Received: 24 April 2021 | Revised: 6 September 2021 | Accepted: 17 September 2021
DOI: 10.1002/we.2687

RESEARCH ARTICLE

WILEY

Model updating of wind turbine blade cross sections with invertible neural networks

Pablo Noever-Castelos[1] | Lynton Ardizzone[2] | Claudio Balzani[1]

[1]Institute for Wind Energy Systems, Leibniz University Hannover, Hanover, Germany

[2]Visual Learning Lab, Heidelberg University, Heidelberg, Germany

Correspondence
Pablo Noever-Castelos, Leibniz University Hannover, Institute for Wind Energy Systems, Appelstr. 9A, Hanover 30167, Germany.
Email: research@iwes.uni-hannover.de

Funding information
Bundesministerium für Wirtschaft und Energie, Grant/Award Numbers: 0324032C, 0324335B; Leibniz University Hannover; Lower Saxony Ministry of Science and Culture (MWK); German Research Foundation (DFG)

Abstract

Fabricated wind turbine blades have unavoidable deviations from their designs due to imperfections in the manufacturing processes. Model updating is a common approach to enhance model predictions and therefore improve the numerical blade design accuracy compared to the built blade. An updated model can provide a basis for a digital twin of the rotor blade including the manufacturing deviations. Classical optimization algorithms, most often combined with reduced order or surrogate models, represent the state of the art in structural model updating. However, these deterministic methods suffer from high computational costs and a missing probabilistic evaluation. This feasibility study approaches the model updating task by inverting the model through the application of *invertible neural networks*, which allow for inferring a posterior distribution of the input parameters from given output parameters, without costly optimization or sampling algorithms. In our use case, rotor blade cross sections are updated to match given cross-sectional parameters. To this end, a sensitivity analysis of the input (material properties or layup locations) and output parameters (such as stiffness and mass matrix entries) first selects relevant features in advance to then set up and train the *invertible neural network*. The trained network predicts with outstanding accuracy most of the selected cross-sectional input parameters for different radial positions; that is, the posterior distribution of these parameters shows a narrow width. At the same time, it identifies some parameters that are hard to recover accurately or contain intrinsic ambiguities. Hence, we demonstrate that *invertible neural networks* are highly capable for structural model updating.

KEYWORDS
Bayesian optimization, blade cross section, invertible neural network, machine learning, model updating, sensitivity analysis, wind turbine rotor blade

1 | INTRODUCTION

Wind turbine blades are huge and complex structures that are exposed to extreme load conditions. Thus, an accurate blade design is of fundamental importance for the turbine's safety and reliability. As for most engineering structures, primarily numerical models form the design basis for rotor blades. However, manufacturing deviations lead to a mismatch in the structural behavior of the numerically designed rotor blades and those produced in real life.[1] These deviations may prove crucial even within the allowed tolerances and material parameter uncertainties. Consequently, enhancing virtual models by means of model updating is an important aspect of a modern blade design procedure. Model updating seeks to correct the inaccurate parameters of the numerical model in order to improve test result predictions.[2] This method is applied either for calibrating the model with conducted real-life tests[3,4] or to detect damage in terms of structural health monitoring.[5] The updated model provides a basis for a digital twin of the rotor blade produced.[6]

Model updating most commonly takes the form of an optimization problem: This optimization can either directly manipulate the modeling parameters (e.g., material properties and layup) or take corrective action in the final model itself (e.g., stiffness or mass matrix of a beam model).[7] For both approaches, metaheuristic algorithms such as genetic or particle swarm algorithms are commonly used to solve the optimization problem.[8] Such deterministic model updating algorithms (e.g., global pattern search[9]) have been applied successfully in the field of rotor blade damage detection. However, all these algorithms yield exactly one result for the model parameters and do not cover possible result ambiguity, that is, multiple model parameter sets that lead to the same output parameters. This can emerge due to a lack of sensitive output parameters. This uncertainty worsens the user's confidence in the updated model parameters, as more than one configuration may yield the given output results.[10] Depending on the algorithm, it even may get stuck in local optima and depend on the randomness of the starting samples.[11] Bayesian inference algorithms solve this issue by predicting posterior distributions for the updated parameters, which lets the user estimate the prediction confidence. Popular methods for this include Bayesian model updating[12] and approximate Bayesian computation.[13]

All the aforementioned approaches for model updating suffer from the same general drawback: the prohibitively high computational cost of repeatedly simulating the physical model. This is especially severe for the probabilistic algorithms such as Bayesian model updating, where techniques like Markov Chain Monte Carlo sampling are needed. Practitioners try to avoid this problem by using surrogate models, which are faster to compute than the full physical model, to cut down on the computational costs.[14] These surrogate models can take the form of reduced order models[15] or other reduction techniques such as the response surface method.[4] However, the surrogate model approach, in turn, sacrifices physical input–output linkage of the original model and may lead to a loss in accuracy depending on the abstraction level and the model complexity itself.[16]

Machine learning techniques, specifically artificial neural networks (ANNs), can help address these issues of model updating in various ways. Most importantly, they can be trained to map the relationship between input and output parameters highly accurately, without knowledge of the physical connections.[17] In this way, they can serve as surrogate models that may be substantially faster to compute or more accurate than other types of surrogates.[18–21] They have also been successfully applied as surrogates in Bayesian model updating.[22,23] However, combining ANNs and model updating algorithms in such a way requires a considerably amount of implementation effort, tuning, and software engineering. Even then, using ANNs as surrogates may not achieve the desired speed-up, as they do not remove the fundamental limitations of having to compute the surrogate model many times, which is part of optimization-based or sampling-based model updating. In principle, ANNs could also be trained to predict the desired parameters directly, circumventing the need for an optimization procedure altogether. While they are orders of magnitude faster than any traditional model updating techniques, the main problem is that they lack indications of confidence, uncertainty, or goodness of fit and are hard to verify rigorously. Due to this, standard ANNs are rarely used in this direct way for the purpose of model updating.

Within this difficult setting, we present the main idea of this paper: We use *invertible neural networks* (INNs) as probabilistic models to produce a posterior distribution of the input parameters directly. During training, the network receives the model parameters as inputs, as would be the case with a surrogate model ANN. At test time, however, the network can be inverted to produce samples from the posterior directly, without having to carry out additional algorithms. This approach offers a significant potential speed-up over traditional model updating techniques, even ones using ANNs as surrogate models. At the same time, we obtain a full Bayesian posterior that allows among other things the determination of confidence intervals and revealing of ambiguities in the same way that is otherwise reserved for computationally expensive Bayesian model updating algorithms. In contrast, this is not possible with existing direct ANN approaches or standard optimization-based model updating procedures.

This is a feasibility study for model updating with INNs and thus relies on a reduced set of material and geometrical input parameters. It is based on a low structural blade model level, the blade cross section. This represents a generic and not a real world application, as cross-sectional (CS) properties—if possible at all—are not intended to be measured, and may also require destructive test/measuring methods. However, our experiments clearly demonstrate the practical applicability and benefit of INNs in the research field of structural rotor blade model updating: The INN predicts highly accurately selected material and layup parameters based on CS beam properties, as well as offering verifiable uncertainty estimates, and identifying some ambiguous and unrecoverable parameters.

To the best of our knowledge, ANNs have not yet been applied for the structural model updating of wind turbine blades, especially not in the form of probabilistic models such as INNs. Instead, the major application of ANNs in the context of wind turbines is the field of controls (e.g., model predictive control,[24] adaptive control,[25] yaw control,[26] and aerodynamic coefficient prediction for control[27]) and for condition or structural health monitoring considering fault or damage prediction.[28–33] INNs have been introduced relatively recently, even in the field of machine learning itself,[34–38] but have seen rapidly growing research attention in the last years. They have been successfully applied in a broad field of applications, commonly in image processing but also in scientific studies. In this paper, we specifically adapt and apply the *conditional invertible neural network* (cINN)[39] implemented in the FrEIA Framework.[40]

The application example evaluated in this feasibility study has some specific limitations, which are summarized in the following: (i) The updating procedure does not account for full rotor blades but links material parameters and layup topologies with blade CS properties for exemplarily chosen cross sections of a real blade. (ii) The blade cross sections are simplified by omitting adhesive joints in order to keep the investigations simple. (iii) The analyzed parameter space is limited to a number of material parameters (Young's moduli, shear moduli, Poisson's ratios, and mass densities) and the ply positions in the in-plane directions of the cross sections. It should be noted the aforementioned limitations have been included in order to keep the application example as simple as possible but still representative for a real wind turbine rotor blade. The method itself does not require these limitations. For instance, adhesive joints can be included by refining the underlying physical models accordingly. Also, the application of the proposed concept to updating of full blade models (i.e., finite beam element models or full 3D finite element models) should generally be possible and is subject of ongoing research.

Section 2 of this paper covers the overall workflow description, with explanations of the feature selection method based on a sensitivity analysis. The approach and architecture of a cINN are briefly addressed in Sections 2.2 and 2.3, respectively. Subsequently, the feature selection results are presented in Section 3. The cINN parameter definition, training, and evaluation are reported in Section 4, followed by the conclusion in Section 5.

2 | MODEL UPDATING METHODOLOGY WITH INNs

This section describes the methods used in this investigation to analyze the input and output parameters of the model updating procedure and how a neural network is structured and trained for inverse problems. In this feasibility study, we will restrict the problem to rotor blade CS analysis with a reduced set of input parameters and evaluate the capability of *INNs* for structural model updating on a first level in wind turbine blade structural design processes. This work is intended to reveal the potential of the presented methodology in a structural wind turbine-related environment, while still keeping the model updating problem rather simple. Figure 1 illustrates the overall workflow for this study, which the following subsections will discuss in more detail. Briefly summarized, the approach consists of a data preprocessing step in the form of a sensitivity analysis to identify relevant input and output features of the model. Following the sensitivity analysis, all features are individually, simultaneously, and randomly sampled with the physical model, to represent all possible parameter combinations. Based on these input samples, the CS properties of the wind turbine blade at a particular radial position are calculated. The input and output features are filtered according to the feature selection. The workflow splits these sample sets of input and output features into training, validation, and testing subsets for the cINN (the validation set is used to check the progress of the training and tune the network settings. The test set is only used for the final evaluation of the method, so as to avoid biasing the results). The data generation is based on a rotor blade model within the in-house modeling tool Model Creation and Analysis Tool for Wind Turbine Rotor Blades (MoCA)[41] and its interface to BEam cross-section Analysis Software (BECAS).[42]

2.1 | Sensitivity analysis of blade CS properties

Data preprocessing plays an important role in building a proper dataset specially for neural networks and for machine learning problems in general.[43] This contribution focuses on a sensitivity analysis to perform feature selection.[44] The feature selection technique reduces the number of

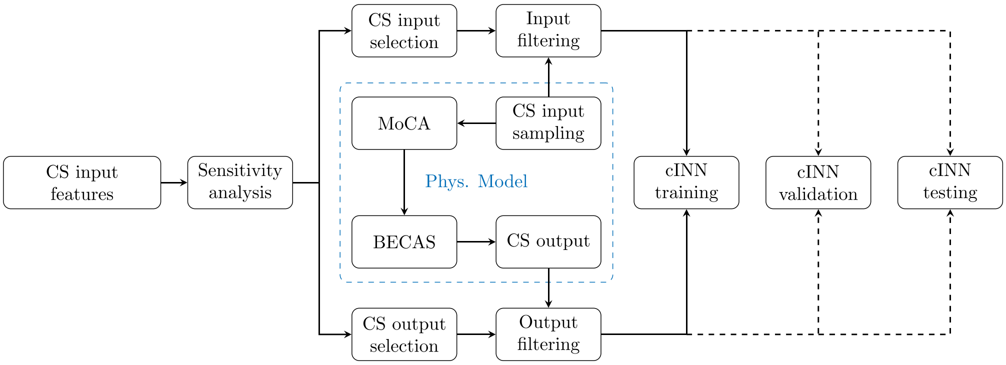

FIGURE 1 Overall workflow of this study. Based on all cross-sectional input features, a sensitivity analysis is performed to determine the relevant input and output features. The physical model is then used to generate sample sets of input and output features, which are then filtered by the feature selection of the sensitivity analysis. All samples are finally split into training, validation, and test sets for the *conditional invertible neural network* (cINN)

input features to a subset which has a significant impact on the output features, based on the assumption that some data contains irrelevant or redundant information.[45] Additionally, the output features are reduced by all insensitive components, as these cannot be expressed with respect to the given input features.

In our use case example of CS model updating, we will be focusing on material and geometrical composite layup parameters. These parameters include Young's modulus E_{11} (for anisotropy additionally: E_{22}), shear modulus G_{12}, Poisson's ratio ν_{12}, and density ρ, which are varied for all applied materials. However, the Young's modulus E_{33} of the anisotropic materials is excluded, as through thickness stresses are not covered in the CS analysis. Additionally, the geometrical layup parameters are described by the layup division point locations on the cross section's circumference. These division points subdivide the composite layup in the CS direction in our model. All selected parameters will be further addressed in a subsequent section. Geometric blade shape parameters, such as blade chord and thickness, are neglected, as 3D laser scanning can offer an accurate measurement of the blade outer shell/mold,[46] and thus, the overall blade shell geometry is assumed to be known. The authors are aware that these selected input features do not cover the full range of varying parameters (e.g., adhesive and longitudinal ply positioning) due to manufacturing but should be reasonably representative for a feasibility study on CS model updating. Figure 2 illustrates the sensitivity analysis for a simplified example of three input features x: Young's modulus E, density ρ, and one division point P, as well as three output features y: mass M, stiffness K, and area A.

During the sensitivity analysis, all selected input features x are varied individually in a one-at-a-time manner. The CS property response of all created parameter subsets is calculated with MoCA and BECAS for a particular blade radius. All these subsets are then concatenated to a full database, labeled as *CS results* in Figure 2. Each output feature y is then standardized to $\overline{y} = 0$ and $\sigma = 1$ across the full database and denoted as $\hat{y}$. This simplifies the sensitivity evaluation of each feature, as $\hat{y}$ describes the output feature's deviation magnitude for each sample in relation to all other samples.

