

Updating structural wind turbine blade models via invertible neural networks



Supervisors

Prof. Dr.-Ing. Andreas Reuter
Prof. Dr. Ullrich Köthe

Pablo Noever Castelos

01 (2023)

Contact:

Leibniz University Hannover
Institute for Wind Energy Systems
Appelstraße 9A
30167 Hannover
Germany

E-Mail: research@iwes.uni-hannover.de

URL: www.iwes.uni-hannover.de

Cover image: Sander Weeteling, Unsplash, "Early morning windmill sunrise in the mist", 2021

Bibliographic information of the German National Library:

The German National Library has listed this publication in its Deutsche Nationalbibliographie; detailed bibliographic data is available on the internet at: www.dnb.de

Publication series: Dissertations of the Institute for Wind Energy Systems

ISSN 2940-2689 (Print)

ISSN 2940-2697 (Online)

Online version accessible via:



Updating structural wind turbine blade models via invertible neural networks © 2022
by Pablo Noever Castelos is licensed under CC BY 4.0.

To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>

Thesis design and implementation by Michael Kuhn and Pablo Noever Castelos.

Supported by:



Federal Ministry
for Economic Affairs
and Climate Action

on the basis of a decision
by the German Bundestag

This work was partially funded by the Federal Ministry for Economic Affairs and Climate Action (BMWK) in the frames of the projects *SmartBlades2* (funding reference no.: 0324032C) and *ReliaBlade* (funding reference no.: 0324335B).

Updating structural wind turbine blade models via invertible neural networks

Von der Fakultät für Bauingenieurwesen und Geodäsie der Gottfried Wilhelm Leibniz Universität Hannover zur Erlangung des Grades Doktor-Ingenieur - Dr.-Ing. - genehmigte Dissertation

Von

Master of Science

Pablo Noever Castelos

2023



Hauptreferent

Prof. Dr.-Ing.

Andreas Reuter

Leibniz Universität Hannover
Institut für Windenergiesysteme

Korreferent

Prof. Dr.

Ullrich Köthe

Universität Heidelberg
Interdisziplinäres Zentrum für wissens. Rechnen

Tag der mündlichen Prüfung:
25. November 2022

A mi querido padre

Vorwort

Mit dieser Arbeit schließe ich meine Promotion ab, die ich im Rahmen meiner Forschung als wissenschaftlicher Mitarbeiter am Institut für Windenergiesysteme im Zeitraum von 2016 bis 2022 durchgeführt habe. Was zunächst mit dem Forschungsschwerpunkt auf Modellierung und Analyse von Rotorblattverklebung begonnen hatte, fokussierte sich im Laufe der Jahre auf die Modellierung gesamter Rotorblätter und das Modelupdating dieser. Speziell im letzten Drittel der Arbeit ist, mit der Anwendung von Machine Learning, ein in sich abgeschlossener Forschungskomplex geworden. Diesen habe ich in mehreren Publikationen veröffentlicht und als kumulative Dissertation eingereicht. An dieser Stelle möchte ich nun die Gelegenheit nutzen und mich bei den Personen bedanken, die mich auf diesem Wege begleitet haben.

Meinem Doktorvater, Herrn Professor Andreas Reuter, gilt in mehrfacher Hinsicht mein ausdrücklicher Dank. Zunächst für alle Möglichkeiten und Freiheiten, die er mir eröffnet hat, um mich wissenschaftlich entfalten zu können. Jedoch auch für die regelmäßige thematische Betreuung, (Neu-)Orientierung, Eingrenzung und Fokussierung auf mein Promotionsvorhaben. Ebenso möchte ich meinen Dank an Professor Ullrich Köthe ausdrücken, der sich des Korreferats angenommen und damit den Expertenpart sowie die Betreuung hinsichtlich der invertierbaren neuronalen Netze übernommen hat. Auch den Herren Professoren Raimund Rolfes und Nabil A. Fouad möchte ich für die Ausübung des Prüfungsbeisitzes und -vorsitzes danken. Ferner möchte ich auch meinen Koautoren, Bernd Haller, Lynton Ardizzone und David Melcher danken, die mich bei der Erstellung meiner promotionsrelevanten Publikationen fachlich unterstützt haben.

Für die kollegiale und produktive Arbeitsatmosphäre im Institut möchte ich mich bei allen Kollegen, die meine Zeit dort geteilt haben, bedanken. Besonders hervorheben möchte ich hierbei Claudio Balzani, der als Gruppenleitung einen beachtlichen Teil meiner Betreuung übernommen hat. Durch seine erfolgreiche Projektakquise, war immer für ausreichend Finanzierung gesorgt, sodass ich mich auf meine Forschung konzentrieren konnte. Er hat mich immer in meinen Entscheidungen gestärkt und unterstützt. Außerdem möchte ich mich bei meinen langjährigen Bürokollegen, Michael Kuhn, bedanken, mit dem ich viele fachliche sowie private Diskussionen genießen durfte. In dieser Zeit war er stets meine erste Anlaufstelle für fachliche und auch stilistische Fragestellungen. Mein Dank gilt außerdem Philipp Dauer, der mir mit seinem Enthusiasmus für künstliche Intelligenz und maschinelles Lernen den finalen Anstoß für den Abschluss meiner Promotion gegeben hat.

Nicht zuletzt und in einem großen Maße möchte ich mich bei meiner Familie und meinen Freunden bedanken, die immer für mich da waren, wenn ich sie gebraucht habe. Dabei gilt ein außerordentlicher Dank natürlich meiner Ehefrau Lena, die mich mit viel Liebe und Geduld bei meiner Arbeit begleitet und mir stets den Rücken frei gehalten hat. Und natürlich auch meinen Kindern Paula und Bruno, die mir viel notwendige Heiterkeit in der Zeit geschenkt haben.

Abschließend möchte ich meiner Mutter und meinem Vater meinen tiefsten Dank aussprechen, die in guten sowie in schlechten Zeiten immer an mich geglaubt haben und mir die Möglichkeit gaben, nun hier an dieser Stelle des Lebens zu stehen.

Keywords

Model updating,
wind turbine,
rotor blade,
invertible neural
network,
reliability,
digital twin

Abstract

Wind energy plays a major role in the transition to a renewable energy supply and is already holding significant shares of the overall energy production. In the near future, a dependence on this power resource requires a high availability of the turbines. In consequence, an enhanced reliability of all wind turbine components gains significant importance. Especially, the rotor blades are huge and complex composite structures that are exposed to exceptionally high loads, both extreme and fatigue loads. These can result in damages causing severe downtimes or repair costs. It is thus of utmost importance that the blades are carefully designed, including uncertainty analyses in order to produce safe, reliable, and cost-efficient wind turbines.

An accurate reliability assessment should already start during the design and manufacturing phases. Recent developments in digitalization give rise to the concept of a digital twin, which replicates a product and its properties into a digital environment. Model updating is a technique, which helps to adapt the digital twin according to the measured behavior and characteristics of the real structure. Current model updating techniques are most often based on heuristic optimization algorithms, which are computationally expensive, can only deal with a relatively small parameter space, or do not estimate the uncertainty of the computed results. However, recent developments in the field of inverse problems and a solution using invertible neural networks offer a chance to efficiently couple the model updating procedure for complex wind turbine blades with uncertainty analysis.

The specific objective of this thesis is to present a computationally efficient model updating method that recovers parameter deviation. This method is able to consider uncertainties and a high fidelity degree of the rotor blade model. The basis for this study is a fully parameterized model generator, which is validated at the beginning of this thesis. The model generator is used to perform a physics-informed training of a conditional invertible neural network. This network finally represents a surrogate of the physical model, which then can be used to recover model parameters based on structural responses of the blade.

The presented research consists of different studies with increasing levels of fidelity of the physical model used for the training of the invertible neural network. All generic model updating applications show excellent results, predicting the *a posteriori* distribution of the significant model parameters accurately. However, the application of the presented approach on real experimental data results in less accurate predictions and requires further analysis in future research.

Kurzfassung

Die Windenergie spielt eine wichtige Rolle beim Wandel zu einer erneuerbaren Energieversorgung und hat bereits einen erheblichen Anteil an der Gesamtenergieerzeugung. Die künftig zu erwartende Abhängigkeit von dieser Energiequelle erfordert eine hohe Verfügbarkeit der Windenergieanlagen. Hierfür muss die Zuverlässigkeit aller Anlagenkomponenten verbessert werden. Insbesondere Rotorblätter sind gewaltige und komplexe Faserverbundstrukturen, die außergewöhnlichen Extrem- als auch Ermüdungslasten ausgesetzt sind. Diese können zu Schäden führen, die erhebliche Ausfallzeiten oder Reparaturkosten verursachen. Daher ist ein sorgfältiger Entwurf unter Berücksichtigung von Unsicherheiten von größter Bedeutung, um sichere, zuverlässige und kosteneffiziente Windenergieanlagen herzustellen.

Eine genaue Zuverlässigkeitsbewertung sollte bereits bei der Konstruktion und Fertigung beginnen. Im Rahmen der Digitalisierung erhält der digitale Zwilling Einzug, der ein Produkt und seine Eigenschaften in einer digitalen Umgebung abbildet. In diesem Fall ist die Modellaktualisierung eine Technik zur Kalibrierung des digitalen Zwillings entsprechend dem gemessenen Verhalten und den Eigenschaften der realen Struktur. Aktuelle Modellaktualisierungstechniken basieren meist auf heuristischen Optimierungsalgorithmen, die rechenintensiv sind, nur einen kleinen Parameterraum abdecken können oder die Unsicherheit der berechneten Ergebnisse nicht abschätzen. Jüngste Entwicklungen auf dem Gebiet inverser Probleme und deren Lösung mit invertierbaren neuronalen Netzen bieten jedoch die Möglichkeit, die Modellaktualisierung für komplexe Rotorblätter mit gleichzeitiger Unsicherheitsbetrachtung effizient anzugehen.

Das Ziel dieser Arbeit ist die Vorstellung einer recheneffizienten Methode der Modellaktualisierung zur Ermittlung der Parameterabweichung. Diese soll außerdem eine Unsicherheitsbewertung der berechneten Ergebnisse für ein hochaufgelöstes Rotorblattmodell beinhalten. Die Grundlage für die Untersuchungen bildet ein umfänglich parametrisierter Modellgenerator, der zu Beginn dieser Arbeit validiert wird. Der Modellgenerator wird verwendet, um ein physikalisch informiertes Training eines konditionierten invertierbaren neuronalen Netzes durchzuführen. Dieses Netz stellt schließlich ein Ersatzmodell des invertierten physikalischen Modells dar, das dann zur Aktualisierung von Modellparametern auf der Grundlage des strukturellen Verhaltens des Blattes verwendet werden kann.

In der vorliegenden Arbeit werden verschiedene Studien mit zunehmender Genauigkeit des physikalischen Modells dargestellt, das für das Training des invertierbaren neuronalen Netzes verwendet wird. Alle generischen Anwendungen zur Modellaktualisierung liefern ausgezeichnete Ergebnisse, indem sie die *a-posteriori* Wahrscheinlichkeiten der signifikanten Modellparameter genau vorhersagen. Die Anwendung auf experimentelle Daten führt jedoch zu ungenaueren Vorhersagen und erfordert weitere Untersuchungen.

Contents

1	Introduction	1
1.1	Motivation	2
1.2	State of the art	3
1.2.1	Wind turbine blade modeling	3
1.2.2	Model updating approaches	7
1.3	Research gap and objectives	12
1.3.1	Research gap	12
1.3.2	Objectives	13
1.4	Thesis concept	13
1.4.1	Outline	14
1.4.2	Declaration on the included publications	15
2	1st Paper: Validation of a modeling methodology for wind turbine rotor blades based on a full scale blade test	17
P1-1	Introduction	18
P1-1.1	State-of-the-art 3D finite element modeling of wind turbine blades	19
P1-1.2	Objectives of this paper	19
P1-1.3	Outline	20
P1-2	Model creation framework	20
P1-3	Modeling of the test blade	22
P1-4	Test description and virtual modeling	24
P1-4.1	Mass and center of gravity	24
P1-4.2	Modal analysis	24
P1-4.3	Static bending and torsion test configuration	25
P1-4.4	Blade segment mass and center of gravity measurement	27
P1-5	Comparison of experimental and simulation results	28
P1-5.1	Blade mass, center of gravity, and eigenfrequency	28
P1-5.2	Static bending tests	29
P1-5.3	Static torsion tests	31
P1-5.4	Local strain comparison	32
P1-5.5	Segment mass and CoG comparison	34
P1-6	Summary and conclusion	34
P1-7	Appendix A: Static bending test results	36
P1-8	Appendix B: Local strain results	37

3	2nd Paper: Model updating of wind turbine blade cross sections with invertible neural networks	41
	P2-1 Introduction	42
	P2-2 Model updating methodology with INNs	44
	P2-2.1 Sensitivity analysis of blade CS properties.	44
	P2-2.2 INNs for inverse problems	45
	P2-2.3 Architecture and training of the conditional INN	46
	P2-3 Input and output feature selection	48
	P2-4 INN structure, training, and evaluation	52
	P2-4.1 Identifying network hyperparameters.	52
	P2-4.2 Invertible network evaluation	56
	P2-5 Conclusion.	56
	P2-6 Appendix A: Sensitivity analysis	60
	P2-7 Appendix B: Evaluation of cINN	61
4	3rd Paper: Model updating of a wind turbine blade finite element Timoshenko beam model with invertible neural networks	69
	P3-1 Introduction	70
	P3-1.1 Model updating of wind turbine blades	70
	P3-1.2 Drawbacks of current updating approaches	71
	P3-1.3 Model updating via invertible neural networks	71
	P3-1.4 Outline	72
	P3-2 Sensitivity analysis of modal response	72
	P3-2.1 Sobol' sensitivity method	72
	P3-2.2 Rotor blade finite element beam model	73
	P3-2.3 Feature subspace selection with Sobol indices	74
	P3-3 Invertible neural network architecture	76
	P3-4 Model updating of a rotor blade beam model	78
	P3-4.1 General analysis of the updating results.	78
	P3-4.2 Intrinsic model ambiguities.	79
	P3-4.3 Model robustness	83
	P3-4.4 Resimulation analysis	84
	P3-4.5 Replacing sensitivity analysis	85
	P3-5 Conclusion.	87
	P3-6 Appendix A: Tables & Figures	88
5	Model updating benchmark of invertible neural networks versus optimization algorithms	93
	5.1 Benchmarking basis	94
	5.2 Surrogate model	95
	5.3 Training and checking the cINN	96
	5.4 Optimization algorithms for the benchmark.	96
	5.5 Benchmarking with surrogate model.	97
	5.6 Benchmarking adapted to physical model	103
	5.7 Discussion	104

6	Model updating of a 3D finite element wind turbine blade model with invertible neural networks	105
6.1	Setting up the model updating problem	106
6.2	Invertible neural network configuration	108
6.3	Generic model updating of a 3D-FEM blade model.	108
6.3.1	General analysis of the updating results.	108
6.3.2	Resimulation analysis	111
6.4	Model updating based on modal experimental data.	113
6.4.1	Updating results	113
6.4.2	Root cause analysis of the updating inaccuracies	113
7	Conclusion and Outlook	119
7.1	Conclusion.	120
7.2	Outlook.	122
	List of Figures	xiii
	List of Tables	xv
	List of Abbreviations	xvii
	Nomenclature	xix
	Bibliography	xxi

1

Introduction

This chapter introduces the thesis by establishing an overall motivation for the topic. In a second step, the state of the art of wind turbine blade modeling is presented, followed by current model updating techniques. Based on the current state of the art, a research gap is identified and the objectives for this thesis are declared. Finally, the concept of the thesis with the outline and the declaration of contributions to the incorporated publications is presented.

1.1	Motivation	2
1.2	State of the art	3
1.3	Research gap and objectives.....	12
1.4	Thesis concept	13

Wind energy is one of the key drivers to transform the fossil energy production [105] to a renewable energy fundament. The decarbonisation of the energy production is a major prerequisite for a long-term transition to climate-neutrality by mid of this century, as decided in the Paris Agreement on 12 December 2015 [193] and integrated into the European Strategic Energy Technology Plan [76]. A shift to an energy production system based on renewable resources — attributing important significance to wind energy — helps to ensure a general access to affordable, reliable, sustainable and modern energy for all, which is demanded by the United Nations in the 7th goal of their 2030 Agenda for Sustainable Development [194].

Within these frameworks, several platforms and associations from the wind energy sector — such as the European Energy Research Association [62], the European Technology and Innovation Platform on Wind Energy [64, 65], the European Academy of Wind Energy [195], and the International Energy Agency [104] — identified several short to long term research challenges for the wind energy research community. One key focus is on modern and reliable turbine technology. These experts emphasize, among others, the importance of reducing or tracking uncertainties during design and manufacturing as well as the reliability modeling and digitalization.

The present thesis will be settled in this research field. Structural uncertainties within a manufactured rotor blade are hard to quantify due to the structure's complexity. However, a probabilistic and efficient model updating approach can improve the quantification of these uncertainties, reveal extreme manufacturing deviations and enhance a digital twin of the blade. This research gap is addressed by the present thesis.

This first chapter is introduced by a short motivation, which is followed by the state of the art of wind turbine blade modeling and model updating. Subsequently, the research gap and the objectives are described. Finally, the last section contains the conceptual design of this thesis.

1.1 Motivation

To make wind turbines fully competitive and establish wind energy as a stable, renewable form of power generation, the wind turbine reliability is of utmost importance [199]. Especially, the rotor blades are huge and complex composite structures, which carry extreme loading. Thus, an accurate blade design is necessary to minimize manufacturing and material costs, while maximizing power extraction. However, rotor blades are also among the top three most crucial components with respect to severe downtime after a failure [114, 153, 162, 189]. According to a study of Dao et al. [52], the rotor blades and hub contribute to 24.2% of the onshore and 33.3% of the offshore downtime. Hence, the rotor blades reliability has to be maximized [83]. The wind industry is highly focused on condition monitoring of crucial sub-components such as rotor blades or gearboxes [68] to improve the predictive maintenance and thereby decrease downtime significantly [74, 102, 200, 217].

Reliability analysis should already start beforehand at the design, modeling and manufacturing stage. All designed rotor blade prototypes are typically tested in a full scale test against fatigue and extreme loads to experimentally verify the blade design [41]. Due to the increasing size of modern blades [145], recent research proposes segmented blade tests [89] and subcomponent-tests [9, 14, 37, 46, 166, 214] of the rotor blades. These reduce the test dimensions, time, and costs, while trying to achieving similar boundary conditions and loads compared to the full scale tests. In any of the mentioned tests, the prototyping and experimental costs are high, thus, such tests are only performed once at the final certification process [106]. This requires the final design to rely on precise models, that represent the real blade characteristic as accurately as possible, to avoid any unexpected surprise during the tests. Thus, the finally designed rotor blade is expected to be sufficiently accurate and reliable before even performing experimental tests [45].

Although quality management in manufacturing should ensure similar blades during production, which range within a defined tolerance, a lot of uncertainties originate either from the manufacturing process itself or from the uncertain material properties and can provoke non-negligible deviations

in structural blade characteristics [33]. For example, Gundlach and Govers [88] revealed variations in modal response of different blades of the same rotor, i.e., production series. On the other hand, Willberg et al. [206] point on manufacturing deviations, as the blade's structural behavior during experiments does not agree with the predicted results from finite element analysis. However, these deviations in a rotor blade can appear as material property variations, geometric differences or even manufacturing defects [139]. All deviations from the design require a method to capture and track these differences and replicate the correct blade with adequate uncertainties to provide a reliable basis for any further analysis or condition monitoring during operation. This can be accomplished by the novel concept of a digital twin [47, 171].

Digitalization and digital twins receive increasing attention throughout the industries [82]. According to Solman et al. [180], the digital twinning also emerged most recently in the wind energy sector and is gaining significant attention in its research community [66, 124, 140, 159, 171]. A mayor method to derive a digital twin is via model updating of the physical model with measured information of the component, as shown in diverse finite element applications, e.g., for a naval vessel [197], bridges [77] or an offshore structure [192]. To achieve a meaningful digital twin the process has to compromise three principal aspects:

- An extensive and accurate (physical) model
- A sufficient tracking of the manufacturing process and condition state of the product via sensors
- An efficient, reliable and probabilistic model updating procedure to feedback the gathered physical information into the model

However, till now, the combination of an extensive model and an efficient model updating procedure is contradictory. Current model updating procedures are based on optimization methods. In general, these optimization methods require the number of parameters to be as small as possible, since fewer parameters improve the stability of the optimization solution [84]. This present thesis will focus on combining both extensive rotor blade models and an efficient updating method.

1.2 State of the art

This section gives an overview on the state of the art of both wind turbine blade modeling and model updating approaches. First, modeling techniques for wind turbine blades are reviewed and categorized according to their fidelity level. Next, different model updating approaches are assessed. Some examples will focus on wind energy applications, while others are more generally chosen to show the broad possibilities of model updating methods.

1.2.1 Wind turbine blade modeling

As already emphasized, a correct and accurate modeling is an essential first step for the reliability of wind turbine blades. Although the lifetime and operational reliability is one important aspect, better predictive models could also lead to a reduction of safety factors imposed by certification guidelines such as DNV GL AS [58] and thereby push the blade designs more to the limit, which would reduce material weight as well as costs and consequently also beneficially impact the design of other turbine components. Therefore, wind energy research and industry have established a huge range of tools to accurately design blades and predict their characteristics from aeroelastic codes [75, 110, 123] to optimization toolboxes [32, 34]. However, accurate models are computationally expensive and do not necessarily contribute additional information for particular design stages. Thus, the different applications call for different model fidelity levels.

A pre-stage in modeling wind turbine blades are cross sectional analysis tools used to generate cross sectional properties of anisotropic beam structures. This directly leads to the first fidelity level of a full blade model, the Euler-Bernoulli beam [86] or the more sophisticated Timoshenko beam model [182], which can both be built based on the cross sectional properties generated beforehand. The next higher fidelity level are fully resolved finite-element models represented by shell elements, which are a good compromise between accuracy and computational cost. The best representation of wind turbine blades can be achieved by 3D finite element models with continuum shell elements or layered solid elements. These can replicate the exact geometry, laminate layup and tapering. However, they require the most computational effort. In the following, the state of the art for these four fidelity levels will be discussed. Further information can also be found in Chapter 2 (i.e., the first paper).

Cross sectional model

Although cross sectional analysis codes do not directly calculate blade responses, in conjunction with beam models, these were very early stages of analyzing the wind turbine blade behavior, as for example in the computer program HANBA2 [79]. Other early codes [25, 26, 158] are based on classical laminate theory [150] or derivations of it. Since then more algorithms were published [13, 131, 204] in research, while the most popular codes, at least in the wind energy research community, are the Beam Cross Sectional Analysis Software - BECAS [27, 28, 29] (exemplarily depicted in Figure 1.1) and the Variational Asymptotic Beam Sectional Analysis - VABS [44]. While all of the aforementioned codes return individual structural properties, such as the shear or mass center, more sophisticated approaches such as BECAS and VABS compute fully populated 6x6 mass and stiffness matrices.

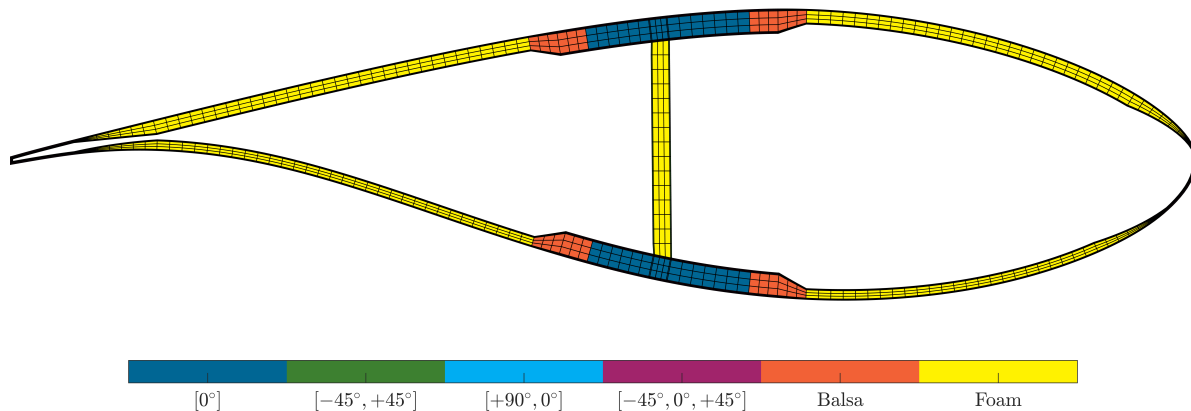


Figure 1.1: Exemplary plot of a wind turbine blade cross section generated with BECAS, showing the element distribution and material assignment.

BECAS, as well as the tool NABSA [212], was developed based on the formulations published by Giavotto et al. [79]. These formulations were established to account for anisotropy and inhomogeneity of beam structures by extending the Saint Venant hypotheses for homogeneous and isotropic beams. VABS is also capable of dealing with anisotropic, non-homogeneous materials and to represent general cross sectional geometries, but was formulated according to Hodges [99], who included the theories of the geometrically exact beam [97] and the variational asymptotic methods [23]. The software VABS was continuously tested, revised and validated [98, 210, 211, 212]. Chen et al. [45] compared the accuracy of VABS against different other tools such as PreComp [25], FAROB [158], and CROSTAB [130] and BPE [131], with the result that VABS outperforms the others in terms of accuracy.

Finite element beam model

Wind turbine blades being slender and elongated structures are predestined to be modeled as beam-like components [209]. Typically, these are chosen as either Euler-Bernoulli beams [95] or Timoshenko beams [182]. The beam theory according to Euler-Bernoulli assumes that the cross sectional deformation is limited only to rotation due to beam bending, i.e., the cross sections stay perpendicular to the beam axis in the deformed configuration. Shear deformations from lateral forces are additionally taken into account in the Timoshenko beam theory. Both models are linear beam models that assume small deflections. However, modern wind turbine blades are highly flexible and are exposed to partially very large deflection, which makes that assumption invalid. Formulations such as the geometrically exact beam theory [98] also account for large deflections. Compared to 3D finite element models, beam models are computationally more efficient, while providing sufficiently accurate solutions considering the general turbine dynamics [53, 201]. The relevant cross sectional information to create a beam model can be derived from the aforementioned cross sectional analysis tools. Figure 1.2 illustrates an exemplary finite element beam model, where BECAS was used to calculate the cross sectional properties for the section, as shown for the cross section highlighted in red.

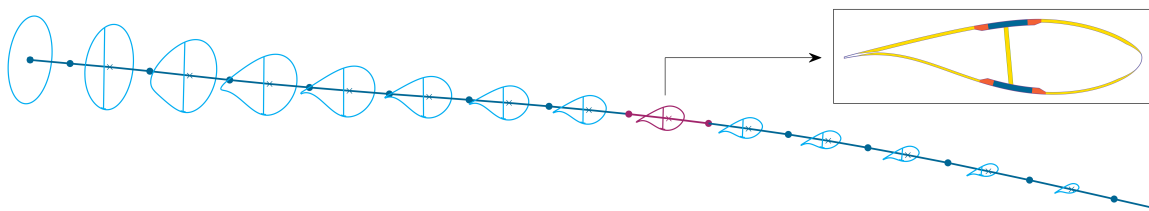


Figure 1.2: Exemplary plot of a coarse wind turbine blade finite element beam model [148]. The blue airfoil shapes depict the corresponding structural topology, whereas for the cross section highlighted in red, the BECAS plot is shown.

Due to their computational efficiency and reasonable accuracy, this type of model fidelity is popular for current aeroelastic analysis codes, such as BeamDyn [203], HAWC2 [123], FAST [181], GH Bladed [75], MBDyn [137], Dymore [18], and others. In general, these codes construct multi-body-systems of the complete turbine to calculate load histories under predefined environmental conditions, which constitute the design loads for the structural components, e.g., the rotor blade. Changing the structure directly leads to changes in the beam model and thus the aeroelastic analysis has to be processed again to obtain the design loads. This procedure is repeated iteratively until a certain convergence is achieved [175]. Since this procedure comprises several different operation states of the wind turbine, these codes make use of the computational efficiency of beam models.

Finite element shell model

Despite their high computational efficiency, beam models contain the risk of over-reducing the complex composite blade structure [107] and decouple the structural properties from the real blade design [70]. However, it may occur that problems originating from the design with simplified models are not revealed until later design stages, when higher fidelity models are applied [165].

As the composite thickness to chord length ratio of a rotor blade is typically small, a finite element model with shell elements containing a layup definition can be used [45]. Although it is usually referred to as 2D finite element model, such a model represents the three dimensional shape of the blade in contrast to a beam model, which only follows the three dimensional blade axis. Shell element models obviously result in a higher number of degrees-of-freedom, but it is still half as much

as in solid finite element models. Apart from torsion, shell finite element models predict the overall global blade behavior almost as well as solid finite element models [156].

Since these models offer an excellent compromise between accuracy and computational costs, they are state of the art when applied for detailed analyses [206] and represent the lowest necessary fidelity level. This is also the reason why a lot of research or commercial model generation tools apply shell models, such as NuMAD [24, 163], FOCUS [61], FEPROC [167] or a tool from the Ghent University [157]. Shell element models are often built in conjunction with solid elements for the adhesive volumetric joints [145, 146, 167]. Figure 1.3 shows an exploded view of such a hybrid model, where the composite parts are modeled as shell elements, i.e., the components shell, shear webs and trailing edge, and the adhesive joints (yellow) are represented by solid elements. However, the shell element models suffer from a significant drawback: the element node offsets from the mid-plane lead to significant misprediction of the structure's torsional behavior [38, 81, 122, 154]. This can present a dramatic problem, especially as modern bend-twist coupled blades are intentionally designed to twist during bending in order to reduce aerodynamic loads [69, 141, 183]. Additionally, shell elements overlap in convex or tight structures, e.g., leading or trailing edge, or the shear web corners, and provoke excessive weight and stiffness. Hence, these types of models are reaching their limits and have to be applied with caution for modern wind turbine blades.

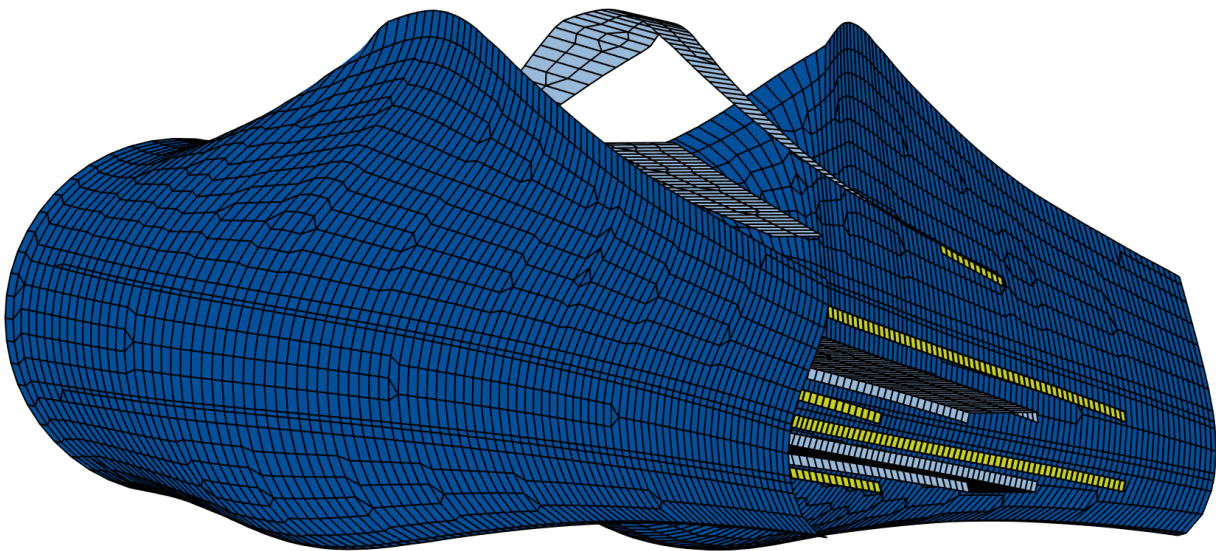


Figure 1.3: Exemplary exploded view of a coarse wind turbine blade hybrid finite element model with shell elements for the composite parts (blue, light blue) and solid elements for the adhesive parts (yellow).

Finite element solid model

The most accurate, but also computationally most expensive modeling approach is (layered) solid element modeling [45]. Whenever very detailed analyses are necessary, such as accurate stresses in adhesive joints [156] or through-thickness stresses for interlaminar fractures and delamination [93, 151, 152], the use of solid element formulations is unavoidable. In contrast to homogeneous solids, layered solid elements can be assigned a laminate layup and thus easily represent a composite section. Recent research follows the trend to use solid element models [34, 94, 156] and even integrate them in their model generators such as the tool from the Ghent University [157] and the modeling code within CP-Max [32]. Figure 1.4 shows an example of a solid finite element model segment used by Haselbach et al. [94].

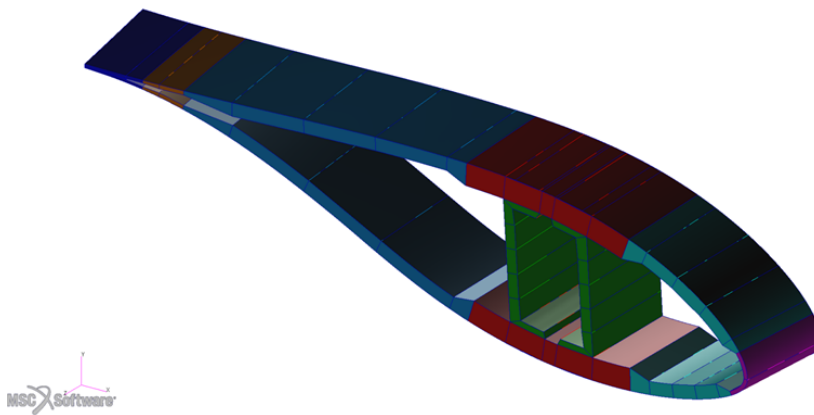


Figure 1.4: Exemplary segment of a wind turbine blade finite element model with solid elements [94].

Models using continuum shell elements (also referred to as solid shell elements) are intermediate solutions between shell and solid element models [161]. These can represent the correct shape of the blade also as solid elements, but include shell kinematics, thus should only be used for thin-walled structures.

1.2.2 Model updating approaches

After a structural model is designed, simulated, and verified, the structure is finally ready for the production process. After this structure is tested, it may have significant differences to the simulated model's properties or response. This is when model updating (also referred to as calibrating) comes into action, seeking to correct the inaccurate model parameters with the aim to improve the model prediction of the real structure [142]. This of course does also apply to other science disciplines. Model updating is a huge scientific field due to a vast amount of available algorithms. There are also different ways of categorizing the model updating problem and the solving approach, though, this section divides it according to Figure 1.5 following Marwala [135], Sarker et. al [170], Lin et. al [129] and Guantara [87]. First, the approaches are categorized into deterministic, heuristic and probabilistic methods. The deterministic methods rely on the analytical properties of a problem and systematically generate a sequence of points trying to converge to an optimum [129], examples are linear programming [49], nonlinear programming [19], and mixed-integer programming [128]. Heuristic approaches rather search for the optimum solution and are applied whenever the deterministic approaches fail to solve a complex optimization problem. Usually they are inspired by natural process and can generally be categorized in *evolutionary*, *swarm intelligence*, *physical/natural processes*, *direct search methods* [91]. Both aforementioned model updating approaches, deterministic and heuristic, solve an optimization problem. The probabilistic model updating is typically performed with the Bayesian model updating [135]. According to Baye's theorem the posterior distribution of the model parameters is inferred by expensive Markov Chain Monte Carlo sampling [178], thus, it benefits from an uncertainty estimation the other approaches cannot offer. A different updating method, which addresses uncertainties, are fuzzy models, though these are non-probabilistic [113].

The deterministic and heuristic approaches can be further divided into multi-objective and single-objective optimizations. The multi-objective optimization has more than one observed model output. Depending on the approach these can either be kept as individual optimization targets, which would result in a Pareto-optimization or can be grouped to a single objective function, which is called scalarization. The objective function, sometimes also referred to as fitness function, is then treated similar to a single-objective optimization. Further, this problems can be divided into linear problems (LP) and non-linear problems (NLP), which defines if the objective function is linear or non-linear.

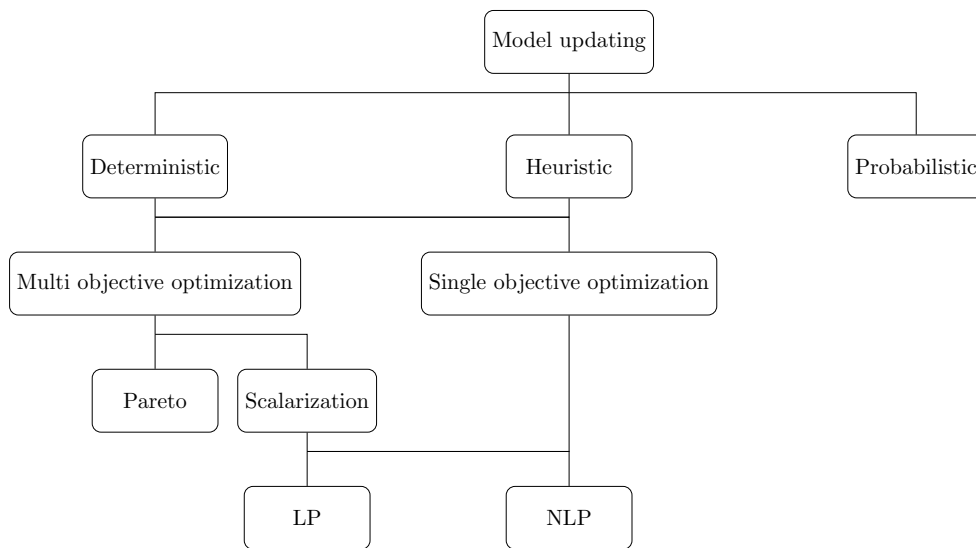


Figure 1.5: Categories of optimization methods used for model updating.

This does by far not cover all ways of categorizing model updating or especially optimization problems. Due to the problem complexity of updating wind turbine blade models, the further state-of-the-art review will focus on model updating with heuristic global optimization algorithms and limit any discussed applications at least to finite element structural model updating. For further information on the other methods and categorization please refer to [21, 35, 87, 118, 129, 134].

Heuristic optimization algorithms in model updating

Heuristic approaches are widely used whenever problem-specific algorithms are not feasible or simply too expensive to develop [160]. *Heuristic* derives from the Greek *heuriskein*, which means *to discover* [213] and describes well the random search nature of these algorithms [36]. They act as universal solver, though, simple linear convex problems are probably better solved in a conventional deterministic way [125]. Heuristic methods seek for a satisfying solution, which not necessarily is always the converged optimum [126]. Another big advantage of heuristic methods is, that they do not need specific insight of the model and can thus deal with black box problems, where only input and output or an objective function are given [17].

Evolutionary algorithms generally follow the principal *survival of the fittest* [138]. From there also derives the term fitness function instead of objective function. The two most popular algorithms are the *evolutionary computation* [170] and the *genetic algorithm* [59]. These algorithms compute an initial solution (often referred to as population), based on this solution a new one is generated by perturbation. If the new solution improved in terms of the fitness value, it is set as the current solution. The iteration is repeated until the fitness function change reaches a defined tolerance. The different algorithms have different strategies for the generation of the initial solution and the perturbation [170, 208].

Swarm intelligence algorithms are sometimes also assigned to *evolutionary algorithms*, however, they deserve an individual category [112]. These algorithms are inspired by the collective intelligence behavior of a self-organized swarm species [31]. It relies on the self-organized interaction and exchange of information between the individuals in order to achieve a superior goal [1]. Here the best-known algorithms are the *ant colony optimization* and the *particle swarm optimization* [30]. *Ant colony optimization* mimics the behavior of an ant colony where ants depositing pheromone on

their way to mark beneficial paths to be followed by other ants [60]. Whereas the *particle swarm optimization* imitates the movement and reaction of, e.g., fish schooling or bird flocking [50]. Both are population-based stochastic optimization strategies.

Additionally to the aforementioned algorithms, other optimization methods are inspired by *physical and natural processes*, from different fields such as magnetism, chemistry, and thermodynamics [169]. Here, the most preferred algorithm is the *simulated annealing*, originally developed by Kirkpatrick et al. [116] and Černý [43], which is based on the thermodynamic analogy of annealing of solids [196].

Unlike gradient based algorithms, the *direct search methods* search from the latest point a set of surrounding points looking for a better solution [127]. Most of them rely on geometric strategies to explore the solution space [16]. Well-known algorithms are the *Nelder-Mead* [144] and the *pattern search* algorithm [101]. The *Nelder-Mead* algorithm starts off with a simplex. A simplex is the most simple geometrical form representing a hypertetrahedron with $n + 1$ vertices in the n dimensional parameter space, e.g., a triangle in a 2D space or a pyramid in a 3D space. The algorithm evaluates the objective function for each corner point of the simplex. Then, the worst point is then replaced by its geometric reflection over the centroid of the rest points generating a new simplex. Combined with some geometric extension and contraction processes, this procedure is done iteratively until it reaches a tolerance [121]. It basically moves the simplex through the parameter space and when it is approaching the optimum, it reduces the size of the simplex until convergence.

Apart from these pure heuristic optimization methods, any combination of those can be imaginable as hybrid approaches. Most likely a global search is used for exploration followed by a more local search for exploitation of the optimum [111]. Examples show combinations of bee colony algorithm & pattern search [111], pattern search & simulated annealing [96], or evolutionary algorithms & Nelder-Mead search [67, 119].

Model updating in structural engineering

Structural engineering nowadays employs model generation and finite element simulation to obtain responses of a certain structural component. Gradient information of the transformation from input to response data is hardly available, which directly suggests the use of non-gradient methods [91], e.g., heuristic optimization approaches for model updating. Usually, the updating of structural models is performed on structural dynamics, thus natural frequencies and mode shapes [174]. Therefore, the most popular metric to define an objective function is the modal assurance criterion (MAC) [2] or derivatives thereof [6]. The MAC-value is a statistical indicator, that describes the degree of consistency between mode shapes [155].

Marwala [135] gives a broad overview and good introduction into the field of finite element structural model updating, while applying all types of heuristic optimization methods, hybrid combinations of them, neural networks, or Bayesian approaches. There are many fields of application for finite element model updating of structures. Bridges are probably the predominant ones, e.g., Deng and Cai [54] used a hybrid response surface method and genetic algorithm, others used a particle swarm algorithm [176, 187, 188, 207]. However, this state-of-the-art review will focus on wind energy related applications.

Several updating studies are presented on the full wind turbine multi-body model, e.g., a FAST model is adapted with a simplex search method based on inertial measurements from operational monitoring [205], Velazquez and Swartz [198] use a simulated annealing algorithm to update the mod-

els for structural health monitoring, whereas others apply a multi objective pareto front optimization to account individually for each mode shape consistency [132], or a gradient based approach [143] to update offshore wind turbine multi-body models. Rinker et al. [164] calibrated a full HAWC2 model of a Vestas V52 turbine with manual step by step tuning and sequential least square method until meeting the desired properties of each single component.

The tower, foundation and, in case of offshore turbines, the substructure also received research attention in terms of model updating. Tamizifar et al. [185] applied model updating for calibrating the material's Young's modulus E and density ρ of a wind turbine tower by minimizing the objective function based on the MAC and using a genetic algorithm. Others used model updating for damage identification applying Levenberg-Marquardt optimization [216] or sequential quadratic programming [172]. Augustyn et al. [15] used a sensitivity based model updating for offshore jacket structures.

Latest research publications show, model updating for wind turbine rotor blades is performed on different fidelity levels depending on the application, but, without exception, (to the best of the author's knowledge) all studies are based on structural dynamic response. Model updating on the beam level is primary used for damage identification and localization. Hofmeister et al. [100] successfully used model updating with different approaches such as particle swarm, genetic algorithm, global pattern search, and sequential quadratic programming to identify stiffness drops in a generic finite element Euler-Bernoulli beam model of the NREL offshore 5-MW rotor blade [109]. In parallel, Bruns et al. [39, 40] compared different optimization algorithms for wind turbine blade model updating of generic two and five dimensional updating problems based on a similar model of the NREL offshore 5-MW. Schröder et al. [173] performed a hybrid model updating with Simulated Quenching as global exploration algorithm and an adaptive Sequential Quadratic Programming as local exploitation algorithm on a Timoshenko beam model. This way they successfully tried to locate local ice accretion on rotor blades, by identifying additional masses on the experimentally tested blades. The last presented publication is that of Bottasso et al. [33], who applied Sequential Quadratic Programming to update a Timoshenko beam model. However, they used shape functions to adapt a spanwise baseline property in order to reduce the dimensionality of the problem and make the effects less local than discrete properties.

Apart from the aforementioned publications all covering finite element beam model updating, a few publications on model updating of higher fidelity models, i.e., shell or solid element models, exists. The first is presented by Turnbull and Omenzetter [190, 191], using fuzzy finite element model updating of a simplified laboratory blade structure. They applied both, a particle swarm optimization as well as a firefly optimization, and compared the results. Knebusch et al. [117] published a model updating study considering a 3D finite element model with shell elements representing the composites and with the largest parameter space including 59 fields of the blade shell where stiffness and density can be varied, i.e., 118 parameters in total. The response used for the model updating with a gradient-based optimization were experimental high fidelity modal characterization tests of the blade [88].

Model updating - an inverse problem

Most model updating approaches and applications tackle the updating process by minimizing the response errors between the updated model and the target response. But at the end, model updating is an *inverse problem* [178], which however, most of the times is ill-posed [63]. The forward process, i.e., mapping of a response to a given set of input, is typically an established physical model/process and well known. In wind turbine blade modeling this, so called *forward process*, is briefly depicted in Figure 1.6 and includes computing the outer shell geometry, before defining additional components such as webs, after which the material and composite layup is mapped onto the corresponding blade

sections. The structure is then divided into finite elements forming the model, which undergoes a modal analysis before extracting the relevant nodal displacements of the mode shapes and their corresponding natural frequencies.

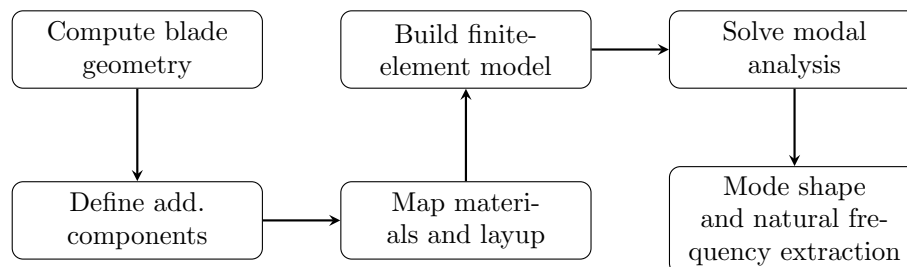


Figure 1.6: Brief description of the forward process in wind turbine blade modeling and model evaluation.

The *inverse problem* is the inverse of the *forward process* and is most likely intractable [10], due to ambiguities, non-linearities or non-invertible processes, such as the finite element modal analysis or even the geometrical modeling itself. Directly approaching the *inverse problem* requires the prediction of a conditional posterior distribution, which can be achieved by statistics and more precisely by Bayesian methods. All of these methods are based on the Bayes' theorem [103]:

$$p(\theta|D) = \frac{p(D|\theta) \cdot p(\theta)}{p(D)} \quad (1.1)$$

It describes the posterior probability $p(\theta|D)$ of the model parameters θ given the data D as a product of the likelihood $p(D|\theta)$ and the prior probability $p(\theta)$, normalized by the model evidence $p(D)$, i.e., the probability of the given data D . The prior describes the prior knowledge or a guess for the model parameters and typically is chosen to be a known distribution to easily sample from it. As the model evidence $p(D)$ is only used to normalize the expression and normally posterior relations are relevant, the above equation reduces to:

$$p(\theta|D) \propto p(D|\theta) \cdot p(\theta) \quad (1.2)$$

However, evaluating the likelihood $p(D|\theta)$ for complex applications is sometimes not feasible or at least computationally very expensive. Therefore, the best-known approach is approximated Bayesian computation (ABC) method, which estimates the posterior probability without the need to evaluate the likelihood function [92], while using a rejection sampling algorithm to compute the posterior distribution [20]. This requires a lot of sampling, first because a significant amount of generated samples is rejected and second it needs to converge to an approximated likelihood. However, this algorithm could predict the true posterior with infinite samples. Newer methods, propose combining the ABC algorithm with the Markov chain Monte Carlo technique (ABC-MCMC) [133], with the sequential Monte Carlo technique (ABC-SMC) [186], or with the population Monte Carlo technique (ABC-PMC) [168]. Although, these extensions improve the computational efficiency by reducing the rejection rate, i.e., less function evaluations are necessary, the overall computational effort is still huge depending on the complexity of the model and the convergence limit defined.

The ABC-method and its derivatives are relatively new approaches, especially as the fast growing computational power makes such heavy sampling techniques feasible. However, it already finds its application in the structural model updating (see, e.g., [22, 71, 136]) and even in wind energy, where Camargo et al. [42] updated a concrete tower finite element model via ABC-SMC.

Inverse modeling with artificial neural networks

Recent research in artificial neural networks provides alternatives to the ABC-methods to approximate the posterior distribution of an inverse problem. Several approaches such as variational autoencoder (VAE) [7, 80, 184] or generative adversarial networks (GAN) [3, 177] were proposed to deal with inverse problems. However, this work will focus on the field of invertible neural networks (INN). In contrast to VAE or GAN, the INN are bidirectional due to the mathematical structure of their coupling blocks [57]. According to Ardizzone et al. [11], these types of artificial neural networks have the following characteristics:

- they are bijective, i.e., they can be unambiguously inverted
- the forward and inverse path are efficiently computable
- they have tractable Jacobians for both paths

The tractable Jacobians and their determinants present the opportunity to train the INNs using the maximum likelihood loss [56, 57] and by that offer a basis to probabilistic inverse modeling according to the Bayes theorem, though without the need of ABC-methods and their cost intensive computation. Ardizzone et al. [10] presented an extension of these INNs, to feed an observation as condition into the coupling blocks, and then calling the complete network conditional invertible neural network (cINN). Although, invertible neural networks are relatively new, they already can demonstrate several successful applications such as, image processing [5, 10, 12, 56, 115], medical applications [4, 55, 85, 202], geophysics [8, 215], and astrophysics [51, 90, 120]. Recently, INNs have also been applied to classical engineering problems such as material science [72] or aerodynamic design for gas turbine airfoils [78]. Further information on invertible neural networks is elaborated in the Chapters 3 and 4 (i.e., second and third paper).

1.3 Research gap and objectives

This section will define the research gap from the presented state of the art for wind turbine blade modeling and model updating. Subsequently, the specific objectives of this thesis are presented.

1.3.1 Research gap

Reliability plays a major role when trying to improve a relatively mature component such as a wind turbine blade. However, it is one of the most crucial characteristics for a stable operation of a wind turbine and with significant impact on turbine downtimes, especially offshore. Reliability analysis is directly connected to uncertainty estimation, without this insight no prediction of the system's condition can be estimated. Though, uncertainty analysis already starts at the manufacturing level. Any additional information on the real state of the manufactured blade, such as, e.g., in-situ material properties, mass distributions, or defect detection, can highlight extreme manufacturing deviations/errors, which could lead to fatal damages during operation. Additionally, this information can be applied for digital twin creation or enhancement.

Model updating is a tool to recover structural or material properties from a given component's response, which in most structural problems is a dynamic analysis. Commonly used methods for model updating are optimization approaches such as heuristic optimization algorithms to minimize the error between the model's response and the target response. However, these classical optimization approaches lack an uncertainty quantification of their predicted model parameters. This raises the necessity to apply probabilistic model updating methods, most likely based on the Bayes' theorem,

such as the approximated Bayesian computation. Although, these algorithms can return a good prediction of the posterior distribution of the model parameters, they are computationally very expensive.

This expense increases with the model's complexity. Therefore, published model updating applications for wind turbine rotor blades typically rely on beam models of the blade, as these represent a type of model reduction and by that reduce the model complexity for the updating process. However, updating a higher fidelity model such as a shell or solid finite element model, would give more detailed insight into the condition of a manufactured blade. To the best of the author's knowledge, only Knebusch et al. [117] published a model updating of a full scale wind turbine blade finite element shell model by applying gradient based optimization. This study also had the highest number of model parameters to update (118 parameters), but, did not account for a proper uncertainty analysis of the results.

Thus, a significant research gap consists in efficiently recovering manufacturing deviations and uncertainties of manufactured rotor blades. This in consequence would help to enhance digital twins and manufacturing quality assurance as well as improve reliability of wind turbine blades overall.

1.3.2 Objectives

With the research gap defined above, the overall objective of this thesis can be stated as follows:

Establishment of a computationally efficient approach to recover the deviation and uncertainty of rotor blade design parameters by updating higher fidelity models.

This concept aims for several improvements compared to the state-of-the-art approaches. The uncertainty evaluation of model parameters after manufacturing enables a more reliable blade assessment. While the higher fidelity level of the finite element models used for the model updating give more detailed insights into the structural properties. Besides, a computational efficient updating method can enhance digital twin creation by feeding in the blade setup as built, not only as designed.

However, the combination of model updating of high fidelity models, uncertainty evaluation and computational efficiency seems to be a contradiction. Therefore, this work is structured via the following work packages to break down the thesis effectively:

1. Development of a fully parameterized model generator, that can efficiently create wind turbine blade models at different fidelity levels,
2. Validation of the modeling methodology,
3. Establishment of a model updating approach with uncertainty evaluation for structural wind turbine blade models at a state-of-the-art fidelity level (beam model),
4. Evaluation of the efficiency and accuracy of the presented approach by comparison with current model updating techniques,
5. Application of the model updating procedure to a higher fidelity model (3D-model).

1.4 Thesis concept

After defining the research gap and the objectives of this work, the thesis concept is summarized in this section. The thesis is presented in a cumulative way, with three papers published in peer-reviewed journals. This section includes the outline of the thesis, discusses the link between the publications, and highlights the author's contribution to each paper.

1.4.1 Outline

The content of the presented thesis is divided in five chapters plus the introduction and conclusion. The second to fourth chapter consists of the published papers. The chapters sequentially deal with the work package, which were defined in the objectives in the previous section, in order to fulfill the overall objective. Figure 1.7 depicts the outline of this thesis with its different chapters and a brief description of the respective content. The complete thesis will be based on a 20 m demonstration blade, that was built and tested during the SmartBlades2 research project [179].

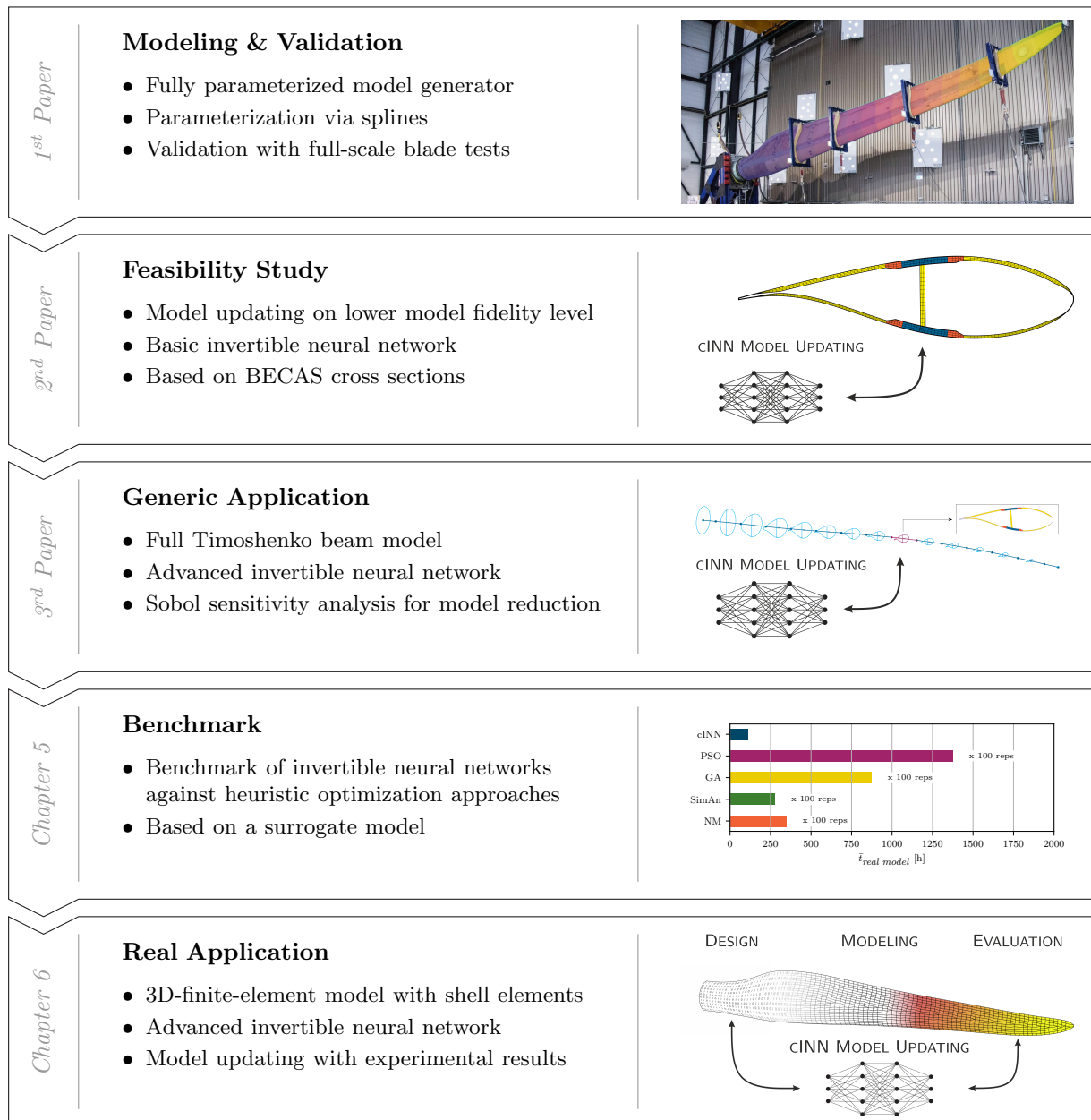


Figure 1.7: Thesis Outline

After the introduction, the second chapter, i.e., the first paper, deals with the two first work packages by describing and validating the modeling tool. The developed fully parameterized model generator is first described and then applied to create a 20 m finite element shell model of a real rotor

blade. An experimental, full-scale, dynamic modal analysis and a static load test were performed on the rotor blade. The same tests were conducted in a simulation using the generated model. Subsequently, the model was then validated against the experimental results.

