



**HAL**  
open science

# Probabilistic approach for fatigue life estimation of wind turbine blades using deep neural network

Wilson Javier Veloz Parra

## ► To cite this version:

Wilson Javier Veloz Parra. Probabilistic approach for fatigue life estimation of wind turbine blades using deep neural network. Mechanical engineering [physics.class-ph]. Normandie Université, 2020. English. NNT : 2020NORMIR29 . tel-03272434

**HAL Id: tel-03272434**

**<https://theses.hal.science/tel-03272434>**

Submitted on 28 Jun 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Normandie Université

## THÈSE

Pour obtenir le diplôme de doctorat

Spécialité Génie mécanique

Préparée au sein de L'institut national des sciences appliquées de Rouen

### Titre de la thèse

Probabilistic approach for fatigue life estimation of wind turbine blades using deep neural network

Présentée et soutenue par  
Wilson Javier VELOZ PARRA

Thèse soutenue publiquement le 17/12/2020 devant le jury composé de		
M. Dimitri LEFEBVRE	Professeur des universités, Université du Havre, GMP	Président du jury
M. John Dalsgaard SØRENSEN	Professeur des universités, Aalborg University, Reliability and Risk Analysis Research Group	Rapporteur
M. Pierre-Alain BOUCARD	Professeur des universités, ENS Paris-Saclay, LMT	Rapporteur
M. Laurent GUILAUMAT	Professeur des universités, Arts et Métiers ParisTech, ENSAM	Examineur
M. Paul DEGLAIRE	Directeur de recherche, Siemens Gamesa Renewable Energy, TLS	Examineur
M. Younes AOUES	Maitre des conférences, NSA Rouen Normandie, LMN	Co-directeur de thèse
M. Didier LEMOSSE	Maitre des conférences HDR, NSA Rouen Normandie, LMN	Directeur de thèse

Thèse dirigée par Didier LEMOSSE (et co-encadrée par Younes AOUES), laboratoire LMN





PROBABILISTIC APPROACH FOR FATIGUE LIFE  
ESTIMATION OF WIND TURBINE BLADES  
USING DEEP NEURAL NETWORK

Wilson Javier VELOZ PARRA

A thesis submitted for the degree of Doctor of  
Philosophy

17 December 2020

# Acknowledgments

The submission of this dissertation is a requirement for obtaining the degree of Doctor of Philosophy. This study was carried out between October 2017 and September 2020 at the Laboratory of Mechanics of Normandy, INSA of Rouen Normandie, thanks to the CALIOPE scholarship program created by agreements between the Dominican Ministry of Higher Education, Science and Technology, the French Embassy, and INSA of Rouen Normandie. The financial support is gratefully appreciated.

This study's scope could not have been integrated into this thesis without my advisors' scientific supervision. I would like to thank my advisor Didier LEMOSSE, who has taught me the state of the art of mechanical theory and modelization techniques and how to find the best solutions to complicated problems. I also would like to thank my co-advisor, Younes AOUES, for his support and advice in probability and reliability theory. Their academic and personal help since my arrival to this country and continuous inspiration has allowed me to continue work in the most challenging moments.

Special thanks to Eduardo SOUZA DE CURSI for his dedication and work in creating the CALIOPE scholarship program. Despite not having much free time, he was always willing to discuss and give me new ideas to advance my research. I want to thank the GPM professors: Taleb LAKHDAR, Clement KELLER, Fabrice BARBE, and last but not least, Benoit VEILLE, for all the help offered in materials mechanics theory and also their welcome since my arrival at the university.

Thanks to all my lab colleagues and friends, especially Hao BAI and Hongbo ZHANG. All the discussions and ideas shared inside and outside the laboratory helped me think more creatively and rigorously and to learn more about their fantastic culture. Thanks to the INSA staff for helping me, as a doctoral student, and as a professor's assistant, specially to Elisabeth LESSAGE. Thanks to my colleagues of the CALIOPE program.

I want to deeply thank my wife "Puchi" for all the support she has given me during these three years. Her love and patience have given me support in the most difficult moments where the results were not obtained. I hope to return the days I spend in front of the computer working and not at home with her. Thanks to my parents and siblings for their support, although we are in different countries, we are still family.

Finally, thanks to Jehovah, who is always present at all times, and his grace never left my home.



# Abstract

In the last few decades, wind turbines have evolved, increasing their size, being more cost-effective, and more Eco-friendly. As a consequence of this massive growth, design-engineers encounter new challenges to continue the cost-effectiveness of turbines for all applications. This dissertation proposes an original approach for the prediction of the fatigue lifetime of a composite wind turbine blade considering wind speed uncertainty. The proposed probabilistic approach is based on Monte Carlo simulations and a deep neural network to surrogate the multi-physical simulation model and the multi-axial fatigue analysis of the blade. In order to achieve this goal, several challenges have been taken on, including development of a regression approach to model jump discontinuities, development of a methodology to calculate the global sensitivity analysis of high dimensional problems, and development of an approach to estimate the probabilistic fatigue lifetime by using a detailed fatigue damage analysis of the composite wind turbine blades.

The regression model for jump discontinuities is developed by using the Gaussian mixture model and a deep neural network. These discontinuities are found in the response of 10 minutes of aero-elastic simulations in steady and unsteady environmental conditions. The proposed method is compared with other regression machine learning methods, and shows more efficiency and presents a lower mean squared error.

To estimate the global sensitivity of the called load application methods used to transfer the aero-elastic wind loads from a 1D beam to a 2D shell blade model, an original approach is developed to calculate the Sobol indices based on a deep neural network surrogate model for high dimensional problems. Two networks are developed, one to estimate the maximum aero-elastic loads considering a 10 minutes mean wind speed, and the second network to estimate the stresses at the blade root for given composite material thicknesses and the maximum aero-elastic loads predicted from the first network.

Using the global sensitivity results, the load application method with the lower sensitivity parameters is used to carry out the fatigue life prediction considering wind speed uncertainty. A deep neural network surrogate model is trained to estimate the fatigue damage for a 10 minutes mean wind speed of unsteady aero-elastic simulations, employed to calculate the 10 minutes stress history of the 2D shell blade model and then the fatigue damage considering the non-proportionality of the stresses.

The probabilistic fatigue life prediction is proposed by combining a deep neural network surrogate model and Monte Carlo simulations for wind speed between cut-in and cut-out. Finally, the probability density function of the fatigue lifetime is estimated at the maximum damaged point composite layer of the blade.

**Keywords :** Fatigue damage of composite wind turbine blade, global sensitivity analysis, fatigue lifetime prediction, wind speed uncertainty, deep neural network application, load application method for blade 2D shell models.



# Résumé

Ces dernières années, les éoliennes ont beaucoup évolué, leurs tailles ont beaucoup augmenté afin de les rendre plus rentables et plus écologiques. En raison de cette croissance massive, les ingénieurs concepteurs sont confrontés à de nouveaux défis pour maintenir la rentabilité des éoliennes, qu'elles soient terrestres ou maritimes. Ce travail de thèse propose le développement d'une approche probabiliste de la durée de vie en fatigue d'une pale d'éolienne en matériaux composites. Cette approche est basée sur l'apprentissage par un réseau de neurones profond et les simulations de Monte Carlo en considérant l'incertitude de la vitesse du vent. Afin d'atteindre cet objectif, plusieurs défis ont été relevés. Il s'agit des développements suivantes : a) une approche de régression pour la prise en compte des discontinuités de saut ; b) une méthodologie pour l'analyse de sensibilité globale des problèmes de haute dimension ; c) une méthodologie pour estimer la densité de probabilité de la durée de vie en fatigue de la pale d'une éolienne en matériaux composites.

Le modèle de régression développé pour prendre en compte les discontinuités de saut, utilise le modèle de mélange gaussien (*Gaussian Mixture*) et les réseaux de neurones profonds (*Deep Neural Network*). Ces discontinuités sont trouvées dans la réponse d'une simulation aéro-élastique de 10 minutes en tenant compte d'un écoulement laminaire ou turbulent du vent. La méthode proposée présente une erreur quadratique moyenne inférieure par rapport à d'autres méthodes de régression et d'apprentissage automatique utilisées.

Une méthode originale pour l'analyse des sensibilités globales pour les problèmes de grande dimension est développée pour estimer la sensibilité des méthodes d'application de charge (*LAM: Load Application Methods*). Ces méthodes sont utilisées pour transférer les charges sur la pale, issues des simulations aéro-élastiques 1D au modèle éléments finis en coque 2D. L'approche développée vise à calculer les indices de Sobol en utilisant un modèle de substitution basé sur un réseau de neurones profond. Cette méthodologie est basée sur le développement de deux réseaux de neurones, le premier estime les charges aéro-élastiques maximales pour une vitesse moyenne du vent sur 10 minutes, et le second consiste à estimer les contraintes mécaniques au pied de la pale en tenant compte des épaisseurs des matériaux composites et des charges aéro-élastiques maximales obtenues à partir du premier réseau de neurones. La durée de vie en fatigue est estimée en utilisant la méthode d'application de la charge qui présente la sensibilité la plus faible et l'analyse de fatigue multiaxiale. En outre, 10 minutes de simulations aéro-élastiques en considérant un écoulement turbulent du vent pour estimer l'historique des contraintes en utilisant un modèle par élément fini de coque, puis l'endommagement par fatigue est calculé en tenant compte de la non proportionnalité des contraintes. Un modèle de substitution basé sur un réseau de neurones profond est développé pour estimer le dommage par fatigue à 10 minutes et puis à 20 ans. La densité de probabilité de la durée de vie en fatigue au point d'endommagement maximal est obtenue en utilisant le modèle de substitution basé sur un réseau de neurones profond et des simulations de Monte Carlo.

**Mots clés:** Dommage par fatigue de la pale d'éolienne en matériaux composites, analyse

de sensibilité globale, estimation de la durée de vie en fatigue, application de réseaux de neurones profonds, incertitude de la vitesse du vent, méthode d'application de charge pour les modèles de coque 2D de pale.

# Résumé étendu

## Introduction générale

Au cours des dernières décennies, les éoliennes ont beaucoup évolué, leurs tailles ont beaucoup augmenté afin de les rendre plus rentables et plus écologiques. Actuellement, cette technologie de production d'énergie est devenue une tendance dans les politiques énergétiques à travers le monde, pour réduire l'utilisation des centrales électriques à énergies fossiles et ainsi diminuer l'effet de serre.

Conséquence de cette croissance massive, les ingénieurs concepteurs sont confrontés à de nouveaux défis pour maintenir la rentabilité des éoliennes pour tous les sites, sur des sites à faible vitesse de vent ou des fermes offshores, produisant de l'énergie en grande quantité. De plus, il y a un manque d'expérience opérationnelle car ces éoliennes sont récentes. Par conséquent, les concepteurs doivent s'appuyer sur des outils de simulation pour modéliser des comportements complexes et prendre des décisions en fonction de ceux-ci, ce qui rend ces tâches d'une importance capitale.

Les outils de simulation tels que FAST, développé par le NREL (National Renewable Energy Laboratory) [Jonkman and Buhl Jr, 2005] et HAWC2 développé par DTU (Denmark Technological University) [Larsen and Hansen, 2007] reproduisent la réponse dynamique des systèmes éoliens. Ces outils couplent les modèles de l'aéro-élasticité, de l'électricité, du contrôle et de l'hydrodynamique à l'aide de la théorie simplifiée de la dynamique des éléments de poutre et la théorie BEM (*Blade Element Momentum theory*). Ces outils de simulation permettent de calculer la réponse dynamique temporelle non-linéaire des déplacements et des chargements agissant sur l'éolienne dans les différentes conditions environnementales décrites dans les normes de certification des éoliennes [Guideline and Lloyd, 2010]. En outre, une analyse générale des pales d'éoliennes pourrait être effectuée en utilisant ces outils de simulation (i.e. FAST), pour obtenir les déplacements, les modes et les fréquences de vibration et l'analyse des charges extrêmes. Cependant, pour faire une analyse de contraintes ou de fatigue, il est nécessaire d'utiliser un modèle aux éléments finis et des charges réparties agissant le long de la pale. Le code CFD (*Computational Fluid Dynamics*) de FAST calcule l'interaction fluide-structure dans un espace 2D ou 3D. Toutefois, ce code n'a pas intégré le calcul de la réponse dynamique couplée de la pale modélisée en 2D. De plus, en fonction des conditions de maillage, le temps de calcul CPU nécessaire pour les simulations de type CFD est prohibitif en comparaison aux simulations aéro-élastiques de la pale modélisée en 1D. Pour cela, l'approche utilisée pour considérer la réponse couplée de la pale d'éolienne est basée sur l'utilisation des charges estimées par la simulation aéro-élastiques de FAST ; de les transférer en un champ de chargement réparti équivalent au modèle d'éléments finis (*FEM: Finite Element Method*) de coque 2D de la pale. Cette modélisation aux éléments finis permet d'obtenir une répartition des contraintes tout au long de la pale.

Différents auteurs [Bottasso et al., 2014, Hu et al., 2016b] définissent leur méthode pour transférer les charges issues du modèle 1D au modèle 2D, pour calculer la durée de vie en fatigue ou faire une optimisation de conception basée sur la fiabilité (*RBDO: Reliability-Based Design Optimization*), la distribution des contraintes et les réponses des déplacements sur la pale varient d'une méthode à l'autre. Ainsi, le choix de la méthode pour estimer la durée de vie en fatigue ou effectuer une analyse de fiabilité de la pale d'éolienne est crucial. En effet, en fonction de la façon dont est répliquée le chargement de la simulation dynamique couplée sur une coque 2D FEM de la pale, il y a le risque d'introduire des singularités qui peuvent introduire des erreurs dans la réponse du modèle numérique. Les pales modélisées en utilisant la méthode d'éléments finis (FEM) 2D ou 3D, permettent d'obtenir les répartitions des contraintes et des déplacements sur toute la pale. Les auteurs [Morató et al., 2019, Teixeira et al., 2017] ont utilisé le modèle de substitution de Krigeage (Processus Gaussien) pour remplacer le modèle de simulation qui nécessite un temps de calcul considérable. En effet, l'analyse de fiabilité ou les approches de propagation des incertitudes dans la durée de vie nécessitent l'utilisation répétitive du modèle aux éléments finis. Cette procédure est impraticable car elle exige un temps de calcul prohibitif. A cet égard, les modèles de substitution peuvent être utilisés pour remplacer le modèle de simulation (MEF, analyse de fatigue, etc.) car ils peuvent prédire précisément la réponse de la structure après un apprentissage adéquat. D'ailleurs, cette prédiction de la réponse est donnée en un temps de calcul réduit par rapport au modèle de simulation initial. Actuellement, avec le développement croissant des nouvelles méthodes d'apprentissage automatique et d'apprentissage profond, différentes stratégies et méthodologies peuvent être explorées pour la substitution des simulateurs multi-physiques et l'obtention d'une solution précise en moins de temps possible.

L'analyse de fatigue des pales d'éoliennes est un vaste domaine, nécessitant une approche multidisciplinaire en raison de la combinaison de différents composants avec des fonctions mécaniques et aérodynamiques. Par conséquent, avec un projet à durée limitée, la présente thèse se concentrera sur l'étude et l'estimation probabiliste de la durée de vie en fatigue des pales d'éoliennes en matériaux composites à l'aide d'un modèle de substitution basé sur un réseau de neurones profond.

## Objectifs

Le premier objectif de cette étude est de proposer une méthodologie d'estimation probabiliste de la durée de vie d'une pale d'éolienne soumise à de l'endommagement par fatigue. Cette analyse comprend : la simulation de 10 minutes du champ de vent à une vitesse moyenne, l'analyse aérodynamique, l'analyse élastique et électrique couplée au contrôle de la pale, le filtrage des charges aéro-élastiques, l'analyse des contraintes de la pale à l'aide d'un modèle aux éléments finis et l'évaluation du dommage par fatigue en tenant compte d'un état de contraintes multiaxial et non-proportionnel. Cependant, le temps de calcul nécessaire pour l'estimation de cette durée de vie en fatigue est très élevé. En outre, l'estimation de la densité de probabilité de la durée de vie en considérant l'incertitude de la vitesse du vent, en utilisant le modèle de simulation original est impraticable. Pour cela, un modèle de substitution est

proposé pour remplacer le modèle de simulation original, basé sur un réseau de neurones profond. En utilisant la méthodologie proposée, un dommage par fatigue sur 10 minutes est déterminé par une simulation aéro-élastique de 10 minutes à une vitesse moyenne donnée et un modèle de turbulence normal pour les conditions environnementales. Différentes méthodes d'application des charges pour le modèle aux éléments finis de coque 2D sont étudiées et comparées dans une analyse des contraintes de la pale.

Le deuxième objectif vise à proposer une méthodologie d'analyse de sensibilité globale pour les modèles de grande dimension. Nous proposons une approche basée sur deux modèles de substitution de réseaux de neurones profonds. Cette méthode de sensibilité permet de sélectionner la méthode d'application des charges avec les paramètres qui génèrent le moins de sensibilité. Trois méthodes d'application de charges sont utilisées pour comparer leurs réponses. Chacune d'entre elles a une approche différente pour appliquer les charges aéro-élastiques au modèle de coque 2D, d'une distribution de pression simple à une distribution réaliste sur toute la pale. Cette étude propose d'estimer les indices de sensibilité de Sobol en utilisant un modèle de substitution à deux étapes, un modèle de réseau de neurones pour estimer les charges aéro-élastiques maximales dans des conditions environnementales laminaires et un autre pour prédire les contraintes de Von Mises au pied de la pale.

Le troisième objectif est de proposer une méthode de régression basée sur les réseaux de neurones profonds pour la prise en compte des discontinuités du modèle initial. Cette méthode permet l'apprentissage et la prédiction de la réponse des modèles possédant des discontinuités dans leur domaine d'étude. La méthodologie proposée présente une erreur quadratique moyenne inférieure par rapport aux autres modèles de substitution utilisés.

## Structure de la thèse

Cette thèse est structurée comme suit:

- Le but, les travaux précédents et les objectifs de la thèse sont résumés dans l'introduction.
- Le chapitre 2 présente les simulations aéro-élastiques couplées utilisées pour une pale d'éolienne en matériaux composites. En outre, nous présentons les différentes spécifications de distribution des matériaux composites à travers la pale, le calcul des matrices de rigidité des sections transversales, les modes et les fréquences propres de vibration. De plus, il présente un état de l'art des méthodes d'application des charges utilisées pour transférer les charges des outils de simulation 1D vers un modèle d'éléments finis à coque 2D de la pale.
- Le chapitre 3 présente brièvement les réseaux de neurones profonds (*DNN: Deep Neural Network*) et les méthodes d'optimisation de ses hyperparamètres. Le DNN est le principal modèle de substitution de cette thèse pour l'analyse des sensibilités et de fatigue. Cette étude développe une nouvelle méthodologie pour traiter les discontinuités de saut en utilisant les réseaux de neurones, par le classement des zones avec une

réponse continue à l'aide d'un modèle de mélange gaussien (*Gaussian Mixture Model*) et l'entraînement du réseau en ajoutant la valeur de la classification en tant que paramètre d'entrée.

- Le chapitre 4 propose une nouvelle méthodologie pour l'analyse de sensibilité globale des méthodes d'application des charges utilisées dans le modèle aux éléments finis de coque 2D de la pale. Cette étude propose deux modèles de substitution, le premier vise à prédire les charges aéro-élastiques maximales simulées dans des conditions environnementales laminaires, le second DNN remplace le modèle aux éléments finis de la pale pour prédire les contraintes de Von Mises à partir des charges simulées précédemment et les épaisseurs des matériaux comme paramètres d'entrée. Les charges aéro-élastiques sont appliquées à l'aide de trois méthodes d'application des charges dans le modèle de coque 2D pour calculer la contrainte de Von Mises sur le pied de pale. Le calcul des indices de Sobol permet de donner la sensibilité des paramètres d'entrée de chaque méthode d'application des charges. Cette étude compare les sensibilités des paramètres d'entrée de différentes méthode d'application de charge pour déterminer celle qui introduit la moins de sensible au problème numérique. Une analyse de Morris utilisant directement le modèle aux éléments finis de coque 2D nous a permis de valider les résultats des indices de Sobol obtenus à l'aide du réseau de neurones profond.
- Le chapitre 5 propose une méthodologie pour estimer le dommage par fatigue à partir d'une simulation sur 10 minutes de la vitesse moyenne du vent. Une simulation de 10 minutes utilisant un modèle de turbulence normal est effectuée en couplant le comportement aérodynamique, élastique, électrique et de commande (servo) de l'éolienne pour calculer les charges résultantes du modèle 1D. Les charges sont transférées au modèle aux éléments finis de coque 2D en utilisant les méthodes d'application des charges afin d'estimer l'historique des contraintes. Un critère de fatigue multiaxial est utilisé pour l'estimation du dommage par fatigue afin de prendre en compte la non-proportionnalité de l'état de contraintes. Les diagrammes de durée de vie constante estiment le nombre de cycles jusqu'à la rupture à différentes amplitudes de contraintes et de contraintes moyennes. La règle de Miner permet de cumuler les dommages par fatigue. Un modèle de substitution basé sur un réseau de neurones profond est utilisé pour prédire les cycles de dommages de fatigue par vitesse du vent en ajoutant la fonction de distribution cumulative empirique de chaque distribution de dommage comme paramètres d'entrée. Les hyperparamètres du réseau sont optimisés à l'aide d'un algorithme d'optimisation basé sur un processus Gaussien. Le réseau de neurones profond est validé et testé pour prédire une nouvelle distribution des dommages en utilisant un historique temporel de champ de vent différent. Ensuite, l'analyse de la distribution d'endommagement de la pale est effectuée au point où le dommage est maximal. Une estimation du dommage par fatigue pour 20 ans est effectuée pour différentes couches des matériaux composites au point sélectionné. Les simulations de Monte Carlo sont effectuées en utilisant le modèle de substitution (DNN) pour estimer la distribution de probabilité de la durée de vie en fatigue.
- Le chapitre 6 résume les conclusions les plus importantes de l'étude et propose des perspectives à ces travaux de recherche.

---

## Chapitre 2

Ce chapitre explique les méthodes utilisées pour transférer les chargements obtenus d'une simulation aéro-élastique à partir d'une modélisation poutre a un modèle de coque 2D de la pale de l'éolienne. Le contenu du chapitre est résumé ci-dessous :

### Méthodes de simulation pour l'éolienne

Pour simuler l'éolienne en considérant différentes vitesses moyennes du vent avec un niveau de turbulence normal, des modèles aéro-élastiques sont utilisés pour déterminer le comportement de l'éolienne en tenant en compte les effets aérodynamiques, élastiques et les effets produits par le contrôleur de production d'énergie. La représentation simplifiée de l'éolienne est effectuée en un modèle 1D où la pale est modélisée par des éléments finis de poutres qui permettent d'obtenir les chargements résultant sur des nœuds localisés aux centres aérodynamiques. Pour exécuter ces simulations, il faut déterminer les propriétés mécaniques de l'éolienne, ainsi que les conditions environnementales auxquelles elle sera soumise. Cette partie explique comment sont estimées les conditions environnementales en utilisant la méthode de Veer et la dépendance entre la vitesse moyenne du vent et l'écart-type de la vitesse moyenne et l'intensité de turbulence. Ce chapitre présente aussi la simplification de la distribution de matériaux composites de la pale de l'éolienne 10 MW de référence développée par DTU, le calcul des propriétés mécanique par section et ses modes de vibration.

