Wind turbine blade design with airfoil shape control using invertible neural networks

John Jasa, Andrew Glaws, Pietro Bortolotti, Ganesh Vijayakumar and Garrett Barter

National Renewable Energy Laboratory, Golden, CO 80401, USA

E-mail: john.jasa@nrel.gov

Abstract. Wind turbine blade design is a highly multidisciplinary process that involves aerodynamics, structures, controls, manufacturing, costs, and other considerations. More efficient blade designs can be found by controlling the airfoil cross-sectional shapes simultaneously with the bulk blade twist and chord distributions. Prior work has focused on incorporating panel-based aerodynamic solvers with a blade design framework to allow for airfoil shape control within the design loop in a tractable manner. Including higher fidelity aerodynamic solvers, such as computational fluid dynamics, makes the design problem computationally intractable. In this work, we couple an invertible neural network trained on high-fidelity airfoil aerodynamic data to a turbine design framework to enable the design of airfoil cross sections within a larger blade design problem. We detail the methodology of this coupled framework and showcase its efficacy by aerostructurally redesigning the IEA 15-MW reference wind turbine blade. The coupled approach reduces the cost of energy by 0.9% compared to a more conventional design approach. This work enables the inclusion of high-fidelity aerodynamic data earlier in the design process, reducing cycle time and increasing certainty in the performance of the optimal design.

1. Introduction

Wind turbine blades are designed to meet a variety of objectives and constraints in the fields of aerodynamics, aeroacoustics, structures, controls, materials, manufacturing, and eventually costs. Wind turbine blade design is therefore a highly multidisciplinary process where trade-offs are constantly weighed against one another. Recent advances in computational modeling have fueled the development of integrated turbine analysis frameworks including HAWTOpt2 [1], Cp-Max [2], ATOM [3], and others. As part of these frameworks, aeroservoelastic simulations are used to estimate blade loads, with a common representation of the blades as a combination of elastic beam and aerodynamic lifting line. International wind turbine design standards, such as the International Electrotechnical Commission standard IEC 61400-1, are set at this fidelity level.

In a lifting line model, airfoils are represented as look-up tables of lift, drag, and moment coefficients as a function of angle of attack and Reynolds number. These coefficients can be obtained via wind tunnel tests or by running numerical solvers. During design, a set of airfoils is typically preselected and placed along the blade span. Airfoil shapes are then often kept frozen, with the optimizer controlling either their position along span [4] or the distribution of relative thickness, which is parameterized similar to chord and twist [1, 3]. These approaches avoid the

need to embed an airfoil solver within the optimization loop, but their results heavily depend on the initial set of airfoils assumed by the user. Also, often airfoils cannot change their order along span, further limiting the solution space.

The limitation of using fixed airfoil designs has been overcome in the past by parameterizing the shape of the airfoils with the shape control points exposed to an optimizer. This was made tractable by integrating lower-fidelity airfoil integral boundary layer methods, such as the one implemented in the popular XFOIL model [5]. A notable example coupling a blade-element momentum solver and XFOIL is presented by Zhu et al. [6], where authors adopted perturbation functions to parameterize the airfoil shapes. A second example is presented by Sartori et al. [7], where airfoil shapes are represented with Bezier curves [8]. At each station where the airfoil is designed, 13 design variables are passed to the optimizer. The framework adopted an aerostructural approach and highlighted interesting trade-offs, such as the adoption of flatback airfoils in the inboard regions of the blade.

Integrating higher-fidelity approaches, such as 2D computational fluid dynamics (CFD), within an iterative design optimization is generally computationally impractical, especially if using finite differences for gradient estimation. An exception is presented by Barrett et al. [9], where a free-form shape control approach is coupled to both XFOIL and Reynolds-averaged Navier-Stokes (RANS) CFD during the optimization. This approach was made computationally tractable thanks to analytic gradients, though the cost of the RANS CFD still greatly increased the optimization cost. Frameworks implementing 3D blade-resolved simulations exist for analysis purposes and researchers have made inroads to inclusion within a design optimization, although the high computational requirements and software complexities have so far limited the application to niche design scenarios [10, 11].

Alternative approaches have focused on using surrogate models or response surfaces to characterize the airfoil design space using fewer computational resources [12–15]. Li et al. [12] parameterized airfoil shapes using eight b-spline control points and then trained a polynomial response surface on airfoil performance from XFOIL with those eight points as inputs. Their work was limited by the ability to accurately fit the data using the response surface and by XFOIL's fidelity level. Han et al. [13] used a Kriging-based surrogate trained on RANS CFD data to enable shape optimization for 2D airfoil cross sections. This was limited to only 2D airfoil analysis and was not extended to 3D shape or full blade optimization.