After standardization, the full database is split into the subset again, that is, variation of one input feature x. Figure 2 contains the calculation of the standard deviation $\sigma_{y,\,x}$ across each subset's output feature, denoted as σ, which is then collected in the sensitivity matrix. Through this, all input features x have a single value for each output feature y showing the input feature's impact on that respective output feature. Finally, by defining a threshold λ, the sensitivity analysis identifies irrelevant input features, in case $\sigma_{y,\,x} < \lambda$ for all y. On the other hand, an insensitive output feature y is discarded if $\sigma_{y,\,x} > \lambda$ applies for all x. Additionally, the algorithm reduces all linearly dependent output features to one, as the others do not include further information for the training process of the neural network.

2.2 | INNs for inverse problems

The general setting described in the introduction is shared across many fields in engineering and natural science: The problem is well understood and modeled in the forward process; that is, the observed response y can be readily calculated based on some parameters x that describe a system (from mechanics, physics, chemistry, medicine). However, scientists are commonly interested in the corresponding *inverse problem*, that is,

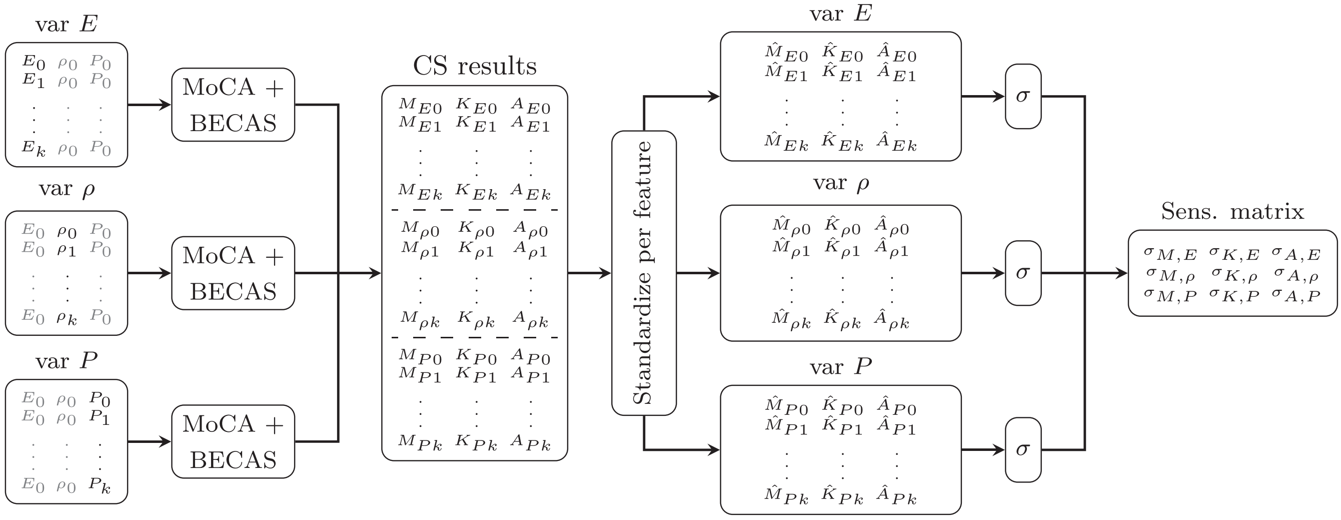

FIGURE 2 The feature selection process based on a sensitivity analysis applied to a simplified cross-sectional (CS) example with three input features: Young's modulus E, density ρ, and division point position P. The algorithm varies all individually and calculates their corresponding CS characteristics, here, as an example, the mass M, stiffness K, and area A. After global feature standardization, splitting into the previous sets, and computing of the standard deviation, the process returns a reduced sensitivity matrix. This can be used for feature selection

computing possible states x of the underlying system given observations y. Computing this inverse can be a highly challenging task, and common approaches such as classical model updating have some intrinsic shortcomings, as was briefly laid out in the introduction. Firstly, they are often computationally very expensive, as the forward process has to be computed or simulated many times to fit a set of system parameters $\hat{x}$ that matches a given set of observations. Secondly, especially for safety-relevant applications, obtaining a single estimate $\hat{x}$ is not sufficient: Ideally, any ambiguities in the solution as well as its uncertainty should be captured and precisely quantified. This can be fulfilled by a (Bayesian) posterior distribution $p_x(x \mid y)$.[47] The posterior quantifies the probability that any system state x could have led to the observations y and makes it possible to produce confidence intervals or discover ambiguous or unrecoverable system parameters.[48]

An approach that alleviates both of these difficulties, and has seen growing adoption in recent years, is the use of *cINNs* to model the full posterior distribution reliably in a computationally efficient way. Such networks were first successfully applied to image processing such as inpainting,[34] colorization of grayscale images,[39] and synthetic image generation.[36] More recently, they have entered other scientific fields such as astrophysics,[49] particle physics,[50] medical imaging,[51] and most recently in epidemiology.[52] In short, cINNs rely on a simple reference distribution $p_z(z)$ called the *latent distribution*, most commonly a Gaussian. The cINN f then conditionally transforms and reshapes between the posterior p_x $(x \mid y)$ and the latent distribution $p_z(z)$ (see Figure 3). The cINN can be understood as an inverse surrogate model of the well-known physical model. The output of the physical model can be passed as a conditional observation y to the cINN to infer the posterior distribution $p_x(x|y)$. Finally, the established inverse model, which can be evaluated again at any time, is a striking benefit over the optimization-based model updating algorithms applied for one particular set of parameters.

From this construction, the posterior that the network represents can be exactly computed through the change-of-variables formula as follows:[34]

$$p_x(x|y) = p_z(f(x;y))\left|\det\left(\frac{\delta f}{\delta x}\right)\right| \tag{1}$$

Here, $\det\left(\frac{\delta f}{\delta x}\right)$ denotes the determinant of the model's Jacobian, $\det(J)$ for short from here on. Similarly, samples from the posterior can be drawn by first sampling z from the latent distribution $p_z(z)$ and then using the *inverted* cINN to transform them to the domain of the posterior: $x = f^{-1}(z;y)$. As with many classic probabilistic modeling techniques, the cINN can be trained through maximum likelihood training. This means that given existing pairs of (x_i, y_i), the model's posterior $p_x(x \mid y)$ will match the true posterior of the inverse problem $p^*(x \mid y)$ if the average log-likelihood of the models posterior is maximized or, as done in practice, the negative logarithmic likelihood (NLL) is minimized. Together with the change-of-variables formula and a Gaussian latent distribution $p_z(z) \propto \exp(\|z\|^2/2)$, we arrive at the following objective:

$$\mathcal{L}_{\text{NLL}} = \mathbb{E}[-\log(p(x_i|y_i))] = \mathbb{E}\left[\frac{\|f(x_i;y_i)\|^2}{2} - \log|\det(J_i)|\right] + const. \tag{2}$$

For a more detailed explanation and derivation of the objective function, see, for example, Ardizzone et al.[39]

2.3 | Architecture and training of the conditional INN

From the previous section, we can conclude that the neural network we use to represent f must be invertible and have a way of readily computing the Jacobian determinant. In the following, we describe the implementation of the cINN architecture that satisfies these requirements. In general,

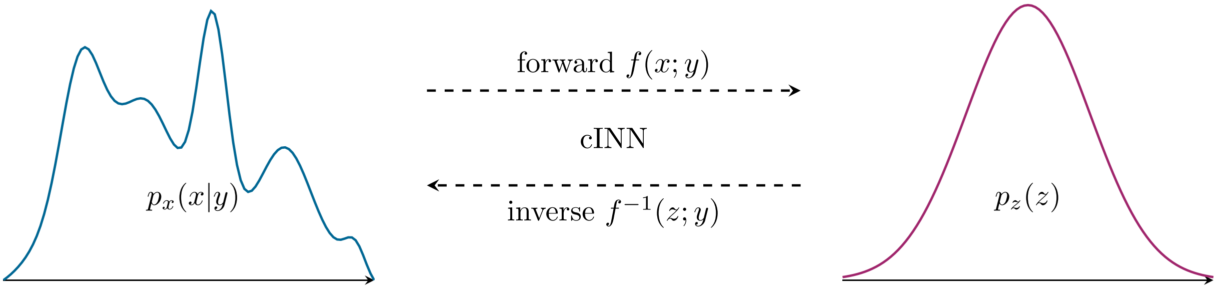

FIGURE 3 Schematic illustration of the principle of a cINN. The cINN f conditionally transforms and reshapes between the posterior $p_x(x \mid y)$ and the latent distribution $p_z(z)$

the cINN consists of a sequence of so-called coupling blocks, specifically affine coupling blocks in our case. To this end, the (unconditional) coupling blocks introduced in RealNVP[35] can be extended to include the condition y, shown below in Figure 4.

This block first splits the input data u into $[u_1, u_2]$ and applies affine transformation according to the following functions:

$$v_1 = u_1 \odot \exp(s_1(u_2, y)) + t_1(u_2, y) \tag{3}$$

$$v_2 = u_2 \odot \exp(s_2(v_1, y)) + t_2(v_1, y) \tag{4}$$

The results $[v_1, v_2]$ are concatenated afterwards to v. Inverting the set of equations yields these inverse operations:

$$u_2 = (v_2 - t_2(v_1, y)) \oslash \exp(s_2(v_1, y)) \tag{5}$$

$$u_1 = (v_1 - t_1(u_2, y)) \oslash \exp(s_1(u_2, y)) \tag{6}$$

The internal functions s_j and t_j always take as input the corresponding variables u_2 or v_1 and additionally the conditional data y. As these functions must not be inverted, they can be replaced by any arbitrary mathematical expression: in our case, by shallow standard neural networks that will be referred to as subnetworks. One big advantage of this coupling block is the simplicity to compute the logarithm of the Jacobian determinant being the sum of s_1 and s_2 over the inputs dimension.[35] The cINN architecture then consists of the aforementioned sequence of conditional coupling blocks CC as depicted in Figure 5, each of them fed with the condition y (CS output). This cINN can than be evaluated in forward and inverse direction between the input x (CS input) and latent space z. In addition to the plain coupling blocks, we include a number of technical improvements common to invertible network architectures, such as fixed permutations between variables. In order to improve generalization of

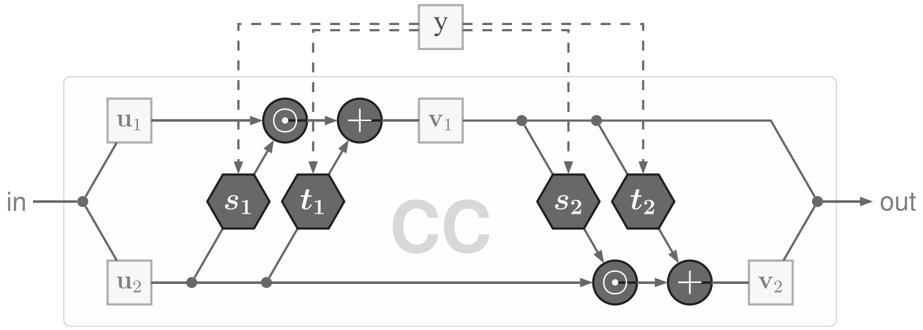

FIGURE 4 Structure of a conditional affine coupling block (CC)[39]

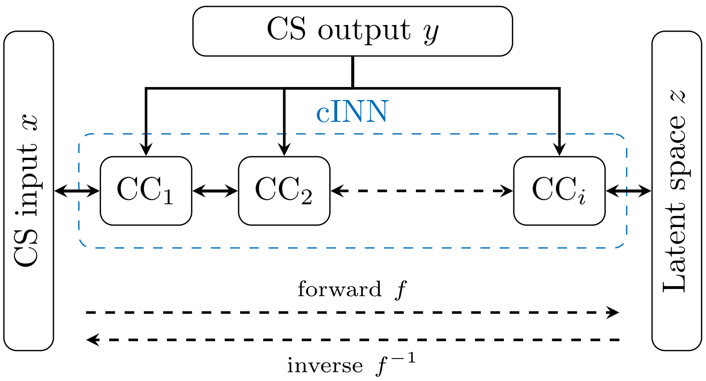

FIGURE 5 The *conditional invertible neural network* (cINN) structure applied to cross-sectional model updating. In the forward path f, cross-sectional (CS) input features x are processed over a sequential concatenation of conditional coupling (CC) blocks, which represents the cINN. The CS output features y contribute as coupling block conditions, and the cINN result is a latent space z. This path can be inverted, which is defined as f^{-1}

the cINN, we apply dropout layers in the subnetworks as well as L2 weight regularization.[53] Gradient clipping avoids exploding gradients in back propagation,[54] and an optimizer's learning rate scheduler improves the convergence.[53]

3 | INPUT AND OUTPUT FEATURE SELECTION

The sensitivity analysis and feature selection as described in Section 2.1 will focus in this study particularly on the cross section at a radial position of R = 6 m of the SmartBlades2 DemoBlade. This cross section is depicted in Figure 6 as a BECAS output with material assignment and division point (*P*) location. In a cross section, division points divide the shell into different sections with a constant material layup or define subcomponent positions such as the web location.

In addition to the CS view, Tables A1 and A2 in Appendix A contains the layup of each CS subcomponent at R = 6 m for a full insight into the analyzed structure. This may enhance the interpretation of the following sensitivity analysis in this section. The spar cap is prefabricated with balsa transition pieces on each side of the unidirectional glass fiber (UD) material and trimmed to the correct size, before placing it in the blade mold.