### Méthodes d'application des charges

Les méthodes d'application des charges (LAM: *Load Application Methods*) classifiées par Caous [Caous et al., 2018] sont utilisées pour transférer les charges résultantes des outils de simulation aéro-élastique dans une répartition équivalente des charges sur le modèle aux éléments finis (FEM) de coque. La réponse du modèle FEM de coque doit être égale à celle calculée à l'aide de l'outil de simulation aéro-élastique. La représentation de la pale par un modèle aux éléments finis de coque permet de calculer la répartition détaillée du champ de contraintes sur la pale en matériaux composites. Cette analyse des contraintes à l'aide d'un modèle FEM permettra ensuite d'effectuer une analyse de fatigue multiaxiale.

Ces méthodologies sont classées par Caous [Caous et al., 2018] en fonction de la façon dont les charges sont appliquées sur le modèle de coque en quatre groupes principaux :

- Application des charges par sections et sur un point de chaque section. [Forcier and Joncas, 2012], [Haselbach et al., 2016],[Griffith and Ashwill, 2011],[Shokrieh and Rafiee, 2006], [Lindgaard and Lund, 2010], [Lund and Stegmann, 2005].
- Application des charges ponctuelles, mais répartition physique sur la section. [Bottasso et al., 2014], [Caous and Valette, 2014].

- Application continue sur la pale d'une charge de surface orientée (pression orientée dans une direction spécifique). [Forcier and Joncas, 2012], [Hu et al., 2012], [Sørensen et al., 2014], [Dimitrov, 2013], [Berggreen et al., 2007], [McKittrick et al., 2001].
- Dissociation des charges inertielles et aérodynamiques avec application d'un champ d'accélération et répartition de la pression sur toute la pale. [Chen et al., 2013], [Grujicic et al., 2010], [Hu et al., 2016b], [Knill, 2005], [Caous et al., 2018].

Dans ce travail de thèse, nous avons sélectionné trois méthodes d'application des charges (LAM) nécessaires pour l'analyse FEM de la pale : du groupe 1) la méthode des éléments de corps rigide (RBE), du groupe 2) la méthode d'application des charges sur 4 nœuds (4NO) et du groupe 4) la méthode de répartition de la pression définie par Caous (PD).

Ces méthodes sont appliquées et comparées au niveau des éléments du tenseur des contraintes de la pale. La méthode "PD" produit la distribution la plus réaliste des charges parmi les trois méthodes étudiées.

## Chapitre 3

Ce chapitre présente l'apprentissage des réseaux de neurones profonds (DNN: Deep Neural Network) sur la réponse aéro-élastique simulée de la pale d'éolienne. Ensuite, l'utilisation de ce réseau de neurones profond pour la prédiction des charges. Ces dernières années les réseaux de neurones profonds ont été intensivement utilisés dans de nombreux domaines différents, y compris des composants d'éolienne, l'énergie éolienne, la prévision de la vitesse du vent, etc. Un état de l'art est présenté dans ce chapitre sur l'application de DNN sur les éoliennes.

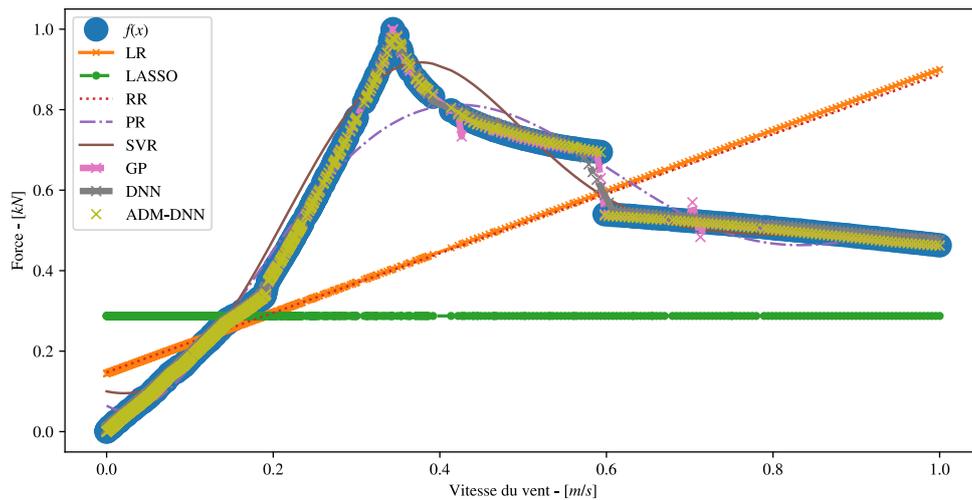
Une introduction du fonctionnement et de l'architecture des DNN est présentée. Ainsi, que les différents algorithmes d'optimisation pour l'apprentissage du réseau. Nous présentons dans ce chapitre les problèmes de sur-apprentissage et les différentes approches pour les éviter. Ensuite, un exemple académique est utilisé pour comparer les performances des algorithmes d'optimisation des hyperparamètres du DNN. Cet exemple, montre que l'algorithme d'optimisation bayésienne est plus efficace par rapport à l'algorithme de la recherche aléatoire (random search algorithm [Bergstra and Bengio, 2012]).

### Modélisation de la discontinuité des charges simulées des éoliennes

La simulation des écoulements laminaires et turbulents sur l'éolienne avec FAST produit des sauts dans la réponse des chargements en fonction de la vitesse du vent. Ce comportement produit une réponse avec une discontinuité, cela rend l'apprentissage des DNN plus difficile. Cette discontinuité est produite par le contrôleur utilisé pour la pale. Ce contrôleur est développé par HAWC2 [GL, 2015] et transformé en fichier compatible pour le logiciel FAST. Cependant, ce contrôleur n'est pas optimisé, puisque les charges résultantes présentent un saut

à partir de la vitesse du vent entre  $16\text{m/s}$  et  $17\text{m/s}$ . Dans cette plage de vitesse du vent, le contrôleur met les pales en arrêt en prenant la valeur maximale de l'angle d'incidence de la pale (*pitch angle*). Pour cela, une nouvelle méthode est proposée pour améliorer l'apprentissage du DNN dans cette zone de discontinuité de la réponse aéro-élastique.

La méthode proposée nommée (ADM-DNN, *Automatic Discontinuity Modelling - DNN*) consiste à déterminer la zone des discontinuités automatiquement dans le problème analysé, en utilisant un modèle de mélange gaussienne (*GMM: Gaussian Mixture Model*). GMM permet de classifier et d'étiqueter les différentes zones du domaine où la réponse est continue. Avec cette classification, chaque valeur est étiquetée et elle est ajoutée comme une nouvelle variable d'entrée du modèle. Ensuite, un modèle de DNN est entraîné et validé avec le changement de dimension des paramètres d'entrée. Cette méthodologie proposée est utilisée et comparée à d'autres méthodes d'apprentissage automatique pour la régression en utilisant un exemple académique d'une part ; d'autre part à la prédiction des chargements résultant de la simulation aéro-élastique de la pale. Les deux exemples montre l'efficacité de la méthode proposée par rapport aux autres méthodes, où la prédiction de la réponse est réalisée avec le moins d'erreur.



Comparaison des méthodes de régression pour prédire le chargement aéro-élastique de la pale.

## Chapitre 4

L'analyse de sensibilité des éoliennes est effectuée en utilisant différentes méthodologies en fonction de la complexité du problème de conception. Les méthodes d'analyse de sensibilité globale sont souvent basées sur la méthode des simulation de Monte Carlo (MCS), pour l'estimation des indices de Sobol. Ces méthodes sont utilisées pour des problèmes de faible dimension avec un modèle à coût de calcul élevé [Ziegler et al., 2015] ou pour des problèmes de grande dimension avec un modèle à coût de calcul faible [Tran and Smith, 2018], ce qui signifie

que de grands échantillons peuvent être produits avec un coût de calcul modéré. Cependant, pour les problèmes liés à la modélisation de problèmes à grandes dimensions et à coût de calcul élevé, ces méthodes sont inabordables. L'analyse de sensibilité peut-être effectuée dans une grille de valeurs sélectionnées par l'utilisateur comme dans l'étude effectuée par Geng et al. [Geng et al., 2018], ces méthodes dites "locales" permettent à faible coût de calcul d'estimer la sensibilité des paramètres d'entrée sur la réponse dans cet intervalle seulement. Toutefois, les méthodes de sensibilité dites "globales" utilisent les simulations de Monte Carlo pour évaluer la réponse du modèle en différents points du domaine. Pour cela, elles sont plus coûteuses en temps de calcul,

L'une des méthodes de sensibilité dites "locale" est l'analyse de «criblage» de Morris [Morris, 1991] ou One-at-a-Time (OAT) utilisée pour quantifier les effets de la sensibilité des paramètres d'entrée sur le paramètre de sortie. Cette méthode est évaluée dans un sous-ensemble du domaine des paramètres d'entrée et ne fait varier qu'une seule variable par itération ; Velarde et al. [Velarde et al., 2019] présente une étude de criblage dans les charges de fatigue des fondations d'une éolienne. Les résultats obtenus sont faciles à comprendre, et comme le nombre d'itérations est linéaire par rapport au nombre de paramètres d'entrée, il est fréquemment utilisé pour les problèmes de grande dimension. Cependant, la méthode de Morris n'évalue pas l'interaction entre les variables de manière quantitative ; autrement dit, on ne sait pas avec exactitude quelle est la valeur de l'effet de la sensibilité des variables d'entrée sur la sortie. Pour cette raison, la méthode de Morris est utilisée comme une étape préalable pour réduire le nombre de variables d'entrée et sélectionner les variables qui ont un effet de la sensibilité élevé pour les utiliser par la suite dans une analyse de sensibilité basée sur la variance ; Hübler et coll. [Hübler et al., 2017] présente un exemple de cette méthodologie. Pour une analyse de sensibilité globale des éoliennes offshore basée sur des simulations aéro-élastiques du domaine temporel. Premièrement l'étude commence avec plus de 50 paramètres pour l'analyse de criblage de Morris et ensuite 20 paramètres les plus sensibles sont sélectionnés pour l'analyse de sensibilité basée sur la variance.

Néanmoins, pour certains problèmes, les paramètres d'entrée peuvent avoir des interactions entre eux. En ignorant certaines paramètres d'entrée après la réduction, leurs interactions internes ne seront pas mesurées en utilisant l'analyse de sensibilité basée sur la variance. Il existe différentes analyses de sensibilité basées sur la variance comme la méthode Sobol [Sobol, 1993] et le test de sensibilité d'amplitude de Fourier [Cukier et al., 1973, Cukier et al., 1975, Cukier et al., 1978]. Ces méthodes quantifient l'effet de sensibilité de chaque paramètre d'entrée et leurs interactions en calculant des indices de différents degrés, représentant l'effet sur la variance de sortie produite par les variables d'entrée et leurs interactions. Par rapport à l'analyse de criblage de Morris, ces méthodes sont évaluées dans tout le domaine des paramètres d'entrée, généralement à l'aide des simulations de Monte Carlo, ce qui rend leur coût de calcul onéreux. Echevarria et al. [Echeverría et al., 2017] présente un étude utilisant cette méthode pour estimer la sensibilité des variables de géométrie des pales sur les performances de l'éolienne avec 21 paramètres d'entrée. D'autres auteurs utilisent un modèle de substitution pour remplacer le modèle d'origine et estimer l'analyse de sensibilité globale basée sur le modèle de substitution, réduisant ainsi le temps de calcul due à l'évaluation du modèle d'origine. McKay et coll. [McKay et al., 2014] a étudié la sensibilité globale de la

---

puissance de sortie des éoliennes en se basant sur des données expérimentales et en ne considérant que huit paramètres d'entrée pour entraîner leur modèle de substitution. La dimension des paramètres d'entrée de ce problème est petite (moins de 50 variables) et aussi les données peuvent être obtenues par expérimentation, leur coût de calcul est plus faible que celui des problèmes où les résultats des réponses sont obtenues à l'aide d'outils de simulation aéro-élastique, où l'analyse de sensibilité doit être effectuée avec un grand nombre des paramètres d'entrée (plus de 50 variables).

Cette étude présente une nouvelle méthodologie pour calculer la sensibilité globale des problèmes de grande dimension sans la réduction du nombre de paramètres d'entrée. Cette méthodologie est utilisée pour déterminer la sensibilité des méthodes LAM.

La méthode est divisée en deux étapes. La première étape consiste à l'apprentissage d'un modèle DNN pour prédire le chargement aéro-élastique obtenu par FAST du modèle 1D de la pale. Nous obtenons 54 chargements dans différents nœuds de la pale modélisée en 1D (modèle de poutre). Ensuite, le second DNN est entraîné en utilisant les chargements précédents et la distribution des matériaux composites de la pale (24 variables d'épaisseur) comme variables d'entrée (78 en total) du modèle FEM pour calculer la contrainte de Von Mises au pied de la pale. Une fois le DNN entraîné et validé, il est utilisé pour prédire le tenseur des contraintes. En utilisant ce dernier modèle de DNN, la sensibilité des paramètres des méthodes LAM est calculée par la méthode de Sobol.

Les résultats obtenus indiquent que la méthode LAM PD est la moins sensible, suivie par RBE et enfin 4NO. LAM PD présente une sensibilité linéaire par rapport aux charges appliquées entre le milieu et le bout de la pale dans la direction de la portance. Cependant, RBE et 4NO présentent un comportement non-linéaire significatif indiquant l'impact des interactions des variables d'entrée sur la sensibilité de la contrainte. RBE présente une plus faible interaction non-linéaire que 4NO.

Pour valider et comparer les résultats obtenus avec la méthodologie développée, une analyse de criblage de Morris est effectuée en utilisant directement le modèle FEM de coque 2D de la pale. Les résultats de l'analyse de sensibilité de Morris montrent que les charges avec l'effet élémentaire moyenne  $\mu^*$  le plus élevés sont les mêmes que celles trouvées par la méthode proposée dans ce travail pour le calcul de l'indice total de Sobol. Ce qui indique que l'apprentissage du modèle DNN a bien représenté la relation entre les variables d'entrée et de sortie. Campolongo [Campolongo et al., 2007] a proposé une comparaison entre la méthode de sensibilité de Morris qui est qualitative et la méthode de la décomposition de la variance basée sur l'estimation des indices de Sobol. Cependant, il est utile de clarifier que la méthode de Morris est réalisée sur un sous-domaine et les indices de Sobol sont estimés sur le domaine entier, ce qui permet de quantifier l'interaction entre les variables.

La méthode LAM PD est ensuite utilisée pour la suite de la thèse dans l'analyse de fatigue et l'estimation probabiliste de la durée de vie en fatigue de la pale en matériaux composites.



---

aéro-élastiques résultantes agissant sur le modèle 1D de la pale. Ensuite, ces charges sont transférées vers un modèle FEM de coque 2D de la pale en utilisant deux méthodes LAM: 4 noeud et PD. Selon le LAM utilisé, les charges réparties sont appliquées au modèle de coque de la pale pour l'analyse des contraintes. Le calcul des dommages de fatigue prend en compte l'état des contraintes non-proportionnelles par un critère de fatigue multiaxiale, un comptage de cycles par la méthode Range-Mean et la règle de Miner pour le cumul du dommage. Le dommage par fatigue est calculé au niveau des couches du stratifié, ce qui indique la défaillance à une position locale.

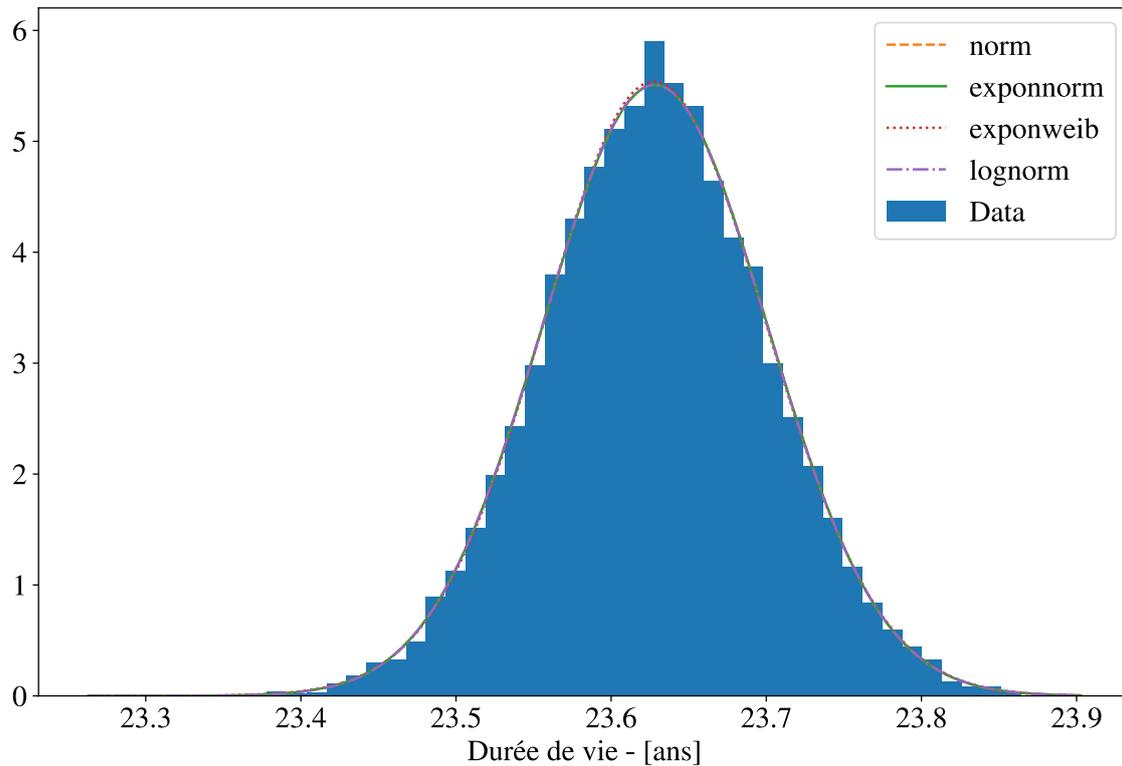
Après avoir étudié la distribution des contraintes et de la fatigue sur la pale, une méthodologie est développée pour permettre l'apprentissage du DNN afin de l'utiliser ensuite pour prédire l'endommagement par fatigue pendant 10 minutes. Cette méthodologie consiste à trouver la fonction de répartition empirique de l'endommagement pour chaque vitesse moyenne du vent et à ajouter la valeur de la probabilité cumulée comme variable d'entrée. Cette approche permet d'estimer l'endommagement par fatigue pendant 10 minutes avec une bonne précision. Ce DNN est testé pour prédire l'endommagement pendant 10 minutes d'un historique du vent à une vitesse moyenne différente de celle utilisée pour l'apprentissage. Les erreurs maximales trouvées sont de moins de 14% pour la couche de matériau P2B et 9% pour la couche de matériau QQ1.

En utilisant ce DNN, il est possible d'estimer la durée de vie probabiliste de la pale, où 10 000 échantillons de 10 minutes d'endommagement par fatigue sont générés pour chaque vitesse du vent entre 4 et 24  $m/s$ , soit pour un total de 210 000. Des données expérimentales du projet open-data Engie Renewable à la Haute Borne, Grand Est, France [Engie, 2019] ont été utilisées afin d'estimer les répartitions de chaque vitesse pendant une année. Ces données ont été collectées pour une éolienne de 2 MW à une hauteur de moyeu (Hub) de 80m. L'éolienne utilisée dans cette étude a une hauteur ou moyeu de 119m. Il est supposé que ce pourcentage de durée de chaque vitesse du vent soit la même durant 20 ans.

L'endommagement par fatigue pendant 20 ans est estimé sur les différentes couches des matériaux composites de la pale. Le niveau d'endommagement le plus élevé est situé sur la couche du matériau composite QQ1. Cette couche est utilisée pour estimer la durée de vie d'un point de la pale. Nous avons fait l'hypothèse que la pale génère de l'électricité sans s'arrêter pendant 20 ans. La distribution probabiliste de la durée de vie dans ce point suit une loi de distribution normale. Les tests de Kolmogorov-Smirnov et d'Anderson-Darling n'ont pas rejeté l'hypothèse de la distribution normale. La durée de vie estimée au fractile de 5% en ce point est de 23,51 ans.

## Conclusions

Dans ce travail de thèse, nous avons développé une procédure d'analyse de fatigue pour les pales d'éolienne. Cette procédure est basée sur plusieurs outils de simulation : i) la simulation par FAST du champ du vent turbulent, ii) simulation la réponse dynamique temporelle couplée en utilisant des modèles aérodynamiques, des modèles de contrôle et de système élec-



Estimation probabiliste de la durée de vie de la pale pour la couche QQ1 dans la position  $Z = 51.257m$  de la longueur de la pale.

trique (servo) et structurels (élastiques) ; iii) transfert des charges obtenue par les simulations aéro-élastiques 1D au modèle aux éléments finis de coque 2D de la pale ; iv) estimation des historiques du tenseur des contraintes dans les différents plis des stratifiées des matériaux composites de la pale ; v) analyse de fatigue multiaxial en tenant compte de la non proportionnalité des contraintes.

Un modèle de régression est développé pour prendre en compte les discontinuités de saut. Ces discontinuités de la réponse sont dues au fonctionnement du contrôleur des pales. Le modèle proposé utilise le modèle de mélange gaussien (Gaussien Mixture) et les réseaux de neurones profonds (Deep Neural Network). La méthode proposée présente une erreur quadratique moyenne inférieure par rapport à d'autres méthodes de régression d'apprentissage automatique utilisées.

Ce travail de thèse propose une méthode originale pour l'analyse de sensibilité globale pour les problèmes à grande dimension. Cette approche est utilisée pour l'analyse des sensibilités globale des méthodes d'application de charges (LAM), utilisées pour transférer les charges du vent issues des simulations aéro-élastiques 1D au modèle aux éléments finis de coque 2D de la pale. L'approche développée emploie la décomposition de la variance en estimant les indices de Sobol. Ces indices de Sobol sont estimés en utilisant un modèle de substitution

---

basé sur un réseau de neurones profond. Cette méthodologie est basée sur le développement de deux réseaux de neurones, le premier estime les charges aéro-élastiques maximales pour une vitesse moyenne du vent donnée et le second consiste à estimer les contraintes mécaniques au pied de la pale à l'aide d'un modèle aux éléments finis, en tenant compte des épaisseurs des matériaux composites et des charges aéro-élastiques maximales obtenues à partir du premier réseau de neurones. Les résultats obtenus par cette analyse de sensibilité en utilisant un réseau de neurones profonds sont validés et comparés aux résultats de la méthode de Morris.

De plus, les résultats de sensibilité obtenus des méthodes LAM testées montrent que la méthode PD est la moins sensible par rapport aux autres méthodes (4NO ET RBE). Cette étude a permis de sélectionner cette méthode d'application des charges pour l'analyse de fatigue de la pale.