Recently, neural networks (NNs) have been studied as possible tools to decrease the cost of airfoil shape control into the design process [14, 16, 17]. Oh [14] used an artificial neural network trained using XFOIL to optimize airfoil shapes and compare them to response surface methods. They found that the NN accuracy was better than the response surface for complex data sets, but not for smaller dimensional spaces. Additionally, this work was limited to only airfoil performance and did not extend to full blade design. Bouhlel et al. [17] used a gradient-enhanced NN to predict subsonic and transonic airfoil performance, leading to drastically lower computational costs for airfoil optimization. Sessarego et al. [16] used 3D aeroelastic analysis to train a neural network to design curved turbine blades. They used a vortex particle method and suggested further work in neural network accuracy and optimization method improvements.

Until now, CFD-trained neural networks have not yet been combined with full wind turbine design tools. Doing this would allow for full blade design control in a computationally tractable manner while outputting relevant cost metrics. In this work we extend the state of the art by exploiting recent advancements in artificial NNs based on conditional generative adversarial networks [18, 19]. We train a NN on 2D airfoil CFD results, and by inverting the NN, we gain the ability to rapidly generate 2D airfoil designs for desired performance at high fidelity and low computational cost. Next, we couple the NN-based airfoil design method to a wind turbine design framework. This novel design approach unlocks the performance improvements needed to accurately optimize blades of increasing size and flexibility in a practical manner.

This paper continues by explaining the methodology behind the invertible neural network (INN) training and implementation and the coupled design framework in Sec. 2. With the methodology established, Sec. 3 details the optimization problems studied and dissects the resulting airfoil and blade designs. Lastly, Sec. 4 discusses the conclusions, the main takeaways of this work, and how this design approach could integrate into other workflows.

2. Methodology

2.1. INN training and implementation

Our NN-based design uses a specialized INN architecture to accelerate design iterations. The INN framework learns a bijective mapping between input airfoil shapes, operating condition, and output aerodynamic and structural considerations [20, 21]. Given a forward mapping $\mathbf{f} = f(\mathbf{x})$, the INN approximates the function $f(\mathbf{x}) \approx f_{INN}(\mathbf{x}; \mathbf{\Theta})$, where $\mathbf{\Theta}$ represents the collection of trainable network weights and biases. The INN architecture comprises specialized invertible blocks [22]. These blocks give the network a closed-form inverse function, f_{INN}^{-1}, while sharing all of the model parameters, $\mathbf{\Theta}$, between the forward and inverse evaluations of the network. A key component of ensuring invertibility is the use of latent space variables, $\mathbf{z}$, that parameterize the generally ill-posed inverse mapping f_{INN}^{-1}. A final component of the INN is the collection of simulation parameters, $\mathbf{y}$, that help characterize the input-output relationship but are not considered design or unknown quantities. These quantities are allowed to pass through the INN and can be specified on either forward or inverse evaluations of the model. Figure 1 contains a schematic representation of the INN architecture.

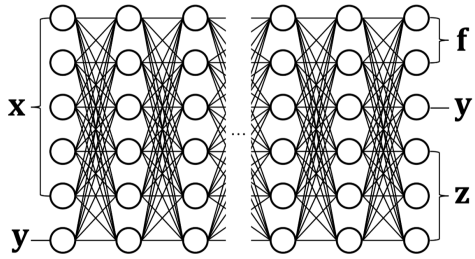

Figure 1. A schematic of the INN architecture that learns a bijective mapping between model inputs, $\mathbf{x}$, and outputs, $\mathbf{f}$. Invertibility is ensured by the use of latent variables, $\mathbf{z}$. $\mathbf{y}$ contains simulation parameters that can be specified in either direction.