The input feature variation is selected based on manufacturing tolerances for materials and the layup of wind turbine blades. The manufacturing documentation of the SmartBlades2 DemoBlade allows tolerance thresholds of max, ±5% deviation for material parameters, such as densities and stiffnesses. The ply positioning tolerances in CS direction, that is, division point locations, depend on the material; valid tolerances for core material are ±5 mm, whereas spar cap and web location may vary ±5–10 mm maximum. In order to account for even higher inaccuracies, the analysis range was extended for each parameter as stated in Table 1. As the spar cap is prefabricated, all related positions varied together; that is, all suction side division points from $P_{\text{SS,TE,core}}$ to $P_{\text{SS,LE,core}}$ are moved simultaneously; the same is true for the pressure side, respectively.

After generating the model in MoCA and processing it with BECAS, the output features in Table 2 are available. These include CS locations of shear, elastic, area, and mass center, as well as total mass, total area, inertias, and principal axis orientation. However, the most important output is probably the stiffness and mass matrices, which serve as input for finite element beam models.

Following the sensitivity algorithm described in Section 2.1, a sensitivity matrix is computed based on the parameter variation listed in Table 1 and the CS output variables in Table 2. The full sensitivity matrix is given in Tables A3–A5 in Appendix A. The sensitivity analysis is part

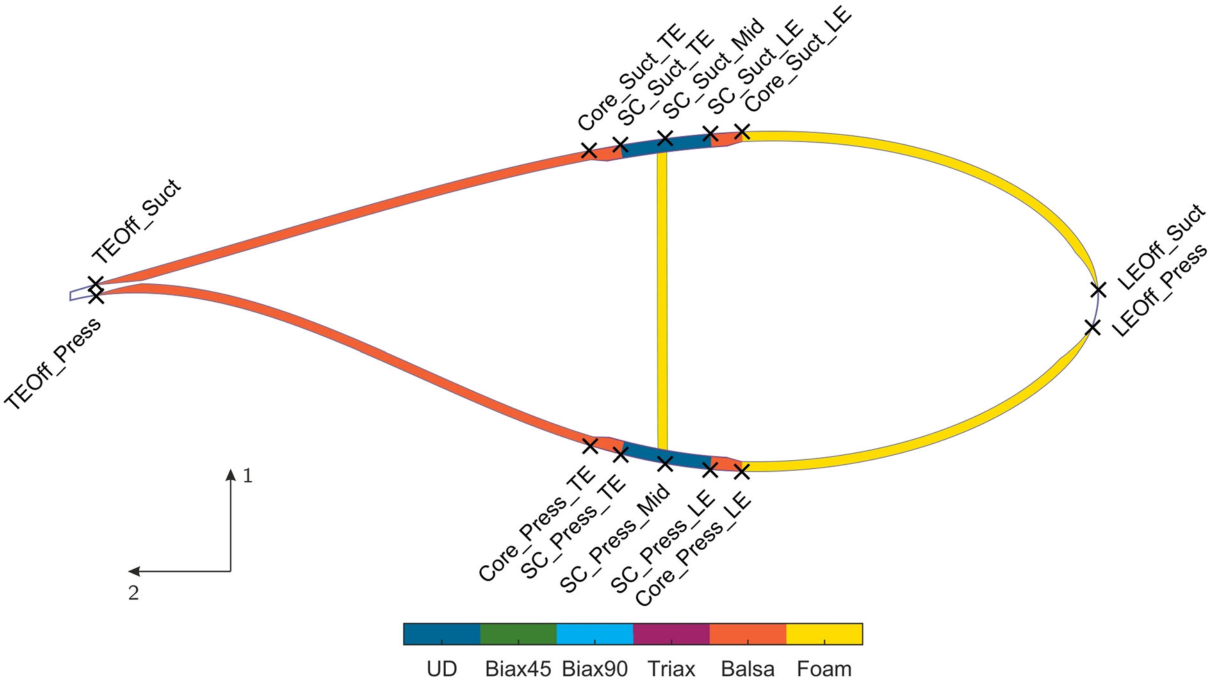

FIGURE 6 This is a cross section of the SmartBlades2 DemoBlade at a radial position of $R = 6$ m. The division points on the circumference divide the blade shell into sections of equal material layup as follows: $P_{\text{SS,TE,offset}}$ to $P_{\text{SS,TE,core}}$ is the suction side (SS) sandwich panel located at the trailing edge (TE); $P_{\text{SS,TE,spar cap}}$ to $P_{\text{SS,LE,spar cap}}$ denotes the location of the spar cap; this is flanked by balsa transition pieces in-between $P_{\text{SS,TE,core}}$ to $P_{\text{SS,LE,core}}$; followed by the sandwich panel located to the leading edge (LE) from $P_{\text{SS,LE,core}}$ to $P_{\text{SS,LE,offset}}$. The pressure side (PS) is constructed accordingly. As the outer face laminates are hard to identify due to their small thickness, the layup is shown in Tables A1 and A2 in Appendix A. Additionally, the coordinate system is defined according to blade coordinate system in DNVGL AS[55]

TABLE 1 Parameter variation range for sensitivity analysis and neural network training

Parameter	Attribute	Max. variation
UD	E_{11}, E_{22}, G_{12}, ν_{12}, ρ	± 10%
Biax 45°	E_{11}, E_{22}, G_{12}, ν_{12}, ρ	± 10%
Biax 90°	E_{11}, E_{22}, G_{12}, ν_{12}, ρ	± 10%
Triax	E_{11}, E_{22}, G_{12}, ν_{12}, ρ	± 10%
Balsa	E_{11}, G_{12}, ν_{12}, ρ	± 10%
Foam	E, G, ρ	± 10%

Parameter	Attribute	Max. variation	Note
$P_{\mathrm{SS,TE,offset}}$	Location	± 10 mm	
$P_{\mathrm{SS,TE,core}}$	Location	± 15 mm	Prefabr.: varied synchr.
$P_{\mathrm{SS,TE,spar\ cap}}$	Location	± 15 mm	
$P_{\mathrm{SS,Mid,spar\ cap}}$	Location	± 15 mm	
$P_{\mathrm{SS,LE,spar\ cap}}$	Location	± 15 mm	
$P_{\mathrm{SS,LE,core}}$	Location	± 15 mm	
$P_{\mathrm{SS,LE,offset}}$	Location	± 10 mm	
$P_{\mathrm{PS,LE,offset}}$	Location	± 10 mm	
$P_{\mathrm{PS,LE,core}}$	Location	± 15 mm	Prefabr.: varied synchr.
$P_{\mathrm{PS,LE,spar\ cap}}$	Location	± 15 mm	
$P_{\mathrm{PS,Mid,spar\ cap}}$	Location	± 15 mm	
$P_{\mathrm{PS,TE,spar\ cap}}$	Location	± 15 mm	
$P_{\mathrm{PS,TE,core}}$	Location	± 15 mm	
$P_{\mathrm{PS,TE,offset}}$	Location	± 10 mm	

Note: The prefabricated spar cap is varied synchronous for each shell side.

TABLE 2 BECAS cross-sectional output parameters

Variable	Description
SC_x, SC_y	Shear center (SC) coordinates
EC_x, EC_y	Elastic center (EC) coordinates
M_{total}	Total mass
CoG_x, CoG_y	Center of gravity (CoG) coordinates
I_{xx}, I_{yy}, I_{xy}	Mass moment of inertia
A_x, A_y	Area center cordinates
A_{xx}, A_{yy}, A_{xy}	Area moment of inertia
A_{total}	Total areas
$\alpha_{PC,Ref}$	Orientation of principal axis
$\alpha_{PC,EC}$	Orientation of principal axis w.r.t. EC
$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} \\ & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} \\ & & K_{33} & K_{34} & K_{35} & K_{36} \\ & & & K_{44} & K_{45} & K_{46} \\ & & & & K_{55} & K_{56} \\ & & & & & K_{66} \end{bmatrix}$	Stiffness matrix
$\begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} & M_{15} & M_{16} \\ & M_{22} & M_{23} & M_{24} & M_{25} & M_{26} \\ & & M_{33} & M_{34} & M_{35} & M_{36} \\ & & & M_{44} & M_{45} & M_{46} \\ & & & & M_{55} & M_{56} \\ & & & & & M_{66} \end{bmatrix}$	Mass matrix

of the overall methodology, enabling also users without in-depth knowledge of the underlying physical models to perform model updating of cross sections.

A threshold value $\lambda = 0.25$ was chosen to identify irrelevant features, which are then excluded in the matrix of Table 3. This led to discarding the input features E_{22} and the Poisson's ratio ν of each material. The Young's modulus E of both core materials is sorted out as its magnitude only reaches a fraction <0.2% of the glass fiber laminates. As the prefabricated spar cap was moved simultaneously, the algorithm additionally rejects 4 of 5 linearly dependent division points, keeping the $P_{\mathrm{XX,Mid,spar\ cap}}$ as representative for each shell side.

TABLE 3 Reduced sensitivity matrix of cross section at R = 6 m

	SC_x	SC_y	EC_x	EC_y	CoG_x	CoG_y	$Area_x$	$Area_y$	A_{xx}	A_{yy}	A_{xy}	$Area_{total}$	$\alpha_{PC,Ref}$	$\alpha_{PC,EC}$	K_{11}	K_{12}	K_{22}
$E_{11,\ UD}$	**0.8**	**0.8**	**4.6**	**4.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**2.8**	**2.9**	**0.1**	0.0	**0.2**
ρ_{UD}	0.0	0.0	0.0	0.0	**3.5**	**2.6**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\ Biax45}$	**2.0**	**2.3**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**5.5**	0.0	0.0
ρ_{Biax45}	0.0	0.0	0.0	0.0	**0.6**	**0.4**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$E_{11,\ Biax90}$	**0.4**	**0.3**	**1.9**	**1.7**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**1.1**	**1.2**	0.0	0.0	**0.1**
$G_{12,\ Biax90}$	**0.6**	**0.7**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.7**	**1.3**	**1.9**
ρ_{Biax90}	0.0	0.0	0.0	0.0	**0.9**	**0.6**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$E_{11,\ Triax}$	**0.3**	**0.5**	**2.7**	**2.4**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**1.7**	**1.7**	**0.1**	0.0	**0.1**
$G_{12,\ Triax}$	**1.9**	**2.2**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**2.1**	**3.8**	**5.5**
ρ_{Triax}	0.0	0.0	0.0	0.0	**0.9**	**0.6**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\ Balsa}$	**0.7**	**0.8**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.1**	**0.6**	**0.7**
ρ_{Balsa}	0.0	0.0	0.0	0.0	**3.9**	**4.2**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G_{Foam}	**0.2**	**0.2**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.1**	0.0	0.0
ρ_{Foam}	0.0	0.0	0.0	0.0	**1.7**	**2.4**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$P_{SS,TE,offset}$	**0.2**	0.0	**0.1**	0.0	**0.3**	**0.3**	**2.8**	**2.8**	**3.2**	**3.3**	**3.4**	**2.9**	0.0	0.0	0.0	0.0	0.0
$P_{SS,Mid,spar\ cap}$	**1.0**	**3.3**	**0.8**	**2.3**	**0.1**	**1.2**	**1.1**	**1.9**	**1.9**	**1.2**	**2.0**	**0.5**	**3.2**	**3.5**	0.0	**3.2**	**0.3**
$P_{SS,LE,offset}$	**0.1**	0.0	0.0	**0.1**	**0.1**	**0.2**	**0.9**	**2.4**	**1.8**	**0.3**	**0.2**	**2.9**	0.0	0.0	0.0	0.0	0.0
$P_{PS,LE,offset}$	0.0	**0.1**	**0.1**	0.0	**0.3**	**0.2**	**4.2**	**2.3**	**1.7**	**3.4**	**1.9**	**2.9**	0.0	0.0	0.0	0.0	0.0
$P_{PS,Mid,spar\ cap}$	**4.9**	**3.3**	**1.4**	**2.2**	**1.3**	**1.2**	**1.0**	**2.0**	**1.9**	**0.9**	**2.2**	**0.3**	**3.5**	**3.1**	**0.1**	**2.8**	**0.2**
$P_{PS,TE,offset}$	**0.3**	0.0	0.0	0.0	**0.4**	**0.4**	**2.6**	**2.8**	**3.3**	**3.1**	**3.2**	**3.0**	0.0	0.0	0.0	0.0	0.0

Note: Values above the the threshold $\lambda = 0.25$ are marked bold and only the finally selected features are displayed. A maximum of ±10% variation applies for material input features, ±15 mm for the spar cap division points $P_{XX,Mid,spa\ cap}$, and ±10 mm for the other division points.