After creating and validating the modeling tool, a feasibility study of the model updating via invertible neural networks is realized in the third chapter (second paper). This study is aimed to show the applicability of invertible neural networks for structural model updating problems with uncertainty treatment in the context of rotor blade engineering. Therefore, the application was limited to the lowest fidelity level of wind turbine modeling, the cross sectional model in combination with a very basic invertible neural network.

The fidelity level is increased to state-of-the-art Timoshenko beam models in the fourth chapter (third paper), completing the third work package defined in the objectives. The invertible neural network is extended to include an additional conditional network. The results of this generic application of model updating are analyzed and discussed revealing the potentials of the approach.

In order to prove the efficiency of the presented method (fourth work package) a benchmark is performed in the fifth chapter, comparing the model updating via invertible neural networks with optimization-based updating using heuristic methods, which represents the state of the art. This benchmarking is conducted on the basis of a surrogate model of the model generator and evaluation tool chain, as all the different approaches demanded high sampling numbers.

In the sixth chapter, the application is elevated to the 3D finite element shell model level, representing the highest fidelity level in this thesis. To fulfill the last work package of the objectives, the approach was tested for a generic updating problem. Additionally, it is also applied to existing experimental test data to update the respective model parameters representing the blade as-built. However, updating the real blade showed inaccuracies, thus a root cause analysis is conducted to the end of this chapter.

Finally, the conclusion summarizes the outcome of the thesis and an outlook for possible future research is given, covering open questions raised by this work.

1.4.2 Declaration on the included publications

The overall goal of the publications is to establish and evaluate the model updating framework for rotor blades via invertible neural networks, before it is then evaluated in terms of its efficiency and applied to a high fidelity model in the additional chapters to fulfill the work packages and objectives of this thesis. All three papers are published in open-access, peer-reviewed, scopus-listed, and wind energy-specific journals, the Wind Energy Science journal and the Wind Energy journal. Although, the author's contribution is already declared in a statement at the end of each paper, they will be additionally highlighted in this section.

The first paper *Validation of a modeling methodology for wind turbine rotor blades based on a full scale blade test* [147] establishes the basic tool for producing sample models of rotor blades with varying model parameters. The tool MoCA – Model Creator and Analyzer is presented, which includes a fully parameterized modeling methodology for wind turbine blades. Within the Smart-Blades2 research project [179] a 20 m demonstration blade was built and tested. These conducted tests are briefly documented in this paper to settle the necessary boundary conditions and for reproducibility in the simulation. After that the test is replicated in simulations and used to validate the generated model and by this the modeling methodology. The author of this thesis implemented the parametrization and modeling methodology of MoCA, conducted the numerical simulations, compared the simulations with the tests, and wrote the core paper. Bernd Haller planned, executed, and documented the tests. Claudio Balzani guided the author in the conception of the ideas and supervised the writing, structuring and reviewing of the paper.

After the modeling tool is established, the second paper *Model updating of wind turbine blade cross sections with invertible neural networks* [149] consists of a feasibility study for structural model updating with invertible neural networks. This feasibility study is performed on the lowest fidelity level, the blade cross section. In the first instance a one-at-a-time sensitivity analysis is performed to identify relevant input parameters and output parameters in order to reduce the updating problem. After the invertible neural network is trained, the model updating capabilities are tested on different cross sections along the blade and with different available sets of output parameters. The author of this thesis established the framework, performed the sensitivity analysis for the parameter subspace selection, trained and applied the neural network, analyzed the updating results, and wrote the core paper. Lynton Ardizzone, participated in the conceptualization of the paper and supported the author on the invertible neural networks as well as the writing and reviewing of the paper. Claudio Balzani supervised all steps of the research and the writing of the paper.

Since the feasibility study on the new model updating methodology was successful, the complexity and scope was increased to a higher model fidelity in the third paper, *Model updating of a wind turbine blade finite element Timoshenko beam model with invertible neural networks* [148]. Here the model updating procedure is applied to a full finite element Timoshenko beam model. A global sensitivity analysis approach, the Sobol' method, is applied to identify a relevant parameter subspace. Subsequently, the invertible neural network is extended by an additional conditional network in order to enhance the information extraction from the conditions, i.e., outputs. The model updating is performed on generic problems and evaluated with respect to, accuracy, ambiguities, cross-correlation, and robustness. It is also identified if a sensitivity analysis is necessary for this approach at all. The author of this thesis prepared the concept, established the methodology, conducted the analysis, wrote the paper and processed the review. David Melcher, supported the author in transferring the cross sectional properties to an ANSYS beam model and in conducting the finite element analysis. Claudio Balzani supervised all steps of the research and the writing of the paper.



Validation of a modeling methodology for wind turbine rotor blades based on a full scale blade test

1st Paper

Detailed 3D finite element simulations are state-of-the-art for structural analyses of wind turbine rotor blades. It is of utmost importance to validate the underlying modeling methodology in order to obtain reliable results. Validation of the global response can ideally be done by comparing simulations with full scale blade tests. However, there is a lack of test results for which also the finite element model with blade geometry and layup, as well as the test documentation and results are completely available.

P1-1	Introduction	18
P1-2	Model creation framework	20
P1-3	Modeling of the test blade	22
P1-4	Test description and virtual modeling	24
P1-5	Comparison of experimental and simulation results	28
P1-6	Summary and conclusion.....	34
P1-7	Appendix A: Static bending test results	36
P1-8	Appendix B: Local strain results.	37



Wind Energ. Sci., 7, 105–127, 2022
<https://doi.org/10.5194/wes-7-105-2022>
 © Author(s) 2022. This work is distributed under
 the Creative Commons Attribution 4.0 License.



Validation of a modeling methodology for wind turbine rotor blades based on a full-scale blade test

Pablo Noever-Castelos¹, Bernd Haller², and Claudio Balzani¹

¹Institute for Wind Energy Systems, Leibniz University Hannover, Appelstr. 9A, 30167 Hanover, Germany

²Department of Rotor Blades, Fraunhofer Institute for Wind Energy Systems (IWES),
 Am Seedeich 45, 27572 Bremerhaven, Germany

Correspondence: Pablo Noever-Castelos (research@iwes.uni-hannover.de)

Received: 16 March 2021 – Discussion started: 29 April 2021

Revised: 25 October 2021 – Accepted: 13 December 2021 – Published: 21 January 2022

Abstract. Detailed 3D finite-element simulations are state of the art for structural analyses of wind turbine rotor blades. It is of utmost importance to validate the underlying modeling methodology in order to obtain reliable results. Validation of the global response can ideally be done by comparing simulations with full-scale blade tests. However, there is a lack of test results for which also the finite-element model with blade geometry and layup as well as the test documentation and results are completely available.

The aim of this paper is to validate the presented fully parameterized blade modeling methodology that is implemented in an in-house model generator and to provide respective test results for validation purpose to the public. This methodology includes parameter definition based on splines for all design and material parameters, which enables fast and easy parameter analysis. A hybrid 3D shell/solid element model is created including the respective boundary conditions. The problem is solved via a commercially available finite-element code. A static full-scale blade test is performed, which is used as the validation reference. All information, e.g., on sensor location, displacement, and strains, is available to reproduce the tests. The tests comprise classical bending tests in flapwise and lead–lag directions according to IEC 61400-23 as well as torsion tests.

For the validation of the modeling methodology, global blade characteristics from measurements and simulation are compared. These include the overall mass and center of gravity location, as well as their distributions along the blade, bending deflections, strain levels, and natural frequencies and modes. Overall, the global results meet the defined validation thresholds during bending, though some improvements are required for very local analysis and especially the response in torsion. As a conclusion, the modeling strategy can be rated as validated, though necessary improvements are highlighted for future works.

1 Introduction

Rotor blades are major components of wind turbines. They are susceptible to damages, which, in case they need repair, can result in severe turbine downtime (Reder et al., 2016). It is thus crucial to develop a blade design that withstands all expected loads without damage. Though a blade prototype is always tested at the full blade scale in the certification process (International Electrotechnical Commission, 2014), such tests are very costly and time-consuming, especially for growing blade dimensions (Ha et al., 2020). For this reason, full-scale blade tests are executed one time only per blade

design. Hence, a reliable and fast virtual blade design procedure is required. Full 3D finite-element (FE) analysis is accurate but computationally expensive. A widely used approach for wind turbine blade design is to carry out two-dimensional cross-sectional analyses which offer a reduced level of complexity but are a fast and efficient alternative for rotor blade pre-designs (Chen et al., 2010). Tools like VABS (Yu et al., 2002) or BECAS (Blasques and Stolpe, 2012) compute cross-sectional properties based on a 2D FE analysis, which is necessary to feed the aeroelastic models in order to recalculate the design loads on the turbine blades and close

the design iteration loop. Nevertheless, at a final stage 3D FE analyses have to be performed in order to obtain a reliable blade design and account for structural details, such as adhesive joints, longitudinal geometric discontinuities, ply drops, or local buckling analysis, which are not considered in a 2D FE analysis.

1.1 State-of-the-art 3D finite-element modeling of wind turbine blades

Automated model creation is state of the art and a key to enhancing the design process significantly by reducing computational time, increasing the possible number of design loops, and avoiding modeling errors caused by the user during a manual model creation. Among a vast selection of common software tools originated from the scientific community, QBlade (Marten et al., 2013) for example focuses on the aerodynamic blade design, applying only an Euler–Bernoulli beam approach for the structure. Sandia’s NuMAD (Berg and Resor, 2012) additionally contains a more sophisticated structural description taking into account a composite layup definition for the blades’ sub-components. The same holds for the software package FOCUS developed by WMC Laboratories, now part of LM Wind Power (Duineveld, 2008), which is a state-of-the-art tool used for blade design in many engineering offices. In FOCUS the user discretizes stations in the spanwise direction with all necessary geometrical information of these particular cross section and in between the tool interpolates linearly all missing data. Hence a high discretization of stations along the blade span (e.g., 45 stations for a 20 m blade) is necessary to correctly reproduce non-linear changing geometrical or material information in the spanwise direction.

Another more advanced tool is the optimization framework Cp-Max; see Bottasso et al. (2014). The parameterization is based on mathematical functions for the blade design description in the spanwise direction. This method has the advantage of reducing the number of stations along the blade without losing information in between, while enabling the framework to efficiently manipulate the parameters during optimization. The focus of the optimization framework is to find the best compromise between accuracy and costs. A similar blade parameterization is used within the FUSED-Wind framework (Zahle et al., 2020), which contains spline descriptions for each parameter as shown in the prominent example of the DTU 10 MW reference blade design (Bak et al., 2013). An interface to the framework was later incorporated into the Python tool FEPROC, and the correct modeling process was verified against the DTU 10 MW reference blade (Rosemeier, 2018). Another blade modeling tool developed at Ghent University also relies on function-based descriptions of the blade parameters and focuses on a modular principle of finite-element (FE) constellations for modeling the different blade components and joints in the structure (Peeters et al., 2018). The latter algorithm and Cp-Max are

capable of generating solid element models, while the others rely on more common shell element representations.

A lot of scientific contributions deal with FE modeling but focus on structural details such as trailing edge adhesive joints. Eder and Bitsche (2015) for instance use a local model with fracture analysis to deduce the debonding between shell and adhesive due to buckling and validate the behavior against experimental results. Ji and Han (2014) also apply fracture mechanics and use a detailed model at the shear web adhesive joint to analyze crack propagation in the bond line. Most of these locally detailed models are used within a global–local modeling approach like in Chen et al. (2014) to reduce the global model complexity while keeping a high level of detail at local spots.

1.2 Objectives of this paper

Though some of these model creation frameworks may work with functions or splines describing the blade’s geometrical or layup information, most of them work with a reasonably high number of airfoils/stations that in addition to the blade’s geometry yield the outer blade shape by a global linear or higher-order interpolation between the airfoils.

The presented method combines and extends several aspects of the different aforementioned software packages. The benefits are the following.

- It generates airfoils independent from any neighboring geometry and uses the relative thickness distribution to position these along the span. This ensures the geometry distribution, as it avoids any overshoot due to spanwise geometry interpolation.
- Any parameter which may vary over the radius can be defined as spline, e.g., relative blade thickness, layer thickness, material density, or stiffness.
- It enables flexible and easy parameter studies due to the simple parameter variation based on splines.
- It is designed for research, as different modules can be easily replaced by an alternative code, e.g., airfoil interpolation, adhesive modeling.
- It generates an FE model in MATLAB and already provides an interface to Ansys Parametric Design Language (APDL ANSYS Inc., 2021) and BECAS (Blasques and Stolpe, 2012); however, interfaces to other FE software can easily be implemented.

Different FE modeling procedures can result in different deformation and stress solutions, though based on the same model parameters; see (Lekou et al., 2015). Hence, it is important to validate modeling strategies by comparing simulations with full blade tests, which is the aim of this paper. A quasi-static full-scale blade test is performed, including not only bending tests in the flapwise and lead–lag direction – as

are usually executed in the context of blade certification (International Electrotechnical Commission, 2014) – but also torsion tests. This allows for an exceptionally detailed and thorough validation. Unlike other blade tests reported in literature (Chen et al., 2021, 2017; Jensen et al., 2006; Overgaard and Lund, 2010; Overgaard et al., 2010), the aim of the tests in this work is not the validation of failure models. Hence, the blade is not loaded up to failure.

The paper focuses on measuring and validating primarily the global elastic response of the blade, expressed in terms of deflections, mass distribution, and modal characteristics. That is why neither global–local modeling approaches nor very local mesh refinements are considered for this study. However, secondly the local behavior in terms of strain levels along defined cross sections is measured and compared to find a qualitative agreement with the model. The blade under investigation is the SmartBlades 2 DemoBlade, a 20 m blade from the SmartBlades 2 project (SmartBlades2, 2016–2020), which includes prebending and a presweep towards the trailing edge. The blade is modeled with our in-house blade model creation tool MoCA (Model Creation and Analysis Tool for Wind Turbine Rotor Blades), taking into account some major manufacturing-related deviations. The test setup and the load introduction are approximated via a combination of suitable boundary conditions and multiple point constraints. The simulation results are thoroughly compared with the test measurements.

1.3 Outline

The modeling strategy is addressed in Sects. 2 and 3. The test setup is described in Sect. 4. The blade was cut into segments after the tests in order to accurately measure the mass distribution and the locations of the centers of gravity along the blade. These measurements are also described in Sect. 4. The simulation versus test comparison is reported in Sect. 5, followed by the conclusions in Sect. 6.

2 Model creation framework

A framework to automatically generate fully parameterized 3D FE models of wind turbine rotor blades from a set of parameters was developed at the Institute for Wind Energy Systems at Leibniz University Hannover. The purpose of this tool called MoCA (Model Creation and Analysis Tool for Wind Turbine Rotor Blades) is to enable users to investigate and analyze different blade designs or design parameter variations in an easy way, including structural details such as adhesive joints. The following section presents a brief description of the framework.

MoCA is based on a set of input parameters categorized as *geometry*, *plybook*, *structure*, and *material*. In general all parameters that describe a distribution along the blade are stored as splines over the blade's arc length, but even material parameters may be varied over the blade arc if necessary

by using a spline. The parameter set *geometry* contains all information on the outer geometry of the blade, i.e., all involved airfoils and their positions along the blade as well as the distributions of the relative thickness, chord length, twist angle, threading point location, prebend and presweep. The *structure* set is associated with the structural description of the blade. This includes the specification of shear webs, adhesive joints, and additional masses as well as cross-sectional division points that are mainly used to subdivide cross sections into different regions of interest. The *plybook* parameters contain the stacking information of different composite layups used in the blade. The parameter set *material* comprises all material properties assigned for the different materials. These can be either isotropic or anisotropic on the macroscopic scale. The user can also specify a composite material based on microscopic characteristics of the fiber and matrix constituents, which are then transformed to a laminate via the well-known rule of mixtures.

In Figs. 1–4, each block is labeled with an index, which will be used in the following description for reference to the blocks of the respective figures. The flowchart in Fig. 1 depicts the structure of the finite-element creation procedure implemented in MoCA on the basis of the parameter sets described above. First, the blade segmentation, i.e., the discretization in the spanwise direction, is defined. For each blade segment edge, a cross section of the blade is calculated (5–7) by evaluating the *planform* data (1). Then a finite-element discretization of the cross sections (8) is executed using the information of the *structure* (2), *material* (3), and *plybook* (4) parameter blocks. At this stage, an interface to the BECAS (Blasques and Stolpe, 2012) software (9) can be utilized to calculate the full 6×6 stiffness and mass matrices of a beam model. However, since our aim is to create a 3D blade model (12), we continue with the finite-element discretization in the spanwise direction (10–11) utilizing a hybrid shell element/solid element strategy. Therein, we use shell elements to model the composite laminates and solid elements for the adhesives. The 3D FE mesh includes the node-to-element connectivity and elemental material assignments. The boundary conditions are added and the FE model is translated to an input file for the finite-element solver of choice (13), which in our case is Ansys Mechanical (ANSYS Inc., 2021). In the following, we describe the different steps of this overall procedure in more detail.

Figure 2 visualizes the process of cross-section geometry calculation. After the blade segmentation, the *geometry* data splines (2–7) are evaluated (8) for the particular blade arc positions (1) of the segment edges. According to the spline-based interpolation of the relative thickness t_{rel} (10), an airfoil (AF) is linearly interpolated (16) between the basic input airfoils (9), which have the next higher and lower relative thickness. In contrast to a global blade shape interpolation, the use of a blade-independent airfoil interpolation enables the user to implement an own sub-function and replace the former. The interpolated airfoils are then scaled by the chord

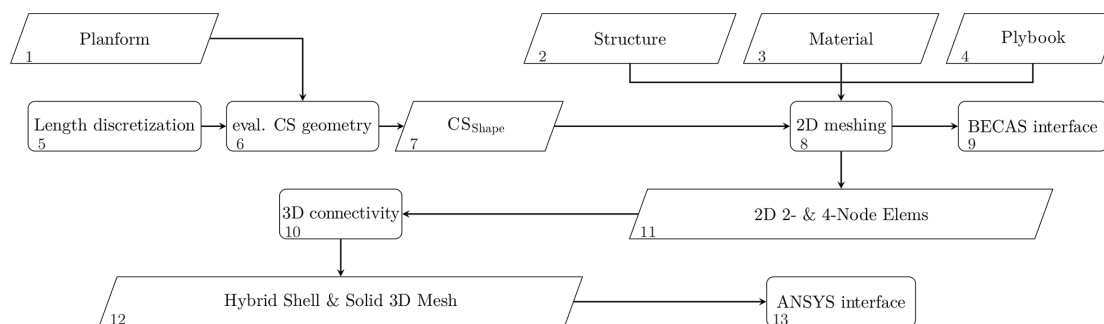


Figure 1. Flowchart of the finite-element model creation procedure in MoCA.

length c^* calculated via the respective spline (11, 17), shifted along the chord to the correct threading point by the coordinate tp^* (12, 18), and twisted by the twist angle θ^* (13, 19).

Until here, all transformations are performed in a 2D chord coordinate system with its final origin in the threading point. The cross sections are now shifted to the correct 3D position (20), locating the 2D cross-sectional threading center on the prebent (14) and preswept (15) global blade axis. By doing so, the 2D chord coordinate system is still parallel to the blade root plane. Hence, the cross sections are rotated by the slope angles of the prebend and presweep spline functions so that they are perpendicular to the threading axis. These shifted and rotated cross sections are the final cross-sectional shapes denoted by CS_{Shape} (21).

According to Fig. 1, the next step is the 2D cross-sectional meshing, which is executed using the cross-sectional shapes CS_{Shape} (7) and the parameter sets *structure*, *material*, and *plybook* (2–4). This process is presented in Fig. 3. As before, all data (2–6) are evaluated (7) for the particular arc positions (1) at the blade segment edges. The division points are generated on the cross-sectional shapes (9, 14). They serve to subdivide the cross sections into regions of different material layups and are also used to define the positions of the shear webs. Then the shapes of the shear web (10, 15) and the web and trailing edge adhesive joints (11, 16) are computed. The computation of the blade's outer geometry and its structural topology is now finished. After inclusion of the *material* (12) and *plybook* (13) information, the FE discretization (18) on the 2D cross-section level can be conducted. This yields either a two-dimensional mesh with four-noded plane elements for the BECAS (Blasques, 2012) interface (19) or a cross-sectional node map representing a hybrid 2D mesh with two-noded elements for the composite laminates and four-noded elements for the adhesives (20, 21).

The last step in the creation of a 3D finite-element model is to connect the 2D cross-sectional models; see Fig. 1 (10). The 2D line elements on the cross-sectional level yield four-noded shell elements on 3D level after the 3D extension, and the four-noded plane elements on cross-sectional level become 3D solid elements, respectively.

An additional module called TestRig is included in MoCA to model the boundary conditions similar to a full-scale blade test. Full clamping of the blade root represents the geometrical boundary conditions; i.e., all degrees of freedom are fixed at the blade root. Figure 4 shows the process of the TestRig module for the introduction of force-like boundary conditions. In the real blade test, a number of load frames introduces loads that approximate the target bending moment distribution (or torsional moment distribution, respectively). The TestRig module approximates the load frames by means of appropriate multiple point constraints (MPCs) and additional masses. For each load frame, the position along the blade (arc position) (1), the load frame width (4), the center of gravity (CoG), and the resulting mass (5) are specified as well as the load (2) and sensor points (3).

In the range where the load frame is located, MoCA searches all elements of the blade shell (6, 9) and defines 2D slave elements (12) that share their nodes. An additional cross section is created at the desired load frame position (7, 10) according to the procedure depicted in Fig. 2. In this additional cross section, the position of the load introduction (load point), the sensor points, and the center of gravity of the load frame are given in the blade coordinate system (11). These points are defined as master nodes (14). MPCs that connect the degrees of freedom of the master nodes and the slave nodes (13) by means of a rigid connection are included; i.e., there are no relative displacements between the master and the slave nodes. The additional mass of the load frame is applied to the CoG node (17), while the load is applied to the position where the load is introduced (16) in the real test (load point). In this way, we model solid and quasi-rigid load frames (15) and their effects on the blade response without adding detailed models of the load frames themselves, which is beneficial in the context of computational costs. The rigid connection implies that the deformation of the blade at the load frames is restricted. Similar to the real blade test according to IEC-64100 International Electrotechnical Commission (2014), the load frames neighboring blade sections cannot be evaluated, as the structural response is influenced by the quasi-rigid constraints.

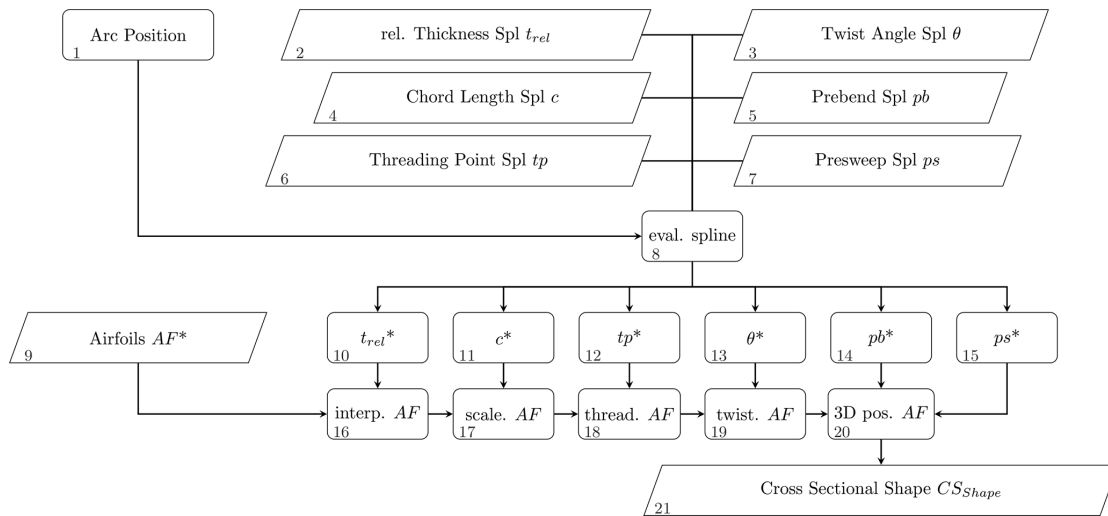


Figure 2. Flowchart of the calculation of the cross-sectional shapes CS_{Shape} .

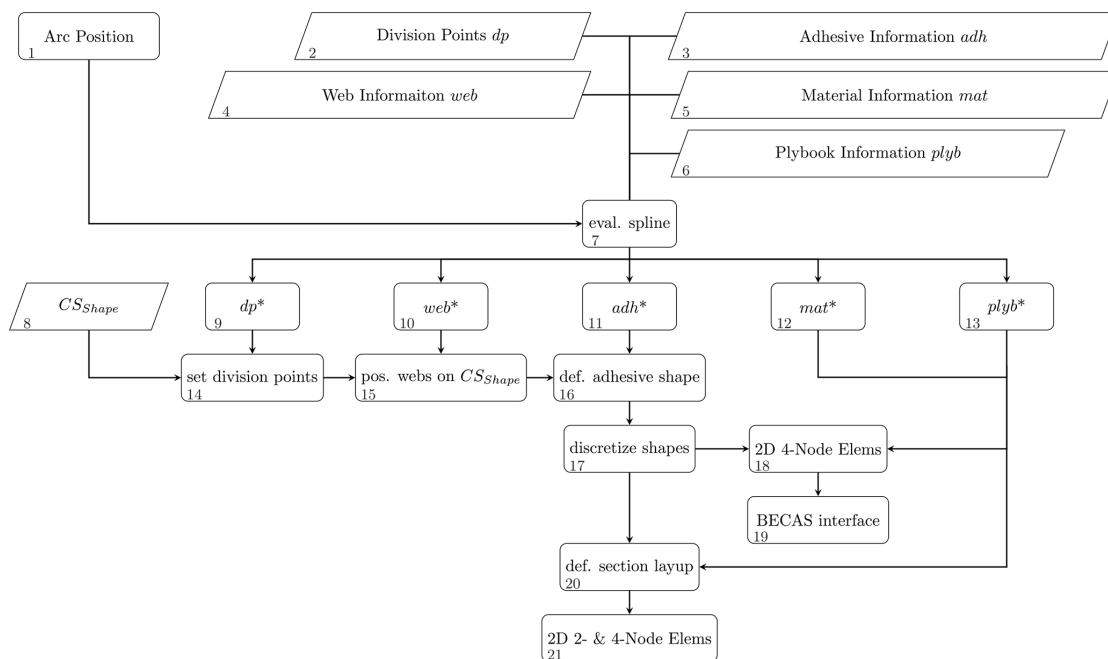


Figure 3. Flowchart of the 2D cross-sectional meshing routine in MoCA.

The 3D finite-element model including the mesh and the boundary conditions is translated to an input file for the finite-element solver of choice via an integrated interface.

3 Modeling of the test blade

This section briefly describes the blade under consideration, which is the SmartBlades DemoBlade, a 20 m long blade

with prebend and presweep. It was designed and manufactured in the coordinated research projects Smart Blades (Teßmer et al., 2016) and SmartBlades2 (SmartBlades2, 2016–2020). The blade is referred to as the DemoBlade in the following.

The DemoBlade was designed to investigate bend–twist coupling effects in wind turbine rotor blades. Therefore a presweep of 1 m towards the trailing edge at the tip is in-

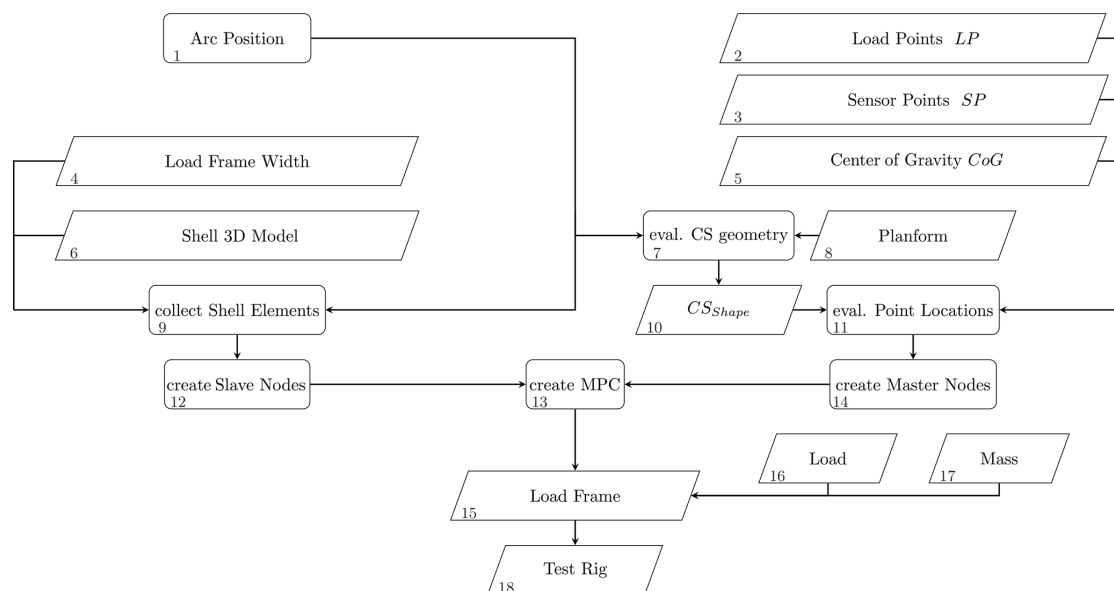


Figure 4. Flowchart of the procedure to model the boundary conditions in the TestRig module.

tended to introduce a torsional twist into the blade. The offset between the aerodynamic centers of the swept airfoils and the pitch axis introduces a torsional moment and thus a torsional deformation, i.e., a twist in the outer part of the blade. The twist reduces the angle of attack of the respective airfoils and hence the aerodynamic coefficients. In this way the aerodynamic loads can be reduced.

During the manufacturing procedure and the latter testing, several different properties of the blade are captured. From these, the FE model only covers the geometrical deviations such as the chord, thickness, and adhesive geometry deviations. However, mass and stiffness adaption, to meet the measured natural frequencies and masses, or displacements will not be covered, as this would demand a thorough model updating procedure, which would go beyond the scope of this work. Therefore, this section will refer to the geometrical measurements, and the rest will be covered in Sect. 5

The full blade design of the DemoBlade as designed and the manufacturing documentation is available to the authors. In order to allow precise modeling of the DemoBlade as built, laser scanning of the blade mold was carried out to determine the geometry deviations. The derived chord length and absolute thickness distributions for the DemoBlade as designed and as built can be found in Noever-Castelos et al. (2021). Though the manufacturing deviations in the outer geometry are negligibly small (chord length < 10 mm and thickness < 8 mm), they will be considered in the modeling process.

After the full-scale blade tests, the DemoBlade was cut into segments. The masses and the centers of gravity were determined for all blade segments. The respective proce-

cedure will be addressed later in this paper; see Sects. 4.4 and 5.5. Besides the weighing, the geometry was measured thoroughly in each cut cross section in order to guarantee the correct positioning of the shear webs in the FE model and to determine deviations from the design due to manufacturing errors. In particular, the dimensions of the shear web/spar cap adhesive joints on the pressure side of the blade showed significant deviations to the blade design and had to be adjusted in the FE model. Figure 5, for instance, shows the cut at a radial position of 5.2 m ($r_{\text{norm}} = 26\%$). On the suction side (bottom) we see a nice, very thin, and over-laminated shear web/spar cap bonding. However, on the pressure side (top) the shear web/spar cap bonding (which was the blind bond, marked in red) is much thicker than specified in the design. Moreover, there is a lack of adhesive in large portions of the blade, so that the shear web flanges were not covered entirely by adhesive material. This was actually found throughout the whole blade, where the thickness varied between 20 and 30 mm. The design defined a thickness of 9 ± 3 mm. Noever-Castelos et al. (2021) contains all the measured dimensions of the pressure side web adhesive.

In the FE model, we apply concentrated and line-distributed additional masses to cover any type of add-ons installed on the blade such as the lightning protection cable or reflectors of an optical sensor system. Noever-Castelos et al. (2021) include a table with all additional masses and the respective modeling methods. Furthermore, MoCA predefines node positions in the blade that correspond to strain gauges installed on the blade. These are documented in Haller and Noever-Castelos (2021). They allow for accurate and easy extraction of strain results at the correct positions.

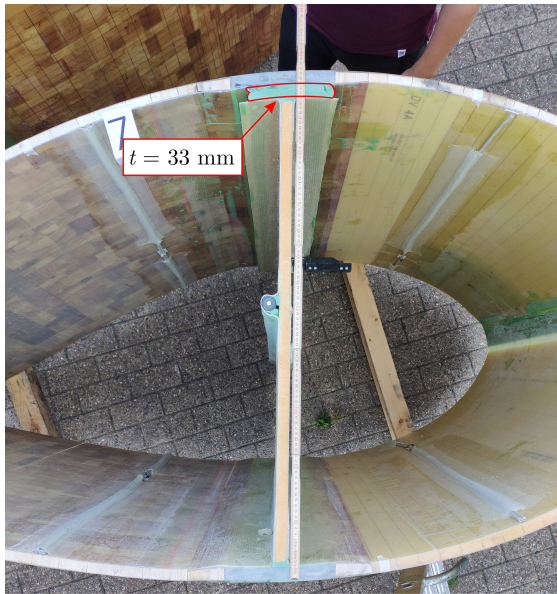


Figure 5. Cut cross section at a radial position of 5.2 m ($r_{26\%}$) with an erroneous manufactured shear web/spar cap adhesive joint on the pressure side of the blade. The width should cover the complete web flange, and the designed thickness is 9 ± 3 mm; however, the real thickness is measured to be 33 mm.

A mesh convergence study was performed in advance to ensure a satisfying mesh density. As stated before, the purpose is to primarily validate the global blade behavior and only secondarily the local response. Therefore, no local mesh refinement will be performed, but the overall mesh density should yield acceptable convergence even at local level. Taking this into account, the convergence was first based on the global blade response in terms of total mass, center of gravity, tip deflection, and the first 10 natural frequencies. Second, the nodal strain results are examined for convergence at several positions covering the whole blade. The element dimensions are halved each step. At the finally chosen mesh size the deviations to the next step are for global responses less than 1.5% and for strains less than 2.1μ strains. It has to be stated that for exact local strain measurements a modeling approach with solid shells or layered solid elements is required to replicate the correct and detailed geometry of the structure. The resulting base model of the DemoBlade consists of 77 693 elements and 71 781 nodes. A total of 71 016 four-noded shell elements (SHELL181 elements in Ansys), with offset nodes on the outer blade surface, represent the composite components, and 6260 eight-noded solid elements (SOLID185 elements in Ansys) model the adhesive joints. Figure 6 depicts a cross-sectional view of the FE model at $r = 8$ m ($r_{\text{norm}} = 40\%$). All other elements are used to model additional masses in the blade. The only boundary conditions

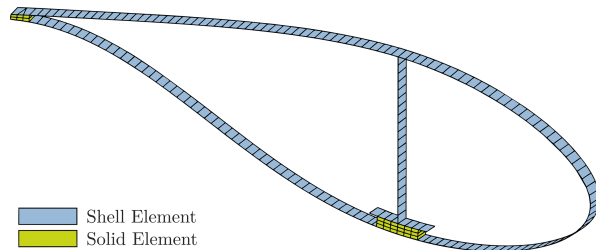


Figure 6. Cross-sectional cut of the FE model at $r = 8$ m ($r_{\text{norm}} = 40\%$).

of the base model are the geometric boundary conditions at the blade root (full clamping as described above).

4 Test description and virtual modeling

Several test configurations of the full-scale blade test were performed to characterize the blade behavior under different load conditions and to prove that the blade design meets all requirements of the certification guidelines (International Electrotechnical Commission, 2014). These configurations are then replicated in the virtual test setup and are described in this section.

4.1 Mass and center of gravity

The first structural characterization considers the blade's mass and center of gravity (CoG). An indoor crane equipped with load cells at every hook lifted two points on each root and tip transport structure as shown in Fig. 7. As the blade remained still and horizontally suspended, the force at each suspension point and its radial position were recorded. After individually weighing the transport structures, the loading chains, and the shackles, the weight was subtracted from the total recorded load at the measurement devices to obtain the total blade mass. Additionally, the mass of the blade bolts was subtracted from the total mass.

The CoG is obtained by calculating the moment equilibrium with the measured loads with respect to a pivot point, in this case the blade root center. This procedure was performed for the z direction (along the span) and y direction (along the chord).

The mass and CoG of the FE model are calculated during every analysis by default and can be extracted directly from the Ansys log file.

4.2 Modal analysis

The experimental modal characterization was carried out by the German Aerospace Center (DLR) for different boundary conditions. The methodology is described briefly in the following. For details please refer to Gundlach and Govers (2019).



Figure 7. Setup for mass and center of gravity measurements.

Free-free boundary conditions were applied after the blade manufacturing by means of elastic suspensions connected to lifting straps. The blade was excited using an impact hammer with soft tip at a total of eight excitation points. Sensors distributed along the blade recorded the deformations, and the mode frequencies and shapes were extracted from the measurements.

The blade was then transported to Fraunhofer Institute for Wind Energy Systems (IWES) and mounted on the test rig. The aim was a second modal characterization with the boundary conditions of the full-scale blade test. Electrodynamical long stroke shakers were employed for the excitation of the blade, and sensor outputs were evaluated for the calculation of the mode frequencies and shapes.

During the FE modal analysis, the boundary conditions are adapted to the different characterization tests. In the free-free configuration, no boundary conditions are applied at all, partially resulting in zero eigenvalues related to rigid body motions. These are not considered in the validation process. For the test rig configuration, the blade root is fully clamped; i.e., all seven degrees of freedom of the shell elements are fixed, for the sake of simplicity. Note that we neglect flexibility of the bolts and the test rig in this way, which we have to keep in mind when evaluating the simulation results.

4.3 Static bending and torsion test configuration

The SmartBlades2 DemoBlade was loaded with extreme loads in four directions. These four load cases correspond to maximum and minimum edgewise loading (MXMAX and MXMIN) as well as maximum and minimum flapwise loading (MYMAX and MYMIN). Furthermore, three static torsion tests were conducted, in which a torsional moment was applied only at one load frame at a time. The tests are referred to as MZLF2, MZLF3, and MZLF4, where LFX indicates the particular load frame, in which the torsional moment was introduced. The static tests provide the necessary information



Figure 8. Photo of a static blade test configuration in flapwise direction.

on the structural blade behavior required to validate the virtual model and test setup.

The tests were performed in the facilities of Fraunhofer IWES. The experimental quasi-static loading of the blade is accomplished with a series of horizontally mounted hydraulic cylinders. These are connected to the load cells via cables which are attached to the load frames mounted on the rotor blade. Each cable runs through pulleys that are mounted on the floor and redirect the forces from a horizontal to a vertical orientation. By attaching the load cells to the load frames (load point), the actual load applied to the rotor blade is measured, and friction and weight of the loading cables do not affect the measurements. The general test setup is shown in Fig. 8.

In the following, some general information is given that is valid for all test setups. The test block angle (cone angle) is 7.5° upwards. The coordinate system referred to in this paper has its origin in the center of the blade root. The y axis is facing vertically upwards, the z axis points horizontally from the origin towards the blade tip (parallel to the floor, not to the pitch axis), and the x axis follows from the right-hand rule (pointing left watching towards the tip). After turning the blade to the correct position and waiting for a static state, the signals of the load cells and the strain gauges are reset to zero. In the virtual test this is achieved by activating gravity, extracting the deformed nodal coordinates, and taking these as the undeformed and stress-free state for the load tests. Gravity is thus not applied in the further analysis, and the nodal displacements are virtually reset to zero so that it is easier to post-process the results. Preliminary verification showed that the corresponding error in the displacement is less than 0.5 % with respect to a simulation that accounts for gravitation throughout the whole simulation.

In the tests, four steel load frames with wooden inlays that follow the blade shape at the respective spanwise positions are used to introduce the loads; see Haller and Noever-Castelos (2021). In the following, we refer to the load frames (LF) as LF1 ($r = 6.7$ m, $r_{\text{norm}} = 33.5\%$), LF2 ($r = 9.7$ m, $r_{\text{norm}} = 48.5\%$), LF3 ($r = 14.0$ m, $r_{\text{norm}} = 70.0\%$), and LF4 ($r = 17.7$ m, $r_{\text{norm}} = 88.5\%$), where r denotes the spanwise position along the blade. Depending on the test setup, not all load frames are installed. Please refer to Noever-Castelos et al. (2021) to find an overview of all test setups. Each load frame is equipped with two eye bolts to attach the load cables. These bolts are roughly positioned at the shear center position in the blade's cross section to avoid unintended torsional loads. Detailed information on the load frames, such as mass, center of gravity, and the corresponding shear center position in the blade's cross section, is given in Haller and Noever-Castelos (2021).

The test setup is equipped with two different kinds of displacement measurements, an optical displacement measurement system and draw-wire sensors (DWSs). For the model validation in this paper, the DWS signals are considered. Using LINK11 elements in Ansys provides a simple and exact model of the draw wires by defining the attachment points only. The deformation measured by the DWS is then modeled by the element length variations of the link elements.

All necessary sensor positions (SP) and load introduction points (LP) on the load frames for the different test setups can be found in Haller and Noever-Castelos (2021). At each load frame position, either with or without a load frame installed, two DWSs are attached. One is connected to a point closest to the front bottom corner, i.e., negative y direction, and one at the rear bottom corner, i.e., positive y direction, of the load frames, or blade shells in case no load frame is installed. These two DWSs will be referred to as front and rear DWS in the following. At the blade tip, one DWS is attached and referred to as the tip DWS. Note that during several load cases, one or the other load frame is not applied due to the setup design; thus the respective DWSs have to be attached directly to the blade shell.

The angle between the loading cable and the blade axis can be adjusted in the experiment by changing the pulley block location within a discrete set of fixing points on the floor. Prior to the test setup, the optimal position for each pulley was determined based on the predicted blade deformation and the desired loading cable angle. The applied loads should be aligned to the load frame planes in the most deformed configuration. The DWS floor attachment and pulley block positions are specified for each test setup individually.

In addition to the DWS and the optical measurement system, several cross sections along the blade are equipped with strain gauges; see Haller and Noever-Castelos (2021). The cross sections at $r = 5$ m ($r_{\text{norm}} = 25\%$) and $r = 8$ m ($r_{\text{norm}} = 40\%$) are instrumented with strain gauge rosettes (biaxial strain gauges) with $0^\circ/90^\circ$ and $\pm 45^\circ$ orientations. Figures B1 and B3 depict the distributions, respectively. The

angles 0 and 90° denote the spanwise and the cross-section-wise direction, where $\pm 45^\circ$ is defined accordingly. The $0^\circ/90^\circ$ rosettes are positioned every approx. 250–300 mm along the shell circumference. The $\pm 45^\circ$ rosettes are located at each web position as well as the leading and trailing edges. Details on strain gauge positions can be found in Haller and Noever-Castelos (2021).

All load cases have the same basic experimental procedure. They were designed to ensure that the actual test matches the specification requirements as closely as possible. Prior to each load case, the rotor blade is rotated to the desired position and mounted to the test stand (with the aforementioned 7.5° cone angle). The load cable pulley blocks are fixed to the appropriate fixation points on the floor. The load cells are installed between the load frames and the loading cables and are then connected to the data acquisition system. Each of the DWSs is attached to the blade. The DWS base is positioned so that the wires run perpendicular to the floor. Finally, the loading cables are connected to the hydraulic cylinders.

The tests are then executed in the following order:

1. check functionality of load cells and displacement sensors;
2. compensate for load cell and strain gauge measurements (reset to zero);
3. start data acquisition;
4. ramp up loads until 100 % of the target load, pausing at 40 %, 60 %, and 80 % partial loads for 10 s each;
5. ramp down loads, pausing at the same load fractions as during ramp-up;
6. stop data acquisition and save measurement data to log file.

The process is similar in the simulation. Starting from the base model, which does not have a cone angle and has the blade positioned with the trailing edge pointing upwards, the steps are as follows:

1. install necessary load frames;
2. rotate blade around z axis to desired position;
3. include cone angle of test rig (incline the blade by 7.5° upwards around x axis);
4. apply gravity and extract new nodal coordinates;
5. replace old nodal coordinates by the extracted new nodal coordinates (equal to resetting sensors to zero);
6. apply and ramp up loads onto the LINK11 elements acting as loading cables;

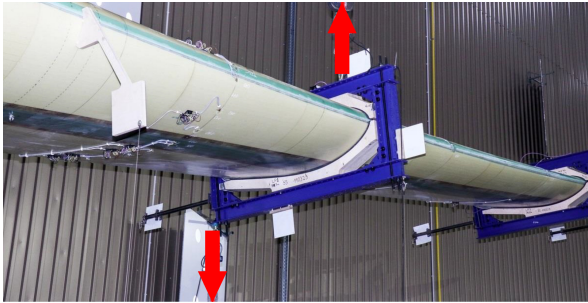


Figure 9. Configuration example of a static torsional loading on the blade with marked-up and downward-facing forces.

7. extract element length variation of the LINK11 elements acting as DWS for 40 %, 60 %, 80 %, and 100 % of the target load.

All individual setups for the simulation with modifications to the base model, all necessary load frames, load points, sensor positions, and forces, and the corresponding ground positions of the pulley blocks and the DWS attachments are summarized in Haller and Noever-Castelos (2021). The ground position coordinates are given in the blade coordinate system of the base model (no cone angle or rotation) described above at the beginning of this subsection.

In other torsion tests, e.g., Tiedemann and Chen (2021), a load is applied on a lever to introduce a combined torsion and flapwise bending, with a subsequent test loading with pure flapwise bending only. The torsional deformation can then be found by subtracting the flapwise motion from the combined motion. However, our test setup follows the idea of introducing pure torsion by suspending the load frame approximately at the shear center location of the blade cross section and inducing torsion by an offset load as shown in Fig. 9.

Because the blade is still mounted at a block angle of 7.5° the axis of the torsional moment is not fully aligned with the pitch axis, as the forces do not act exactly in the cross-sectional plane. The load cable oriented upwards was attached to a ceiling crane and to the load frame at approximately the shear center position. As the ceiling crane location is hard to record but the load rope is perpendicular to the ground, it was assumed that the location is 18 m above ground (y direction, approximately crane height). By this, deflections parallel to the floor, due to load application, would only result in small angle deviation of the perpendicular force. The force facing downwards was applied onto the load frame corner to create the lever with respect to the shear center. Our procedure is similar to a combination of the pure torsion and locked torsion test presented by Berring et al. (2007). However, this method may imply some errors from

1. numerical shear center calculation;

2. not suspending exactly at the cross-sectional shear center but on the frame, which leads to an offset of the suspending force when the blade is twisted and thus an induced counteracting torsion;
3. no exact perpendicular downward-facing force; and
4. inclination of the blade.

Regarding point 2, the offset of the load application point of the suspending cable from the numerically calculated shear center after twisting the blade yields 0.9 %, 3.0 %, and 5.3 % of the respective lever for the downward-facing force on LF2, LF3, and LF4, respectively. Theoretically, expecting a similar force pulling upwards as downwards, the induced torsion is reduced by the same relative values for the respective load cases MZLF2, MZLF3, and MZLF4.

The magnitude of the induced torsion was designed to be the maximum allowable torsion (respecting safety margins) at the particular cross section rather than a possible bend-twist-induced torsion magnitude. This was motivated by the idea of reducing any relative measurement error when proofing at the maximum allowable torsion, i.e., the maximum deformation.

Nevertheless, overall the aforementioned errors do not practically affect the validation of the model, as all DWS and load cables are modeled as LINK11 elements, with all attachment points modeled at their correct global location of the test setup. This ensures that the forces and displacement measurement direction is always correct throughout the test, all under the assumption that the model behaves the same as the real blade. Thus, no corrections of any kind to measurements or FE results were applied.

4.4 Blade segment mass and center of gravity measurement

After finishing the full blade tests, as discussed in Sect. 4.3, the blade was cut into 17 segments for further characterization. Figure 5 shows a cut surface of the seventh segment at a spanwise position of $r = 5.2$ m. To determine the 3D center of gravity (CoG), the segment was suspended at one point with a flexible rope, so that the CoG settled exactly underneath this point (like a pendulum). Hence, the vector in direction of the suspension rope defines an axis on which the CoG must be located (CoG axis). This procedure was repeated with different suspension points at least two times. The CoG was then found in the intersection point of the different CoG axes. The measurement setup can be seen in Fig. 10 as well as a digital representation of the intersection of different CoG axes.

To measure the vectors and analyze the data, an optical measurement system (photogrammetry) was used. Every segment was equipped with several coded and uncoded reflecting marks to obtain the shape of the segment, the suspension points, and a plumb that was used to get the CoG

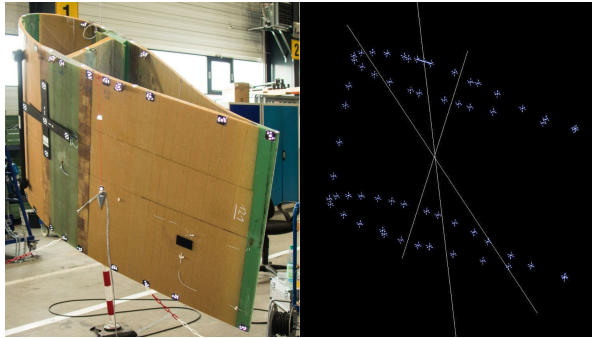


Figure 10. Measurement setup of cross-sectional analysis (left) and extracted vectors in CAD with intersection point defining the center of gravity (right).

axes. All the point clouds were analyzed in Autodesk Inventor and Siemens NX. All segments were aligned in CAD, and the CoG was extracted for each segment with regard to the blade coordinate system. In this way we obtained the distribution of the segment CoGs along the blade.

Considering the model validation, MoCA is able to generate the respective segments at their correct positions in the blade, so the segment masses and CoGs are a natural output of Ansys.

5 Comparison of experimental and simulation results

In this section, we compare the experimental results with the simulations. The observation scale will continuously decrease from a global to a more local scale. We start with the global blade characteristics such as natural frequencies, total mass, and global center of gravity. These give a rough estimate of the modeling correctness. Then the blade deformations by means of bending and twist distributions during the static extreme load tests will be analyzed. Finally, the strain levels in two cross sections during the extreme load tests and the masses and centers of gravity of the cut blade segments are compared, which give a more detailed view on a local scale.

In order to rate the validity of the model, it is necessary to identify specific thresholds. However, these are hardly found in the literature, especially as different applications and fidelity levels may demand other thresholds. For example, Safarian (2015) reports validation requirements for finite-element analysis according to the Federal Aviation Regulations of the US government, where a displacement deviation of $< 5\%$ between the simulation and experiment is typically acceptable for global effects, and local effects measured in the form of strains allow for a maximum of 10% deviation, whereas strains exceeding these values require a re-evaluation of the model. These regulations refer to avia-

Table 1. Comparison of the total mass and the center of gravity (CoG).

	Experiment (in kg)	Uncertainty (in kg)	MoCA (in kg)	Difference (in kg)
Mass	1793	45	1673.5	-115.5
	Experiment (in m)	Uncertainty (in m)	MoCA (in m)	Difference (in m)
y	0.10	0.04	0.10	0.00
z	6.58	0.20	6.35	0.23

tion applications, which also apply finite-element shell models for the analysis comparable to our use case. Therefore, we will apply a 5% threshold for global displacements, whereas a 10% threshold will be applied on the cross-sectional strain results. These margins should also cover measurement uncertainties, as the DWS and the strain gauges offer a quite narrow uncertainty band, 0.6% and 2% , respectively. Thresholds for masses are harder to define as these depend on the measurement setup, in our case with up to 2.5% uncertainty. In addition, not all additional masses were correctly documented, and thus they were not modeled. The same problem holds for natural frequencies ω : following $\omega = \sqrt{\frac{k}{m}}$ and respecting unknown mass variation and typically a maximum of 5% material tolerances (including density and stiffness according to private communication with manufacturers), it is also hard to define thresholds for the frequencies. Therefore, both mass and frequencies will be discussed individually.

5.1 Blade mass, center of gravity, and eigenfrequencies

Table 1 lists the total blade mass and the location of the center of gravity in longitudinal (z) and chord direction (y) as well as the measurement uncertainties and the deviation of the numerical model. We see that the model from MoCA is 115.5 kg lighter than the real blade, which corresponds to a 6.44% relative difference related to the measurement. In contrast, the measurement uncertainty is 45 kg . The mass difference is likely due to manufacturing deviations and/or additional masses (e.g., sensor wires and installations) that have not been considered in the numerical model. The location of the CoG matches perfectly in the chord direction, i.e., with a precision to the nearest two decimal places. There is only little deviation of 230 mm in the spanwise direction, which is almost within the measurement uncertainty range of $\pm 200\text{ mm}$. All measurement uncertainties are based on given sensor uncertainties and taking the worst case scenario in the combination of those.

The results of the modal analysis, both experimental and numerical, are listed in Table 2. The experimental results are taken from Gundlach and Govers (2019). The flapwise frequencies are in acceptable agreement with deviations of less

Table 2. Comparison of the modal analyses for the free–free (top) and the test rig (bottom) configuration. Experimental results are taken from Gundlach and Govers (2019).

Mode	Experiment (in Hz)	MoCA (in Hz)	Difference	
			(in Hz)	(in %)
free–free				
1st flapwise	4.8	5.08	0.28	5.83 %
1st edgewise	10.1	9.74	−0.36	−3.56 %
1st torsion	16.9	15.95	−0.95	−5.62 %
Mode	Experiment (in Hz)	MoCA (in Hz)	Difference	
test rig			(in Hz)	(in %)
1st flapwise	2.2	2.37	0.17	7.73 %
2nd flapwise	6.8	7.34	0.54	7.94 %
1st edgewise	3.1	3.25	0.15	4.84 %
2nd edgewise	10.9	10.81	−0.09	−0.83 %
1st torsion	18.7	16.50	−2.20	−11.76 %

than 8 %. The largest deviation in flapwise modes is found for the second mode in the test rig configuration (7.94 %, which corresponds to an absolute deviation of 0.54 Hz). The smallest deviation can be observed for the first flapwise mode in the free–free configuration, which is 5.83 % or 0.28 Hz, respectively. In edgewise direction, the approximation is even better. The largest relative deviation is seen for the first edgewise mode in the test rig configuration, which is 4.84 % (or 0.15 Hz in absolute numbers). The second edgewise mode is only 0.83 % (or 0.09 Hz in absolute numbers) smaller in the simulation compared to the experiment in the test rig configuration, which is an excellent agreement. The largest absolute deviation is present in the free–free configuration, where the first edgewise mode is 0.36 Hz lower than the measured value. Anyways, the deviation of the edgewise modes is less than 5 % in all cases, which is a very good agreement. The first torsion mode is quite well approximated in the free–free configuration, where the simulation is 5.62 % lower than the experiment. However, in the test rig configuration the deviation is −11.76 % (more than 2 Hz less compared to the test), which is relatively high. In general, the simulations agree better with the test results in the free–free configuration than in the test rig configuration. This is likely due to the rigid representation of the test rig and the connection bolts, as already mentioned in Sect. 4.2. Similar deviation ranges of the natural frequencies can be found in Knebusch et al. (2020) for the same blade, but with a different model, with errors between 1.8 % and 9.7 % for flapwise and edgewise modes and up to 22 % for the torsion mode.

5.2 Static bending tests

The results of the static bending tests will be illustrated by means of deflection lines. For each test setup, two lines exist: one for the front and one for the rear DWS. The deflections

in the front DWS are plotted in Fig. 11 for each pausing load during ramp-up (40 %, 60 %, 80 %, and 100 % of the target load as described in Sect. 4.3). The plots for the rear DWS are added in Appendix A. A table is added in each of the figures that show the differences between the simulations and the tests (in absolute and relative numbers). The tip DWS values are the same for the rear and the front DWS, as only one DWS is installed at the blade tip.

Figure 11a shows the result of the front DWS during the MXMAX load case. For this scenario a maximum deflection of 180 mm at the blade tip is reached. The simulation shows excellent agreement for the front DWS sensors, with a maximum absolute difference of −2.3 mm at the tip for 100 % load and a maximum relative difference of −4.0 % at LF1, whereas the deviations in all other positions are well below 2.0 %. The rear DWS results shown in Fig. A1a in Appendix A have slightly higher errors with a maximum of −5.5 % at LF1 for full load.

For the load case MXMIN, Fig. 11b illustrates the front DWS results. Except for LF1, the results are in very good agreement with a maximum deflection error of −1.6 % at LF2 at full load. However, the results in LF1 return maximum errors of 3.8 % at 40 % load, which decreases to 1.8 % at full load. Similar behavior is found for the rear DWS (Fig. A1b); excluding LF1 the maximum error is 1.7 % in LF3 and the tip during 40 % load.