For the airfoil design problem, we define the input vector $\mathbf{x}$ to be the collection of parameters defining the airfoil shapes as well as the operational angle of attack α. We use a data-driven Grassmannian shape representation of the airfoils that identifies dominant modes of spatial variation and builds a low-dimensional parameterization over those modes [23, 24]. Figure 2 shows how airfoils change along each of the six parameters in $\mathbf{x}$ that define the shape. The angle of attack, α, is the final entry of the input vector $\mathbf{x}$ and can vary from -4° to 20°. The collection of outputs, $\mathbf{f}$, under consideration include the coefficient of drag (C_D), the ratio of lift to drag (L/D), the stall margin ($\alpha_{stall} - \alpha$), and the maximum thickness-to-chord ratio (t/c). The aerodynamic quantities are obtained for each airfoil shape using CFD simulations generated by the HAM2D solver [25]. For this work, the simulation parameters, $\mathbf{y}$, only contain the Reynolds number (Re), which ranges from $[3, 12] \times 10^6$. The final component of the network is the latent variables, $\mathbf{z}$, which are assumed to be realizations of a standard Gaussian distribution.

The data used to train the neural network as well as to build the Grassmannian basis come from perturbations to nine baseline airfoils from the National Renewable Energy Laboratory's (NREL's) 5-MW wind turbine blades and the International Energy Agency Wind Technology

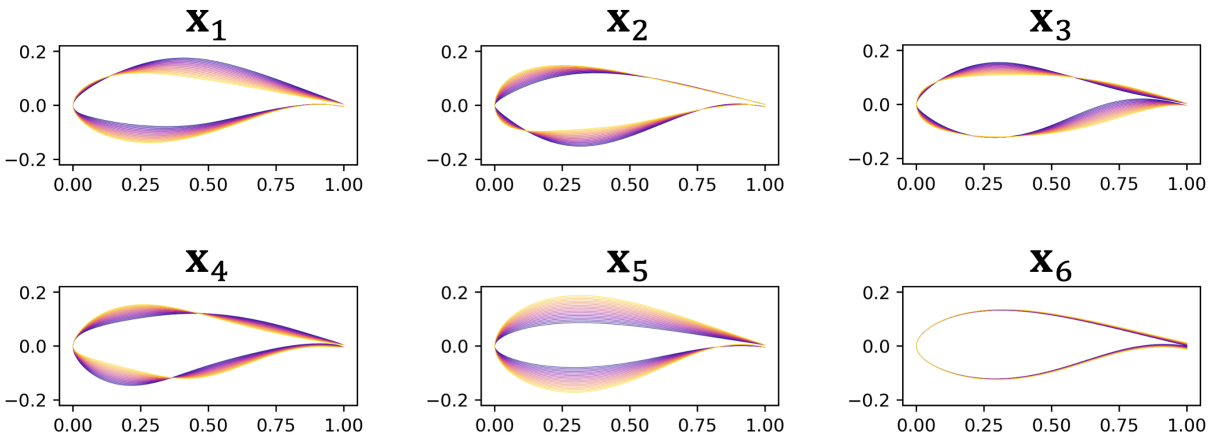

Figure 2. A visualization of sweeps through the Grassmann parameter space. For each $\mathbf{x}_i$, airfoil shapes transitioning from blue to yellow reflect changes in the associated parameter from its minimum to maximum value while all of the other parameters are held constant at their mean.

Collaboration Programme (IEA Wind) 15-MW reference wind turbine blades. New airfoils are generated by randomly perturbing the class-shape transformation [26] parameters defining these baseline shapes by up to $\pm 20\%$. In total, this provides us with 724 airfoils to define our Grassmannian basis representation and to use for INN training and validation. We randomly select roughly 70 airfoils to exclude from network training for validation. The INN is trained in alternating forward and inverse steps that seek to minimize prediction errors in $\mathbf{f}$ and pass-through reconstruction of $\mathbf{y}$. The shape parameters, $\mathbf{x}$, and the latent variables, $\mathbf{z}$, are trained against the target distributions of the training data and a standard Gaussian, respectively. Once trained, the INN model is able to rapidly design airfoil shapes along with the operational angle of attack for a desired set of performance features, including the drag coefficient, L/D, stall margin, and t/c. This capability enables the coupled design of airfoils within the larger blade design loop.

2.2. Coupled INN+WISDEM framework

We couple the INN to NREL's Wind-Plant Integrated System Design Engineering Model (WISDEM®) framework[1] for wind turbine design as shown in figure 3. New analysis blocks are shown in green and existing analysis blocks are yellow. The optimizer is shown in blue, and the gray off-diagonal blocks represent data flow between the modules. The optimizer can supply the parameters needed for airfoil inverse design that are used to generate the airfoil shapes and aerodynamic polars using the INN. Those shapes are then passed to WISDEM to compute the rotor performance metrics, including cost, which are fed back to the optimizer.