TABLE 3 (Continued)

	K_{33}	K_{34}	K_{44}	K_{35}	K_{45}	K_{55}	K_{16}	K_{26}	K_{66}	M_{11}	M_{44}	M_{45}	M_{55}	M_{16}	M_{26}	M_{66}
$E_{11,\ \mathrm{UD}}$	**4.7**	0.0	0.0	**2.2**	**0.2**	**5.4**	**0.7**	**0.8**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{UD}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**3.4**	0.0	**0.1**	**4.7**	0.0	**0.9**	**0.9**
$G_{12,\ \mathrm{Biax45}}$	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Biax45}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.5**	0.0	0.0	0.2	0.0	0.1	0.0
$E_{11,\ \mathrm{Biax90}}$	**1.7**	**2.7**	**3.1**	**2.8**	**2.2**	**1.0**	**0.3**	**0.5**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\ \mathrm{Biax90}}$	0.0	0.0	0.0	0.0	0.0	0.0	**1.0**	**1.6**	**1.9**	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Biax90}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**2.0**	**2.5**	**2.2**	**1.4**	**1.6**	**1.9**	**2.5**
$E_{11,\ \mathrm{Triax}}$	**3.1**	**4.4**	**5.0**	**4.4**	**3.7**	**2.1**	**0.4**	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\ \mathrm{Triax}}$	0.0	0.0	0.0	0.0	0.0	0.0	**3.1**	**4.8**	**5.6**	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Triax}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**3.2**	**3.6**	**3.2**	**2.5**	**2.3**	**2.6**	**3.7**
$G_{12,\ \mathrm{Balsa}}$	0.0	0.0	0.0	0.0	0.0	0.0	**0.8**	**0.9**	**0.4**	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Balsa}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**2.7**	**3.8**	**3.7**	**1.9**	**4.8**	**4.7**	**3.7**
G_{Foam}	0.0	0.0	0.0	0.0	0.0	0.0	**0.3**	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Foam}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**1.3**	**0.5**	0.2	**0.9**	**1.1**	**0.6**	**0.6**
$P_{\mathrm{SS,TE,offset}}$	0.0	0.0	0.0	0.1	0.1	0.0	0.0	0.8	0.0	0.1	**0.4**	**0.3**	0.0	**0.3**	**0.3**	**0.3**
$P_{\mathrm{SS,Mid,spar\ cap}}$	0.0	**2.0**	0.0	**0.8**	**2.7**	0.1	**3.5**	**1.4**	0.1	0.0	0.0	**1.7**	0.0	**0.9**	0.0	0.0
$P_{\mathrm{SS,LE,offset}}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.3**	0.0	0.1	0.1	0.0	0.0	0.1	0.0	0.1
$P_{\mathrm{PS,LE,offset}}$	0.0	0.0	0.0	0.1	0.0	0.0	0.1	0.1	0.0	0.1	0.1	0.1	0.0	0.1	0.2	0.1
$P_{\mathrm{PS,Mid,spar\ cap}}$	0.0	**2.0**	0.0	**1.4**	**2.9**	0.2	**3.3**	**1.4**	0.1	0.1	0.0	**1.8**	**0.3**	**0.9**	**1.0**	0.1
$P_{\mathrm{PS,TE,offset}}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**1.2**	0.0	0.1	**0.4**	**0.4**	0.0	**0.3**	**0.3**	**0.3**

Note: Values above the the threshold $\lambda = 0.25$ are marked bold and only the finally selected features are displayed. A maximum of ±10% variation applies for material input features, ±15 mm for the spar cap division points $P_{\mathrm{XX,Mid,spa\ cap}}$, and ±10 mm for the other division points.

Regarding the output, the algorithm sorts out several insensitive stiffness terms (K_{13}, K_{23}, K_{14}, K_{24}, K_{15}, K_{25}, K_{36}, K_{46}, K_{56}). As only isotropic and orthotropic materials aligned with the CS normal axis are applied in the blade design and no additional off-axis layers to induce, for example, bend-twist coupling effects, the respective coupling terms in the stiffness matrix are zero, and thus, the sensitivity is zero. Considering the mass matrix, the features M_{12}, M_{13}, M_{23}, M_{14}, M_{24}, M_{15}, M_{25}, M_{36}, M_{46}, and M_{56} are always zero.[56] Lastly, due to linear dependencies, M_{11} was kept, but M_{total}, M_{22}, and M_{33} were discarded; all these features represent the total mass. Additionally I_{xx}, I_{yy}, I_{xy}, M_{34}, and M_{35} were sorted out in favor of M_{44}, M_{55}, M_{45}, M_{16}, and M_{26}, as these again represent the same physical parameter, respectively. This feature selection yields the final reduced sensitivity matrix in Table 3. However, all parameters are varied for the sample generation for training and validation of the cINN, only the remaining selected 14 input features and 33 output features from the respective samples are passed as input and conditions to the cINN.

The physical soundness of the input–output relations is discussed in the following. The explanations are exemplarily given for the impact of the input parameters $E_{11,\,UD}$ (axial Young's modulus of the spar caps), $G_{12,\,Biax45}$ (in-plane shear modulus of the biax layers in the shear web), and $G_{12,\,Triax}$ (in-plane shear modulus of the triax layers in the shell) on the output parameters. For a complete picture, the reader is referred to existing literature on physical modeling employing generalized composite Timoshenko beam formulations.[56,57]

There is a strong separation of functions for the different blade subcomponents. The spar caps shall provide stiffness against flapwise bending and carry bending-related axial normal stresses. Hence, $E_{11,\,UD}$ should have an impact on the related stiffness matrix entry K_{55}. Additional axial stiffness in the spar caps will also affect the stiffness matrix entry linked to axial stretching, K_{33}, and the position of the elastic center, EC_x and EC_y, respectively. The function of the shear web is to carry lateral forces (and the respective shear stresses) in flapwise direction (perpendicular to the rotor plane). Therefore, the in-plane shear modulus of its biax layers, $G_{12,\,Biax45}$, shall contribute to the corresponding stiffness matrix entry K_{11} and the shear center location. The shell is designed to withstand the shear stresses, especially due to torsion, and due to lateral forces, primarily in edgewise direction (parallel to the rotor plane) and secondarily in flapwise direction. Modification of the in-plane shear modulus in the triax layers positioned in the shell should therefore have an impact on the related stiffness matrix entries, K_{11}, K_{22}, and K_{66}, and the related coupling entries K_{12}, K_{16}, and K_{26}, respectively. The position of the shear center should also be affected. All of these considerations are confirmed by Table 3. It can thus be concluded that the physical model that gives the outputs as a function of the inputs is physically meaningful.

4 | INN STRUCTURE, TRAINING, AND EVALUATION

After having identified the significant in- and output features of the model, in this section, we seek appropriate cINN hyperparameters for the subsequent training and evaluation of the *INN*. Furthermore, the network is selected with respect to its computational training costs and applicability on other related scenarios.

4.1 | Identifying network hyperparameters

While the network parameters (i.e., the network weights) are produced by the training process, the network size and structure, the length of the training procedure, and other settings have to be set by the user beforehand. These are known as hyperparameters, and we describe our choices in the following sections. All the programming is done within Pytorch[58] including the FrEIA for INNs.[40] As previously stated, the cINN is a sequence of conditional affine coupling blocks (CC), with subnetworks acting as internal functions. The subnetworks are represented by standard feed-forward neural networks consisting of a number of hidden layers (network depth), each with a certain number of nodes (network width). Every hidden layer is followed by a dropout layer to improve generalization and an activation layer. To find a set of well-performing hyperparameters for the cINN, we trained networks with various different depths and widths as depicted in Table 4. The hyperparameter tuning revealed that shallow but wide subnetworks are favorable for this application. The AdaGrad[59] optimization algorithm gave the best and fastest convergence, which is finally improved by a learning rate scheduler. Due to huge magnitude differences between the features, all passed samples are standardized per feature ($\bar{x} = 0$ and $\sigma_x = 1$) to equalize the contribution magnitude of each one. For further information on ANN's terminology, please refer for example to Chollet.[17]

In order to determine the necessary network depth of the cINN, we will evaluate the four trained models from Table 4 against their prediction quality of the input feature's posterior. If not otherwise stated, all given results will be shown as standardized values to improve direct comparability between the features. Considering that all input features were sampled uniformly within their respective symmetric maximum variation $\pm x_{max}$ from Table 1, the standard deviation is defined as follows:

$$\sigma_x = |x_{max}| \cdot \sqrt{\frac{1}{3}} \tag{7}$$

Equation (7) helps to estimate the real range of the respective input feature's posterior distribution. Furthermore, to enhance the understanding of the upcoming discussion, we use Figure 7 to explain the interpretation of the results for two exemplarily chosen features: $E_{11,Triax}$ and ρ_{Biax90}. The left two graphs represent the posterior distribution of the respective features after evaluating the cINN inversely. Each sample is a set

TABLE 4 Final cINN hyperparameter set

Model	No. CC	Subnet nodes	Subnet layer	Activation function	Dropout rate	Optimizer	Learning rate	Batch size	Epochs	Samples	Training time
0	2	100	1	PReLu	0.05	AdaGrad	0.2	32	1000	20 000	50 min
1	4	200	1	PReLu	0.05	AdaGrad	0.2	32	1000	20 000	107 min
2	8	400	1	PReLu	0.05	AdaGrad	0.2	32	1000	20 000	168 min
3	16	800	1	PReLu	0.05	AdaGrad	0.1	32	1000	20 000	279 min

Note: The parameters where chosen subjectively, involving the number of epochs and the learning rate and its scheduler, although focusing more on computationally cheaper hyperparameter sets may also achieve reasonably good accuracy with lower computational costs.

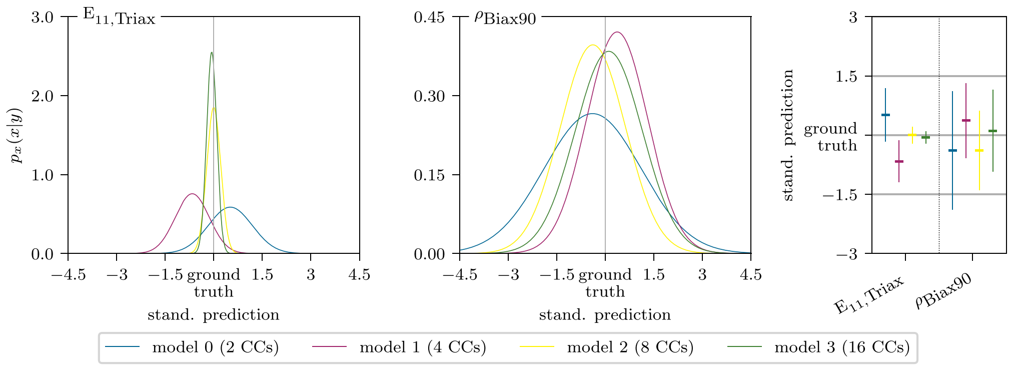

FIGURE 7 Exemplary standardized prediction for the two input features $E_{11,Triax}$ and ρ_{Biax90} of one sample computed with all four models from Table 4. The left two graphs show the predicted distribution of the corresponding feature x. The right graph summarizes the same results as error bars with $1 \cdot \sigma_x$ width around the predicted mean value $\bar{x}$. The results are related to the ground truth value

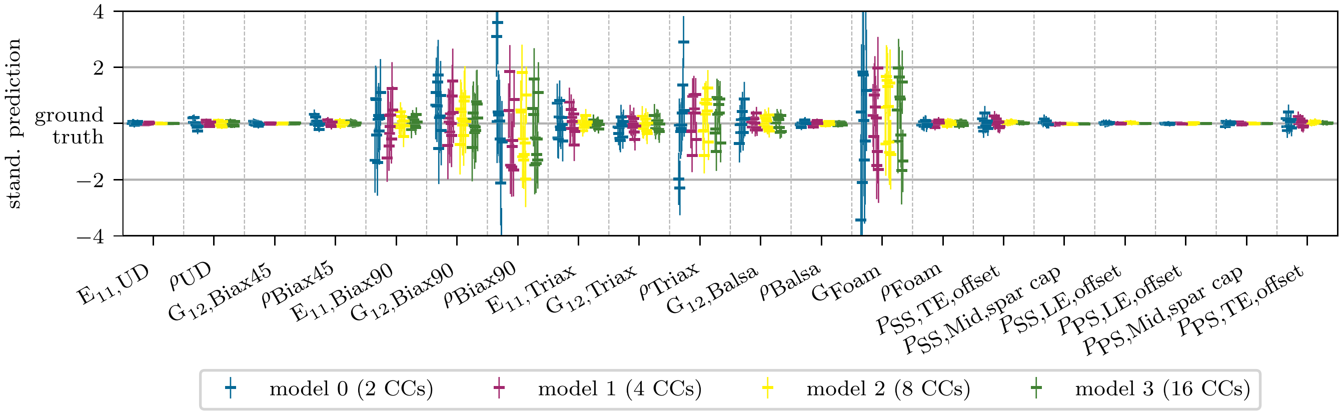

FIGURE 8 Input feature prediction showed for 10 random samples of four different models with full output as conditional features. The four *invertible neural networks* increase in depths

of input and output features. This original input feature value is the so-called ground truth, which the cINN tries to predict as accurately as possible. Therefore, all given results are related to the ground truth value of each sample feature. This enables the comparison of different samples with varying ground truth values, and it lets the reader recognize the accuracy of the prediction at a glance.

What is striking about the left graph is the improved prediction with increasing network depth. The mean value approximates the ground truth with increasing depth, while the standard deviations describe narrower distributions, with the shallowest model ($\sigma_{E_{11,Triax}} = 0.72$) is much higher than that of the deepest ($\sigma_{E_{11,Biax90}} = 0.16$). That means the shallowest model has a poor prediction confidence for $E_{11,Triax}$ compared to the deeper models. In contrast, the input feature ρ_{Biax90} is predicted similarly by all models. Here, the width of the posteriors is generally also much higher, indicating that ρ_{Biax90} cannot be recovered with as great of a precision as $E_{11,Triax}$, even for the deeper and more powerful cINN architecture. Another interesting fact is that the posteriors are all approximately Gaussian (as opposed to having multiple peaks, skewed shape, etc.). This could help justify even simpler methods in future that may only provide Gaussian uncertainty estimates. Without producing the full nonparametric posteriors first with the cINN, such simplifying assumptions could not be made. Both first graphs can be summarized as presented in the right graph. There, the prediction moves to the y-axis, and for each feature, the posterior is depicted as error bars with $1\cdot\sigma_x$ width around the mean value $\bar{x}$, making it easy to compare several features and models at a glance.

Having explained how to interpret the inverted model results, we will now move on to define the best network depth from the given models in Table 4. Therefore, Figure 8 shows all four models' prediction of each input feature's posterior for 10 randomly chosen samples. It is directly apparent from this figure that most of the features are predicted extremely accurately with a high confidence by the deeper models, except for both densities of the glass fiber plies Biax90 and Triax and the shear modulus G_{12} of Biax90 and Foam.