The results of the front DWS during the maximum flapwise setup (MYMAX, Fig. 11c) are in very good agreement, when excluding the LF1 data. The LF1 results tend to show the highest errors. This might be due to the smallest absolute deflection values, as a systematic sensor/measurement inaccuracy will have a higher impact on relative errors. Concerning the other load frames the maximum error is found to be −2.6 % for the LF4 DWS at full load, which corresponds to −22.4 mm deflection error at a maximum deflection of 875 mm in the experiment. All other values range between −0.9 % and −2.4 %. The excluded LF1 results show higher errors of up to 9.0 % for 60 % load. For the rear DWS (Fig. A1c), though excluding LF1 (maximum error −17.6 %), the LF2 results show errors above 6.7 % with the highest reaching −8.8 % during full load. For the other two load frames the errors are low again and are between −0.9 % and −2.4 %. If taking a closer look at the LF2 full load deflection d in the test and experiment, the front DWS shows $d_{\text{Exp}, f} = 165$ mm and $d_{\text{Sim}, f} = 161$ mm, whereas the rear DWS returns $d_{\text{Exp}, r} = 175$ mm and $d_{\text{Sim}, r} = 160$ mm. That means the overall deflection of the simulation is less than in the experiment, but the difference between rear and front is $\Delta d_{\text{Exp}} = d_{\text{Exp}, r} - d_{\text{Exp}, f} = 10$ mm and $\Delta d_{\text{Sim}} = -1$ mm; i.e., the simulation shows a positive rotation around the z axis, while the experiment returns a much higher negative rotation. The rotation angle Θ can be calculated by the relationship

$$\Theta = \arcsin\left(\frac{\Delta d}{l_{\text{SP}}}\right), \quad (1)$$

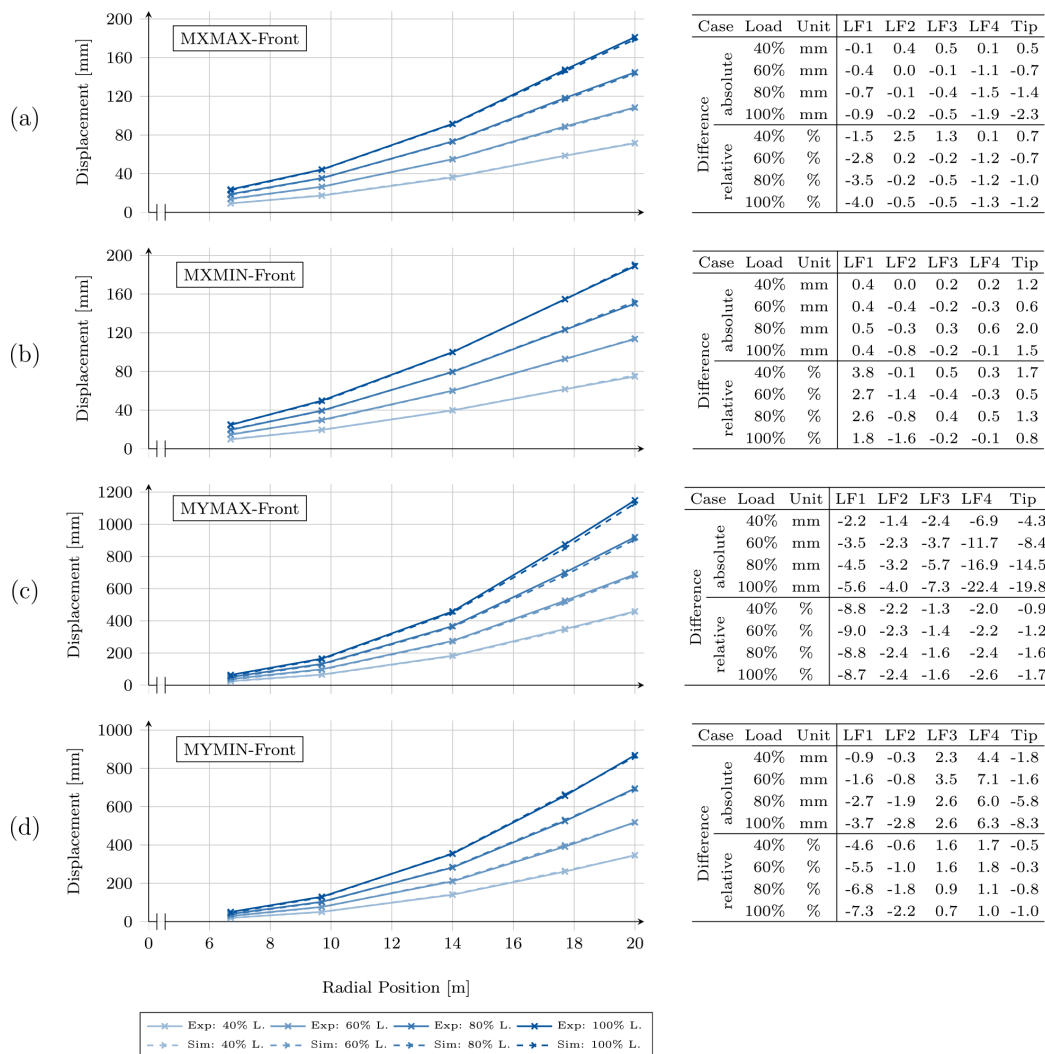


Figure 11. Bending lines extracted from the front draw-wire sensor for the (a) MXMAX, (b) MXMIN, (c) MYMAX, and (d) MYMIN experiment and simulation. Results are shown for 40 %, 60 %, 80 %, and 100 % of the target load. The table on the right shows the differences between the simulation and the test.

where l_{SP} is the distance of both front and rear DWS attachment points on the load frame. The rotation angle becomes $\Theta_{Exp, LF2} = -0.268^\circ$ in the experiment and $\Theta_{Sim, LF2} = 0.042^\circ$ in the simulation. Assuming the pivot point is at the shear center (SC), a correction could be calculated to see if the bad results of the rear DWS at LF2 are due to the wrong predicted rotation along the z axis. All necessary geometric data can be found in Haller and Noever-Castelos (2021). Following Eq. (1) and using the distance of the front or rear DWS attachment to the shear center, the front absolute difference during 100 % loading is increased to -7.87 mm, which results in an error of -4.8 %, and the rear deflection is reduced to -7.7 mm or an error of -4.4 %. By this correc-

tion due to a wrong predicted rotation angle the rear DWS approximation improves by 4 %, while the accuracy of the front sensor decreases by only 2.4 %. This correction is introduced to evaluate the accuracy for the bending prediction and only holds for the LF2 position, as the other positions have different rotation angle deviations. Additionally, it has to be noted that during the flapwise loading the DWS attachment distances to the shear center are much higher than for the edgewise loading; i.e., the influence from rotation angle deviations is amplified significantly.

Figure 11d shows the front DWS results comparison during the minimum flapwise loading scenario (MYMIN). All load frames are installed and thus can be evaluated, and the

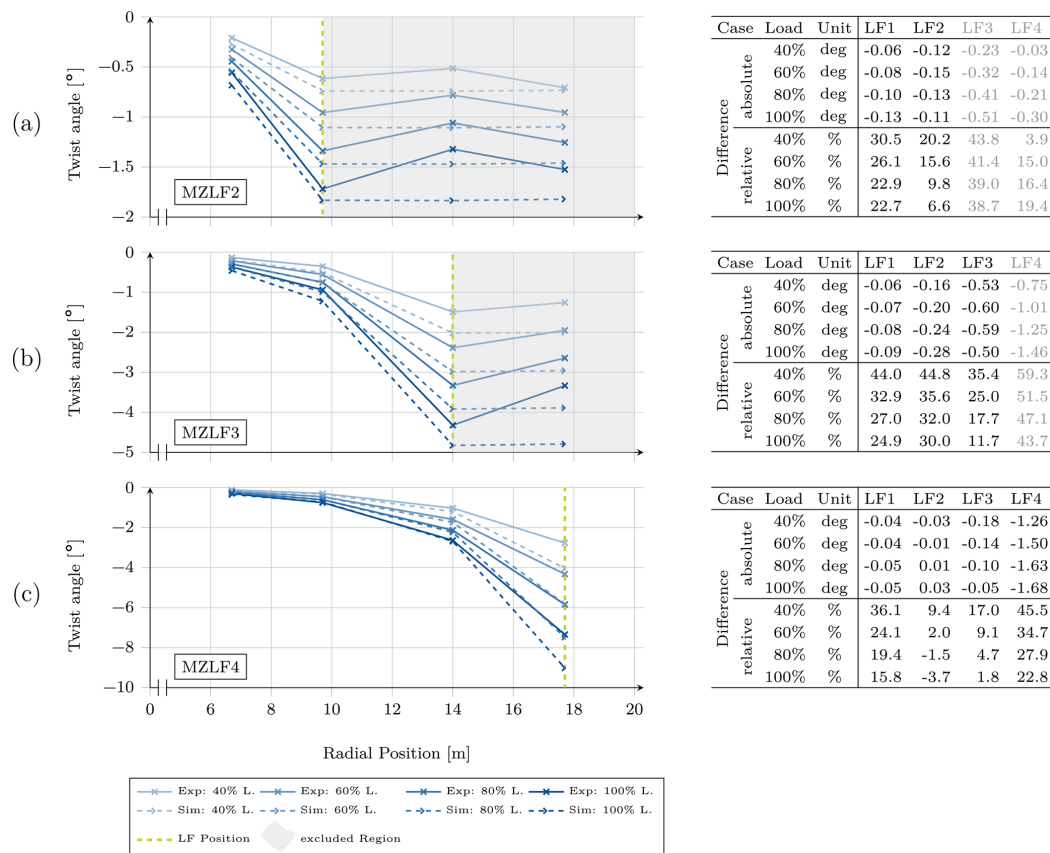


Figure 12. Twist angles calculated from the draw-wire sensor results for the (a) LF2, (b) LF3, and (c) LF4 torsional loading experiment and simulation. Results are shown for 40 %, 60 %, 80 %, and 100 % of the target load. The table on the right shows the differences between the simulation and the test.

results show a very good agreement with errors below 2.2 % for all DWS, except LF1. At this first load frame, again the results have significantly higher errors of up to -7.3% at full load. Figure A1d contains the rear DWS result of the MYMIN load case and lists throughout higher deviations of up to -13% for the LF1 sensor. Here, again, by analyzing the rotation behavior of the blade along the z axis all load frames show significant rotation differences and after estimating a correction, e.g., the accuracy of the LF1 front sensor, would decrease to a deviation of -11% , while that of the rear sensor increases to -10.4% . This is the worst approximation of the simulation for the static extreme load bending setups. The other load frames are in very good agreement, with most of the results (excluding LF1) being below the 5 % threshold defined at the beginning of Sect. 5.

5.3 Static torsion tests

Full-scale blade tests in pure torsion are usually not included in certification processes according to International

Electrotechnical Commission (2014) and are thus rarely available. As described in Sect. 4.3 the blade is twisted during three different setups successively at the load frames LF2, LF3, and LF4. The results of the tests and the simulations are plotted in Fig. 12. The structural behavior behind the actual loaded frame position to the tip will not be addressed in this paper and is highlighted as gray-colored areas, as the areas loaded in torsion are located between the root and the respective load frame. However, the raw results similar to the static bending experiments are the DWS length variation; for these torsional experiments the more relevant twist angles are calculated and plotted according to Eq. (1). Figure 12a shows the first torsional test loaded at LF2. The absolute angle deviation from experiment to simulation is between -0.06° and -0.15° but yields high relative deviation up to 30 % due to the small twist angles of -0.55° at LF1 and -1.72° at LF2 during 100 % load.

Moving the load application to LF3 (Fig. 12b) does not change the situation. At the load application position the absolute error is high, with up to -0.6° at a maximum twisting

of -4.3° . All errors exceed -10% dramatically. However, the experiment with torsional loading on LF4 (see Fig. 12c) shows reasonably good results for the twist angle at LF2 and LF3 with angle deviations of 3.7% and 1.8% , respectively. The results at LF4 where the load is applied and which shows the highest twist angle keep high deviations of about 20% for full load. Such high errors during torsional loading may be based on the shell element with a node offset to the exterior surface used for this model. Greaves and Langston (2021), Branner et al. (2007), Pardo and Branner (2005), and especially Laird et al. (2005) already stressed the high inaccuracy of shell elements with node offsets from the mid-plane to predict the structural behavior of hollow structures subjected to torsional loading. However, the twisting is generally overestimated throughout the three torsional tests, which is in line with the aforementioned references.

5.4 Local strain comparison

As stated in Sect. 4.3 the highly instrumented cross sections at $r = 5\text{ m}$ and $r = 8\text{ m}$ offer a more detailed view on the local strain levels in the rotor blade. The strain results are used to compare the simulations with the tests and to verify that local effects are correctly reproduced. We have selected a few representative load cases in this section. The remaining load cases can be found in Appendix B.

In Fig. 13a the strain in 0° (spanwise direction, in blue) and 90° (crosswise, in yellow) directions for the MXMIN simulation (solid lines) and experiment (circles) are plotted over the normalized airfoil circumference (denoted by S) for the 5 m cross section, starting at the suction side trailing edge ($S = 0$), moving along the suction side to the leading edge ($S \approx 0.5$), and then along the pressure side to the pressure side trailing edge ($S = 1$). This cross section shows the following general characteristics in all load cases.

1. In the simulation at $S = 0$, there is a strain peak in the 90° direction, because the sandwich core material vanishes suddenly towards the trailing edge, due to the shell elements and their missing capability of tapering single materials in their sections as done in the real layup.
2. In the simulation at $S = 0-0.25$, there is an excessive or wrong curvature in the 90° strain curve, for which we do not have a feasible explanation.
3. In the simulation at $S = 0.25-0.35$, there is a stepped dip or raise of the 90° strain, because the sandwich core material is substituted by core and unidirectional (UD) layers and then completely by the UD spar cap and vice versa.
4. In the simulation at $S = 0.5$, there is a strain peak in the 90° direction, because the sandwich core material vanishes around the leading edge.
5. In the experiment at $S = 0.5-0.65$, there is a strain dip in the 0° direction, for which we do not have a feasible explanation. The structure should be symmetric next to the leading edge.
6. In the simulation at $S = 1$, there is a strain peak in the 90° direction, because the sandwich core material vanishes towards the trailing edge.

Apart from the unclear dip around the suction side leading edge panel ($S = 0.5-0.65$), the longitudinal strain (in 0° direction) differs along the circumference in mean only about $\pm 107\ \mu\text{m m}^{-1}$. This is about 13.3% related to the maximum measured absolute strain of $811\ \mu\text{m m}^{-1}$. However, the crosswise strains (in 90° direction) reach deviations of up to $\pm 90\ \mu\text{m m}^{-1}$ in mean, which corresponds to about 26.2% related to its measured maximum. The MXMAX results (Fig. B2a) are slightly better concerning mean strain errors, with 12.4% for the 0° direction and 20.3% for the 90° direction.

Figure 13b shows the MYMAX load case. Unlike the edgewise case a failure of the strain gauge at $S = 0.3$ was recorded in the experiment, which can be seen in the discontinuity of the experimental results. The flapwise bending of the blade in general is more excessive compared to the edgewise bending and provokes the highest longitudinal strains in the spar cap positions reaching maximum values of up to $1800\ \mu\text{m m}^{-1}$ in the outer shell layer. Consequently the crosswise strain also increases with absolute mean errors to $\pm 208\ \mu\text{m m}^{-1}$ (11.6%) in 0° direction and $\pm 217\ \mu\text{m m}^{-1}$ (36.0%) in 90° direction, both approximately twice as much as in the edgewise load case. All other aforementioned issues are also present here, some more and some less pronounced. The same conclusion also holds for the MYMIN case in Fig. B2b, though the mean error values are lower, due to smaller load sets. In 0° direction a mean error of 8.2% was calculated and 15.8% in 90° direction.

Taking a look at the torsion tests, in particular for the MZLF3 load case plotted in Fig. 13c, the longitudinal strain shows a relatively good agreement with the test, except for $S = 0.5-0.65$ and at the pressure side trailing edge panel ($S = 0.85-1$). The crosswise strain shows partially good agreement with the experiments, except for the aforementioned characteristics, which are more dominant than in the bending tests; e.g., the peaks at the trailing edge are more pronounced. As for the longitudinal strain, the crosswise strain shows a disagreement between simulation and experimental results, which is even stronger due to a shifted curvature in the plot. These can also be seen during the remaining two torsion tests. The MZLF4 load case in Fig. B2d is very similar to the MZLF3 load case, whereas the MZLF2 load case (Fig. B2c) shows all of the stated characteristics in a more pronounced manner as the load introduction is shifted closer to the evaluated section at $r = 5\text{ m}$.

The next highly equipped cross section is at $r = 8\text{ m}$. While the previous cross section was located at maximum

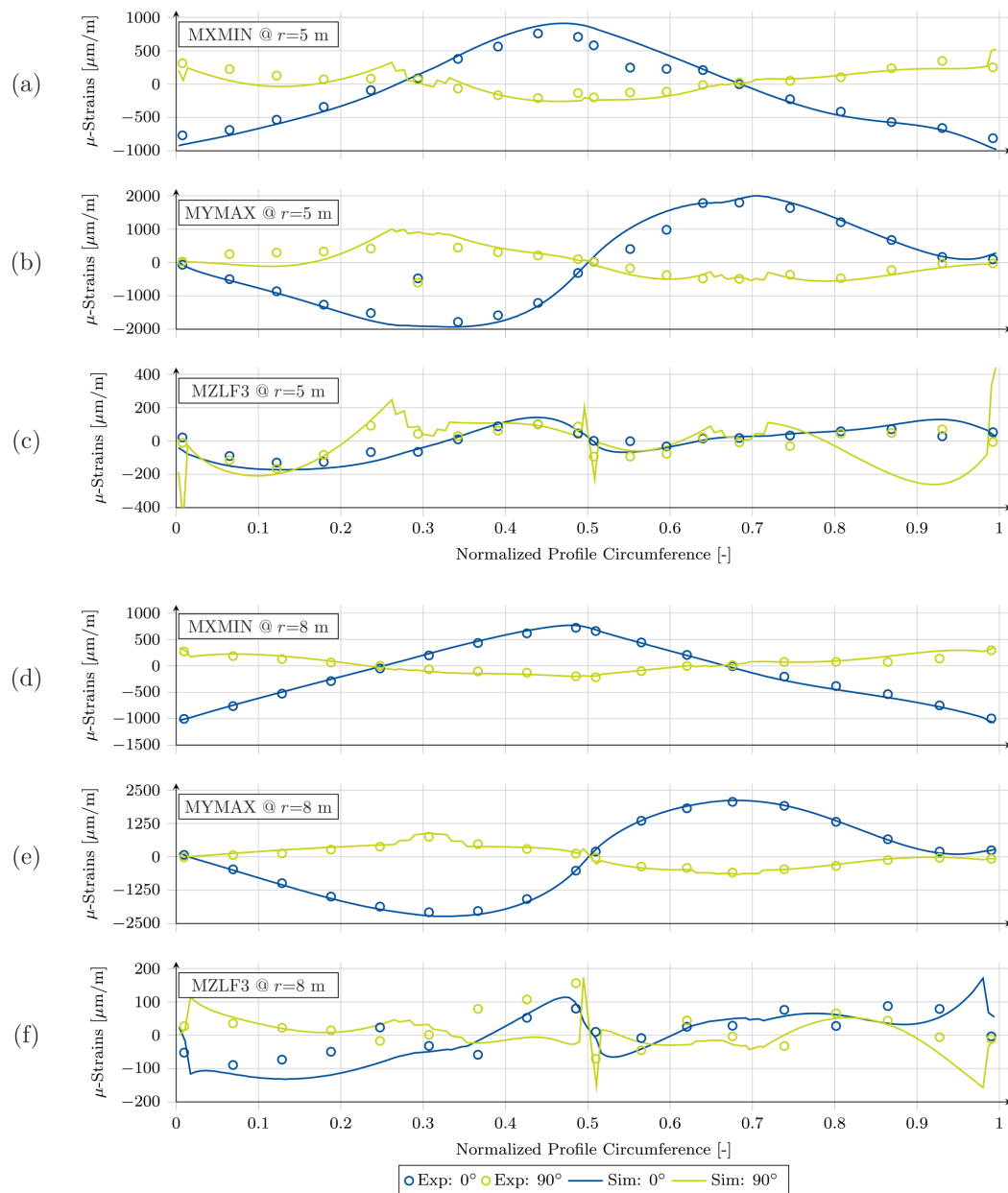


Figure 13. Spanwise and crosswise strains of the simulation and the test, plotted against the normalized profile circumference of the cross section at $r = 5$ m for the (a) MXMIN, (b) MYMAX, and (c) MZLF3 load case and at $r = 8$ m for the (d) MXMIN, (e) MYMAX, and (f) MZLF3 load case.

chord, this one is already in a region where geometric curvatures are smoother. For direct comparison the same three load cases were selected for this cross section. As depicted in Fig. 13d the longitudinal and the crosswise strains during the MXMIN test follow very well the experimental results, both qualitatively and quantitatively. Strain levels are similar to the cross section at $r = 5$ m, but the strain errors of the sim-

ulation compared to the experiments are much lower (mean error $\pm 29 \mu\text{m m}^{-1}$ or 3.0 % in 0° direction and $\pm 32 \mu\text{m m}^{-1}$ or 11.0 % in 90° direction). The same holds for the MXMAX loading (see Fig. B4a), where the mean strain error is even between 2.0 % and 9.5 %, respectively. Although these are not very pronounced, the peaks at the trailing and leading

Table 3. Segment mass and center of gravity (CoG) differences between experiment and simulation. The relative distances of the CoG are given with respect to their corresponding geometrical cross-section parameter, i.e., absolute thickness (X), chord length (Y), and spanwise segment length (Z).

Section no.	r_1 (in m)	r_2 (in m)	Mass		Center of gravity					
			(in kg)	(in %)	X (in m)	Y (in m)	Z (in m)	X (in %)	Y (in %)	Z (in %)
1	0.0	0.9	34.6	9.8 %	-0.030	0.000	0.003	-2 %	0 %	0 %
2	0.9	2.0	-7.36	-5.1 %	-0.003	0.009	0.035	0 %	1 %	3 %
3	2.0	3.0	-10.96	-9.3 %	-0.031	-0.004	0.065	-3 %	0 %	6 %
4	3.0	3.5	-4.74	-8.0 %	-0.066	0.000	-0.007	-6 %	0 %	-1 %
5	3.5	4.0	-3.419	-6.1 %	-0.076	-0.005	0.021	-8 %	0 %	4 %
6	4.0	5.2	-7.39	-5.9 %	-0.094	-0.060	0.055	-10 %	-3 %	5 %
7	5.2	6.5	-6.07	-4.9 %	-0.102	-0.036	0.054	-13 %	-2 %	4 %
8	6.5	8.5	-9.81	-5.8 %	-0.074	-0.008	0.071	-12 %	0 %	4 %
9	8.5	9.5	-3.572	-4.8 %	-0.050	0.007	0.040	-10 %	0 %	4 %
10	9.5	10.5	-5.236	-7.3 %	-0.049	0.004	0.132	-11 %	0 %	13 %
11	10.5	11.5	-3.685	-5.4 %	-0.041	-0.005	0.108	-10 %	0 %	11 %
12	11.5	12.5	-4.087	-6.6 %	-0.031	0.003	0.090	-9 %	0 %	9 %
13	12.5	16.0	-18.59	-9.9 %	-0.036	0.007	0.091	-13 %	1 %	3 %
14	16.0	16.5	4.007	16.3 %	-0.003	-0.048	0.128	-1 %	-4 %	26 %
15	16.5	17.5								
16	17.5	19.0	-4.405	-9.1 %	-0.025	0.094	0.195	-15 %	11 %	13 %
17	19.0	20.0	1.104	9.5 %	0.010	0.023	0.041	10 %	4 %	4 %

edges as well as the stepped dips or raises can be identified as consistent characteristics throughout all test setups.

Comparing the results of the MYMAX test depicted in Fig. 13e, the good agreement between the simulation and the test are evident. Even the stepped raise at the two spar caps ($S = 0.3$ and $S = 0.67$) exist in the experimental results. The mean strain error is $\pm 53 \mu\text{m m}^{-1}$ (2.5 %) in 0° direction and $\pm 63 \mu\text{m m}^{-1}$ (8.4 %) in 90° direction, which is much less than for the other cross section, while having slightly higher maximum strain levels of $2080 \mu\text{m m}^{-1}$ in 0° direction and $753 \mu\text{m m}^{-1}$ in 90° direction. This excellent agreement is also found in Fig. B4b for the MYMIN load case.

However, the results from the torsional tests do not agree. As seen in Fig. 13f the simulation results of the longitudinal strain during the MZLF3 test may follow some correct trend of the experiments but has significant differences. The same applies to the crosswise strains. Although the strain errors are in the same range as the bending test results, compared to the absolute strain levels these have the same magnitude as the error. The remaining torsional test results (Fig. B4c and d) show similar problems.

5.5 Segment mass and CoG comparison

In this section, we compare the experimental mass and CoG measurement of each segment with the respective simulation results. Table 3 contains the segment numbers, the segment locations along the blade defined by the spanwise positions of the left and the right cutting sections r_1 and r_2 , respec-

tively, and the differences of the segment masses and the CoG locations (in absolute and relative numbers).

The relative difference of the mass is related to the measured segment mass, and the CoG positions are with respect to the corresponding geometrical mid-cross-sectional dimensions, i.e., absolute thickness (X), chord length (Y), and radial segment length (Z). It was not possible to measure segment 15. The mass differs from -4.8 % to -9.9 % except for segment 1, 14, and 17, where the mass was overestimated. Unfortunately it was not possible to calculate an overall blade mass as one segment result was missing. Concerning the CoG differences, the coordinate in cross-section thickness direction (X) varied up to -15 % but was most of the time predicted closer to the suction side. The CoG location in chord direction (Y) agreed very well with the measurement, except for segment 16, where the variations were below ± 4 %. The radial locations match well for most of the segments (≤ 6 %). However, the sections 10, 11, 12, 14, and 16 resulted in higher variations, predicting the CoG position closer to the tip by more than 10 % of the segment length.

6 Summary and conclusion

The aim of this paper was the validation of a parameterization and modeling methodology for wind turbine rotor blades. This methodology was implemented in the in-house 3D finite-element model generator MoCA, which creates hybrid shell/solid finite-element models.

Full-scale blade tests were performed on the SmartBlades DemoBlade as an experimental reference. The blade has a length of 20 m and is designed with prebend and presweep. The following magnitudes were determined experimentally: the total mass and the center of gravity of the full blade, the mass and center of gravity distributions along the blade by weighing of blade segments, the natural frequencies in a free-free and a clamped cantilever configuration, the deflection curves along the blade for both flapwise and edgewise bending as well as torsion, and the strains in the cross-sectional and longitudinal direction close to the maximum chord position. The governing parameters such as geometry, material layout, manufacturing deviations, additional sensor, and load frame masses were extracted from the blade and test documentations. These were fed into MoCA. Finite-element models for all test setups were created, and the simulations were executed in the commercially available finite-element code Ansys. Then, the simulations were compared with the experimental results.

The mass and center of gravity of the full blade compared very well (error of -6%). The masses and centers of gravity of the blade segments, i.e., the mass and center of gravity distributions along the blade, were also in good agreement (error of 5% – 10%). Modal analysis concluded for the natural frequencies with free-free boundary conditions also well (error $< 6\%$) matching results, and those for the clamped cantilever configuration matched reasonably well (error $< 8\%$ for bending, 11.7% for torsion).

The deflections for bending in edgewise direction were in excellent agreement (error $< 4\%$), while the deflection curve for bending in flapwise direction showed a comparably large deviation of 13% at the root, which decreased substantially towards the tip (error at the tip $< 4\%$). A reason for that was an elastic twist during the test that was not replicated in the simulations. In general, the errors mostly comply with the validation threshold of 5% defined at the beginning.

For both flapwise and edgewise bending the strains in the spanwise direction were in reasonably good agreement, taking into account that no local mesh refinement or global-local modeling strategy was followed. Strain gauges were distributed along the circumference of the cross sections at spanwise positions of 5 and 8 m, respectively, in order to measure the cross-sectional deformations. Especially at a span of 8 m, the authors observed a very good agreement of the simulation and the experiments, with nearly all mean strain errors below the 10% threshold defined for local comparison. The cross section at a span of 5 m produced approximately twice the errors of the 8 m section. However, for both sections some local effects close to the spar caps could not be resolved in the simulations.

During torsion, the authors identified quite large deviations in the global elastic twist distributions along the blade. Also the first torsional natural frequency has the highest discrepancy to the test with -11.76% . During the torsion test the strain measurements showed quite large deviations ex-

ceeding 30% mean errors at 8 m span and reaching up to 295% mean error at 5 m span. Although, the longitudinal strains agreed better than the transverse strains, at least qualitatively. As the literature reports, all this may be traced back to the shell elements being inappropriate to model torsional behavior, due to the offset of the nodes to the element's mid-plane.

Generally speaking, the authors observed good agreement between the simulations and the experiments in almost all situations and global bending observations and acceptable agreement in local observations. The parameterization and modeling methodology can thus be rated as validated, in the capabilities of the proposed modeling technique.

However, the modern flexible blade design, which is driven to its material and structural integrity limits and includes intentional torsion for load alleviation, requires accurate predictions for all load cases in order to be reliable. Looking a step further, fatigue damage calculation especially needs correct strain or stress predictions of the models. The authors currently work on evaluating blade modeling by means of solid elements and/or solid shell elements. Although we loose computational efficiency of the shell element models, this way the accuracy in torsional response should be improved significantly. Additionally, the correct representation of geometrical shape and 3D tapering can be realized. This should shed light on the discrepancy in torsion and some of the bending load cases, where we were unable to identify their origin, for instance wrong curvatures in the strain distributions or numerical steps/peaks at material tapering. However, such very local effects as material discontinuities and numerical strain/stress peaks probably require a global-local modeling approach to capture every smaller-scaled detail. Subsequently, a sensitivity study of relevant geometry, material, or modeling parameters can further enhance the understanding of local inaccuracies.

Appendix A: Static bending test results

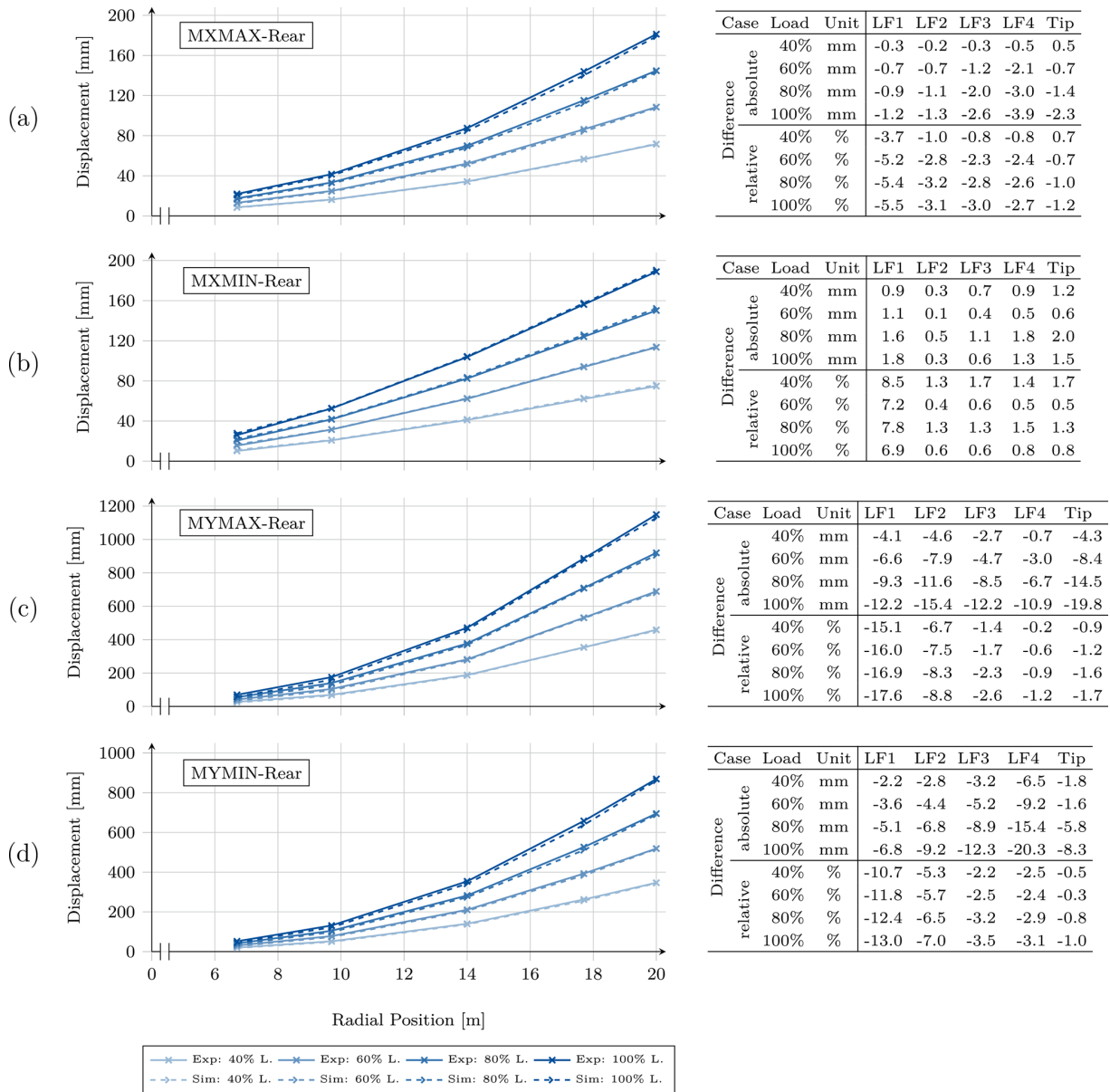


Figure A1. Bending lines extracted from the rear draw-wire sensor for the (a) MXMAX, (b) MXMIN, (c) MYMAX, and (d) MYMIN experiment and simulation. Results are shown for 40%, 60%, 80%, and 100% of the target load. The table on the right shows the differences between the simulation and the test.

Appendix B: Local strain comparison

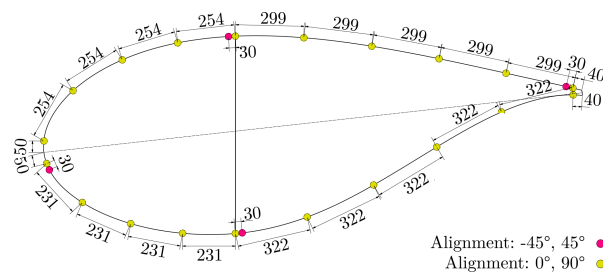


Figure B1. Cross-sectional sensor distribution at $r = 5$ m ($r_{\text{norm}} = 25\%$) (Haller and Noever-Castelos, 2021).

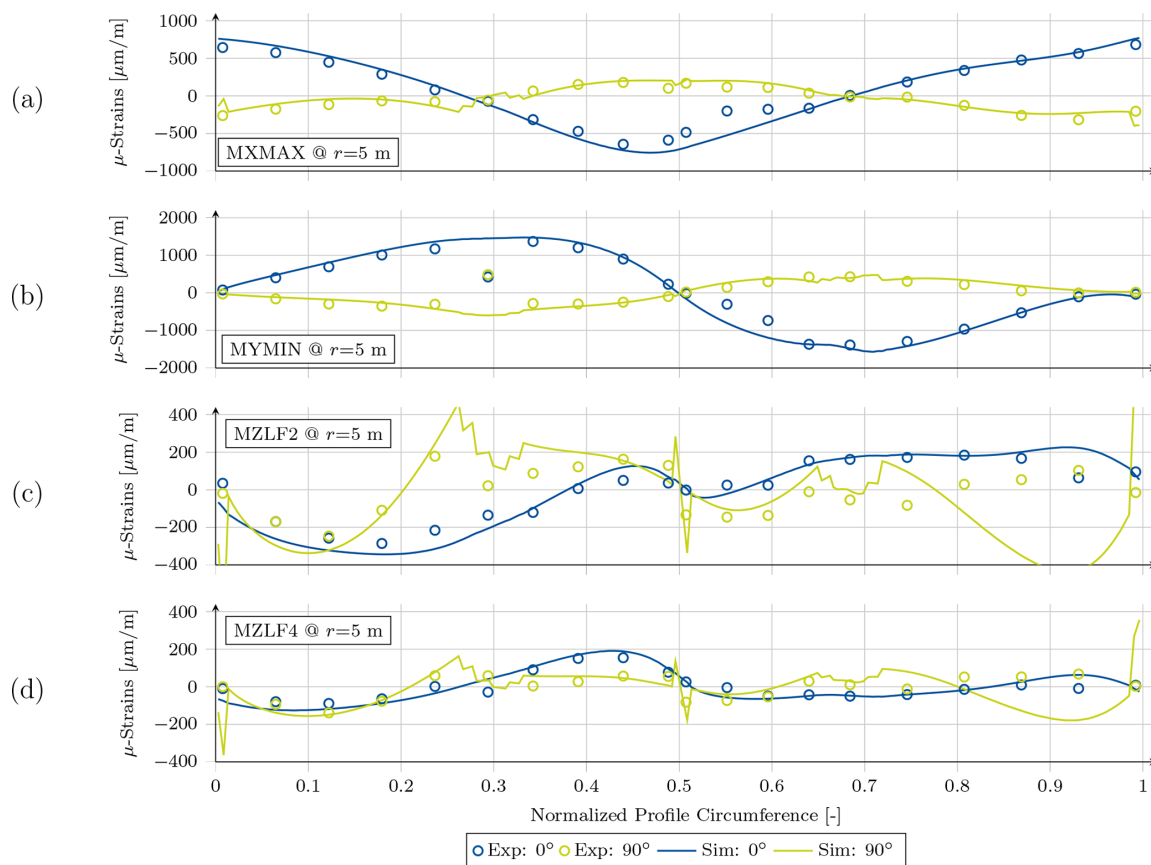


Figure B2. Spanwise and crosswise strains of the simulation and the test, plotted against the normalized profile circumference of the cross section at $r = 5$ m for the (a) MXMAX, (b) MYMIN, (c) MZLF2, and (d) MZLF4 load case.

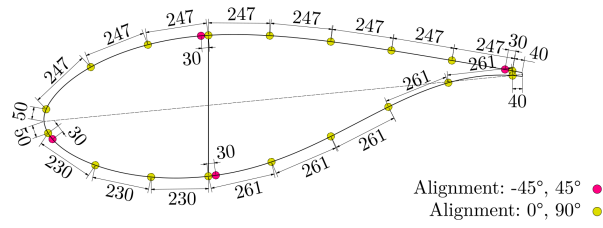


Figure B3. Cross-sectional sensor distribution at $r = 8$ m ($r_{\text{norm}} = 40\%$) (Haller and Noever-Castelos, 2021).

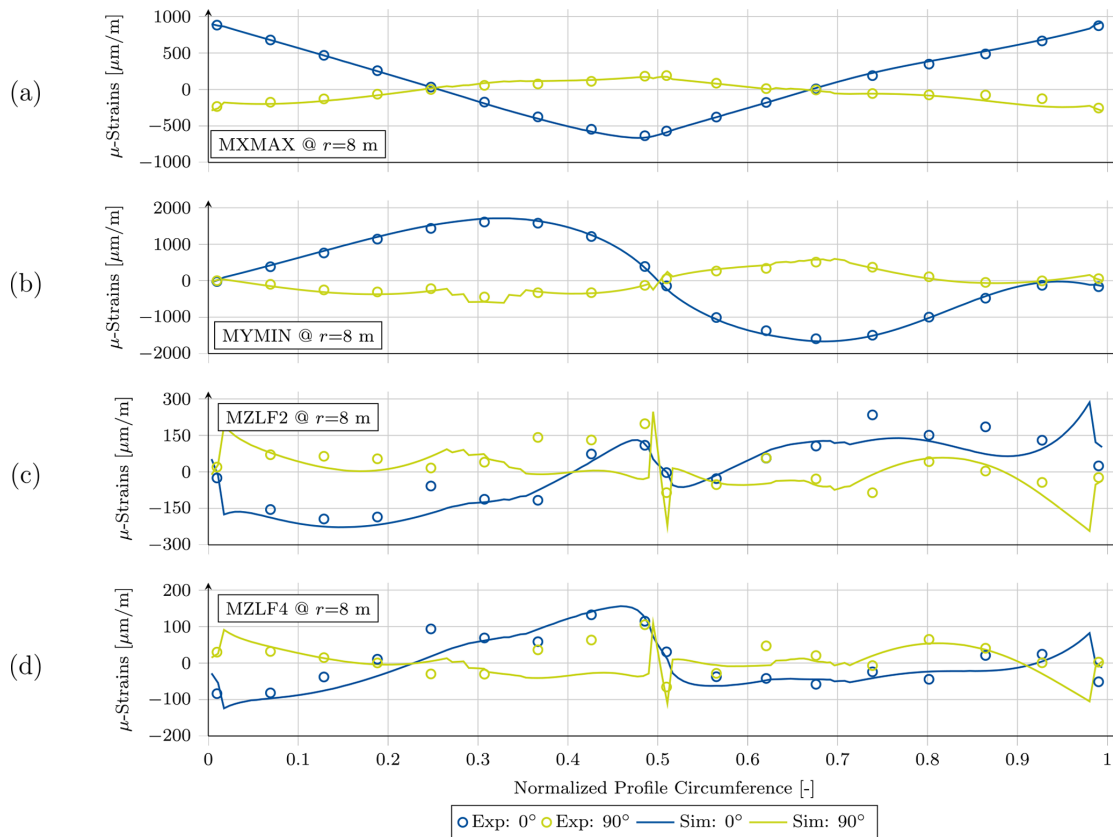


Figure B4. Spanwise and crosswise strains of the simulation and the test, plotted against the normalized profile circumference of the cross section at $r = 8$ m for the (a) MXMAX, (b) MYMIN, (c) MZLF2, and (d) MZLF4 load case.

Code and data availability. The code of MoCA is not publicly available but may be made available on request at conditions that need to be agreed upon. All processed experimental and simulation data that support the results of this research as well as the baseline finite-element model of the blade as an Ansys Mechanical input file are available at <https://doi.org/10.25835/0048541> (Noever-Castelos et al., 2021). All further detailed experimental information and documentation are published at <https://doi.org/10.5281/zenodo.4605409> (Haller and Noever-Castelos, 2021).

Author contributions. PNC implemented the parametrization and modeling methodology in MoCA, conducted the numerical simulations, compared the simulations with the tests, and wrote the paper. BH planned, executed, and documented the tests. CB is the supervisor and guided PNC in the conception of the ideas and participated in the specification of strain gauge positions as well as in writing, structuring, and reviewing the paper.

Competing interests. The contact author has declared that neither they nor their co-authors have any competing interests.

Disclaimer. The information in this paper is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Acknowledgements. The authors acknowledge the financial support by the Federal Ministry for Economic Affairs and Energy of Germany in the project SmartBlades2 (project numbers 0324032B/C). The authors further acknowledge the coordination effort of the German Aerospace Center (DLR), the very good cooperation with the project partners, and the fruitful discussions within the project consortium.

Financial support. The publication of this article was funded by the open-access fund of Leibniz Universität Hannover.

Review statement. This paper was edited by Joachim Peinke and reviewed by Sarah Barber and Martin Eder.

References

- ANSYS Inc.: Ansys[®] Academic Research Mechanical, Release 2021 R2, 2021.
- Berg, J. C. and Resor, B. R.: Numerical manufacturing and design tool (NuMAD v2.0) for wind turbine blades : user's guide, United States 2012, <https://doi.org/10.2172/1051715>, 2012.
- Berring, P., Branner, K., Berggreen, C., and Knudsen, H. W.: Torsional Performance of Wind Turbine Blades: Part I: Experimental Investigation, in: 16th International Conference on Composite Materials, Kyoto, Japan, 8–13 July 2007.
- Blasques, J. P. A. A.: User's Manual for BECAS: A cross section analysis tool for anisotropic and inhomogeneous beam sections of arbitrary geometry. Risø DTU – National Laboratory for Sustainable Energy, Denmark, Forskningscenter Risøe, Risøe-R No. 1785(EN), 2012.
- Blasques, J. P. and Stolpe, M.: Multi-material topology optimization of laminated composite beam cross sections, *Compos. Struct.*, 94, 3278–3289, <https://doi.org/10.1016/j.compstruct.2012.05.002>, 2012.
- Bottasso, C. L., Campagnolo, F., Croce, A., Dilli, S., Gualdoni, F., and Nielsen, M. B.: Structural optimization of wind turbine rotor blades by multilevel sectional/multibody/3D-FEM analysis, *Multibody Syst. Dyn.*, 32, 87–116, <https://doi.org/10.1007/s11044-013-9394-3>, 2014.
- Branner, K., Berring, P., Berggreen, C., and Knudsen, H. W.: Torsional performance of wind turbine blades – Part II: Numerical validation, in: 16th International Conference on Composite Materials, Kyoto, Japan, 8–13 July 2007.
- Bak, C., Zahle, F., Bitsche, R., Kim, T., Yde, A., Henriksen, L. C., Hansen, M. H., Blasques, J. P. A. A., Gaunaa, M., and Natarajan, A.: The DTU 10-MW Reference Wind Turbine, Published by DTU Wind Energy (DTU Wind Energy Report-I-0092), 2013.
- Chen, H., Yu, W., and Capellaro, M.: A critical assessment of computer tools for calculating composite wind turbine blade properties, *Wind Energy*, 13, 497–516, <https://doi.org/10.1002/we.372>, 2010.
- Chen, X., Zhao, W., Zhao, X., and Xu, J.: Failure Test and Finite Element Simulation of a Large Wind Turbine Composite Blade under Static Loading, *Energies*, 7, 2274–2297, <https://doi.org/10.3390/en7042274>, 2014.
- Chen, X., Zhao, X., and Xu, J.: Revisiting the structural collapse of a 52.3 m composite wind turbine blade in a full-scale bending test, *Wind Energy*, 20, 1111–1127, <https://doi.org/10.1002/we.2087>, 2017.
- Chen, X., Semenov, S., McGugan, M., Hjelm Madsen, S., Cem Yeniceli, S., Berring, P., and Branner, K.: Fatigue testing of a 14.3 m composite blade embedded with artificial defects – Damage growth and structural health monitoring, *Compos. Part A-Appl. S.*, 140, 106189, <https://doi.org/10.1016/j.compositesa.2020.106189>, 2021.
- Duineveld, N. P.: Structure and Possibilities of the FOCUS Design Package, WMC, 2008.
- Eder, M. A. and Bitsche, R. D.: Fracture analysis of adhesive joints in wind turbine blades, *Wind Energy*, 18, 1007–1022, <https://doi.org/10.1002/we.1744>, 2015.
- Greaves, P. and Langston, D.: Torsional testing of wind turbine blades, Conference presentation, Wind Energy Science Conference 2021 in Hannover, 25–28 May 2021.
- Gundlach, J. and Govers, Y.: Experimental modal analysis of aeroelastic tailored rotor blades in different boundary conditions, *J. Phys. Conf. Ser.*, 1356, 012023, <https://doi.org/10.1088/1742-6596/1356/1/012023>, 2019.
- Ha, K., Bätge, M., Melcher, D., and Czichon, S.: Development and feasibility study of segment blade test methodology, *Wind Energ. Sci.*, 5, 591–599, <https://doi.org/10.5194/wes-5-591-2020>, 2020.
- Haller, B. and Noever-Castelos, P.: Full scale blade test of a 20 m wind turbine blade within the SmartBlades2 project, Zenodo [data set], <https://doi.org/10.5281/zenodo.4605409>, 2021.
- International Eletrotechnical Commission: IEC 61400-23: Wind turbines – Part 23: Full-scale structural testing of rotor blades, ISBN 978-2-8322-4924-6, Edition 1, 2014.
- Jensen, F. M., Falzon, B. G., Ankersen, J., and Stang, H.: Structural testing and numerical simulation of a 34 m composite wind turbine blade, *Compos. Struct.*, 76, 52–61, <https://doi.org/10.1016/j.compstruct.2006.06.008>, 2006.
- Ji, Y. M. and Han, K. S.: Fracture mechanics approach for failure of adhesive joints in wind turbine blades, *Renew. Energ.*, 65, 23–28, <https://doi.org/10.1016/j.renene.2013.07.004>, 2014.
- Knebusch, J., Gundlach, J., and 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, 2159–2174, EASDAthens, in Athens Greece, 23–26 November 2020, <https://doi.org/10.47964/1120.9175.19508>, 2020.
- Laird, D., Montoya, F., and Malcolm, D.: Finite Element Modeling of Wind Turbine Blades, p. 354, 43rd AIAA

- Aerospace Sciences Meeting and Exhibit, Reno, Nevada, US, <https://doi.org/10.2514/6.2005-195>, 10–13 January 2005.
- Lekou, D. J., Bacharoudis, K. C., Farinas, A. B., Branner, K., Berring, P., Croce, A., Philippidis, T. P., and de Winkel, G. D.: A Critical Evaluation of Structural Analysis Tools used for the Design of Large Composite Wind Turbine Rotor Blades under Ultimate and Cycle Loading, in: Proceedings of the 20th International Conference on Composite Materials ICCM20 Secretariat, Copenhagen, Denmark, 19–24 July 2015.
- Marten, D., Wendler, J., Pechlivanoglou, G., Nayeri, C. N., and Paschereit, C. O.: QBlade: An open source tool for design and simulation of horizontal and vertical axis wind turbines, *International Journal of Emerging Technology and Advanced Engineering*, 3, Feb 2013, 264–269, 2013.
- Noever-Castelos, P., Haller, B., and Balzani, C.: Supplement to: Validation of a modeling methodology for wind turbine rotor blades based on a full scale blade test, Research Data Repository of the Leibniz University Hannover [data set], <https://doi.org/10.25835/0048541>, 2021.
- Overgaard, L. and Lund, E.: Structural collapse of a wind turbine blade. Part B: Progressive interlaminar failure models, *Compos. Part. A-Appl. S.*, 41, 271–283, <https://doi.org/10.1016/j.compositesa.2009.10.012>, 2010.
- Overgaard, L., Lund, E., and Thomsen, O. T.: Structural collapse of a wind turbine blade. Part A: Static test and equivalent single layered models, *Compos. Part. A-Appl. S.*, 41, 257–270, <https://doi.org/10.1016/j.compositesa.2009.10.011>, 2010.
- Pardo, D. R. and Branner, K.: Finite Element Analysis of the Cross-Section of Wind Turbine Blades; A Comparison between Shell and 2D-Solid Models, *Wind Engineering*, 29, 25–31, <https://doi.org/10.1260/0309524054353700>, 2005.
- Peeters, M., Santo, G., Degroote, J., and van Paepegem, W.: High-fidelity finite element models of composite wind turbine blades with shell and solid elements, *Compos. Struct.*, 200, 521–531, <https://doi.org/10.1016/j.compstruct.2018.05.091>, 2018.
- Reder, M. D., Gonzalez, E., and Melero, J. J.: Wind Turbine Failures – Tackling current Problems in Failure Data Analysis, *J. Phys. Conf. Ser.*, 753, 072027, <https://doi.org/10.1088/1742-6596/753/7/072027>, 2016.
- Rosemeier, M.: FEPROC Blade Model Verification – 3D Shell and Beam Model, Zenodo, <https://doi.org/10.5281/ZENODO.1493936>, 2018.
- Safarian, P.: Finite Element Modeling and Analysis Validation, Conference presentation, FEMAP Symposium, Seattle, Washington, US, 23 September 2015.
- SmartBlades2: Fabrication, Testing, and Further Development of Smart Rotor Blades, coordinated research project (project numbers 0324032A-H), supported by the Federal Ministry for Economic Affairs and Energy of Germany due to a decision of the German Bundestag, 2016–2020.
- Teßmer, J., Icpinar, C., Sevinc, A., Daniele, E., Riemschneider, J., Hölling, M., and Balzani, C.: Schlussbericht Smart Blades: Technical Report, 2016.
- Tiedemann, M. and Chen, X.: Experimental study on torsional stiffness of a wind turbine blade through combined loading, Conference presentation, Wind Energy Science Conference 2021, Hannover, 25–28 May 2021.
- Yu, W., Volovoi, V. V., Hodges, D. H., and Hong, X.: Validation of the Variational Asymptotic Beam Sectional Analysis, *AIAA Journal*, 40, 2105–2112, <https://doi.org/10.2514/2.1545>, 2002.
- Zahle, F., Réthoré, P.-E., Graf, P., Dykes, K., and Ning, A.: FUSED-Wind dev, available at: <https://github.com/FUSED-Wind> (last access: 19 January 2022), 2020.

3

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

2nd Paper

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. Classical optimization algorithms represent the state of the art in updating structural models. However, this feasibility study approaches the updating task by inverting the model through the application of invertible neural networks, without costly optimization or sampling algorithms. In our use case, rotor blade cross sections are updated to match given cross sectional parameters.

P2-1	Introduction	42
P2-2	Model updating methodology with INNs	44
P2-3	Input and output feature selection	48
P2-4	INN structure, training, and evaluation	52
P2-5	Conclusion	56
P2-6	Appendix A: Sensitivity analysis.	60
P2-7	Appendix B: Evaluation of cINN	61



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-Castelos¹  | Lynton Ardizzone²  | Claudio Balzani¹ ¹Institute for Wind Energy Systems, Leibniz University Hannover, Hanover, Germany²Visual 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,

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2021 The Authors. *Wind Energy* published by John Wiley & Sons Ltd.

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 layout) 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 search⁹) 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 updating¹² 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 models¹⁵ 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 layout 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 control²⁷) 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

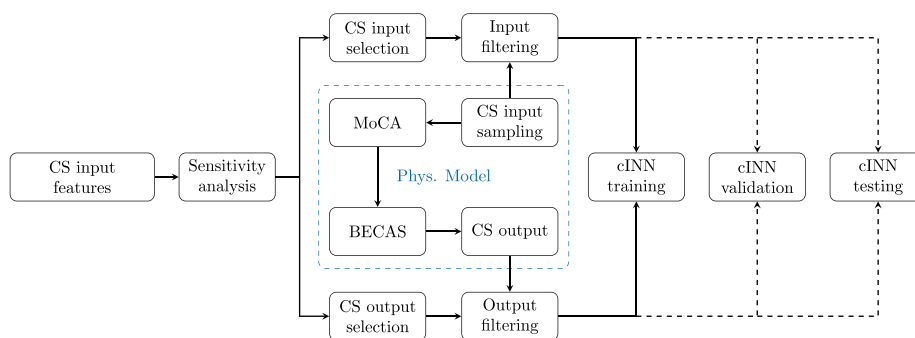


FIGURE 1 Overall workflow of this study. Based on all cross-sectional input features, a sensitivity analysis is performed to determine the relevant input and output features. The physical model is then used to generate sample sets of input and output features, which are then filtered by the feature selection 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 $\bar{y} = 0$ 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,

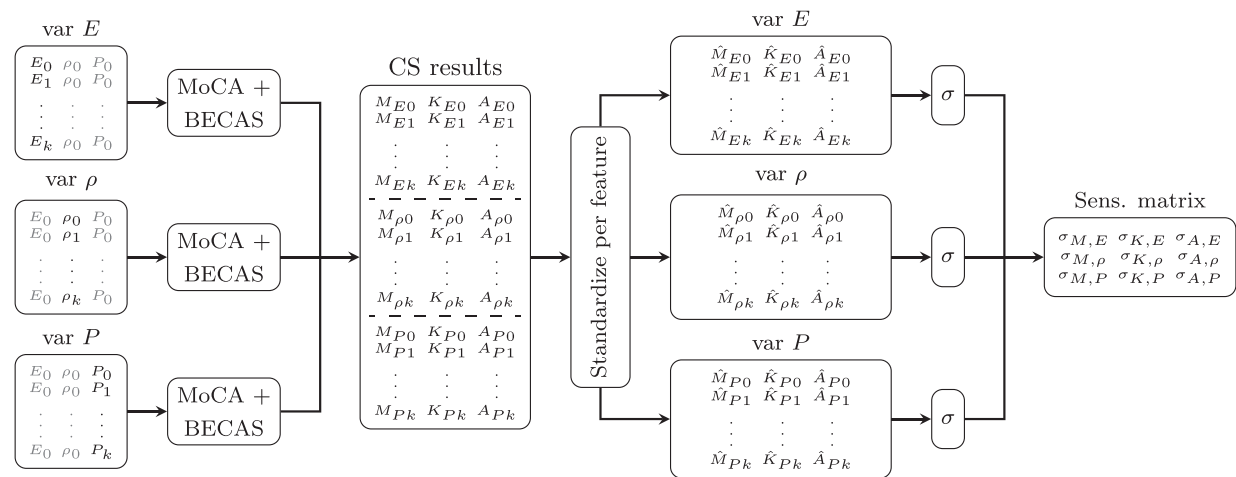


FIGURE 2 The feature selection process based on a sensitivity analysis applied to a simplified cross-sectional (CS) example with three input features: Young's modulus E , density ρ , and division point position P . The algorithm varies all individually and calculates their corresponding CS characteristics, 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 $p_x(x|y)$.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 $p_x(x|y)$ 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

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

Here, $\det \left(\frac{\partial f}{\partial x} \right)$ 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 $p_x(x|y)$ will match the true posterior of the inverse problem $p^*(x|y)$ if the average log-likelihood of the model's 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:

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

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,

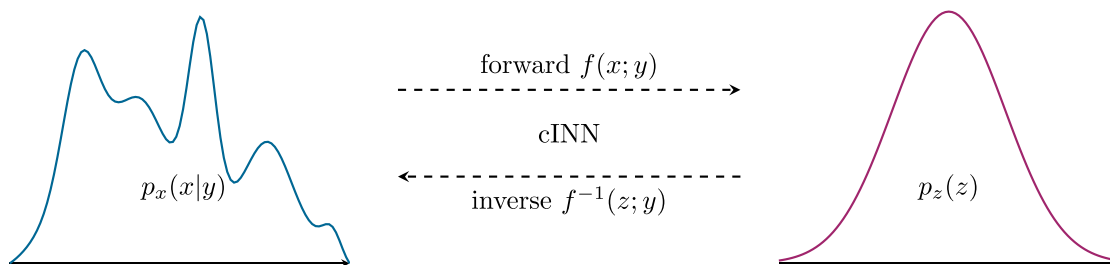


FIGURE 3 Schematic illustration of the principle of a *cINN*. The *cINN* f conditionally transforms and reshapes between the posterior $p_x(x|y)$ and the latent distribution $p_z(z)$

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 RealNVP³⁵ 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:

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

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

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

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

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

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 then 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

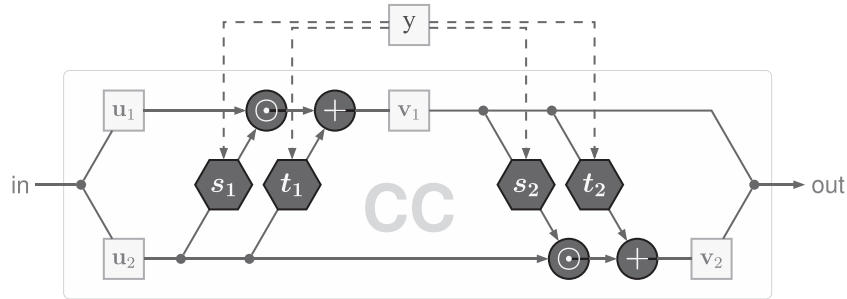


FIGURE 4 Structure of a conditional affine coupling block (CC)³⁹

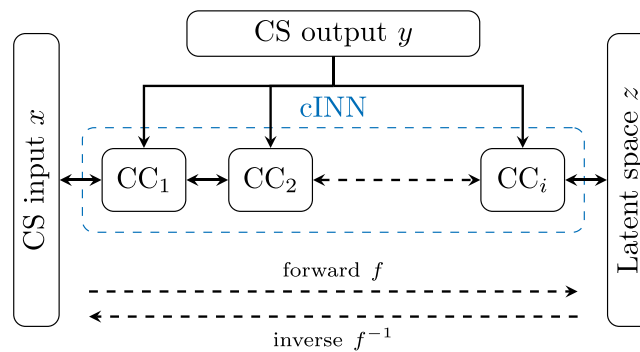


FIGURE 5 The conditional invertible neural network (cINN) structure applied to cross-sectional model updating. In the forward path f , cross-sectional (CS) input features x are processed over a sequential concatenation of conditional coupling (CC) blocks, which represents the cINN. The CS output features 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, $\pm 5\%$ 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 $P_{SS,TE,core}$ to $P_{SS,LE,core}$ 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

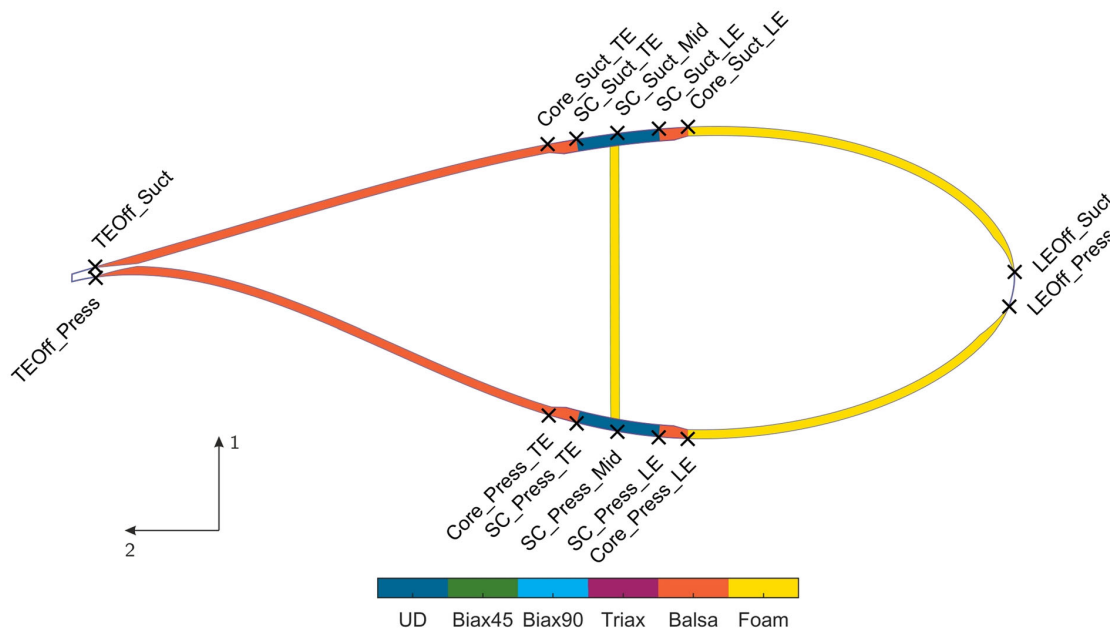


FIGURE 6 This is a cross section of the SmartBlades2 DemoBlade at a radial position of $R = 6$ m. The division points on the circumference divide the blade shell into sections of equal material layup as follows: $P_{SS,TE,offset}$ to $P_{SS,TE,core}$ is the suction side (SS) sandwich panel located at the trailing edge (TE); $P_{SS,TE,spar\ cap}$ to $P_{SS,LE,spar\ cap}$ denotes the location of the spar cap; this is flanked by balsa transition pieces in-between $P_{SS,TE,core}$ to $P_{SS,LE,core}$; followed by the sandwich panel located to the leading edge (LE) from $P_{SS,LE,core}$ to $P_{SS,LE,offset}$. 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 A5⁵⁵

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

Parameter	Attribute	Max. variation	Parameter	Attribute	Max. variation	Note
UD	$E_{11}, E_{22}, G_{12}, \nu_{12}, \rho$	$\pm 10\%$	$P_{SS,TE,offset}$	Location	± 10 mm	
Biax 45°	$E_{11}, E_{22}, G_{12}, \nu_{12}, \rho$	$\pm 10\%$	$P_{SS,TE,core}$	Location	± 15 mm	
Biax 90°	$E_{11}, E_{22}, G_{12}, \nu_{12}, \rho$	$\pm 10\%$	$P_{SS,TE,spar\ cap}$	Location	± 15 mm	Prefabr.: varied synchr.
Triax	$E_{11}, E_{22}, G_{12}, \nu_{12}, \rho$	$\pm 10\%$	$P_{SS,Mid,spar\ cap}$	Location	± 15 mm	
Balsa	$E_{11}, G_{12}, \nu_{12}, \rho$	$\pm 10\%$	$P_{SS,LE,spar\ cap}$	Location	± 15 mm	
Foam	E, G, ρ	$\pm 10\%$	$P_{SS,LE,core}$	Location	± 15 mm	
			$P_{SS,LE,offset}$	Location	± 10 mm	
			$P_{PS,LE,offset}$	Location	± 10 mm	
			$P_{PS,LE,core}$	Location	± 15 mm	
			$P_{PS,LE,spar\ cap}$	Location	± 15 mm	Prefabr.: varied synchr.
			$P_{PS,Mid,spar\ cap}$	Location	± 15 mm	
			$P_{PS,TE,spar\ cap}$	Location	± 15 mm	
			$P_{PS,TE,core}$	Location	± 15 mm	
			$P_{PS,TE,offset}$	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 coordinates
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 $P_{XX,Mid,spar\ cap}$ 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
/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
/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
/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
/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
/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
/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 threshold $\lambda = 0.25$ are marked bold and only the finally selected features are displayed. A maximum of $\pm 10\%$ variation applies for material input features, ± 15 mm for the spar cap division points $P_{XX,Mid,spar\ cap}$, 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 _{1,1} , 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
<i>ν</i> _{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 , 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
<i>ν</i> _{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 _{1,1} , 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 , 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
<i>ν</i> _{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 _{1,1} , 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 , 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
<i>ν</i> _{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 , 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
<i>ν</i> _{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
<i>ν</i> _{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 _{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 _{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 _{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 _{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 _{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 _{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 threshold $\lambda = 0.25$ are marked bold and only the finally selected features are displayed. A maximum of $\pm 10\%$ variation applies for material input features, ± 15 mm for the spar cap division points P_{XX,Mid,spar 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 Pytorch⁵⁸ 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 AdaGrad⁵⁹ 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 $\pm x_{\text{max}}$ from Table 1, the standard deviation is defined as follows:

$$\sigma_x = |x_{\text{max}}| \cdot \sqrt{\frac{1}{3}} \quad (7)$$

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 were 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.