Ce travail de thèse propose le développement d'une approche probabiliste de la durée de vie en fatigue d'une pale d'éolienne en matériaux composites. L'utilisation de la chaîne de simulation décrite ci-dessus pour propager l'incertitude de la vitesse du vent dans la durée de vie en fatigue est impraticable en utilisant les simulations de Monte Carlo. Pour cela, un modèle de substitution basé sur le réseau de neurones profond est utilisé pour prédire la durée de vie en fatigue de l'éolienne. L'apprentissage du réseau de neurones profond est effectué en développant une approche originale, basée sur l'estimation de la fonction de répartition empirique de l'endommagement pour chaque vitesse moyenne du vent, cette fonction de répartition de probabilité cumulée est ajoutée aux variables d'entrée du réseau de neurones profond. Les simulations de Monte Carlo en considérant l'incertitude de la vitesse du vent sont alors effectuées en utilisant le modèle de substitution. La distribution probabiliste de la durée de vie suit une loi de distribution normale et la durée de vie de la pale estimée au fractile de 5% est de 23,5 ans.

## Recommandations pour les travaux futurs

Plusieurs améliorations peuvent être apportées aux travaux de recherche développées. Premièrement, la méthode d'application de la répartition de la pression développée par Caous, pourrait être implémentée dans une autre logiciel d'analyse par éléments finis où les forces d'inertie pourraient être appliquées comme suggérées par l'auteur en tant que champs d'accélération. Contrairement à l'implémentation qui a été faite dans ce travail dans le logiciel CODE-ASTER, les forces d'inertie sont appliquées en utilisant des forces réparties proportionnelles à la surface. De plus, comme le contrôle de la pale peut présenter un comportement énergétique non-productif, le coefficient de pression à un angle d'attaque plus élevé, ce qui pourrait être calculé à l'aide du logiciel 2D Xfoil. Dans cette étude, l'angle maximum calculé est  $\pm 32^\circ$  et après cet angle, le coefficient de pression est supposé être ce dernier. Pour les recherches futures, le contrôle de la pale doit être optimisé afin d'assurer une simulation de la production d'énergie pour toutes les vitesses du vent entre les vitesses d'enclenchement et de coupure. Deuxièmement, la méthodologie d'analyse de sensibilité globale présentée dans cette thèse est validée à l'aide d'une analyse de Morris, pour les recherches futures, il est recommandé de

calculer la sensibilité globale en utilisant le modèle aux éléments finis de coque 2D de la pale pour comparer et valider les indices de Sobol obtenues par l'approche proposée en utilisant le DNN. L'utilisation d'un solveur GPU pour le modèle de simulation FEM afin d'effectuer un calcul parallèle à haute performance. L'analyse de sensibilité a été réalisée en supposant des conditions environnementales d'écoulement laminaire, pour les recherches futures, une étude en tenant compte d'un écoulement turbulent pourrait produire des résultats différents introduisant plus d'incertitude dans les charges du vent. Troisièmement, les modèles de substitution utilisés pour l'analyse de fatigue ont été entraînés uniquement en considérant un écoulement turbulent du vent et un seul historique temporel du champ de vent. Ce modèle de substitution a présenté une bonne prédiction par rapport à la quantité de données entraînées. Il est recommandé pour les futures recherches d'augmenter le nombre d'échantillons d'apprentissage en tenant compte les différentes conditions environnementales, différents historiques de champ de vent et les différentes incertitudes des paramètres. Ainsi, le modèle de substitution peut prédire le dommage par fatigue avec une meilleure précision et pourrait-être utilisé pour une analyse de fiabilité.

Pour les travaux futurs, la méthodologie développée pour calculer la distribution probabiliste de la durée de vie en fatigue pourrait être utilisée pour une analyse de fiabilité, puis une conception d'optimisation basée sur la fiabilité (RBDO) de la pale d'éolienne. Dans ce cas, le poids ou le coût total de la pale est minimisé sous la contrainte que la probabilité de défaillance de l'état limite de fatigue soit inférieure à une valeur cible. Cette approche d'optimisation fiabiliste permettrait une distribution optimale des matériaux composites de la pale.

# List of publications and presentations

## International conferences

- Veloz, W. J., Bai, H., Aoues, Y., & Lemosse, D. (2021, January). Application of artificial neural network for fatigue analysis in wind turbine blade. In 14th World Congress in Computational Mechanics (WCCM) ECCOMAS Congress 2021.
- Veloz, W. J., Zhang, H., Bai, H., Aoues, Y., & Lemosse, D. (2020, June). Uncertainty Propagation in Wind Turbine Blade Loads. In International Symposium on Uncertainty Quantification and Stochastic Modeling (pp. 337-346). Springer, Cham.
- Parra, W. J. V., Aoues, Y., & Lemosse, D. (2019, July). Sensitivity Analysis of Load Application Methods for Shell Finite Element Models. In World Congress on Global Optimization (pp. 1001-1010). Springer, Cham.
- Veloz Parra W.J., Aoues Y., & Lemosse D. (2019, Oct.). Sensitivity Analysis of Load Application Methods for Blade Shell Finite Element Model. In 15th EAWE PhD seminar, Nantes, France.

## International presentation

- Veloz Parra W.J., Aoues Y., & Lemosse D. (2019, Nov.) Sensibilidad en métodos de aplicación de cargas usados para aspas de generadores eólicos. In XVIII Jornada de Investigación Científica, Santo Domingo, República Dominicana.

## Journal submissions

- Veloz Parra W. J., Bai H., Zhang H., Aoues Y., & Lemosse D. (2020, Aug). Two-stages deep learning global sensitivity approach for wind turbine blade load application methods. In Journal of Wind Engineering & Industrial Aerodynamics. Under review.



# Contents

<b>Abstract</b>	<b>iii</b>
<b>Résumé</b>	<b>v</b>
<b>Résumé étendu</b>	<b>vii</b>
<b>List of publications and presentations</b>	<b>xxi</b>
<b>List of abbreviations and symbols</b>	<b>xxxix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Context . . . . .	1
1.2 Previous work . . . . .	2
1.3 Proposed study . . . . .	6
1.4 Outline of the thesis . . . . .	7
<b>2 Load application methods from beam to shell FEM</b>	<b>9</b>
2.1 Simulation codes for wind turbine . . . . .	10
2.2 Load Application Methods - LAM . . . . .	23
2.3 Classification . . . . .	24
2.4 Application of LAM to a composite wind turbine blade . . . . .	37
2.5 Conclusions . . . . .	41
<b>3 Deep learning application for wind turbine blade response</b>	<b>43</b>
3.1 Deep learning in Wind Turbine . . . . .	44
3.2 Introduction to deep neural network . . . . .	45
3.3 Tuning hyperparameters . . . . .	57

3.4	Discontinuity modeling for wind turbine simulated loads . . . . .	61
3.5	Conclusions . . . . .	69
<b>4</b>	<b>Sensitivity of LAMs for wind turbine using surrogate models</b>	<b>71</b>
4.1	Introduction to sensitivity analysis . . . . .	72
4.2	Sensitivity analysis in wind turbine . . . . .	81
4.3	Sensitivity analysis of high dimensional models using surrogate models . . . . .	83
4.4	Conclusions . . . . .	103
<b>5</b>	<b>Fatigue life estimation of wind turbine blades using DNN</b>	<b>105</b>
5.1	Aero-elastic unsteady simulation . . . . .	106
5.2	LAM wind turbine blade 2D shell FEM for stress analysis . . . . .	110
5.3	Multi-axial fatigue criteria for composite materials . . . . .	115
5.4	DNN: 10-minutes damage . . . . .	124
5.5	20 years fatigue damage estimation using DNN . . . . .	138
5.6	Probabilistic fatigue life estimation for LAM PD . . . . .	143
5.7	Conclusions . . . . .	145
<b>6</b>	<b>Conclusions</b>	<b>147</b>
6.1	Conclusions . . . . .	147
6.2	Recommendations for future works . . . . .	151
<b>A</b>	<b>Damage Distribution per Layer</b>	<b>153</b>
A.1	LAM 4NO 10 minutes Damage P2B Layer: Inferior Level . . . . .	153
A.2	LAM 4NO 10 minutes Damage QQ1 Layer: Inferior Level . . . . .	159
A.3	LAM PD 10 minutes Damage P2B Layer: Inferior Level . . . . .	165
A.4	LAM PD 10 minutes Damage QQ1 Layer: Inferior Level . . . . .	171
<b>B</b>	<b>20 years Damage per Wind Speed</b>	<b>177</b>

B.1 LAM PD 20 years Damage P2B Layer: Inferior Level . . . . . 177

B.2 LAM PD 20 years Damage QQ1 Layer: Inferior Level . . . . . 182

**Bibliography**

**187**



# List of Figures

1	Comparaison des méthodes de régression pour prédire le chargement aéro-élastique de la pale. . . . .	xiii
2	Indices de Sobol d'ordre total en utilisant LAM RBE et 4NO pour estimer la contrainte au pied de pale. . . . .	xvi
3	Indices de Sobol d'ordre total en utilisant LAM PD pour estimer la contrainte au pied de pale. . . . .	xvi
4	Estimation probabiliste de la durée de vie de la pale pour la couche QQ1 dans la position $Z = 51.257m$ de la longueur de la pale. . . . .	xviii
1.1	Methodology for analyzing the structural reliability of wind turbine blades [Dimitrov, 2013]. . . . .	2
1.2	Methodology for structural blade optimization [Bottasso et al., 2014] . . . . .	4
1.3	Methodology for reliability-based optimization of wind turbine blade [Hu et al., 2016a]	4
1.4	Methodology for surrogate model uncertainty in wind turbine reliability assessment. . . . .	5
1.5	Proposed methodology for probabilistic fatigue damage estimation of a composite wind turbine blade. . . . .	6
2.1	Wind ramp simulation results for the 10 MW DTU simplified wind turbine blade using FAST. . . . .	16
2.2	Circumferential region division of the wind turbine blade for the composite material distribution [Bak et al., 2013] . . . . .	17
2.3	Blade span-wise partition for a constant thickness composite material distribution	18
2.4	Region definition and sequence ordering for airfoil section $n^o21$ . . . . .	20
2.5	2D representation of blade section $n^o10$ a) Elastic, shear and tension centers and b) material distribution (2 = BALSA, 3 = QQ1 and 4 = P2B). . . . .	21
2.6	Blade mass distribution per sections. . . . .	21
2.7	Stiffness distribution per section in a) edge-wise and b) flap-wise directions. . .	22
2.8	Polynomial coefficient calculation for the 2nd shape mode of the blade. . . . .	24

2.9	Wind turbine blade assessment process [Caous et al., 2018]. . . . .	25
2.10	Approaches for load application methods in a shell FEM of the blade [Caous et al., 2018].	25
2.11	Loads distribution per section using Rigid Body Elements . . . . .	26
2.12	Load application method 2: (a) Recovery of inertial and aerodynamic loads from sectional beam FEM loads and (b) inertial load application to spar caps nodes using the RBE3 element [Bottasso et al., 2014] . . . . .	28
2.13	Loads distribution on 4 nodes per section. . . . .	28
2.14	Aerodynamic pressure computation and correction [Caous et al., 2018]. . . . .	32
2.15	Geometrical properties: chord, thickness, aerodynamic center ( $C$ , $B$ and $o$ ) and pressure components $P_x$ and $P_y$ (blue and red arrows respectively) of a segment in an airfoil. . . . .	33
2.16	Maximum and minimum $\sigma_{22}$ stress of first layer of composite material P2B of 10MW Wind turbine blade at Gaussian points. . . . .	39
2.17	Von Mises plane stress comparison for the LAMs at fixed coordinate a) $X = -0.5$ and b) $X = 0$ . . . . .	40
2.18	Deflection comparison of LAM and aero-elastic response in flapwise direction. . . . .	41
3.1	Description linear threshold unit [Géron, 2019] . . . . .	46
3.2	Perceptron diagram for regression [Géron, 2019]. . . . .	47
3.3	Perceptron diagram for regression [Géron, 2019]. . . . .	48
3.4	Activation functions used in LTUs [Géron, 2019]. . . . .	50
3.5	Leaky ReLU activation function, $alpha = 0.1$ [Géron, 2019]. . . . .	51
3.6	ELU activation function, $alpha = 1$ [Géron, 2019]. . . . .	52
3.7	Benchmark function for training a Deep Neural Network. . . . .	59
3.8	Load $S_{pn1FLxb1}$ (Span location 1, force in lift direction for blade 1) distribution between wind speeds from $4m/s$ to $25m/s$ . . . . .	62
3.9	Aero elastic simulation for $19m/s$ normal turbulence wind speed in X-direction, power generated and pitch angle of the blade (from up to down respectively). . . . .	63
3.10	Benchmark function: $10sign(x)$ . . . . .	66
3.11	Gaussian mixture cluster classification. . . . .	67

3.12	Comparison of regression methods predicting a $f(x) = 10\text{sign}(x)$ . . . . .	68
3.13	Comparison of regression methods predicting an aero-elastic load $Spn1FLxb1$ on a wind turbine blade. . . . .	69
4.1	Morris method classification of input parameters. . . . .	75
4.2	Simulation methodology to calculate and compare the Sobol global sensitivity analysis of LAMs: 4 nodes (4NO),rigid body elements (RBE) and pressure distribution Caous method (PD). . . . .	84
4.3	Wind turbine blade cross section in the span-wise direction equally located at the gages from FAST simulation. . . . .	85
4.4	Load $Spn1FLxb1$ (Span location 1, force in lift direction for blade 1) maximum value extraction for a steady simulation at $11m/s$ . . . . .	85
4.5	a) Wind speed histogram at wind turbine hub height and b) prediction of load histogram at span location 1 acting in X-direction. . . . .	90
4.6	Validation of all predicted aero-elastic loads. . . . .	90
4.7	Relative error of all predicted aero-elastic loads. . . . .	91
4.8	Input parameters domain for surrogate model 2, material thickness and aero-elastic loads. . . . .	92
4.9	10MW wind turbine blade and shell local coordinate system. . . . .	93
4.10	Validation of predicted and simulated stress for LAM a) RBE and b) 4 Nodes. . . . .	94
4.11	Validation of predicted and simulated stress for LAM PD. . . . .	95
4.12	Comparison of first order Sobol index for RBE and 4NO methods to estimate the detailed stress distribution in a blade shell FEM. . . . .	96
4.13	First order Sobol index using LAM PD to estimate the detailed stress distribution in a blade shell FEM. . . . .	96
4.14	Comparison of total order Sobol index using RBE and 4NO methods to estimate the detailed stress distribution in a blade shell FEM. . . . .	97
4.15	Total order Sobol index using LAM PD to estimate the detailed stress distribution in a blade shell FEM. . . . .	98
4.16	Second order Sobol index for Von Mises stress at blade root using LAM 4 node. . . . .	98
4.17	Second order Sobol index for Von Mises stress at blade root using LAM RBE. . . . .	99

4.18	Second order Sobol index for Von Mises stress at blade root using LAM PD. . . . .	99
4.19	Morris sensitivity analysis for LAM: RBE, 4 nodes and PD. . . . .	101
4.20	Zoom of Morris sensitivity analysis for LAM PD. . . . .	102
5.1	Methodology proposed for 10 minutes fatigue damage estimation. . . . .	106
5.2	environmental condition parameters distribution. . . . .	108
5.3	Turbsim wind field generation for wind speed $11m/s$ and normal turbulence, DLC 1.2. . . . .	109
5.4	FAST load output time history for $S_{pn1FLxb1}$ at $11m/s$ with a time step of $0.015s$ and re-sampled to a time step of $0.63s$ , and a zoom between 200 and 300 seconds. . . . .	111
5.5	Location node 606 on blade root. . . . .	112
5.6	Longitudinal $\sigma_1$ stress distribution through laminate thickness at node 606 using LAM 4NO for a) wind speed $11m/s$ and b) wind speed $16m/s$ . . . . .	113
5.7	Stress history at node 606, second ply, inferior position at wind speed $11m/s$ using LAM 4NO. . . . .	113
5.8	Stress distribution through laminate thickness at node 606 using LAM PD for a) wind speed $11m/s$ and b) wind speed $16m/s$ . . . . .	114
5.9	Stress history at node 606, second ply, inferior position at wind speed $11m/s$ using LAM PD. . . . .	114
5.10	Range-mean cycle counting example. . . . .	116
5.11	95% confidence band for the S-N curve of QQ1 in longitudinal direction with $R = -0.5$ . . . . .	118
5.12	Constant life diagram of QQ1 in longitudinal direction using lower bound 95% S-N curves. . . . .	119
5.13	Constant life diagram of QQ1 in transversal direction using lower bound 95% S-N curves. . . . .	120
5.14	Constant life diagram of P2B in longitudinal direction using lower bound 95% S-N curves. . . . .	121
5.15	Constant life diagram of P2B in transversal direction using lower bound 95% S-N curves. . . . .	122

5.16	Constant life diagram of QQ1 in shear direction using lower bound 95% S-N curves. . . . .	123
5.17	Constant life diagram of P2B in shear direction using lower bound 95% S-N curves. . . . .	123
5.18	a) Damage distribution $D^i$ per wind speed and b) ECDF per wind speed of Damage using LAM 4NO at Node 506. . . . .	125
5.19	Convergence curve of hyper-parameter optimization a) first run b) second run starting from optimum of first run. . . . .	127
5.20	Hyper-parameters space search points for DNN 10 Minute Damage. . . . .	128
5.21	Blade nodes (red dots) where the damage is calculated using LAMs 4NO and PD. . . . .	132
5.22	10 minutes damage distribution along the blade Z coordinate as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO. . . . .	133
5.23	10 minutes damage distribution along the blade Z coordinate as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO. . . . .	134
5.24	10 minutes damage distribution along the blade Z coordinate as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . .	135
5.25	10 minutes damage distribution along the blade Z coordinate as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . .	136
5.26	20 years damage distribution at node 606, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . .	141
5.27	20 years damage distribution along the blade in Z-direction at first layer (P2B), inferior level by using LAM PD. . . . .	141
5.28	20 years damage distribution at node 606, section 1 of the blade as a function of the wind speed at first layer (QQ1), inferior level by using LAM PD. . . . .	142
5.29	20 years damage distribution along the blade in Z-direction at first layer (QQ1), inferior level by using LAM PD. . . . .	142
5.30	Kolmogorov-Smirnov test for fatigue life estimation distribution of node 549 at position $Z = 51.257m$ and material QQ1 a) CDF and b) PDF. . . . .	143
A.1	10 minutes damage distribution at node 606, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO. . . . .	153

- 
- A.2 10 minutes damage distribution at node 603, section 2 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO. . . . 154
- A.3 10 minutes damage distribution at node 595, section 3 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO. . . . 154
- A.4 10 minutes damage distribution at node 586, section 4 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO. . . . 155
- A.5 10 minutes damage distribution at node 580, section 5 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO. . . . 155
- A.6 10 minutes damage distribution at node 567, section 6 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO. . . . 156
- A.7 10 minutes damage distribution at node 549, section 7 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO. . . . 156
- A.8 10 minutes damage distribution at node 531, section 8 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO. . . . 157
- A.9 10 minutes damage distribution at node 516, section 9 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO. . . . 157
- A.10 10 minutes damage distribution at node 506, section 10 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO. . . . 158
- A.11 10 minutes damage distribution at node 606, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO. . . . 158
- A.12 10 minutes damage distribution at node 606, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO. . . 159
- A.13 10 minutes damage distribution at node 603, section 2 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO. . . 159
- A.14 10 minutes damage distribution at node 595, section 3 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO. . . 160
- A.15 10 minutes damage distribution at node 586, section 4 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO. . . 160
- A.16 10 minutes damage distribution at node 580, section 5 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO. . . 161
- A.17 10 minutes damage distribution at node 567, section 6 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO. . . 161

- 
- A.18 10 minutes damage distribution at node 549, section 7 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO. . . 162
- A.19 10 minutes damage distribution at node 531, section 8 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO. . . 162
- A.20 10 minutes damage distribution at node 516, section 9 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO. . . 163
- A.21 10 minutes damage distribution at node 506, section 10 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO. . . 163
- A.22 10 minutes damage distribution at node 606, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO. . . 164
- A.23 10 minutes damage distribution at node 606, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 165
- A.24 10 minutes damage distribution at node 603, section 2 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 165
- A.25 10 minutes damage distribution at node 595, section 3 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 166
- A.26 10 minutes damage distribution at node 586, section 4 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 166
- A.27 10 minutes damage distribution at node 580, section 5 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 167
- A.28 10 minutes damage distribution at node 567, section 6 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 167
- A.29 10 minutes damage distribution at node 549, section 7 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 168
- A.30 10 minutes damage distribution at node 531, section 8 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 168
- A.31 10 minutes damage distribution at node 516, section 9 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 169
- A.32 10 minutes damage distribution at node 506, section 10 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 169
- A.33 10 minutes damage distribution at node 606, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 170

- 
- A.34 10 minutes damage distribution at node 606, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . 171
- A.35 10 minutes damage distribution at node 603, section 2 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . 171
- A.36 10 minutes damage distribution at node 595, section 3 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . 172
- A.37 10 minutes damage distribution at node 586, section 4 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . 172
- A.38 10 minutes damage distribution at node 580, section 5 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . 173
- A.39 10 minutes damage distribution at node 567, section 6 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . 173
- A.40 10 minutes damage distribution at node 549, section 7 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . 174
- A.41 10 minutes damage distribution at node 531, section 8 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . 174
- A.42 10 minutes damage distribution at node 516, section 9 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . 175
- A.43 10 minutes damage distribution at node 506, section 10 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . 175
- B.1 20 years damage distribution at node 603, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 177
- B.2 20 years damage distribution at node 595, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 178
- B.3 20 years damage distribution at node 586, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 178
- B.4 20 years damage distribution at node 580, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 179
- B.5 20 years damage distribution at node 567, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 179
- B.6 20 years damage distribution at node 549, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 180

- 
- B.7 20 years damage distribution at node 531, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 180
- B.8 20 years damage distribution at node 516, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 181
- B.9 20 years damage distribution at node 506, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD. . . . . 181
- B.10 20 years damage distribution at node 603, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . . 182
- B.11 20 years damage distribution at node 595, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . . 182
- B.12 20 years damage distribution at node 586, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . . 183
- B.13 20 years damage distribution at node 580, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . . 183
- B.14 20 years damage distribution at node 567, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . . 184
- B.15 20 years damage distribution at node 549, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . . 184
- B.16 20 years damage distribution at node 531, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . . 185
- B.17 20 years damage distribution at node 516, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . . 185
- B.18 20 years damage distribution at node 506, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD. . . . . 186



# List of Tables

2.1	Parameters of the DTU 10 MW Reference Wind Turbine [Bak et al., 2013] . . .	15
2.2	Apparent mechanical properties of the multi-directional plies and core material of reference 10MW [Bak et al., 2013] and fatigue database composite material [Mandell and Samborsky, 2016]. . . . .	17
2.3	Composite material distribution in trailing panel region along the blade. . . . .	18
2.4	Composite material distribution in leading panel region along the blade. . . . .	19
2.5	Composite material distribution in shear webs region along the blade. . . . .	19
2.6	Comparison of natural frequencies between reference blade [Bak et al., 2013] and simplified 10MW. . . . .	22
2.7	Advantages and disadvantages of load application methods [Caous et al., 2018]	36
2.8	Comparison of blade tip displacement . . . . .	40
3.1	Hyper-parameter searching space. . . . .	60
3.2	Tuning hyperparameters using Random search and Bayesian Optimization varying the number of training epochs with a fixed number of <i>iteration</i> = 30. . . . .	60
3.3	Tuning hyperparameters using Random search and Bayesian optimization varying the number of iteration with a fixed training <i>epochs</i> = 100. . . . .	61
3.4	Regression comparison of a benchmark function. MSE represents the mean squared error and <i>r</i> <sup>2</sup> represents the coefficient of determination. . . . .	67
3.5	Comparison of state of arts regression methods to predict discontinuity in loads generated using aero-elastic simulation tools. . . . .	69
4.1	Lower and upper boundary for maximum steady aero-elastic loads for wind speed between <i>4m/s</i> and <i>25m/s</i> . . . . .	87
4.2	Lower and upper boundary for LAM PD input parameter. . . . .	88
5.1	Wind turbine surrogate model inputs parameters. . . . .	107
5.2	Example Range-Mean Counting . . . . .	116
5.3	Hyper-parameter searching space for DNN Damage. . . . .	126