The two density features cannot be recovered accurately enough by any of the given models due to an ambiguity resulting from their relatively similar mass contribution and quasi identical position in the cross section. Recovering from Figure 6 above and Tables A1 and A2 in Appendix A, the Triax and Biax90 are placed directly upon each other in the shell sandwich laminate with a similar nominal thickness. Thus, they counteract each other, that is, if one density increases, the other decreases to achieve the same total weight and inertia contribution together. This behavior is clearly confirmed by Figure 9, where the mean value of the learned posterior of the samples is scattered along a thin line against each other; that is, the features are negatively proportional and highly correlated $R^2 = 0.9985$. From this, we can conclude that the cINN has correctly detected the ambiguity and represents it accordingly in the posterior. However, it is able to predict a merged density of both values quite precisely, as following results will show. Another interesting point due to a greater nominal thickness ($+38\%$) and its existence in the prefabricated spar cap the Triax is slightly more dominant, which is reflected in a marginally better prediction and thus narrower posterior distribution compared to the Biax90 (cf. Figure 8).

As stated above, the overall inference performance of the models is strikingly accurate, especially from model 1 (4 CCs) on. Figure 8 shows similar result qualities for models 1–3, with only minor improvements in the standard deviation in each step. In addition, Table 4 describes a computational time increase by 57% from model 1 to 2 and approx. 66% from model 2 to 3. Thus, the authors decided to choose the model 2 (8 CCs) hyperparameter set as the cINN design. In addition to the depth selection, the necessary training sample size and number of epochs for a fixed learning rate of 0.2 are analyzed to cut down computational costs even further. All scenarios showed satisfying posteriors; therefore, a sample size of 10 000 and 1000 epochs was chosen as a trade-off between computational time (112 min) and accuracy. Table 5 summarizes the final cINN

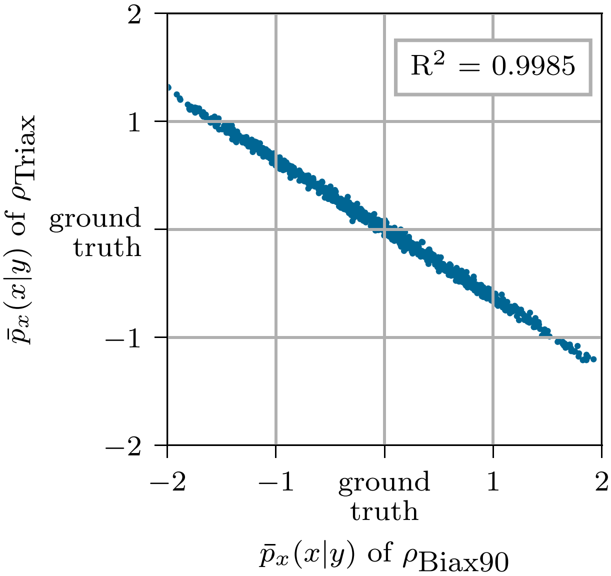

FIGURE 9 Counteraction of the mean predicted densities ρ_{Biax90} and ρ_{Triax}. The highly correlating samples show that the cINN correctly detects this ambiguity

TABLE 5 Final cINN hyperparameter set

No. CC	Subnet nodes	Subnet layer	Activation function	Dropout rate	Optimizer	Learning rate	Batch size	Epochs	Samples	Training time
8	400	1	PReLu	0.05	AdaGrad	0.2	32	1000	10 000	112 min

Note: The parameters where chosen subjectively, involving the number of epochs and the learning rate and its scheduler, although focusing more on computationally cheaper hyperparameter sets may also achieve reasonably good accuracy with lower computational costs.

hyperparameter set. Considering the complete process to create a cINN, the sample generation has to be taken into account, which is a significant cost driver for classical iterative model updating techniques. On a 40-node computing cluster, generating 10 000 samples with MoCA and BECAS with the given mesh density of 500 elements per circumference takes approx. 38 min.

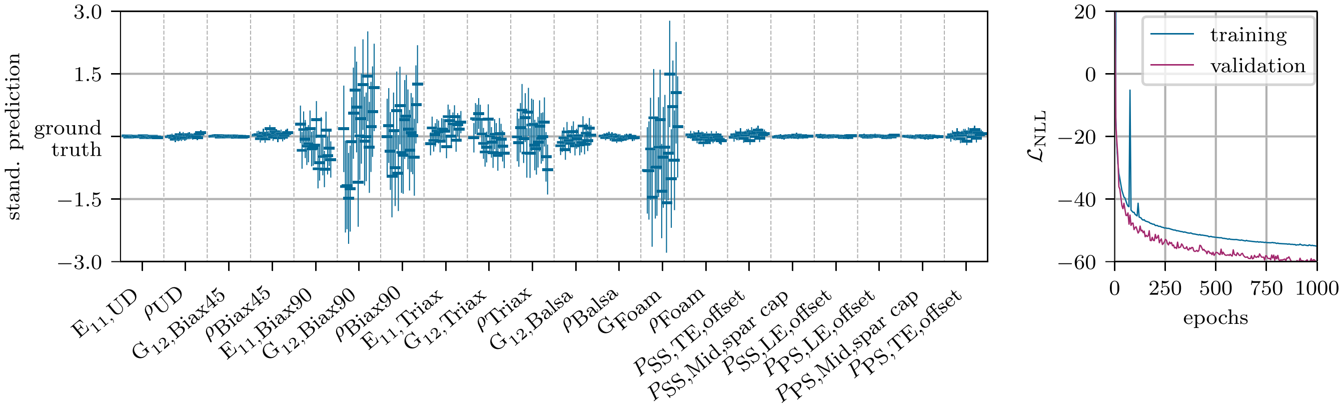

FIGURE 10 Final model: input feature prediction on the left graph for 30 random samples with full output as conditional features. Right graph depicts the negative logarithmic likelihood loss curve of the cINN for training and validation samples

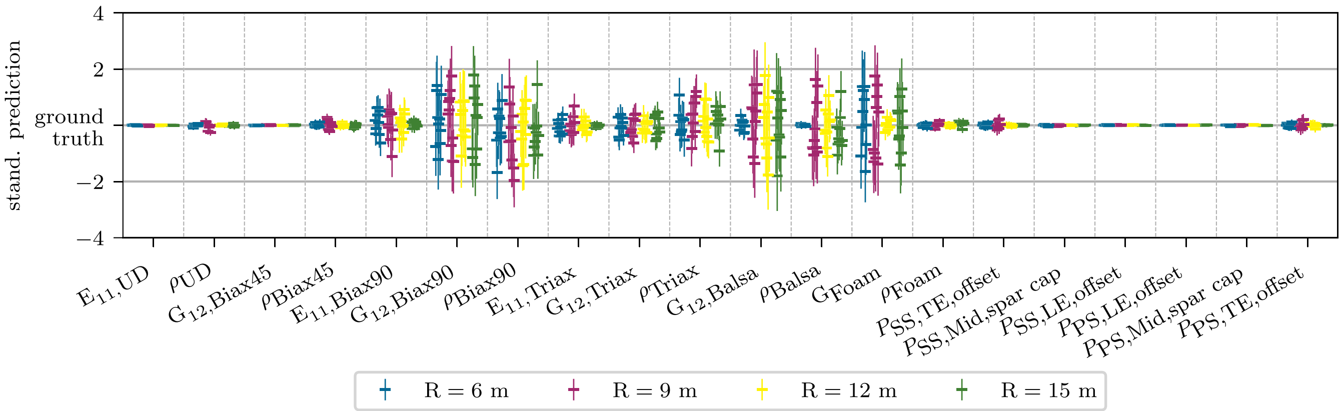

FIGURE 11 Input feature prediction showed for 10 random samples at four different radial positions with full output as conditional features. The four cINNs were each trained individually for their respective radius

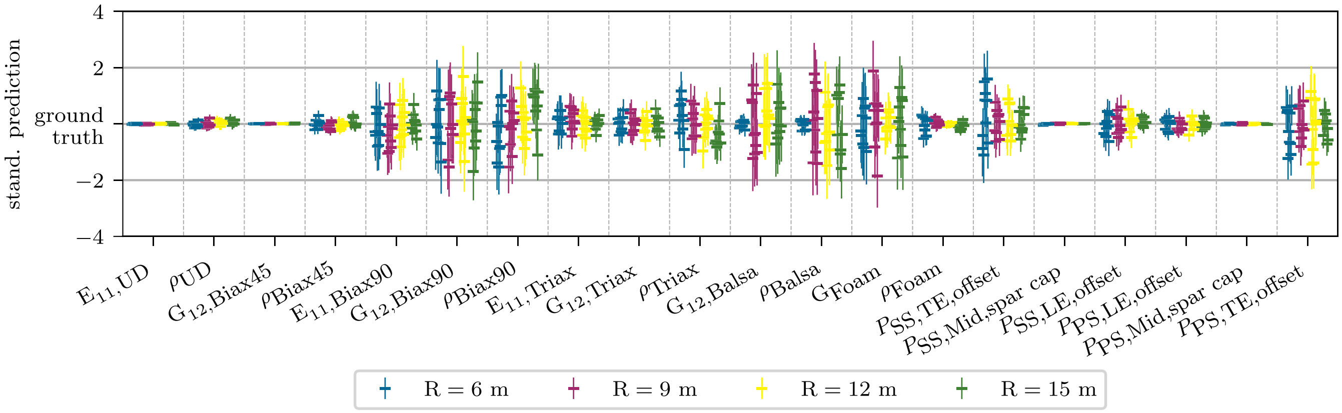

FIGURE 12 Input feature prediction showed for 10 random samples at four different radial positions only considering the stiffness and mass matrix as conditional features. The four cINNs were each trained individually for their respective radius

4.2 | Invertible network evaluation

After having defined our cINN, this section will continue with the evaluation of its performance and applicability for different scenarios. First of all, we will recap the input feature prediction of the finally chosen model and its training process. The good prediction quality is again apparent from the data in the left graph of Figure 10, with the exception of the two counteracting input features: $G_{12,\text{Biax90}}$, G_{Foam}, ρ_{Biax90}, and ρ_{Triax}. The latter two features are mostly unrecoverable due to an inherent ambiguity of the problem, as discussed above. During training, the NLL loss curve on the right graph is monotonically decreasing with little steps every 100th epoch, where the scheduler reduces the learning rate by 20% of the actual rate. The validation loss is even lower due to the averaging effect of the dropout layers mimicking multiple trained models.[54] However, both seem to have nearly converged to their optimum.

In order to extend the analysis of the posterior distribution, the correlation of an input feature prediction (mean value $\bar{x}$) against its ground truth value can be investigated. Therefore, we use the coefficient of determination (R^2). Figure B1 in Appendix B compares the linear correlation of each inferred input feature for 1000 random samples, also stating each feature's R^2 score, which in most cases is $R^2 > 0.92$, with the exception previously named. The computed correlation confirms the previous outstanding predictions for a wide range of samples. Next, these predicted mean values are used to recalculate the output features with MoCA and BECAS to evaluate its accuracy. All recalculated values match extremely well the ground truth values as proved by the given R^2 scores, which are all approx. 1 with a roundoff error at the fifth decimal digit. Interestingly, the inaccuracies of ρ_{Biax90} and ρ_{Triax} seem to cancel each other out due to their counteraction. Considering the proximity of both laminates and that both are infused together, a merged or averaged density for both materials could improve the prediction of such a parameter in future applications.

So far, we have demonstrated an excellent model updating capability of the finally designed cINN for a cross section at $R = 6$ m. Hereafter, the same cINN will be trained for cross sections at the following positions: $R = 9$ m, 12 m, and 15 m. Figure 11 presents the posterior predictions for all four radial locations considering 10 random samples. Here again, the predictions are outstanding, except for ρ_{Balsa}. This rests upon the fact that balsa is replaced by foam in the trailing edge panels after $R = 6$ m and only appears in the transition pieces of the spar cap, as shown exemplarily in Figure B2 in Appendix B. Hence, the contribution to any mass property and sensitivity is comparatively low to gain enough information to recover this input feature. The wide posteriors produced by the cINN show that it has correctly understood and modeled this uncertainty, instead of having the same high confidence as for the other parameters.

As a final aspect, the possible updated features for a future case of a finite element beam model updating would be the stiffness and mass matrix. Therefore, analyzing the prediction quality trained only with these two matrices as conditional features indicates if the basic material and layup input features can be subsequently inferred. Figure 12 shows the already familiar posterior prediction graph for the four previously analyzed CS positions, only considering the stiffness and mass matrix features. The overall width of the posterior distributions increases slightly for all cross sections and features but is still reasonably accurate. Only the original cINN for the cross section at $R = 6$ m noticeably loses accuracy at several features ($P_{\text{SS,TE,offset}}$, $P_{\text{SS,LE,offset}}$, $P_{\text{PS,LE,offset}}$, and $P_{\text{PS,TE,offset}}$). This only presents the easiest and most straight forward way of inferring the input posteriors, though recovering the full output parameter set from the stiffness and mass matrix before inferring the input features is also possible with a few calculations.[56,57]

5 | CONCLUSION

This feasibility study set out to reveal the capability of *INNs* to be successfully applied in the field of wind turbine blade structural model updating. The study was based on an example of blade cross sections, being one of the first structural model levels of rotor blades.

A feature selection was carried out using a sensitivity analysis that yielded a sensitivity matrix. This analysis covered a limited set of input parameters, including material properties such as Young's moduli, shear moduli, Poisson's ratios, and densities of all materials and layup variations (CS layup division points). All parameters were varied within extended manufacturing tolerances. Based on the sensitivity matrix and a chosen threshold value, the significant input and output features were identified. Although from an engineering point of view most of the sensitivities might be deducible, the sensitivity analysis in the overall approach is universally applicable to retrieve sensitive parameters for the feature selection without a priori knowledge of the physical model behavior. The physical soundness of the underlying physical model was discussed though and shown by exemplary input–output relations.

Furthermore, the general architecture and principles of an *cINN* were explained. Subsequently, the necessary cINN structure was investigated considering the trade-off between computational time and prediction accuracy. A cINN with shallow feedforward subnetworks was selected that took approx. 140 min for sample generation and training on a computing cluster equipped with 40 CPUs and an NVIDIA Tesla P100 GPU. In contrast to optimization methods for model updating, this cINN finally establishes an inverse model of the physical model represented by MoCA and BECAS, which can be evaluated any time without having to perform the complete process again.