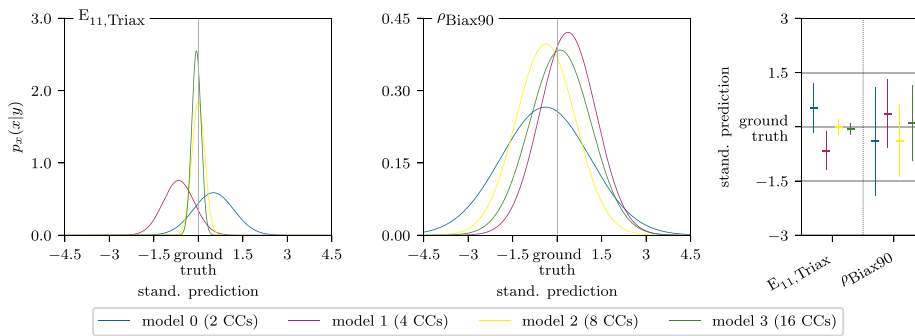


FIGURE 7 Exemplary standardized prediction for the two input features $E_{11, \text{Triax}}$ and ρ_{Biax90} of one sample computed with all four models from Table 4. The left two graphs show the predicted distribution of the corresponding feature x . The right graph summarizes the same results as error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

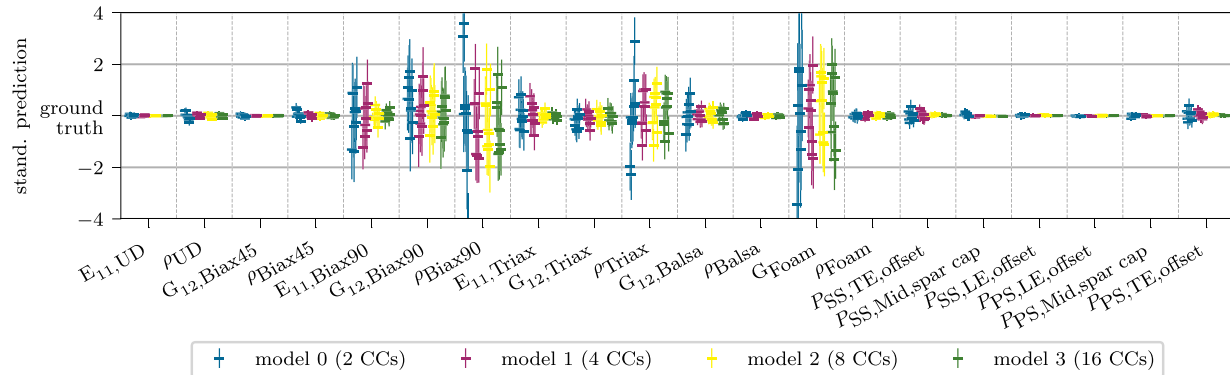


FIGURE 8 Input feature prediction showed for 10 random samples of four different models with full output as conditional features. The four invertible neural networks increase in depths

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 ($\sigma_{E_{11, \text{Triax}}} = 0.72$) is much higher than that of the deepest ($\sigma_{E_{11, \text{Biax90}}} = 0.16$). That means the shallowest model has a poor prediction confidence for $E_{11, \text{Triax}}$ 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 $E_{11, \text{Triax}}$, 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

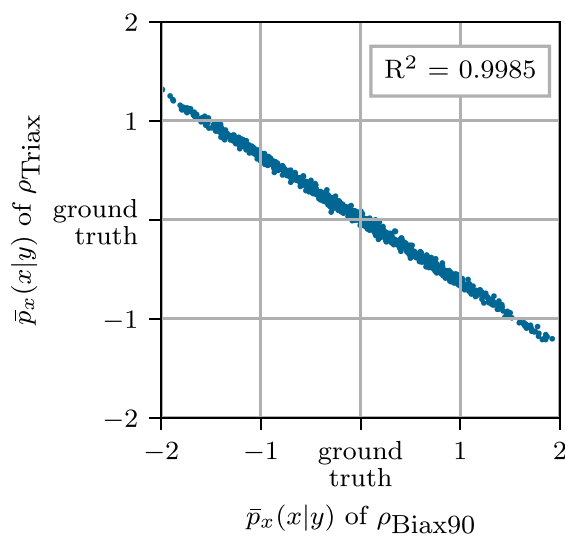


FIGURE 9 Counteraction of the mean predicted densities ρ_{Biax90} and ρ_{Triax} . The highly correlating samples show that the cINN correctly detects this ambiguity

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 were 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.

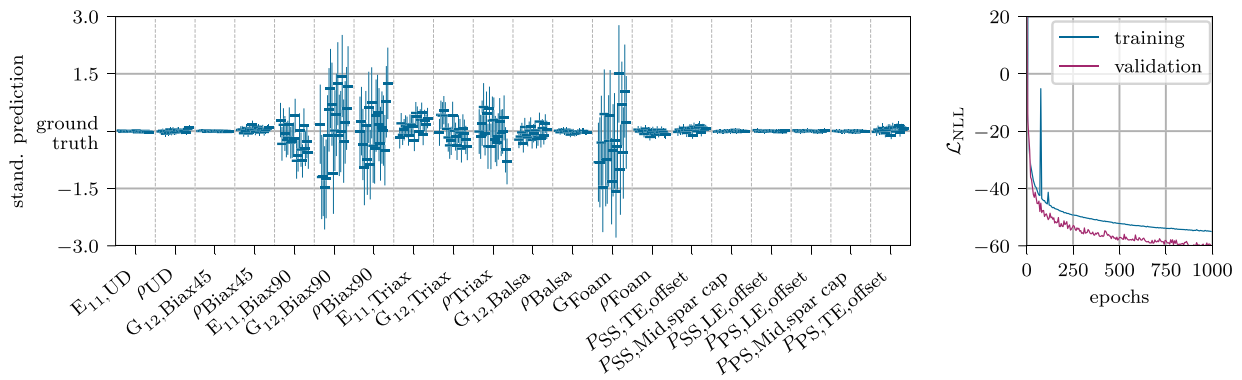


FIGURE 10 Final model: input feature prediction on the left graph for 30 random samples with full output as conditional features. Right graph depicts the negative logarithmic likelihood loss curve of the cINN for training and validation samples

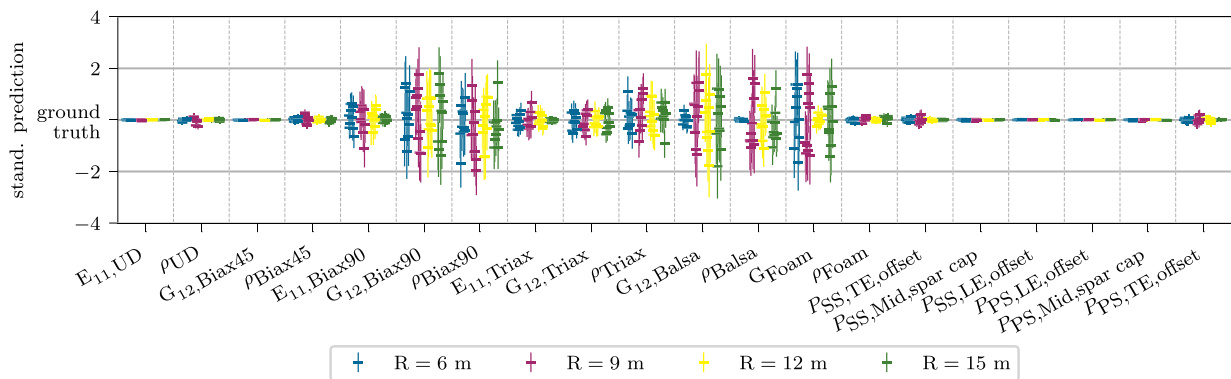


FIGURE 11 Input feature prediction showed for 10 random samples at four different radial positions with full output as conditional features. The four cINNs were each trained individually for their respective radius

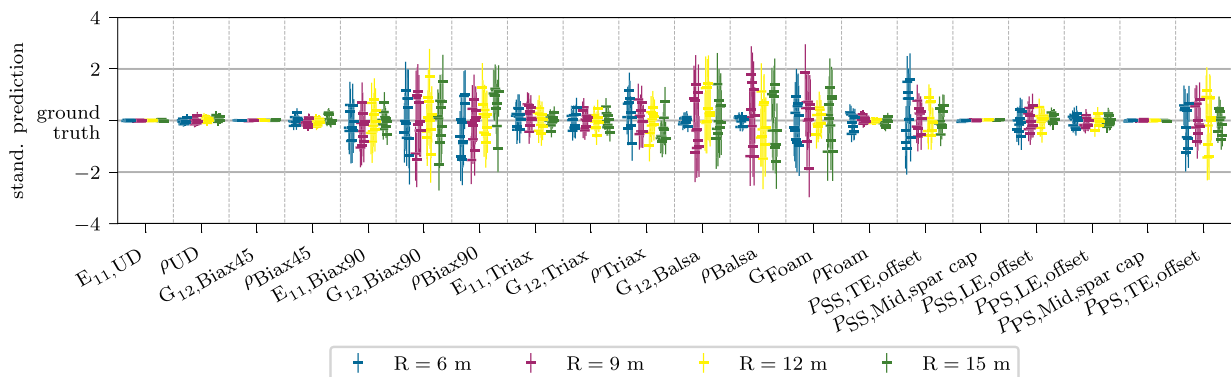


FIGURE 12 Input feature prediction showed for 10 random samples at four different radial positions only considering the stiffness and mass matrix as conditional features. The four cINNs were each trained individually for their respective radius

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: $G_{12,Biax90}$, 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 ($P_{SS,TE,offset}$, $P_{SS,LE,offset}$, $P_{PS,LE,offset}$, and $P_{PS,TE,offset}$). 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 layout 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-clNN>.

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

- Gundlach J, Govers Y. Experimental modal analysis of aeroelastic tailored rotor blades in different boundary conditions. *J Phys Conf Ser.* 2019;1356:12023.
- Mottershead JE, Friswell MI. Model updating in structural dynamics: A survey. *J Sound Vib.* 1993;167(2):347-375.
- 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.
- 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.
- 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.
- 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.
- Yang YB, Chen YJ. A new direct method for updating structural models based on measured modal data. *Eng Struct.* 2009;31(1):32-42.
- Bruns M, Hofmeister B, Griebmann 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.
- Hofmeister B, Bruns M, Rolfes R. Finite element model updating using deterministic optimisation: a global pattern search approach. *Eng Struct.* 2019;195:373-381.
- Mojtahedi A, Lotfollahi Yaghin MA, Hassanzadeh Y, Etefagh 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.
- 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.
- Sun H, Büyükköztürk 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.
- Sunnåker M, Busetto AG, Numminen E, Corander J, Foll M, Dessimoz C. Approximate bayesian computation. *PLoS Comput Biol.* 2013;9(1):e1002803.
- Serna A, Bucher C. Advanced surrogate models for multidisciplinary design optimization. In: 6th weimar optimization and stochastic days 2009; 2009.
- 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.
- Trehan S, Carlberg KT, Durlafsky LJ. Error modeling for surrogates of dynamical systems using machine learning. *International Journal for Numerical Methods in Engineering.* 2017;112(12):1801-1827.
- Chollet F. *Deep learning with python*, Safari Tech Books Online. Shelter Island, NY: Manning; 2018. <http://proquest.safaribooksonline.com/9781617294433>
- Marwala T. *Finite-element-model updating using computational intelligence techniques: Applications to structural dynamics*. London: Springer; 2010.
- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- 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. For-skningsscenter Risøe. Risøe-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, Köthe 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 $0^\circ/90^\circ$	1	0.65 mm
Foam/Balsa	1	20 mm
Biax $0^\circ/90^\circ$	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 $R = 6\text{ m}$ – 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}	$\sigma_{PC,Ref}$	$\sigma_{PC,EC}$
$E_{1,1, UD}$	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
$E_{2,2, UD}$	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
$G_{1,2, UD}$	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
$\nu_{1,2, UD}$	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
$E_{1,1, Biax45}$	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
$E_{2,2, Biax45}$	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
$G_{1,2, Biax45}$	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
$\nu_{1,2, Biax45}$	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
$E_{1,1, Biax90}$	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
$E_{2,2, Biax90}$	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
$G_{1,2, Biax90}$	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
$\nu_{1,2, Biax90}$	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
$E_{1,1, Triax}$	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
$E_{2,2, Triax}$	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
$G_{1,2, Triax}$	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
$\nu_{1,2, Triax}$	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
$E_{1,1, Balsa}$	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
$G_{1,2, Balsa}$	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
$\nu_{1,2, Balsa}$	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
ν_{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
$P_{S,TE,offset}$	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
$P_{S,TE,core}$	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
$P_{S,TE,sp, cap}$	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
$P_{S,Mid,sp, cap}$	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
$P_{S,LE,sp, cap}$	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
$P_{S,LE,core}$	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}	$\sigma_{PC,Ref}$	$\sigma_{PC,EC}$
P _{SSL,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,spat 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,spat 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,spat 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 $R = 6\text{ m}$ – 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}		
$E_{11, UD}$	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	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	0.0	
$G_{12, UD}$	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	0.0	0.1	
$\nu_{12, UD}$	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	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	0.0	
$E_{11, Biax45}$	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	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	0.0	0.0
$G_{12, Biax45}$	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	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	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	0.0	0.0
$E_{11, Biax90}$	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	0.0	0.0
$E_{22, Biax90}$	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	0.0	0.0
$G_{12, Biax90}$	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	0.0	1.9	0.0
$\nu_{12, Biax90}$	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	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	0.0	0.0
$E_{11, Triax}$	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	0.0	0.0
$E_{22, Triax}$	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	0.0	0.0
$G_{12, Triax}$	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	0.0	5.6	0.0
$\nu_{12, Triax}$	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	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	0.0	0.0
$E_{11, Balsa}$	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	0.0	0.0
$G_{12, Balsa}$	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	0.0	0.4	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	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	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	0.0	0.0	0.0	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	0.0	0.1	0.0	0.0	0.0	0.0	0.1	0.0
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	0.0	0.0
$P_{SS,TE,offset}$	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	0.0	0.0
$P_{SS,TE,core}$	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	0.0	0.1	0.0
$P_{SS,TE,spar\ cap}$	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	0.0	0.1	0.0
$P_{SS,Mid,spar\ cap}$	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	0.0	0.1	0.0
$P_{SS,LE,spar\ cap}$	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	0.0	0.1	0.0
$P_{SS,LE,core}$	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	0.0	0.1	0.0

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 _{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 _{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 _{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 _{PS,LE,spat 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 _{PS,Mid,spat 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 _{PS,TE,spat 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 _{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 _{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 (Continued)

	M ₁₁	M ₁₂	M ₁₃	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	0.1	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	0.1	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	0.9	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} spare cap	0.1	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} spare cap	0.1	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} spare cap	0.1	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	0.9	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	0.3	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

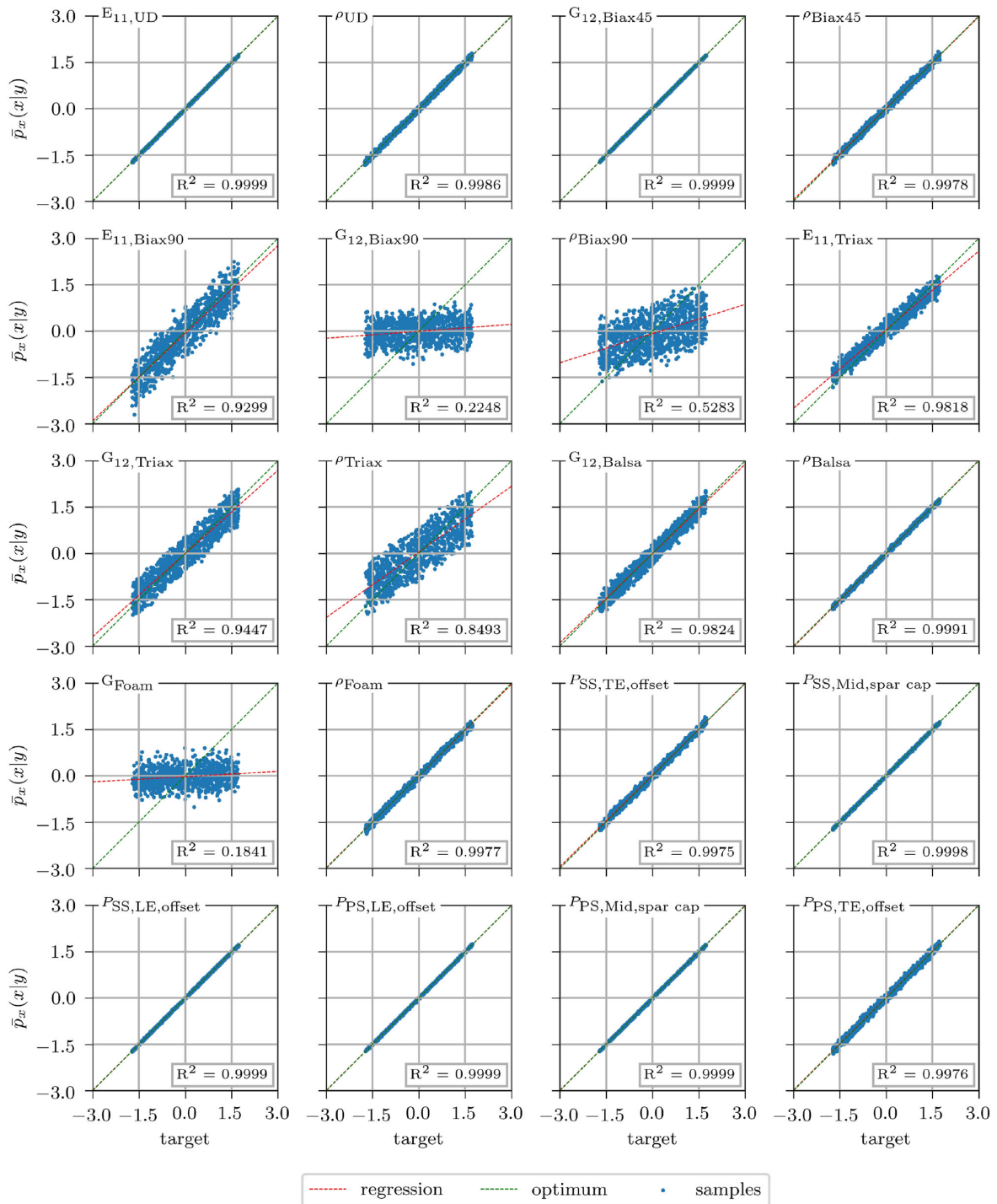


FIGURE B1 Correlation between input feature prediction and ground truth measured with R^2 . Optimum values will be located on $f(x) = m \cdot x$

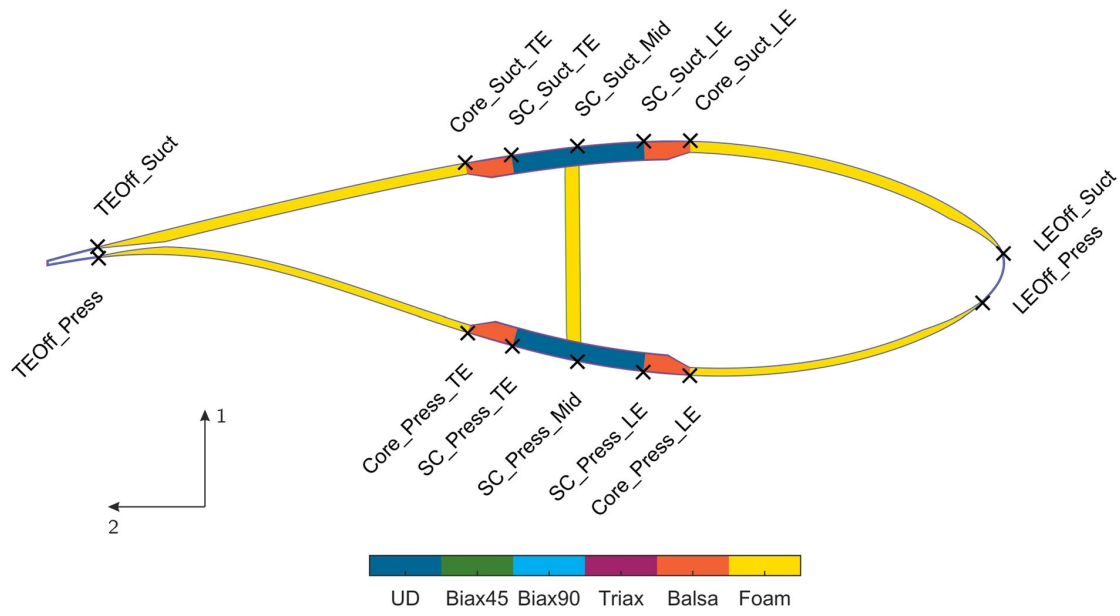


FIGURE B2 Cross section of the SmartBlades2 DemoBlade at a radial position of $R = 12$ m

4

Model updating of a wind turbine blade finite element Timoshenko beam model with invertible neural networks

3rd Paper

In the wind energy industry, a digital twin is fast becoming a key instrument for the monitoring of a wind turbine blade's life cycle. Here, our introduced model updating with invertible neural networks provides an efficient and powerful technique to represent the real blade as built. This method is applied to a full finite element Timoshenko beam model of a blade to successfully update material and layup parameters. The advantage over state-of-the-art methods is the established inverse model.

P3-1	Introduction	70
P3-2	Sensitivity analysis of modal response	72
P3-3	Invertible neural network architecture	76
P3-4	Model updating of a rotor blade beam model	78
P3-5	Conclusion	87
P3-6	Appendix A: Tables & Figures ..	88



Wind Energ. Sci., 7, 623–645, 2022
<https://doi.org/10.5194/wes-7-623-2022>
 © Author(s) 2022. This work is distributed under
 the Creative Commons Attribution 4.0 License.



Model updating of a wind turbine blade finite element Timoshenko beam model with invertible neural networks

Pablo Noever-Castelos¹, David Melcher², and Claudio Balzani¹

¹Institute for Wind Energy Systems, Leibniz University Hannover, Appelstr. 9A, 30167 Hanover, Germany

²Department of Rotor Blades, Fraunhofer IWES, Fraunhofer Institute for Wind Energy Systems, Am Seedeich 45, 27572 Bremerhaven, Germany

Correspondence: Pablo Noever-Castelos (research@iwes.uni-hannover.de)

Received: 4 August 2021 – Discussion started: 17 September 2021

Revised: 17 December 2021 – Accepted: 8 February 2022 – Published: 16 March 2022

Abstract. Digitalization, especially in the form of a digital twin, is fast becoming a key instrument for the monitoring of a product's life cycle from manufacturing to operation and maintenance and has recently been applied to wind turbine blades. Here, model updating plays an important role for digital twins, in the form of adjusting the model to best replicate the corresponding real-world counterpart. However, classical updating methods are generally limited to a reduced parameter space due to low computational efficiency. Moreover, these approaches most likely lack a probabilistic evaluation of the result.

The purpose of this paper is to extend a previous feasibility study to a finite element Timoshenko beam model of a full blade for which the model updating process is conducted through the novel approach with invertible neural networks (INNs). This type of artificial neural network is trained to represent an inversion of the physical model, which in general is complex and non-linear. During the updating process, the inverse model is evaluated based on the target model's modal responses. It then returns the posterior prediction for the input parameters. In advance, a global sensitivity study will reduce the parameter space to a significant subset on which the updating process will focus.

The finally trained INN excellently predicts the input parameters' posterior distributions of the proposed generic updating problem. Moreover, intrinsic model ambiguities, such as material densities of two closely located laminates, are correctly captured. A robustness analysis with noisy response reveals a few sensitive parameters, though most can still be recovered with equal accuracy. And, finally, after the resimulation analysis with the updated model, the modal response perfectly matches the target values. Thus, we successfully confirmed that INNs offer an extraordinary capability for structural model updating of even more complex and larger models of wind turbine blades.

1 Introduction

Wind turbine blades are enormous composite structures exposed to extreme and harsh environmental conditions. Due to these circumstances, structural health or condition monitoring plays a critical role in reliably ensuring the endurance of the rotor blade. However, this raises the need for an accurate model representation of the structure as built. In this context, the digital twin is emerging as a powerful instrument (Grieves, 2019) for these monitoring systems during operational time, though it can already be involved in early stages

of manufacturing (Sayer et al., 2020). The concept of model updating is central to achieving a digital product twin, for example, by updating the preliminary blade design based on sensor responses from blade characterization tests. This process of model updating ensures that the current stage of the digital twin represents the blade as built.

1.1 Model updating of wind turbine blades

Model updating has grown in importance in light of digitalization of the wind turbine blades; however, it has only

been marginally explored in the literature. Similarly to other structural dynamic model updating applications (Sehgal and Kumar, 2016), the publications on rotor blade model updating typically follow metaheuristic optimization techniques and define the objective function based on the modal assurance criterion (MAC), which represents a common metric for the quantitative comparison of modal shapes (Pastor et al., 2012). Other related modal metrics can be found in Allemang (2003). The most recent publications, such as Hofmeister et al. (2019) and Bruns et al. (2019), apply classical metaheuristic optimization algorithms to update the model parameters and localize damage in a generic problem with a finite element beam blade model. These publications evaluate a global pattern search and compare it to evolutionary, particle swarm, and genetic optimization algorithms. The objective function is based upon the natural frequencies and the MAC value. Furthermore, the MAC and the coordinate modal assurance criterion (COMAC) are applied in the model updating process of a finite element shell model of a rotor blade conducted by Knebusch et al. (2020). That study aims to update the blade model of a built blade along with high-fidelity modal measurements and a gradient-based optimization approach. Another approach presented by Schröder et al. (2018) uses a two-stage metaheuristic optimization to detect damage and ice accretion on a rotor blade. A global optimization with a simulated quenching algorithm is followed by a local method (sequential quadratic programming) to minimize the objective function, consisting of natural frequencies and mode shapes. Omenzetter and Turnbull (2018) implemented metaheuristic optimization methods (fireflies and virus optimization) to detect damage in a finite element beam model of a blade and compare the performance to a simplified beam experiment. Other publications cover simplified model updating procedures of low-level wind turbine blade models (Velazquez and Swartz, 2015; Liu et al., 2012; Lin et al., 2018). While most of the referred contributions focus on the field of damage detection, the model updating conducted by Luczak et al. (2014) highlights the impact of a flexible support structure of the test setup of modern blades, which was also revealed by Knebusch et al. (2020).

1.2 Drawbacks of current updating approaches

Most of these publications encounter three major problems:

1. Due to the aforementioned computational effort, the studies have been restricted to simple models.
2. They typically lack an efficient probabilistic approach to evaluate the uncertainty in the results.
3. All approaches only address one particular state of the blade at a defined condition and not a generalized inverse model.

The aforementioned approaches can be classified as deterministic and thus lead to results which are not necessarily the

global optima. Therefore, these methodologies may require the process to be run several times to ensure result validity (Schröder et al., 2018; Omenzetter and Turnbull, 2018). This is especially problematic since metaheuristic optimization algorithms are computationally expensive due to their iterative model evaluation (Chopard and Tomassini, 2018). As a reference, Bruns et al. (2019) performed 500 iterations for two updating parameters and 1500 iterations for five updating parameters, while in Omenzetter and Turnbull (2018) the firefly optimization of two updating parameters required 157 iterations until convergence and the virus optimization required 5000 iterations. Newer model updating techniques involve probabilistic approaches such as a sensitivity-based method (Augustyn et al., 2020) or Bayesian optimization (Marwala et al., 2016). The latter is based on sampling techniques such as Markov chain Monte Carlo methods to cover the complete parameter space. However, these approaches typically require even more model evaluations as stated in Patelli et al. (2017). There, a relatively simple model of a 3-degree-of-freedom (DOF) mass-spring system demanded 12 000 samples for the Bayesian solution, which was approximately 10 times higher than for the sensitivity-based method. Iterations are always model dependent, but to give a reference for real-time consumption, the model generator used in this publication (Noever-Castelos et al., 2021) takes on average approx. 80 s on a single-core device for one iteration, i.e., model creation. And finally, from the model updating we obtain one solution of input parameters for a particular set of model response parameters. However, if the model response changes, the whole optimization process has to be repeated. While in most applications a solution for a particular model is sufficient, an inverted model, which maps model responses to input parameters, can be beneficial, e.g., in quality management during serial production. This reveals a niche for an efficient method to invert the physical model, enabling a fast evaluation of the model states at any time.

1.3 Model updating via invertible neural networks

This research framework is based on Noever-Castelos et al. (2021), a feasibility study on a first structural level of wind turbine blades. The research performs a model updating with *conditional invertible neural networks* (cINNs) (Ardizzone et al., 2019b) for four selected cross-sections of a wind turbine blade. Noever-Castelos et al. (2021) consider a set of material and layup parameters as updateable inputs and take cross-sectional structural beam properties, such as stiffness and mass matrix, as model outputs to define the objective values. A sensitivity analysis following a one-at-a-time approach identified a parameter subspace selection of 19 significant input parameters for the updating process. The specific objective of this current investigation in contrast to the aforementioned publication (Noever-Castelos et al., 2021) is to

1. extend the feasibility study and methodology to a complete three-dimensional finite element Timoshenko beam model of a wind turbine blade as applied in real-world problems, instead of analyzing isolated cross-sections;
2. introduce parameter splines for the input variation along the blade;
3. use modal blade shapes and frequencies as the model response;
4. replace the sensitivity analysis for the parameter subspace selection by the global variance-based Sobol' method (Sobol', 1993), which takes interactions of input parameters into account;
5. implement a preprocessing feed-forward neural network for the cINN conditions;
6. analyze the potential of replacing or neglecting the sensitivity analysis by training the cINN on the full parameter space.

However, this investigation is still designed to reveal the feasibility with respect to a complex full three-dimensional Timoshenko beam model, before applying the method to a high dimensional real-world and non-generic problem.

1.4 Outline

This study will follow the outline of Noever-Castelos et al. (2021). The first section after the Introduction presents the sensitivity analysis procedure and discusses the physical model built in MoCA (Model Creation and Analysis Tool for Wind Turbine Rotor Blades) (Noever-Castelos et al., 2022) and BECAS (BEam Cross section Analysis Software) (Blasques and Stolpe, 2012). The chosen architecture of the cINN is explained in Sect. 3. Section 4 covers the discussion of results, with a general analysis of the updating results in Sect. 4.1. Section 4.2 reveals intrinsic model ambiguities before the model robustness to noisy model responses is examined in Sect. 4.3. A resimulation analysis to ensure high updating quality is performed in Sect. 4.4. Section 4.5 presents a method to replace the computationally expensive sensitivity analysis. This is then all followed by the conclusion in Sect. 5.

2 Sensitivity analysis of modal responses of a rotor blade finite element beam model

Typically a physical model consists of several input parameters defining the model behavior. The model is then evaluated, or simulations are performed, which yield a model response. However, not all input parameters equally contribute to the particular model response. A sensitivity analysis helps

to identify the most significant input parameters. It is an underestimated powerful tool to reduce the model dimensions without losing significant information. Especially for model updating purposes this can make a huge difference in performance. This section will discuss the applied sensitivity method as well as the applied model and parameter subspace selection.

2.1 Sobol' sensitivity method

Noever-Castelos et al. (2021) performed a sensitivity analysis to reflect how input distributions influence the output distribution's variance and mean value in order to identify relevant input and output features for the model updating process with the invertible neural network. There, a one-at-a-time approach is used, where values vary individually and their impact on the output is analyzed. In contrast to Noever-Castelos et al. (2021), this contribution will make use of a variance-based approach, called the Sobol' method, or Sobol' index (Sobol', 1993, 2001). This method is widely used in research and is used here, as it also applies globally to non-linear models and analyzes the influence of input parameter interaction on the model response. For a multivariate function $y = f(x_1, \dots, x_n)$, Sobol' derived the first-order Sobol' index S_i for the variable x_i as follows:

$$S_i = \frac{V[\mathbb{E}(y|x_i)]}{V(y)}. \quad (1)$$

This is a measure of to what extent the impact of varying x_i will have on the output y . On the basis of a random sampling of the parameters x , $\mathbb{E}(y|x_i)$ represents the expectation E of all y values for a constant value of x_i . It can be understood as an average of y corresponding to a slice of the x_i domain in the parameter space. $V[\mathbb{E}(y|x_i)]$ is then the variance of all expectations over the range of values of x_i , i.e., slices of the x_i domain (Saltelli et al., 2008). This variance is finally related to the overall variance of y . The first-order Sobol' index range is $0 \leq S_i \leq 1$. Higher-order Sobol' indices can also be extracted, see Saltelli et al. (2008), which measure the sensitivity of parameter interactions. For instance the second-order Sobol' index shows the joint effect of two parameters on the output, whereas third indices express the joint effect of three parameter interactions and so on. Although these indices can give a significant insight into the model, such as existing collinearities, the number of indices grows geometrically with the number of parameters, which quickly makes the computation intractable. However, the total Sobol' index S_{T_i} gathers the total sensitivity for a parameter including the first-order and all higher-order interactions. According to Saltelli et al. (2008) the total index S_{T_i} is calculated as follows:

$$S_{T_i} = 1 - \frac{V[\mathbb{E}(y|x_{\sim i})]}{V(y)}, \quad (2)$$

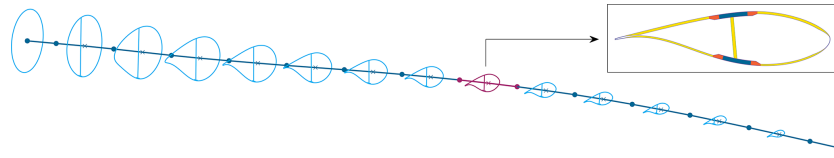


Figure 1. Exemplary finite element beam with reduced number of elements and exemplary cross-sectional illustration. The detail shows a cross-sectional BECAS output (Blasques and Stolpe, 2012) as used in the feasibility study (Noever-Castelos et al., 2021).

where $V[\mathbb{E}(y|x_{\sim i})]$ describes the variance of all expectations over the range where x_i is not included. If the model is purely additive for a particular parameter, the corresponding total Sobol' index should be equal to the first-order index. While the total index does not provide the information about which interaction is significant, it does identify if any interaction exists, with the benefit that it is computed alongside the first-order Sobol' index without any significant additional computational effort.

For a multivariate function with multiple outputs $(y_1, \dots, y_m) = f(x_1, \dots, x_n)$, Eqs. (1) and (2) can be expressed, respectively, as

$$S_{ij} = \frac{V[\mathbb{E}(y_j|x_i)]}{V(y_j)}, \quad (3)$$

$$S_{T_{ij}} = 1 - \frac{V[\mathbb{E}(y_j|x_{\sim i})]}{V(y_j)}. \quad (4)$$

2.2 Rotor blade finite element beam model

The necessary model generation and variation are performed with the model creator MoCA (Noever-Castelos et al., 2022) and its interface to BECAS (Blasques and Stolpe, 2012) to create cross-sectional beam properties, which are assembled into a finite element beam (FE beam) and evaluated in ANSYS Mechanical (ANSYS Inc., 2021a). We will be performing the analysis on the DemoBlade of the SmartBlades2 project (SmartBlades2, 2016–2020). Figure 1 depicts a coarse version of the FE beam used in this study. In contrast to this simplified visualization in Fig. 1, the applied FE beam model is built of 50 three-dimensional linear beam elements (BEAM188) (ANSYS Inc., 2021a) with higher mesh density to the root and tip sections of the blade, where greater geometrical and material changes are expected. Thus, the finite element model consists of 51 nodes (N_{FE}). The input parameter selection of Noever-Castelos et al. (2021) was slightly expanded to cover more material properties, which will be discussed in detail later. The input parameter selection spans a space with a maximum dimension of $\mathbb{D}_{CS} = 33$ for each cross-section, though varying these for each of the 50 cross-sections would result in $\mathbb{D}_{tot} = 1650$ parameters. Assuming a smooth variation in each parameter over the radius, Akima splines (Akima, 1970) were introduced to represent the parameter variation along the blade. An exemplary spline is depicted in Fig. 2. The spline is built based upon

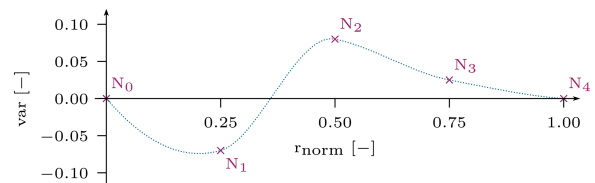


Figure 2. Exemplary variation spline with five nodes.

five equidistant nodes that may vary in the y direction within the given variation range of the parameter. The number of spline nodes can be chosen arbitrarily; however, a high number increases the computational costs (more updating parameters) and can lead to collinear behavior if the nodes are too near, whereas a low number reduces the flexibility to adapt to short-distance changes. For this study the number were chosen based on experience as a trade-off between computational costs and a sufficient approximation of a global parameter variation.

Table 1 summarizes all the investigated input parameters x_i and corresponding properties. Moreover, Table 1 lists the number of spline nodes with their respective normalized radial range and variance limits for each property. In this feasibility study, we consider the most significant independent elastic properties for each material – the density ρ , Young's modulus E_{11} , the shear modulus G_{12} , and Poisson's ratio ν_{12} – which may be varied over all five nodes in a range of $\pm 10\%$. Here, we have neglected all thickness-related elastic constants, i.e., parameters including the index/direction 3 and E_{22} , as these parameters offer no significant contribution to the stiffness terms of the beam cross-sectional properties according to Hodges (2006) and Noever-Castelos et al. (2021). Since foam is modeled as an isotropic material, only two independent elastic properties E and G and the density ρ are considered. In addition to the material properties, the division points are also varied. These subdivide the shell in the cross-sectional direction into different sections with a constant material layout or define sub-component positions such as the web or adhesive (Noever-Castelos et al., 2022). The division point parameters P are allowed to vary on the three mid-nodes by the given absolute range. The root and tip nodes cannot be varied due to model generation issues within MoCA; thus the variance for node N_0 and N_4 will be kept at zero, similarly to in Fig. 2. All applied varia-

Table 1. Input feature list analyzed in this study. Each feature and property builds a distribution spline based on the given number of equidistant nodes within the given normalized radial range of the blade. Each node value may then vary in the listed variance range.

Parameter	Property	Nodes	Norm. range	Variance
UD	$\rho, E_{11}, G_{12}, \nu_{12}$	5	[0, 1]	$\pm 10\%$
Biax45°	$\rho, E_{11}, G_{12}, \nu_{12}$	5	[0, 1]	$\pm 10\%$
Biax90°	$\rho, E_{11}, G_{12}, \nu_{12}$	5	[0, 1]	$\pm 10\%$
Triax	$\rho, E_{11}, G_{12}, \nu_{12}$	5	[0, 1]	$\pm 10\%$
Flange	$\rho, E_{11}, G_{12}, \nu_{12}$	5	[0, 0.1]	$\pm 10\%$
Balsa	$\rho, E_{11}, G_{12}, \nu_{12}$	5	[0, 1]	$\pm 10\%$
Foam	ρ, E, G	5	[0, 1]	$\pm 10\%$
$P_{SS,TE,offset}$	Location	3	[0.25, 0.75]	± 10 mm
$P_{SS,Mid,spar\ cap}$	Location	3	[0.25, 0.75]	± 15 mm
$P_{SS,LE,offset}$	Location	3	[0.25, 0.75]	± 10 mm
$P_{PS,TE,offset}$	Location	3	[0.25, 0.75]	± 10 mm
$P_{PS,Mid,spar\ cap}$	Location	3	[0.25, 0.75]	± 15 mm
$P_{PS,LE,offset}$	Location	3	[0.25, 0.75]	± 10 mm

UD: uni-directional layer; Biax45°: biaxial layer with $-45, +45^\circ$; Biax90°: biaxial layer with $0, 90^\circ$; Triax: triaxial layer with $-45, 0, 45^\circ$; SS: suction side; PS: pressure side; TE: trailing edge; LE: leading edge.

tions are approximately twice the permitted manufacturing tolerances (Noever-Castelos et al., 2021) in order to assure some flexibility of the inverse model. Summing up all parameters and nodes, the problem spans a parameter space of $\dim(x) = 153$. The sensitivity study is conducted based on the Python package ‘‘SALib’’ (Herman and Usher, 2017) and a random sampling dimension of $n = 2^9 = 512$ samples. SALib uses the quasi-random sampling with a low-discrepancy sequence technique from Saltelli et al. (2008) for the sensitivity analysis. To compute the Sobol’ index, the algorithms require a variation in each input feature individually for each of the n samples, which results in a total sample size of $n_{total} \cdot (\dim(x) + 2) = 79\,360$ to compute the first-order and total Sobol’ indices. The sensitivity study as well as the updating process is based on the modal beam response y , as described in Gundlach and Govers (2019), in a free–free and a clamped scenario, which are comparable to an elastic suspension of the blade and a fixation of its root to a test rig, respectively. In each case, the first 10 eigenmodes are extracted, excluding the rigid body motion modes in the free–free scenario. For all 10 mode shapes of each configuration (free–free and clamped), the natural frequency and the three deflections and three rotations of each finite element beam node N_{FE} are saved. These are collected in a response matrix with $\dim(y) = (10 + 10) \cdot (1 + 6) = 140$ columns. Throughout this paper, input parameters and model responses will also be referred to as input and output features or conditions, respectively.

2.3 Feature subspace selection with Sobol’ indices

After computing the first-order and total Sobol’ index S_{ij} and S_{Tij} , respectively, for each input feature x_i and output feature y_i at every N_{FE} position, we obtain a matrix of size

$140 \times 51 \times 153$, i.e., $\dim(y) \times \dim(N_{FE}) \times \dim(x)$. For the subspace selection we follow two selection methods:

1. by computing the maximum-appearing first-order Sobol’ index of each input feature and comparing it to a threshold;
2. by performing singular value decomposition (SVD) on the total Sobol’ sensitivity matrix to identify the most relevant contributions and mapping these back onto the input feature with a QR factorization with column pivoting (Chakroborty and Saha, 2010; Olufsen and Ottesen, 2013).

The selected subspaces are merged into a final subspace, which is applied for the model updating process.

For the first selection method the sensitivity matrix containing the first-order Sobol’ index is condensed into a single maximum value $S_{max,i}$ for each input feature x_i . Therefore, it is reduced to identify relevant input features y by computing the maximum value along the other non-corresponding dimensions, i.e., dimensions 2 and 3. Subsequently, an arbitrary threshold S_{thld} is defined to reject all features with a lower maximum index $S_{max,i}$. By this, we aim to consider only features which have a significant impact during at least one event at one location, thus containing enough information for the updating process. Based on experience, we have chosen $S_{thld} = 0.1$.

The second method follows a combination of SVD and QR factorization on the sensitivity matrix of the total Sobol’ index according to Chakroborty and Saha (2010) for a given set of n input parameters x . Here each mode shape is analyzed individually. Therefore, the sensitivity matrix is divided and reshaped; the first dimension, i.e., the 6 DOFs plus frequency, and second dimension, i.e., the node positions N_{FE} , are flattened, while the third dimension, i.e., input features, is kept yielding an $(m \times n)$ matrix. Given this individual total Sobol’ sensitivity matrix S_T for each mode shape, the singular value decomposition according to Golub and van Loan (2013) is

$$S_T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \quad (5)$$

\mathbf{U} and \mathbf{V} denote the left and right singular vector matrices, each column corresponding to the singular values in $\mathbf{\Sigma} = \text{diag}\{s_1, s_2, \dots, s_p\}$ with $p = \min(m, n)$. According to Chakroborty and Saha (2010), the criterion *percentage of energy explained by the singular values* is used to identify the g most relevant features. The percentage of energy P_{ex} is calculated as the normalized cumulative sum of the singular values:

$$P_{ex} = \frac{\sum_{i=1}^g s_i^2}{\sum_{i=1}^p s_i^2} \cdot 100\%. \quad (6)$$

The number of relevant singular values g is equal to the highest number g complying with $P_{ex} \leq 99\%$. The rest of the singular values, $p - g$, only contribute to 1% of the total energy and are therefore insignificant for the result.

A subsequent QR factorization with column pivoting (Golub and van Loan, 2013) is used according to Olufsen and Ottesen (2013) and Chakroborty and Saha (2010) to extract the order of the original input vector \mathbf{x} , by sorting the columns of the left singular vector matrix \mathbf{V} of size $n \times n$ in order of maximum Euclidean norm in successive orthogonal directions:

$$\mathbf{V}^T \mathbf{P} = \mathbf{Q}\mathbf{R}. \tag{7}$$

Here, \mathbf{Q} is a matrix with orthonormal columns, \mathbf{R} is an upper triangular matrix, and \mathbf{P} is the permutation matrix. In this particular case of a square matrix \mathbf{V} , all matrices are of the same dimension as \mathbf{V} . The permutation matrix \mathbf{P} is finally applied to the input parameter vector \mathbf{x} to re-sort the vector according to sensitivity significance:

$$\mathbf{x}_s = \mathbf{x}^T \cdot \mathbf{P}. \tag{8}$$

The sorted input vector \mathbf{x}_s is then reduced to the first g entries, representing the most significant parameters for the analyzed mode shape following the criterion explained above. After computing all x_s values for each mode shape, they are all merged into a final set of input parameters determined to be relevant for at least one mode shape. With this SVD-QR method applied to the total Sobol' indices matrix, the authors tried to identify parameters that are significant either on their own or in interaction with others. However, the significance is not measured as the maximum value on one occasion, such as in the first method, but rather contributes substantially on average over a complete mode shape.

Both methods lead to the 49 selected features depicted in Table 2 with their respective $S_{\max,j}$ values and a checkmark showing the selection by the SVD-QR method.

When analyzing the rejections, it has to be noted that all structural properties are condensed to cross-sectional beam properties. That means, for example, Biax45° as a face layer of the shear web is typically located near the elastic and gravitational center of the cross-sections and thus does not contribute in excess to the mass inertia according to the Steiner theorem or to the overall bending stiffness (Gross et al., 2012). Consequently, a variation in ρ_{Biax45} and $E_{11,\text{Biax45}}$ will not significantly impact the modal response of the beam model. However, its shear modulus $G_{12,\text{Biax45}}$ does have an impact when dealing with the shear forces from flap-wise loading. Regarding foam and balsa as sandwich core materials, the stiffness contribution to the sandwich panels is approximately 1% compared to the GFRP (glass-fiber-reinforced plastic) face sheets, and this makes their variations neglectable, while the mass contributions depending on the layup can reach up to 66%–100%, which is why a few of the density spline nodes are kept. Summarizing the sensitivity analysis reduced the input feature space to $\dim(x_{\text{sel}}) = 45$,

Table 2. Selected feature list from sensitivity study with the features' respective maximum first-order Sobol' indices $S_{\max,j}$ (values shown in bold meet the given threshold $S_{\text{thd}} = 0.1$) and the selection mark for the SVD-QR method.

Feature	$S_{\max,j}$	SVD	Feature	$S_{\max,j}$	SVD	Feature	$S_{\max,j}$	SVD	Feature	$S_{\max,j}$	SVD
ρ_{UD,N_1}	0.248	✓	$G_{12,\text{Biax45},N_3}$	0.149	✓	ρ_{Trax,N_3}	0.462	✓	G_{12,Trax,N_3}	0.593	✓
ρ_{UD,N_2}	0.381	✓	ρ_{Biax90,N_3}	0.240	✓	ρ_{Trax,N_4}	0.804	✓	G_{12,Trax,N_4}	0.343	✓
ρ_{UD,N_3}	0.278	✓	ρ_{Biax90,N_4}	0.109	✓	E_{11,Trax,N_0}	0.312	✓	ρ_{Foam,N_2}	0.214	✓
E_{11,UD,N_0}	0.109	✓	$E_{11,\text{Biax90},N_0}$	0.116	✓	E_{11,Trax,N_1}	0.375	✓	$\rho_{\text{PSS,Mid,spat cap},N_0}$	0.669	✓
E_{11,UD,N_1}	0.434	✓	$E_{11,\text{Biax90},N_1}$	0.142	✓	E_{11,Trax,N_2}	0.485	✓	$\rho_{\text{PSS,Mid,spat cap},N_1}$	0.620	✓
E_{11,UD,N_2}	0.432	✓	$E_{11,\text{Biax90},N_2}$	0.132	✓	E_{11,Trax,N_3}	0.351	✓	$\rho_{\text{PSS,Mid,spat cap},N_2}$	0.087	✓
E_{11,UD,N_3}	0.371	✓	$E_{11,\text{Biax90},N_3}$	0.218	✓	E_{11,Trax,N_4}	0.527	✓	$\rho_{\text{PSS,Mid,spat cap},N_3}$	0.413	✓
$G_{12,\text{Biax45},N_0}$	0.099	✓	$G_{12,\text{Biax90},N_3}$	0.112	✓	G_{12,Trax,N_0}	0.371	✓	$\rho_{\text{PSS,Mid,spat cap},N_4}$	0.485	✓
$G_{12,\text{Biax45},N_1}$	0.224	✓	ρ_{Trax,N_1}	0.291	✓	G_{12,Trax,N_1}	0.607	✓	$\rho_{\text{PSS,Mid,spat cap},N_5}$	0.044	✓
$G_{12,\text{Biax45},N_2}$	0.262	✓	ρ_{Trax,N_2}	0.211	✓	G_{12,Trax,N_2}	0.521	✓	$\rho_{\text{PSS,Mid,spat cap},N_6}$	0.331	✓
									ρ_{Balsa,N_1}	0.015	✓
									ρ_{Foam,N_2}	0.207	✓
									$\rho_{\text{PSS,Mid,spat cap},N_7}$	0.162	✓
									$\rho_{\text{PSS,Mid,spat cap},N_8}$	0.669	✓
									$\rho_{\text{PSS,Mid,spat cap},N_9}$	0.433	✓
									$\rho_{\text{PSS,Mid,spat cap},N_{10}}$	0.458	✓
									$\rho_{\text{PSS,Mid,spat cap},N_{11}}$	0.491	✓
									$\rho_{\text{PSS,Mid,spat cap},N_{12}}$	0.549	✓
									$\rho_{\text{PSS,Mid,spat cap},N_{13}}$	0.433	✓

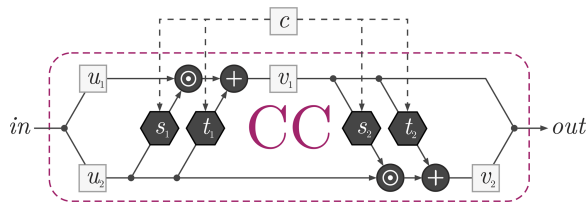


Figure 3. The conditional coupling blocks (CCs) with their embedded sub-network s_1 , t_1 , s_2 , and t_2 . This CC architecture can easily be inverted (Ardizzone et al., 2019b).

approximately 30 % of $\dim(x)$. The output features were all kept according to the feature selection approach. However, a reduced set of radial positions can be applicable as the intrinsic information might be repeated in neighboring N_{FE} . This repeated information does not necessarily improve the updating results but reduces the computational performance. Therefore, the output of each third node is selected, ending up with a radial output dimension of $\dim(N_{FE,sel}) = 17$. Thus, the final dimension for the model updating process of the input feature space is $\dim(\text{input}) = \dim(x_{sel}) = 45$ and of the output feature space is $\dim(\text{output}) = \dim(N_{FE,sel}) \times \dim(y) = 17 \times 140$.

3 Invertible neural network architecture

Before proceeding to the model updating process, it is necessary to define the invertible neural network architecture. Similarly to Noever-Castelos et al. (2021), this work will build on conditional invertible neural networks (cINNs) (Ardizzone et al., 2019b) implemented in FrEIA – the Framework for Easily Invertible Architectures (Visual Learning Lab Heidelberg, 2021). A basic cINN consists of a sequence of *conditional coupling blocks* (CCs), as shown in Fig. 3. Each of these represents affine transformations that can easily be inverted. The embedded *sub-networks* s_1 , t_1 , s_2 , and t_2 embody the trainable functions of this type of artificial neural network.

These sub-networks stack the conditions c and the input slice u_2 or v_1 and transform them for further processing. The stacking necessarily requires similar spacial dimensions of c and u_2 or v_1 , respectively. For a further brief introduction to cINNs with a topic-related application, please refer to Noever-Castelos et al. (2021). A more in-depth explanation can also be found in Ardizzone et al. (2019b, 2018).

After an extensive hyperparameter study, the presented investigation applies the network depicted in Fig. 4. Hyperparameters describe the network or architecture parameters of artificial neural networks, like the number of layers or perceptrons. It consists of a cINN (blue) with a sequence of 15 CCs, grouped into clusters of 3. This cINN transforms between the beam input x and the *latent space* z . However, unlike the underlying feasibility study of Noever-Castelos et al.

(2021), an additional feed-forward network is implemented, referred to as a *conditional network* (orange). The idea is to preprocess the raw conditions c , i.e., beam responses, before passing them to the sub-networks in the CCs. It is trained in conjunction with the cINN to extract relevant feature information optimally for each stage. The conditional network architecture is inspired by Ardizzone et al. (2019b) and should extract higher-level features of c to feed into the sequential CCs, which, according to Ardizzone et al. (2019b), should relieve the sub-networks from having to relearn these higher-level features each time again. With a conditional beam response c of shape $\dim(c) = \dim(N_{FE,sel}) \times \dim(y)$, the conditional network applies 1D convolutions (conv 1D) to process the data, which gradually increase in size to progressively extract higher-level features, which are fed into the different clusters of the cINN.