Prior to this work, WISDEM used either static airfoil polars provided by the user or directly queried XFOIL [5]. In either case, WISDEM could optimize the spanwise location of the airfoils [27] and apply the corrections on the lift coefficients to account for 3D effects based on the Du/Selig model [28]. WISDEM, however, did not have the ability to vary the airfoil shapes automatically. The inclusion of the CFD-trained INN within WISDEM allows the airfoil shapes to be controlled during the design optimization, impacting every downstream component within WISDEM, including the aerodynamic, structural, manufacturing, and cost modules.

[1] `https://github.com/WISDEM/WISDEM/`

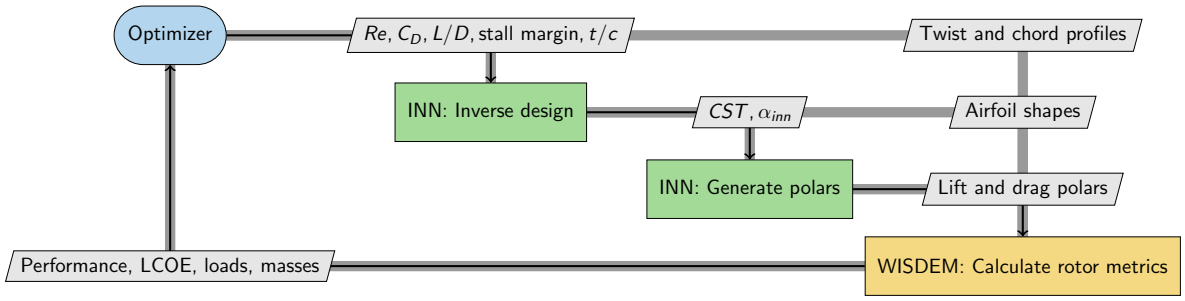

Figure 3. A schematic view of the workflow for the coupled INN+WISDEM framework. New capabilities added as part of this work are shown in green. (CST = class-shape transformation; LCOE = levelized cost of energy)

Additionally, the added cost of the INN is much lower than the cost of including XFOIL and provides higher-fidelity performance information.

3. Problem formulation and results

3.1. Optimization problem formulation

To showcase the coupled INN+WISDEM design process, we construct and solve two separate optimization problems outlined in table 1. Both optimize the rotor of the fixed-bottom version of the IEA Wind 15-MW reference turbine [29]. The first problem, labeled "Blade planform only" (WISDEM), features only the design variables available prior to this work. The second, labeled "Airfoil shape + blade planform" (INN+WISDEM), includes the airfoil shape design variables detailed earlier in addition to the blade shape and structural thickness variables. In each case the objective is levelized cost of energy (LCOE), and constraints limit blade strains to 3,500 microstrains, stall margin to 3 °, and tip deflection to stay within the blade tower clearance with a combined safety factor of 1.42. Although the optimization problems differ, the analyses are the same between the cases. To achieve a fair comparison, the INN is queried to obtain airfoil aerodynamic data for both cases.

In both problems, we control the chord of the blade, the spar cap thickness, and the spanwise location of the defined airfoils. In the INN+WISDEM problem, the design variables also parameterize the spanwise distributions of lift over drag ratio (L/D), the drag coefficient (C_D), the margin to stall, and the thickness-to-chord ratio (t/c). Next, a set of design variables controls the latent variables $\mathbf{z}$, which is the only parameter that does not have an immediate physical meaning as it affects the INN inverse mapping process. The design variables representing the airfoil positions are active in both optimization problems. Finally, the blade aerodynamic twist is controlled by a set of explicit design variables in the WISDEM case, whereas it is implicitly optimized in the INN+WISDEM problem via the calculated angle of attack. In the INN+WISDEM case, the angle of attack provided by the INN is used to invert the blade-element-momentum equations and back-calculate the corresponding twist for each airfoil along blade span [30].

For each design variable except the airfoil position and z values, we use a b-spline parameterization along the blade span. This allows us to control the optimization parameterization independently of the analysis discretization; for example, we can perform analysis at 30 locations along the span while exposing only four b-spline control points to the optimizer. Each variable is controlled using six b-spline points along the span, with the innermost two points held fixed. The chord at blade tip is also fixed.

Table 1. The design variables, constraints, and objective for each optimization problem.