A total number of 10 000 samples based on randomly varied input feature sets were generated with MoCA and BECAS for testing. The cINN inferred remarkably accurate input feature values from the given test samples, except for two ambiguous density values of glass fiber-reinforced plastic materials. The results revealed that the ambiguity rests upon the two densities counteracting each other. However, a recalculation of output parameters from the inferred values affirmed again correct predictions. The inaccuracies of the densities canceled each other out. Another advantage over classical model updating techniques is that cINNs generate posterior distribution and not single values. This gives the user an instrument to evaluate the model's confidence on the predicted value and to reveal unrecoverable parameters. These findings were further confirmed by investigating cross sections at different radial positions, showing similarly accurate results. The study found the posterior for the density of balsa was very wide for the other radii, which is due to a significantly reduced balsa application in these cross sections and thus lower contribution to the mass-related output features, and therefore a source of uncertainty correctly captured by the cINN. Moreover, this paper studied a reduced output feature set, training the model only with stiffness and mass matrix as conditional feature. This scenario becomes relevant whenever a finite beam model updating can predict these values and a further inference to the material and layup level is desired. Here, the standard deviation of the posterior distribution increases slightly, that is, the confidence of the prediction diminishes. However, the results are still satisfactory.

In conclusion, this feasibility study was able to show that cINNs are generally applicable and provide good results. The example of wind turbine blade CS model updating proved outstanding performance for cINNs in this research field. Although we have limited the parameter space and model complexity, the cINN is flexibly scalable to cover further parameters, for example, parameters linked to geometry or adhesive joints, as well as more complex models. This fact and the excellent results make model updating with cINN a feasible and promising approach to tackle more complex model updating problems. In ongoing research, the authors work on the extension of the updated parameter space, the increase of the model complexity to a full-scale blade model, and to update blade models with experimental data. A successful application of this methodology on the global blade model and having an extensive inverse model of the blade can offer several application opportunities, for example, digital twins and/or structural health monitoring by continuous training and updating or quality assurance after manufacturing.

ACKNOWLEDGEMENTS

This work was supported by the compute cluster, which is funded by the Leibniz University Hannover, the Lower Saxony Ministry of Science and Culture (MWK), and the German Research Foundation (DFG). This work was also supported by the Federal Ministry for Economic Affairs and Energy of Germany (BMWi) projects SmartBlades2 (grant number 0324032C) and ReliaBlade (grant number 0324335B).

CONFLICT OF INTEREST

The authors declare that they do not have any conflicts of interests.

AUTHOR CONTRIBUTIONS

P.N-C. and L.A. performed the conceptualization and helped with the software. P.N-C. performed the methodology, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, and visualization. P.N-C., L.A., and C.B. did the and writing—review and editing. C.B. did the supervision, project administration, and funding acquisition. All authors have read and agreed to the published version of the manuscript.

ABBREVIATIONS

The following abbreviations are used in this manuscript:ANN artificial neural network
BECAS beam cross section analysis software
CC conditional coupling block
cINN conditional invertible neural network
CS cross section
EC elastic center
FrEIA Framework for Easily Invertible Architectures
CoG center of gravity
INN invertible neural network
K stiffness matrix
LE leading edge
MoCA Model Creation and Analysis Tool for Wind Turbine Rotor Blades
M mass matrix
NLL negative logarithmic likelihood
PS pressure side
SC shear center
SS suction side
TE trailing edge
UD unidirectional

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1002/we.2687.

DATA AVAILABILITY STATEMENT

Data available are in a publicly accessible repository: https://github.com/IWES-LUH/CS-ModelUpdating-cINN.

ORCID

Pablo Noever-Castelos https://orcid.org/0000-0002-3353-6165
Lynton Ardizzone https://orcid.org/0000-0001-9777-1773
Claudio Balzani https://orcid.org/0000-0003-3432-3476

REFERENCES

1. Gundlach J, Govers Y. Experimental modal analysis of aeroelastic tailored rotor blades in different boundary conditions. *J Phys Conf Ser*. 2019;1356: 12023.
2. Mottershead JE, Friswell MI. Model updating in structural dynamics: A survey. *J Sound Vib*. 1993;167(2):347-375.
3. Knebusch J, Gundlach J, Govers Y. A systematic investigation of common gradient based model updating approaches applied to high-fidelity test-data of a wind turbine rotor blade. In: Proceedings of the xi international conference on structural dynamics. EASDAthens; 2020:2159-2174.
4. Luczak M, Manzato S, Peeters B, Branner K, Berring P, Kahsin M. Updating finite element model of a wind turbine blade section using experimental modal analysis results. *Shock Vib*. 2014;2014:1-12.
5. Schröder K, Grove S, Tsiapoki S, Gebhardt CG, Rolfes R. Structural change identification at a wind turbine blade using model updating. *J Phys Conf Ser*. 2018;1104:12030.
6. Sayer F, Antoniou A, Goutianos S, Gebauer I, Branner K, Balzani C. Reliablade project: a material's perspective towards the digitalization of wind turbine rotor blades. *IOP Conf Ser: Mater Sci Eng*. 2020;942:12006.
7. Yang YB, Chen YJ. A new direct method for updating structural models based on measured modal data. *Eng Struct*. 2009;31(1):32-42.
8. Bruns M, Hofmeister B, Grießmann T, Rolfes R. Comparative study of parameterizations for damage localization with finite element model updating. In: Proceedings of the 29th european safety and reliability conference (esrel) Beer M, Zio E, eds. Research Publishing Services; 2019; Singapore:1125-1132.
9. Hofmeister B, Bruns M, Rolfes R. Finite element model updating using deterministic optimisation: a global pattern search approach. *Eng Struct*. 2019; 195:373-381.
10. Mojtahedi A, Lotfollahi Yaghin MA, Hassanzadeh Y, Ettefagh MM, Aminfar MH, Aghdam AB. Developing a robust shm method for offshore jacket platform using model updating and fuzzy logic system. *Appl Ocean Res*. 2011;33(4):398-411.
11. Fuchs C, Spolaor S, Nobile MS, Kaymak U. A swarm intelligence approach to avoid local optima in fuzzy c-means clustering. In: 2019 ieee international conference on fuzzy systems (fuzz-ieee). IEEE; 2019:1-6.
12. Sun H, Büyüköztürk O. Bayesian model updating using incomplete modal data without mode matching. In: Health monitoring of structural and biological systems 2016, SPIE Proceedings. SPIE; 2016:98050D.
13. Sunnåker M, Busetto AG, Numminen E, Corander J, Foll M, Dessimoz C. Approximate bayesian computation. *PLoS Comput Biol*. 2013;9(1):e1002803.
14. Serna A, Bucher C. Advanced surrogate models for multidisciplinary design optimization. In: 6th weimar optimization and stochastic days 2009; 2009.
15. van Damme CI, Allen MS, Hollkamp JJ. Updating geometrically nonlinear reduced-order models using nonlinear modes and harmonic balance. *AIAA J*. 2020;58(8):3553-3568.
16. Trehan S, Carlberg KT, Durlofsky LJ. Error modeling for surrogates of dynamical systems using machine learning. *International Journal for Numerical Methods in Engineering*. 2017;112(12):1801-1827.
17. Chollet F. *Deep learning with python*, Safari Tech Books Online. Shelter Island, NY: Manning; 2018. http://proquest.safaribooksonline.com/9781617294433
18. Marwala T. *Finite-element-model updating using computional intelligence techniques: Applications to structural dynamics*. London: Springer; 2010.
19. Sung H, Chang S, Cho M. Reduction method based structural model updating method via neural networks. In: Aiaa scitech 2020 forum. American Institute of Aeronautics and Astronautics; 2020; Reston, Virginia.
20. Sung H, Chang S, Cho M. Efficient model updating method for system identification using a convolutional neural network. *AIAA J*. 2021:59(9):1-10.
21. Lu Y, Tu Z. A two-level neural network approach for dynamic fe model updating including damping. *J Sound Vib*. 2004;275(3-5):931-952.
22. Goller B, Broggi M, Calvi A, Schuëller GI. A stochastic model updating technique for complex aerospace structures. *Finite Elem Anal Des*. 2011;47(7): 739-752.
23. Yin T, Zhu H-P. An efficient algorithm for architecture design of Bayesian neural network in structural model updating. *Comput-Aided Civil Infrastruct Eng*. 2020;35(4):354-372.
24. Han B, Kong X, Zhang Z, Zhou L. Neural network model predictive control optimisation for large wind turbines. *IET Gener Transm Distrib*. 2017;11(14): 3491-3498.
25. Jafarnejadsani H, Pieper J, Ehlers J. Adaptive control of a variable-speed variable-pitch wind turbine using radial-basis function neural network. *IEEE Trans Control Syst Technol*. 2013;21(6):2264-2272.
26. Saenz-Aguirre A, Zulueta E, Fernandez-Gamiz U, Lozano J, Lopez-Guede J. Artificial neural network based reinforcement learning for wind turbine yaw control. *Energies*. 2019;12(3):436.
27. Gambier A, Behera A. Modelling the aerodynamic coefficients of wind turbines by using neural networks for control design purposes. *J Phys Conf Ser*. 2018;1037:32032.
28. Shihavuddin ASM, Chen X, Fedorov V, et al. Wind turbine surface damage detection by deep learning aided drone inspection analysis. *Energies*. 2019; 12(4):676.

29. Bangalore P, Tjernberg LB. An approach for self evolving neural network based algorithm for fault prognosis in wind turbine. In: 2013 ieee grenoble conference. IEEE; 2013:1-6.
30. Malik H, Mishra S. Application of probabilistic neural network in fault diagnosis of wind turbine using fast, turbsim and simulink. *Procedia Comput Sci*. 2015;58:186-193.
31. Malik H, Mishra S. Artificial neural network and empirical mode decomposition based imbalance fault diagnosis of wind turbine using turbsim, fast and simulink. *IET Renew Power Gener*. 2017;11(6):889-902.
32. Lu Y, Sun L, Zhang X, Feng F, Kang J, Fu G. Condition based maintenance optimization for offshore wind turbine considering opportunities based on neural network approach. *Appl Ocean Res*. 2018;74:69-79.
33. Qiu B, Lu Y, Sun L, Qu X, Xue Y, Tong F. Research on the damage prediction method of offshore wind turbine tower structure based on improved neural network. *Measurement*. 2020;151:107141.
34. Dinh L, Krueger D, Bengio Y. Nice: Non-linear independent components estimation. http://arxiv.org/pdf/1410.8516v6; 2014.
35. Dinh L, Sohl-Dickstein J, Bengio S. Density estimation using real nvp. http://arxiv.org/pdf/1605.08803v3; 2016.
36. Kingma DP, Dhariwal P. Glow: generative flow with invertible 1x1 convolutions. In: Curran Associates I, ed. *Advances in neural information processing systems*; 2018.
37. Rezende D, Mohamed S. Variational inference with normalizing flows. In: Proceedings of the 32nd international conference on machine learning; 2015; Lille, France.
38. Jacobsen J-H, Smeulders A, Oyallon E. i-revnet: Deep invertible networks. ICLR 2018 - International Conference on Learning Representations, http://arxiv.org/pdf/1802.07088v1; 2018.
39. Ardizzone L, Lüth C, Kruse J, Rother C, Köthe U. Guided image generation with conditional invertible neural networks. http://arxiv.org/pdf/1907.02392v3; 2019.
40. Visual Learning Lab Heidelberg. Freia - framework for easily invertible architectures *Edited by* GitHub. https://github.com/VLL-HD/FrEIA; 2021.
41. Noever-Castelos P, Haller B, Balzani C. Validation of a modelling methodology for wind turbine rotor blades based on a full scale blade test; 2021.
42. Blasques JP. *User's manual for becas: a cross section analysis tool for anisotropic and inhomogeneous beam sections of arbitrary geometry*, Denmark. Forskningscenter Risoe. Risoe-R: Risø DTU – National Laboratory for Sustainable Energy; 2012.
43. Raschka S, Mirjalili V. *Python Machine Learning: Machine Learning and Deep Learning with Python*, *Scikit-learn*, *and Tensorflow 2*. 3. Birmingham: Packt; 2019.
44. Huang J, Li Y-F, Xie M. An empirical analysis of data preprocessing for machine learning-based software cost estimation. *Inf Softw Technol*. 2015;67:108-127.
45. Chen Z, Menzies T, Port D, Boehm B. Feature subset selection can improve software cost estimation accuracy. *ACM SIGSOFT Softw Eng Notes*. 2005;30(4):1-6.
46. Magerramova L, Vasilyev B, Kinzburskiy V. Novel designs of turbine blades for additive manufacturing. In: Proceedings of asme turbo expo 2016: Turbomachinery technical conference and exposition; 2016.
47. Radev ST, Graw F, Chen S, Mutters NT, Eichel VM, Bärnighausen T, Köthe U. Model-based bayesian inference of disease outbreak dynamics with invertible neural networks. http://arxiv.org/pdf/2010.00300v3; 2020.
48. Ardizzone L, Mackowiak R, Rother C, Köthe U. Training normalizing flows with the information bottleneck for competitive generative classification. In: 32nd conference on neural information processing systems (neurips 2018), Vol. 33; 2018:7828-7840.
49. Ksoll VF, Ardizzone L, Klessen R, et al. Stellar parameter determination from photometry using invertible neural networks. *Mon Not R Astron Soc*. 2020;499(4):5447-5485.
50. Bellagente M, Butter A, Kasieczka G, et al. Invertible networks or partons to detector and back again. *SciPost Physics*. 2020;9(5):74.
51. Gröhl J, Schellenberg M, Dreher K, Maier-Hein L. Deep learning for biomedical photoacoustic imaging: A review. *Photoacoustics*. 2021;22:100241.
52. Radev ST, Mertens UK, Voss A, Ardizzone L, Kothe U. Bayesflow: learning complex stochastic models with invertible neural networks. *IEEE transactions on neural networks and learning systems*. 2020;1050:17.
53. Patterson J, Gibson A. *Deep Learning: A Practitioner's Approach*. 1. Beijing and Boston and Farnham and Sebastopol and Tokyo: O'Reilly; 2017.
54. Ravichandiran S. *Hands-on Deep Learning Algorithms with Python: Master Deep Learning Algorithms With Extensive Math by Implementing Them Using Tensorflow*. Birmingham and Mumbai: Packt Publishing; 2019.
55. DNVGL AS. Dnvgl-st-0376 - rotor blades for wind turbines. https://rules.dnvgl.com/docs/pdf/DNVGL/ST/2015-12/DNVGL-ST-0376.pdf; 2015.
56. Hodges DH. *Nonlinear Composite Beam Theory*, Progress in astronautics and aeronautics, vol. 213. Reston, Va.: American Institute of Aeronautics and Astronautics; 2006.
57. Blasques JP, Stolpe M. Multi-material topology optimization of laminated composite beam cross sections. *Compos Struct*. 2012;94(11):3278-3289.
58. Paszke A, Gross S, Massa F, et al. Pytorch: an imperative style, high-performance deep learning library. http://arxiv.org/pdf/1912.01703v1; 2019.
59. Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. *J Mach Learn Res*. 2011;12:2121-2159.