In general, the beam input would also be available in a 2D shape (property \times spline nodes), though the feature selection of the sensitivity analysis reduced the splines irregularly. Thus, a 2D shape cannot be maintained anymore, as not all splines have the same number of nodes. Therefore, the selected beam input x for the updating process going into the cINN is flattened to a vector and is not present in a 2D shape, as for example the beam response c . A consequence is that the sub-networks cannot make use of convolutional layers but have to include feed-forward layers. However, this will not have any significant impact on the result. As mentioned before, the conditions and input features are stacked in the sub-networks, which thus need a similar spacial shape. Consequently, the conditional network has to flatten the shape to a vector for each output in order to agree with the input shape in the sub-networks. Before flattening the output, the conditional network activates the convolutional layer output with a parametric rectified linear unit (PReLU) (He et al., 2015) and halves the dimension with an average 1D pooling layer (Chollet, 2018) (avg. pool 1D). After flattening, the dimension is additionally reduced with a fully connected layer (fc-layer) to subsequently relieve the sub-network's computation.

Within the cINN, the CCs are clustered into groups, which are then each fed by the progressively processed conditions c . All sub-networks have one hidden fc-layer, followed by a batch normalization to improve generalization and a PReLU (Chollet, 2018) activation layer, as depicted in Fig. 5. As previously explained the conditional network processes the conditions c and has five outputs at different stages of the processing. Each of these outputs is fed into a cluster of three CCs. The configuration for each cluster and the corresponding hyperparameters for the conditional network, cINN, and sub-networks are summarized in Table 3.

The training is performed with an AdaGrad optimizer (Duchi et al., 2011) and an initial learning rate of 0.3, which is gradually decreased throughout the training process. The optimization minimizes the negative logarithmic likelihood (NLL) given in Eq. (9) in order to match the model's

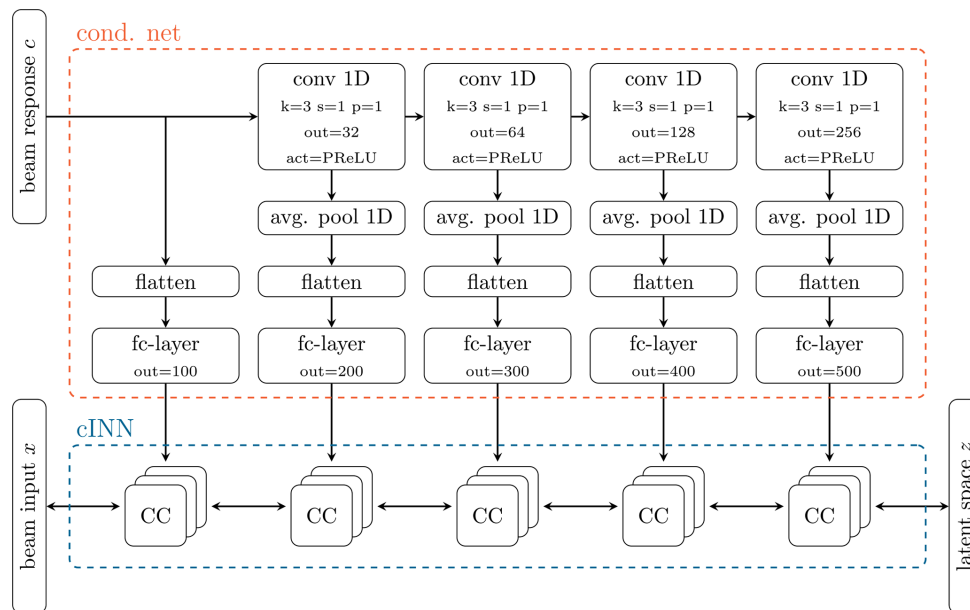


Figure 4. Conditional invertible neural networks (cINNs, in blue frame) with sequentially connected conditional coupling blocks (CCs). The conditional feed-forward network (cond. net, in orange frame) preprocesses the condition y with 1D convolutional layers and PReLU (parametric rectified linear unit) activations. Average 1D pooling is performed on the output before it is flattened and reduced in dimensions with a fully connected layer (fc-layer) to be then fed into the sub-networks of the CCs. The convolutions gradually increase in size in order to progressively extract higher-level features from the condition c .

Table 3. Hyperparameter set of the complete network, including conditional network, conditional invertible neural networks (cINNs), and sub-network. The cINN is divided into five clusters, for which the hyperparameters are listed separately. In Cluster 1, the conditions are directly fed into the conditional coupling blocks (CCs), without a prior convolutional layer (see Fig. 4).

			Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Conditional network	Conv 1D	kernel k		3	3	3	3
		stride s		1	1	1	1
		padding p		1	1	1	1
		out chan. "out"		32	64	128	256
	Activation			PReLU	PReLU	PReLU	PReLU
Average 1D pooling	kernel k	stride s		2	2	2	2
		padding p		2	2	2	2
				0	0	0	0
Flatten		✓	✓	✓	✓	✓	
Fully connected	nodes	100	200	300	400	500	
cINN	Conditional coupling block (CC)		3	3	3	3	3
Sub-network	Fully connected	nodes	400	500	600	700	800
	Batch normalization		✓	✓	✓	✓	✓
	Activation		PReLU	PReLU	PReLU	PReLU	PReLU

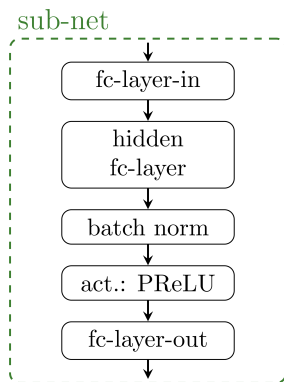


Figure 5. Sub-network with one hidden fully connected layer (fc-layer), batch normalization, and a PReLU activation layer. Each conditional coupling block (CC) has such a sub-network embedded.

posterior prediction of $p_x(x|y)$ with the true posterior of the inverse problem (Noever-Castelos et al., 2021).

$$\begin{aligned} \mathcal{L}_{\text{NLL}} &= \mathbb{E}[-\log(p(x_i | y_i))] \\ &= \mathbb{E}\left[\frac{\|f(x_i; y_i)\|^2}{2} - \log \|\det(J_i)\| \right] + \text{const.} \quad (9) \end{aligned}$$

4 Model updating of a rotor blade beam model

Having selected the significant features with the sensitivity analysis and defined the cINN architecture, we will now move on to the model updating process and its evaluation. Therefore, the workflow of the cINN is briefly explained along with the schematic view of the transformations performed by the cINN in Fig. 6. The presented cINN in Sect. 3 is trained and tested with sample sets of input features x and their corresponding conditions c in the form of the modal beam responses as described in Sect. 2. The concept and training of the cINN are based on Bayes' theorem to infer a posterior distribution $p_x(x|c)$ from a set of conditions c . Therefore, the cINN learns the conditioned transformation from the posterior distribution $p_x(x|c)$ onto the latent distribution $p_z(z)$, as depicted in Fig. 6. This mapping can be achieved through maximum likelihood training. The training is performed over 150 epochs, i.e., training iterations, with a samples size of 30 000 training samples in order to minimize the negative log likelihood \mathcal{L}_{NLL} (given in Eq. 9). For a more detailed description of the inherent method of cINNs please refer to Noever-Castelos et al. (2022) or Ardizzone et al. (2019a). Additionally a sample set of 5000 test samples, which have not been seen by the cINN during its training, are used for validating and testing the cINN after the training. All input features are always sampled randomly and independently but at the same time in order to span the complete parameter space. However, only features selected by the sen-

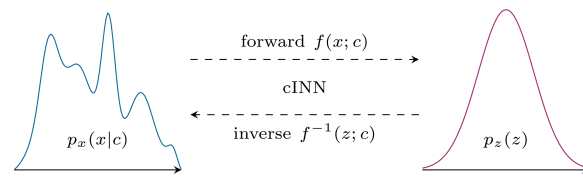


Figure 6. Schematic view of the transformation between the input features x and the latent space z for a given condition c performed by the conditional invertible neural network (Noever-Castelos et al., 2021).

sitivity study (see Table 2) are passed on to the cINN as the other parameters are identified to be less relevant.

As the cINN is trained to map the input features x into a normally distributed latent space $p_z(z)$, during the inverse evaluation the process is reversed: the latent space is sampled from a Gaussian normal distribution (e.g., 50–100 samples), which the cINN then transforms along with the beam response as condition c to the posterior prediction of the input features. This prediction results in a distribution for each input feature $p_x(x|y)$ as depicted in Fig. 6. In order to generalize the data for the training process and make them more comparable for the evaluation, all input features and conditions are standardized to zero mean and a standard deviation of 1 over the complete training set. The necessary scaling factors are additionally saved in the cINN to transform back and forth any input features or conditions used in the cINN besides the training process. Consequently, all features and conditions during the evaluation of the cINN are related to the complete training set's mean and standard deviation. Generally, the posterior predictions are also depicted with respect to their ground truth, i.e., target value of the sample, to align multiple samples for enhanced comparison.

This section first analyses the overall updating results of the model. The identified inference ambiguities are then highlighted and discussed before the model is checked against its robustness to noisy conditions c_{noisy} . Based on the predicted posterior distribution of the input features $p(x|y)$, a resimulation analysis is performed where the updated parameters are used to feed the physical model and calculate the beam response in order to check the quality of the updating and sensitivity analysis results. Finally, a method for avoiding the computational intensive sensitivity analysis is presented.

4.1 General analysis of the updating results

In the first instance the posterior distributions have to be examined. Figure 7 shows as an example the predicted posterior distribution of four different input features as a histogram and fitted Gaussian distribution. The ground truth on the x axis represents the real value used to generate the sample, while the distribution is obtained from the cINN. For the further analysis, the type of distribution must be known in advance

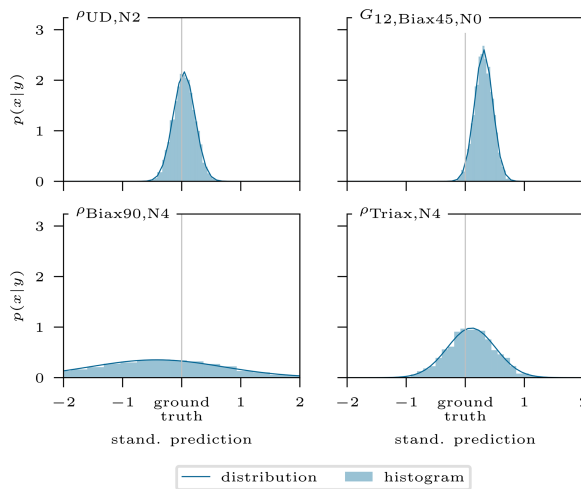


Figure 7. Conditional invertible neural network's standardized posterior prediction distributions $p(x|y)$ for four input features of one example. Plotted as a histogram and fitted Gaussian distribution.

for it to be possible to apply the correct metric, e.g., mean or median. In this case of a Gaussian distribution, the mean is the most significant value and will thus be applied in this study to reduce the posterior prediction distribution to a single value accompanied by the standard deviation as a measure of uncertainty.

By shifting the former x axis from Fig. 7 onto the y axis and reducing the distributions to their mean and standard deviation, as stated before, we obtain the graphs depicted in Fig. 8 for the same exemplary sample but with all updated parameters. Most values range close to their ground truth value and with a narrow distribution, which is desired. For some input features, e.g., $\rho_{\text{Biax}90,N_4}$, the prediction is less accurate. However, the overall posterior prediction in this example is very good, as approx. 70% of the predictions are within a range of ± 0.05 (standardized scale) of the ground truth.

After having checked the results in detail for one exemplary sample, Fig. 9 shows the prediction result of all selected input features for the 5000 test samples. The graphs scatter the standardized mean posterior predictions $\bar{p}(x|y)$ against their corresponding target value from the sample set. Thus, the ideal case would correlate to an exact line with a slope $m = 1$ and an interception $b = 0$. Each graph is equipped with the coefficient of determination R^2 and shows a corresponding regression line with slope m . Approximately 70% of the selected features reach a very satisfying linear correlation with $R^2 > 0.9$ while showing a slope m of approx. 0.9 or higher. For the rest of the discussion we will be sticking with the R^2 value for the accuracy, as the slope accuracy correlates with the R^2 value.

In the following we will create the link between the sensitivity results to enhance comprehension and explain pos-

sible discrepancies. In general a high sensitivity of $S_{\max,i} > 0.35$ leads to a high prediction accuracy ($R^2 > 0.9$). A second major metric to fully understand the prediction accuracy is the cross-correlation of the input features, which reveals collinearities within the physical model. These present a problem for the inversion of the model as the output response of the physical model is ambiguous and can be traced back to different combinations of input features. However, this will be addressed in Sect. 4.2. Input features that do not have any substantial cross-correlation but high $S_{\max,i}$ reach prediction accuracies of $R^2 \approx 1.0$, e.g., all spar cap position points $P_{\text{Mid,sc}}$ or Young's modulus of the UD material $E_{11,\text{UD}}$. For instance, ρ_{Flange,N_1} has one of the highest sensitivity index values, $S_{\max,i} = 0.62$, but a comparably poor prediction accuracy of $R^2 = 0.82$. This fact is due to a strong collinearity with the input features ρ_{Flange,N_0} and ρ_{Flange,N_2} . In contrast, the feature $G_{12,\text{Biax}45,N_0}$ has a low sensitivity $S_{\max,i} = 0.1$ but an excellent prediction accuracy of $R^2 = 1.0$. The reason for this is that this feature does not show any collinearity to other features. Although the $S_{\max,i}$ is low, according to the first-order Sobol' index matrix it has at three nodes N_{FE} the second- and third-highest contribution of all input features for a particular mode shape and DOF, reaching a magnitude of 50%–75% of the maximum value for that DOF. That shows the powerful capability of the cINN to learn the mapping of an input feature to only very few output features out of the complete response data. Table 4 completes this list of examples with the most striking discrepancies of the sensitivity index and prediction accuracy of the input features. Hence, the sensitivity analysis is a good indication to detect a significant parameter subspace for the model updating, though high sensitivities do not directly promise highly accurate inverse prediction.

4.2 Intrinsic model ambiguities

Ambiguities can originate from different sources, such as little significant responses or modeling issues (Ardizzone et al., 2019a). Noever-Castelos et al. (2021) revealed some intrinsic model ambiguities of counteracting density values of the Biax90° and Triax layer in the blade cross-section. This was also handled by the cINN in this study, although it was only checked for the two spline nodes N_3 and N_4 as these coincide in the feature selection. The results are depicted in Fig. 10, showing the standardized mean posterior prediction for the 5000 test samples related to their ground truth and the linear regression as well as the corresponding slope m in the label. While the mean posterior predictions at node N_3 were detected rather accurately (see Fig. 8), i.e., representing a circular area in Fig. 10, the values of node N_4 spread more and correlate to the plotted regression line.

In addition to the density, another ambiguity was detected in Young's modulus E_{11} of both these materials, shown in Fig. 11 for the nodes N_{0-3} . Here, the correlation of the mean posterior predictions is reasonably well described by

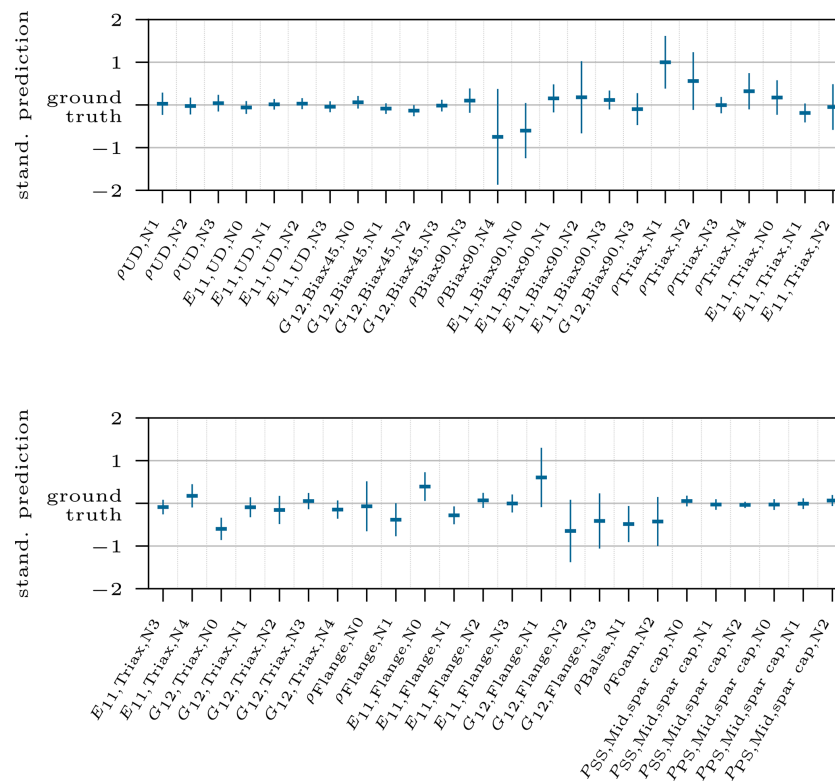


Figure 8. The two graphs show the standardized posterior prediction for all updated input features related to the target value with 1σ standard deviation as error; thus the mean value marks the distance to the target value, i.e., ground truth.

Table 4. Most striking discrepancies of the sensitivity and prediction accuracy of input features.

Feature	$S_{\max,i}$	R^2	XCORR_{\min}	Explanation
E_{11,UD,N_0}	0.110	1.000	-0.663	Low $S_{\max,i}$; however, for two sensors it has the third-highest contribution in a DOF during one mode shape. The values reach a magnitude of 66 % and 50 % of the maximum value in their corresponding DOF.
$G_{12,Biax45,N_0}$	0.100	1.000	-0.179	Low $S_{\max,i}$; however, for three sensors it has the second- and third-highest contribution in a DOF during one mode shape. The values reach a magnitude of 75 %, 55 %, and 53 % of the maximum value in their corresponding DOF.
$G_{12,Biax45,N_3}$	0.149	1.000	-0.383	Low $S_{\max,i}$; however, for one sensor it has the third-highest contribution in a DOF during one mode shape. The value reaches a magnitude of 83 % of the maximum value in its corresponding DOF.
ρ_{Triax,N_1}	0.292	0.790	-0.537	Mid- $S_{\max,i}$; mixed collinearity with ρ_{Biax90,N_1} and ρ_{Balsa,N_1}
ρ_{Triax,N_2}	0.211	0.770	-0.678	Mid- $S_{\max,i}$; mixed collinearity with ρ_{Biax90,N_2} and ρ_{Balsa,N_2}
ρ_{Flange,N_0}	0.214	0.660	-0.952	Mid- $S_{\max,i}$; strong collinearity with ρ_{Flange,N_1}
ρ_{Flange,N_1}	0.620	0.820	-0.952	High $S_{\max,i}$; strong collinearity with ρ_{Flange,N_0} and ρ_{Flange,N_2}
$G_{12,Flange,N_1}$	0.332	0.720	-0.857	Mid- $S_{\max,i}$; strong collinearity with $G_{12,Flange,N_1}$
$G_{12,Flange,N_2}$	0.276	0.690	-0.876	Mid- $S_{\max,i}$; strong collinearity with $G_{12,Flange,N_0}$ and $G_{12,Flange,N_2}$

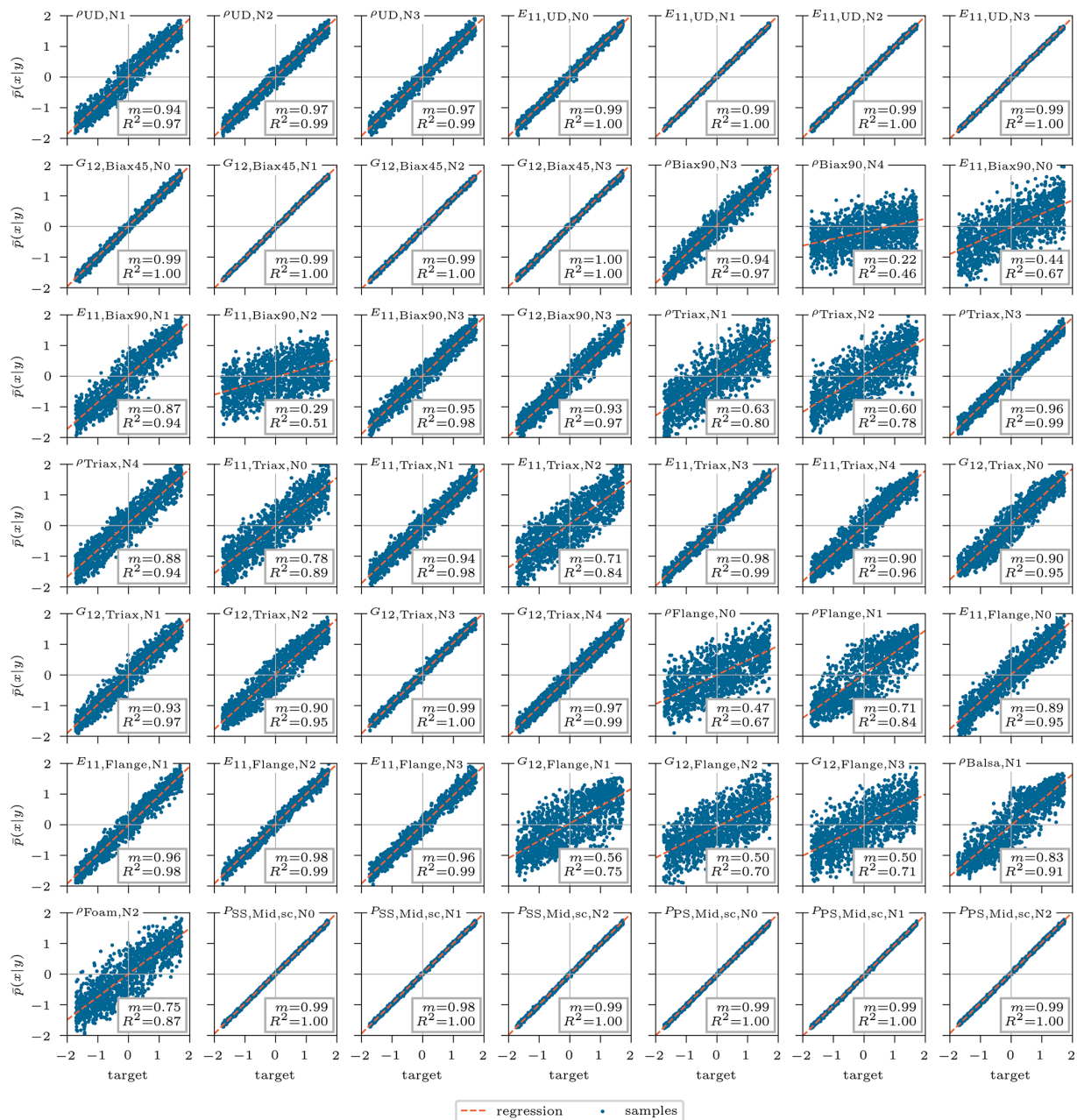


Figure 9. Standardized mean of posterior prediction \bar{p} of the updated inputs over the corresponding target standardized value for the 5000 test samples. The coefficient of determination and a corresponding linear regression line are shown. The corresponding parameter description of the features can be found in Table 2.

the calculated regression lines. Finally, the last correlation was found for the shear modulus $G_{12,N3}$ between the same materials (Fig. 12).

All ambiguities rely on the same fact that the Biax90° and Triax layers appear subsequently in the stacking of the sandwich panels of the blade shell. The stacking

is schematically illustrated in Fig. 13 with a detailed view of the shell, showing the stacking in an exploded view. Together, these layers build the symmetric inner and outer face sheets of the shell, with a layer thickness of $t_{\text{Biax90}} = 0.651$ mm and $t_{\text{Triax}} = 0.922$ mm, the same density $\rho_{\text{Biax90}} = \rho_{\text{Triax}} = 1875$ kg m⁻³

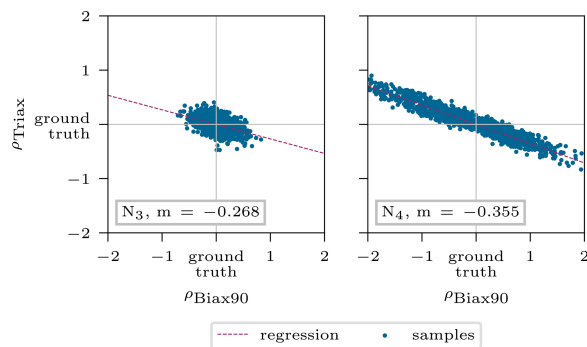


Figure 10. Interaction of density ρ_{Biax90} and ρ_{Triax} describing the intrinsic model ambiguities. The depicted values correspond to the standardized mean posterior prediction for the 5000 test samples.

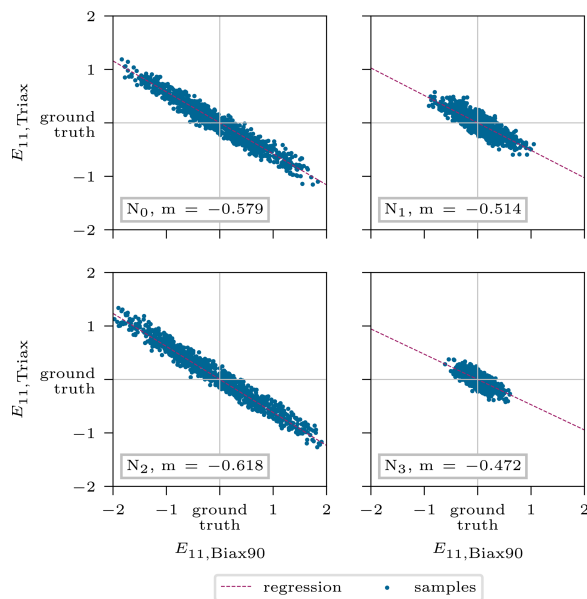


Figure 11. Interaction of stiffness $E_{11,\text{Biax90}}$ and $E_{11,\text{Triax}}$ describing the intrinsic model ambiguities. The depicted values correspond to the standardized mean posterior prediction for the 5000 test samples.

Young's modulus $E_{11,\text{Biax90}} = 26430 \text{ N mm}^{-2}$, and $E_{11,\text{Triax}} = 29873 \text{ N mm}^{-2}$, and shear modulus $G_{12,\text{Biax90}} = 3464 \text{ N mm}^{-2}$ and $G_{12,\text{Triax}} = 6918 \text{ N mm}^{-2}$.

The contributions of the properties to the model behavior must be analyzed for it to be possible to understand these ambiguities further. As described in Sect. 2, a finite element beam model is composed of beam elements containing cross-sectional properties (Blasques and Stolpe, 2012). These basically consist of mass and stiffness terms, which can be directly linked to ρ and E_{11} or G_{12} , respectively (Hodges, 2006). The upcoming deductions follow classical mechanics

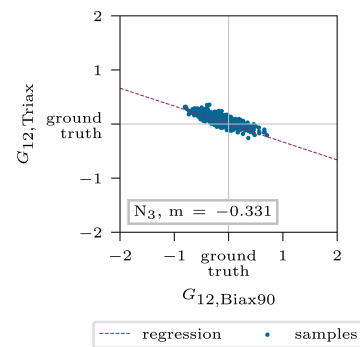


Figure 12. Interaction of shear stiffness $G_{12,\text{Biax90}}$ and $G_{12,\text{Triax}}$ describing the intrinsic model ambiguities. The depicted values correspond to the standardized mean posterior prediction for the 5000 test samples.

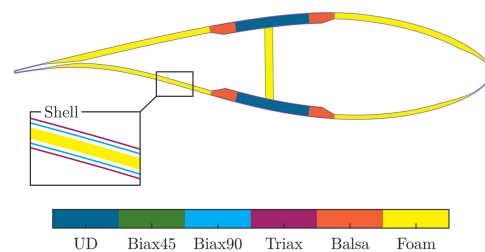


Figure 13. Schematic blade cross-sectional view at a radial position of $r = 12 \text{ m}$ with a detailed explosion drawing of the shell.

theories found for example in Gross et al. (2012). First, considering the mass contribution, we stick with the simplified example of the center of gravity:

$$x_s = \frac{1}{m_{\text{tot}}} \int x^2 dm = \frac{1}{m_{\text{tot}}} \sum x_j^2 m_j, \quad (10)$$

where x_j represents the center of gravity of each component and m_i the corresponding mass. Due to the very thin thickness of both layers and the overall cross-sectional dimension being about 10^3 greater for both materials, it can be assumed that $x_j = x_s$. And by expecting that the cINN correctly predicts the total mass m_{tot} , Eq. (10) yields

$$x_s = \frac{1}{m_{\text{tot}}} \cdot x_s \sum m_j, \quad (11)$$

$$m_{\text{tot}} = \sum m_j = k_{\text{Biax90}} \cdot t_{\text{Biax90}} \cdot \rho_{\text{Biax90}} + k_{\text{Triax}} \cdot t_{\text{Triax}} \cdot \rho_{\text{Triax}}. \quad (12)$$

And this obviously leads to the summation of all individual masses to the total mass, where k represents the number of layers. This of course holds for higher-order moments of mass due to the given proximity of both layers. Thus, a ratio between both materials can be expressed:

$$k_{\text{Biax90}} \cdot t_{\text{Biax90}} \cdot \rho_{\text{Biax90}} : k_{\text{Triax}} \cdot t_{\text{Triax}} \cdot \rho_{\text{Triax}}. \quad (13)$$

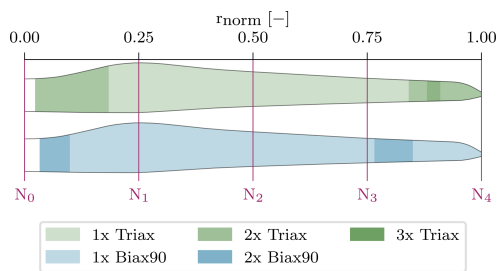


Figure 14. Layup of the sandwich laminate face sheets of the blade shell, consisting of Triax and Biax90°. The inner and outer face sheets are symmetric.

A similar behavior is also found for the stiffness. This is explained in a simplified example for the flexural rigidity of a beam in Eq. (14), which extends with the Steiner theorem to Eq. (15).

$$EI_{\bar{x}} = \sum E_j I_{\bar{x},j} \quad (14)$$

$$= \sum E_j (I_{x,j} + x_s^2 \cdot A_j) \quad (15)$$

Assuming the layers have a rectangular shape, the area moment of inertia is $I_{x,j} = \frac{w \cdot t^3}{12}$, though the width w of the layer is large and the thickness t is 10^{-3} smaller, and thus this term vanishes. With that, Eq. (15) reduces to Eq. (16). As stated before, x_s can be assumed to be constant, and the same holds for the width w_j as in the cross-sectional direction both material layers cover the complete circumference of the blade. This results in the proportionality in Eq. (17):

$$EI_{\bar{x}} = \sum E_j (x_s^2 \cdot A_j) = \sum E_j (x_s^2 \cdot k_j \cdot t_j \cdot w_j), \quad (16)$$

$$EI_{\bar{x}} \propto \sum E_j \cdot k_j \cdot t_j. \quad (17)$$

Similarly to the total mass m_{tot} , we expect the cINN to predict the global $EI_{\bar{x}}$ accurately, and, consequently, we can establish the following ratio for the stiffness:

$$k_{\text{Biax90}} \cdot t_{\text{Biax90}} \cdot E_{\text{Biax90}} : k_{\text{Triax}} \cdot t_{\text{Triax}} \cdot E_{\text{Triax}}. \quad (18)$$

Analog derivations can be made for the shear modulus, which ends up in the following ratio:

$$k_{\text{Biax90}} \cdot t_{\text{Biax90}} \cdot G_{\text{Biax90}} : k_{\text{Triax}} \cdot t_{\text{Triax}} \cdot G_{\text{Triax}}. \quad (19)$$

Figure 14 shows the number of each layer for the respective material along the blade, which corresponds to both the inner and outer face sheet of the shell. The corresponding spline nodes positions are also depicted. Table 5 shows the ratios according to Eqs. (13), (18), and (19) of the different possible stacking options in Fig. 14.

Looking back to the identified ambiguities in Fig. 10 of the density at node N_4 , the linear regression shows a slope of $m = -0.355$. Assuming each spline node contributes to

Table 5. Ratio between Biax90° and Triax layers for density and stiffness contribution, considering different layer constellations.

k_{Biax90}	1	1	1	2	2	2
k_{Triax}	1	2	3	1	2	3
$\frac{k_{\text{Biax90}} \cdot \rho_{\text{Biax90}} \cdot t_{\text{Biax90}}}{k_{\text{Triax}} \cdot \rho_{\text{Triax}} \cdot t_{\text{Triax}}}$	$\frac{0.706}{1}$	$\frac{0.353}{1}$	$\frac{0.235}{1}$	$\frac{1.412}{1}$	$\frac{0.706}{1}$	$\frac{0.471}{1}$
$\frac{k_{\text{Biax90}} \cdot E_{\text{Biax90}} \cdot t_{\text{Biax90}}}{k_{\text{Triax}} \cdot E_{\text{Triax}} \cdot t_{\text{Triax}}}$	$\frac{0.625}{1}$	$\frac{0.312}{1}$	$\frac{0.208}{1}$	$\frac{1.249}{1}$	$\frac{0.625}{1}$	$\frac{0.416}{1}$
$\frac{k_{\text{Biax90}} \cdot G_{\text{Biax90}} \cdot t_{\text{Biax90}}}{k_{\text{Triax}} \cdot G_{\text{Triax}} \cdot t_{\text{Triax}}}$	$\frac{0.354}{1}$	$\frac{0.177}{1}$	$\frac{0.118}{1}$	$\frac{0.707}{1}$	$\frac{0.354}{1}$	$\frac{0.236}{1}$

the variance of half of the space to the left and right of it, the given slope agrees extremely well with the ratio of $k_{\text{Biax90}} = 1$ and $k_{\text{Triax}} = 2$. This corresponds to the stacking shown near the node N_4 in Fig. 14. Due to the poor linear regression of node N_3 in Fig. 10, the slope is not reliable; thus no conclusion can be drawn.

However, the counteracting Young's moduli in Fig. 11 can be very accurately captured by the ratios for most spline nodes. Starting with node N_2 (Fig. 11, bottom left), which is clearly affected by only one layer to the left and right of it (see Fig. 14), the line slope $m = -0.618$ matches the value in Table 5 ($k_{\text{Biax90}} = 1$; $k_{\text{Triax}} = 1$) of 0.625. Node N_0 has a slope of $m = -0.579$, which agrees well with the value corresponding to $k_{\text{Biax90}} = 2$ and $k_{\text{Triax}} = 2$ but tends towards $k_{\text{Biax90}} = 1$ and $k_{\text{Triax}} = 2$, which is also in the scope of this node according to the layup in Fig. 14. Similar behavior is found for node N_1 . Node N_3 does not fully agree with this argumentation, though the point scatters less and the regression line might not be accurate enough. The same holds for the shear modulus in Fig. 12.

As assumed in the derivation of the ratios, we can state that the cINN should correctly predict the total mass and the stiffness contributions in a global manner but suffers from an intrinsic model ambiguity affected by the counteracting densities ρ , Young's moduli E_{11} , and shear moduli G_{12} of the neighboring materials Biax90° and Triax. However, it offers posterior predictions for these features but with a wide distribution expressing the uncertainty in the cINN based on the given ambiguity. Merging both materials to a face sheet material following laminate theory would avoid these ambiguities and improve the prediction qualities for the overall laminate. It is assumed that, based on the relatively low layer thickness, the infusion and therefore the fiber volume fraction of both layers are very similar, so this approach should be valid.

4.3 Model robustness

So far the analysis of this feasibility study was conducted on the exact test sample data; i.e., for a given input sample the corresponding exact output sample is generated with the tool chain MoCA + BECAS + ANSYS. In future studies, this presented method should be applied to real measured data of a blade, and these normally suffer from measurement

uncertainties. It is thus important to analyze the model robustness with respect to a measurement error in the output features. Therefore, an error of 5% as normally distributed random noise is applied to the clean output response of each sample, which is then used as a condition to infer the posterior prediction of the input features. The results are shown in Fig. A1 in the Appendix, comparing the noisy (orange) and the clean (blue) mean posterior predictions $\bar{p}(x|y)$ against their corresponding targets for all 5000 test samples. The graphs show some features that are sensitive to noise, such as $E_{11,Flange,N_{0-3}}$ and $G_{12,Flange,N_3}$. As visually confirmed in Fig. A1, the other features do not show a wider spread (orange) than the original values (blue) and therefore do not suffer from any accuracy loss. Additionally, tests were performed resuming the training of the cINN with noisy conditions in order to improve the prediction quality, though no benefit was identified.

4.4 Resimulation analysis

A resimulation analysis aims to utilize the posterior predictions of the cINN based on the original response to resimulate/recalculate the response with the physical model in order to compare it to the original response used to perform the prediction. For all samples, the correct input features and their corresponding response features are known, which we will be referring to as targets. The target response is used as a condition for the cINN to infer the posterior prediction of the selected input features. From these inferred input features we can create new input splines for each input, as depicted exemplarily in Fig. 15. However, the prediction is not a discrete value but a Gaussian distribution as we have seen before in Sect. 4.1. Additionally, there are nodes that the sensitivity analysis excluded from the updating process; these may take every value within their variation range as they were sampled uniformly. Hence, for each input feature we obtain a range of possible splines as Fig. 15 illustrates. Here, the orange spline represents the target variance of the input parameter and the dark blue area represents the expected value, i.e., the mean prediction from the updated nodes. In the case of the first spline for ρ_{UD} , nodes N_0 and N_4 were excluded from the updating process and can thus take any value in the range of $\pm 10\%$ as we do not have any prediction for them. As such the blue area covers all possible splines a user would take as the result from the model updating process. However, the purpose of this first evaluation of the resimulation analysis is to examine if sampling splines from the given distributions will all lead to appropriate results. Therefore, the 1σ uncertainty displayed in light blue shows the standard deviation of the predicted nodes. In this first analysis, we sample from a uniform distribution for the non-updated nodes (dark blue range) and from a normal distribution for the updated nodes (light blue) to create a spline. This will be done 1000 times for the same given target response of the selected single test sample. Subsequently, these 1000 sets of input splines are

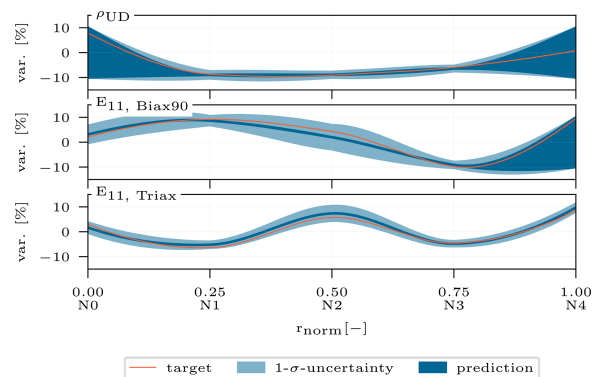


Figure 15. Exemplary inferred spline prediction range for ρ_{UD} , $E_{11,Biax90}$, and $E_{11,Triax}$. The graphs depict the target spline in orange, the mean prediction in dark blue, and the 1σ uncertainty in light blue, for the updated spline nodes.

then used to create the model and calculate its modal response. For the sake of completeness, Table A2 gathers the identified mode shapes of both configurations. The resultant mode shapes of the free–free and the clamped configurations are then compared to the target response with the help of the modal assurance criterion (MAC) (Allemang, 2003).

$$MAC_{ij} = \frac{|\Phi_i \cdot \Phi_j|^2}{|\Phi_i \cdot \Phi_i| \cdot \|\Phi_j \cdot \Phi_j\|} \quad (20)$$

The MAC is the scalar product of two normalized vectors, each representing all the model’s degrees of freedom of a particular mode shape. It is basically an orthogonality check: equal mode shapes reach a value of $MAC = 1$, and a value of $MAC > 0.8$ is already assumed to show good coherence (Pastor et al., 2012). For a multiple number of modes, a MAC matrix summarizes all MAC values of all mode shapes compared against each other.

In our use case, the MAC matrix is computed individually for all responses of the previously generated 1000 samples against the target response. For the free–free configuration, Fig. 16 illustrates the mean value of the MAC matrix over all samples in the top graph. The corresponding standard deviation is depicted below. The main diagonal ideally takes values of $MAC_{ij} = 1$, as the same mode shape of the sample and the target is compared. Additionally, the matrix should be symmetric, as the comparison of $MAC_{ij} = MAC_{ji}$ represents the same two mode shapes. Figure 16 confirms this ideal symmetric matrix structure for the resimulated samples, with mean values $\overline{MAC} > 0.9975$ in the diagonal and extremely low standard deviations of $\sigma_{MAC} < 0.003$. For the clamped configuration, the values on the diagonal are also strikingly close to 1 ($MAC > 0.9960$; $\sigma_{MAC} < 0.005$) and the overall matrix appears symmetric. In this way, sampling from the distribution predicted by the cINN for each selected input feature and arbitrarily choosing a value for the non-updated

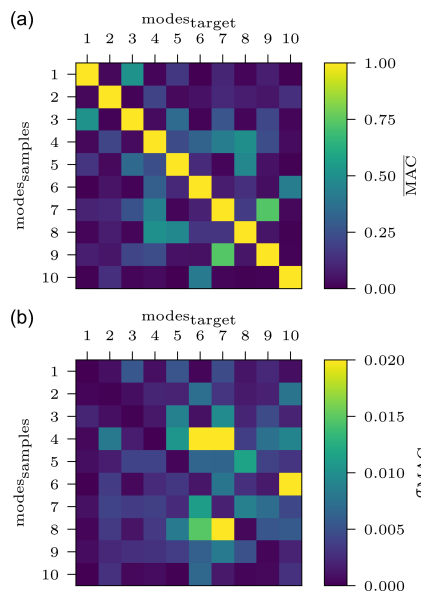


Figure 16. Mean values (a) and standard deviations (b) of the MAC matrix for the free–free modal configuration based on 1000 spline samples inferred for one target response.

values yield an exact coherence of target and computed mode shapes.

After having analyzed a single target sample, the resimulation is expanded to more samples to show the cINN’s general performance. Therefore, posterior predictions for the 5000 test samples of the test set are inferred with the cINN. Contrary to the resimulation case before, only one input is generated for each of the samples by choosing the mean value of the prediction and, in the case of excluded variables, a node value of zero (i.e., no variation). That represents a typical choice a user would make, based on predicted posterior distributions. Figure 17 depicts the mean (horizontal marker) and max and min values (bar) of the diagonal entries of the MAC matrices, which are computed for all samples and both configurations, to compare the resimulated model and its respective target response. Again, all mean values are close to 1 (90 % with a $\overline{MAC} \geq 0.995$), so an overall excellent updating performance can be stated. Single predictions lead to worse results, as depicted by the minimum value (4.3 % of all have a $MAC \leq 0.98$), especially for the higher-order modes, though the MAC value of less than 0.8 is only obtained for the 10th eigenmode of the free–free configuration.

The generally good performance is also confirmed by the predicted corresponding natural frequencies. Figure 18 shows the relative error from the resimulated frequencies to the target frequencies of each mode for both configurations, giving the mean and standard deviation over all resimulated samples. The range of the mean error is $|\bar{e}_f| < 0.25\%$ and the standard deviation $\sigma_{e_f} < 1.50\%$.

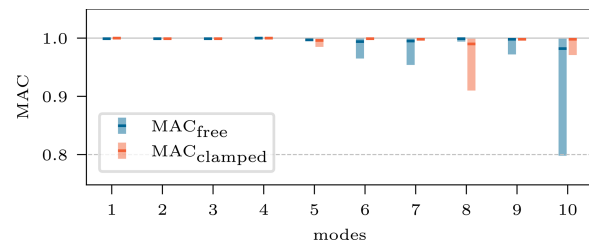


Figure 17. Mean, maximum, and minimum diagonal entries of the MAC matrices computed for 1000 target responses.

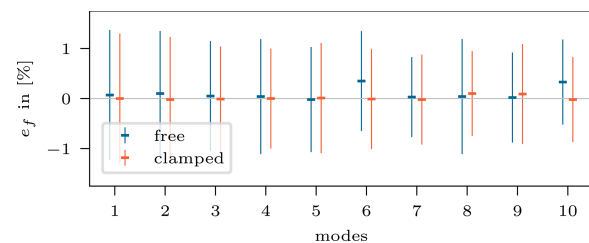


Figure 18. Mean and standard deviation of the natural frequency error e_f computed for 1000 target responses.

The results of the presented resimulation analysis show the following:

1. The counteracting intrinsic model ambiguities discussed in Sect. 4.2 cancel each other out; i.e., the overall shell laminate properties are correctly predicted, although the individual stiffness or density of the layers (Biax90 and Triax) is not predicted accurately. So the cINN still correctly captures the global model behavior with respect to mass and stiffness distribution.
2. As expected, the insensitive and thereby excluded input parameters really do not have an impact on the results and can be chosen arbitrarily (see Fig. 15).
3. The overall cINN updating performance is strikingly good, with on average 90 % of the mode shapes of the resimulated samples showing a $\overline{MAC} \geq 0.995$. The frequencies were recovered with a mean error of $|\bar{e}_f| < 0.25\%$.

4.5 Replacing sensitivity analysis

Similarly to other model updating studies such as Luczak et al. (2014), this work relies on a sensitivity study to reduce the parameter space of the updating problem to significant parameters. This so-called feature selection is performed in this particular investigation with the aforementioned Sobol’ method. A quasi-random sampling with low-discrepancy sequences (Dick and Pillichshammer, 2010) is applied to compute the Sobol’ indices, which is a computationally efficient

and space-efficient sampling method for the sensitivity analysis. However, the sampling set to train the cINN in general has to span a real random sampling space, where all features are varied independently but simultaneously. That means, despite the 79 360 samples for the sensitivity analysis, an additional set of 30 000 samples has to be generated for training purposes and a second variably sized set for validation and testing of the cINN. In total, this results in approximately 115 000 samples and thus model evaluations. This is crucial considering that the model evaluation in general is the computational bottleneck. Although a classical optimization algorithm would also need a feature selection to reduce the updating problem complexity on top of its usual model evaluation number for the optimization process, the overhead of the sensitivity cuts down the computational benefit of the cINN. A single model evaluation from creating the input parameter set to importing the modal response of the model took on average approx. 80 s on a single-core device. We generated the 115 000 samples on a 40-core computing cluster in slightly less than 2.66 d. In contrast, the cINN training for 150 epochs took only 0.67 h on an NVIDIA Tesla P100 GPU.

To reduce the computational sampling time, the idea is to apply the cINN to the full input parameter set x to identify relevant parameters. The cINN implicitly detects irrelevant features by predicting an uncertain posterior distribution, i.e., high standard deviation, due to missing information for the inference in the response. However, Sect. 4 and 4.2 showed that intrinsic model ambiguities lead to wider distributions, without being inaccurate in the global model behavior. This means the respective input parameters should not be rejected due only to a widely distributed posterior prediction. Therefore, we combine three metrics to perform the feature selection on the posterior predictions of the full input parameter set with respect to standardized values:

1. root mean square error (RMSE) of the predicted posterior's mean and target value,
2. standard deviation of the predicted posterior distribution,
3. cross-correlation matrix of the predicted posterior's mean values.

The RMSE should reject features that might have a narrow predicted posterior distribution but do not match the target value. This is more a security or backup metric. The standard deviation is a metric for the confidence of the cINN and should reject features that are not significantly included in the information of the modal beam response. And finally, a cross-correlation matrix should reveal intrinsic model ambiguities from feature interactions, in order to keep the respective features, though the other two metrics would reject them. The cross-correlation matrix of this inverse problem is depicted in Fig. 19. The inputs feat_{40-54} and feat_{60-74} in the matrix correspond to $\rho_{\text{Biax}90, N_0-N_4}$, $E_{11, \text{Biax}90, N_0-N_4}$, and

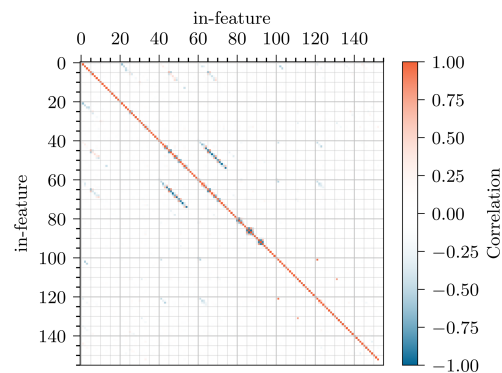


Figure 19. Cross-correlation of all input features based on mean posterior prediction of the 5000 test samples.

$G_{12, \text{Biax}90, N_0-N_4}$ and to $\rho_{\text{Triax}, N_0-N_4}$, $E_{11, \text{Triax}, N_0-N_4}$, and $G_{12, \text{Triax}, N_0-N_4}$, respectively, which show the high negative correlation of the interacting features discussed in Sect. 4.2. This matrix also helps to detect other relevant correlations. Especially nearby nodes of the same feature (e.g., feat_{85-87} , $E_{11, \text{Flange}, N_0-2}$) can counteract each other, as these have to predict in combination the spline behavior in between them; i.e., if one increases, the other has to diminish. Similar behavior was already detected in Bruns et al. (2019).

Similarly to the Sobol' threshold $S_{ij, \text{thld}} = 0.1$, thresholds for the given metrics can be chosen arbitrarily again and rely on experience. In this case we have chosen $\text{RMSE}_{\text{thld}} = 0.5$, $\sigma_{\text{thld}} = 0.5$, and $\text{XCORR}_{\text{thld}, \text{max}} = -0.75$. Table A1 lists all features selected by the sensitivity analysis and the cINN in comparison. The sensitivity analysis selects 49 features, while the cINN includes 54 features. Most of the features agree for both selection methods, except those included in Table 6. The cINN, for example, includes the input features E_{11, UD, N_4} and $G_{12, \text{Balsa}, N_1}$, which can be very well predicted by the cINN but which are not detected by the sensitivity analysis to be significant for the response variations. Additionally, it detects a few highly negative correlating features – $E_{11, \text{Biax}90, N_4}$ and $G_{12, \text{Biax}90, N_0-2,4}$ – which follow the similarly ambiguous behavior shown in Sect. 4.2, counteracting the respective Triax properties. However, the features $\rho_{\text{Triax}, N_{1,2}}$ and ρ_{Foam, N_1} , detected by the sensitivity analysis, were excluded by the cINN, though at least the first two show a significant $S_{\text{max}} > 0.200$.

Finally, this procedure is based on 30 000 samples and the same cINN architecture and hyperparameters. Figure A2 shows the correlation results for all features included in the sensitivity analysis, where the orange scatter represents the prediction with the model trained on the full input set and the blue scatter the prediction by the former model based on the feature selection from the sensitivity analysis. Only very few features show a significant loss in accuracy compared to the original model and most likely for the feature

Table 6. Feature selection discrepancies between both methods – sensitivity analysis (SA) and the cINN-based approach – and their corresponding metrics. All values depicted in bold meet their corresponding threshold and are thus selected by the respective approach.

Feature	SA	S_{\max}	SVD	cINN	RMSE	σ	XCorrmin
$E_{11,UD,N4}$		0.006		✓	0.340	0.354	-0.4407
$E_{11,Biax90,N4}$		0.051		✓	0.913	0.881	-0.9524
$G_{12,Biax90,N0}$		0.040		✓	0.862	0.833	-0.8341
$G_{12,Biax90,N1}$		0.062		✓	0.454	0.374	-0.8889
$G_{12,Biax90,N2}$		0.078		✓	0.941	0.920	-0.986
$G_{12,Biax90,N4}$		0.009		✓	1.014	0.991	-0.9485
$\rho_{Triax,N1}$	✓	0.292	✓		0.648	0.531	-0.5367
$\rho_{Triax,N2}$	✓	0.211	✓		0.652	0.604	-0.6785
$G_{Balsa,N1}$		0.017		✓	0.285	0.230	-0.2985
$\rho_{Foam,N2}$	✓	0.163	✓		0.623	0.538	-0.4732
$\rho_{Foam,N3}$		0.072		✓	0.478	0.483	-0.5273

with a worse prediction quality. Thus, there is no need to perform a second training process with a reduced data set for the sensitivity-free procedure, though the selection of the samples should still reveal the significant parameters of the model. Relying on the same computing resources mentioned above, the overall process in this particular case adds up to a complete computation time of approximately 20h, which corresponds to a reduction of 69%. It also reveals that the cINN can handle a higher number of parameters while extracting the relevant information from the response to predict the significant input features. On account of that, there is no need for a pre-analyzing sensitivity study in future investigations. This gives cINNs a huge advantage over common approaches as discussed in the Introduction. They rely on a sensitivity analysis to identify a significant subspace to reduce the problem dimension. With 30 000 model evaluations for a total of 49 updated features, the cINN is quite efficient. A stochastic updating approach demanded 1200–12 000 evaluations for a simple three-feature updating problem (Augustyn et al., 2020; Marwala et al., 2016). Higher dimensional problems could explode in computational costs for common deterministic approaches, relying even more on an additional preprocessed subspace selection (here, 79 000 model evaluations). However, to the best of the authors knowledge, no model updating was found in the literature for such a high parameter space as presented in this work.

5 Conclusions

The current study aims to extend the feasibility study of model updating with *invertible neural networks* presented in Noever-Castelos et al. (2021) to a more complex and application-oriented level in the form of a Timoshenko beam. The model updating was performed on a global level. This took into account five-noded splines for input feature representation over the blade span of material density and stiffness, as well as layup geometry. The blade response used for the updating process is in the form of modal shapes and

frequencies. The outstanding updating results presented in this study strengthen the conclusion in Noever-Castelos et al. (2021) that invertible neural networks are highly capable of efficiently dealing with wind turbine blade model updating for the given global fidelity level.

In comparison with Noever-Castelos et al. (2021), this investigation increased the model complexity from a single cross-sectional representation to a finite element Timoshenko beam model of the complete blade. The update parameter space was only slightly expanded for the materials to cover the most relevant, independent elastic properties of orthotropic materials. These, however, are varied over the complete blade length with three- to five-noded splines. Moreover, an established, global, variance-based sensitivity analysis with the Sobol' method was performed to determine the relevant update parameters. A total of 49 input parameters were updated based on modal responses of the blade in a free-free boundary configuration and a root-clamped configuration. The applied cINN approximately doubled its depth, and an additional feed-forward network was implemented to preprocess the conditions of the cINN in order to improve the network's flexibility and accuracy.

The result analysis of the predicted parameters shows strikingly high coherence with the target values with R^2 scores over 0.9 for 75% of the updated parameters. The very high updating certainty of the network is reflected in the narrow predicted posterior distributions of the updated parameters. Moreover, this study revealed more intrinsic model ambiguities of material properties (E_{11} , G_{12} , ρ) of the laminate face sheets Biax90° and Triax due to their proximity in the layup. The cINN learns and understands the intrinsic collinearities of the physical model, which result in ambiguous inverse paths. However, the cINN is still not able to distinguish from which parameter the individual contribution comes. Nevertheless, in contrast to a deterministic approach, the user can see how uncertain the cINN is about the prediction due to its wide spreading of affected features' prediction. In future contributions this can be handled by updating a joint density or stiffness variation, instead of individual features. However, the resimulation analysis revealed the modal response of the updated models matches the target results exceptionally well: 90% of the mode shapes of the resimulated samples show a $\overline{MAC} \geq 0.995$ and a mean error in the natural frequencies of $|\bar{\epsilon}_f| < 0.25\%$ over 1000 randomly chosen test samples. Finally, this study presents a method for avoiding the computationally expensive sensitivity analysis by fully exploiting the opportunities of the cINN. For this reason, the full parameter set of $D_{\text{tot}} = 153$ was used for the update process. Thanks to the underlying probabilistic approach of the cINN, a similar set of significant input features was detected from the complete parameter space, based on the predicted posterior distributions and a cross-correlation between the input feature to identify the ambiguities. Thus, the necessary sample number for the complete process was

reduced to 30 000 samples and the computational time by 69 % while maintaining similar outstanding updating results.

Referring back to the three major problems of the approaches studied in the Introduction, the cINN tackles these in the following ways:

1. It has a high computational efficiency in relation to the model complexity, i.e., updating parameter space, and even more by evading computationally expensive sensitivity analysis. The cINN only demanded 30 000 model evaluations (≈ 20 h) for a total of 49 updated features within an original space of 153 features.
2. It makes an inherent probabilistic evaluation, as it follows Bayes' theorem and is trained to minimize the negative log likelihood of the mapping between posterior distribution and latent distribution.
3. A surrogate of the inverted model is represented. By that, the cINN can be evaluated for any given response (in the model boundaries) at practically no additional cost after training. Any other approach is solved only for one particular model response and has to be repeated in the case of a different set of responses.

In conclusion, the feasibility study was highly successfully extended to a full-blade beam model, though with a still limited parameter set. The cINN proved to be extremely capable of performing efficient model updating with a larger parameter space. The physical model complexity in the form of a Timoshenko finite element beam is already at the state-of-the-art level applied in industry. However, to ensure that the cINN learns the complete inverted physical model, it is important that all possibly relevant parameters have to be varied so that the cINN is trained for all circumstances of variations for the model updating. Therefore, ongoing and future investigations should bring this method to a real life application, where the parameter space will span more relevant aspects of blade manufacturing deviations, such as adhesive joints. Moreover, the combined laminate properties of the face sheets might be able to prevent the model ambiguities and even improve the already good prediction accuracy. One possible application scenario could be a final quality control after manufacturing if the response generation can be automated. The benefit would be to find rough manufacturing deviations and even provide individually updated models for each blade, which could for example enhance turbine controls.

Appendix A: Tables and figures

Table A1. Comparison of the feature selection performed by the sensitivity analysis (SA) and directly with the cINN applied to the full input parameter set.