5.4	MSE of DNNs Damage using LAM 4NO and LAM PD. . . . .	129
5.5	Average relative error for LAM 4NO 10 minutes damage prediction testing using a different seed number for generation of the 3D wind. . . . .	130
5.6	10 minutes damage prediction for a missing wind speed using LAM PD at blade root node for P2B and QQ1 layers at inferior level. . . . .	131
5.7	Wind speed proportion distribution over a measured year at la haute borne, France . . . . .	139
5.8	Kolmogorov-Smirnov goodness fit test for fatigue life estimation distribution of QQ1 node at $Z = 51.257$ . . . . .	144
5.9	Anderson-Darling test for fatigue life estimation distribution of QQ1 node at $Z = 51.257\text{m}$ . . . . .	144

# List of abbreviations and symbols

<b>4NO</b>	4 node physical load distribution
<b>ANN</b>	Artificial Neural Networks
<b>ADM</b>	Automatic Discontinuity Modeling
<b>ASTM</b>	American Society for Testing and Materials
<b>BECAS</b>	BEam Cross section Analysis Software
<b>BEM</b>	Blade Element Momentum
<b>BModes</b>	Software for Computing Rotating Beam Coupled Modes
<b>CDF</b>	Cumulative Distribution Function
<b>CFD</b>	Computational Fluid Dynamics
<b>CNN</b>	Convolutional Neural Network
<b>CLD</b>	Constant Life Diagram
<b>Cp-Lambda</b>	Code for Performance, Loads, AeroElasticity by Multi-Body Dynamic Analysis
<b>DEL</b>	Damage equivalent loads
<b>D</b>	Fatigue Damage per Half Cycle
<b>DLC</b>	Design Load Case
<b>DNN</b>	Deep Neural Network
<b>DOE</b>	U.S. Department of Energy
<b>DTU</b>	Denmark Technological University
<b>EE</b>	Elementary Effect
<b>ECDF</b>	Empirical Cumulative Density Function
<b>ELU</b>	Exponential Linear Unit
$E_1$	Longitudinal Young's Modulus
$E_2$	Transversal Young's Modulus
$E_3$	Normal Young's Modulus

<b>EM</b>	Expectation-Maximization Algorithm
<b>FAST</b>	Fatigue, Aerodynamics, Structure and Turbulence
<b>FEM</b>	Finite Element Model
<b>GP</b>	Gaussian Process
<b>GMM</b>	Gaussian Mixture Model
<b>HAWC2</b>	Horizontal Axis Wind turbine simulation Code 2nd generation
<b>HAWT</b>	Horizontal Axis Wind Turbine
<b>IEC</b>	International Electrotechnical Commission
<b>LAM</b>	Load Application Method
<b>LP</b>	Leading Panel
<b>LTU</b>	Linear Threshold Unit
<b>LR</b>	Linear Regression
<b>MCS</b>	Monte Carlo Simulation
<b>MSE</b>	Mean Squared Error
<b>B*</b>	Morris Trajectory Generation Matrix
<b>MLP</b>	Multi-Layer Perceptron
<b>NREL</b>	National Renewable Energy Laboratory
<b>NAG</b>	Nesterov Accelerated Gradient
<b>NTM</b>	Normal Turbulence Model
<b>PCE</b>	Polynomial Chaos Expansion
<b>PR</b>	Polynomial Regression
<b>PDF</b>	Probability Density Function
$P_f$	Probability of Failure
<b>PD</b>	Pressure Distribution Caous Method
<b>PSD</b>	Power Spectral Density
<b>RBE</b>	Rigid Body Element
<b>ReLU</b>	Rectified Linear Unit
<b>r2</b>	r2 Metric

---

<b>FL</b>	Resultant Force Aero-elastic Simulation
<b>ML</b>	Resultant Moment Aero-elastic Simulation
<b>RReLU</b>	Randomized Rectified Linear Unit
<b>PReLU</b>	Parametric Rectified Linear Unit
<b>RR</b>	Ridge Regression
<b>RBDO</b>	Reliability-Based Design Optimization
$u(t)$	Random time series signal
<b>SQP</b>	Sequential Quadratic Programming
<b>SNL</b>	Sandia National Laboratory
<b>SW</b>	Shear Web
<b>Spn</b>	Span Location along Blade Z-coordinate
<b>SVM</b>	Support Vector Machine
<b>SVR</b>	Epsilon-Support Vector Regression
<b>TP</b>	Trailing Panel
<b>VABS</b>	Variational Asymptotic Beam Section
<b>W</b>	Weight Vector
<b>X</b>	Input Parameter Vector



# Introduction

---

## 1.1 Context

In the last few decades, wind turbine machines have evolved, increasing their size, being more cost-effective, more eco-friendly, and more reliable. Currently, this energy generation technology has become trending in government policies worldwide, to reduce the use of fossil fuel power plants and decrease the greenhouse effect.

As a consequence of this massive growth, design-engineers encounter new challenges to continue the cost-effectiveness of turbines for all applications, at sites with low wind speeds or offshore farms, producing energy at a large amount. Also, there is a lack of operational experience because these wind turbines are brand new. Therefore, designers have to rely on simulation tools to model complex behavior and make decisions based on them, making these tasks of paramount importance.

Simulation tools as FAST developed by NREL (National Renewable Energy Laboratory) [Jonkman and Buhl Jr, 2005] and HAWC2 developed by DTU (Denmark Technological University) [Larsen and Hansen, 2007] reproduce the coupled dynamic response of wind turbines. Joining models for aerodynamic, electrical and control, and hydrodynamic using simplified Blade Element Momentum (BEM) theory. This wind turbine model calculates the non-linear time-domain response of displacements and loads acting in the machine under different environmental conditions testing all possible cases described in certification standards for wind turbines [Guideline and Lloyd, 2010]. As these tools allow us to simulate numerous conditions in a considered time, a general analysis for wind turbine blades could be performed, obtaining the deflection, natural modes, and extreme load analysis. However, a detailed analysis of the blade as fatigue or buckling analysis needs a tri-dimensional model and distributed loads acting along the blade. Computational Fluid Dynamics (CFD) codes calculate the interaction of fluid-structure in a 2D or 3D space. However, there will not be a coupled dynamic response, also depending on the mesh conditions, it costs more CPU calculation time. To consider the coupled response of the wind turbine blade is required to use the multi-body outputs, transfer them as an equivalent distributed load field to the 2D shell finite element model (FEM), to obtain a stress distribution along the blade.

As different authors [Bottasso et al., 2014, Hu et al., 2016b] define their method to transfer loads and calculate the fatigue life or do a reliability-based design optimization (RBDO), the stress distribution and displacement responses on the blade varies from one method to another,

creating uncertainty by the type of method used. Selecting a method to estimate the fatigue life or perform a reliability analysis of the wind turbine blade's fatigue failure is crucial. Because depending on how is replicated the coupled multi-body dynamic simulation on a 2D shell FEM of the blade, increasing the complexity, and methods based on gradient would have problems to converge, adding uncertainty to the evaluated model. Moreover, blades shell FEM return distribution of stress and displacement all over the blade, and treating all these outputs is a time-consuming task for reliability analysis. In this case, authors [Morató et al., 2019, Teixeira et al., 2017] have used the Kriging (Gaussian process) surrogate model to replace a high time-consuming model and being able to predict new outputs without the need of running the original model, making the estimation of fatigue life, reliability, and fatigue analysis more doable in terms of calls to the original model. Also, with the increasing development of new machine learning and deep learning methods to create a surrogate model, different strategies and methodologies could be explored to solve these problems with a faster and more accurate solution.

Fatigue analysis of wind turbine blades is a broad field, requiring an approach of multiple disciplines due to the combination of components with mechanical and aerodynamic functions, and a corresponding variety of material and load conditions. Therefore, with a limited time frame project, the present thesis will focus on narrowing the study's scope to the probabilistic fatigue life estimation of composite wind turbine blades using surrogate models.

## 1.2 Previous work

Different methodologies to analyze the fatigue damage on wind turbines have been approached differently by some authors, with the most difference in applying the loads along the wind turbine blade and the failure criteria applied. This state of the art presents only previous works in the last decade. First, a structural reliability study and evaluation of wind turbine blades made by Dimitrov [Dimitrov, 2013] is shown in figure 1.1.

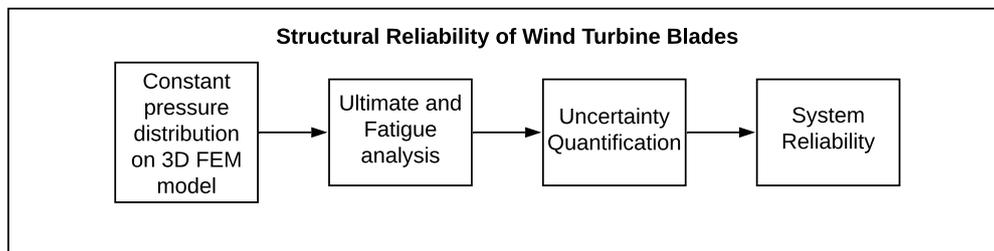


Figure 1.1: Methodology for analyzing the structural reliability of wind turbine blades [Dimitrov, 2013].

In this study, Dimitrov defined a constant uniform pressure distribution as a linear function of the flap-wise bending moment and applied to the 3D FEM of a composite material wind turbine blade. Then, using the Hashin criterion [Hashin, 1980] calculates the ultimate failure of composite laminate, which involves the computation of four failure indices describing the material conditions in fiber and matrix tension and compression. Moreover, the author used a progressive failure analysis, following the development of matrix-related failure events, until the first fiber failure occurs. In fatigue failure for laminates is used as a fatigue analog to the static Hashin failure criterion [Hashin, 1981] with a progressive update of stiffness in the laminates. Subsequently, treatment of uncertainty in different assessments as uncertainties in wind climate variables in estimating extreme and fatigue loads, in the ultimate limit state, and fatigue limit state. Finally, performing a reliability analysis for blade-tower collision and a sandwich panel subjected to compression and traverse pressure. As the loads are applied using a constant uniform pressure distribution, this force's value is proportional to the surface area. Its direction is not necessarily perpendicular to the element surface. However, it is oriented in a specific direction, meaning that flap-wise and edge-wise behavior are studied separately [Caous et al., 2018]. On the other hand, the main idea of the Hashin criterion is the failure mechanism of the composite, making differences between failures of the fiber and matrix both in tension and compression. However, for the matrix failure, the author proposes a quadratic criterion because a linear criterion underestimates. The material's strength and a polynomial of higher degree would be too complicated to manage in his experience. This criterion was developed based on logical reasoning to reach an applicable criterion, rather than to continue with the mechanism of failure to establish the macro-variables associated with it and propose a criterion based on them [París and Jackson, 2001].

Another study for the structural blade optimization was performed by [Bottasso et al., 2014] as shown in figure 1.2. Divided into two main parts, first, a 2D FEM section and beam modeling are optimized, and then a 3D FEM is used to update the constraints of the 2D FEM optimization. In the first part, to analyze the wind turbine is necessary to define the sectional design of the cross-section (airfoil) using code HANBA 2D FEM [Giavotto et al., 1983] to compute the  $6 \times 6$  stiffness matrices, to create a geometrically exact beam model. Subsequently, is defined as a complete HAWT (Horizontal Axis Wind Turbine) Cp-Lambda (Code for Performance, Loads, AeroElasticity by Multi-Body Dynamic Analysis) multi-body model [Bauchau et al., 2001, Bottasso, 2006] performing different DLCs (Design Load Conditions) simulations and extracting: load envelope, DELs (Damage equivalent loads), and max tip deflection.

Then, is minimized the cost (mass) of the wind turbine blade using the sequential quadratic programming (SQP) method until convergence. Subsequently, it generates a 3D CAD model using the optimized parameters to perform a detailed analysis: max tip deflection, max stress/strain, fatigue, and buckling; founding a violation of the 3D FEM constraint, is updated and relaunched the 2D optimization to Minimize both analyses under the defined constraints.

For the 3D FEM analysis, Bottasso has developed an approach to apply equivalent loads from the beam multi-body model analysis to the 3D FEM blade by recovering the inertial and aerodynamics resultants loads applied to all sectional nodes and the sole skin nodes, re-

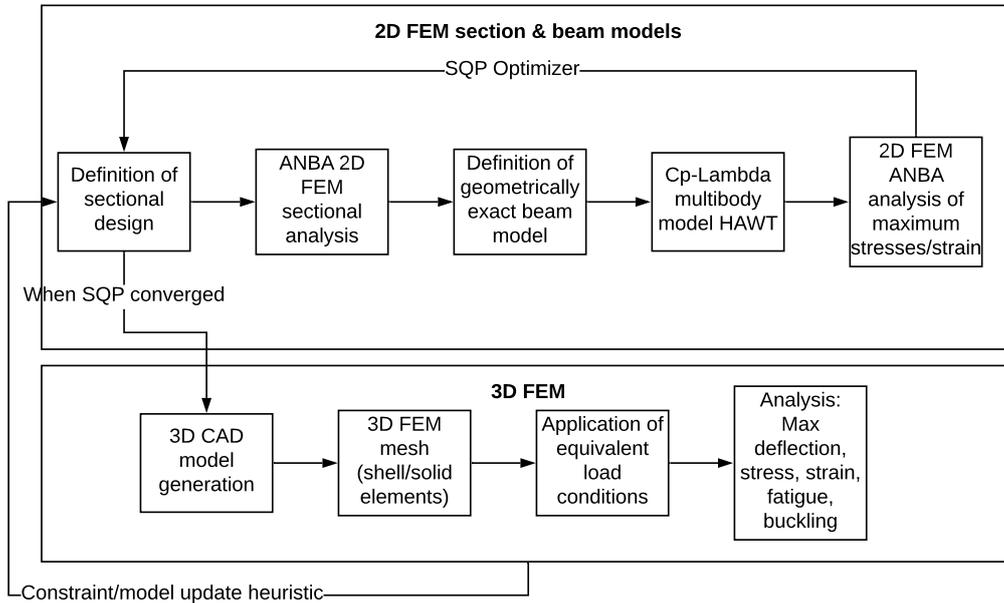


Figure 1.2: Methodology for structural blade optimization [Bottasso et al., 2014]

spectively. It computes the fatigue analysis by exploiting the linear superposition of static unit load cases applied to the FEM with load histories obtained from the beam model, using rain-flow counting and the associated Markov matrices. As mentioned by the author, a more realistic way of representing aerodynamics loads would be to reconstruct the chord-wise pressure distribution, using experimental measurements or numerical models such as Xfoil [Drela, 1989] that can calculate the pressure distribution on the airfoil. However, Xfoil could not accurately predict the stall because of extensive flow separation and possibly unsteady effects [Petrone et al., 2011].

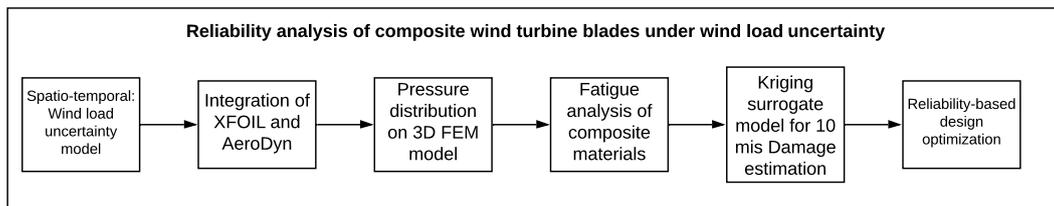


Figure 1.3: Methodology for reliability-based optimization of wind turbine blade [Hu et al., 2016a]

A study for reliability-based optimization in wind turbine blades for fatigue life was made by Hu [Hu et al., 2016a], as shown in figure 1.3. A wind load uncertainty model representing realistic spatiotemporal annual wind load variation represents a joint probability density function (PDF) of 10-minutes mean wind speed and 10-minutes turbulence intensity. That

estimate the lift, drag, and moment coefficients integrating Xfoil and aerodynamic code AeroDyn [Moriarty and Hansen, 2005] for a different angle of attack for each blade section. Then, a pressure distribution along the 3D FEM model of the blade is applied to calculate the fatigue damage using a multi-axial fatigue damage model for isotropic and anisotropic materials [Liu and Mahadevan, 2007]. Due to the high amount of variables on his fatigue damage model, the author used a Kriging surrogate model [Zhao et al., 2011] on defined hotspots, to then estimate the reliability-based optimization of the wind turbine blade. The correction made by the author, concerning lift, drag, and aerodynamic moment coefficients may differ from final forces once applying the pressure because of blade geometry variations. Segments of the blades defined in AeroDyn are beam FEM lead to a simplified geometry, which may differ from the blade modeled with finite shell elements. These geometries differ in terms of local chord length or twist, or terms of global dimension variation through the length, leading to a radial force when pressure is applied [Caous et al., 2018]. Also, as specified by the author, the method does not consider the aerodynamic force coefficient variation at one angle of attack. Instead is used a fifth-order polynomial regression model.

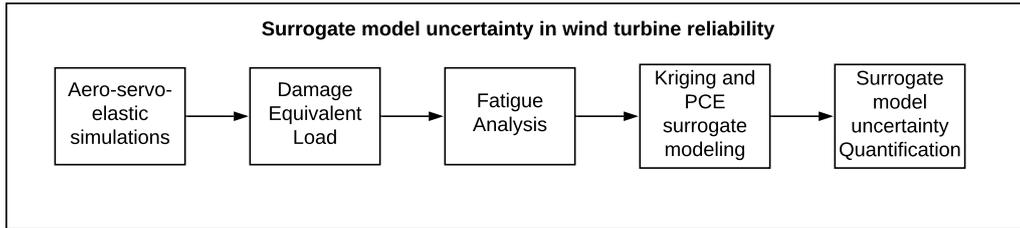


Figure 1.4: Methodology for surrogate model uncertainty in wind turbine reliability assessment.

A more recent study about surrogate models in wind turbine reliability assessment [Slot et al., 2020] is presented in figure 1.4. In this study, the authors simulate the wind turbine using FAST (Fatigue, Aerodynamics, Structure, and Turbulence) code [Jonkman and Buhl Jr, 2005] from National Renewable Energy Laboratory (NREL) to obtain resultant loads in a beam FEM of the wind turbine. Subsequently, Uses DELs implicitly assuming that a linear S-N curve can model fatigue strength of materials. The Miner's rule [Miner, 1945] is used to accumulate fatigue damage. Subsequently, a surrogate model using two different approaches are used to predict the damage at different location of the blade, Kriging [Santner et al., 2003] and PCE (Polynomial Chaos Expansion) [Sudret, 2014]. Then, making quantification of the surrogate model's uncertainty for different conditions on the wind turbine blade outputs. Modeling the wind turbine blade using beam FEM, the fatigue damage analysis is faster than using a 3D FEM. However, there is no detailed information on stress distribution along with the wind turbine, especially the blade, which has the most complex geometry, and estimating this distribution is difficult to calculate.

### 1.3 Proposed study

The first objective of this study is to propose a methodology for probabilistic life estimation based on fatigue damage, which includes: wind field simulation, coupled aerodynamic, elastic, and electrical and control (servo) analysis, filter aero-elastic loads, stress analysis by finite element analysis, and fatigue damage evaluation based on deep neural network surrogate model considering a multi-axial non-proportional stress state. Using the proposed methodology, a 10 minutes fatigue damage is determined by a 10 minutes aero-elastic simulation assuming a normal turbulence model for the environmental conditions. Different load application methods for the 2D shell finite element model are specially studied and compared in a stress and fatigue analysis in this procedure. A methodology diagram is illustrated in figure 1.5.

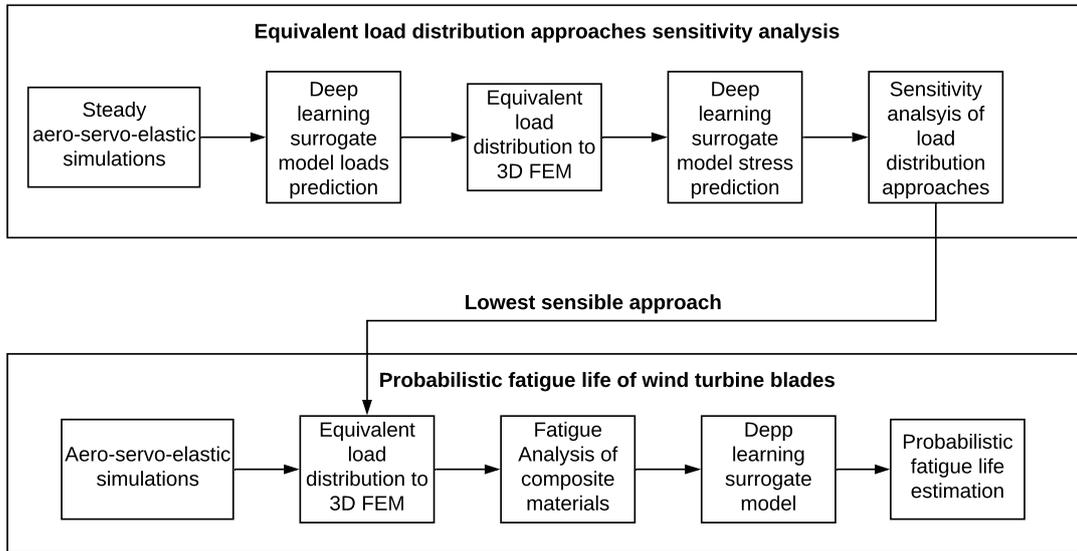


Figure 1.5: Proposed methodology for probabilistic fatigue damage estimation of a composite wind turbine blade.

The second objective is to propose a methodology for sensitivity analysis for high dimension models based on the neural network surrogate model to select the load application methods that generate the least sensitivity, selecting this method as the primary approach for the fatigue life estimation. Three load application methods are selected to compare their responses. Each of them has a different approach to applying the aero-elastic loads to a 2D shell model from simple to realistic pressure distribution across the whole blade. This study proposes to estimate the Sobol sensitivity indexes, a two-stages surrogate modeling approach, one to estimate the maximum aero-elastic loads in steady environmental conditions and another one to predict the Von Mises stress at the blade root. This proposed methodology applies to other wind turbine components as a tower and jacket.

The third objective is to propose an algorithm for modeling jump discontinuity using deep

neural networks. This model the variables found to have jump discontinuities in their domain. The proposed methodology presents an improvement in mean squared error and R2 metrics using a benchmark problem and aero-elastic loads for sensitivity analysis.

## 1.4 Outline of the thesis

This thesis follows the structure presentd bellow:

Chapter 2 presents the coupled aero-servo-elastic simulations used for a composite wind turbine blade, including all the specifications in material distribution across the blade, calculation of cross-section stiffness matrices, and natural frequencies modes. Moreover, it presents a state of the art of the load application methods used to transfer the loads from the simulation tools to a blade 2D shell finite element model.