		Blade planform only (WISDEM)	Airfoil shape + blade planform (INN+WISDEM)
Design variables	twist	4	(determined from stall margin)
	chord	3	3
	spar cap thickness	4	4
	airfoil position	4	4
	lift over drag ratio		4
	drag coefficient		4
	stall margin		4
	thickness to chord		4
	z		3
	Total	15	30
Constraints	strain	6	6
	stall	23	23
	tip deflection	1	1
	Total	30	30
Objective	LCOE	1	1

3.2. Optimization results

We now discuss the results of these two design problems presented in table 1. We use the COBYLA algorithm defined by Powell [31] and implemented in NLopt [32] to solve each problem. COBYLA is a derivative-free nonlinear optimizer that handles arbitrarily constrained problems. It operates by constructing a linear approximation of the problem by creating a simplex of $n_{dv}+1$ dimensions then iterating through a trust-region method.

Figure 4 shows how performance metrics from the two cases converge as the optimizer progresses. By plotting the blade cost and AEP along the optimization history, we can see how the optimizer varies the blade shape and power production to enhance the LCOE objective. The WISDEM case finds an optimal result in 374 iterations, whereas the INN+WISDEM case takes 704 iterations to explore the larger design space. Each iteration takes approximately 22 core-seconds on a notebook workstation with an Intel Core i7-9850H CPU. Although the INN+WISDEM case takes more iterations, it finds a lower LCOE value before the WISDEM case is done converging. The initial iteration histories of both cases show COBYLA exploring the design spaces with large variation in performance until it finds a region of convergence.

Table 2 compares the two final designs against the initial performance. Compared to the WISDEM case, the INN+WISDEM case further reduces LCOE by 0.9%. This reduction in LCOE comes from both a further reduction in blade cost as well as an increase in annual energy production (AEP), showing that having airfoil shape control allows for additional design improvements when included in an aerostructural blade optimization. Again, these optimization results consider only the blade and airfoil design; these gains are found without including any tower, platform, or controller design variables. Further decreases in LCOE could be found using the coupled INN+WISDEM method when considering the full turbine design.

Examining the resulting airfoil designs in more detail, figure 5 shows the initial and optimal airfoil polars and shapes for four sections along the blade span. Figure 6 shows design quantities along the span for both cases, including blade twist, chord, lift-to-drag ratio, thickness-to-chord ratio, and spar cap thickness. The spline control points are highlighted in the WISDEM twist curve to show the amount of design flexibility used in these studies.

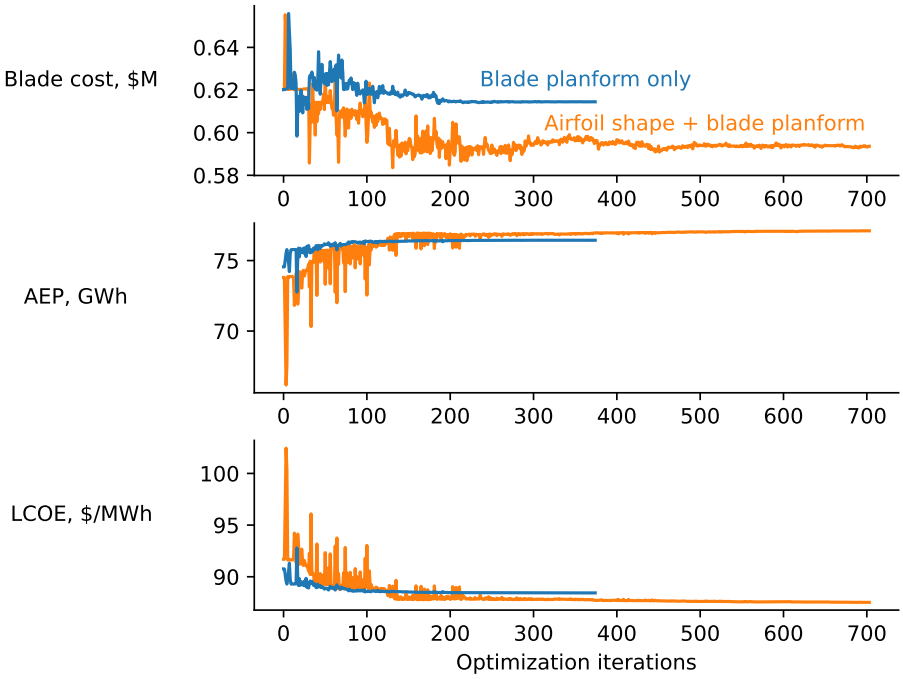

Figure 4. Convergence of the high-level performance metrics for each design case. The INN-enhanced procedure is shown in orange and the industry standard is shown in blue. The INN+WISDEM case finds a lower LCOE value in more optimization iterations.