Feature	$S_{max,i}$	SVD	cINN	Feature	$S_{max,i}$	SVD	cINN
$\rho_{UD,N0}$	✓	✓	✓	$E_{11, Triax,N1}$	✓	✓	✓
$\rho_{UD,N2}$	✓	✓	✓	$E_{11, Triax,N2}$	✓	✓	✓
$\rho_{UD,N3}$	✓	✓	✓	$E_{11, Triax,N3}$	✓	✓	✓
$E_{11, UD,N0}$	✓	✓	✓	$E_{11, Triax,N4}$	✓	✓	✓
$E_{11, UD,N1}$	✓	✓	✓	$G_{12, Triax,N0}$	✓	✓	✓
$E_{11, UD,N2}$	✓	✓	✓	$G_{12, Triax,N1}$	✓	✓	✓
$E_{11, UD,N3}$	✓	✓	✓	$G_{12, Triax,N2}$	✓	✓	✓
$E_{11, UD,N4}$	✓	✓	✓	$G_{12, Triax,N3}$	✓	✓	✓
$G_{12, Biax45,N0}$	✓	✓	✓	$G_{12, Triax,N4}$	✓	✓	✓
$G_{12, Biax45,N1}$	✓	✓	✓	$\rho_{Flange,N0}$	✓	✓	✓
$G_{12, Biax45,N2}$	✓	✓	✓	$\rho_{Flange,N1}$	✓	✓	✓
$G_{12, Biax45,N3}$	✓	✓	✓	$E_{11, Flange,N0}$	✓	✓	✓
$\rho_{Biax90,N3}$	✓	✓	✓	$E_{11, Flange,N1}$	✓	✓	✓
$\rho_{Biax90,N4}$	✓	✓	✓	$E_{11, Flange,N2}$	✓	✓	✓
$E_{11, Biax90,N0}$	✓	✓	✓	$E_{11, Flange,N3}$	✓	✓	✓
$E_{11, Biax90,N1}$	✓	✓	✓	$G_{12, Flange,N1}$	✓	✓	✓
$E_{11, Biax90,N2}$	✓	✓	✓	$G_{12, Flange,N2}$	✓	✓	✓
$E_{11, Biax90,N3}$	✓	✓	✓	$G_{12, Flange,N3}$	✓	✓	✓
$E_{11, Biax90,N4}$	✓	✓	✓	$\rho_{Balsa,N1}$	✓	✓	✓
$G_{12, Biax90,N0}$	✓	✓	✓	$G_{Balsa,N1}$	✓	✓	✓
$G_{12, Biax90,N1}$	✓	✓	✓	$\rho_{Foam,N2}$	✓	✓	✓
$G_{12, Biax90,N2}$	✓	✓	✓	$\rho_{Foam,N3}$	✓	✓	✓
$G_{12, Biax90,N3}$	✓	✓	✓	$P_{SS, Mid, spar cap,N0}$	✓	✓	✓
$G_{12, Biax90,N4}$	✓	✓	✓	$P_{SS, Mid, spar cap,N1}$	✓	✓	✓
$\rho_{Triax,N1}$	✓	✓	✓	$E_{11, Triax,N1}$	✓	✓	✓
$\rho_{Triax,N2}$	✓	✓	✓	$P_{SS, Mid, spar cap,N2}$	✓	✓	✓
$\rho_{Triax,N3}$	✓	✓	✓	$P_{PS, Mid, spar cap,N0}$	✓	✓	✓
$\rho_{Triax,N4}$	✓	✓	✓	$P_{PS, Mid, spar cap,N1}$	✓	✓	✓
$E_{11, Triax,N0}$	✓	✓	✓	$P_{PS, Mid, spar cap,N2}$	✓	✓	✓

Table A2. Identified mode shapes of the first 10 modes (excluding rigid body motion) of the free-free and the clamped modal configuration.

Mode no.	Free-free	Clamped
1	1. Flap	1. Flap
2	1. Edge	1. Edge
3	2. Flap	2. Flap
4	1. Torsion	2. Edge
5	3. Flap	3. Flap
6	2. Edge	1. Torsion
7	4. Flap	4. Flap
8	2. Torsion	2. Torsion
9	5. Flap	3. Torsion
10	3. Edge	5. Flap

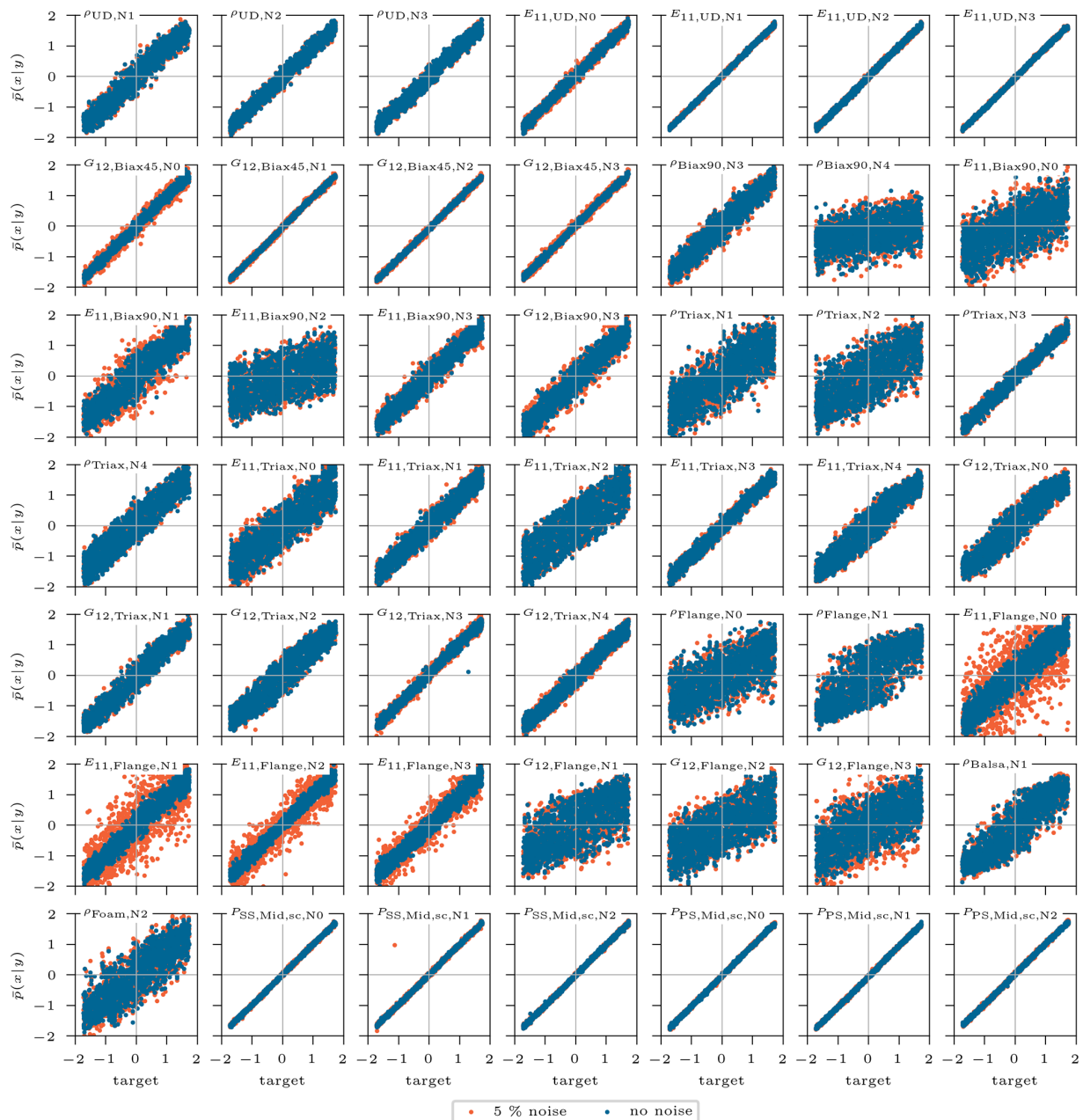


Figure A1. Standardized mean of posterior prediction \bar{p} of the updated inputs over the corresponding target standardized value for the 5000 test samples. The original samples predicted with clean conditions are in blue, compared to samples with noisy flawed conditions (5% random noise) in orange. The noisy conditions are intended to simulate measurement inaccuracies of the modal beam response.

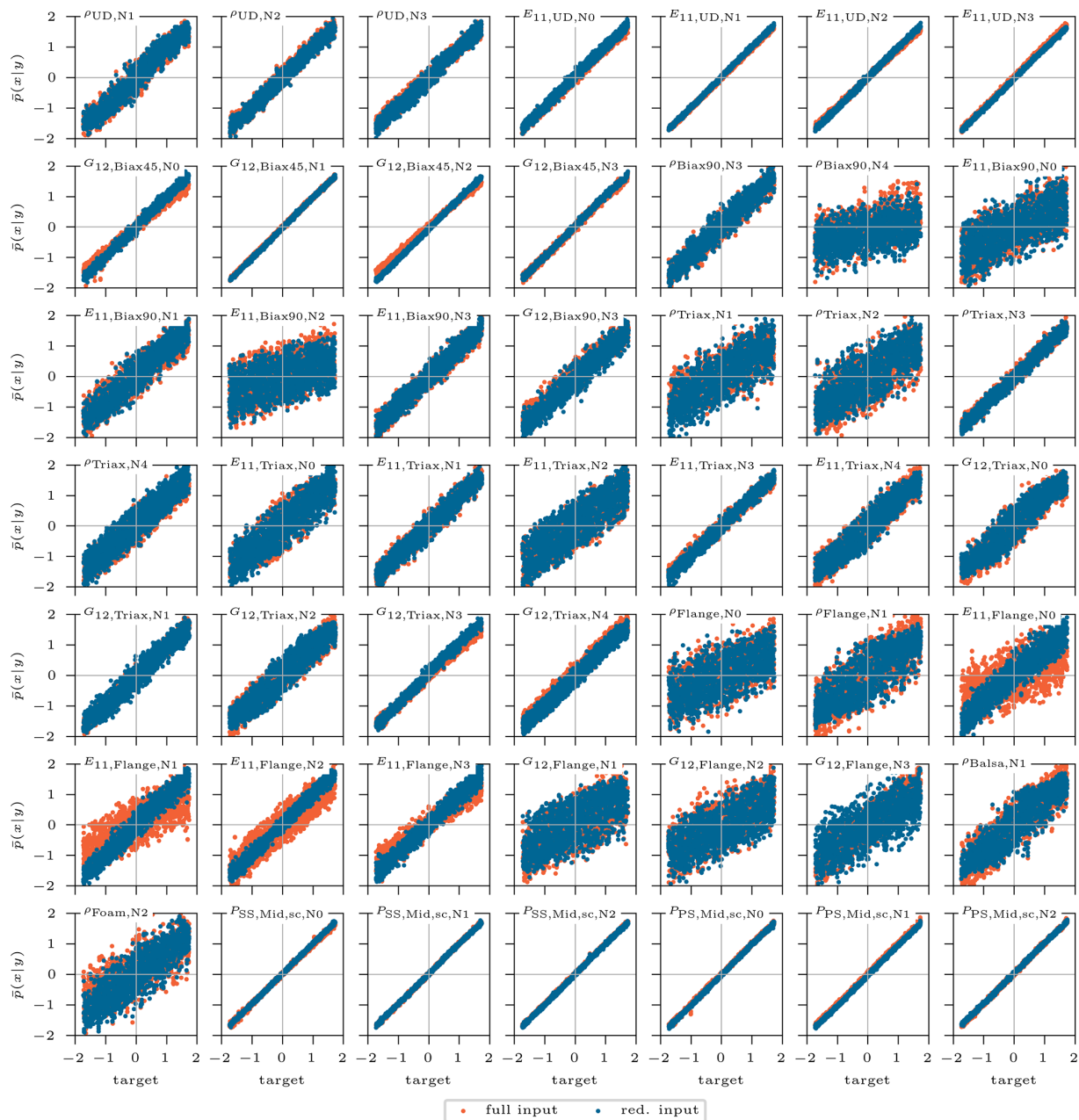


Figure A2. Standardized mean of posterior prediction \bar{x} of the inputs selected by the sensitivity analysis, over the corresponding target standardized values for the 5000 test samples. The original samples predicted with the reduced input set according to the sensitivity analysis selection are depicted in blue. They are compared with the inputs predicted by the cINN trained on the full input set (in orange).

Code and data availability. Code is available in a publicly accessible repository: <https://doi.org/10.5281/zenodo.6351906> (Noever-Castelos, 2022a). Data are available at <https://doi.org/10.25835/0042221> (Noever-Castelos, 2022b).

Author contributions. PNC prepared the concept and methodology, conducted the analysis, wrote the paper, and processed the review. DM supported PNC in transferring the BECAS beam information to ANSYS and in conducting the FE-analysis. CB is the supervisor and guided PNC in the conception of the paper.

Competing interests. The contact author has declared that neither they nor their co-authors have any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

The information in this paper is provided as is, and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at their own risk and liability.

Acknowledgements. This work was supported by the Federal Ministry for Economic Affairs and Climate Action of Germany (BMWK) in the project ReliaBlade (grant number 0324335A/B).

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 Association (DFG).

Financial support. The publication of this article was funded by the open-access fund of Leibniz Universität Hannover.

Review statement. This paper was edited by Carlo L. Bottasso and reviewed by Sarah Barber and Stefano Cacciola.

References

- Akima, H.: A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures, *J. ACM*, 17, 589–602, <https://doi.org/10.1145/321607.321609>, 1970.
- Allemang, R. J.: The modal assurance criterion – twenty years of use and abuse, *Sound Vib.*, 37, 14–23, 2003.
- ANSYS Inc.: Ansys® Academic Research Mechanical, Release 2021 R2, Help System, ANSYS Mechanical APDL Element Reference, <https://www.ansys.com/academic/terms-and-conditions> (last access: 14 March 2022), 2021a.
- Ardizzone, L., Mackowiak, R., Rother, C., and 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, 7828–7840, <https://researchr.org/publication/nips-2020> (last access: 14 March 2022), 2018.
- Ardizzone, L., Kruse, J., Wirkert, S., Rahner, D., Pellegrini, E. W., Klessen, R. S., Maier-Hein, L., Rother, C., and Köthe, U.: Analyzing Inverse Problems with Invertible Neural Networks, in: Seventh International Conference on Learning Representations, <https://doi.org/10.48550/arXiv.1808.04730>, 2019a.
- Ardizzone, L., Lüth, C., Kruse, J., Rother, C., and Köthe, U.: Guided Image Generation with Conditional Invertible Neural Networks, <https://doi.org/10.48550/arXiv.1907.02392>, 2019b.
- Augustyn, D., Smolka, U., Tygesen, U. T., Ulriksen, M. D., and Sørensen, J. D.: Data-driven model updating of an offshore wind jacket substructure, *Appl. Ocean Res.*, 104, 102366, <https://doi.org/10.1016/j.apor.2020.102366>, 2020.
- Blasques, J. P. and Stolpe, M.: Multi-material topology optimization of laminated composite beam cross sections, *Compos. Struct.*, 94, 3278–3289, <https://doi.org/10.1016/j.compstruct.2012.05.002>, 2012.
- Bruns, M., Hofmeister, B., Griebmann, T., and 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), edited by: Beer, M. and Zio, E., 1125–1132, Research Publishing Services, Singapore, https://doi.org/10.3850/978-981-11-2724-3_0713-cd, 2019.
- Chakraborty, S. and Saha, G.: Feature selection using singular value decomposition and QR factorization with column pivoting for text-independent speaker identification, *Speech Commun.*, 52, 693–709, <https://doi.org/10.1016/j.specom.2010.04.002>, 2010.
- Chollet, F.: Deep learning with Python, Safari Tech Books Online, Manning, Shelter Island, NY, ISBN 9781617294433, 2018.
- Chopard, B. and Tomassini, M. (Eds.): An Introduction to Metaheuristics for Optimization, Natural Computing Series, Springer International Publishing, Cham, <https://doi.org/10.1007/978-3-319-93073-2>, 2018.
- Dick, J. and Pillichshammer, F.: Digital nets and sequences: Discrepancy theory and quasi-Monte Carlo integration, Cambridge Univ. Press, Cambridge, ISBN 9780511761188, 2010.
- Duchi, J., Hazan, E., and Singer, Y.: Adaptive Subgradient Methods for Online Learning and Stochastic Optimization, *J. Mach. Learn. Res.*, 12, 2121–2159, 2011.
- Golub, G. H. and van Loan, C. F.: Matrix computations, Johns Hopkins studies in the mathematical sciences, 4th Edn., The Johns Hopkins University Press, Baltimore, ISBN 1421407949, 2013.
- Grieves, M. W.: Virtually Intelligent Product Systems: Digital and Physical Twins, in: Complex Systems Engineering: Theory and Practice, vol. 411, edited by: Flumerfelt, S., Schwartz, K. G., Mavris, D., and Briceno, S., American Institute of Aeronautics and Astronautics, Inc, Reston, VA, 175–200, <https://doi.org/10.2514/5.9781624105654.0175.0200>, 2019.
- Gross, D., Hauger, W., Schröder, J., and Wall, W. A.: Technische Mechanik 2: Elastostatik, Springer-Lehrbuch, 11. bearb. Aufl., Springer, Berlin, ISBN 9783642005640, 2012.
- Gundlach, J. and Govers, Y.: Experimental modal analysis of aeroelastic tailored rotor blades in different boundary conditions, *J. Phys.: Conf. Ser.*, 1356, 012023, <https://doi.org/10.1088/1742-6596/1356/1/012023>, 2019.

- He, K., Zhang, X., Ren, S., and Sun, J.: Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, <https://doi.org/10.48550/arXiv.1502.01852>, 2015.
- Herman, J. and Usher, W.: SALib: An open-source Python library for Sensitivity Analysis, *J. Open Source Softw.*, 2, 97, <https://doi.org/10.21105/joss.00097>, 2017.
- Hodges, D. H.: Nonlinear composite beam theory, in: vol. 213 of *Progress in astronautics and aeronautics*, American Institute of Aeronautics and Astronautics, Reston, VA, ISBN 1563476975, 2006.
- Hofmeister, B., Bruns, M., and Rolfes, R.: Finite element model updating using deterministic optimisation: A global pattern search approach, *Eng. Struct.*, 195, 373–381, <https://doi.org/10.1016/j.engstruct.2019.05.047>, 2019.
- Knebusch, J., Gundlach, J., and 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, 2159–2174, <https://doi.org/10.47964/1120.9175.19508>, 2020.
- Lin, J., Leung, L. K., Xu, Y.-L., Zhan, S., and Zhu, S.: Field measurement, model updating, and response prediction of a large-scale straight-bladed vertical axis wind turbine structure, *Measurement*, 130, 57–70, <https://doi.org/10.1016/j.measurement.2018.07.057>, 2018.
- Liu, X., Leimbach, K. R., and Hartmann, D.: System identification of a wind turbine using robust model updating strategy, in: *Proceedings of 19th International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering*, Weimar, 250–260, https://e-pub.uni-weimar.de/opus4/frontdoor/deliver/index/docId/2457/file/IKM2012_pdfa.pdf (last access: 14 March 2022), 2012.
- Luczak, M., Manzato, S., Peeters, B., Branner, K., Berring, P., and Kahsin, M.: Updating Finite Element Model of a Wind Turbine Blade Section Using Experimental Modal Analysis Results, *Shock Vibrat.*, 2014, 1–12, <https://doi.org/10.1155/2014/684786>, 2014.
- Marwala, T., Boukhaibet, I., and Adhikari, S.: *Probabilistic Finite Element Model Updating Using Bayesian Statistics*, John Wiley & Sons, Ltd, Chichester, UK, <https://doi.org/10.1002/9781119153023>, 2016.
- Noever-Castelos, P.: IWES-LUH/Beam-ModelUpdating-cINN, Zenodo [code], <https://doi.org/10.5281/zenodo.6351906>, 2022a.
- Noever-Castelos, P.: Training data and models for cINN model updating of finite element beam models of wind turbine blades, Research Data Repository of the Leibniz Universität Hannover [data set], <https://doi.org/10.25835/0042221>, 2022b.
- Noever-Castelos, P., Ardizzone, L., and Balzani, C.: Model updating of wind turbine blade cross sections with invertible neural networks, <https://doi.org/10.15488/11045>, 2021.
- Noever-Castelos, P., Haller, B., and Balzani, C.: Validation of a modeling methodology for wind turbine rotor blades based on a full-scale blade test, *Wind Energ. Sci.*, 7, 105–127, <https://doi.org/10.5194/wes-7-105-2022>, 2022.
- Olufsen, M. S. and Ottesen, J. T.: A practical approach to parameter estimation applied to model predicting heart rate regulation, *J. Math. Biol.*, 67, 39–68, <https://doi.org/10.1007/s00285-012-0535-8>, 2013.
- Omenzetter, P. and Turnbull, H.: Comparison of two optimization algorithms for fuzzy finite element model updating for damage detection in a wind turbine blade, in: *Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, Civil Infrastructure, and Transportation XII*, edited by: Shull, P. J., SPIE, p. 60, <https://doi.org/10.1117/12.2295314>, 2018.
- Pastor, M., Binda, M., and Harčarik, T.: Modal Assurance Criterion, *Proced. Eng.*, 48, 543–548, <https://doi.org/10.1016/j.proeng.2012.09.551>, 2012.
- Patelli, E., Govers, Y., Broggi, M., Gomes, H. M., Link, M., and Mottershead, J. E.: Sensitivity or Bayesian model updating: a comparison of techniques using the DLR AIRMOD test data, *Arch. Appl. Mech.*, 87, 905–925, <https://doi.org/10.1007/s00419-017-1233-1>, 2017.
- Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S.: *Global Sensitivity Analysis: The Primer*, John Wiley & Sons, New York, NY, ISBN 9780470059975, 2008.
- Sayer, F., Antoniou, A., Goutianos, S., Gebauer, I., Branner, K., and Balzani, C.: ReliaBlade Project: A Material’s Perspective towards the Digitalization of Wind Turbine Rotor Blades, *IOP Conf. Ser.: Mater. Sci. Eng.*, 942, 012006, <https://doi.org/10.1088/1757-899X/942/1/012006>, 2020.
- Schröder, K., Grove, S., Tsiapoki, S., Gebhardt, C. G., and Rolfes, R.: Structural Change Identification at a Wind Turbine Blade using Model Updating, *J. Phys.: Conf. Ser.*, 1104, 012030, <https://doi.org/10.1088/1742-6596/1104/1/012030>, 2018.
- Sehgal, S. and Kumar, H.: Structural Dynamic Model Updating Techniques: A State of the Art Review, *Arch. Comput. Meth. Eng.*, 23, 515–533, <https://doi.org/10.1007/s11831-015-9150-3>, 2016.
- SmartBlades2: Fabrication, Testing, and Further Development of Smart Rotor Blades, coordinated research project (project numbers 0324032A-H), supported by the Federal Ministry for Economic Affairs and Energy of Germany due to a decision of the German Bundestag, 2016–2020.
- Sobol’, I. M.: Sensitivity estimates for nonlinear mathematical models, *Math. Model. Comput. Exp.*, 1, 407–414, 1993.
- Sobol’, I. M.: Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates, *Math. Comput. Simul.*, 55, 271–280, [https://doi.org/10.1016/S0378-4754\(00\)00270-6](https://doi.org/10.1016/S0378-4754(00)00270-6), 2001.
- Velazquez, A. and Swartz, R. A.: Operational model updating of low-order horizontal axis wind turbine models for structural health monitoring applications, *J. Intel. Mater. Syst. Struct.*, 26, 1739–1752, <https://doi.org/10.1177/1045389X14563864>, 2015.
- Visual Learning Lab Heidelberg: FrEIA – Framework for Easily Invertible Architectures, GitHub, <https://github.com/VLL-HD/FrEIA> (last access: 14 March 2022), 2021.

5

Model updating benchmark of invertible neural networks versus optimization algorithms

Model updating with cINN has proven to be efficient and accurate in the previous Chapter 4. In this chapter a benchmarking study is presented that puts the cINN model updating approach into the context of commonly used optimization based approaches and compares their efficiency and accuracy.

5.1	Benchmarking basis	94
5.2	Surrogate model	95
5.3	Training and checking the cINN .	96
5.4	Optimization algorithms for the benchmark	96
5.5	Benchmarking with surrogate model	97
5.6	Benchmarking adapted to physical model.....	103
5.7	Discussion	104

The previous chapter approved with excellent results the efficient model updating of complex beam models of wind turbine blades via cINN. The state-of-the-art algorithms for model updating rely on optimization approaches (see section 1.2.2). In order to position the presented model updating method in the state of the art, this chapter will cover a benchmarking of cINN against optimization based model updating approaches. This benchmarking will not examine the full spectrum of optimization approaches, optimization tuning, or model updating techniques, but intends to reasonably position the cINN based model updating in this thematic field and to reveal the potential benefits of it.

5.1 Benchmarking basis

Following the previous chapter, this benchmark is based on the beam model and its modal response as described in section P3-2.2. In order to compare the computational efficiency and prediction accuracy of the selected approaches, the benchmark makes use of the following metrics for the comparison:

- Number of model evaluations n_{eval}
- Time consumption on a single core device with GPU-support, when applying a surrogate model $t_{\text{surrogate}}$ and interpolated to the real physical model $t_{\text{real model}}$
- Root-Mean-Square-Error (RMSE) of prediction to true value of a single sample e_x
- Correlation for multiple updated samples and corresponding true values
- MAC values and natural frequency errors e_f of re-simulation analysis

Section P3-4.5 showed that the cINN is capable of dealing with the complete input parameter space without losing accuracy. Therefore, it does not depend on a prior sub-space selection of the input parameters. However, the optimization based approaches typically demand a prior sub-space selection in order to reduce the model complexity in favor of computational time. Thus the benchmark of the model updating is performed with the sub-space selection as shown in Table P3-2 in Section P3-2.3. Additionally, the boundaries for the variation of each parameter is set according to Table P3-1.

As a single model evaluation of the complete tool chain *MoCA - Ansys* requires approx. 80 s on a single-core device, this benchmarking study makes use of a second approach to reduce the model complexity/computational costs: the tool chain *MoCA - Ansys* transforms the input parameter space into the degrees of freedom of the modal responses of the blade. In the following this tool chain will be referred to as *physical model*. Prior to the benchmarking, a surrogate model of the *physical model* is derived and is used for the benchmarking process in exchange for the *physical model*. By this, the accuracy of the model update against the *physical model* input parameters will suffer. However, it does not influence the benchmarking process, as the sampling for the cINN and the metrics computation will all refer to the surrogate model as baseline.

The benchmarking process will be performed on a GPU-cluster node with a NVIDIA Tesla P100 GPU. This allows for fast evaluation of ANNs. Moreover, no further parallelization of the code was applied.

5.2 Surrogate model

Surrogate models are used to replace physical models with an adequate accuracy, while being highly efficient with respect to computational costs for evaluating the model. Classic types of surrogate models are, i.a., polynomial response surface models, kriging models, or artificial neural networks (ANN). [73, 108] This particular application uses ANNs, as these in general are highly capable of modeling complex relationships of multidimensional input and output.

After a hyperparameter study, the architecture of the ANN is fixed to a rather simple system as depicted in Figure 5.1. It consists of an input (in: 49, out: 1,000 nodes) and an output (in: 1,000, out: 2,060 nodes) fully connected layer. In-between, a batch-normalization layer and a ReLU activation is applied. As the physical model's outputs are modal responses, this study applies a loss function for the training process, which is based on the MAC values and natural frequencies f similar to the fitness function proposed by [40]. The modal loss is derived as follows:

$$\mathbf{e}_{\text{MAC}} = \text{diag}(\text{MAC}(\mathbf{d}_p, \mathbf{d}_t)) - 1 \quad (5.1)$$

$$e_{\text{mse},f}^2 = \frac{1}{n_{\text{modes}}} \sum_{i=1}^{n_{\text{modes}}} \left(1 - \frac{f_{p,i}}{f_{t,i}}\right)^2 \quad (5.2)$$

$$\mathcal{L}_{\text{Modal}} = \frac{1}{2} \sqrt{\frac{\mathbf{e}_{\text{MAC}} \cdot \mathbf{e}_{\text{MAC}}^T}{n_{\text{modes}}}} + \frac{1}{2} \sqrt{e_{\text{mse},f}^2} \quad (5.3)$$

In equation (5.1) the \mathbf{d}_p and \mathbf{d}_t describe the matrices containing the predicted and target displacement vectors, respectively. For these, the diagonal vector of the computed MAC matrix is extracted. The difference of the MAC entries to the ideal value of 1 expresses the MAC-error \mathbf{e}_{MAC} . The mean squared error (mse) of the natural frequencies $e_{\text{mse},f}^2$, is computed by the sum of the squared relative error of the predicted frequency as depicted in equation (5.2). The modal loss $\mathcal{L}_{\text{Modal}}$ (equation (5.3)) represents the mean of the Root-Mean-Square-Error of both aforementioned errors.

The surrogate model is trained over 75 epochs with an AdaGrad optimizer, a batch size of 32 and a sample size of 20,000 samples, which are generated with the *physical model*. Figure 5.2 shows the loss curves of the training and validation.

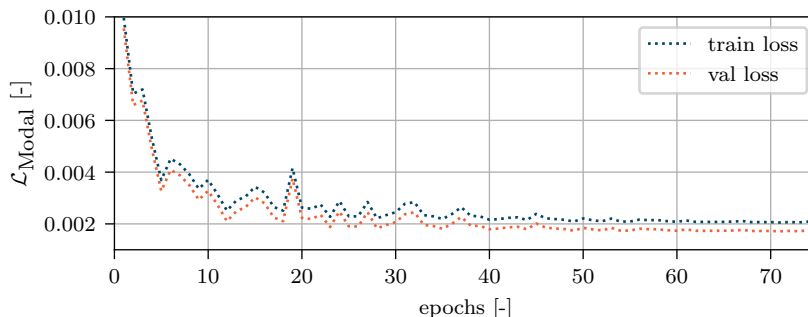


Figure 5.2: Training and validation loss of the surrogate model trained over 75 epochs. The applied loss $\mathcal{L}_{\text{Modal}}$ describes the modal coincidence of the model prediction and the target values.

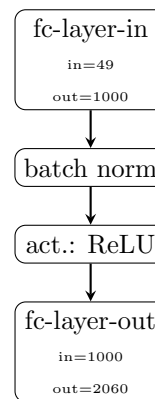


Figure 5.1: Structure of the surrogate model to replace the *physical model*. It consists of a fully connected (fc-layer) input and output layer and in-between a batch normalization and ReLU-activation layer.

The training and validation quickly converge and show little discrepancies (i.e., $\mathcal{L}_{\text{Modal}} > 0$) between the surrogate model and the *physical model*. However, as stated before, the focus of this study is the comparison of the model updating approaches, rather than the accuracy to the *physical model*. Therefore, it is important to have a model, that behaves somehow comparable and with a similar range of parameters, but at the same time makes it affordable to computed expensive iterative algorithms. This surrogate model cuts down the computational time for one evaluation to 0.00025 s/eval on a NVIDIA Tesla P100 GPU. Compared to approx. 80 s/eval for the *physical model* on a single Intel core i5 processor, this is a reduction by a factor of 320,000.

5.3 Training and checking the cINN

The cINN architecture for this Benchmark is fully taken from the cINN presented in Section P3-3, due to the similarity of the surrogate model. However, it is newly trained with samples generated by the surrogate model. The cINN will be trained with an AdaGrad optimizer to minimize the negative log-likelihood loss \mathcal{L}_{NLL} over 75 epochs. The optimizer's learning rate is set to 0.3, which is reduced with a scheduler by 25% after each 10 epochs. In order to find the minimum number of necessary training samples, the training was processed with a different training size, from 500 to 10,000 random samples picked from the surrogate model. The validation was always computed with the same 3,000 samples. Figure 5.3 depicts the loss curves for training and validation with increasing training sample numbers $n_{\text{train samp}}$. It clearly shows that the cINN already sufficiently generalizes with 2,000 samples, as the validation loss gets equal to the training loss. With increasing number of samples the \mathcal{L}_{NLL} is driven down until it reaches more or less a convergence with a stable loss at about 5,000 samples and $\mathcal{L}_{\text{NLL}} = -93$. Hence, the benchmarking reference will be set to the cINN model with 5,000 training samples.

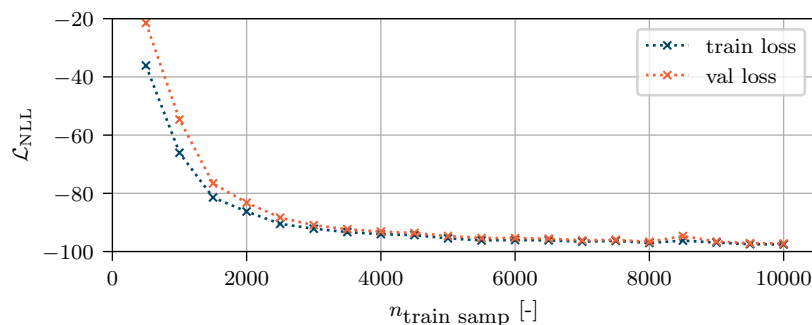


Figure 5.3: Convergence of the training and validation loss for increasing number of training samples of the cINN. The training was performed over 75 epochs minimizing the negative log-likelihood loss \mathcal{L}_{NLL} of the cINN.

5.4 Optimization algorithms for the benchmark

During this benchmarking study, four optimization algorithms will be compared with the cINN. These are heuristic methods, that can be applied to complex, non-linear black-box models and are gradient-free optimization algorithms. For each of the heuristic categories presented in Section 1.2.2 — swarm intelligence, evolutionary, physical/natural processes, direct search methods — one of the most popular algorithms is picked: particle-swarm-optimization (PSO), genetic-algorithm (GA), simulated annealing optimization (SimAn), and Nelder-Mead algorithm (NM). Without exhausting and analyzing in full depth the hyper-parameter variation of each algorithm, the following configurations were chosen for each optimization approach:

- PSO: swarm-size: 1,000, max-iteration: 2,000, search-radius-scale: 1
- GA: no-generations: 1,000, chromosomes-per-pop: 100, mutation: 40%, parents-mating: 10%
- SimAn: max-iteration: 1,000, max-fun-eval: 200,000
- NM: max-fun-eval: 200,000

All algorithms are defined to stop if they reach a convergence limit of $\Delta\mathcal{L}_{\text{Modal,lim}} = 1\text{e-}7$ to the previous iteration, or generation in case of the GA.

5.5 Benchmarking with surrogate model

After defining the benchmarking basics and selecting the benchmarking candidates, the model updating is performed in the first instance for one randomly chosen sample of input and output set computed with the surrogate model. In contrast to the cINN, the aforementioned optimization approaches yield a discrete input parameter prediction and do not give a probabilistic measure of the obtained results. Therefore, it is of utmost importance to double check the given results with repeated runs of the optimization. This creates a predicted distribution of each updated input parameter for each heuristic optimization approach. Such a distribution shows the user the degree of uncertainty in the given results. In this study the evaluation will be based on 100 runs of each optimization approach in order to establish a statistically relevant outcome. However, this number of repetitions is a subjective choice, therefore most of the specific benchmarking metrics will be averaged to a single run. The cINN already includes a probabilist evaluation of the inverse model. Therefore, the latent space sampling will be based on 100 random samples, to have a comparable histogram.

In the second instance, the author conducts the updating for 200 different, randomly chosen samples. This should reveal the generalization of the approaches to update the surrogate model in full extend, and not one potentially beneficial sample. However, for each of this randomly chosen sample only a single run is performed, as well as the cINN's latent space is only sampled once.

The first overall comparison is the convergence curve of each approach. Figure 5.4 depicts the best run of each algorithm, i.e., lowest final $\mathcal{L}_{\text{Modal}}$ and shows it's convergence curve. The NM algorithm (orange line) demonstrates a fast convergence and the lowest final loss of $\mathcal{L}_{\text{Modal}} = 6.9\text{e-}5$ after $n_{\text{eval}} = 22,400$ number of model evaluations. Also the PSO reaches relatively fast ($n_{\text{eval}} = 30,000$) a low loss level of $\mathcal{L}_{\text{Modal}} \leq 15.0\text{e-}5$, but takes another 32,000 evaluations to slowly reach the convergence limit and a final loss of $\mathcal{L}_{\text{Modal}} = 13.1\text{e-}5$. The other two approaches, GA and SimAn, only achieve a loss of $\mathcal{L}_{\text{Modal}} = 71.0\text{e-}5$ after $n_{\text{eval}} = 28,000$ and $\mathcal{L}_{\text{Modal}} = 62.3\text{e-}5$ after $n_{\text{eval}} = 96,800$, respectively. Hence, the GA has the worst results and SimAn needs the most evaluations until convergence. One may argue, that the convergence limit $\Delta\mathcal{L}_{\text{Modal,lim}}$ is not set properly, as for example the PSO runs half of the n_{eval} only to reach a slight improvement in the loss, while the other algorithms need the time to get out of their local optima (plateaus in the convergence curve), such as the GA and especially the SimAn algorithm. Even the best approach, the NM algorithm converges to a local minimum at about $n_{\text{eval}} = 17,000$, before it explores a better solution. A lower $\Delta\mathcal{L}_{\text{Modal,lim}}$ would prevent the algorithms to have time to search for a better optimum. And again, it has to be noted, this is only an excerpt of the best computed optimization run for one randomly chosen sample.

The cINN curve does not represent the convergence against $\mathcal{L}_{\text{Modal}}$, as it is trained against a different loss, the \mathcal{L}_{NLL} . However, for the trained models with different training sample sizes as referred to in Figure 5.3 in Section 5.3, the $\mathcal{L}_{\text{Modal}}$ is computed for the same randomly chosen sample, as in case of the optimization algorithms. The cINN demonstrates an even faster convergence than

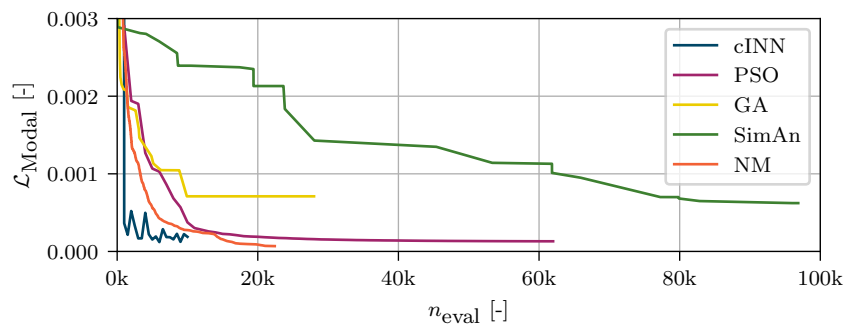


Figure 5.4: Convergence of all optimization approaches and the cINN against the modal loss $\mathcal{L}_{\text{Modal}}$. The number of evaluations n_{eval} is increased until the $\Delta\mathcal{L}_{\text{Modal},\text{lim}}$ is reached. The depicted convergence curves belong to the run with the best final fitness value.

the NM algorithm at a slightly higher loss of $\mathcal{L}_{\text{Modal}} = 15.0 \text{e-}5$ with $n_{\text{eval}} = n_{\text{trainsamp}} = 5,000$ training samples. With higher training sizes the $\mathcal{L}_{\text{Modal}}$ oscillates around this loss value.

Model updating of a single random sample

As described above, the first benchmarking scenario considers only one random sample, for which the updating is performed with the discussed approaches, but with 100 repetitions, each until reaching the maximum iteration or the convergence tolerance. Figure 5.5 shows the resulting average number of model evaluations \bar{n}_{eval} from the 100 conducted optimization runs. The bars clearly show that the SimAn algorithm, closely followed by the NM algorithm, need on average the least number of evaluation with approximately $\bar{n}_{\text{eval}} = 12,500$ and $\bar{n}_{\text{eval}} = 15,500$ samples, respectively. The PSO took the most number of samples on average with $\bar{n}_{\text{eval}} = 62,000$ and the GA needed $\bar{n}_{\text{eval}} = 39,000$. The cINN, however, only required a training sample size of 5,000 samples, which more than halves the amount of samples necessary for the SimAn.

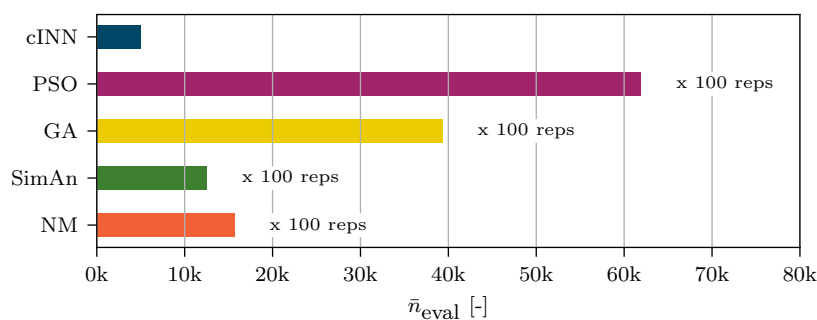


Figure 5.5: Average number of model evaluations \bar{n}_{eval} necessary for each model updating approach to convergence. Additionally, the number of repetitions necessary for a reasonable probabilistic distribution is denoted. The cINN does not need any repetitions.

The next considered metric for the benchmarking is the average time consumption of a single optimization run, which is depicted in Figure 5.6. For the optimization approaches this is close to an even split between surrogate model evaluation time and overhead time. This means it linearly scales with the \bar{n}_{eval} from above. The overhead of the optimization approaches originate in picking the current best solution of an iteration and preparing new parameter sets for evaluation during the next iteration step. The even time split is not the case for the cINN, the evaluation, i.e., sampling time, is scaled linearly to the number of samples, though, the overhead is much higher, as the cINN consists of the training process over several epochs.

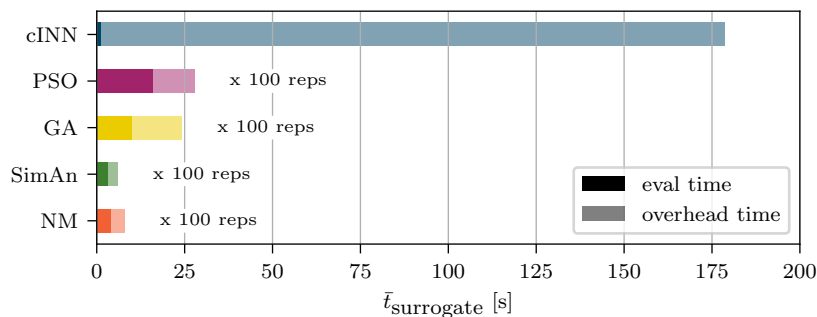


Figure 5.6: Average time $\bar{t}_{\text{surrogate}}$ necessary for a run of each model updating approach until convergence using the surrogate model. The bars are splitted into sampling time (depicted dark) and overhead time (depicted light) for each approach. Additionally, the number of repetitions necessary for a reasonable probabilistic distribution is denoted. The cINN does not need any repetitions.

However, the cINN is then finally finished and can be evaluated in the inverse path at nearly no costs, whereas the optimization approaches need to be fully run several times (here 100 repetitions) to establish a probabilistic parameter prediction as presented in Figure 5.7. In this figure the true value is depicted as a dashed line. The x-values are this time normalized to the parameter boundaries defined for the updating process and centered on the true values. Here, we see that only for a very few parameters, such as $E_{11,UD,N2}$, $E_{11,UD,N3}$ or $E_{11,Triax,N0}$, the optimization algorithms reach a roughly significant peak near the true value, but for most others the distributions seem random. In contrast, the cINN predicts the parameters very close to the true value and with high confidence (narrow distribution).

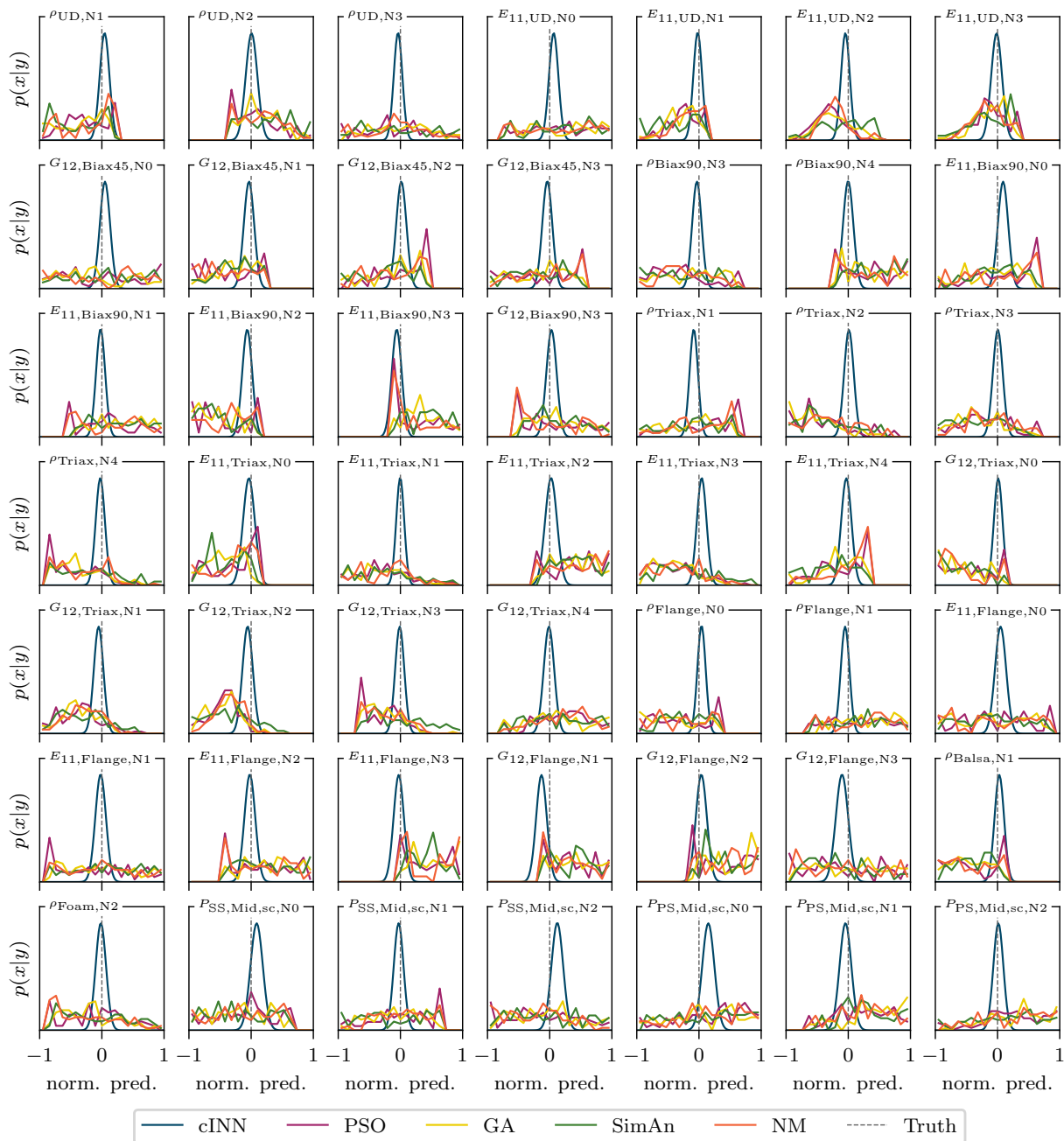
As the optimization algorithms are all evaluated against the $\mathcal{L}_{\text{Modal}}$, it can be expected that all model results with the obtained input predictions show a good correlation with the target mode shapes and frequencies. Table 5.1 shows exactly this, the mean and standard deviation of the $\mathcal{L}_{\text{Modal}}$ is relatively small. However, the Root-Mean-Square-Error of the input predictions given in equation 5.4 shows a significantly high mean value and standard deviation for all the optimization approaches over the 100 runs.

$$e_x = \sqrt{\frac{1}{n} \sum_{i=1}^{n_x} \left(\frac{x_{t,i} - x_{p,i}}{x_{b,i}} \right)^2} \quad (5.4)$$

The error describes the RMSE of the difference between predicted input $x_{p,i}$ and target input $x_{t,i}$ normalized to the boundaries $x_{b,i}$. The high accuracy in the fitness function and the relatively poor prediction accuracy with respect to the true values of the corresponding input, probably point to multiple identified local optima, with a different constellation of input parameters, which however lead to a similar mode shape and thus low $\mathcal{L}_{\text{Modal}}$. The cINN in contrast has the big advantage that it is trained against \mathcal{L}_{NLL} and learns conditional probabilities of the input given the raw output, i.e., mode shape displacements. This background knowledge of dependencies or conditions is reflected in the good input predictions of the cINN.

Table 5.1: Mean and standard deviation of model loss $\mathcal{L}_{\text{Modal}}$ and input prediction error e_x for all approaches.

	$\bar{\mathcal{L}}_{\text{Modal}}$	$\sigma_{\mathcal{L}_{\text{Modal}}}$	\bar{e}_x	σ_{e_x}
cINN	0.00087	0.00040	0.0968	0.0101
PSO	0.00031	0.00010	0.7167	0.0809
GA	0.00101	0.00013	0.6650	0.0676
SimAn	0.00272	0.00088	0.7265	0.0730
NM	0.00025	0.00012	0.6894	0.0779

**Figure 5.7:** Input predictions $p(x|y)$ based on a model updating with the fitness $\mathcal{L}_{\text{Modal}}$. The update was repeated 100 times to see if a local or global maximum is reached. The output is normalized to the evaluation boundaries of each parameter.

Model updating of multiple random samples

Next the model updating will be applied to 200 random samples, but only performing the optimization algorithm once per sample. The cINN is also only performed with one random sample in the latent space to make it comparable, though a higher latent space sampling would not impact on the computational costs in contrast to the optimization approaches.

In the first instance, Figure 5.8 shows the correlation of predicted values x_p over true values x_t for each parameter and model updating approach as a point in the scatter plot. All values of a parameter are again normalized to their corresponding defined boundaries for the optimization process. An ideal solution would map all points onto a straight thin diagonal line with the slope $m = 1$.

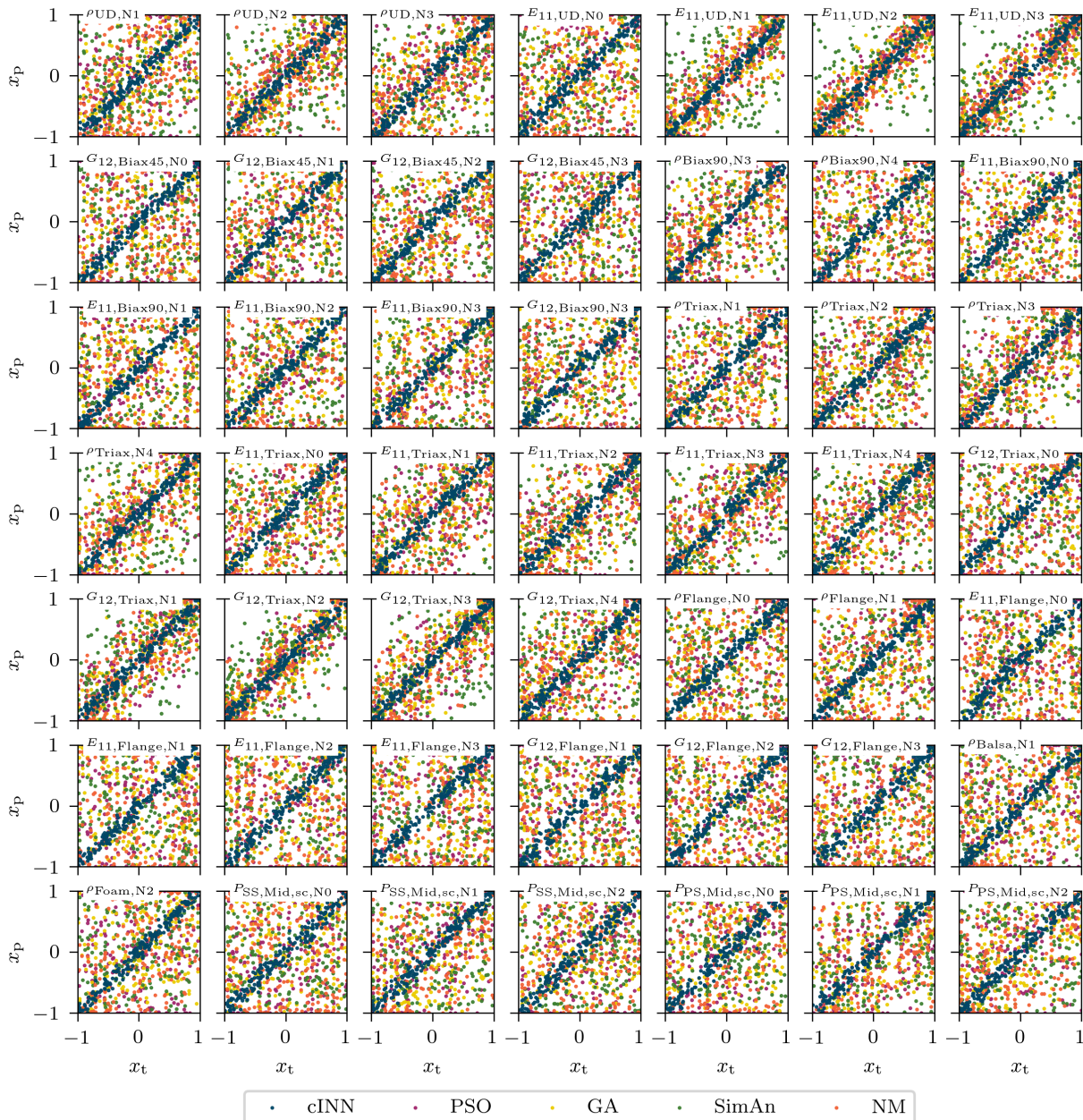


Figure 5.8: Correlation of input prediction x_p and corresponding target input x_t for all approaches and 200 random samples. All values are normalized to their respective optimization boundaries. Ideally all results should coincide with a straight line with a slope of $m = 1$.

The distributions reflect the previous result given in Table 5.1. The optimization algorithm results spread a lot in their prediction compared to the target values, except for a few parameters, such as $E_{11,UD,N1-3}$ or $G_{12,Triax,N1-3}$, for which the scatter is noticeably more bundled to the ideal solution. Whereas the cINN results correlate exceptionally well, thus match great with the ideal solution. The author refrains from computing coefficients of determination as the figures are obviously clear to interpret.

Similar again to the results in Table 5.1, the MAC values show a nearly absolute agreement of the target mode shapes to the resimulated mode shapes based on the prediction. For the first 10 mode shapes of the free and the clamped test configuration Figure 5.9 presents the minimum, maximum and mean values of all 200 test samples. And for nearly all approaches, even the minimum MAC values are all above $MAC > 0.995$. Except the SimAn algorithm shows lower values but still all $MAC \geq 0.985$, which would also be considered as a very good result. In general the cINN and the PSO provides the best results, being almost always $MAC = 1$.

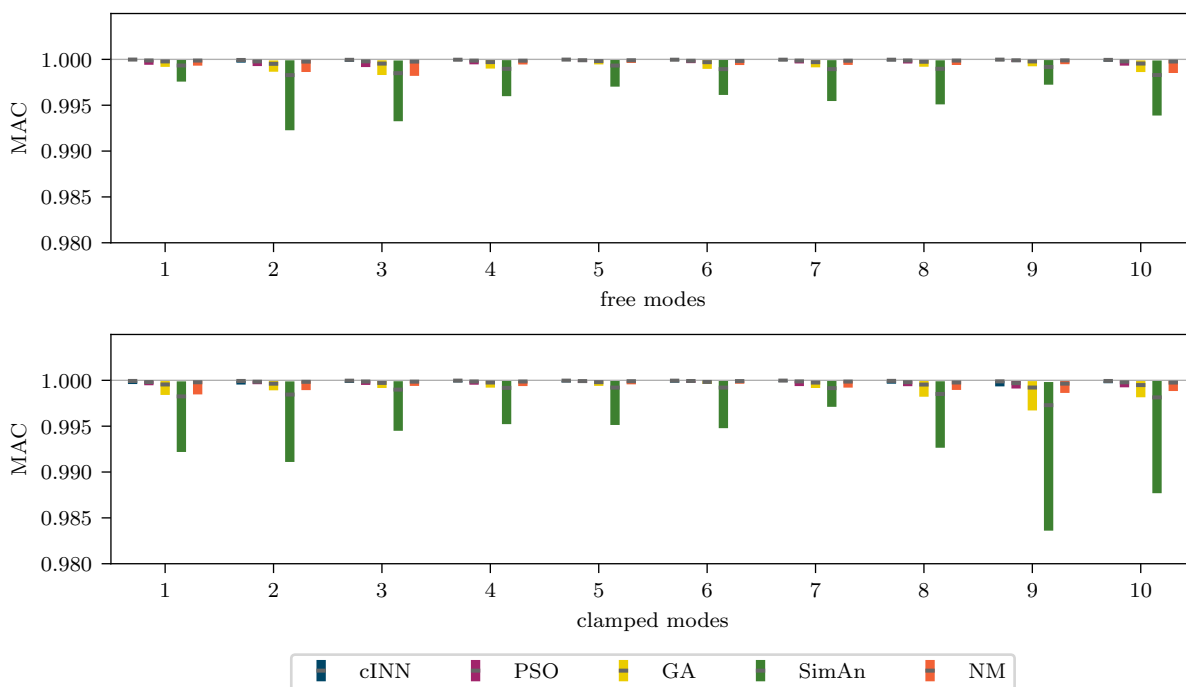


Figure 5.9: Maximum, minimum and mean MAC value of each model shape of the re-simulated models based on the predictions for the 200 random samples.

Similar results are also presented in Figure 5.10, which show the mean and $1-\sigma$ -range for the natural frequency error $e_{f,i} = 1 - \frac{f_{p,i}}{f_{t,i}}$ for each mode i . Here the errors for all modes are relatively low with the $1-\sigma$ -range limited to a maximum of $e_f = \pm 0.25\%$, except for the SimAn algorithm, which reaches up to $e_f = \pm 0.7\%$. But in contrast to the results in Figure 5.9, the PSO and the NM algorithm outperforms the cINN, showing nearly no error $|e_f| < 0.01\%$. This is a result from the chosen fitness function, as the frequency errors contribute to half of the modal loss $\mathcal{L}_{\text{Modal}}$ (see equation 5.3). On the other hand, the frequencies constitute only a small amount of the observed response in the cINN training and thus have less importance compared to the contribution in the modal loss of the optimization algorithms. In consequence, these algorithms tend to put a higher effort in matching the correct frequency. This effect could be regulated by choosing a different weighting of MAC-error and frequency error in equation 5.3.