Chapter 3 introduces the reader briefly to deep neural networks (DNN) and how to optimize their hyperparameters. DNN is the primary surrogate model for this thesis for sensitivity and fatigue analysis. This study develops a new methodology to deal with jump discontinuities by using neural networks, classifying the zones with a continuous response using a Gaussian Mixture Model, and training the network by adding the classification value as an input parameter.

Chapter 4 proposes a new methodology for sensitivity analysis of load application methods used in the blade 2D shell finite element model. This study proposes two surrogate models to predict the maximum aero-elastic simulated loads in steady environmental conditions used as input parameters with the material thicknesses for the second surrogate model. The aero-elastic loads are applied using three different load application methods in the 2D shell model to calculate the Von Mises stress on the blade root. Another surrogate model replaces the 2D shell model to predict the Von Mises stress and is used to calculate each load application method's sensitivity using the Sobol method. This study compares all sensitivity analyses to determine the least sensitive method. A Morris analysis is used in the 2D shell model to validate the Sobol indexes' results using neural networks.

Chapter 5 proposes a methodology to estimate the fatigue damage based on 10 minutes simulation mean wind speed. 10-minutes unsteady simulation assuming a normal turbulence model is performed coupling aerodynamic, elastic, and electrical and control (servo) behavior of the wind turbine to calculate the resultant loads on a beam finite element model. Loads are transferred to a 2D shell FEM using the least and most sensible load application method to estimate the blade's detailed stress time history. Then uses a multi-axial criterion for fatigue analysis to take into account the non-proportionality of the stress state. Constant life diagrams estimate the number of cycles to failure at different stress amplitude and mean stress. Miner's rule calculates the accumulated fatigue damage. A surrogate model based on a deep neural network is used to predict the fatigue damage cycles per wind speed by adding the empirical cumulative distribution function of each damage distribution as input parameters. The network hyperparameters are optimized using an optimization algorithm based on a

Gaussian process. The network is validated and tested to predict a new damage distribution using a different wind field time history. Next, analyzing a blade damage distribution at the hotspot where the damage is maximum due to the load application method strategy. A 20 years fatigue damage estimation is performed for different composite material layers at a selected hotspot. It then performs a Monte Carlo Simulation, considering surrogate models to estimate the fatigue life probability distribution.

Chapter 6 summarizes the most important conclusions from the study and suggests topics for future research.

# Load application methods from beam to shell FEM

---

## Sommaire

<b>2.1</b>	<b>Simulation codes for wind turbine</b> . . . . .	<b>10</b>
2.1.1	Pre-processor tools for simulation codes . . . . .	10
2.1.2	Wind turbine blade model . . . . .	14
<b>2.2</b>	<b>Load Application Methods - LAM</b> . . . . .	<b>23</b>
<b>2.3</b>	<b>Classification</b> . . . . .	<b>24</b>
2.3.1	Application of loads by sections and on one point of each section . . . . .	26
2.3.2	Application of loads by sections but physical distribution on sections . . . . .	27
2.3.3	Constant pressure application on the blade of an oriented surface load . . . . .	29
2.3.4	Dissociation of inertial and aerodynamic loads with the application of an acceleration field and pressure distribution across the whole blade . . . . .	30
<b>2.4</b>	<b>Application of LAM to a composite wind turbine blade</b> . . . . .	<b>37</b>
2.4.1	Blade stress distribution . . . . .	37
<b>2.5</b>	<b>Conclusions</b> . . . . .	<b>41</b>

---

To analyze wind turbine blade for different environmental design cases computational tools estimate the forces and displacement of the blade. This response considers the aerodynamics, elastic effect and the controller action on the blade. To use this tools is needed to define the wind turbine characteristics for mechanics properties, geometry, environmental conditions. This tools then, create a 1D beam finite element model of the blade to simulate this coupled behavior in a reduced computational time. This chapter presents the simulation codes and loads application methods used to analyze wind turbine blades. Section 2.1 presents the aero-elastic simulation codes and pre-processor tools to model a wind turbine. This study uses a wind turbine model as a reference and explains the modification in the composite material distribution along the blade. An explanation and classification of the called load application methods (LAM) are presented in Sections 2.2 and 2.3. Section 2.4 presents an application of different load application methods in a composite wind turbine blade. As found in the literature, these approaches applies the loads from the 1D aero-elastic simulations to a 2D shell model, and depending their strategy it could be more realistic and also more time consuming.

## 2.1 Simulation codes for wind turbine

Wind turbine certification standards for design [Guideline and Lloyd, 2010] suggest different design situation as power production, start-up, normal shut-down, emergency shut-down, where different design load cases (DLC) could occur during the wind turbine life. Each DLC specifies the type of analysis needed to perform, fatigue, or ultimate analysis. However, as environmental conditions have uncertainties, the wind turbine must be analyzed at different combinations of wind speed, intensity turbulence, and shear exponent, resulting in an extensive, time-consuming task for design. Numerical simulation tools deal with this issue, assuming a simplified representation of the wind turbine and coupling multi-physics behavior.

Some simulation codes run numerous simulations to evaluate DLCs in the wind turbine. HAWC2 [Larsen and Hansen, 2007] (Horizontal Axis Wind turbine Code 2nd generation) developed by DTU (Denmark Technological University) Wind Energy, is an aero-elastic code intended for calculating wind turbine response in the time domain. Cp-Lambda [Bottasso, 2006] (Code for Performance, Loads, Aero-Elastic by Multi-Body Dynamics Analysis). FAST [Jonkman and Buhl Jr, 2005] (Fatigue, Aerodynamics, Structure, and Turbulence) code developed by NREL (National Renewable Energy Laboratory) is a multi-fidelity tool for simulating the coupled dynamic response of wind turbines. Coupling computational modules for aerodynamics, hydrodynamics, control, electrical system (servo) dynamics, and structural dynamics enable coupled nonlinear simulation in the time domain. All of these simulation codes have implemented the blade element momentum theory. However, the main difference appears in how the multi-body formulation models the wind turbine; some use a finite element implementation of the Timoshenko beam theory. Others make use of a response shape formulation. The last one linearizes the response of the individual turbine elements (e.g., blades and tower), excluding a subset of degrees of freedom that may be important for a large and flexible wind turbine [GL, 2015]. In this study, the FAST simulation code analyzes the deflection and loads (forces and moments) acting on the wind turbine blade.

### 2.1.1 Pre-processor tools for simulation codes

Before executing the simulation tool FAST to calculate the loads and displacement time history, it is necessary to generate some input files that describe the wind conditions to analyze and generates the properties that describe the wind turbine as a beam FEM.

#### 2.1.1.1 Wind environmental conditions

Wind speed variation in time occurs in different scales, variation in location, and wind direction follows. Inter-annual variation in wind speed occur over time scales greater than one year. The annual variation shows the variation in seasonal or monthly averaged wind speed. Diurnal variation occurs in tropical and temperate latitudes due to differential heating of the earth's surface during the daily radiation cycle. Short-term variations over intervals of ten minutes or less

that have a stochastic character represent turbulence in the wind speed. The fatigue damage of wind turbines most frequently considers short-term variations intervals of 10-minutes averaged wind speed [Griffith and Ashwill, 2011, Hu et al., 2016a, Dimitrov, 2013, Slot et al., 2020]. In this study, a 10-minutes simulation at an averaged wind speed calculates the fatigue damage on the wind turbine blade.

To generate a 10-minutes wind field is based on 10-minutes mean wind speed  $V_{10}$ , 10-minutes turbulence intensity  $I_{10}$ , and shear exponent  $\alpha$ . The 10-minutes turbulence intensity is the ratio of the wind speed standard deviation to the mean wind speed, determined from the same set of measured data samples of wind speed, and taken over a specified time [Standard et al., 2005] as

$$I_{10} = \frac{\Sigma_{10}}{V_{10}} \quad (2.1)$$

Where  $\Sigma_{10}$  is the standard deviation of the mean wind speed.

To calculate the 10-minutes standard deviation there is a dependency with the  $V_{10}$  described in standard [Standard et al., 2005]. For normal turbulence model, this dependency is given by the local statistical moments of  $\sigma_1$  as:  $\mathbf{E}(\sigma_1|WS) = TI_{ref}(0.75WS + 3.8)$  and  $\mathbf{V}(\sigma_1|WS) = (1.4TI_{ref})^2$ . The parameters of the  $\sigma_1$  density probability distribution are given in the following equations as a function of  $WS$  and  $I_{ref}$  is the expected value of the turbulence intensity at  $15m/s$  depending on wind turbine class [Standard et al., 2005].

$$\sigma_{\sigma_1} = \left[ \ln \left( \frac{\mathbf{V}(\sigma_1|WS)}{\mathbf{E}^2(\sigma_1|WS)} + 1 \right) \right]^{1/2} \quad (2.2)$$

$$\mu_{\sigma_1} = \ln[\mathbf{E}(\sigma_1|WS)] - \frac{\sigma_{\sigma_1}^2}{2} \quad (2.3)$$

The shear exponent  $\alpha$  or power-law exponent express the assumed wind speed variation with height above ground, taking into account topography and roughness [Standard et al., 2005]. Meaning that knowing the profile at different mean wind speeds is necessary to know the type of wind field that analyzes the wind turbine, which is not always available. A simplified correlation between  $V_{10}$  and mean shear exponent  $\alpha$  based on a joint distribution defined by Dimitrov et al. [Dimitrov et al., 2015] as

$$\mu_{\alpha} = 0.088(\ln(V_{10}) - 1) \quad (2.4)$$

$$\sigma_{\alpha} = (1/V_{10}) \quad (2.5)$$

Subsequently, this procedure utilizes stochastic, full-field, turbulent wind simulations named Turbsim [Jonkman and Buhl Jr, 2006]. It numerically simulates time series of 3-dimensional

wind velocity vectors at points in a regular vertical grid. The theory behind this tool, based on Veer's method [Veers, 1988], generates a 3-dimensional field from a power spectral density (PSD) function and the coherence function. Veer's method uses a general random process simulation method developed by Shinozuka and Jan [Shinozuka and Jan, 1972]. To generate the random time-series signal using the following equation:

$$u(t) = \bar{u} + \sum_{j=1}^{M-1} A_j \cos(2\pi n_j t + \phi_j) \quad (2.6)$$

$$A_j = \sqrt{2} l_j \quad (2.7)$$

$$l_j = \sqrt{\left[ \frac{S(n_j) + S(n_{j+1})}{2} \right] (n_{j+1} - n_j)} \quad (2.8)$$

A frequency interval has to be selected from the PSD between  $n_1$  to  $n_M$  and then discretize in  $M - 1$  sub-intervals, each frequency  $n_j$  correspond to a value  $S_j$  from the spectral density function,  $\bar{u}$  is the average wind speed and  $\phi_j$  is a random variable uniformly distributed between 0 and  $2\pi$ . For each node of the vertical grid and at every time instant observed, it will consider the contribution of every frequency selected between  $n_1$  and  $n_M$ .

As mentioned before, a spectral density function is needed to complete the random generation process. In Turbsim [Jonkman and Buhl Jr, 2006], there are several different models available listed below:

- The IEC Spectral Models [Standard et al., 2005]: The IEC Kaimal Model is defined as:

$$\frac{f S_k(f)}{\sigma_k^2} = \frac{\frac{4f L_k}{V_{hub}}}{\left(1 + \frac{6f L_k}{V_{hub}}\right)^{5/3}}, \quad (k = u, v, w \text{ and } f > 0) \quad (2.9)$$

where  $f$  is the frequency in Hertz;  $k$  is the index referring to the velocity component direction (i.e  $u$  = longitudinal,  $v$  = lateral and  $w$  = upward);  $S_k$  is the single-sided velocity component spectrum;  $V_{hub}$  is the mean wind speed at hub height;  $L_k$  is the velocity component integral scale parameter, determined by

$$L_k = \begin{cases} 8.1\Lambda_1, & k = u \\ 2.7\Lambda_1, & k = v \\ 0.66\Lambda_1, & k = w \end{cases} \quad (2.10)$$

where  $\Lambda_1$  is the longitudinal scale parameter at hub height given by  $0.7z$   $z \leq 60m$  or  $42m$   $z \geq 60m$ .  $\sigma_k$  is the velocity component standard deviation given by

$$\sigma_k = \begin{cases} \sigma_u, & k = u \\ 0.8\sigma_u, & k = v \\ 0.5\sigma_u, & k = w \end{cases} \quad (2.11)$$

where  $\sigma_u$  is calculated used equation 2.1.

The IEC Von Karman isotropic model [Standard et al., 2005]:

$$\begin{cases} S_u(f) = \frac{2L_u\sigma_u^2}{\pi(1+(1.339L_u f)^2)^{5/6}} \\ S_k(f) = \frac{2L_k\sigma_k(1+8/3(2.678L_k f)^2)}{\pi(1+(2.678L_k f)^2)^{11/6}}, k = v, w \end{cases} \quad (2.12)$$

both spectral models are defined in the IEC standard, assuming neutral atmospheric conditions.

- The RisøSmooth-Terrain Model (SMOOTH) based on work by Højstrup et al. [Højstrup, 1982, Olesen et al., 1984], this spectral model shows more flexibility in the atmospheric conditions than the IEC spectral models. This spectral model has separate equations for stable/neutral and unstable flows. The SMOOTH model defines the velocity spectra using local height and wind speed; this contrasts with the IEC models, which use the hub's wind speed and height to define the spectra at all points.

For stable and neutral conditions (Gradient Richardson Number  $\geq 0$ ), the SMOOTH model velocity spectra for the three wind components,  $K$ , are given by

$$S_K(f) = VStar^2 \frac{S_{1,K}(\frac{z}{\bar{v}\phi_M})(\frac{\phi_E}{\phi_M})^{2/3}}{1.0 + S_{2,K}(\frac{fz}{\bar{v}\phi_M})^{5/3}} \quad (2.13)$$

$VStar$  is the friction velocity input parameter,  $\bar{v}$  is the mean wind speed at height  $z$ , and  $\phi_E$  and  $\phi_M$  are functions of the stability parameter.

- The NREL National Wind Technology Center Model [Kelley et al., 2002], for neutral and stable flows, the spectra models are defined by adding scaled versions of the SMOOTH-model spectra:

$$S_K(f) = \sum_{i=1}^{NumPeaks_K} p_{i,K} S_{K,SMOOTH}(F_{i,K}f) \quad (2.14)$$

where  $NumPeaks_K = 2$  for all wind components  $K$ . All of the  $p_{i,K}$  and  $F_{i,K}$  scaling factors are functions of gradient Richardson number.

- The NREL Great Plains Model Low-Level Jet Model [Kelley et al., 2004], defines vertical profiles of stability and of shear velocity. For stable and neutral flows, the spectra are defined by adding peaks from the form of the SMOOTH-model spectra:

$$S_K(f) = \frac{v_*^2}{VStar^2} \sum_{i=1}^{NumPeaks_K} p_{i,K} S_{K,SMOOTH}(F_{i,K}f) \quad (2.15)$$

Using the local stability parameter,  $z/L$ , to determine the values of functions  $\phi_E$  and  $\phi_M$  (instead of using the Richardson number as the SMOOTH model does).

Also a coherence function defined by the complex magnitude of the cross-spectral density of the longitudinal wind velocity components at two spatially separated points divided by the autospectrum function needs to be selected. The most commonly used spatial coherence models are:

- Coherence for IEC Spectral Models: The root coherence function for the u-component of the IEC spectral model is defined as

$$Coh_{j,k} = \exp\left(-b\sqrt{\left(\frac{fr}{V_{hub}}\right)^2 + \left(0.12\frac{r}{L_c}\right)^2}\right) \quad (2.16)$$

where  $f$  is the frequency,  $r$  is the distance between points  $j$  and  $k$ ,  $b$  is the coherence decrement (either 12 or 8.8, depending on the IEC 61400-1 edition number), and  $L_c$  is the IEC 61400-1 length scale.

The coherence function for the v- and w-components of the IEC spectral models is

$$Coh_{j,k} = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases} \quad (2.17)$$

- Coherence for non-IEC spectral models: The root coherence function for all three of the wind components for non-IEC spectral models is:

$$Coh_{j,k} = \exp\left(-0.5b\left(\frac{r}{z_m}\right)^a \frac{fr}{V_m}\right) \quad (2.18)$$

where  $b$  is the input coherence decrement (it may be different for each wind component),  $r$  is the distance between points  $j$  and  $k$ ,  $z_m$  is the mean height of the two points,  $a$  is the input coherence exponent, and  $V_m$  is the mean wind speed of the two points.

After selecting the spectral model and the spatial coherence model, Turbsim generates an output binary file with the full-field wind speed that will be interpreted by FAST to calculate the loads and displacements in the wind turbine blade. The IEC spectral model Kaimal and the IEC spatial coherence model are the primary models in this study.

### 2.1.2 Wind turbine blade model

The DTU 10MW reference wind turbine blade [Bak et al., 2013] is the model used in this study. The design of this wind turbine was an upscaling of the NREL 5MW reference turbine [Jonkman et al., 2009]. Table 2.1 shows the critical parameters of this wind turbine.

The DTU Wind Energy controller, as described by Hansen [Hansen and Henriksen, 2013], was implemented as a DLL to interact with the FAST model. The controller enables both partial and full load operation, with switching mechanisms that streamline the transition

Table 2.1: Parameters of the DTU 10 MW Reference Wind Turbine [Bak et al., 2013]

Parameter	DTU 10MW RWT
Wind Regime	IEC Class 1A
Rotor Orientation	Clockwise rotation - Upwind
Control	Variable Speed, Collective Pitch
Cut in wind speed	4 m/s
Cut out wind speed	25 m/s
Rated wind speed	11.4 m/s
Rated power	10 MW
Number of blades	3
Rotor Diameter	178.3 m
Hub Diameter	5.6 m
Hub Height	119.0 m
Drivetrain	Medium Speed, Multiple-Stage Gearbox
Minimum Rotor Speed	6.0 rpm
Maximum Rotor Speed	9.6 rpm
Maximum Generator Speed	480.0 rpm
Gearbox Ratio	50
Maximum Tip Speed	90.0 m/s
Hub Overhang	7.1 m
Shaft Tilt Angle	5.0 deg
Rotor Precone Angle	2.5 deg
Blade Prebend	3.332 m
Rotor Mass	227,962 kg
Nacelle Mass	446,036 kg
Tower Mass	628,442 kg

between the two modes of operation. The controller uses the collective blade pitch angle and electro-magnetic generator torque to control the wind turbine based on proportional-integral control with additional filters. However, no tuning modifications were carried out, and the present implementation considers the same controller as that in HAWC2 [GL, 2015].

Figure 2.1 presents a uniform wind ramp simulation generated using IECwind pre-processing tool from FAST, showing the behavior of the wind turbine controller. The wind speed starts at  $4\text{ m/s}$  and increases  $2\text{ m/s}$  every  $100\text{ s}$  until it reaches the cut-out wind speed. The wind turbine starts generating a rated power of  $10\text{ MW}$  after the wind speed is greater than the rated wind speed. Also, the pitch control is adjusted to maintain this rated power as the wind speed increases. Although the controller for this wind turbine blade is out of tune, it works well with perturbation in the system.

The composite material distribution used on this wind turbine blade is glass fiber reinforced composites and balsa wood, which serves as a sandwich core material, assembling the composite layup in terms of stacking-sequence of layers representing multi-directional plies.

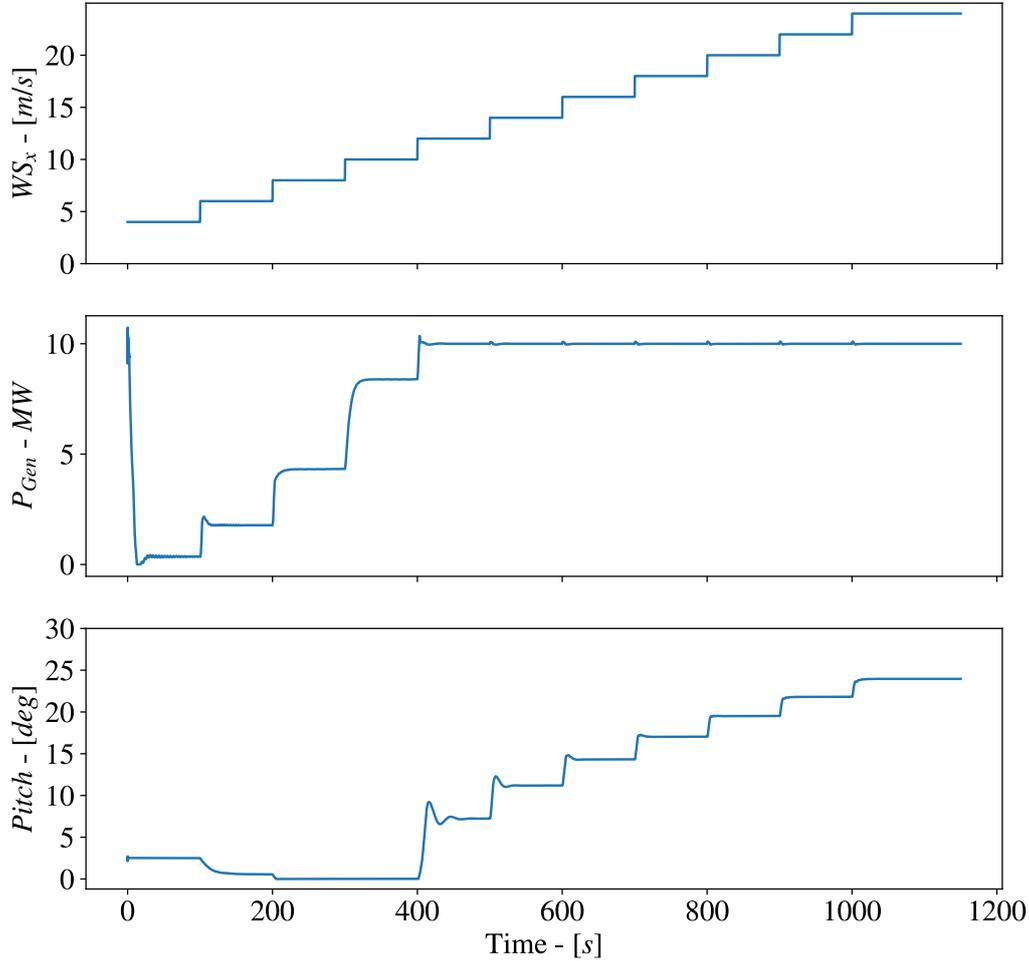


Figure 2.1: Wind ramp simulation results for the 10 MW DTU simplified wind turbine blade using FAST.

Their apparent material properties were derived based on the constituent materials' specific properties using simple micro-mechanics equations from [Chamis, 1983]. The resultant properties for Uniax, Biax, Triax, and Balsa wood are summarized in table 2.2. The composite layup divisions for the blade are 11 regions circumstantially 2.2, and the stacking sequence varies in thickness along the blade length resulting in complex composite material distribution, with a high dimensional thicknesses space. To simplify this complex distribution to estimate the probabilistic fatigue life, the number of variables thicknesses where reduced.

Moreover, this study uses a new material composite for the blade, replacing Uniax, Biax, and Triax by composite materials from DOE/MSU fatigue database [Mandell and Samborsky, 2016].