How to cite this article: Noever-Castelos P, Ardizzone L, Balzani C. Model updating of wind turbine blade cross sections with invertible neural networks. *Wind Energy*. 2021;1-27. doi:10.1002/we.2687

APPENDIX A: SENSITIVITY ANALYSIS

TABLE A1 Shell layup of the DemoBlade at cross section R = 6 m

Laminate	No. of plies	Nom. thick.
Triax	1	0.9 mm
Biax 0°/90°	1	0.65 mm
Foam/Balsa	1	20 mm
Biax 0°/90°	1	0.65 mm
Triax	1	0.9 mm

TABLE A2 Prefabricated spar cap layup of the DemoBlade at cross section R = 6 m

Laminate	No. of plies	Nom. thick.
Triax	1	0.9 mm
UD	32	26.2 mm
Triax	1	0.9 mm

TABLE A3 Full sensitivity matrix of cross section at $R = 6$ m — Part A: Cross-sectional properties

	SC_x	SC_y	EC_x	EC_y	$Mass_{total}$	CoG_x	CoG_y	I_{xx}	I_{yy}	I_{xy}	$Area_x$	$Area_y$	A_{xx}	A_{yy}	A_{xy}	$Area_{total}$	$\alpha_{PC,Ref}$	$\alpha_{PC,EC}$
$E_{11,\,UD}$	**0.8**	**0.8**	**4.6**	**4.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**2.8**	**2.9**
$E_{22,\,UD}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\,UD}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\nu_{12,\,UD}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{UD}	0.0	0.0	0.0	0.0	**3.4**	**3.5**	**2.6**	0.0	**4.7**	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$E_{11,\,Biax45}$	0.0	0.0	0.2	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$E_{22,\,Biax45}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\,Biax45}$	**2.0**	**2.3**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\nu_{12,\,Biax45}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Biax45}	0.0	0.0	0.0	0.0	**0.5**	**0.6**	**0.4**	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$E_{11,\,Biax90}$	**0.4**	**0.3**	**1.9**	**1.7**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**1.1**	**1.2**
$E_{22,\,Biax90}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\,Biax90}$	**0.6**	**0.7**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\nu_{12,\,Biax90}$	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Biax90}	0.0	0.0	0.0	0.0	**2.0**	**0.9**	**0.6**	**2.5**	**1.4**	**2.2**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$E_{11,\,Triax}$	**0.3**	**0.5**	**2.7**	**2.4**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**1.7**	**1.7**
$E_{22,\,Triax}$	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\,Triax}$	**1.9**	**2.2**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\nu_{12,\,Triax}$	0.0	0.0	0.2	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.2
ρ_{Triax}	0.0	0.0	0.0	0.0	**3.2**	**0.9**	**0.6**	**3.6**	**2.5**	**3.2**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$E_{11,\,Balsa}$	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\,Balsa}$	**0.7**	**0.8**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\nu_{12,\,Balsa}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Balsa}	0.0	0.0	0.0	0.0	**2.7**	**3.9**	**4.2**	**3.8**	**1.9**	**3.7**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
E_{Foam}	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G_{Foam}	0.2	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Foam}	0.0	0.0	0.0	0.0	**1.3**	**1.7**	**2.4**	**0.5**	**0.9**	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$P_{SS,TE,offset}$	0.2	0.0	0.1	0.0	0.1	**0.3**	**0.3**	**0.4**	0.0	**0.3**	**2.8**	**2.8**	**3.2**	**3.3**	**3.4**	**2.9**	0.0	0.0
$P_{SS,TE,core}$	**1.0**	**3.3**	**0.8**	**2.3**	0.0	0.1	**1.2**	0.0	0.0	**1.7**	**1.1**	**1.9**	**1.9**	**1.2**	**2.0**	**0.5**	**3.2**	**3.5**
$P_{SS,TE,spar\ cap}$	**1.0**	**3.3**	**0.8**	**2.3**	0.0	0.1	**1.2**	0.0	0.0	**1.7**	**1.1**	**1.9**	**1.9**	**1.2**	**2.0**	**0.5**	**3.2**	**3.5**
$P_{SS,Mid,spar\ cap}$	**1.0**	**3.3**	**0.8**	**2.3**	0.0	0.1	**1.2**	0.0	0.0	**1.7**	**1.1**	**1.9**	**1.9**	**1.2**	**2.0**	**0.5**	**3.2**	**3.5**
$P_{SS,LE,spar\ cap}$	**1.0**	**3.3**	**0.8**	**2.3**	0.0	0.1	**1.2**	0.0	0.0	**1.7**	**1.1**	**1.9**	**1.9**	**1.2**	**2.0**	**0.5**	**3.2**	**3.5**
$P_{SS,LE,core}$	**1.0**	**3.3**	**0.8**	**2.3**	0.0	0.1	**1.2**	0.0	0.0	**1.7**	**1.1**	**1.9**	**1.9**	**1.2**	**2.0**	**0.5**	**3.2**	**3.5**

TABLE A3 (Continued)

	SC_x	SC_y	EC_x	EC_y	$Mass_{total}$	CoG_x	CoG_y	I_{xx}	I_{yy}	I_{xy}	$Area_x$	$Area_y$	A_{xx}	A_{yy}	A_{xy}	$Area_{total}$	$\alpha_{PC,Ref}$	$\alpha_{PC,EC}$
$P_{SS,LE,offset}$	0.1	0.0	0.0	0.1	0.1	0.1	0.2	0.1	0.0	0.0	**0.9**	**2.4**	**1.8**	**0.3**	0.2	**2.9**	0.0	0.0
$P_{PS,LE,offset}$	0.0	0.1	0.1	0.0	0.1	**0.3**	0.2	0.1	0.0	0.1	**4.2**	**2.3**	**1.7**	**3.4**	**1.9**	**2.9**	0.0	0.0
$P_{PS,LE,core}$	**4.9**	**3.3**	**1.4**	**2.2**	0.1	**1.3**	**1.2**	0.0	**0.3**	**1.8**	**1.0**	**2.0**	**1.9**	**0.9**	**2.2**	**0.3**	**3.5**	**3.1**
$P_{PS,LE,spar\ cap}$	**4.9**	**3.3**	**1.4**	**2.2**	0.1	**1.3**	**1.2**	0.0	**0.3**	**1.8**	**1.0**	**2.0**	**1.9**	**0.9**	**2.2**	**0.3**	**3.5**	**3.1**
$P_{PS,Mid,spar\ cap}$	**4.9**	**3.3**	**1.4**	**2.2**	0.1	**1.3**	**1.2**	0.0	**0.3**	**1.8**	**1.0**	**2.0**	**1.9**	**0.9**	**2.2**	**0.3**	**3.5**	**3.1**
$P_{PS,TE,spar\ cap}$	**4.9**	**3.3**	**1.4**	**2.2**	0.1	**1.3**	**1.2**	0.0	**0.3**	**1.8**	**1.0**	**2.0**	**1.9**	**0.9**	**2.2**	**0.3**	**3.5**	**3.1**
$P_{PS,TE,core}$	**4.9**	**3.3**	**1.4**	**2.2**	0.1	**1.3**	**1.2**	0.0	**0.3**	**1.8**	**1.0**	**2.0**	**1.9**	**0.9**	**2.2**	**0.3**	**3.5**	**3.1**
$P_{PS,TE,offset}$	**0.3**	0.0	0.0	0.0	0.1	**0.4**	**0.4**	**0.4**	0.0	**0.4**	**2.6**	**2.8**	**3.3**	**3.1**	**3.2**	**3.0**	0.0	0.0

TABLE A4 Full sensitivity matrix of cross section at $R = 6$ m — Part B: Stiffness matrix

	K_{11}	K_{12}	K_{22}	K_{13}	K_{23}	K_{33}	K_{14}	K_{24}	K_{34}	K_{44}	K_{15}	K_{25}	K_{35}	K_{45}	K_{55}	K_{16}	K_{26}	K_{36}	K_{46}	K_{56}	K_{66}
$E_{11,\text{ UD}}$	0.1	0.0	0.2	0.0	0.0	**4.7**	0.0	0.0	0.0	0.0	0.0	0.0	**2.2**	0.2	**5.4**	**0.7**	**0.8**	0.0	0.0	0.0	0.0
$E_{22,\text{ UD}}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\text{ UD}}$	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1
$\nu_{12,\text{ UD}}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{UD}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$E_{11,\text{ Biax45}}$	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0
$E_{22,\text{ Biax45}}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\text{ Biax45}}$	**5.5**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0
$\nu_{12,\text{ Biax45}}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Biax45}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$E_{11,\text{ Biax90}}$	0.0	0.0	0.1	0.0	0.0	**1.7**	0.0	0.0	**2.7**	**3.1**	0.0	0.0	**2.8**	**2.2**	**1.0**	**0.3**	**0.5**	0.0	0.0	0.0	0.0
$E_{22,\text{ Biax90}}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\text{ Biax90}}$	**0.7**	**1.3**	**1.9**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**1.0**	**1.6**	0.0	0.0	0.0	**1.9**
$\nu_{12,\text{ Biax90}}$	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.1	0.1	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Biax90}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$E_{11,\text{ Triax}}$	0.1	0.0	0.1	0.0	0.0	**3.1**	0.0	0.0	**4.4**	**5.0**	0.0	0.0	**4.4**	**3.7**	**2.1**	**0.4**	0.1	0.0	0.0	0.0	0.0
$E_{22,\text{ Triax}}$	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.1	0.1	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\text{ Triax}}$	**2.1**	**3.8**	**5.5**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**3.1**	**4.8**	0.0	0.0	0.0	**5.6**
$\nu_{12,\text{ Triax}}$	0.0	0.0	0.0	0.0	0.0	**0.3**	0.0	0.0	**0.4**	**0.5**	0.0	0.0	**0.4**	**0.3**	0.2	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Triax}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$E_{11,\text{ Balsa}}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\text{ Balsa}}$	0.1	**0.6**	**0.7**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.8**	**0.9**	0.0	0.0	0.0	**0.4**
$\nu_{12,\text{ Balsa}}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Balsa}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
E_{Foam}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0
G_{Foam}	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.3**	0.1	0.0	0.0	0.0	0.1
ρ_{Foam}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$P_{\text{SS,TE,offset}}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.0	0.0	**0.8**	0.0	0.0	0.0	0.0
$P_{\text{SS,TE,core}}$	0.0	**3.2**	**0.3**	0.0	0.0	0.0	0.0	0.0	**2.0**	0.0	0.0	0.0	**0.8**	**2.7**	0.1	**3.5**	**1.4**	0.0	0.0	0.0	0.1
$P_{\text{SS,TE,spar cap}}$	0.0	**3.2**	**0.3**	0.0	0.0	0.0	0.0	0.0	**2.0**	0.0	0.0	0.0	**0.8**	**2.7**	0.1	**3.5**	**1.4**	0.0	0.0	0.0	0.1
$P_{\text{SS,Mid,spar cap}}$	0.0	**3.2**	**0.3**	0.0	0.0	0.0	0.0	0.0	**2.0**	0.0	0.0	0.0	**0.8**	**2.7**	0.1	**3.5**	**1.4**	0.0	0.0	0.0	0.1
$P_{\text{SS,LE,spar cap}}$	0.0	**3.2**	**0.3**	0.0	0.0	0.0	0.0	0.0	**2.0**	0.0	0.0	0.0	**0.8**	**2.7**	0.1	**3.5**	**1.4**	0.0	0.0	0.0	0.1
$P_{\text{SS,LE,core}}$	0.0	**3.2**	**0.3**	0.0	0.0	0.0	0.0	0.0	**2.0**	0.0	0.0	0.0	**0.8**	**2.7**	0.1	**3.5**	**1.4**	0.0	0.0	0.0	0.1

TABLE A4 (Continued)