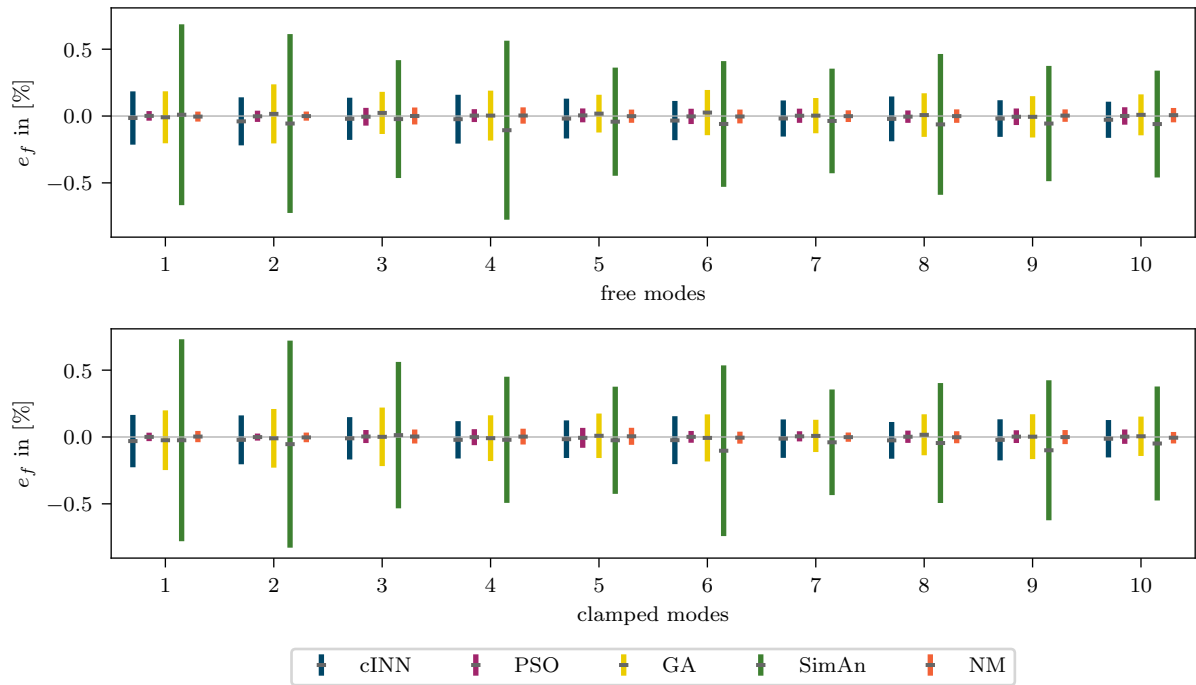


Figure 5.10: Mean and 1- σ -range for the natural frequencies of each model shape of the re-simulated models based on the predictions for the 200 random samples.

5.6 Benchmarking adapted to physical model

After evaluating all the metrics on the basis of the surrogate model, this section gives a brief outlook how much the updating computational costs would evolve in case of applying the *physical model*. Under the assumption the surrogate model fully represents the *physical model* and it's response, the average sample number for a model run would be the same. Only the time for evaluating the model would change from 0.00025 s/eval to approx. 80 s/eval. By scaling the computational time consumption presented in Figure 5.6 for the evaluation time and leaving the overhead equal gives the results depicted in Figure 5.11 (note: the unit is now hours). The overhead time consumption is negligible compared to the necessary time for the model evaluation. Here, all optimization algorithms surpass the cINN time by far. Even the fastest algorithm, the SimAn with $\bar{t}_{physical\ model} = 277$ h, takes more than twice as long as the cINN ($\bar{t}_{physical\ model} = 111$ h). Disregarding any additional run to test against local optima.

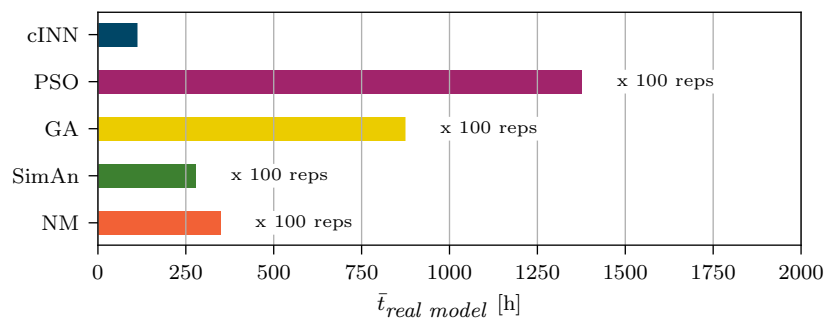


Figure 5.11: Average time $\bar{t}_{physical\ model}$ necessary for a run of each model updating approach until convergence, extrapolated for the *physical model*. Additionally, the number of repetitions necessary for a reasonable probabilistic distribution is denoted. The cINN does not need any repetitions.

5.7 Discussion

This benchmarking clearly proves that for the proposed settings and scenarios the cINN outperforms the chosen optimization approaches for model updating, with the highest accuracy and especially in terms of reliability, while still being relatively fast. All optimization based approaches fail to find the correct posterior input parameters. The posterior distributions of the parameters for one random sample predicted by the optimization algorithms do not have a significant peak, in contrast to the cINN, which shows a normally distributed and narrow posterior prediction around the true value (see Figure 5.7). The same holds for updating multiple random samples (see Figure 5.8), where the true value is only correctly predicted by the cINN. However, all approaches show good results concerning the MAC-values and the natural frequency error. This leads to two different conclusions:

1. The $\mathcal{L}_{\text{Modal}}$ may not be the best fitness function, as it describes a very global blade behavior, which averages the modal information and provokes a loss of information for a correct system identification.
2. The model itself has significant intrinsic ambiguities of the modal response, which prevents an optimization algorithm to correctly predict the posterior.

The cINN has the advantage that it is trained to learn the conditional probability of the input given an observation or response according to the Bayes' theorem. Additionally, these conditions are given as the raw displacements of the modal shapes and no averaging or similar is performed on it.

But, not only the accuracy of the cINN is excellent, even the necessary model evaluations are the least, less than half of the best algorithm on a single run. Although, the overhead time for the cINN for training the model requires more computational time than a single run of any of the other approaches, but these of course should be repeated in order to double check if a global or local optimum is found. As soon as the computational time consumption is extrapolated to the evaluation with the *physical model*, the overhead time becomes insignificant and the cINN outperforms the other algorithms.

The absolute computational time for the real model of course can be easily reduced by parallelization of the model evaluation, as this is the main cost driving part. Most of these algorithms typically have implemented partly or full parallelization of the code. For the cINN a parallelization is straight forward, as the training samples are generate prior to the training process itself. Additionally, the author would like to note that the cINN has the same accuracy applied to the full parameter set without a sub space selection in advance, as stated in Section P3-4.5. This may not necessarily be true for the optimization approaches.

6

Model updating of a 3D finite element wind turbine blade model with invertible neural networks

After successfully applying cINN based model updating on a finite element beam model, this study transfers the approach to a more complex level of a hybrid 3D finite element model with shell and solid elements. The cINN model updating is based on a real experimental setup and will be subsequently applied to the measured experimental data, once it is validated on a generic model. Finally, the model updating errors that arised with experimental data and their possible root causes are discussed.

6.1	Setting up the model updating problem	106
6.2	Invertible neural network configuration	108
6.3	Generic model updating of a 3D-FEM blade model	108
6.4	Model updating based on modal experimental data	113

In this chapter the cINN model updating is applied to the full 3D finite element model as described in section P1-3. First, Section 6.1 describes the model updating problem based on lessons learned within chapter 2 and the experimental setup performed for the real blade. Subsequently, Section 6.2 follows with the cINN architecture for this study. Then the model updating is conducted and checked for the generic model in Section 6.3, after which the experimental data is applied to the inverse model to infer the posterior prediction of the input parameters in Section 6.4. The results are finally analyzed for their validity and discussed.

6.1 Setting up the model updating problem

Again, the SmartBlades2 20 m demonstration blade [179], in the following abbreviated by DemoBlade, represents the physical model for the model updating. In particular the 3D finite element model is used as described and validated in chapter 2. It is a hybrid linear shell and solid model, where the shell elements represent the composite parts of the blade and the solid elements all volumetric adhesive joints. The sensor positions were defined according to the real world experiments, to measure the blade's displacement along its span [88]. Figure 6.1 illustrates the blade finite element mesh and the given sensor positions and measurement directions. In total 16 cross sections and additionally the tip were equipped with sensors. In each cross section the flapwise motion displacement (violet arrows) is measured at four positions of the blade's suction side from leading edge to trailing edge and the edgewise motion (green arrows) is captured at the leading edge sensors. The sensor distribution is used in the experiment to measure the free-free vibrating mode shapes of the blade. These mode shapes and the corresponding natural frequencies serve as response for the model updating procedure and therefore will be replicated by the simulation. After setting up the tool chain to generate the Model with MoCA and evaluate it with ANSYS, it is used to feed the physics-informed cINN and generate representative training samples. This cINN will then be of use to update the input parameters for the measured response for all three real blades.

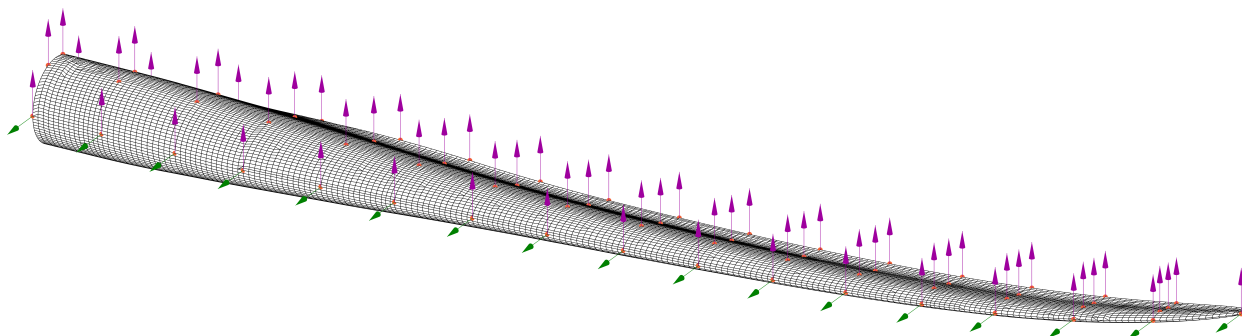


Figure 6.1: Sensor distribution along the DemoBlade for the free-free modal experiment depicted with the 3D finite element model of the blade. Violet arrows represent the flapwise motion sensors and the green arrows the edgewise motion sensors.

Following the conclusion in Section P3-5 of the feasibility study with the Timoshenko beam model, the input parameter space was redesigned in order to avoid ambiguities/co-linearities. In this study the material parameters are changed throughout the complete laminate, hence no ambiguous results occur, e.g., from two interacting face laminates in terms of stiffness or density. In the light of manufacturing processes, this also makes more sense, as the laminate is infused all together. This definition first reduces the parameter space. By adding different layup sections crosswise in the blade as depicted in Figure 6.2, the parameter setup accounts for material variations in the cross sectional direction. This creates more flexibility for the cINN to distribute the densities and stiffnesses in

order to match the real blade. These sections basically represent the leading panel, the trailing panel and the spar cap for each shell side. In addition to this six different sections along the blade, the position and width of the spar cap on each blade side can be varied individually and a specific mass located at the lightning protection cable ($m = 5 \frac{\text{kg}}{\text{m}}$) can be adapted. Finally, Table 6.1 summarizes the parameter space and boundaries linked to the aforementioned parameters in Figure 6.2. This study analyzes the in-plane stiffnesses E_{11} , E_{22} , shear stiffness G_{12} and the density ρ for all materials in each section. Again, the spanwise variation of each parameter is realized with an equidistant 5-noded-spline, except for the spar cap location and width where the last tip node of each spline is set to 0 due to geometric issues arising from the small scales at the tip. During sampling all these parameters will be independently and randomly picked from a uniform distribution within the respective boundaries.

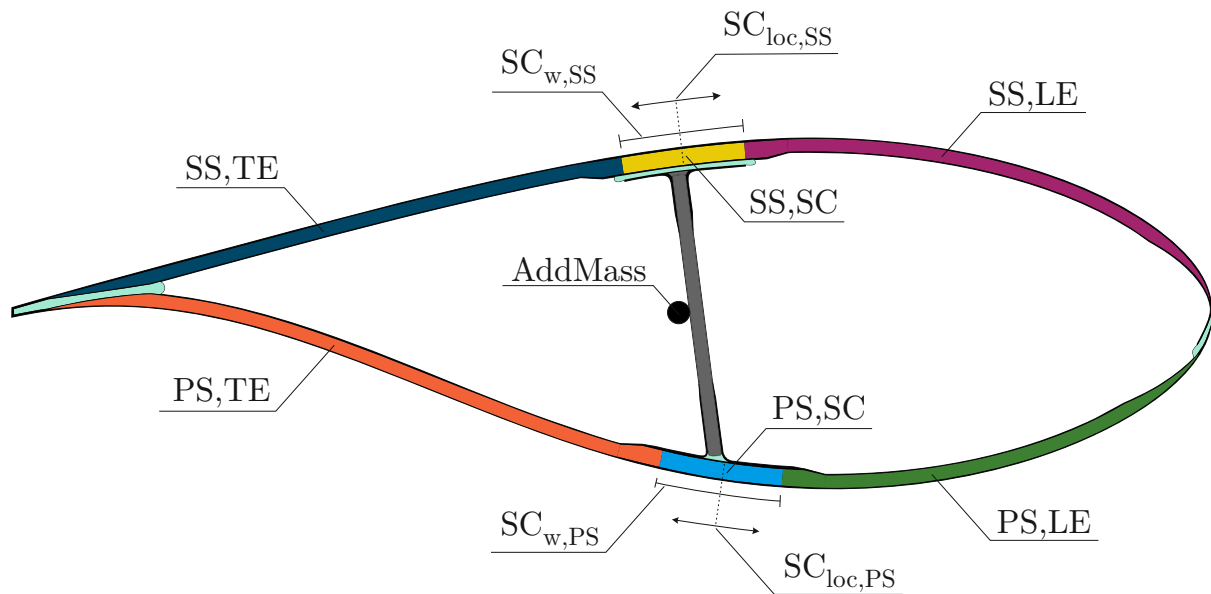


Figure 6.2: Parameter definition for the model updating input parameters illustrated in a schematic cross sectional view of the DemoBlade.

Table 6.1: Input feature list for the 3D finite element blade model updating. Each feature and property builds a distribution spline based on the given number of equidistant spline nodes within the given normalized radial range. Each node value may then vary in the listed variance range.

Parameter	Property	Spl. nodes	Radial range	Variance
SS, TE	$E_{11}, E_{22}, G_{12}, \rho$	5	[0, 1]	$\pm 25\%$
SS, SC	$E_{11}, E_{22}, G_{12}, \rho$	5	[0, 1]	$\pm 25\%$
SS, LE	$E_{11}, E_{22}, G_{12}, \rho$	5	[0, 1]	$\pm 25\%$
PS, LE	$E_{11}, E_{22}, G_{12}, \rho$	5	[0, 1]	$\pm 25\%$
PS, SC	$E_{11}, E_{22}, G_{12}, \rho$	5	[0, 1]	$\pm 25\%$
PS, TE	$E_{11}, E_{22}, G_{12}, \rho$	5	[0, 1]	$\pm 25\%$
SC, SS	loc, w	4	[0, 0.75]	± 1 cm
SC, PS	loc, w	4	[0, 0.75]	± 1 cm
AddMass		5	[0, 1]	$\pm 75\%$

SS: Suction Side; PS: Pressure Side; TE: Trailing Edge; SC: Spar cap; LE: Leading Edge;
AddMass: Additional Mass.

6.2 Invertible neural network configuration

Next, inspired by the presented architecture in Section P3-3, the cINN is defined. However, a hyperparameter analysis showed that a cINN built by only eight conditional coupling blocks (CC) is better suited for this problem. The CCs are divided into clusters of two and in consequence result in only four output stages from the *conditional network*. Therefore the width of each *subnetwork* in the CCs or the channels of each 1D-convolution in the *conditional network* was slightly increased. The corresponding hyperparameter set for this cINN is summarized in Table 6.2. The cINN is trained against 90,000 training samples over 200 epochs and an initial learning rate of 0.6, which is reduced by 50% each 25 epochs. The batch size is set to 32 samples. The training progress is tested against 10,000 unseen validation samples.

Table 6.2: Hyperparameter set of the complete network, including *conditional network*, *conditional invertible neural networks* (cINN), and *sub-network*. The cINN is divided into four clusters (Clst), for which the hyperparameters are listed separately. In Cluster 1, the conditions are directly fed into the *conditional coupling blocks* (CC), without a prior convolutional layer. The structure is inspired by Figure P3-4 and Table P3-3.

			Clst 1	Clst 2	Clst 3	Clst 4
Conditional network	Conv 1D	kernel k		3	3	3
		stride s		1	1	1
		padding p		1	1	1
		out chan. out		256	512	1024
	Activation			PReLU	PReLU	PReLU
	Average 1D pooling	kernel k		2	2	2
		stride s		2	2	2
padding p			0	0	0	
Flatten		✓	✓	✓	✓	
Fully connected	nodes	300	400	500	600	
cINN	Conditional coupling block (CC)		2	2	2	2
Sub-network	Fully connected	nodes	700	800	900	1000
	Batch normalization		✓	✓	✓	✓
	Activation		PReLU	PReLU	PReLU	PReLU

6.3 Generic model updating of a 3D-FEM blade model

In the first instance the cINN is analyzed based on new generic samples extracted from the physical model before processing the model updating with the experimental data. The analysis comprise: examining the general updating results, identifying collinearities of predicted parameters and performing a resimulation analysis.

6.3.1 General analysis of the updating results

For the general analysis, the cINN predicts the input parameter's posterior distribution for 1,000 randomly generated samples, which were computed with the physical model. For each sample, the latent space was sampled 100 times. All resulting posterior distributions $p(x|y)$ are predominantly normally distributed. Therefore, the mean value $\bar{p}(x|y)$ is used as the final prediction value. These predictions are then plotted against the target values x_t for the respective sample and depicted as scatter plots in Figure 6.3 for all spline nodes of four exemplary parameters. Additionally, a regression line (dashed, orange) shows the general trend of the predictions. The quality of the

prediction can be checked visually again; the closer the scattering gets to the optimum line (solid, yellow) the more accurate is the prediction.

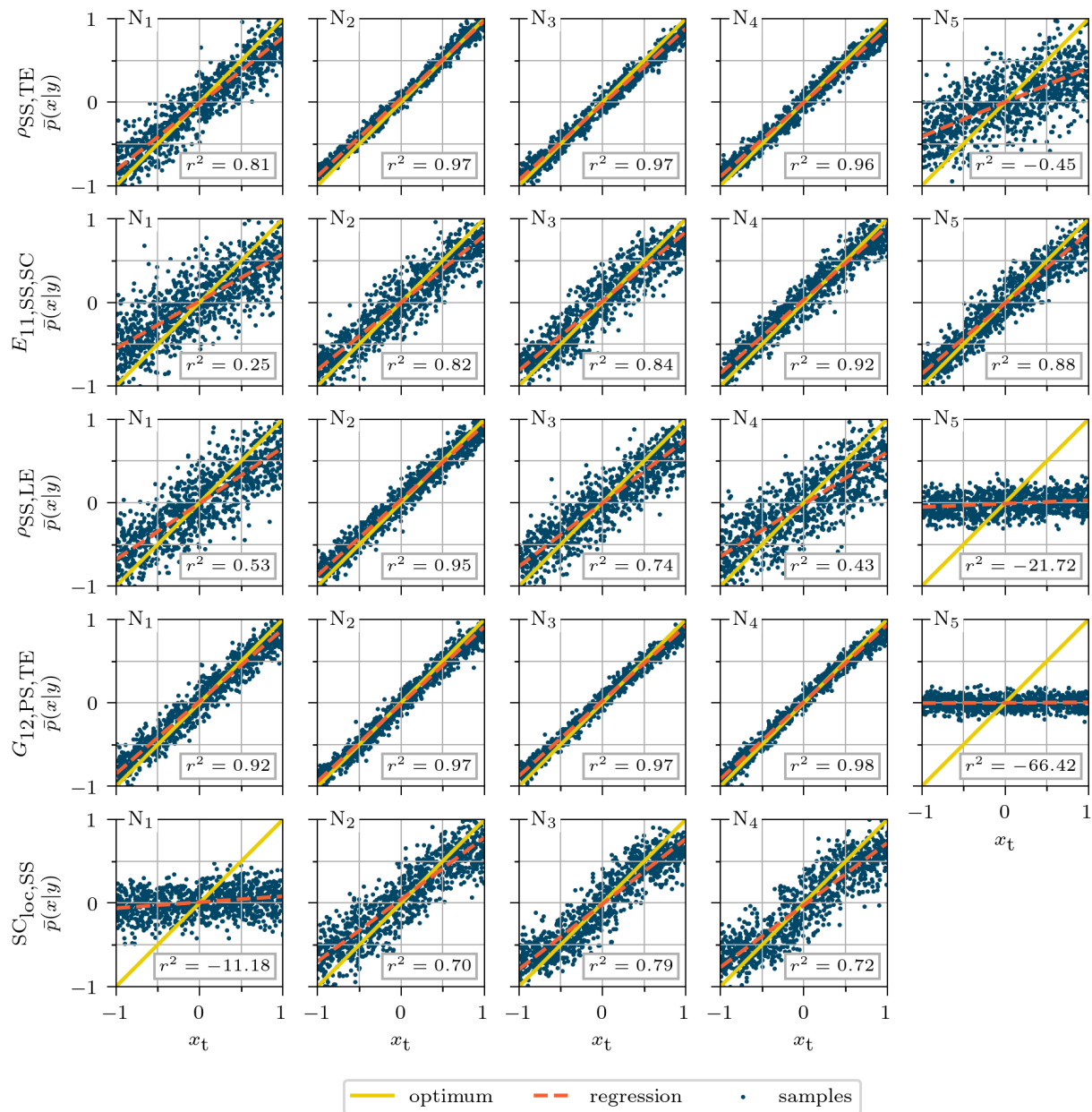


Figure 6.3: Correlation of posterior prediction versus target values for 1,000 random samples. The optimum solution is depicted as a yellow bisectonal straight line. Additionally, the coefficient of determination r^2 (see Equation 6.1) of the parameter prediction against the ideal result is shown.

In contrast to the previous Chapters 3 and 4, here, the coefficient of determination does not represent the accuracy of the samples with respect to the corresponding regression line, but with respect to the optimum line. By this, the coefficient of determination, denoted here as r^2 and given in Equation 6.1, is corrected to directly give the accuracy of the prediction.

$$r^2 = \frac{\sum_{j=1}^{n_{\text{samp}}} (\bar{p}_j(x|y) - x_{j,t})^2}{\sum_{j=1}^{n_{\text{samp}}} (\bar{p}_j(x|y) - \bar{x}_p)^2} \quad \text{with} \quad \bar{x}_p = \frac{1}{n_{\text{samp}}} \sum_{k=1}^{n_{\text{samp}}} \bar{p}_k(x|y) \quad (6.1)$$

Due to this formulation r^2 can become heavily negative. To have a slight orientation the interpretation of r^2 in this context can be defined as follows:

- $r^2 = 1$: Perfect accuracy of prediction to target values
- $r^2 > 0.5$: Still decent accuracy of prediction to target values
- $r^2 > 0$: Correct trend prediction with respect to target values
- $r^2 < 0$: Prediction closer to a baseline model, i.e., $\bar{p}_j(x|y) = 0$, than to the ideal solution

This interpretation is also slightly reflected in the regression lines in Figure 6.3, e.g., N_1 of $E_{11,SS,SC}$ (top left) has an $r^2 \approx 0$ the regression line is approximately the bisector of the optimum and a baseline model (the horizontal x-axis). Therefore, it is exactly on the border. Hence, $r^2 > 0$ means the regression line is inclined to the optimum (cf. N_{2-5} of $E_{11,SS,SC}$) and vice versa, $r^2 < 0$ inclines more to the baseline model (cf. N_1 of $SC_{loc,SS}$).

All this assumes that the y-interception of the regression line is always approximately 0, which is the case for all parameters. In general it can be stated, the trend of any parameter predicted by the cINN is never non-proportional, i.e., the slope of the regression line is always greater than 0. So the cINN understands globally the physics behind the data and in case it does not capture these at all for a particular parameter, the prediction is just scattered close to zero (c.f. N_5 of $G_{12,PS,TE}$) and not randomly guessed.

The complete set of results is summarized in Figure 6.4, condensed to the r^2 values for all predicted parameters over the 1,000 random samples. The parameters span over the columns, whereas the different nodes can then be found in the corresponding row. Figure 6.4 clearly shows the center nodes 2-4 are predicted very accurate for most of the parameters. The tip nodes (N_5) have the worst prediction accuracy, the root nodes (N_1) are a mix with good and bad predictions.

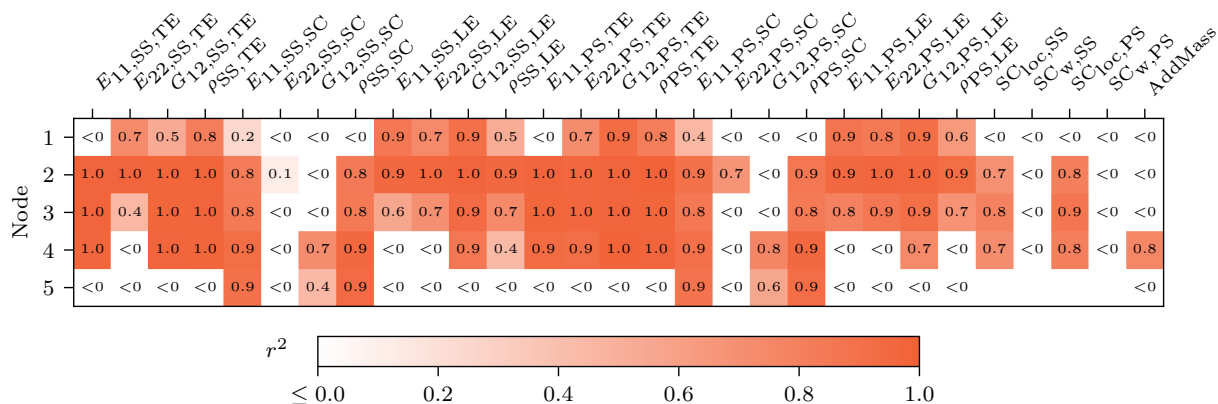


Figure 6.4: Coefficient of determination r^2 of each parameter prediction against the ideal results, i.e., a straight line with $y = x$. The coefficient is measured on the basis of 1,000 random sample predictions.

Roughly explained, the most dominant cause for this is that if the blade is divided into 4 spanwise sections in-between the nodes, then each node contributes always to each adjacent section to 50%.

The root and tip nodes only influence the parameter variation over the blade radius to one side of the node. Hence, the root and tip nodes only have impact on 12.5% and each mid node to 25% of the blade. During inference only these adjacent parts contain pure information for each node, which accordingly results in less information for the root and tip node compared to the mid nodes. This is of course significantly simplified for imagination purpose, as the parameter variation at a specific node may not have such discrete ranges of impact, but more overlapping with other nodes. However, a too dense spline node distribution leads to collinearities in the prediction.

The second reason for the good prediction of the mid nodes can be, that typically most of the deformation or the greatest variety of the blade's mode shapes happen in the mid section of the blade and not to its extremities. Thus the information has a wider spectrum of states and by that a higher information resolution or quality for the inference.

Further, a cross correlation analysis reveals any model ambiguities, i.e., model collinearities. Figure 6.5 shows the cross correlation matrix for all predicted input parameters over the 1,000 random samples. This matrix reveals some slight collinearities of the suction and pressure side parameters at the trailing edge. The density ρ , but also the stiffnesses E_{11} and G_{12} correlate negatively, which is due to its geometrical proximity of both shell sides at the trailing edge. This collinearity increases from the first to the fourth node as the profiles get thinner and the trailing edge panels approach each other. A second collinearity can be found between the spar cap width SC_w and the spar cap stiffness $E_{11,SC}$ or density ρ_{SC} for each shell side. And finally, some adjacent nodes of the same parameter, such as $G_{12,SS,SC} N_{4-5}$ or $E_{11,PS,SC} N_{4-5}$, show interactions, the same phenomena already found in the previous chapter 4 in Section P3-4.5.

6.3.2 Resimulation analysis

A resimulation analysis shows how accurate the predicted input parameters can reconstruct the modal response with the physical model, which was used to predict the input parameters. Without performing any prior feature selection, the full set of the predicted posterior distribution of the parameters is used to resimulate the response and compare it to the analysis. Following the same approach as described in Section P3-4.4, the MAC values and the natural frequencies of the five given free-free eigenmodes are used to compare the resimulated results to the original predictions.

The results for this resimulation analysis are depicted in Figure 6.6, (a) shows the minimum, maximum and mean MAC values for the 1,000 resimulated random samples, whereas (b) depicts the relative error of the natural frequencies. All mean MAC values are above 0.99 which shows an extraordinary good resimulation of the global mode shape. Except for the 3rd flapwise mode with a minimum outlier of $MAC = 0.9$, the others show minimum MAC values of $MAC \geq 0.98$ throughout all the 1,000 samples. The natural frequencies achieve mean errors of approximately 0-3%.

Overall, this can be regarded an excellent result for the generic model updating, as in general the modal behavior is captured accurately and the correlation of the input feature prediction in the previous section also gave very good results with $r^2 > 0.5$ for 62% of the parameter and even 57% with $r^2 > 0.7$.

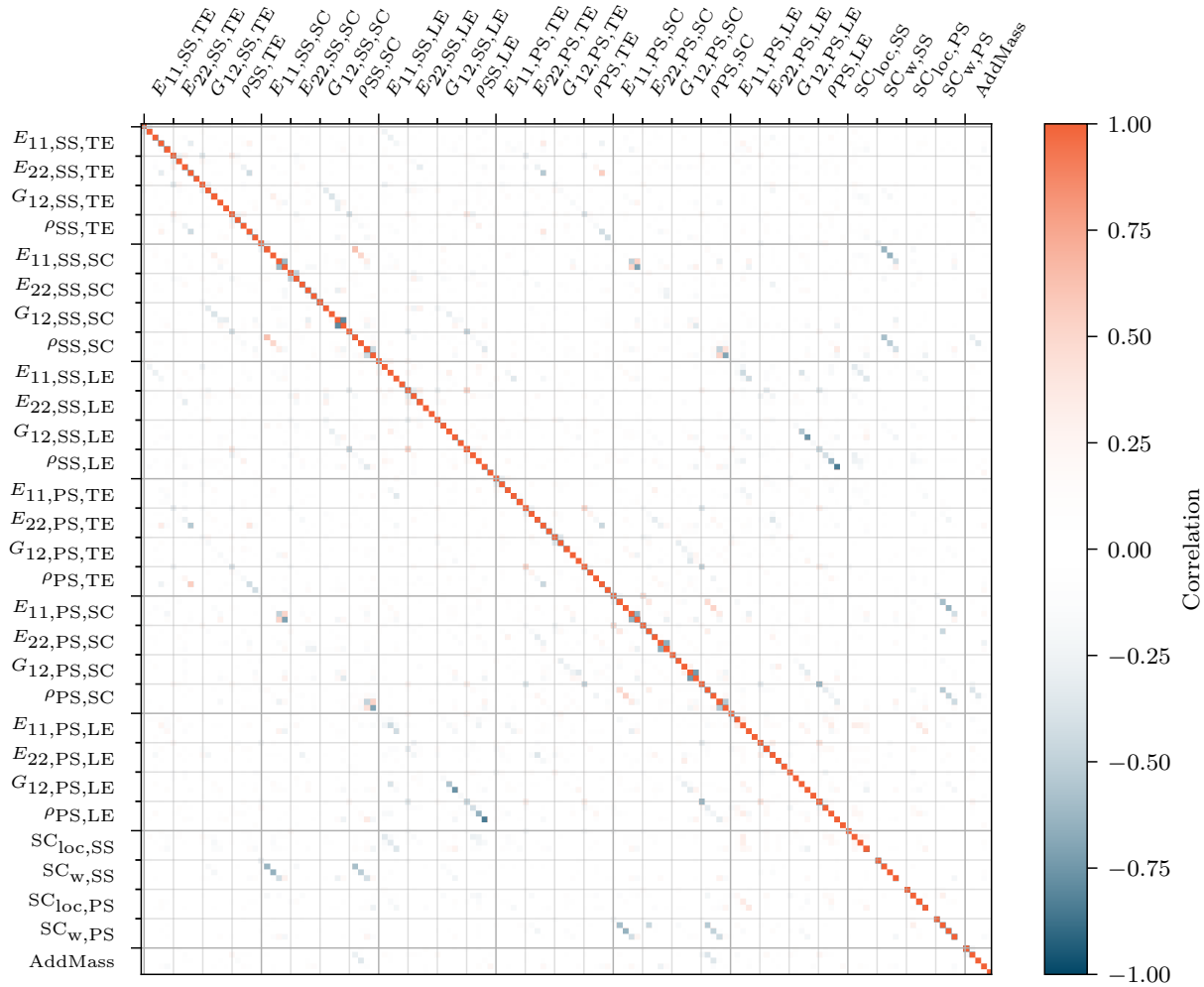


Figure 6.5: Cross correlation matrix of the predicted input parameters to reveal collinearities between predicted parameters.

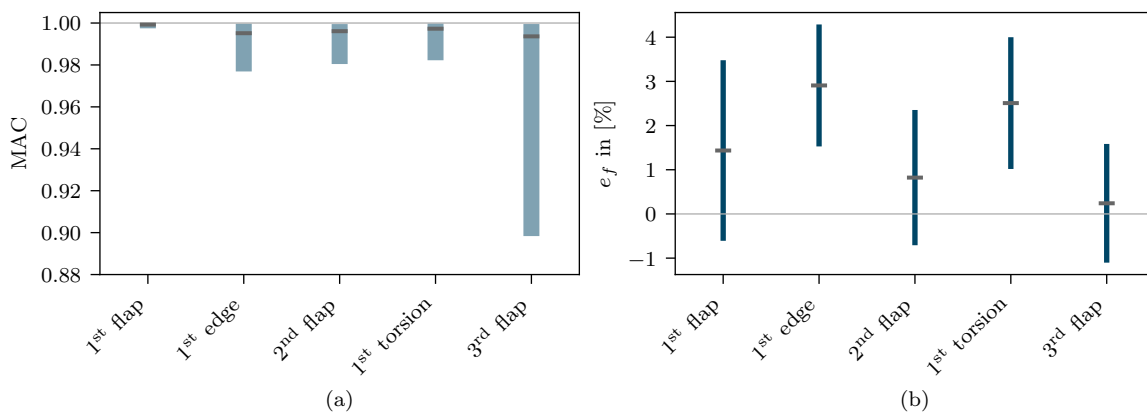


Figure 6.6: Resimulation results for 1,000 random samples. (a) minimum, maximum and mean MAC values of each mode. (b) mean and 1- σ -range of the relative frequency error e_f .

6.4 Model updating based on modal experimental data

As described in Section 6.1, the model updating will be based on free-free vibrating modal responses. The free-free vibration experiments in the SmartBlades2 project are conducted according to [88] with an elastic suspension of the blade and an impact hammer for excitation. This analysis will be applied for the three finished blades number 2-4 of the experiment, while only taking into account the first five extracted eigenmodes: 1st flap, 1st edge, 2nd flap, 1st torsion, 3rd flap.

6.4.1 Updating results

The experimental modal results of the three observed blades was used to predict the posterior distribution of the input features by the trained cINN. The obtained results indicate for most of the features values beyond the training boundaries of the cINN. Considering that the sampling variations are already assumed higher than normally expected during manufacturing, the range should be sufficient to cover possible outliers of the manufactured blades. Especially features with an high r^2 value in the generic model validation (see Figure 6.4) still have unrealistic predictions. Figure 6.7 depicts exemplary three material parameter splines predicted for the three blades. The chosen parameters have very high r^2 values for most of their nodes, but still yield posterior predictions clearly beyond the $\pm 25\%$ variation of the training sampling.

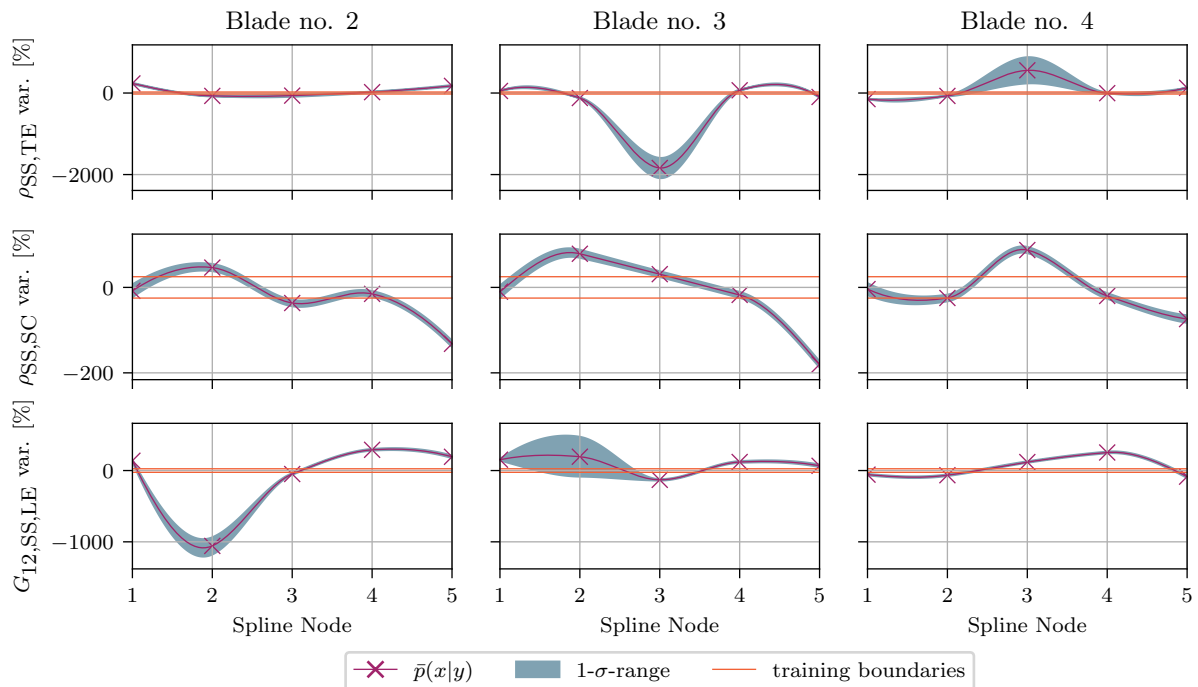


Figure 6.7: Material parameter prediction for all three blades based on the experimental results of the modal analysis. Results are depicted exemplary for three material parameter splines, which are among the best prediction accuracy in the generic updating process as shown in Figure 6.4.

6.4.2 Root cause analysis of the updating inaccuracies

Due to the aforementioned inaccuracies or unrealistic values of the material parameter's posterior prediction, a root cause analysis is conducted to identify possible problems during the updating process. This analysis covers concerns about the cINN itself, the training design and the raw data.

Generalization problems of the cINN

In the first instance, the cINN is checked against its generalization capacity, i.e., if the cINN is overfitting to the training data. The training and validation curve principally do not show any significant overfitting as depicted in Figure 6.8. Both curves are not significantly diverging more with progressed training, which is a good indication for a correct fit. However, it has to be mentioned that both data sets are sampled in the exact same parameter variation range, so it is possible that cINN overfits to this sample design and is not able to generalize beyond this sampling range.

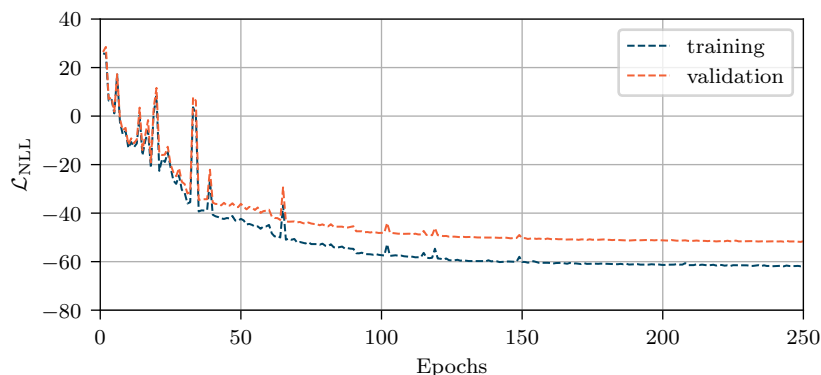


Figure 6.8: Training and validation loss curve of the cINN training.

In order to avoid a possible overfitting, a few typical techniques to reduce this and improve generalization were applied according to [48], such as:

- Weight regularization
- Drop-out layers in the sub-network and/or the conditional network
- Reducing the model capacity, i.e., cINN model size, in order to force the cINN to learn less detailed patterns. But still avoiding any underfitting.
- Apply noise to the modal responses used for the training

However, non of these techniques improved the posterior prediction accuracy of the input features significantly. Therefore, it is assumed, that the cause of the prediction inaccuracy based on the experimental modal response is not primarily found in the cINN design.

Erroneous sample design

The next possible root cause is an erroneous sample design. Obviously the predetermined input parameters, which are varied, do not cover all of the possible manufacturing deviations. Therefore, the input parameter space was extended to consider these additional design parameters:

1. Chord length
2. Relative thickness
3. Twist angle
4. Shear web location
5. Shear web adhesive thickness
6. Shear web adhesive width
7. Trailing edge adhesive width

Together, these aforementioned parameters and the already included ones should cover most of the relevant design variations possible for this particular blades. In the generic model updating the global blade geometry parameters 1.-2. were accurately predicted with $r^2 > 0.8$, the other parameters 3.-7. could only be sufficiently recovered for a few parameter spline nodes ($r^2 > 0.5$), i.e., the model response has no significant information on these input features. However, including all these additional features did not improve the prediction accuracy with the experimental data either. It also has to be mentioned, that the global geometric parameters were included for the sake of completeness, though, the mold was measured in advance to manufacturing and all the deviations were already included in the baseline design. That means, the global geometric parameters, such as chord length, relative thickness and twist angle, were not expected to vary significantly for the manufactured blades.

Discrepancies between raw sample data and experimental data

After discarding the cINN and the sampling design as root cause for the prediction inaccuracies during the model updating, the raw data itself is analyzed on its consistency between sample and experimental data. Therefore, all generated training samples are compared to the experimental data in terms of MAC values and natural frequency errors e_f . The ten best matching samples to the experimental results exemplary for blade no. 2 are listed in Table 6.3. These appear to be quite acceptable, with good agreement according to the relatively good MAC values and frequency errors. As the training is performed including these samples and the general prediction accuracy against this generic data is good, hence it could be expected, that the cINN should find a reasonable solution for the predictable input parameters.

Table 6.3: MAC values and frequency errors for the ten best training samples compared to the experimental results of blade no. 2.

Mode	1 st flap	1 st edge	2 nd flap	1 st torsion	3 rd flap
MAC	0.995	0.971	0.986	0.982	0.964
	0.993	0.965	0.991	0.979	0.964
	0.993	0.970	0.992	0.982	0.957
	0.991	0.958	0.987	0.977	0.974
	0.990	0.964	0.984	0.977	0.967
	0.991	0.970	0.991	0.976	0.960
	0.994	0.964	0.985	0.976	0.965
	0.994	0.958	0.985	0.976	0.973
	0.992	0.967	0.991	0.981	0.958
	0.992	0.962	0.985	0.975	0.969
e_f	-2.6%	3.0%	-2.4%	3.9%	-1.6%
	-0.1%	4.9%	0.2%	6.9%	2.3%
	-2.3%	5.1%	0.6%	5.0%	1.8%
	-3.9%	1.7%	-1.6%	5.1%	0.2%
	-3.2%	4.0%	-1.5%	6.2%	1.2%
	-7.0%	2.8%	-5.2%	3.4%	-2.8%
	-1.4%	4.6%	1.1%	6.7%	2.3%
	-3.0%	2.7%	-1.7%	5.8%	0.6%
	-4.9%	1.7%	-0.9%	5.6%	0.3%
	-6.3%	2.1%	-3.1%	4.2%	-1.5%

However, the MAC values are not part of the loss function the cINN is trained with. Therefore, a closer look is taken on the raw data itself, i.e., the unprocessed sensor data. Figure 6.9 depicts the mean of the displacement difference $\Delta y_{\text{disp}} = y_{\text{disp, samp}} - y_{\text{disp, exp}}$ for each sensor over the three best matching samples, sorted by mode shapes and sensor direction (flapwise/edgewise). The graphs

reveal, the difference Δy_{disp} is independent from the magnitudes of y_{disp} . Whereas, the mode shapes influence the difference, e.g., the flapwise sensors result in lower differences for the flapwise shapes, while the edgewise sensor show higher differences and vice versa during the edgewise shapes. With increasing order of mode shapes the difference also increases, see 1st, 2nd and 3rd flapwise mode. This opens space for the assumption, that flapwise mode shapes result in higher absolute deflections for the flapwise sensors, therefore, an arbitrary absolute signal noise vanishes in magnitude before the values are normalized. Whereas, the edgewise and torsion mode shapes provoke smaller displacement magnitudes for the flapwise sensors, by this any possibly existing noise would result in a significant disturbance, as shown in the graphs. Same holds for higher order mode shapes such as the 3rd flapwise mode, although these are more complex, they generally appear with smaller displacement magnitudes.

Looking further, according to [6], the MAC is formulated to minimize the squared error between two vectors, which makes it insensitive to smaller values, while higher values influence it more severely. Therefore, the MAC is a suitable indication for the global behavior, but does not properly account for any local and small deviations. And as Figure 6.9 clearly shows, there are a lot of sensor responses within a very low magnitude level, 67% with $y_{\text{disp, exp}} < 0.05$. However, when considering a cINN, each of these sensors offer information to infer the posterior distribution of the input parameters. And even small noises with respect to small displacements result in a high relative error. And if all of the sensor signals are deviating significantly, these errors multiply with each weight and pass throughout the network in order to yield a poor posterior prediction. Remembering the model robustness analysis in Section P3-4.3, a 5% random noise error on the responses still gave excellent results, though, in this case the mean relative error is approximately 40%.

In order to exclude two possible numerical and modeling causes for this discrepancies the following aspects were analyzed:

- The convergence study from Section P1-3 was extended to additionally check the displacement convergence at the particular sensor positions. Here the maximum displacement error was only 4% to the previous step for all considered mode shapes.
- The suspended blade is exposed to gravitation and will thus deform, which results in a pre-stress in the blade and a misalignment of sensor orientations compared to an undeformed configuration. Unpublished analyses of the SmartBlades2 project partner DLR showed only very minor static deformations due to dead weight. It was thus assumed that a modal analysis in the undeformed configuration is a sufficiently accurate approximation.

After excluding these two candidates for the significant difference between the numerical solution and the experiments, it has to originate from a more fundamental perspective. Thus, all this condenses to two general questions:

1. *How to model a rotor blade correctly in order to replicate the experimental blade behavior?*

A finite element shell model compromises easy modeling, efficient computation and reasonably good results. Therefore, it lacks an accurate modeling of the geometry, which causes an increased stiffness and mass in convex shapes. Additionally, the node offsets from the mid-plane generate inaccuracies in torsional prediction as already discussed in Section P1-5.3. However, local and accurate responses are necessary for the model updating approach, which might be achieved by continuum shell element or even better solid element modeling.

2. *How to adequately design experiments to provide good data for the model updating?*

Although, the applied experimental results were used to manually or automatically update

finite element models, see Knebusch et al. [117], they were formerly not dedicatedly designed for the use in this particular model updating approach. The primary focus was to best and fully characterize the modal blade behavior and test the experimental setup. For a dedicated experimental design, structural characteristics are desirable, where higher magnitudes are measured to reduced the impact of measurement noise. In case of modal analysis, an increased excitation than that of a modal impact hammer might be a solution. Ideally, the extracted structural characteristics should represent a global response instead of local details. This fidelity level would be more consistent with the updateable input parameters, which are more of a global type.

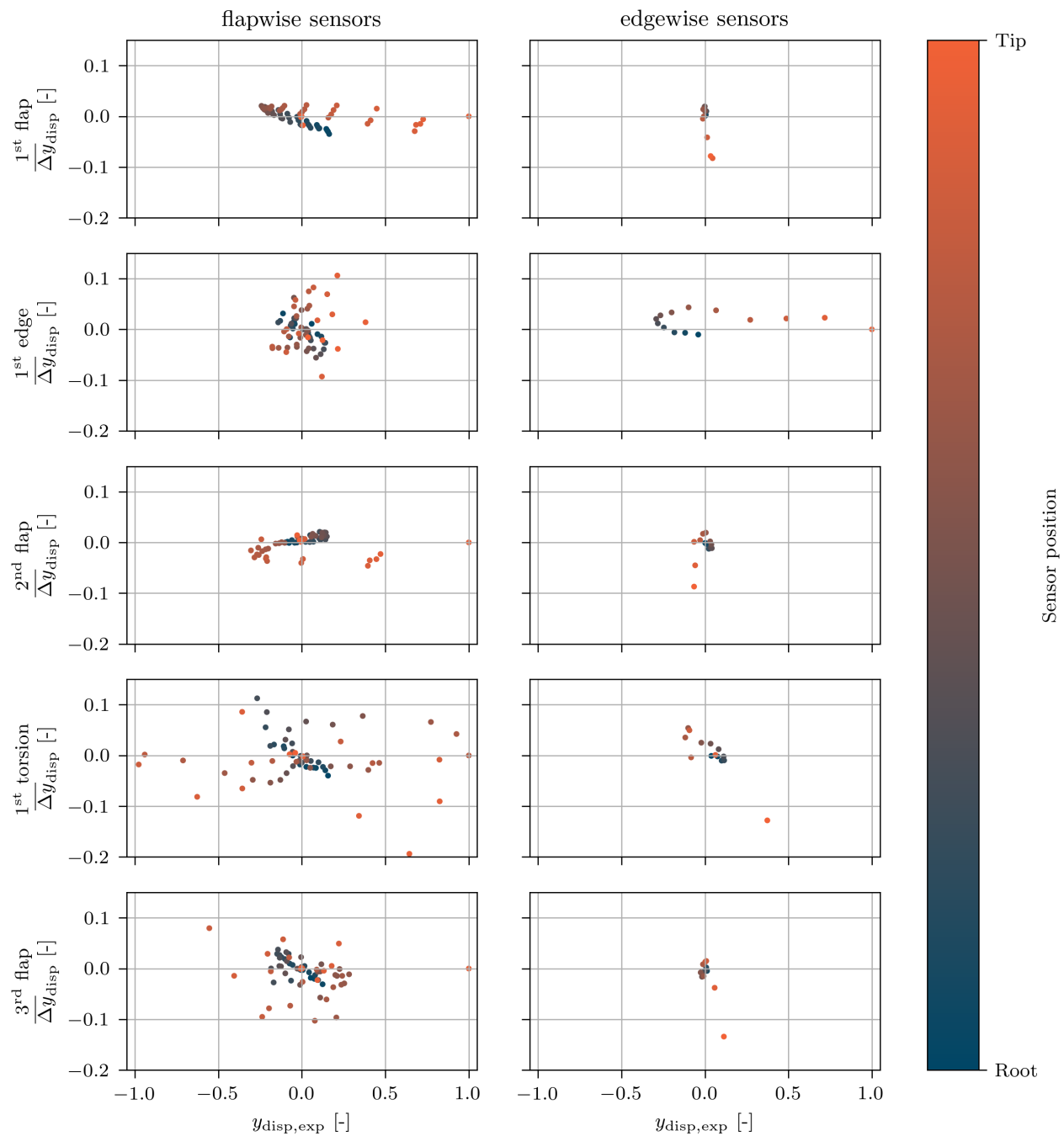


Figure 6.9: Displacement difference Δy_{disp} between training sample $y_{\text{disp,samp}}$ and experiment $y_{\text{disp,exp}}$. This is divided into flapwise and edgewise sensors as well as into the different mode shapes.

7

Conclusion and Outlook

This thesis concludes with a summary of all relevant aspects of the conducted research. And ends with an outlook on future research inspired by open question, as well as possible fields of application for the elaborated model updating approach.

7.1	Conclusion	120
7.2	Outlook	122

To finish this thesis, an overall conclusion for the presented publications and additional chapters is drawn, putting the different results in the lights of the thesis' objective. After that, an outlook will gather open questions raised by this thesis and propose future research focuses.

7.1 Conclusion

The overall aim of this thesis was to **establish a computationally efficient approach to recover the deviation and uncertainty of rotor blade design parameters by updating higher fidelity models**. Therefore, the work is broken down into five significant work packages: 1) development of a model generator for wind turbine blades, 2) validation of the modeling methodology, 3) establishing an updating approach with uncertainty evaluation for state-of-the-art models, 4) proving the efficiency of the approach against current techniques, and finally 5) applying the updating on higher fidelity models.

Focusing on the first two work packages, the first paper *Validation of a modeling methodology for wind turbine rotor blades based on a full-scale blade test* presents a fully parameterized modeling methodology implemented in the software tool MoCA. It provides output for different model fidelity levels, such as BECAS cross sections, finite element Timoshenko beam models and hybrid shell & solid 3D finite element models. The parameterization is defined based on spline distributions for nearly each parameter of the model, thus, it enables an easy way to vary the distribution by simply adding or multiplying spline distributed variations. The subsequent validation is performed on the example of the SmartBlades2 20 m demonstration blade, which was modeled with MoCA and exported as hybrid shell & solid 3D finite element model. A blade was built according to this design during the project, then accurately measured and tested. The test comprised modal analysis as well as static bending and torsion. After the test, the blade is cut into segments, which were characterized for further information. Every known manufacturing deviation and additional information, such as adhesive thickness, geometry deviation or sensor weights, was replicated in the finite element model. Then, the model is successfully validated with the given experimental results. The mass and center of gravity of the full blade and each segment matched very well with only a few measurements exceeding 5% error. The natural frequencies also gave reasonably good results with error of up to 8%. For the experiments under pure static bending in both, flapwise and edgewise direction, the deflection error mostly complied with the validation threshold of 5% and also the local strain measurements in general agreed with the local threshold of 10%. However, torsion resulted in significant deviations, which exceeded the deflection thresholds by up to 12% and local strain threshold by a mean of 30%. The results agree with the poor torsional prediction of shell elements with offset nodes to the element's mid plane. This can only be avoided by layered solid element modeling of the composite parts. Summarizing, this publication accomplished the first work package and generally the second, except for the torsional behavior. All following applications use models derived from the validated MoCA model of the SmartBlades2 demonstration blade.

The next step was to establish a model updating approach with uncertainty evaluation of the updated parameters. As current model updating approaches are either not probabilistic, such as heuristic optimizations, or computationally very expensive, as e.g., Markov Chain Monte Carlo methods for approximated Bayesian computation, an alternative method is presented, the conditional invertible neural networks. These networks are bijective due to their mathematical structure and can thus be easily inverted. While following Bayes' theorem and training against the negative log-likelihood loss, they are capable of representing an inverse surrogate of the forward physical modeling process. They provide an uncertainty evaluation by predicting the posterior parameter distribution based on observed model response and the learned mapping. This method, therefore, fulfills all the requirements of the third work package. It is first tested in a feasibility study in the

second paper *Model updating of wind turbine blade cross sections with invertible neural networks* with excellent results. Then, it is successfully applied to update a state-of-the-art full wind turbine blade model in the third paper *Model updating of a wind turbine blade finite element Timoshenko beam model with invertible neural networks*. The feasibility study was based on cross sectional models of the demonstration blade, with updateable model parameters consisting of material and geometrical properties. The observations in this example are cross sectional properties such as, e.g., stiffness matrix, mass matrix, and elastic center. After reducing the parameter space to a relevant subspace via an one-at-a-time sensitivity analysis, training, validation and test samples are created with MoCA. The trained cINN-model accurately predicts the input parameters with generally low uncertainties. The extension to a model fidelity used in the state-of-the-art model updating is done in the third paper. Here, the cINN proved that it is also capable of accurately recovering model parameters for a full blade model with 153 variable model parameters, in this case a Timoshenko beam model. Therefore, the cINN was extended by a conditional network, which pre-processes the observations before passing them to the coupling blocks. Both networks, the conditional and the cINN, are trained jointly. The method proved to be robust even against 5% noise on the observations. Although some of the results could not be accurately predicted due to inherited model ambiguities, the cINN correctly understands and models the physics behind the data. A cross-correlation of the predicted features reveals ambiguous model parameters and helps to redefine the model updating problem in order to enhance the model updating accuracy. Additionally, this method is also able to deal with full data sets including irrelevant data, without losing accuracy, while common optimization approaches require a parameter space reduction to be able to computationally deal with more complex models at all. With this, the third work package is done and the model updating method is established and successfully tested at current fidelity level.

A benchmark study presented in Chapter 5 briefly compares the cINN model updating method with approaches based on four different heuristic optimization algorithms. Due to the high number of model evaluations, the forward process has to be replaced by an artificial neural network as surrogate model for the MoCA modeling process and evaluation of a full blade beam model. The time reduction for a single evaluation was from approximately 80 s to 0.25 ms with the surrogate model. The results show that the optimization approaches do not even rudimentarily reach the accuracy of the cINN. And when interpolating the performance on real modeling time, the cINN outperforms the fastest algorithm by the factor of 2.5. Thus, the fourth work package, including the proof of computational efficiency compared to state-of-the-art approaches, is accomplished.

In the final chapter, the fidelity level of the considered model is increased a further step to a hybrid shell and solid finite element model to fulfill the last work package. This time, the model parameter definition was changed to reduce the inherited ambiguities. The model updating of a generic problem presented good predictions of the posterior distributions of the model parameters with relatively low uncertainty. The generic problem was designed to replicate an experimental modal analysis performed on a real blade. However, when applying the trained cINN to update the model with experimental modal results, the accuracy drops significantly. A thorough root cause analysis especially of the experimental data compared to the training sample data revealed significant deviations, which might be caused either by relevant noise in the experimental measurements exceeding by far the 5% for low magnitudes, or by modeling-inaccuracies of the finite element model.

Despite its lacking accuracy when applied to modal test data, this thesis has proposed and successfully tested a computationally efficient updating approach on generic problems for high fidelity structural rotor blade models taking into account uncertainties of the predicted model parameters.

7.2 Outlook

Having established the method, further studies could focus on transferring this to a industrial use case. Therefore, Chapter 6 highlighted important future research questions to overcome the significant deviations of experimental and sample data. First: *how to model a rotor blade correctly in order to replicate the experimental blade behavior?* Finite element shell models comprise efficient modeling and computation with reasonably good results, but lacks in accuracy concerning the exact structural properties of a blade. An increased fidelity level of the model with, e.g., layered solid elements, should avoid any geometric discontinuities, additional masses and stiffness of overlapping elements, or inaccuracy in torsional prediction, and thus minimize any modeling errors. The second question is: *how to adequately design experiments to provide good data for the model updating?* Here, the focus should be on the design of experiments. The presented modal analysis data used in Chapter 6 was not originally designed for the purpose of model updating via invertible neural networks, which led to inaccurate parameter predictions. A thorough research must determine an appropriate experimental design with the necessary accuracy and extract, at best, noise insensitive characteristics, which can be used for the model updating process. As the input parameter splines represent a rather global distribution of a property over the blade, the measured response, which is used for the update, should also follow a more global character. The measurement of too local effects may lead to problems in the updating process, as the global updateable parameters may never be tuned to represent such local effects. Beyond these two highlighted questions, the progress in the research of inverse modeling and invertible neural networks should be carefully followed. Here, a field of interest can be generalization capabilities of invertible neural networks, as this could make the updating process less prone to noisy observation.