These materials are QQ1, a glass fiber reinforced epoxy composed by Vantico TDT 177-155 epoxy resin, Saertex U14EU920-00940-T1300-100000 0's and VU-90079-00830-01270-000000 45's, and P2B a carbon/glass-hybrid-fiber-reinforced epoxy laminate composed by Newport carbon NB307-D1-34-600 G300 prepreg 0° and glass NB307-D1-7781-497A [Mandell and Samborsky, 2016]. Also, retaining the balsa wood as the core material in the sandwich panel. These new materials were selected because of the existing experimental fatigue data at different stress ratios, that the original composite materials used for DTU do not have.

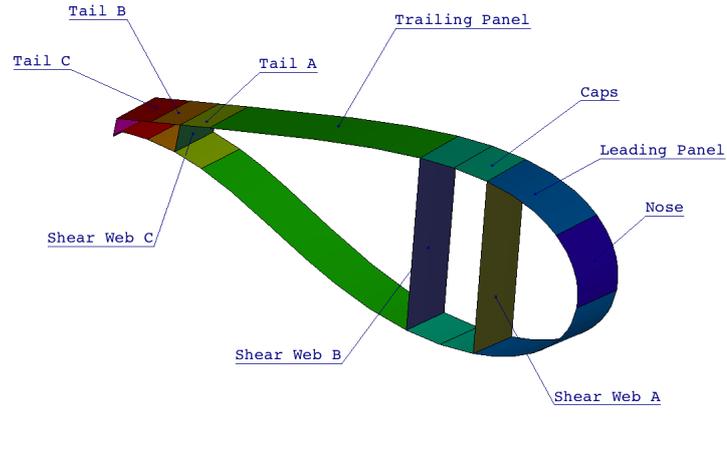


Figure 2.2: Circumferential region division of the wind turbine blade for the composite material distribution [Bak et al., 2013]

Table 2.2: Apparent mechanical properties of the multi-directional plies and core material of reference 10MW [Bak et al., 2013] and fatigue database composite material [Mandell and Samborsky, 2016].

Property	Uniax	Biax	Triax	Balsa	QQ1	P2B	Unit
Young's modulus $E_1$	41,63	13,92	21,79	0,05	33,1	101	Gpa
Young's modulus $E_2$	14,93	13,92	14,67	0,05	17,1	8,86	Gpa
Young's modulus $E_3$	14,93	13,92	14,67	2,73	17,1	8,86	Gpa
Shear modulus $G_{12}$	5,047	11,5	9,413	0,01667	6,29	6,37	Gpa
Shear modulus $G_{13}$	5,04698	4,53864	4,53864	0,15	6,29	6,37	Gpa
Shear modulus $G_{23}$	5,04698	4,53864	4,53864	0,15	6,29	6,37	Gpa
Poisson's ratio $\nu_{12}$	0,241	0,533	0,453864	0,5	0,27	0,22	-
Poisson's ratio $\nu_{13}$	0,241	0,28	0,28	0,013	0,359	0,31	-
Poisson's ration $\nu_{23}$	0,241	0,28	0,28	0,013	0,27	0,22	-
Mass density $\rho$	1915,5	1845	1845	110	1919	1570	$kg/m^3$

The new layup configuration used in this study keeps the same stacking-sequence of

layers as [Bak et al., 2013]. For all regions in the blade's surface, the stacking sequence is  $[P2B, QQ1, BALSAL, QQ1, P2B]$  and for internal regions is  $[QQ1, BALSAL, QQ1]$ , both symmetrical and equal distribution for suction and pressure sides. Also, is regrouped the 11 regions into four main regions that have the same composite material distribution:

- Leading Panel (LP): composed by leading panel and nose.
- Trailing Panel (TP): composed by trailing panel, tail A, B, C, and tail V.
- Shear Web (SW): composed by shear webs A, B, and C.

Furthermore, in the span-wise direction, the thickness of composite material was changed to a constant thickness in three different sections: lower, middle, and upper (see figure 2.3). These constant values are assumed to be the mean value in this section to simplify the complex composite material distribution. The blade thickness distribution is summarized in tables 2.3 2.4 and 2.5.

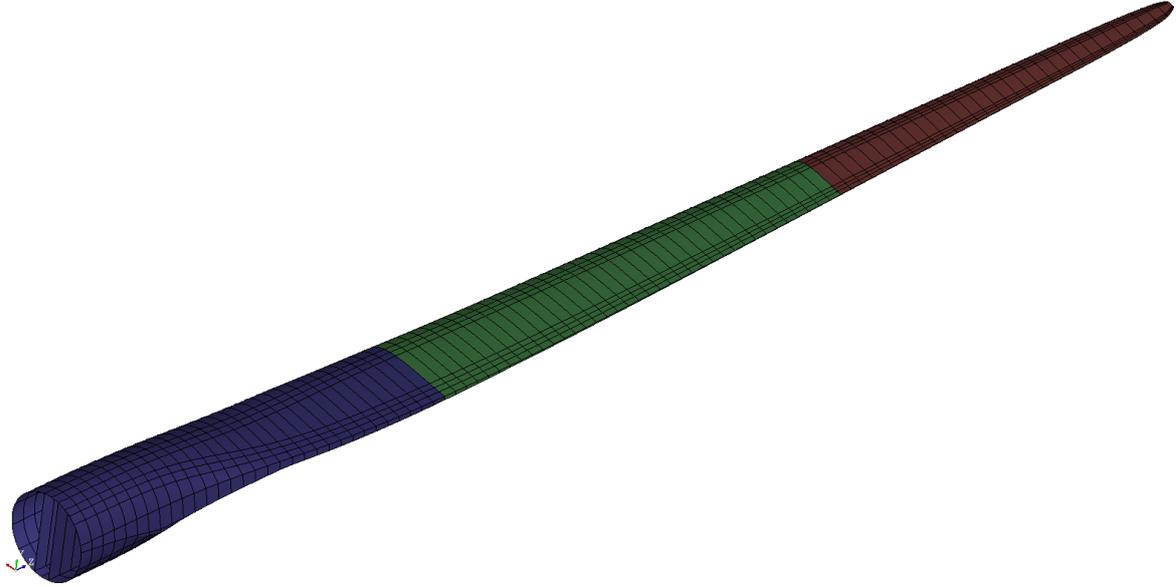


Figure 2.3: Blade span-wise partition for a constant thickness composite material distribution

Table 2.3: Composite material distribution in trailing panel region along the blade.

Trailing Panel						
Z-start	Z-end	P2B	QQ1	BALSAL	QQ1	P2B
[m]	[m]	[mm]	[mm]	[mm]	[mm]	[mm]
2.8	31.301	7.5	8	50	8	7.5
31.301	60.665	5	4	60	4	5
60.665	89.166	2.5	4	25	4	2.5

As the composite material distribution on the blade has changed stiffness, inertia, and mass properties, they are recalculated to recreate the beam model used in FAST.

Table 2.4: Composite material distribution in leading panel region along the blade.

Leading Panel						
Z-start	Z-end	P2B	QQ1	BALSA	QQ1	P2B
[m]	[m]	[mm]	[mm]	[mm]	[mm]	[mm]
2.8	31.301	5	8	25	8	5
31.301	60.665	2.5	4	25	4	2.5
60.665	89.166	2.5	2.5	10	2.5	2.5

Table 2.5: Composite material distribution in shear webs region along the blade.

Shear Webs				
Z-start	Z-end	QQ1	BALSA	QQ1
[m]	[m]	[mm]	[mm]	[mm]
2.8	31.301	2.5	55	2.5
31.301	60.665	5	30	5
60.665	89.166	2.5	10	2.5

### 2.1.2.1 Cross-section blade properties

Introducing the wind turbine blade and tower to FAST, computations of span-variant structural properties are needed. As the wind turbine blades use composite materials, the calculation of these properties is complicated because of the change of layup for different regions, and the use of different composite materials and stacking sequence. To calculate cross-section stiffness properties, exists different numerical codes using a finite element based approach as PreComp by NREL [Bir, 2006], BECAS (BEam Cross-section Analysis Software) by DTU [Blasques, 2012] and VABS (Variational Asymptotic Beam Section) by Yu et al. [Yu et al., 2002]. As the wind turbine blade used as a reference is the WT 10MW DTU, BECAS was selected to determine their cross-section properties.

BECAS can handle an extensive range of arbitrary section geometries, predicting correctly inhomogeneous and anisotropy material effects. Based on a definition of the cross-section geometry and material distribution. It can determine the cross-section stiffness properties while accounting for all geometrical and material induced couplings, inertia properties, and offsets of the blade shear center, tension center, and center of mass concerning the blade pitch axis [Blasques, 2012]. BECAS bases on the theory initially presented by Giavotto [Giavotto et al., 1983] for the analysis of inhomogeneous anisotropic beams. The theory leads to the definition of two types of solutions. Following Saint-Venant's principle, the non-decaying solutions are the basis for evaluating the cross-section stiffness properties.

To use BECAS code, the airfoil coordinates, layup configuration, and material data is input in Airfoil2BECAS python code [Bitsche, 2012], that generates a 2D-mesh of the cross-section of the wind turbine blade with the corresponding material and orientation assignments in BECAS format. Subsequently, the inputs files for BECAS are generated using python code ShellExpander [Branner et al., 2012] based on information in a finite element shell model. It

assumes that the finite element shell model uses layered shell elements and that the shell model nodes have an offset to the layer's outer surface.

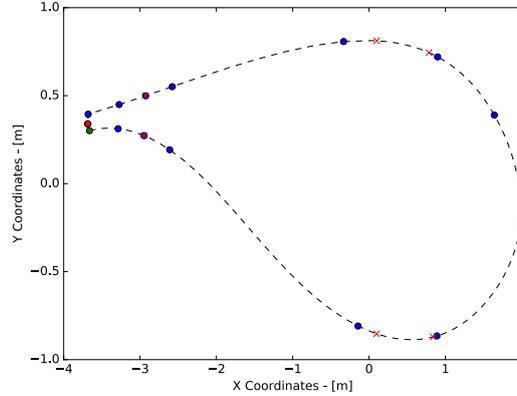


Figure 2.4: Region definition and sequence ordering for airfoil section  $n^{\circ}21$ .

The airfoil nodes must be ordered in sequence, starting from the suction to pressure sides (counterclockwise), then defined nodes represent each region of the airfoil, including shear webs. Also, the definition of material properties and stacking sequence for each region and each cross-section leads to a 2D representation of the wind turbine blade (e.g., figure 2.4). After executing the BECAS code, all cross-sectional properties of the blade can be extracted and compute them into inputs files for FAST code. As shown in figure 2.5, the mesh and mass, elastic, and shear centers are calculated, including stiffness properties and material distribution.

Figure 2.6 shows the mass distribution along the wind turbine blade. As can be seen, there are jumps in the mass due to the simplified composite material distribution, having a total mass of  $44600.75kg$ , which is heavier than the reference 10 MW DTU model  $41716kg$ . Moreover, the change of composite materials made the highest impact in stiffness properties, as shown in figure 2.7 both stiffness, flap-wise and edge-wise, were increased almost in all sections of the blade, resulting in a more resistant wind turbine blade.

### 2.1.2.2 Blade vibration modes

The new composite material distribution on the blade also changed their shape modes, which are recalculated and provided to FAST aero-elastic code. Currently, FAST uses uncoupled modes for flap and edge-wise (lag) degrees of freedom (dof) of the blade and also for the fore and lateral motions of the tower. It ignores the torsion dof and offsets of the shear center, tension center, and center of mass, causing significant dynamic couplings. To overcome these issues, BModes (Software for Computing Rotating Beam Coupled Modes) computes flap-edge-torsion coupled modes by implicitly accounting for the torsion dofs and the offsets mentioned earlier [Bir, 2005].

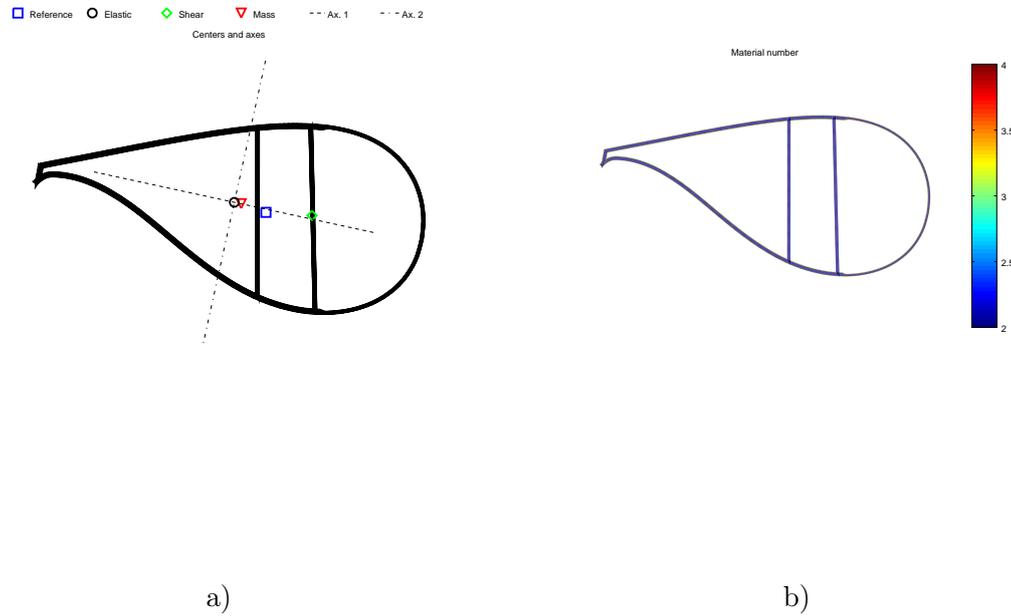


Figure 2.5: 2D representation of blade section  $n^{\circ}10$  a) Elastic, shear and tension centers and b) material distribution (2 = BALSA, 3 = QQ1 and 4 = P2B).

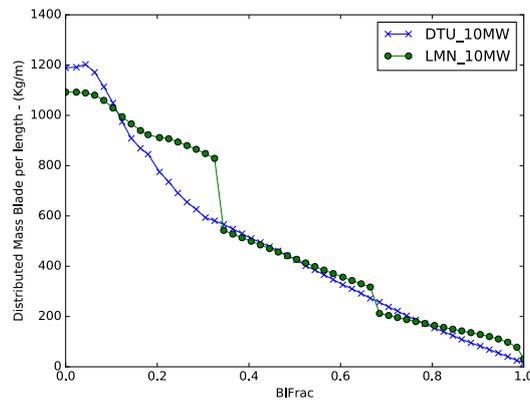


Figure 2.6: Blade mass distribution per sections.

BModes uses a finite-element approach in conjunction with analytical linearization and a special finite-element assembly that accurately captures Coriolis and centrifugal effects [Bir, 2005]. BModes currently offers only one boundary condition to calculate the shape modes: a cantilever boundary condition at the blade root.

To execute BModes is required specification of rotor speed, blade geometry, pre-cone,

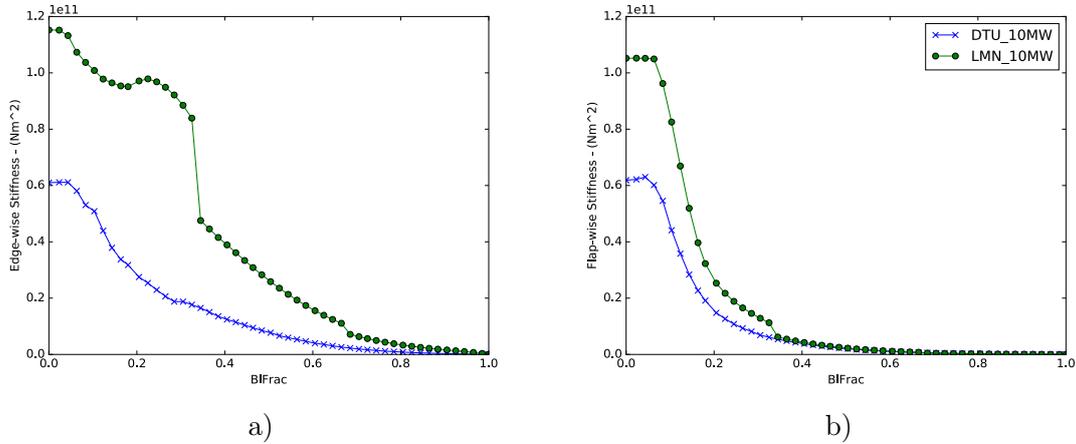


Figure 2.7: Stiffness distribution per section in a) edge-wise and b) flap-wise directions.

pitch control setting, and structural properties distribution along the blade for inputs. The structural properties are specified in terms of the section location (`sec_loc`), structural twist (`str_tw`), flap bending (`flp_stff`), edge bending (`edge_stff`), torsion (`tor_stff`), and axial stiffness (`axial_stff`); mass density (`mass_den`); section moments of inertia flap and edgewise (`flp_iner`, `edge_iner` respectively); and chord-wise offsets of the shear center (`sc_offst`), tension center (`tc_offst`), and center of mass (`cg_offst`) at each section along the blade. BECAS calculates the cross-section properties used for the shape mode calculations.

Table 2.6 shows the result from BECAS to calculate the isolated blade natural frequency and comparison with the frequencies of the reference 10 MW. the first and second shape modes in flap-wise direction are similar to the reference model. However, for the other shape modes, the simplified blade has a greater frequency.

Table 2.6: Comparison of natural frequencies between reference blade [Bak et al., 2013] and simplified 10MW.

Mode	Natural Frequency - [Hz]	
	10MW DTU	10MW Simplified
1st flap mode	0.61	0.603
1st edge mode	0.93	1.175
2nd flap mode	1.74	1.742
2nd edge mode	2.76	3.396
3rd flap mode	3.57	3.772
1st torsion mode	5.69	6.345

To define the wind turbine in FAST is needed to specify two shape modes in flap-wise direction and one in the edge-wise direction in a polynomial representation of the shape. The results obtained using BModes are calculated the flap, edge, and twist displacement as a function of span location for each mode of the blade, choosing the direction with the maximum displacement as the dominant model. They are used to calculate the polynomial coefficients

to define the shape modes in the FAST code. The file of BModes contains an excel sheet *ModeShapePolyFitting.xls* that calculates the polynomial coefficients for mode shapes given deflection data and slope along the length of a flexible beam. After specifying the values for each mode shape, the calculation of the coefficients uses different methods:

- **Direct Method:** does not depend on the given slope, and the resulting mode shape is only valid for small deflections of the beam about the undeflected position. This method is not accurate for all mode shapes and should be avoided when possible.
- **Improved Direct Method:** uses the entered slope and deflection at the bottom of the beam to improve the fit relative to the Direct Method. Like the Direct Method, the resulting mode shape is only valid for small deflections of the beam. If an accurate slope is known and if the deflection data at the bottom of the beam is also accurate, this will most likely be the preferred method for calculating the polynomial coefficients.
- **Projection Method:** depends on the entered slope, the deflection at the bottom of the beam, and the entered factor of  $y$ . This method is the best when trying to find mode shapes about a deflected position. Nevertheless, it is only accurate when the slope and bottom deflection are known accurately.

The wind turbine blade's representation is a cantilever beam, fixed at the bottom. The slope and deflection at the bottom are known accurately, and the Projection Method result for the 6th order polynomial coefficients. The calculation of the coefficients is until the ninth order, but with a sixth-order polynomial, the Projection Method has an excellent fit of the mode shapes (e.g., figure 2.8).

Finally, the blade model's representation uses the defined FAST input file to simulate all different design load cases described in IEC-64001 [Standard et al., 2005]. To analyze the wind turbine blade's behavior, as power production, maximum tip deflection, load uncertainties, and control response. FAST offers different output values at ten span locations of the blade, which can be defined by the user to extract the time series of deflection, loads (forces and moments) of the blade [Jonkman and Buhl Jr, 2005].

However, these outputs are the resultant response applied in a node of the beam representation, meaning that using these results for the fatigue analysis is an approximation of the blade damage and not a detailed distribution of it. The next section explains how to recreate a detailed analysis that needs to transfer this 1D loads to a 3D equivalent load distribution on the blade.

## 2.2 Load Application Methods - LAM

Load Application Methods (LAM) defined by Caous [Caous et al., 2018] are the methodologies defined to transfer the 1D resultant loads from aero-elastic simulation numerical tools into an

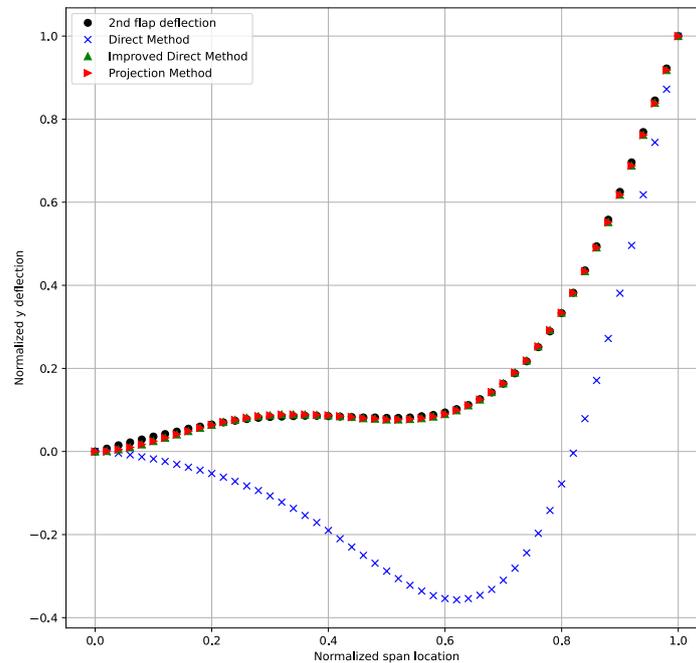


Figure 2.8: Polynomial coefficient calculation for the 2nd shape mode of the blade.

equivalent distribution loads on the shell FEM. The response of the shell model must be equal to the one calculated using the simulation tool. The blade's representation is a shell FEM, calculation of the detailed distribution of stress on the blade leads to buckling, or fatigue analysis (figure 2.9).

## 2.3 Classification

These methodologies are classified by Caous [Caous et al., 2018] depending on how the loads are applied on the shell FEM into four main groups:

- Group 1: Application of loads by sections and on one point of each section (figure 2.10(a)).
- Group 2: Application of loads by sections but physical distribution on sections (figure 2.10(b)).
- Group 3: Continuous application on the blade of an oriented surface load (pressure oriented in a specific direction)(figure 2.10(c)).

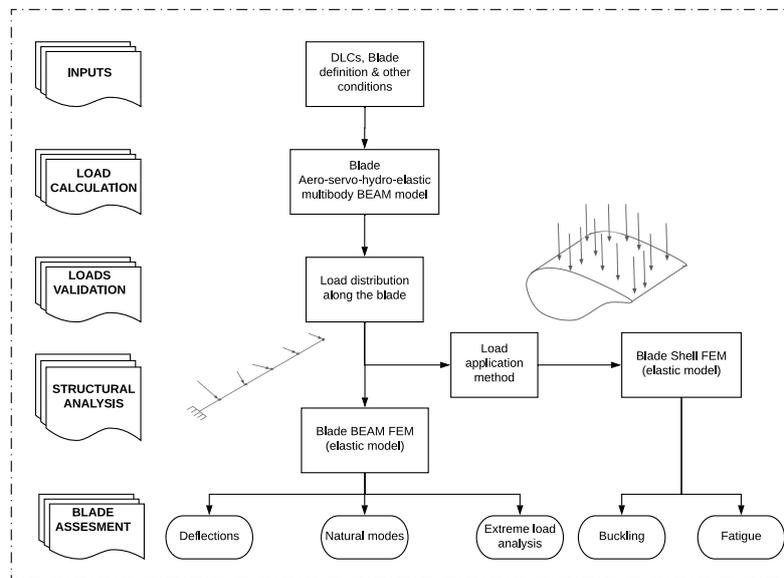


Figure 2.9: Wind turbine blade assessment process [Caous et al., 2018].