Table 2. High-level metrics for the optimized results compared to the initial design show both a decrease in turbine cost and an increase in AEP.

	Blade cost, $M	Turbine cost, $M	AEP, GWh	LCOE, $/MWh
Initial	0.621	22.399	74.554	90.76
Blade planform only	0.607	22.294	76.519	88.34
Airfoil shape + blade planform	0.598	22.170	77.088	87.58

Across the blade span, the INN+WISDEM case increases the lift-to-drag ratio of the airfoils, especially at the operation points. The airfoil shapes do not differ greatly near the midspan of the blade, but outboard the INN+WISDEM method converges to thicker airfoils, whereas inboard the INN+WISDEM airfoils are slightly thinner. Throughout the span, the INN+WISDEM method is able to improve the operating lift-to-drag ratio by fine-tuning the airfoil shape in tandem with blade twist. Overall, the INN+WISDEM approach simultaneously increases the aerodynamic performance and reduces ultimate loads. These loads are estimated in WISDEM by simulating an extreme steady-state "gust" hitting the turbine at rated rotor speed and rated pitch angle. The INN+WISDEM tunes the airfoil shapes and polars so that the angles of attack generated during the steady-state gust fall into the post-stall region and sit in the dip of minimum lift coefficient, in turn limiting the blade deflections toward the tower and the strains along span. The reduced loads allow a decrease in the spar cap thickness and therefore the blade mass. A true blade design process would involve optimizing over many design scenarios to find the aerostructurally limiting case, whereas in this work we used one such case.

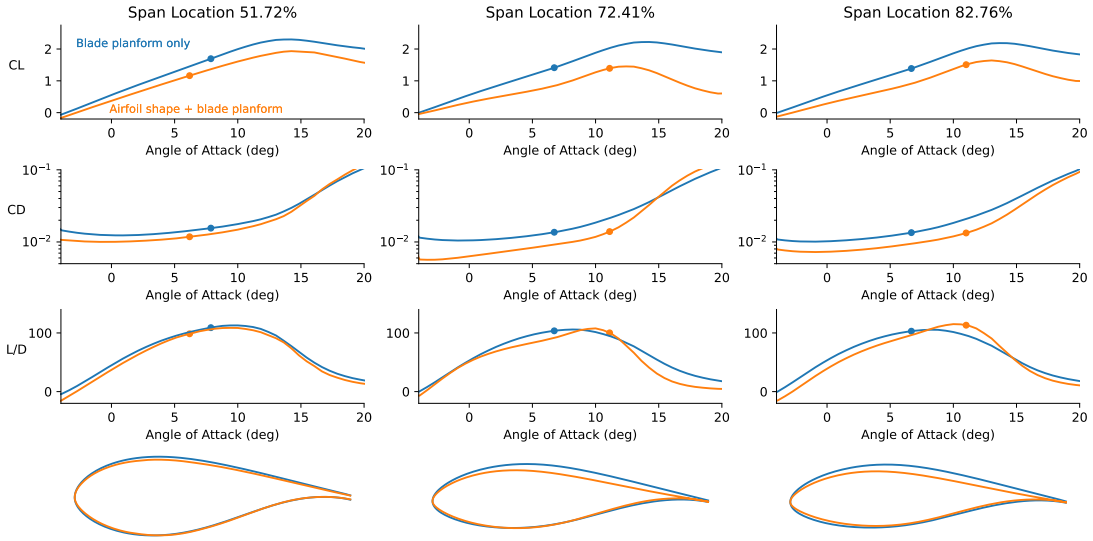

Figure 5. Here we show the two optimized blade designs using WISDEM and INN+WISDEM and their corresponding polars at selected cross sections. The optimized airfoil shapes differ along the span for the INN+WISDEM case and generally have higher operating lift-to-drag ratios. The operational angles of attack for each airfoil are shown with points.