	K_{11}	K_{12}	K_{22}	K_{13}	K_{23}	K_{33}	K_{14}	K_{24}	K_{34}	K_{44}	K_{15}	K_{25}	K_{35}	K_{45}	K_{55}	K_{16}	K_{26}	K_{36}	K_{46}	K_{56}	K_{66}
$P_{\mathrm{SS,LE,offset}}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.3**	0.0	0.0	0.0	0.0
$P_{\mathrm{PS,LE,offset}}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.0
$P_{\mathrm{PS,LE,core}}$	0.1	**2.8**	0.2	0.0	0.0	0.0	0.0	0.0	**2.0**	0.0	0.0	0.0	**1.4**	**2.9**	0.2	**3.3**	**1.4**	0.0	0.0	0.0	0.1
$P_{\mathrm{PS,LE,spar\ cap}}$	0.1	**2.8**	0.2	0.0	0.0	0.0	0.0	0.0	**2.0**	0.0	0.0	0.0	**1.4**	**2.9**	0.2	**3.3**	**1.4**	0.0	0.0	0.0	0.1
$P_{\mathrm{PS,Mid,spar\ cap}}$	0.1	**2.8**	0.2	0.0	0.0	0.0	0.0	0.0	**2.0**	0.0	0.0	0.0	**1.4**	**2.9**	0.2	**3.3**	**1.4**	0.0	0.0	0.0	0.1
$P_{\mathrm{PS,TE,spar\ cap}}$	0.1	**2.8**	0.2	0.0	0.0	0.0	0.0	0.0	**2.0**	0.0	0.0	0.0	**1.4**	**2.9**	0.2	**3.3**	**1.4**	0.0	0.0	0.0	0.1
$P_{\mathrm{PS,TE,core}}$	0.1	**2.8**	0.2	0.0	0.0	0.0	0.0	0.0	**2.0**	0.0	0.0	0.0	**1.4**	**2.9**	0.2	**3.3**	**1.4**	0.0	0.0	0.0	0.1
$P_{\mathrm{PS,TE,offset}}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**1.2**	0.0	0.0	0.0	0.0

TABLE A5 Full sensitivity matrix of cross section at $R = 6$ m — Part C: Mass matrix

	M_{11}	M_{12}	M_{22}	M_{13}	M_{23}	M_{33}	M_{14}	M_{24}	M_{34}	M_{44}	M_{15}	M_{25}	M_{35}	M_{45}	M_{55}	M_{16}	M_{26}	M_{36}	M_{46}	M_{56}	M_{66}
$E_{11,\ UD}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$E_{22,\ UD}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\ UD}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\nu_{12,\ UD}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{UD}	**3.4**	0.0	**3.4**	0.0	0.0	**3.4**	0.0	0.0	0.0	0.0	0.0	0.0	**0.9**	0.1	**4.7**	0.0	**0.9**	0.0	0.0	0.0	**0.9**
$E_{11,\ Biax45}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$E_{22,\ Biax45}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\ Biax45}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\nu_{12,\ Biax45}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Biax45}	**0.5**	0.0	**0.5**	0.0	0.0	**0.5**	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.2	0.0	0.1	0.0	0.0	0.0	0.0
$E_{11,\ Biax90}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$E_{22,\ Biax90}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\ Biax90}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\nu_{12,\ Biax90}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Biax90}	**2.0**	0.0	**2.0**	0.0	0.0	**2.0**	0.0	0.0	**1.6**	**2.5**	0.0	0.0	**1.9**	**2.2**	**1.4**	**1.6**	**1.9**	0.0	0.0	0.0	**2.5**
$E_{11,\ Triax}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$E_{22,\ Triax}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\ Triax}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\nu_{12,\ Triax}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Triax}	**3.2**	0.0	**3.2**	0.0	0.0	**3.2**	0.0	0.0	**2.3**	**3.6**	0.0	0.0	**2.6**	**3.2**	**2.5**	**2.3**	**2.6**	0.0	0.0	0.0	**3.7**
$E_{11,\ Balsa}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$G_{12,\ Balsa}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\nu_{12,\ Balsa}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Balsa}	**2.7**	0.0	**2.7**	0.0	0.0	**2.7**	0.0	0.0	**4.8**	**3.8**	0.0	0.0	**4.7**	**3.7**	**1.9**	**4.8**	**4.7**	0.0	0.0	0.0	**3.7**
E_{Foam}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G_{Foam}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ρ_{Foam}	**1.3**	0.0	**1.3**	0.0	0.0	**1.3**	0.0	0.0	**1.1**	**0.5**	0.0	0.0	**0.6**	0.2	**0.9**	**1.1**	**0.6**	0.0	0.0	0.0	**0.6**
$P_{SS,TE,offset}$	0.1	0.0	0.1	0.0	0.0	0.1	0.0	0.0	**0.3**	**0.4**	0.0	0.0	**0.3**	**0.3**	0.0	**0.3**	**0.3**	0.0	0.0	0.0	**0.3**
$P_{SS,TE,core}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.9**	0.0	0.0	0.0	0.0	**1.7**	0.0	**0.9**	0.0	0.0	0.0	0.0	0.0
$P_{SS,TE,spar\ cap}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.9**	0.0	0.0	0.0	0.0	**1.7**	0.0	**0.9**	0.0	0.0	0.0	0.0	0.0
$P_{SS,Mid,spar\ cap}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.9**	0.0	0.0	0.0	0.0	**1.7**	0.0	**0.9**	0.0	0.0	0.0	0.0	0.0
$P_{SS,LE,spar\ cap}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.9**	0.0	0.0	0.0	0.0	**1.7**	0.0	**0.9**	0.0	0.0	0.0	0.0	0.0
$P_{SS,LE,core}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.9**	0.0	0.0	0.0	0.0	**1.7**	0.0	**0.9**	0.0	0.0	0.0	0.0	0.0

TABLE A5 (Continued)

	M_{11}	M_{12}	M_{22}	M_{13}	M_{23}	M_{33}	M_{14}	M_{24}	M_{34}	M_{44}	M_{15}	M_{25}	M_{35}	M_{45}	M_{55}	M_{16}	M_{26}	M_{36}	M_{46}	M_{56}	M_{66}
$P_{SS,LE,offset}$	0.1	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.1
$P_{PS,LE,offset}$	0.1	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.1	0.1	0.0	0.0	0.2	0.1	0.0	0.1	0.2	0.0	0.0	0.0	0.1
$P_{PS,LE,core}$	0.1	0.0	0.1	0.0	0.0	0.1	0.0	0.0	**0.9**	0.0	0.0	0.0	**1.0**	**1.8**	**0.3**	**0.9**	**1.0**	0.0	0.0	0.0	0.1
$P_{PS,LE,spar\ cap}$	0.1	0.0	0.1	0.0	0.0	0.1	0.0	0.0	**0.9**	0.0	0.0	0.0	**1.0**	**1.8**	**0.3**	**0.9**	**1.0**	0.0	0.0	0.0	0.1
$P_{PS,Mid,spar\ cap}$	0.1	0.0	0.1	0.0	0.0	0.1	0.0	0.0	**0.9**	0.0	0.0	0.0	**1.0**	**1.8**	**0.3**	**0.9**	**1.0**	0.0	0.0	0.0	0.1
$P_{PS,TE,spar\ cap}$	0.1	0.0	0.1	0.0	0.0	0.1	0.0	0.0	**0.9**	0.0	0.0	0.0	**1.0**	**1.8**	**0.3**	**0.9**	**1.0**	0.0	0.0	0.0	0.1
$P_{PS,TE,core}$	0.1	0.0	0.1	0.0	0.0	0.1	0.0	0.0	**0.9**	0.0	0.0	0.0	**1.0**	**1.8**	**0.3**	**0.9**	**1.0**	0.0	0.0	0.0	0.1
$P_{PS,TE,offset}$	0.1	0.0	0.1	0.0	0.0	0.1	0.0	0.0	**0.3**	**0.4**	0.0	0.0	**0.3**	**0.4**	0.0	**0.3**	**0.3**	0.0	0.0	0.0	**0.3**

APPENDIX B: EVALUATION OF CINN

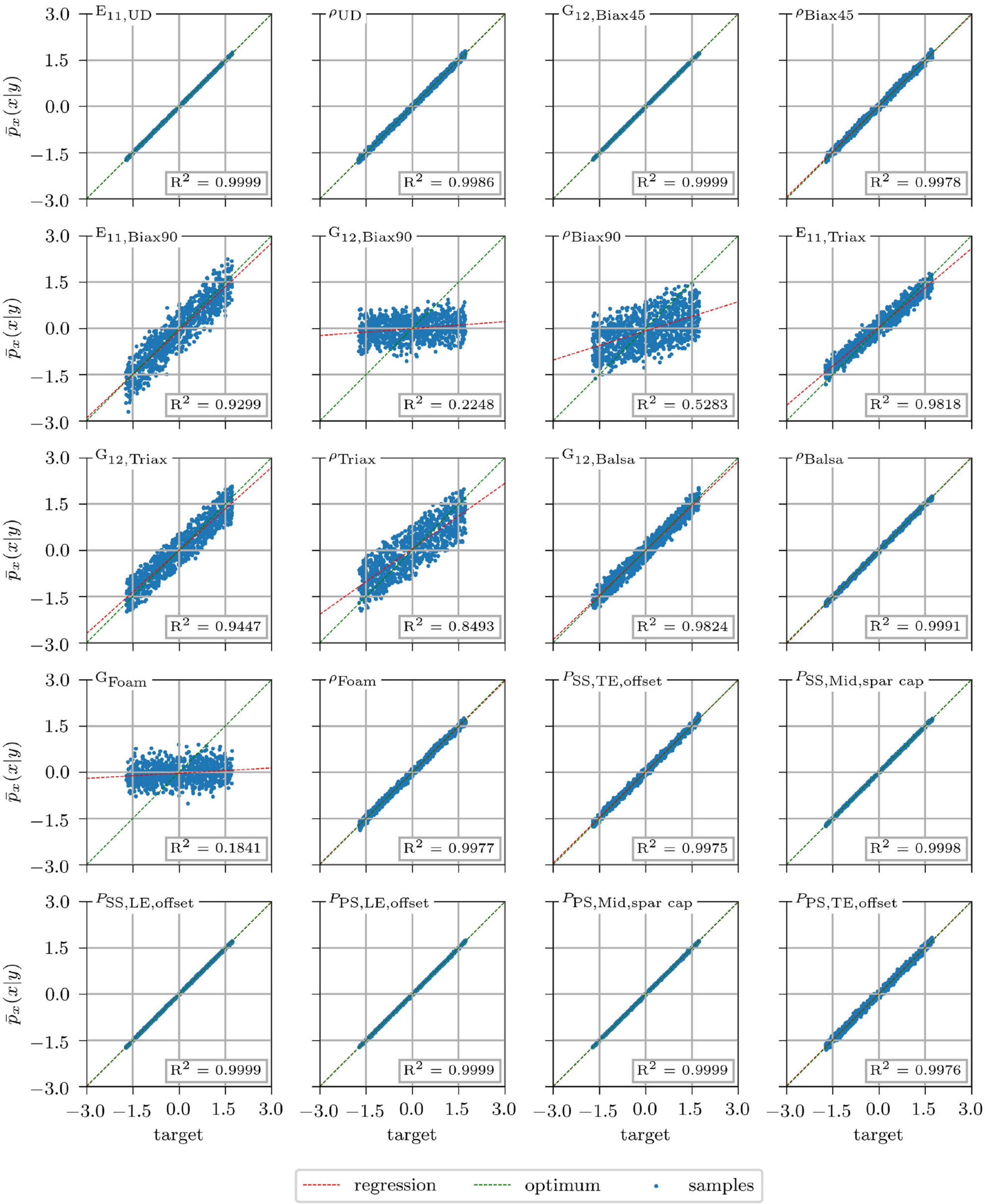

FIGURE B1 Correlation between input feature prediction and ground truth measured with R^2. Optimum values will are located on $f(x) = m \cdot x$

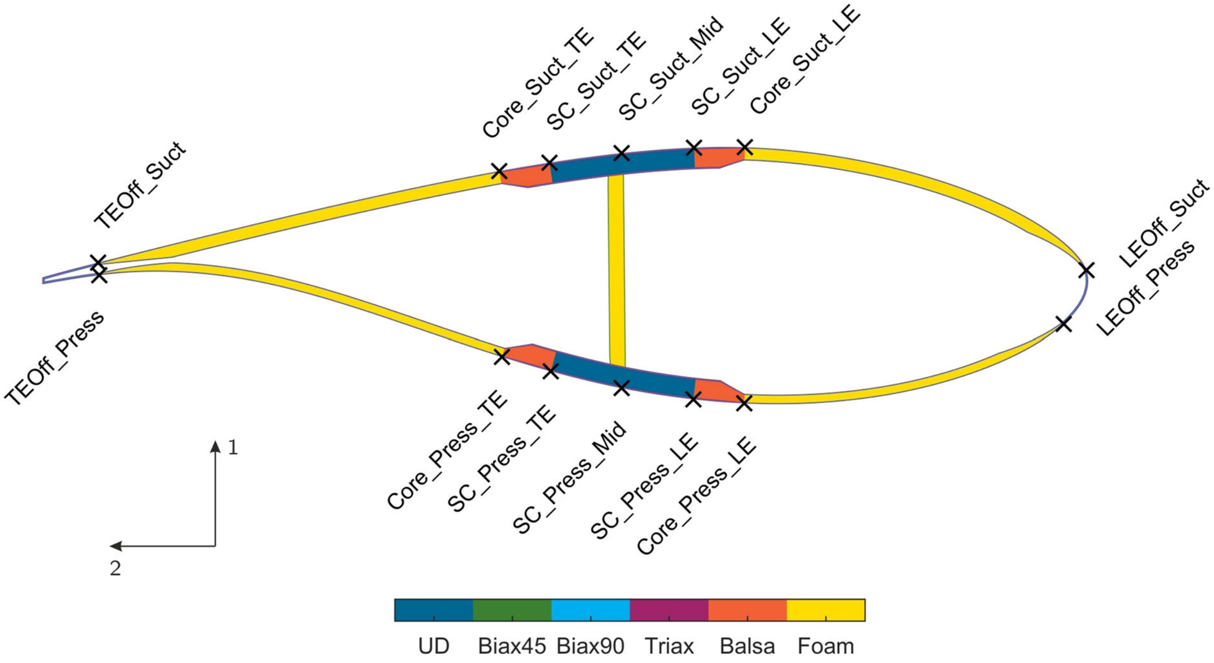

FIGURE B2 Cross section of the SmartBlades2 DemoBlade at a radial position of $R = 12$ m