A successfully trained and tested invertible neural network in conjunction with the correct experimental measurement setup could open different fields of application. As the trained cINN represents a surrogate of the inverse physical model, it can cover all types of parameter variations and thus possible physical states of a blade, whenever it is correctly trained. Additionally, it has the advantage that it can be evaluated at practically no extra cost. One possible application for such an inverse model is to enhance quality assurance during manufacturing, because such an inverse model could easily update a baseline model for each blade of a series. Therefore, a simple and quick test method must be designed, applicable to a blade after manufacturing but before shipment/installation. This may help to detect possible manufacturing deviations rapidly and avoid unnecessary actions, when the blade is already attached to the turbine. Another application, can be of course structural health monitoring, where the cINN can update a digital twin of the blade for the current conditional state. All these approaches should be realized with the aim of evaluating uncertainties of the blade properties and by that improving the overall reliability. At the end, this model updating method is universal. It can easily be transferred to other structures or problems.

List of Figures

1.1	Exemplary cross sectional plot from BECAS	4
1.2	Exemplary beam finite element	5
1.3	Exemplary shell finite element model	6
1.4	Exemplary solid finite element model	7
1.5	Categories of optimization methods	8
1.6	Forward process in wind turbine blade modeling	11
1.7	Thesis Outline	14
P1-1	Flowchart of the model creation with MoCA	21
P1-2	Flowchart of the cross sectional shape generation	22
P1-3	Flowchart of the 2D cross sectional meshing	22
P1-4	Flowchart of the TestRig module	23
P1-5	Cross section cut at radial position of $r = 5.2$ m.	24
P1-6	Cross sectional few of FE model	24
P1-7	Setup for mass and center of gravity measurements	25
P1-8	Static blade test configuration	25
P1-9	Static blade torsional test configuration	27
P1-10	Cross sectional center of gravity measurement setup	28
P1-11	Comparison of bending lines at front draw-wire sensors	30
P1-12	Comparison of torsional twist	31
P1-13	Strain comparison at a radial position of $r = 5$ m and $r = 8$ m	33
P1-14	Segment mass and center of gravity comparison	34
P1-15	Appendix A: Comparison of bending lines at rear draw-wire sensors	36
P1-16	Appendix B: Cross sectional sensor distribution at $r = 5$ m	37
P1-17	Appendix B: Strain comparison at $r = 5$ m	37
P1-18	Appendix B: Cross sectional sensor distribution at $r = 8$ m	38
P1-19	Appendix B: Strain comparison at $r = 8$ m	38
P2-1	Overall workflow of this study	44
P2-2	Sensitivity analysis based feature selection process	45
P2-3	Schematic cINN working principle	46
P2-4	Structure of a conditional affine coupling block	47
P2-5	The conditional invertible neural network structure	47
P2-6	Cross sectional of the SmartBlades2 DemoBlade at a radial position of $R = 6$ m.	48
P2-7	Exemplary standardized prediction for two input features	53
P2-8	Input feature prediction for 10 random samples of the four different models	53
P2-9	Counteraction of the mean predicted densities for ρ_{Biax90} and ρ_{Triax}	54
P2-10	Final model feature prediction for 30 random samples	55
P2-11	Feature prediction for 10 random samples for four different radial positions	55

P2-12	Feature prediction for four different radial positions based only on the stiffness and mass matrix	55
P2-13	Appendix B: Correlation of input feature prediction and ground truth	67
P2-14	Appendix B: Cross sectional of the SmartBlades2 DemoBlade at a radial position of $R = 12$ m.	68
P3-1	Exemplary finite element beam model	73
P3-2	Exemplary parameter variation spline	73
P3-3	Conditional Coupling Block	76
P3-4	Conditional invertible neural network architecture	77
P3-5	Subnetwork architecture	78
P3-6	Schematic view of cINN transformation	78
P3-7	Posterior distribution prediction example	79
P3-8	Posterior prediction for a random sample	80
P3-9	Posterior prediction correlation for multiple samples	81
P3-10	Interaction of ρ_{Biax90} and ρ_{Triax}	82
P3-11	Interaction of $E_{11,\text{Biax90}}$ and $E_{11,\text{Triax}}$	82
P3-12	Interaction of $G_{12,\text{Biax90}}$ and $G_{12,\text{Triax}}$	82
P3-13	Schematic blade cross sectional view	82
P3-14	Rotor blade's layup of Biax90° and Triax	83
P3-15	Exemplary inferred parameter spline prediction	84
P3-16	Mean and standard deviation of MAC matrices	85
P3-17	Mean, minimum and maximum MAC values of multiple samples	85
P3-18	Mean and standard deviation of natural frequency of multiple samples	85
P3-19	Cross-correlation of all predicted features	86
P3-20	Appendix A: Posterior prediction correlation with noise	89
P3-21	Appendix A: Posterior prediction correlation full input space	89
5.1	Structure of the surrogate model	95
5.2	Training and validation loss of surrogate model	95
5.3	Convergence of the training and validation loss of the cINN	96
5.4	Convergence of all optimization approaches	98
5.5	Average number of model evaluations for each updating approach	98
5.6	Average time for each updating approach with surrogate model	99
5.7	Input predictions of a single sample	100
5.8	Input predictions for multiple samples	101
5.9	Re-simulated MAC values of all 20 mode shapes	102
5.10	Re-simulated natural frequencies of all 20 mode shapes	103
5.11	Average time for each updating approach extrapolated to <i>physical model</i>	103
6.1	Sensor distribution along the DemoBlade	106
6.2	Schematic cross section defining the input parameter	107
6.3	Correlation of posterior prediction and target values	109
6.4	Coefficient of determination r^2 of the parameter prediction against the ideal results	110
6.5	Cross correlation of the predicted parameters	112
6.6	MAC values and frequency errors after the resimulation	112
6.7	Material parameter prediction based on experimental data	113
6.8	Training and validation loss curve of the cINN	114
6.9	Displacement difference between best training samples and experiment	118

List of Tables

P1-1	Total mass and center of gravity comparison	28
P1-2	Modal analysis comparison with free-free and the test rig setup	29
P2-1	Parameter variation range for sensitivity analysis and neural network training	49
P2-2	BECAS cross sectional output parameters	49
P2-3	Reduced sensitivity matrix (CS at $R = 6$ m)	50
P2-4	Different cINN hyperparameter sets	53
P2-5	Final cINN hyperparameter set	54
P2-6	Appendix A: Shell layup at $R = 6$ m	60
P2-7	Appendix A: Spar cap layup at $R = 6$ m	60
P2-8	Appendix A: Full sensitivity matrix (CS at $R = 6$ m) - Cross sectional properties	61
P2-9	Appendix A: Full sensitivity matrix (CS at $R = 6$ m) - Stiffness matrix . .	63
P2-10	Appendix A: Full sensitivity matrix (CS at $R = 6$ m) - Mass matrix	65
P3-1	Input feature list	74
P3-2	Feature selection with sensitivity analysis	75
P3-3	Conditional invertible neural network hyperparameter set	77
P3-4	Discrepancies of sensitivity to prediction	80
P3-5	Stiffness and Density ratios of Biax90° and Triax	83
P3-6	Feature selection discrepancies of cINN and sensitivity analysis	87
P3-7	Appendix A: Feature selection comparison	88
P3-8	Appendix A: Identified mode shapes	88
5.1	Mean and standard deviation of model loss $\mathcal{L}_{\text{Modal}}$ and input prediction error e_x for all approaches.	100
6.1	Input feature list for the 3D finite element blade model updating	107
6.2	Hyperparameter set of the complete artificial neural network	108
6.3	MAC values and frequency errors for the ten best training samples compared to the experimental results	115

List of Abbreviations

This abbreviation list contains only those abbreviations included in non-paper sections.

ABC	Approximated Bayesian computation
AdaGrad	Adaptive gradient algorithm
AddMass	Additional mass
ANN	Artificial neural network
BECAS	Beam cross sectional analysis software
Biax45	Bi-axial laminate with $\pm 45^\circ$ fiber directions
Biax90	Bi-axial laminate with 0° and 90° fiber directions
CC	Coupling block
cINN	Conditional invertible neural network
Clst	Cluster
CoG	Center of gravity
Conv	Convolution
CS	Cross section
DLR	German aerospace center
FAST	Fatigue, aerodynamics, structures, and turbulence code
FE	Finite element
FEM	Finite element model
GA	Genetic optimization algorithm
GAN	Generative adversarial network
GPU	Graphics processing unit
HAWC2	Horizontal Axis Wind turbine simulation Code 2 nd generation
INN	Invertible neural network
LE	Leading edge
loc	Location
LP	Linear problem
MAC	Modal assurance criterion
MCMC	Markov chain Monte Carlo technique
MoCA	Model creator and analyzer for wind turbine rotor blades
NLP	Non-linear problem
NM	Nelder-Mead optimization algorithm
NREL	National renewable energy laboratory
PMC	Population Monte Carlo technique
PReLU	Parametric rectified linear unit
PS	Pressure side
PSO	Particle swarm optimization algorithm
ReLU	Rectified linear unit

RMSE	Root-mean-square-error
SC	Spar cap
SimAn	Simulated annealing optimization algorithm
SMC	Sequential Monte Carlo technique
SS	Suction side
TE	Trailing edge
Triax	Tri-axial laminate with -45° , 0° and 45° fiber directions
VAE	Variational autoencoder

Nomenclature

This nomenclature contains only those symbols included in non-paper sections.

Latin Symbols

Symbol	Description	Unit
D	Data	-
\mathbf{d}_p	Matrix with predicted displacement vectors	m
\mathbf{d}_t	Matrix with target displacement vectors	m
e	Root mean square error	-
e_f	Relative frequency error	-
e_{MAC}	Modal assurance error	-
$e_{\text{mse},f}^2$	Mean square error of the natural frequency	Hz
f_p	Predicted natural frequency	Hz
f_t	Target natural frequency	Hz
$\mathcal{L}_{\text{Modal}}$	Modal loss	-
\mathcal{L}_{NLL}	Negative log-likelihood loss	-
N	Spline Node	-
n_{eval}	Number of evaluations	-
n_{modes}	Number of modes	-
p	Probability	-
r^2	Coefficient of determination	-
t	Time	s
x	Input value	-
x_p	Predicted input value	-
x_t	Target input value	-
y	Output value	-

Greek Symbols

Symbol	Description	Unit
σ	Standard deviation	-
θ	Model parameters	-

Bibliography

This bibliography list contains only those publications cited in non-paper sections.

- [1] Mohd Nadhir Ab Wahab, Samia Nefti-Meziani, and Adham Atiyabi. A comprehensive review of swarm optimization algorithms. *PloS one*, 10(5):e0122827, 2015.
- [2] N. A. Z. Abdullah, M. S. M. Sani, M. M. Rahman, and I. Zaman. A review on model updating in structural dynamics. *IOP Conference Series: Materials Science and Engineering*, 100:012015, 2015.
- [3] Shady Abu Hussein, Tom Tirer, and Raja Giryes. Image-adaptive gan based reconstruction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3121–3129, 2020.
- [4] Tim J. Adler, Lynton Ardizzone, Anant Vemuri, Leonardo Ayala, Janek Gröhl, Thomas Kirchner, Sebastian Wirkert, Jakob Kruse, Carsten Rother, Ullrich Köthe, and Lena Maier-Hein. Uncertainty-aware performance assessment of optical imaging modalities with invertible neural networks. *International journal of computer assisted radiology and surgery*, 14(6):997–1007, 2019.
- [5] Zeynep Akata, Andreas Geiger, and Torsten Sattler, editors. *Pattern Recognition*. Lecture Notes in Computer Science. Springer International Publishing, Cham, 2021. ISBN 978-3-030-71277-8.
- [6] R. J. Allemang. The modal assurance criterion - twenty years of use and abuse. *Sound and Vibration*, 37(8):14–23, 2003.
- [7] Manal Almaeen, Yasir Alanazi, Nobuo Sato, W. Melnitchouk, Michelle P. Kuchera, and Yaohang Li. Variational autoencoder inverse mapper: An end-to-end deep learning framework for inverse problems. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021. ISBN 978-1-6654-3900-8.
- [8] Govinda Anantha Padmanabha and Nicholas Zabarar. Solving inverse problems using conditional invertible neural networks. *Journal of Computational Physics*, 433:110194, 2021.
- [9] Alexandros Antoniou, Kim Branner, D. J. Lekou, Iñaki Nuin, and Rogier Nijssen. Methodology for testing subcomponents; background and motivation for subcomponent testing of wind turbine rotor blades. 2016. URL https://orbit.dtu.dk/files/125632907/D71_1_Review_blade_design_revised.pdf.
- [10] Lynton Ardizzone, Jakob Kruse, Sebastian Wirkert, Daniel Rahner, Eric W. Pellegrini, Ralf S. Klessen, Lena Maier-Hein, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. In *Seventh International Conference on Learning Representations*, 2019. URL <http://arxiv.org/pdf/1808.04730v3>.
- [11] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. 2019. URL <http://arxiv.org/pdf/1907.02392v3>.
- [12] Lynton Ardizzone, Jakob Kruse, Carsten Lüth, Niels Bracher, Carsten Rother, and Ullrich Köthe. Conditional invertible neural networks for diverse image-to-image translation. In Zeynep Akata, Andreas Geiger, and Torsten Sattler, editors, *Pattern Recognition*, volume 12544 of *Lecture Notes in Computer Science*, pages 373–387. Springer International Publishing, Cham, 2021. ISBN 978-3-030-71277-8.
- [13] Turaj Ashuri and Jie Zhang. Compsim: Cross sectional modeling of geometrical complex and inhomogeneous slender structures. *SoftwareX*, 6:155–160, 2017.
- [14] Mohamad Eydani Asl, Christopher Niezrecki, James Sherwood, and Peter Avitabile. Design of scaled-down composite i-beams for dynamic characterization in subcomponent testing of a wind turbine blade. In Anders Brandt and Raj Singhal, editors, *Shock & Vibration, Aircraft/Aerospace, Energy Harvesting, Acoustics & Optics, Volume 9*, Conference Proceedings of the Society for Experimental Mechanics Series, pages 197–209. Springer International Publishing, Cham, 2016. ISBN 978-3-319-30086-3.

- [15] Dawid Augustyn, Ursula Smolka, Ulf T. Tygesen, Martin D. Ulriksen, and John D. Sørensen. Data-driven model updating of an offshore wind jacket substructure. *Applied Ocean Research*, 104:102366, 2020.
- [16] Enrique Baeyens, Alberto Herreros, and José Perán. A direct search algorithm for global optimization. *Algorithms*, 9(2):40, 2016.
- [17] Richard S. Barr, Bruce L. Golden, James P. Kelly, Mauricio G. C. Resende, and William R. Stewart. Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics*, 1(1):9–32, 1995.
- [18] O. A. Bauchau, C. L. Bottasso, and Y. G. Nishkov. Modeling rotorcraft dynamics with finite element multibody procedures. *Mathematical and Computer Modelling*, 33(10-11):1113–1137, 2001.
- [19] Mokhtar S. Bazaraa, Hanif D. Sherali, and Chitharanjan M. Shetty. *Nonlinear programming: Theory and algorithms*. Wiley-Interscience, Hoboken, NJ, 3. ed. edition, 2006. ISBN 9780471486008.
- [20] Mark A. Beaumont. Approximate bayesian computation in evolution and ecology. *Annual Review of Ecology, Evolution, and Systematics*, 41(1):379–406, 2010.
- [21] Amir Beck. *Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB*, volume 19 of *MOS-SIAM series on optimization*. SIAM Society for Industrial and Applied Mathematics, Philadelphia, Pa., 2014. ISBN 978-1-611973-64-8. URL <http://www.loc.gov/catdir/enhancements/fy1503/2014029493-b.html>.
- [22] Anis Ben Abdesslem, Nikolaos Dervilis, David Wagg, and Keith Worden. Model selection and parameter estimation in structural dynamics using approximate bayesian computation. *Mechanical Systems and Signal Processing*, 99:306–325, 2018.
- [23] V. L. Berdichevskii. Variational-asymptotic method of constructing a theory of shells. *Journal of Applied Mathematics and Mechanics*, 43(4):711–736, 1979.
- [24] Jonathan C. Berg and Brian R. Resor. Numerical manufacturing and design tool (numad v2.0) for wind turbine blades: User’s guide, 2012.
- [25] Gunjit S. Bir. Computerized method for preliminary structural design of composite wind turbine blades. *Journal of Solar Energy Engineering*, 123(4):372–381, 2001.
- [26] Gunjit S. Bir. User’s guide to precomp (pre-processor for computing composite blade properties). Colorado, 2006. URL <https://www.nrel.gov/docs/fy06osti/38929.pdf>.
- [27] José Pedro Blasques. Multi-material topology optimization of laminated composite beams with eigenfrequency constraints. *Composite Structures*, 111:45–55, 2014.
- [28] José Pedro Blasques and Mathias Stolpe. Multi-material topology optimization of laminated composite beam cross sections. *Composite Structures*, 94(11):3278–3289, 2012.
- [29] José Pedro Blasques and Mathias Stolpe. Multi-material topology optimization of laminated composite beam cross sections. *Composite Structures*, 94(11):3278–3289, 2012.
- [30] Christian Blum, editor. *Swarm Intelligence: Introduction and Applications*. EBL-Schweitzer. Springer, Berlin and Heidelberg, 2008. ISBN 9783540740889. URL <http://swb.ebib.com/patron/FullRecord.aspx?p=364262>.
- [31] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence: From natural to artificial systems*. Santa Fe Institute studies in the sciences of complexity. Oxford Univ. Press, New York, NY, 1999. ISBN 0-19-513159-2. URL <http://www.loc.gov/catdir/enhancements/fy0606/98049821-d.html>.
- [32] C. L. Bottasso, F. Campagnolo, and A. Croce. Multi-disciplinary constrained optimization of wind turbines. *Multibody System Dynamics*, 27(1):21–53, 2012.
- [33] C. L. Bottasso, S. Cacciola, and A. Croce. Estimation of blade structural properties from experimental data. *Wind Energy*, 16(4):501–518, 2013.
- [34] C. L. Bottasso, F. Campagnolo, A. Croce, S. Dilli, F. Gualdoni, and M. B. Nielsen. Structural optimization of wind turbine rotor blades by multilevel sectional/multibody/3d-fem analysis. *Multibody System Dynamics*, 32(1):87–116, 2014.

- [35] Stephen Boyd and Lieven Vandenbergh. *Convex optimization*. Cambridge University Press, Cambridge and New York, NY and Port Melbourne, 2004. ISBN 978-0-521-83378-3. URL <http://www.loc.gov/catdir/description/cam041/2003063284.html>.
- [36] Omid Bozorg-Haddad, Mohammad Solgi, and Hugo A. Loaiciga. *Meta-heuristic and evolutionary algorithms for engineering optimization*, volume v.294 of *Wiley Series in Operations Research and Management Science*. Wiley, Hoboken, New Jersey, 2017. ISBN 9781119387077. URL <https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=5015534>.
- [37] Anders Brandt and Raj Singhal, editors. *Shock & Vibration, Aircraft/Aerospace, Energy Harvesting, Acoustics & Optics, Volume 9*. Conference Proceedings of the Society for Experimental Mechanics Series. Springer International Publishing, Cham, 2016. ISBN 978-3-319-30086-3.
- [38] Kim Branner, Peter Berring, Christian Berggreen, and Henrik Witthøft Knudsen. Torsional performance of wind turbine blades - part ii: Numerical validation. In *16th International Conference on Composite Materials*. 2007.
- [39] Marlene Bruns, Benedikt Hofmeister, Tanja Griefsmann, and Raimund Rolfes. Comparative study of parameterizations for damage localization with finite element model updating. In Michael Beer and Enrico Zio, editors, *Proceedings of the 29th European Safety and Reliability Conference (ESREL)*, pages 1125–1132. Research Publishing Services, 2019. ISBN 978-981-11-2724-3.
- [40] Marlene Bruns, Benedikt Hofmeister, Dorian Pache, and Raimund Rolfes. Finite element model updating of a wind turbine blade—a comparative study. In *EngOpt 2018 Proceedings of the 6th International Conference on Engineering Optimization*, pages 569–580. Springer International Publishing, Cham, 2019.
- [41] Falko Bürkner. *Biaxial dynamic fatigue tests of wind turbine blades: Dissertation*. Fraunhofer IRB-Verlag, 2020. ISBN 978-3-8396-1710-6.
- [42] Edilson Alexandre Camargo, Jens Peder Ulfkjaer, Rune Brincker, Jens Nørgaard, and Simon Steen Gadegaard. Operational modal analysis and finite-element model updating of pilot concrete wind turbine tower. *Journal of Structural Engineering*, 145(2):05018003, 2019.
- [43] V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985.
- [44] Carlos E. S. Cesnik and Dewey H. Hodges. Vabs: A new concept for composite rotor blade cross-sectional modeling. *Journal of the American Helicopter Society*, 42(1):27–38, 1997.
- [45] Hui Chen, Wenbin Yu, and Mark Capellaro. A critical assessment of computer tools for calculating composite wind turbine blade properties. *Wind Energy*, 2010.
- [46] Xiao Chen, Peter Berring, Steen Hjelm Madsen, Kim Branner, and Sergei Semenov. Understanding progressive failure mechanisms of a wind turbine blade trailing edge section through subcomponent tests and nonlinear fe analysis. *Composite Structures*, 214:422–438, 2019.
- [47] Mayank Chetan, Shulong Yao, and D. Todd Griffith. Multi-fidelity digital twin structural model for a sub-scale downwind wind turbine rotor blade. *Wind Energy*, 24(12):1368–1387, 2021.
- [48] François Chollet. *Deep learning with Python*. Safari Tech Books Online. Manning, Shelter Island, NY, 2018. ISBN 9781617294433. URL <http://proquest.safaribooksonline.com/9781617294433>.
- [49] Edwin Kah Pin Chong and Stanislaw H. Żak. *An introduction to optimization*. Wiley series in discrete mathematics and optimization. Wiley, Hoboken, NJ, 4. ed. edition, 2013. ISBN 978-1-118-27901-4.
- [50] Zhihua Cui and Xiaozhi Gao. Theory and applications of swarm intelligence. *Neural Computing and Applications*, 21(2):205–206, 2012.
- [51] Eun Da Kang, Eric W. Pellegrini, Lynton Ardizzone, Ralf S. Klessen, Ullrich Koethe, Simon C. O. Glover, and Victor F. Ksoll. Emission-line diagnostics of h ii regions using conditional invertible neural networks. *Monthly Notices of the Royal Astronomical Society*, 512(1):617–647, 2022.
- [52] Cuong Dao, Behzad Kazemtabrizi, and Christopher Crabtree. Wind turbine reliability data review and impacts on levelised cost of energy. *Wind Energy*, 22(12):1848–1871, 2019.
- [53] Anubhav Datta and Wayne Johnson. Three-dimensional finite element formulation and scalable domain decomposition for high-fidelity rotor dynamic analysis. *Journal of the American Helicopter Society*, 56(2):22003–2200314, 2011.

- [54] Lu Deng and C. S. Cai. Bridge model updating using response surface method and genetic algorithm. *Journal of Bridge Engineering*, 15(5):553–564, 2010.
- [55] Alexander Denker, Maximilian Schmidt, Johannes Leuschner, and Peter Maass. Conditional invertible neural networks for medical imaging. *Journal of imaging*, 7(11), 2021.
- [56] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. 2014. URL <http://arxiv.org/pdf/1410.8516v6>.
- [57] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. 2016. URL <http://arxiv.org/pdf/1605.08803v3>.
- [58] DNV GL AS. Dnvgl-st-0376 - rotor blades for wind turbines, 2015. URL <https://rules.dnvgl.com/docs/pdf/DNVGL/ST/2015-12/DNVGL-ST-0376.pdf>.
- [59] Benjamin Doerr and Frank Neumann. *Theory of Evolutionary Computation*. Springer International Publishing, Cham, 2020. ISBN 978-3-030-29413-7.
- [60] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4):28–39, 2006.
- [61] N. P. Duineveld. Structure and possibilities of the focus desgin package. 2008.
- [62] EERA JP Wind. R&i strategy, 2014. URL https://www.eera.jpwind.eu/wp-content/uploads/JP-Wind-brochure_09_spreads.pdf.
- [63] Heinz W. Engl, Martin Hanke-Bourgeois, and Andreas Neubauer. *Regularization of inverse problems*, volume 375 of *Mathematics and its applications <Dordrecht>*. Kluwer Acad. Publ, Dordrecht, 1996. ISBN 0-7923-4157-0. URL <http://www.loc.gov/catdir/enhancements/fy0822/96028672-d.html>.
- [64] ETIPWind Executive Committee, Alexander Vandenberghe, and Pierre Tardieu. Etipwind roadmap, 2020. URL <https://etipwind.eu/files/reports/ETIPWind-roadmap-2020.pdf>.
- [65] European Technology & Innovation Platform on Wind Energy. Strategic research and innovation agenda 2018, 2018. URL <https://etipwind.eu/wp-content/uploads/2018-Strategic-Research-Innovation-Agenda.pdf>.
- [66] Muhammad Fahim, Vishal Sharma, Tuan-Vu Cao, Berk Canberk, and Trung Q. Duong. Machine learning-based digital twin for predictive modeling in wind turbines. *IEEE Access*, 10:14184–14194, 2022.
- [67] Shu-Kai S. Fan, Yun-Chia Liang, and Erwie Zahara. A genetic algorithm and a particle swarm optimizer hybridized with nelder–mead simplex search. *Computers & Industrial Engineering*, 50(4):401–425, 2006.
- [68] S. Faulstich, B. Hahn, and P. J. Tavner. Wind turbine downtime and its importance for offshore deployment. *Wind Energy*, 14(3):327–337, 2011.
- [69] V. Fedorov and C. Berggreen. Bend-twist coupling potential of wind turbine blades. *Journal of Physics: Conference Series*, 524:012035, 2014.
- [70] Roland Feil, Tobias Pflumm, Pietro Bortolotti, and Marco Morandini. A cross-sectional aeroelastic analysis and structural optimization tool for slender composite structures. *Composite Structures*, 253: 112755, 2020.
- [71] Zhouquan Feng, Yang Lin, Wenzan Wang, Xugang Hua, and Zhengqing Chen. Probabilistic updating of structural models for damage assessment using approximate bayesian computation. *Sensors (Basel, Switzerland)*, 20(11), 2020.
- [72] Victor Fung, Jiaxin Zhang, Guoxiang Hu, P. Ganesh, and Bobby G. Sumpter. Inverse design of two-dimensional materials with invertible neural networks. *npj Computational Materials*, 7(1), 2021.
- [73] S. Shan G. Gary Wang. Review of metamodeling techniques in support of engineering design optimization. 2012.
- [74] Mari Cruz Garcia, Miguel A. Sanz-Bobi, and Javier Del Pico. Simap: Intelligent system for predictive maintenance. *Computers in Industry*, 57(6):552–568, 2006.
- [75] Garrad Hassan. Bladed: Theory manual version 4.6. *DNV-GL*, 2014.

- [76] General Directorate for Research and Innovation (European Commission) and Joint Research Centre (European Commission). The strategic energy technology (set) plan. *Publications Office*, 2019.
- [77] Farid Ghahari, Niloofar Malekghaini, Hamed Ebrahimiyan, and Ertugrul Taciroglu. Bridge digital twinning using an output-only bayesian model updating method and recorded seismic measurements. *Sensors (Basel, Switzerland)*, 22(3), 2022.
- [78] Sayan Ghosh, Govinda A. Padmanabha, Cheng Peng, Steven Atkinson, Valeria Andreoli, Piyush Pandita, Thomas Vandeputte, Nicholas Zabaras, Liping Wang, and Genghis Khan. Pro-ml ideas: A probabilistic framework for explicit inverse design using invertible neural network. In *ATAA Scitech 2021 Forum*. American Institute of Aeronautics and Astronautics, 2021. ISBN 978-1-62410-609-5.
- [79] V. Giavotto, M. Borri, P. Mantegazza, G. Ghiringhelli, V. Carmaschi, G. C. Maffioli, and F. Mussi. Anisotropic beam theory and applications. *Computers & Structures*, 16(1-4):403–413, 1983.
- [80] Hwan Goh, Sheroze Sheriffdeen, Jonathan Wittmer, and Tan Bui-Thanh. Solving bayesian inverse problems via variational autoencoders. 2019.
- [81] Peter Greaves and David Langston. Torsional testing of wind turbine blades: [conference presentation], 2021.
- [82] Michael W. Grieves. Virtually intelligent product systems: Digital and physical twins. In Shannon Flumerfelt, Katherine G. Schwartz, Dimitri Mavris, and Simon Briceno, editors, *Complex Systems Engineering: Theory and Practice*, volume 411, pages 175–200. American Institute of Aeronautics and Astronautics, Inc, Reston, VA, 2019. ISBN 978-1-62410-564-7.
- [83] D. Todd Griffith. Structural dynamics analysis and model validation of wind turbine structures. In *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, 2009. ISBN 978-1-60086-975-4.
- [84] D. Todd Griffith, Thomas G. Carne, and Joshua A. Paquette. Modal testing for validation of blade models. *Wind Engineering*, 32(2):91–102, 2008.
- [85] Janek Gröhl, Melanie Schellenberg, Kris Dreher, and Lena Maier-Hein. Deep learning for biomedical photoacoustic imaging: A review. *Photoacoustics*, 22:100241, 2021.
- [86] Dietmar Gross, Werner Hauger, Jörg Schröder, and Wolfgang A. Wall. *Technische Mechanik 2: Elastostatik*. Springer-Lehrbuch. Springer, Berlin, 11., bearb. Aufl. edition, 2012. ISBN 978-3-642-19983-7.
- [87] Nyoman Gunantara. A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, 5(1):1502242, 2018.
- [88] J. Gundlach and Y. Govers. Experimental modal analysis of aeroelastic tailored rotor blades in different boundary conditions. *Journal of Physics: Conference Series*, 2019.
- [89] Kwangtae Ha, Moritz Bätge, David Melcher, and Steffen Czichon. Development and feasibility study of segment blade test methodology. *Wind Energy Science*, 5(2):591–599, 2020.
- [90] Jonas Haldemann, Victor Ksoll, Daniel Walter, Yann Alibert, Ralf S. Klessen, Willy Benz, Ullrich Koethe, Lynton Ardizzone, and Carsten Rother. Exoplanet characterization using conditional invertible neural networks, 2022.
- [91] Warren Hare, Julie Nutini, and Solomon Tesfamariam. A survey of non-gradient optimization methods in structural engineering. *Advances in Engineering Software*, 59:19–28, 2013.
- [92] Florian Hartig, Justin M. Calabrese, Björn Reineking, Thorsten Wiegand, and Andreas Huth. Statistical inference for stochastic simulation models—theory and application. *Ecology letters*, 14(8):816–827, 2011.
- [93] P. U. Haselbach, R. D. Bitsche, and K. Branner. The effect of delaminations on local buckling in wind turbine blades. *Renewable Energy*, 85:295–305, 2016.
- [94] Philipp U. Haselbach, Frederik Zahle, Peter Berring, Sergei Semenov, Laura Voltá, I. Roqueta, and David Robert Verelst. Blade research and demonstration platform. *Journal of Physics: Conference Series*, 1618(5):052073, 2020.
- [95] Werner Hauger, Volker Mannl, Wolfgang A. Wall, and Ewald Werner, editors. *Aufgaben zu Technische Mechanik 1-3*. Springer-Lehrbuch. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-21185-0.

- [96] Abdel-Rahman Hedar and Masao Fukushima. Heuristic pattern search and its hybridization with simulated annealing for nonlinear global optimization. *Optimization Methods and Software*, 19(3-4): 291–308, 2004.
- [97] Dewey H. Hodges. A mixed variational formulation based on exact intrinsic equations for dynamics of moving beams. *International Journal of Solids and Structures*, 26(11):1253–1273, 1990.
- [98] Dewey H. Hodges. Geometrically exact, intrinsic theory for dynamics of curved and twisted anisotropic beams. *AIAA Journal*, 41(6):1131–1137, 2003.
- [99] Dewey H. Hodges. *Nonlinear composite beam theory*, volume 213 of *Progress in astronautics and aeronautics*. American Institute of Aeronautics and Astronautics, Reston, Va., 2006. ISBN 1-56347-697-5.
- [100] Benedikt Hofmeister, Marlene Bruns, and Raimund Rolfes. Finite element model updating using deterministic optimisation: A global pattern search approach. *Engineering Structures*, 195:373–381, 2019.
- [101] Robert Hooke and T. A. Jeeves. “direct search” solution of numerical and statistical problems. *Journal of the ACM*, 8(2):212–229, 1961.
- [102] Jyh-Yih Hsu, Yi-Fu Wang, Kuan-Cheng Lin, Mu-Yen Chen, and Jenneille Hwai-Yuan Hsu. Wind turbine fault diagnosis and predictive maintenance through statistical process control and machine learning. *IEEE Access*, 8:23427–23439, 2020.
- [103] Jérôme Idier, editor. *Bayesian approach to inverse problems*. ISTE, London, 2008. ISBN 978-1-84821-032-5.
- [104] IEA (2013). Technology roadmap - wind energy 2013. Paris, 2013. URL <https://www.iea.org/reports/technology-roadmap-wind-energy-2013>.
- [105] IEA (2021). Global energy review 2021. Paris, 2021. URL <https://www.iea.org/reports/global-energy-review-2021>.
- [106] International Electrotechnical Commission. Iec 61400-23: Wind turbines - part 23: Full-scale structural testing of rotor blades. 2014.
- [107] John Jasa, Pietro Bortolotti, Daniel Zalkind, and Garrett Barter. Effectively using multifidelity optimization for wind turbine design. *Wind Energy Science*, 7(3):991–1006, 2022.
- [108] Ping Jiang, Qi Zhou, and Xinyu Shao. *Surrogate Model-Based Engineering Design and Optimization*. Springer Tracts in Mechanical Engineering Ser. Springer Singapore Pte. Limited, Singapore, 2020. ISBN 978-981-15-0730-4.
- [109] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-mw reference wind turbine for offshore system development, 2009.
- [110] Jason Jonkman. The new modularization framework for the fast wind turbine cae tool. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, 2013. ISBN 978-1-62410-181-6.
- [111] Fei Kang, Junjie Li, and Haojin Li. Artificial bee colony algorithm and pattern search hybridized for global optimization. *Applied Soft Computing*, 13(4):1781–1791, 2013.
- [112] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. A review on genetic algorithm: past, present, and future. *Multimedia tools and applications*, 80(5):8091–8126, 2021.
- [113] H. Haddad Khodaparast, Y. Govers, I. Dayyani, S. Adhikari, M. Link, M. I. Friswell, J. E. Mottershead, and J. Siens. Fuzzy finite element model updating of the dlr airmod test structure. *Applied Mathematical Modelling*, 52:512–526, 2017.
- [114] Yuka Kikuchi and Takeshi Ishihara. Availability and lcoe analysis considering failure rate and downtime for onshore wind turbines in japan. *Energies*, 14(12):3528, 2021.
- [115] Durk P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In Curran Associates, Inc., editor, *Advances in Neural Information Processing Systems*. 2018.
- [116] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science (New York, N. Y.)*, 220(4598):671–680, 1983.

- [117] Johannes Knebusch, Janto Gundlach, and Yves Govers. 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*, pages 2159–2174. EASDAthens, 2020.
- [118] Mykel J. Kochenderfer and Tim Allan Wheeler. *Algorithms for optimization*. The MIT Press, Cambridge, Massachusetts and London, England, 2019. ISBN 9780262039420.
- [119] Rohit Kshirsagar, Steve Jones, Jonathan Lawrence, and Jim Tabor. Optimization of tig welding parameters using a hybrid nelder mead-evolutionary algorithms method. *Journal of Manufacturing and Materials Processing*, 4(1):10, 2020.
- [120] Victor F. Ksoll, Lynton Ardizzone, Ralf Klessen, Ullrich Koethe, Elena Sabbi, Massimo Robberto, Dimitrios Gouliermis, Carsten Rother, Peter Zeidler, and Mario Gennaro. Stellar parameter determination from photometry using invertible neural networks. *Monthly Notices of the Royal Astronomical Society*, 499(4):5447–5485, 2020.
- [121] Jeffrey C. Lagarias, James A. Reeds, Margaret H. Wright, and Paul E. Wright. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1):112–147, 1998.
- [122] Daniel Laird, Felicia Montoya, and David Malcolm. Finite element modeling of wind turbine blades. page 354, 2005.
- [123] Torben Juul Larsen and Anders Melchior Hansen. How 2 hawc2, the user’s manual. Roskilde, 2007. ISBN 978-87-550-3583-6. URL http://iis-03.risoe.dk/netahtml/risoe/publ_uk.htm.
- [124] Bruce LeBlanc and Carlos Ferreira. Experimental characterization of h-vawt turbine for development of a digital twin. *Journal of Physics: Conference Series*, 1452(1):012057, 2020.
- [125] Kwang Y. Lee and Mohamed A. El-Sharkawi, editors. *Modern heuristic optimization techniques: Theory and applications to power systems*. IEEE Press series on power engineering. Wiley-Interscience, Hoboken, NJ, 2008. ISBN 9780471457114.
- [126] Douglas B. Lenat. The nature of heuristics. *Artificial Intelligence*, 19(2):189–249, 1982.
- [127] Robert Michael Lewis, Virginia Torczon, and Michael W. Trosset. Direct search methods: then and now. *Journal of Computational and Applied Mathematics*, 124(1-2):191–207, 2000.
- [128] Sven Leyffer. *Mixed Integer Nonlinear Programming*, volume v.154 of *The IMA Volumes in Mathematics and Its Applications Ser.* Springer New York, New York, NY, 2012. ISBN 1493902210. URL <https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=884336>.
- [129] Ming-Hua Lin, Jung-Fa Tsai, and Chian-Son Yu. A review of deterministic optimization methods in engineering and management. *Mathematical Problems in Engineering*, 2012:1–15, 2012.
- [130] C. Lindenburg. Stablad - stability analysis tool for anisotropic rotor blade panels, 2008.
- [131] David J. Malcolm and Daniel L. Laird. Extraction of equivalent beam properties from blade models. *Wind Energy*, 10(2):135–157, 2007.
- [132] Simone Manzato, Bart Peeters, Alessandro Toso, Herman van der Auweraer, and Richard Osgood. Model updating methodologies for multibody simulation models: application to a full-scale wind turbine model. In Tom Proulx, editor, *Linking Models and Experiments, Volume 2*, Conference Proceedings of the Society for Experimental Mechanics Series, pages 349–358. Springer New York, New York, NY, 2011. ISBN 978-1-4419-9304-5.
- [133] Paul Marjoram, John Molitor, Vincent Plagnol, and Simon Tavaré. Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, 100(26):15324–15328, 2003.
- [134] R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.
- [135] Tshilidzi Marwala. *Finite-element-model updating using computational intelligence techniques: Applications to structural dynamics*. Springer, London, 2010. ISBN 9781849963220.
- [136] Tshilidzi Marwala, Ilyes Boulkaibet, and Sondipon Adhikari. *Probabilistic Finite Element Model Updating Using Bayesian Statistics*. John Wiley & Sons, Ltd, Chichester, UK, 2016. ISBN 9781119153023.

- [137] Pierangelo Masarati, Marco Morandini, and Paolo Mantegazza. An efficient formulation for general-purpose multibody/multiphysics analysis. *Journal of Computational and Nonlinear Dynamics*, 9(4), 2014.
- [138] Zbigniew Michalewicz. Heuristic methods for evolutionary computation techniques. *Journal of Heuristics*, 1(2):177–206, 1996.
- [139] Leon Mishnaevsky. Root causes and mechanisms of failure of wind turbine blades: Overview. *Materials*, 15(9):2959, 2022.
- [140] Farid K. Moghadam, Geraldo F. de S. Rebouças, and Amir R. Nejad. Digital twin modeling for predictive maintenance of gearboxes in floating offshore wind turbine drivetrains. *Forschung im Ingenieurwesen*, 85(2):273–286, 2021.
- [141] Zhuzhell Montano, Michael Kühn, Elia Daniele, and Jan Stüve. Bend-twist coupling on rotor blades for wind turbines. *Lightweight Design worldwide*, 11(4):42–47, 2018.
- [142] J. E. Mottershead and M. I. Friswell. Model updating in structural dynamics: A survey. *Journal of Sound and Vibration*, 167(2):347–375, 1993.
- [143] Mansureh-Sadat Nabiyan, Faramarz Khoshnoudian, Babak Moaveni, and Hamed Ebrahimiyan. Mechanics-based model updating for identification and virtual sensing of an offshore wind turbine using sparse measurements. *Structural Control and Health Monitoring*, 28(2), 2021.
- [144] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.
- [145] Pablo Noever-Castelos and Claudio Balzani. The impact of geometric non-linearities on the fatigue analysis of trailing edge bond lines in wind turbine rotor blades. *Journal of Physics: Conference Series*, 749:012009, 2016.
- [146] Pablo Noever-Castelos and Claudio Balzani. On the impact of multi-axial stress states on trailing edge bondlines in wind turbine rotor blades. *Journal of Physics: Conference Series*, 753:062002, 2016.
- [147] Pablo Noever-Castelos, Lynton Ardizzone, and Claudio Balzani. Model updating of wind turbine blade cross sections with invertible neural networks. *Wind Energy*, 25(3):573–599, 2022.
- [148] Pablo Noever-Castelos, Bernd Haller, and Claudio Balzani. Validation of a modeling methodology for wind turbine rotor blades based on a full-scale blade test. *Wind Energy Science*, 7(1):105–127, 2022.
- [149] Pablo Noever-Castelos, David Melcher, and Claudio Balzani. Model updating of a wind turbine blade finite element timoshenko beam model with invertible neural networks. *Wind Energy Science*, 7(2):623–645, 2022.
- [150] Andreas Öchsner. *Foundations of classical laminate theory*, volume volume 163 of *Advanced structured materials*. Springer, Cham, 2021. ISBN 3030826309.
- [151] L.C.T. Overgaard and E. Lund. Structural collapse of a wind turbine blade. part b: Progressive interlaminar failure models. *Composites Part A: Applied Science and Manufacturing*, 41(2):271–283, 2010.
- [152] L.C.T. Overgaard, E. Lund, and O. T. Thomsen. Structural collapse of a wind turbine blade. part a: Static test and equivalent single layered models. *Composites Part A: Applied Science and Manufacturing*, 41(2):257–270, 2010.
- [153] Samet Ozturk, Vasilis Fthenakis, and Stefan Faulstich. Failure modes, effects and criticality analysis for wind turbines considering climatic regions and comparing geared and direct drive wind turbines. *Energies*, 11(9), 2018.
- [154] Daniel R. Pardo and Kim Branner. Finite element analysis of the cross-section of wind turbine blades; a comparison between shell and 2d-solid models. *Wind Engineering*, 29(1):25–31, 2005.
- [155] Miroslav Pastor, Michal Binda, and Tomáš Harčarik. Modal assurance criterion. *Procedia Engineering*, 48:543–548, 2012.
- [156] Mathijs Peeters, Gilberto Santo, Joris Degroote, and Wim van Paeppegem. High-fidelity finite element models of composite wind turbine blades with shell and solid elements. *Composite Structures*, 200:521–531, 2018.

- [157] Mathijs Peeters, Gilberto Santo, Joris Degroote, and Wim van Paeppegem. Comparison of shell and solid finite element models for the static certification tests of a 43 m wind turbine blade. *Energies*, 11(6):1346, 2018.
- [158] T. P. Philippidis, A. P. Vassilopoulos, K. G. Katopis, and S. G. Voutsinas. Thin/probeam : A software for fatigue design and analysis of composite rotor blades. *Wind Engineering*, 20(5):349–362, 1996. URL <http://www.jstor.org/stable/43749625>.
- [159] F. Pimenta, J. Pacheco, C. M. Branco, C. M. Teixeira, and F. Magalhães. Development of a digital twin of an onshore wind turbine using monitoring data. *Journal of Physics: Conference Series*, 1618(2):022065, 2020.
- [160] Ronald L. Rardin and Reha Uzsoy. Experimental evaluation of heuristic optimization algorithms: A tutorial. *Journal of Heuristics*, 7(3):261–304, 2001.
- [161] J. N. Reddy. *Theory and Analysis of Elastic Plates and Shells*. CRC Press, 2006. ISBN 9780429127601.
- [162] M. D. Reder, E. Gonzalez, and J. J. Melero. Wind turbine failures - tackling current problems in failure data analysis. *Journal of Physics: Conference Series*, 753, 2016.
- [163] Brian R. Resor and Joshua Paquette. A numad model of the sandia tx-100 blade. 2012.
- [164] Jennifer M. Rinker, Morten H. Hansen, and Torben J. Larsen. Calibrating a wind turbine model using diverse datasets. *Journal of Physics: Conference Series*, 1037:062026, 2018.
- [165] Peter Rohl, Paul Dorman, Mark Sutton, Devesh Kumar, and Carlos Cesnik. A multidisciplinary design environment for composite rotor blades. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference & 20th AIAA/ASME/AHS Adaptive Structures Conference & 14th AIAA*. American Institute of Aeronautics and Astronautics, 2012. ISBN 978-1-60086-937-2.
- [166] M. Rosemeier, A. Antoniou, X. Chen, F. Lahuerta, P. Berring, and K. Branner. Trailing edge subcomponent testing for wind turbine blades—part a: Comparison of concepts. *Wind Energy*, 22(4):487–498, 2019.
- [167] Malo Rosemeier. Feproc blade model verification - 3d shell and beam model. 2018.
- [168] Mojtaba Sadegh and Jasper A. Vrugt. Approximate bayesian computation using markov chain monte carlo simulation: Dream (abc). *Water Resources Research*, 50(8):6767–6787, 2014.
- [169] S. Salcedo-Sanz. Modern meta-heuristics based on nonlinear physics processes: A review of models and design procedures. *Physics Reports*, 655:1–70, 2016.
- [170] Ruhul A. Sarker and Charles Sinclair Newton. *Optimization modelling: A practical approach*. CRC Press/Taylor & Francis, Boca Raton, Fla., 2007. ISBN 978-1-4200-4310-5.
- [171] F. Sayer, A. Antoniou, S. Goutianos, I. Gebauer, K. Branner, and C. Balzani. Reliablade project: A material’s perspective towards the digitalization of wind turbine rotor blades. *IOP Conference Series: Materials Science and Engineering*, 942:012006, 2020.
- [172] Karsten Schröder, Cristian G. Gebhardt, and Raimund Rolfes. Damage localization at wind turbine support structures using sequential quadratic programming for model updating. In *8th European Workshop on Structural Health Monitoring*, 2016.
- [173] Karsten Schröder, Saskia Grove, Stavroula Tsiapoki, Cristian G. Gebhardt, and Raimund Rolfes. Structural change identification at a wind turbine blade using model updating. *Journal of Physics: Conference Series*, 1104:012030, 2018.
- [174] Shankar Sehgal and Harmesh Kumar. Structural dynamic model updating techniques: A state of the art review. *Archives of Computational Methods in Engineering*, 23(3):515–533, 2016.
- [175] Marc Seidel, Sven Voormeeren, and Jan-Bart van der Steen. State-of-the-art design processes for offshore wind turbine support structures. *Stahlbau*, 85(9):583–590, 2016.
- [176] Faisal Shabbir and Piotr Omenzetter. Particle swarm optimization with sequential niche technique for dynamic finite element model updating. *Computer-Aided Civil and Infrastructure Engineering*, 30(5):359–375, 2015.

- [177] Viraj Shah and Chinmay Hegde. Solving linear inverse problems using gan priors: An algorithm with provable guarantees. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4609–4613. IEEE, 2018. ISBN 978-1-5386-4658-8.
- [178] Ellen Simoen, Guido de Roeck, and Geert Lombaert. Dealing with uncertainty in model updating for damage assessment: A review. *Mechanical Systems and Signal Processing*, 56-57:123–149, 2015.
- [179] SmartBlades2. Fabrication, testing, and further development of smart rotor blades, coordinated research project (project numbers 0324032a-h), supported by the federal ministry for economic affairs and energy of germany due to a decision of the german bundestag. 2016-2020.
- [180] Helena Solman, Julia Kirch Kirkegaard, Mattijs Smits, Bas van Vliet, and Simon Bush. Digital twinning as an act of governance in the wind energy sector. *Environmental Science & Policy*, 127:272–279, 2022.
- [181] Michael A. Sprague, Jason M. Jonkman, and Bonnie J. Jonkman. Fast modular framework for wind turbine simulation: New algorithms and numerical examples. In *AIAA SciTech 2015*, 2015.
- [182] Christian Spura. *Einführung in die Balkentheorie nach Timoshenko und Euler-Bernoulli*. essentials. Springer Vieweg, Wiesbaden and Heidelberg, 2019. ISBN 9783658252151. URL <http://www.springer.com/>.
- [183] Alexander R. Stäblein, Morten H. Hansen, and David R. Verelst. Modal properties and stability of bend–twist coupled wind turbine blades. *Wind Energy Science*, 2(1):343–360, 2017.
- [184] Daniel J. Tait and Theodoros Damoulas. Variational autoencoding of pde inverse problems, 2020.
- [185] Mohammad Tamizifar, Masoud Mosayebi, and Saeid Ziaei-Rad. Optimal sensor placement and model updating applied to the operational modal analysis of a nonuniform wind turbine tower. *Mechanical Sciences*, 13(1):331–340, 2022.
- [186] Tina Toni. Abc smc for parameter estimation and model selection with applications in systems biology. *Nature Precedings*, 2011.
- [187] H. Tran-Ngoc, S. Khatir, G. de Roeck, T. Bui-Tien, L. Nguyen-Ngoc, and M. Abdel Wahab. Model updating for nam o bridge using particle swarm optimization algorithm and genetic algorithm. *Sensors (Basel, Switzerland)*, 18(12), 2018.
- [188] H. Tran-Ngoc, Leqia He, Edwin Reynders, S. Khatir, T. Le-Xuan, G. de Roeck, T. Bui-Tien, and M. Abdel Wahab. An efficient approach to model updating for a multispan railway bridge using orthogonal diagonalization combined with improved particle swarm optimization. *Journal of Sound and Vibration*, 476:115315, 2020.
- [189] Alan Turnbull, Conor McKinnon, James Carrol, and Alasdair McDonald. On the development of offshore wind turbine technology: An assessment of reliability rates and fault detection methods in a changing market. *Energies*, 15(9):3180, 2022.
- [190] Heather Turnbull and Piotr Omenzetter. Fuzzy finite element model updating of a laboratory wind turbine blade for structural modification detection. *Procedia Engineering*, 199:2274–2281, 2017.
- [191] Heather Turnbull and Piotr Omenzetter. Damage severity assessment in wind turbine blade laboratory model through fuzzy finite element model updating. In H. Felix Wu, Andrew L. Gyekenyesi, Peter J. Shull, and Tzu-Yang Yu, editors, *Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, and Civil Infrastructure 2017*, SPIE Proceedings, page 101692E. SPIE, 2017.
- [192] Ulf T. Tygesen, Michael S. Jepsen, Jonas Vestermark, Niels Dollerup, and Anne Pedersen. The true digital twin concept for fatigue re-assessment of marine structures. In *Volume 1: Offshore Technology*. American Society of Mechanical Engineers, 2018. ISBN 978-0-7918-5120-3.
- [193] United Nations. Paris agreement: (adopted 12 dec. 2015, entered into force 4 nov. 2016). *United Nations Treaty Collection*, (Chapter XXVII 7 d), 2015.
- [194] United Nations General Assembly. Resolution 70/1: Transforming our world: the 2030 agenda for sustainable development. (A/RES/70/1 (21 October 2015)), 2015. URL undocs.org/en/A/RES/70/1.
- [195] G. A. M. van Kuik, J. Peinke, R. Nijssen, D. Lekou, J. Mann, J. N. Sørensen, C. Ferreira, J. W. van Wingerden, D. Schlipf, P. Gebraad, H. Polinder, A. Abrahamsen, G. J. W. van Bussel, J. D. Sørensen, P. Tavner, C. L. Bottasso, M. Muskulus, D. Matha, H. J. Lindeboom, S. Degraer, O. Kramer, S. Lehnhoff, M. Sonnenschein, P. E. Sørensen, R. W. Künnike, P. E. Morthorst, and K. Skytte. Long-term research challenges in wind energy – a research agenda by the european academy of wind energy. *Wind Energy Science*, 1(1):1–39, 2016.

- [196] Peter J. M. van Laarhoven. *Simulated Annealing: Theory and Applications*, volume 37 of *Springer eBook Collection Mathematics and Statistics*. Springer, Dordrecht, 1987. ISBN 9789048184385.
- [197] Eric VanDerHorn and Sankaran Mahadevan. Digital twin: Generalization, characterization and implementation. *Decision Support Systems*, 145:113524, 2021.
- [198] Antonio Velazquez and R. Andrew Swartz. Operational model updating of low-order horizontal axis wind turbine models for structural health monitoring applications. *Journal of Intelligent Material Systems and Structures*, 26(13):1739–1752, 2015.
- [199] Christopher A. Walford. *Wind turbine reliability: Understanding and minimizing wind turbine operation and maintenance costs*. Albuquerque, New Mexico, 2006.
- [200] Jinjiang Wang, Yuanyuan Liang, Yinghao Zheng, Robert X. Gao, and Fengli Zhang. An integrated fault diagnosis and prognosis approach for predictive maintenance of wind turbine bearing with limited samples. *Renewable Energy*, 145:642–650, 2020.
- [201] Lin Wang, Xiongwei Liu, and Athanasios Kolios. State of the art in the aeroelasticity of wind turbine blades: Aeroelastic modelling. *Renewable and Sustainable Energy Reviews*, 64:195–210, 2016.
- [202] Meng Wang, Feiyan Yang, and Haipeng Liu. Fusion-cinn: A generative fusion method for multimodal medical images based on conditional invertible neural network. *SSRN Electronic Journal*, 2022.
- [203] Qi Wang, Michael A. Sprague, Jason Jonkman, Nick Johnson, and Bonnie Jonkman. Beamdyn: a high-fidelity wind turbine blade solver in the fast modular framework. *Wind Energy*, 20(8):1439–1462, 2017.
- [204] Yanfeng Wang, Ming Liang, and Jiawei Xiang. Damage detection method for wind turbine blades based on dynamics analysis and mode shape difference curvature information. *Mechanical Systems and Signal Processing*, 48(1-2):351–367, 2014.
- [205] Jonathan White, Douglas Adams, and Mark Rumsey. Updating of a wind turbine model for the evaluation of methods for operational monitoring using inertial measurements. In *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, 2010. ISBN 978-1-60086-959-4.
- [206] Christian Willberg, Rakesh Ravi, Johannes Rieke, and Falk Heinecke. Validation of a 20 m wind turbine blade model. *Energies*, 14(9):2451, 2021.
- [207] Zhiyuan Xia, Aiqun Li, Jianhui Li, Huiyuan Shi, Maojun Duan, and Guangpan Zhou. Model updating of an existing bridge with high-dimensional variables using modified particle swarm optimization and ambient excitation data. *Measurement*, 159:107754, 2020.
- [208] Xin Yao. *Evolutionary Computation: Theory and Applications*. WORLD SCIENTIFIC, 1999. ISBN 978-981-02-2306-9.
- [209] Hyeonsoo Yeo, Khiem-Van Truong, and Robert Ormiston. Assessment of 1-d versus 3-d methods for modeling rotor blade structural dynamics. In *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference & 18th AIAA/ASME/AHS Adaptive Structures Conference & 12th*. American Institute of Aeronautics and Astronautics, 2010. ISBN 978-1-60086-961-7.
- [210] Wenbin Yu. Efficient high-fidelity simulation of multibody systems with composite dimensionally reducible components. *Journal of the American Helicopter Society*, 52(1):49–57, 2007.
- [211] Wenbin Yu, Vitali V. Volovoi, Dewey H. Hodges, and Xianyu Hong. Validation of the variational asymptotic beam sectional analysis. *AIAA Journal*, 40(10):2105–2112, 2002.
- [212] Wenbin Yu, Dewey H. Hodges, and Jimmy C. Ho. Variational asymptotic beam sectional analysis – an updated version. *International Journal of Engineering Science*, 59:40–64, 2012.
- [213] Stelios H. Zanakakis and James R. Evans. Heuristic “optimization”: Why, when, and how to use it. *Interfaces*, 11(5):84–91, 1981.
- [214] D. S. Zarouchas, A. A. Makris, F. Sayer, D. van Hemelrijck, and A. M. van Wingerde. Investigations on the mechanical behavior of a wind rotor blade subcomponent. *Composites Part B: Engineering*, 43(2):647–654, 2012.

- [215] Xin Zhang and Andrew Curtis. Bayesian geophysical inversion using invertible neural networks. *Journal of Geophysical Research: Solid Earth*, 126(7), 2021.
- [216] Zhiming Zhang, Chao Sun, and Vahid Jahangiri. Structural damage identification of offshore wind turbines: A two-step strategy via fe model updating. *Structural Control and Health Monitoring*, 29(2), 2022.
- [217] P. Zhou and P. T. Yin. An opportunistic condition-based maintenance strategy for offshore wind farm based on predictive analytics. *Renewable and Sustainable Energy Reviews*, 109:1–9, 2019.

Dissertations of the Institute for Wind Energy Systems

01 (2023): Updating structural wind turbine blade models via invertible neural networks, Dissertation, Pablo Noever Castelos, 2023



Curriculum Vitae

Pablo Noever Castelos 

Born on 6th July 1987
in Krefeld, Germany

Awards

Victor Rizkallah Foundation

*Award for excellent achievements
in the master degree: Wind En-
ergy Engineering*

Scientific Staff

Aug
2022

Institute for Wind Energy Systems (IWES)
Leibniz University Hannover

Apr
2014

Involved Research Projects:
SmartBlades2, Reliablade

Tutorial Lecturer in:
Rotor Blade Design of Wind Turbines
Planning and Erection of Wind Farms

Apr
2016

Wind Energy Engineering

Apr
2014

Master of Science
Leibniz University Hannover

Mar
2014

Scientific Staff

Oct
2013

Fraunhofer Institute for Environmental,
Safety and Energy Technology, Oberhausen

Oct
2013

Mechanical Engineering

Oct
2011

Master of Science
Ruhr-University Hannover

Jul
2011

European Engineering Studies

Oct
2010

Bachelor of Engineering
Coventry University, United Kingdom

Jun
2011

Mechatronics

Sep
2007

Dual Bachelor of Engineering
University of Applied Science Bochum