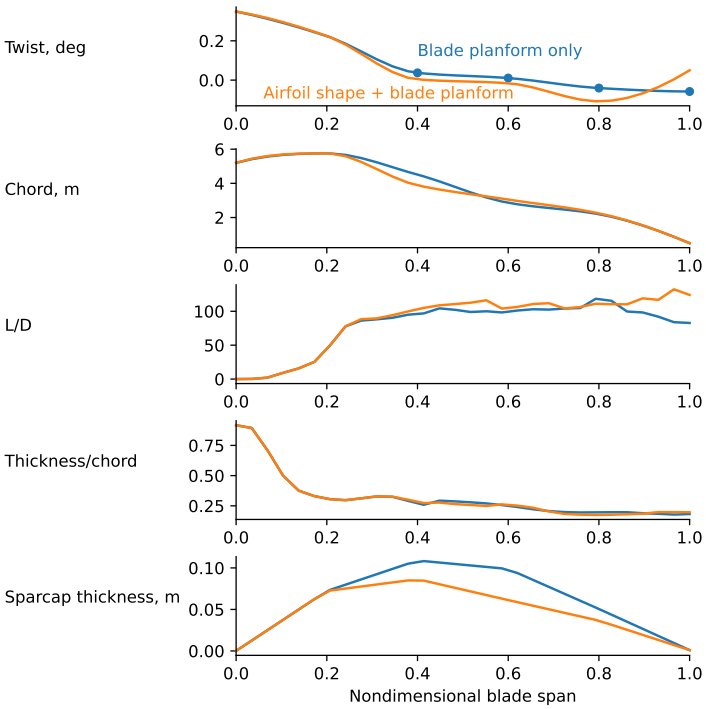

Figure 6. Spanwise design parameters are shown here with the b-spline control points highlighted in the twist plot. The INN+WISDEM case reduces spar cap thickness by using thicker airfoils and reducing ultimate loads, leading to reduced blade mass and cost. Additionally, the L/D for the INN+WISDEM case is generally higher than that for the WISDEM case across the span.

4. Conclusions

Coupling a novel INN to the wind turbine design framework WISDEM allows for CFD-informed airfoil shape control within the blade design process in a computationally tractable manner. The INN is able to parameterize airfoil shapes using a reduced set of design variables while still maintaining meaningful control over the aerodynamic performance. As shown in our application problems, we see a decrease in LCOE by solving this coupled optimization. The INN+WISDEM approach tailors the airfoil shapes to decrease blade mass and cost while increasing AEP. Additionally, we see an aerostructural trade-off where the airfoil cross sections increase in thickness to allow for spar cap thickness to decrease while satisfying structural constraints. This novel approach allows for better exploration of a more complex space earlier in the design process.

This coupled INN+WISDEM approach is now included in the base WISDEM toolset, including installation and usage documentation. Users can enable the INN by activating analysis flags and use it for either evaluation or design optimization within the larger turbine design workflow.

Ongoing work aims to extend the analysis to include unsteady aerodynamics and aeroservoelasticity. The framework Wind Energy with Integrated Servo-control (WEIS)[2], which couples WISDEM to the hydro-aero-servo-elastic solver OpenFAST and to the dynamic Reference OpenSource turbine Controller (ROSCO), will be used. Inclusion of the INN within a turbine design framework also has other benefits not explored in this paper. Specifically, the INN can provide analytic gradients of the outputs with respect to the inputs, which enables efficient gradient-based optimization. Coupled to a framework that provides low-cost gradients, the speed of the design process could be further increased through these improvements. If a designer is interested in controlling the airfoil shapes through specific performance metrics, the INN can be retrained with those metrics as inputs to the inversion process. Although we used L/D, C_D, and t/c as design variables in this work, those are not the required parameters for the framework, and any desired metric could be used instead. Ongoing work is also focused on improving the accuracy of the INN relative to performance data obtained from CFD. Lastly, there is ongoing work to use CFD data from 3D simulations of the blade to train the INN. This would move away from the 2D airfoil approach and allow for more realistic prediction of the 3D aerodynamic performance of the full blade.

Acknowledgments

The authors would like to thank Shreyas Ananthan for his efforts developing this project. We thank Dr. James Baeder and Dr. Yong Su Jung from the University of Maryland for providing us access to the HAM2D CFD solver and mesh generation codes to generate the training data for this work and performing the CFD simulations to generate the training data.

This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Wind Energy Technologies Office. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

The research was performed using computational resources sponsored by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy and located at the National

[2] `https://github.com/WISDEM/WEIS/`

Renewable Energy Laboratory.