- Group 4: Dissociation of inertial and aerodynamic loads with application of an acceleration field and pressure distribution across the whole blade (figure 2.10(d)).

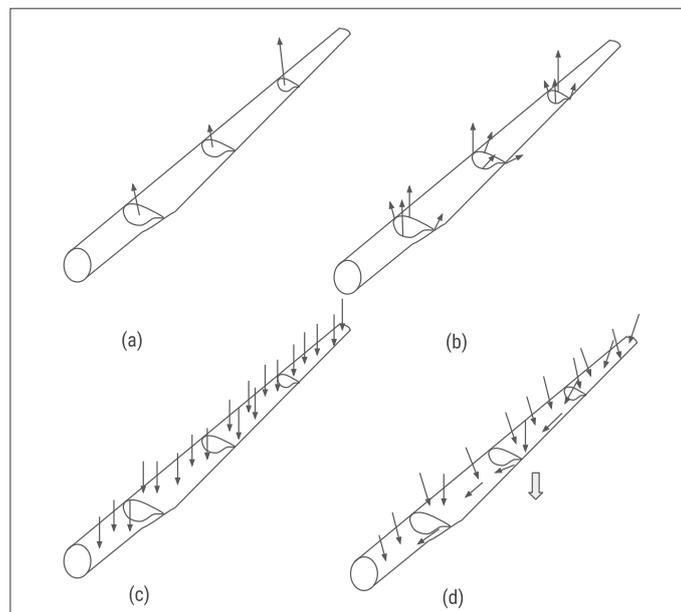


Figure 2.10: Approaches for load application methods in a shell FEM of the blade [Caous et al., 2018].

### 2.3.1 Application of loads by sections and on one point of each section

In the first group, resultants load are applied directly to a small number of sections of the blade, either through a master node which controls the whole section displacement by relations between nodes degrees of freedom ([Forcier and Joncas, 2012],[Haselbach et al., 2016]), directly onto one node that has nearly the same position as the beam FEM node ([Griffith and Ashwill, 2011], [Shokrieh and Rafiee, 2006]), or a small number of nodes of the section of the shell FEM ([Lindgaard and Lund, 2010], [Lund and Stegmann, 2005]).

When using master nodes to control the whole section displacement by relations between node degrees of freedom, it constructs an undeformable section of the structure, called Rigid Body Element (RBE). It imposes linear relationships between the degrees of freedom of the nodes in the undeformable section that the relative displacement between these nodes is zero.

The linear relation is written in a vector form, translating a movement of a rigid body in small rotations:

$$\vec{u}(M) = \vec{u}(A) + \vec{\theta}(A) \times A\vec{M} \quad (2.19)$$

Where  $M$  is a slave node,  $A$  is the master node,  $\vec{u}$  is the displacement, and  $\theta$  is the rotation vector [de France, ]. The resultant loads from fast are applied directly as equation 2.20 at each selected section.

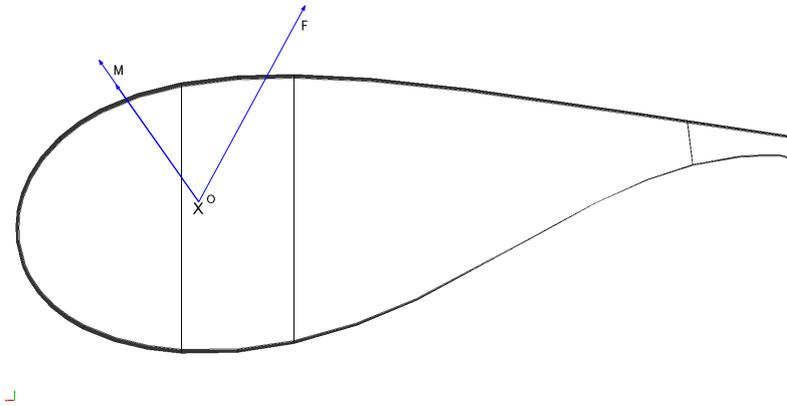


Figure 2.11: Loads distribution per section using Rigid Body Elements

$$\begin{cases} F_i^{Fast}(z) = F_i^{RBE}(z) \\ M_i^{Fast}(z) = M_i^{RBE}(z) \end{cases} \quad (2.20)$$

This first group of approaches is mostly used to model full-scale tests on blades [Branner et al., 2007] or for a fast and straightforward application of loads from the beam FEM to the shell FEM.

These approaches apply load resultants without distinction between inertial and aerodynamic loads [Caous et al., 2018].

### 2.3.2 Application of loads by sections but physical distribution on sections

In the second group (Figure 2.10(b)), resultant loads are no longer applied to sections directly by a few nodes or rigid body elements. However, they have a physical distribution across the nodes of some sections. An example of such an approach was presented by [Bottasso et al., 2014], applying inertial and aerodynamic loads separately through elements that allow an interpolation according to a user law. The computation of identical loading conditions for the FEM is performed by distinguishing between aerodynamic and inertial loads. This way, realistic loading conditions for the blade can be established by limiting the aerodynamic loads' application to the external skin nodes.

Span-wise distributions of inertial loads are recovered by enforcing the equilibrium of a blade portion. With reference to Figure 2.12(a), consider a blade segment  $\eta \in [\eta_i, \eta_{i+1}]$ , where  $\eta \in [0, 1]$  is the span-wise non-dimensional coordinate running along the beam reference line passing through the sectional point E. The internal stress resultants on the  $(i+1)$ th section is  $f_{i+1}$ , and the moment resultant about E is  $m_{i+1}$ , while the ones on the  $i$ th section negative face are  $-f_i$  and  $-m_i$ , respectively. At the span-wise station  $\eta$ , per-unit-span aerodynamic forces  $f^A(\eta)$  and moments  $m^A(\eta)$  are applied at the aerodynamic reference line passing through the sectional point A, which is at a distance  $r^A(\eta)$  from point E on the  $i$ th section. Similarly, per-unit-span inertial forces  $f^I(\eta)$  and moments  $f^I(\eta)$  are applied at the beam reference line, which is at a distance  $r^I(\eta)$  from point E on the  $i$ th section.

The force and moment (about E) equilibrium conditions for the blade segment write.

$$-f_i + f_{i+1} + \int_{\eta_i}^{\eta_{i+1}} (f^A + f^I) d\eta = 0 \quad (2.21)$$

$$-m_i + m_{i+1} + \int_{\eta_i}^{\eta_{i+1}} (m^A + m^I + r^A x f^A + r^I x f^I) d\eta + r x f_{i+1} = 0 \quad (2.22)$$

By using a trapezoidal approximation for the span-wise integrals, one obtains

$$\int_{\eta_i}^{\eta_{i+1}} f^I d\eta \approx \frac{\eta_{i+1} - \eta_i}{2} (f_i^I + f_{i+1}^I) \quad (2.23)$$

Where  $f_i^I$  and  $f_{i+1}^I$  are sectional inertial forces. By inserting (2.23) into (2.21), starting from the blade tip, one may compute each sectional inertial force based on the sectional resultants and aerodynamic loads. Similarly, from (2.22) one may compute each inertial moment.

Once recovering the sectional loads as explained, the application to the blade structure employs RBE3 interpolation elements provided by FE solver MSC Nastran [Nastran, 2004].

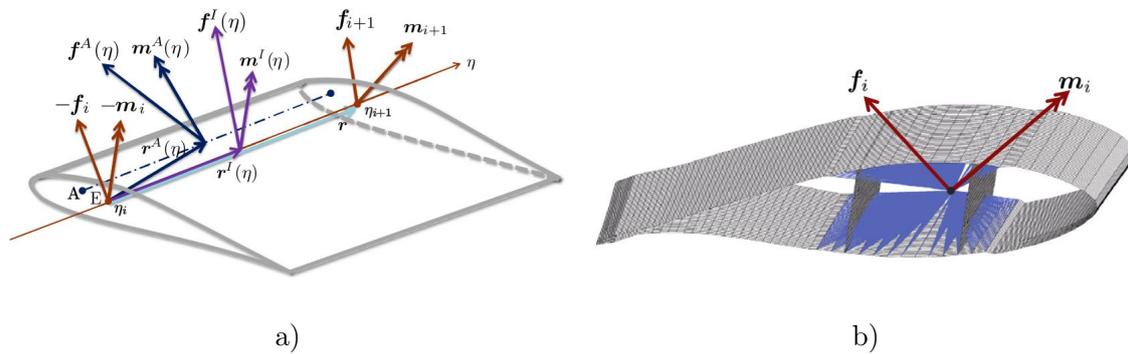


Figure 2.12: Load application method 2: (a) Recovery of inertial and aerodynamic loads from sectional beam FEM loads and (b) inertial load application to spar caps nodes using the RBE3 element [Bottasso et al., 2014]

Different sets of nodes can be associated with different interpolation elements, each one in turn associated with a different set of forces and moments. In all cases, forces are distributed to each node considering local user-defined weighting factors. At the same time, moments are applied as a sum of equilibrated forces on dependent nodes. Figure 2.12(b) shows an example of the application of loads to a shell 3D model's spar cap.

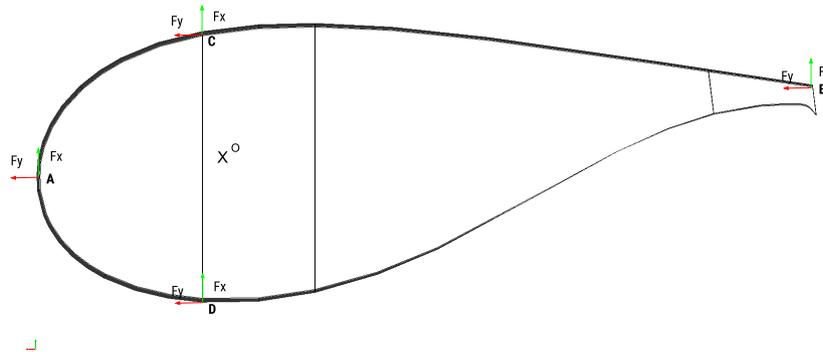


Figure 2.13: Loads distribution on 4 nodes per section.

The second example presenting such an approach is by TENSYL company [Caous and Valette, 2014]. In this approach, the force and moment resultants are distributed between four nodes of each section (e.g., figure 2.13), assuming simplified phenomenological laws. In this method, FAST's resultant loads are applied following the system of equations 2.24.

$$\begin{cases} F_x^{Fast} = F_x^A + F_x^B + F_x^C + F_x^D \\ F_y^{Fast} = F_y^A + F_y^B + F_y^C + F_y^D \\ F_z^{Fast} = F_z^A + F_z^B + F_z^C + F_z^D \\ M_x^{Fast} = y_A \cdot F_z^A + y_B \cdot F_z^B + y_C \cdot F_z^C + y_D \cdot F_z^D \\ M_y^{Fast} = -x_A \cdot F_z^A - x_B \cdot F_z^B - x_C \cdot F_z^C - x_D \cdot F_z^D \\ M_z^{Fast} = x_A \cdot F_y^A - y_A \cdot F_x^A + x_B \cdot F_y^B - y_B \cdot F_x^B + x_C \cdot F_y^C - y_C \cdot F_x^C + x_D \cdot F_y^D - y_D \cdot F_x^D \end{cases} \quad (2.24)$$

where,  $F_x^{Fast}, F_y^{Fast}, F_z^{Fast}, M_x^{Fast}, M_y^{Fast}, M_z^{Fast}$  are the resulting loads from the 1D beam simulation FAST,  $F_x^i, F_y^i, F_z^i$ ,  $i = (A, B, C, D)$  are the nodal forces applied to nodes  $(A, B, C, D)$  and  $x_i, y_i$ ,  $i = (A, B, C, D)$  are the distance from the center  $o$  to each node. In order to have a unique solution, six other equations are required. The missing relations will then be assumed as:

$$\begin{cases} F_x^A = F_x^B = F_x^C = F_x^D \\ F_y^A = 2F_y^C \\ F_y^C = F_y^D \\ F_z^A = F_z^C \end{cases} \quad (2.25)$$

These chosen relations ensure equal forces in the drag direction on all four nodes. The lift is twice as high on the leading edge as the trailing edge and the middle of the suction side to get a lift position close to the quarter-chord length. It depends on the location of point  $C$  and  $D$  [Caous et al., 2018].

This group of approaches has the advantage of applying loads more continuously rather than applying them to one node and not stiffening the blade numerically with rigid elements. However, load application is restricted to sections at discrete positions, and loads are not distributed across the whole blade.

### 2.3.3 Constant pressure application on the blade of an oriented surface load

The third group (Figure 2.10(c)) allows the distribution of the whole blade, applying the aerodynamics loads with a unidirectional surface load. In other words, it applies a force to each element. This force's value is proportional to the element's surface area, and its direction is not necessarily perpendicular to the element surface but oriented in a specific direction. This group consists of approaches where pressure on the blade is constant ([Forcier and Joncas, 2012]; [Hu et al., 2012]; [Sørensen et al., 2014]; [Dimitrov, 2013]) and is distributed through the blade length, for example, according to the variation in pressure coefficient [Berggreen et al., 2007], or distributed through the chord, for example, according to a bi-linear law which assumes a maximal pressure at the quarter-chord point [McKittrick et al., 2001].

In these approaches, surface loads are only applied to pressure or suction side or are directed in a specific direction and only studies flap-wise and edge-wise behavior separately [Caous et al., 2018].

### 2.3.4 Dissociation of inertial and aerodynamic loads with the application of an acceleration field and pressure distribution across the whole blade

The fourth group (Figure 2.10(d)) is used to describe the physics of the blade loading more precisely. It is done by computing the pressure distribution as a function of the wind flow section by section with two-dimensional (2D) aerodynamic codes as XFOIL [Drela, 1989], before applying it to the whole blade surface ([Chen et al., 2013]; [Grujicic et al., 2010]). However, calculating pressure distribution, not computed by the beam finite element aero-elastic simulation code, may lead to differences between the shell and beam FEM load distributions. These differences are due to potentially different assumptions between aerodynamic and aero-elastic simulations: non-viscous flow versus viscous flow, steady-state versus transient or 2D versus 3D effect and different numerical models, and panel method versus blade element momentum method [Caous et al., 2018]. Although it uses computed loads using the beam FEM because they a critical tool and a blade certification body [Jonkman and Buhl Jr, 2005] may have even validated their values. Computing a detailed analysis on a shell FEM with different loading conditions directly using the loads validated as critical is not desirable [Bottasso et al., 2014]. For this reason, it is selected correct pressure distributions methods that induce aerodynamic loads as close as possible to loads computed with the beam FEM, taking into account aero-elastic behavior and control command of the whole blade [Caous et al., 2018].

In the literature are founded several pressure corrections methods. [Hu et al., 2016b] propose a pressure coefficient correction to match lift, drag, and aerodynamic moment coefficients obtained with the aero-elastic beam finite element simulation AeroDyn with XFOIL. [Knill, 2005] corrects the pressure distribution given by the 2D aerodynamic code with a scalar to obtain the same lift force as that given by the aero-elastic simulation. Then are corrected the drag force by applying a force proportional to the surface area of the elements. Inertial forces are applied with loads proportional to element surface area and not mass to simplify load transfer.

The correction by [Hu et al., 2016b] concerns lift, drag, and aerodynamic moment coefficients, which may differ from final forces after applying the pressure because of blade geometry variation. Segments of blades defined in an aero-elastic beam FEM lead to a simplified geometry, which may differ from the blade modeled with finite shell elements. These geometries differ in terms of local chord length or twist or global dimension variation through the length, leading to a radial force when pressure is applied. It also uses the aero-elastic simulation AeroDyn; it only takes into account dynamics stall and blade rotation. It does not take into control and electrical (servo) or elastic effects as FAST does, meaning that in Hu study, the blade is undeformed when calculating the aerodynamic wind load within AeroDyn [Hu et al., 2016a]. The correction proposed by [Knill, 2005] considers this geometry variation computed from the

differences between lift and drag forces. However, the aerodynamic moment is not corrected.

The method proposed by [Caous et al., 2018] overcomes the limitations described through a novel correction of pressure coefficients to match lift force, drag force, and aerodynamic moment. Also, the correction of inertial loads is provided to recover bending moments. This method applies the loads separately to the shell FEM, depending on their source:

- Application of Aerodynamic loading with surface loads.
- Inertial loads, including both gravity and rotational forces, are applied as body loads.

As described by Caous, the first step is to correct the lift force by a coefficient using the method developed by TJ Knill [Knill, 2005]. The coefficient is calculated as:

$$c_{knill} = \frac{L_{node}}{L_{aero}} = \frac{F_x^{node}}{F_x^{aero}} \quad (2.26)$$

Where  $L_{node}$  is the lift force from the beam FEM and the  $L_{aero}$  is the lift force from the pressure distribution of the selected section, calculating this coefficient at each selected cross-section. Subsequently, modification in the pressure distribution using this coefficient as:

$$P = c_p q c_{knill} \quad (2.27)$$

where  $c_p$  is the pressure coefficient,  $q$  is the fluid dynamic pressure calculated as  $q = 1/2\rho V_{rel}^2$ . This modification in the total pressure distribution is going to change both the lift and drag forces. The 2D airfoil tool Xfoil [Drela, 1989] calculates the aerodynamic pressure coefficient of  $c_p$ . From 2D geometry, local wind incidence angle, and Reynolds number at each span cross-section (i.e., airfoil), computing the aerodynamic pressure coefficient distribution on the pressure and the suction side of the airfoil. It is used results obtained for pressure coefficient from the reference 10 MW [Bak et al., 2013] to calculate the airfoil's pressure distribution. These  $c_p$  are calculated for an incidence angle between  $\pm 32^\circ$ . It then chooses the pressure coefficient to calculate the pressure  $P$  as the one obtained with the closest incidence angle calculated using Xfoil. If the incidence angle is greater than  $\pm 32^\circ$ , it uses this last one. For thick and nearly cylindrical profiles, for instance, near the blade root, a simplified law is used based on experimental data [Roshko, 1954, Roshko, 1961]. This law assumes a constant  $c_p$  equal to  $-1$  on the whole downwind surface and a linear variation on the upwind surface from 1 at the upwind stagnation point to  $-1$  at the two points where upwind and downwind surfaces meet.

Once the pressure is calculated and corrected about  $F_x$  at each section, they are interpolated on the shell FEM blade surface. Despite this correction, the resultant aerodynamic force  $F_x$  computed on the shell FEM differs from the beam FEM target, as shown in Fig.2.14.

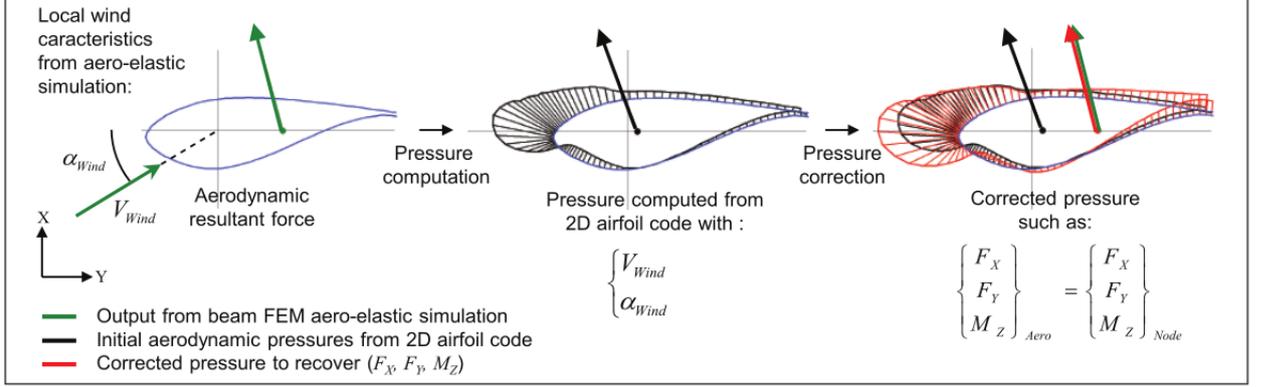


Figure 2.14: Aerodynamic pressure computation and correction [Caous et al., 2018].

Thus, additional pressure is applied to correct the three aerodynamic forces  $F_x$ ,  $F_y$ , and  $M_z$ . Two oriented pressure loads are then applied in order to correct the aerodynamic forces and moment:

- Constant pressure by section, oriented towards the  $X - axis$ , to correct the  $F_x$  forces.
- Chord location-dependent pressure, oriented towards the  $Y - axis$ , to correct the  $F_y$  forces and the  $M_z$  moments.

Let us consider a section with  $x$  the location along the chord, pressure corrections  $P_x$  (along  $X - axis$ ) and  $P_y$  (along  $Y - axis$ ) are applied according to the following equations:

$$\begin{cases} P_x(X) = P_x = const \\ P_y(X) = P_{y1} + x.P_{y2} \end{cases} \quad (2.28)$$

In order to determine these two pressure, is necessary to compute the difference between:

$$\begin{cases} \Delta F_x \\ \Delta F_y \\ \Delta M_z \end{cases} = \begin{cases} F_x \\ F_y \\ M_z \end{cases}_{Node} - \begin{cases} F_x \\ F_y \\ M_z \end{cases}_{Aero} \quad (2.29)$$

From Eq. 2.28, the correction pressure for each component can be calculated as:

$$\begin{cases} \Delta F_x = \int_c (P_x) ds \\ \Delta F_y = \int_c (P_{y1} + x.P_{y2}) ds \\ \Delta M_z = \int_c \vec{op} \wedge (P_x \vec{x} + P_y \vec{y}) ds \end{cases} \quad (2.30)$$

equation 2.30 can be simplified to:

$$\begin{cases} \Delta F_x = a_x \cdot P_x \\ \Delta F_y = a_{y1} \cdot P_{y1} + a_{y2} \cdot P_{y2} \\ \Delta M_z = a_{y3} \cdot P_{y1} + a_{y4} \cdot P_{y2} + b_y \cdot P_x \end{cases} \quad (2.31)$$

and represented as a matrix form:

$$\begin{cases} P_x = \frac{\Delta F_x}{a_x} \\ \begin{pmatrix} P_{y1} \\ P_{y2} \end{pmatrix} = \begin{pmatrix} a_{y1} & a_{y2} \\ a_{y3} & a_{y4} \end{pmatrix}^{-1} \begin{pmatrix} \Delta F_x \\ \Delta M_z - b_y \cdot P_x \end{pmatrix} \end{cases} \quad (2.32)$$

where, the six coefficients  $a_x, a_{y1}, a_{y2}, a_{y3}, a_{y4}$  and  $b_y$  are only dependent on the mesh and geometry of the blade.

$$\begin{aligned} a_x &= \int_c (\vec{n} \cdot \vec{x}) ds \\ a_{y1} &= \int_c (\vec{n} \cdot \vec{y}) ds \\ a_{y2} &= \int_c x \cdot (\vec{n} \cdot \vec{y}) ds \\ a_{y3} &= \int_c (\vec{o}\vec{p} \wedge \vec{y}) ds \\ a_{y4} &= \int_c x \cdot (\vec{o}\vec{p} \wedge \vec{y}) ds \\ b_x &= \int_c (\vec{o}\vec{p} \wedge \vec{x}) ds \end{aligned} \quad (2.33)$$

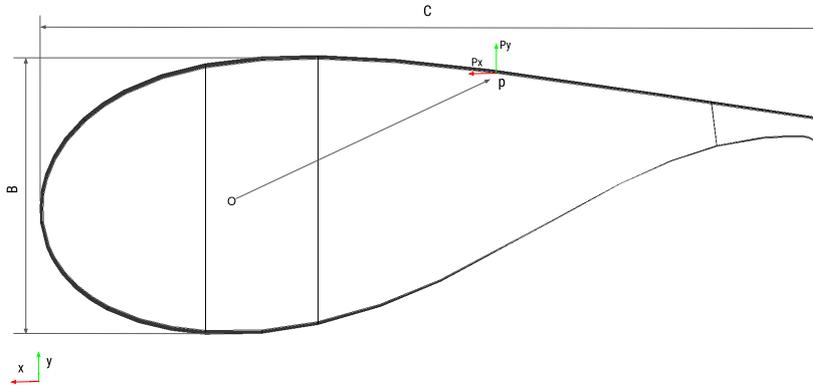


Figure 2.15: Geometrical properties: chord, thickness, aerodynamic center ( $C$ ,  $B$  and  $o$ ) and pressure components  $P_x$  and  $P_y$  (blue and red arrows respectively) of a segment in an airfoil.

