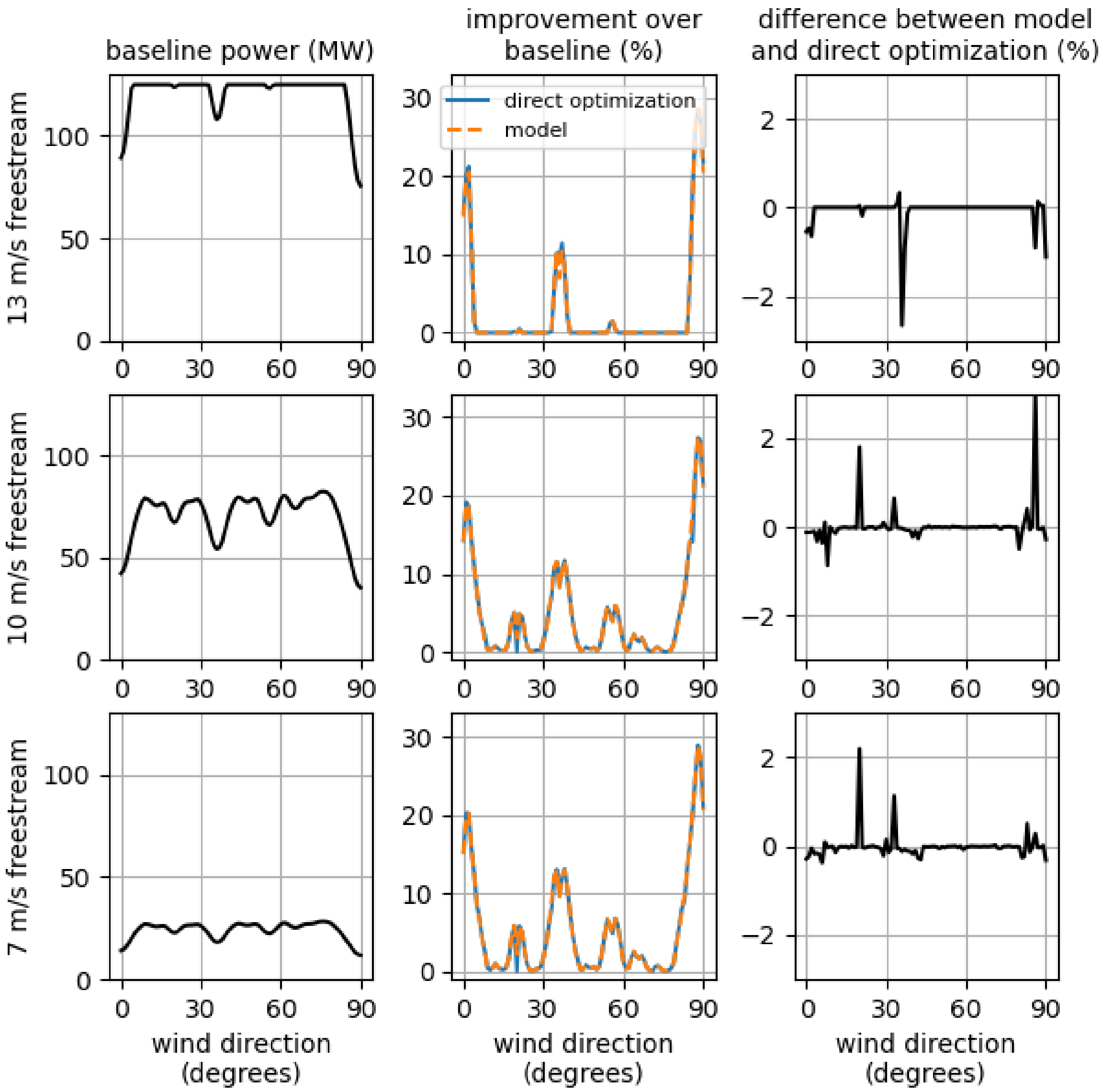

Figure 3. The power production achieved with wake steering using our model prediction compared to directly optimized yaw angles. This figure shows the power for a grid wind farm with a column spacing of 7.5 rotor diameters and row spacing of 5.5 rotor diameters, which is out of sample. The left column shows the baseline power production of the wind farm as a function of the wind direction. The middle column shows the percent power improvement achieved with wake steering using our model and with directly optimized yaw angles. The right column shows the comparison between the power production from our model versus the directly optimized yaw angles. Positive values in this right column indicate that our model produced a better power, negative values indicate that the directly optimized yaw angles performed better. Each row indicates a different freestream wind speed, with 13, 10, and 7 meters per second from top to bottom, respectively.

through wake steering.

A second takeaway is seen in the right column of Fig. 3. This column shows the percent difference between wind farm power production with our trained model and with directly optimized yaw angles for wake steering. Positive values in this column indicate that our model performs better; negative indicate that the direct optimization performs better. For the majority of wind directions the difference in power performance between the two wake steering methods is close to zero. There are a few spikes of larger difference in performance, but across all of the wind speeds these spikes seem relatively balanced between our model outperforming the direct optimization and the other way around.