References

[1] Zahle F, Tibaldi C, Pavese C, McWilliam M K, Blasques J P and Hansen M H 2016 *Journal of Physics: Conference Series* vol 753 (IOP Publishing) p 062008

[2] Bottasso C L, Bortolotti P, Croce A and Gualdoni F 2016 *Multibody System Dynamics* **38** 317–344

[3] Scott S, Macquart T, Rodriguez C, Greaves P, McKeever P, Weaver P and Pirrera A 2019 *Journal of Physics: Conference Series* **1222** 012012 URL `https://doi.org/10.1088/1742-6596/1222/1/012012`

[4] Bortolotti P, Bottasso C L and Croce A 2016 *Wind Energy Science* **1** 71–88

[5] Drela M 1989 *Low Reynolds number aerodynamics* (Springer) pp 1–12

[6] Zhu W J, Shen W Z and Sørensen J N 2014 *Renewable Energy* **70** 172–183

[7] Bottasso C L, Croce A, Sartori L and Grasso F 2014 *Journal of Physics: Conference Series* **524** 012041

[8] Grasso F 2011 *Journal of Aircraft* **48** 248–255

[9] Barrett R and Ning A 2018 *Wind Energy* **21** 663–675

[10] Mangano M, He S, Liao Y, Caprace D G and Martins J R 2022 *AIAA SCITECH 2022 Forum*

[11] Caprace D G, Cardoza A, Ning A, Mangano M, He S and Martins J R 2022 *AIAA SCITECH 2022 Forum*

[12] Li J, Li R, Gao Y and Huang J 2010 *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy* **224** 827–838

[13] Han Z, Zhang K, Song W and Liu J 2013 *2013 AIAA SciTech* p 1108

[14] Oh S 2020 *Applied Sciences* **10** 6277

[15] Sessarego M, Ramos-García N, Yang H and Shen W Z 2016 *Renewable Energy* **93** 620–635

[16] Sessarego M, Feng J, Ramos-García N and Horcas S G 2020 *Renewable Energy* **146** 1524–1535

[17] Bouhlel M A, He S and Martins J R 2020 *Structural and Multidisciplinary Optimization* **61** 1363–1376

[18] Achour G, Sung W J, Pinon-Fischer O J and Mavris D N 2020 *AIAA Scitech 2020 Forum* p 2261

[19] Yang S, Lee S and Yee K 2021 *arXiv preprint arXiv:2108.08500* URL `https://arxiv.org/abs/2108.08500`

[20] Ardizzone L, Kruse J, Wirkert S, Rahner D, Pellegrini E W, Klessen R S, Maier-Hein L, Rother C and Köthe U 2018 *arXiv preprint arXiv:1808.04730* URL `https://arxiv.org/abs/1808.04730`

[21] Glaws A, King R, Vijayakumar G and Ananthan S 2022 *AIAA Journal (in press)*

[22] Dinh L, Sohl-Dickstein J and Bengio S 2016 *arXiv preprint arXiv:1605.08803* URL `https://arxiv.org/abs/1605.08803`

[23] Grey Z J 2019 Ph.D. thesis University of Colorado URL `https://www.proquest.com/docview/2334393179`

[24] Doronina O, Grey Z J and Glaws A 2021 *AAAI 2022 Workshop on AI for Design and Manufacturing (ADAM)* URL `https://arxiv.org/abs/2201.04649`

[25] Jung Y S and Baeder J 2019 *Journal of Aircraft* **56** 1040–1055

[26] Kulfan B M 2008 *Journal of Aircraft* **45** 142–158

[27] Bortolotti P, Johnson N, Abbas N J, Anderson E, Camarena E and Paquette J 2021 *Wind Energy Science* **6** 1277–1290

[28] Du Z and Selig M *1998 ASME Wind Energy Symposium*

[29] Gaertner E, Rinker J, Sethuraman L, Zahle F, Anderson B, Barter G, Abbas N, Meng F, Bortolotti P, Skrzypinski W, Scott G, Feil R, Bredmose H, Dykes K, Shields M, Allen C and Viselli A 2020 Tech. Rep. NREL/TP-75698 International Energy Agency

[30] Bortolotti P, Dixon K, Gaertner E, Rotondo M and Barter G 2020 *Journal of Physics: Conference Series* **1618** 042016

[31] Powell M J 1994 *Advances in optimization and numerical analysis* (Springer) pp 51–67

[32] Johnson S G 2014 The nlopt nonlinear-optimization package URL `http://github.com/stevengj/nlopt`