4. Conclusions

In this paper, we have demonstrated a method to train a machine learning model to predict optimal turbine yaw angles for wake steering. The resultant trained model predicts the optimal yaw angle of a wind turbine as a function of its location relative to other turbines in the wind farm and the inflow wind speed to the turbine. For the results in this paper, we trained and tested the model with grid wind farms of with varied row and column spacing, different inflow wind directions, and freestream wind speeds between 5–15 meters per second. The out-of-sample R^2 value for this trained model was 0.98 in predicting the turbine yaw angles.

Additionally, we have demonstrated that the power production with wake steering from yaw angles predicted by our trained model compares extremely well to the wind farm power using directly optimized yaw angles. The comparison that we show in this paper is for a new wind farm layout that was not used in the training data.

While this method provides a fast evaluation to determine yaw misalignment for wake steering, it may not be the best option to determine set points during wind farm operation. Rather, this new method can be used to enable wind farm layout and yaw control co-design. The trained model requires only a single wake model function call (to evaluate the inflow wind speeds to each turbine) to determine the yaw misalignment angles for wake steering. This allows wind farm layouts to be optimized while assuming they will operate with yaw control, a process that was previously very challenging or completely prohibited by the large number of design variables and associated computational expense.

5. Future Work

There are some important continued steps for this research, some of which have been mentioned in previous sections. First is improvement of the model itself. A more exhaustive search of hyper-parameters in the neural network, exploring additional or different parameters to train the model, and other improvements could lead to a better yaw angle prediction. Second is to test our model for a wide range of wind farm layouts, turbine types, and other wind farm setups to test the model validity in these different scenarios and make improvements where necessary. One important cases to explore are larger wind farms with more turbines and non-grid wind farm layouts. Also included in this future work is better bounds for maximum and minimum yaw misalignment angles. We have assumed maximum and minimum angles of 30 and -30 degrees, which may be outside the acceptable bounds of operation from a structural loading perspective. Third and last is to use this model to optimize wind farm layouts while accounting for yaw control. This includes ensuring the model does not overfit, exploring which types of wind farms benefit the most from control co-design, understanding the performance gains possible through control co-design, and comparing the results with the previous simple geometric yaw model [6] for this same purpose.

Acknowledgements
This work has been supported by the SUDOCO project, which receives funding from the European Union's Horizon Europe Programme under the grant agreement No. 101122256.

References

[1] Perez-Moreno S S, Dykes K, Merz K O and Zaaijer M B 2018 *Journal of Physics: Conference Series* vol 1037 (IOP Publishing) p 042004
[2] Thomas J J, McOmber S and Ning A 2022 *Wind Energy* **25** 678–699
[3] Howland M F, Lele S K and Dabiri J O 2019 *Proceedings of the National Academy of Sciences* **116** 14495–14500
[4] Fleming P, King J, Dykes K, Simley E, Roadman J, Scholbrock A, Murphy P, Lundquist J K, Moriarty P, Fleming K *et al.* 2019 *Wind Energy Science* **4** 273–285
[5] Fleming P, King J, Simley E, Roadman J, Scholbrock A, Murphy P, Lundquist J K, Moriarty P, Fleming K, van Dam J *et al.* 2020 *Wind Energy Science Discussions* **2020** 1–24
[6] Stanley A P, Bay C J and Fleming P 2023 *Wind Energy Science* **8** 1341–1350
[7] Garcia-Sanz M 2019 *Advanced Control for Applications: Engineering and Industrial Systems* **1** e18
[8] Virtanen P, Gommers R, Oliphant T E, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt S J, Brett M, Wilson J, Millman K J, Mayorov N, Nelson A R J, Jones E, Kern R, Larson E, Carey C J, Polat İ, Feng Y, Moore E W, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero E A, Harris C R, Archibald A M, Ribeiro A H, Pedregosa F, van Mulbregt P and SciPy 10 Contributors 2020 *Nature Methods* **17** 261–272
[9] Wu N, Kenway G, Mader C A, Jasa J and Martins J R R A 2020 *Journal of Open Source Software* **5** 2564
[10] NREL 2022 Floris. version 3.2 URL `https://github.com/NREL/floris/releases/tag/v3.2`
[11] Bay C J, Fleming P, Doekemeijer B, King J, Churchfield M and Mudafort R 2022 *Wind Energy Science Discussions* **2022** 1–28
[12] Jonkman J, Butterfield S, Musial W and Scott G 2009 Definition of a 5-mw reference wind turbine for offshore system development Tech. rep. National Renewable Energy Lab.(NREL), Golden, CO (United States)
[13] Yan C, Pan Y and Archer C L 2019 *Wind Energy* **22** 1421–1432
[14] Bleeg J 2020 *Journal of Physics: Conference Series* vol 1618 (IOP Publishing) p 062054
[15] Dou S and Dimitrov N 2022 *Journal of Physics: Conference Series* vol 2265 (IOP Publishing) p 032069
[16] Jensen N O 1983 *A note on wind generator interaction* (Risø National Laboratory)
[17] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M and Duchesnay E 2011 *Journal of Machine Learning Research* **12** 2825–2830