Considering as reference the airfoil of Fig.2.15 and the definition of each coefficient:  $a_x = 2B$  and  $a_{y1} = 2C$ . To calculate the coefficient  $a_{y2}$ , let us define the distance from the nose to point  $o$  (aerodynamic center of the beam FEM) as  $\alpha C$ , then, the integral can be defined as:

$$a_{y2} = \int_{-\alpha C}^{(1-\alpha)C} y ds = (2\alpha^2 - 2\alpha + 1)C^2 \quad (2.34)$$

where  $\alpha$  is the fraction of the distance from nose to the aerodynamic center. For coefficients  $a_{y3}$  and  $b_y$ , let us develop only the coefficient  $a_{y3}$ , because for both coefficient is the same. Defining distance  $\vec{op} = \langle r \cos(\theta), r \sin(\theta) \rangle$  as  $r = |\vec{op}|$  and  $\theta$  the angle of the vector and  $\vec{x} = \langle 1, 0 \rangle$ , the vector product result as:

$$\vec{op} \wedge \vec{x} = \begin{vmatrix} i & j \\ r \cos(\theta) & r \sin(\theta) \\ 1 & 0 \end{vmatrix} = -r \sin(\theta) \quad (2.35)$$

Substituting the defined integral of  $a_{y3}$  by it's Riemann's Sum and also adding the result from Eq.2.35, is obtained:

$$a_{y3} = \sum_{i=1}^n -r_i \sin(\theta_i) \Delta s \quad (2.36)$$

Following the same steps for  $b_y$  is obtained:

$$b_y = \sum_{i=1}^n r_i \cos(\theta_i) \Delta s \quad (2.37)$$

$r_i$  and  $\theta_i$  depends of the size of the mesh of the selected section of the blade.

Last coefficient  $a_{y4}$  is calculated as:

$$a_{y4} = \sum_{i=1}^n -r_i \sin(\theta_i) x_i \Delta s \quad (2.38)$$

In summary, to apply the distributed pressure over the blade surface and correct the aerodynamic loads, it is calculated and corrected the lift force to match with the beam FEM using TJ Knill's method. Then, calculates the constant pressure  $P_x$  and apply it to correct the error introduced from the first method in the drag force. Afterward,  $P_{y1}$  is calculated and applied, and finally,  $P_{y2}$  is applied to erase the error in the lift force and the aerodynamic moment in  $Z$ -direction.

It is worth noting that aerodynamic loads were corrected according to  $F_X$ ,  $F_Y$ , and  $M_Z$ . By computing acceleration correction after applying aerodynamic loads, it is also possible to correct any difference in bending moments observed after pressure correction. This difference is due to the miss-align between the position of the blade beam's pressure center and the shell FEM.

The inertial load's application follows a piece-wise linear law along the blade through body loads, proportional to each element's mass. It performs an acceleration correction by applying

translational accelerations along X-axis and Y-axis. These accelerations  $A_X$  and  $A_Y$  generate, respectively, a force  $F_X$  and a bending moment  $M_Y$  and a force  $F_Y$  and a bending moment  $M_X$ . Each internal bending moment, computed on a section, depends on accelerations applied from this section to the blade tip.

The acceleration correction can be expressed by the following relation:

$$\begin{cases} M_X^{Node}(z_i) = \sum_{j=i}^n c_{ij} \cdot A_Y(z_j) \\ M_Y^{Node}(z_i) = \sum_{j=i}^n d_{ij} \cdot A_X(z_j) \end{cases} \quad (2.39)$$

where  $M_X(z_i)$  and  $M_Y(z_i)$  are the internal bending moments, extracted at section  $i$ , induced by the acceleration correction  $A_X$  and  $A_Y$  and  $c_{ij}$  and  $d_{ij}$  are parameters which depend only on the shell FEM (mesh and geometry).

In order to determine the value of  $c_{ij}$  and  $d_{ij}$  the equation 2.39 is going to be developed. As definition is known that:

$$\Delta M_X = \int_{z=z_i}^{z=z_n} z \cdot m \cdot A_Y dz \quad (2.40)$$

As explained by Caous, this equation is a piece-wise linear function, defining a linear function between two sections as:

$$A_Y = (1 - w(z))A_Y(z_i) + w(z)A_Y(z_{i+1}) \quad (2.41)$$

where  $w(z)$  is a linear function between two section:

$$w(z) = \frac{z - z_{i+1}}{z_i - z_{i+1}} \quad (2.42)$$

Using equations 2.41 and 2.42 in 2.40, this last one can be re-written as:

$$\Delta M_X(z_i) = \int_{z_i}^{z_{i+1}} (z - z_i) \cdot m(z) \cdot ((1 - w(z)) \cdot A_Y(z_i) + w(z) \cdot A_Y(z_{i+1})) dz + \dots + \int_{z_{n-1}}^{z_n} (z - z_{n-1}) \cdot m(z) \cdot ((1 - w(z)) \cdot A_Y(z_{n-1}) + w(z) \cdot A_Y(z_n)) dz \quad (2.43)$$

However, this system cannot be solved because  $j$  ranges from 1 (blade root) to  $n$  (blade tip), whereas  $i$  ranges from 1 to  $n - 1$ . To solve the problem and guarantee the uniqueness of the solution, an additional relation between acceleration correction values on two sections has to be defined. A linear variation is then defined between the rotation point of the turbine, where acceleration correction is chosen as null, and the first two sections of the blade. This can be written in the following form:

$$\begin{cases} A_Y(z_1) = \frac{z_1 - z_{rotation}}{z_2 - z_{rotation}} \cdot A_Y(z_2) \\ A_X(z_1) = \frac{z_1 - z_{rotation}}{z_2 - z_{rotation}} \cdot A_X(z_2) \end{cases} \quad (2.44)$$

Once the internal bending moment corrections to be applied to  $\Delta M_X$  and  $\Delta M_Y$  are determined section by section, and matrix C are computed, acceleration corrections are determined according to:

$$\begin{pmatrix} A_Y(z_1) \\ A_Y(z_2) \\ \dots \\ A_Y(z_n) \end{pmatrix} = \begin{pmatrix} 1 & -\frac{z_1 - z_{rotation}}{z_2 - z_{rotation}} & 0 & \dots & 0 \\ c_{11} & c_{12} & c_{13} & \dots & c_{1n} \\ 0 & c_{22} & c_{23} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & c_{nn} \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ \Delta M_X(z_1) \\ \Delta M_X(z_2) \\ \dots \\ \Delta M_X(z_{n-1}) \end{pmatrix}$$

Given the limitations of the FEA *Code\_Aster* software used in this study, the definition of an acceleration piece-wise linear function is not possible. Only a uniform application of acceleration is possible. In this study, the accelerations were applied proportional to the wind turbine blade's surface area and not proportional as the blade mass, using surface loads.

In summary, based on the above classification, the first two groups concern discrete load application methods, while the third is continuous but with loads oriented along only one direction. Suppose we need a method to apply a continuous loading, taking multi-axial loads into account to calculate the fatigue damage. In that case, an approach from the fourth group must be considered [Caous et al., 2018]. Caous compared three methods from groups 1, 2, and 4 regarding their differences in stress distribution using puck criteria [Puck et al., 2002] and the blade tip deflection, concluding that depending on the load application method could have minor consequences for the conclusion of structural analysis. Discrete load application methods could induce stress concentrations around nodes where loads are applied, thus decreasing safety factors. Also, he noticed that the rigid element method stiffens the blade locally, which influences blade deflection. Positives and negatives aspects of each method are summarized in table 2.7.

Table 2.7: Advantages and disadvantages of load application methods [Caous et al., 2018]

Load application method	Pros	Cons
Group 1	<ul style="list-style-type: none"> <li>- Easy to implement</li> <li>- Fast analysis of the wind turbine blade</li> <li>- Low computational cost</li> </ul>	<ul style="list-style-type: none"> <li>- Stress concentrations</li> <li>- No distinction between inertial and aerodynamic loads</li> </ul>
Group 2	<ul style="list-style-type: none"> <li>- Load applied to several nodes on the section</li> <li>- Physical distribution of the loads in the shell FEM</li> <li>- Low computational cost</li> </ul>	<ul style="list-style-type: none"> <li>- Stress concentrations</li> <li>- Loads are distributed in a section and not along the whole blade</li> <li>- No distinction between inertial and aerodynamic load</li> </ul>
Group 3	<ul style="list-style-type: none"> <li>- Distribution of the loads along the whole blade</li> <li>- No stress concentrations</li> <li>- Medium computational cost</li> </ul>	<ul style="list-style-type: none"> <li>- Pressure applied only in one direction (flapwise or edgewise)</li> <li>- No distinction between inertial and aerodynamic loads</li> </ul>
Group 4	<ul style="list-style-type: none"> <li>- Distribution of the loads along the whole blade</li> <li>- Distinction between inertial and aerodynamic loads</li> <li>- No stress concentration</li> <li>- Pressure applied following a more precise blade loading physics</li> </ul>	<ul style="list-style-type: none"> <li>- Torsion loading not taken into account</li> <li>- High computational cost</li> </ul>

However, the sensitivity induced by those load application methods has not yet been analyzed by other authors, which could introduce more complexity to the blade's reliability analysis if a gradient has to be calculated, for example. Also, as each method differs in the blade's stress and tip deflection behavior, the variance in the response could have a part corresponding to the method itself. In this case, a sensitivity analysis regarding the blade's stress and tip deflection could be a useful criterion to select which method is the most suitable to realize a fatigue life estimation or a reliability analysis. It takes into account the stress distribution on the blade and the sensitivity induced by the variation of their parameters.

## 2.4 Application of LAM to a composite wind turbine blade

This study uses three LAM: rigid body elements (RBE), four nodes method (4NO), and Caous method pressure distribution (PD) to apply the resulting loads from an aero-elastic simulation at a constant speed of  $11m/s$  carried out in the 10MW wind turbine blade with a simplified composite material distribution described before.

It Uses a shell FEM of the blade with 30613 nodes and 46400 elements, nine rigid elements used for the load application method RBE. Linear static FEA is run with *Code\_Aster*, assuming small perturbations.

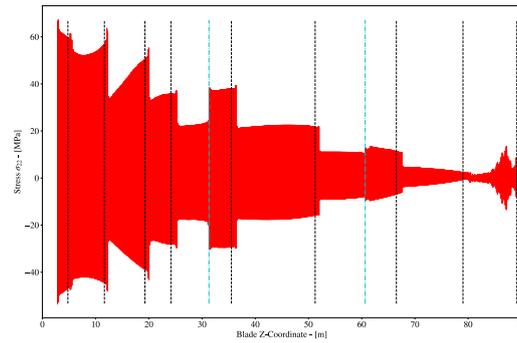
### 2.4.1 Blade stress distribution

In this case of study, is shown the maximum and minimum stress distribution  $\sigma_{22}$  of the first ply (P2B) of the laminate in figure 2.16, extracted at each Gaussian point along the blade span-wise direction (Z-axis), the dash lines represents the position where the loads are applied or corrected, and the cyan dash dotted line presents the position where the thickness change abruptly. First, when using the LAM RBE, figure 2.16(a), the maximum stress magnitude along the blade is higher at the tail A, B, C, and V (end of the trailing edge), and lower stress between the shear webs. Also, the stress pattern presents jumps in stress level. This behavior is due to the assumption of applying the loads at cross-sections acting as rigid bodies. As the forces are applied directly to the section, all nodes have the same displacement, and the stress will be maximum in the zones with the biggest change in geometry. In this case, the end of the trailing edge and the shear webs A and B increase the stiffness on the flap-wise direction, explaining the reduced stress produced in this area. Also there are jumps in stress level due to the discontinuity created by the simplified material distribution. In this scenario there is an abrupt change in thickness that creates a huge variation in stress.

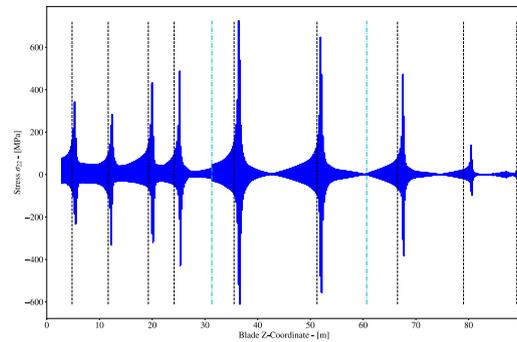
Second, using LAM 4NO, figure 2.16(b), the stress along the blade presents points where the stress increases abnormally. This resulting behavior is due to the approach used by applying the loads directly in 4 nodes of a cross-section. Applying punctual nodes is generated stress concentration on the local site of load application, multiplying the stress by a constant that will depend on the magnitude of the force and the shell element's area. This method

presents the highest stress level of all three methods tested.

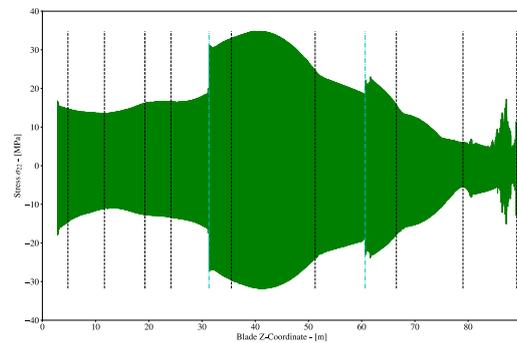
Third, using LAM PD, figure 2.16(c), the stress distribution along the blade is way different from all other methods. In this case, as the loads are applied using a pressure distribution corrected using the Caous method [Caous et al., 2018], the stress along the blade varies not only in the span direction but also in the radial direction. In this case, high stress is found in the middle of the blade leading panel section in the pressure side and decreasing towards the blade tip and the leading edge. Similar to the stress level applying LAM RBE, there are jumps in the stress level due to the change in thicknesses. This method presents a more realistic representation of the stress acting on the blade. Compared with all other literature methods, it is the most accurate to apply the loads from an aero-elastic simulation [Caous et al., 2018].



a) Load application method: RBE



b) Load application method: 4NO



c) Load application method: PD

Figure 2.16: Maximum and minimum  $\sigma_{22}$  stress of first layer of composite material P2B of 10MW Wind turbine blade at Gaussian points.

Caous [Caous et al., 2018] presented a comparison of these three methods in a smaller wind turbine blade, using the distribution of the Puck safety factor [Puck et al., 2002]. In his study, LAM 4NO presented a location as the most damaged one different than LAM RBE and PD. This result is similar, regarding the stress distribution.

Figure 2.17 shows the Von Mises plane stress at a fixed position of X coordinate (a)  $X = -0.5$  and b)  $X = 0$ ) and varying Z coordinate and selecting only positive Y coordinate (pressure side or the blade). As shown in the previous figure, the stress distribution is lower for LAM PD, then follows RBE and the higher stress is found using 4NO. In this case is

compared the exact position between all three methods. This indicates that methods RBE and 4NO will produce a higher stress/fatigue damage compared to LAM PD for the same wind conditions.

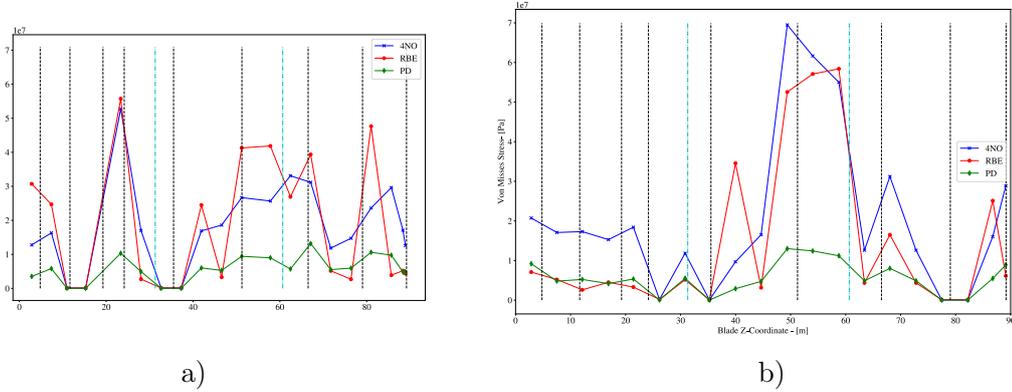


Figure 2.17: Von Mises plane stress comparison for the LAMs at fixed coordinate a)  $X = -0.5$  and b)  $X = 0$ .

Figure 2.18 shows a comparison of the deflection between the three LAMs and the aero-elastic beam response. The deflection response in the flapwise direction at the blade tip for all three methods are similar to the response of the aero-elastic simulation beam model (see table 2.8). However, for the other points of the blade, the deflection calculated for all three methods is greater than the aero-elastic simulation. RBE presents the higher deflection, followed by 4NO and, finally PD. LAM PD is the method that estimates the closest deflection behavior to the aero-elastic simulation compared to RBE and 4NO. These methods could be improved not only regarding the loads but also the deflection along the blade ensuring the exact behavior. These deflections differences could be produced due to the simplified beam model used in the aero-elastic simulation and the 2D shell finite element software shortcomings to correct acceleration. As mentioned by Caous [Caous et al., 2018] the pressure on the real blade surface induces a radial effort that is not calculated in the beam FEM, or the case of FAST, the torsional moment due to inertial loads is not considered in the beam element model. Therefore, although differences may remain between the beam finite element servo-aero-elastic simulation and the shell FEM loads, with the LAM PD, this method seems to provide a loading which describes the physical phenomena better than the beam FEM. However, further investigation on this subject should be carried out, including a comparison with a reference load distribution taken from on-site measurements or computational fluid dynamics (CFD) analysis.

Table 2.8: Comparison of blade tip displacement

Method	Blade tip displacement, flapwise (m)
Aero-elastic simulation	4.413
RBE	4.4898
4NO	4.36855
PD	4.36551

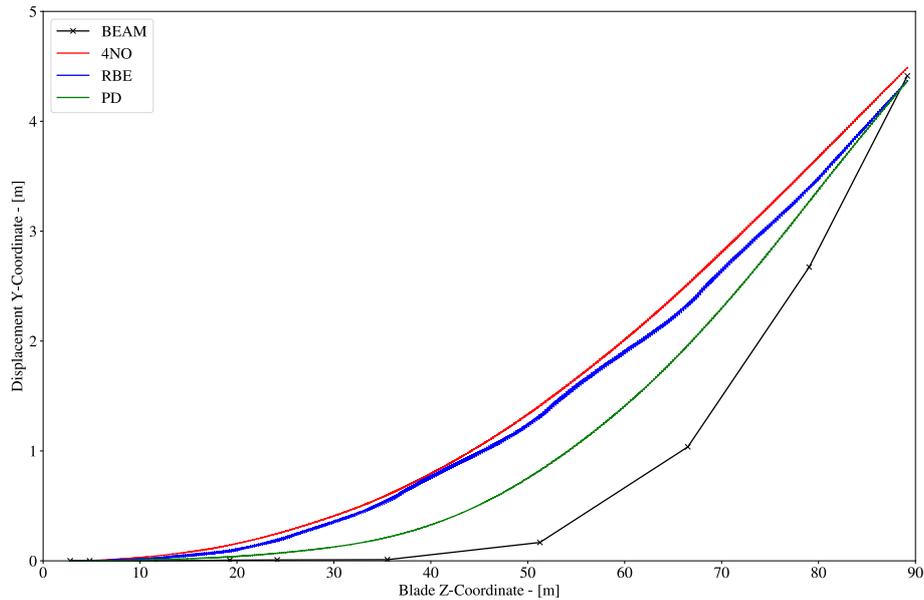


Figure 2.18: Deflection comparison of LAM and aero-elastic response in flapwise direction.

## 2.5 Conclusions

The simplified wind turbine blade model developed in this chapter has a greater mass and stiffness in both principal directions (flapwise and edgewise) than the original model developed by DTU. These changes in mechanical properties increases the natural frequencies. Also, using a simplified material distribution helps to reduce the computational time to create the blade finite element model, moreover, this model is develop to be used in an optimization problem to minimize the weight of the blade. However, this simplified material distribution introduces discontinuities while changing between zones where the thicknesses are different.

The following study uses the three methods tested to estimate the stress in the simplified wind turbine blade or fatigue damage. To clarify, LAM PD is the most accurate method to transfer the aero-elastic loads in our study, and LAM RBE or 4NO are used to compare their results with the primary method.



# Deep learning application for wind turbine blade response

---

## Sommaire

<b>3.1</b>	<b>Deep learning in Wind Turbine</b>	<b>44</b>
<b>3.2</b>	<b>Introduction to deep neural network</b>	<b>45</b>
3.2.1	Perceptron	45
3.2.2	Multi-layer perceptron	47
3.2.3	Activation functions	49
3.2.4	Deep neural network optimizer	51
3.2.5	Overfitting problems	55
<b>3.3</b>	<b>Tuning hyperparameters</b>	<b>57</b>
3.3.1	Bayesian optimization for hyperparameter tuning	57
3.3.2	Random search vs. Bayesian optimization for hyperparameters optimization	59
<b>3.4</b>	<b>Discontinuity modeling for wind turbine simulated loads</b>	<b>61</b>
3.4.1	Pitch control problems for steady and unsteady simulations	61
3.4.2	Automated Discontinuity Modeling Deep Neural Network (ADM-DNN)	64
<b>3.5</b>	<b>Conclusions</b>	<b>69</b>

---

The machine learning regression algorithms create a continuous surrogate model of data behavior. However, if this data has discontinuities jumps, their response is not longer continuous in the domain but is continuous per region. Using a state of the art regression algorithm, it will predict in between the discontinuity introducing error to the surrogate model's response. Some authors determine where the discontinuities are located and train various surrogate models for each continuous region. Others make a combination of these surrogate models. Nevertheless, these approaches need to train multiple surrogate models, and, after their combination, some approaches present a continuous response. This chapter proposes a new methodology to model discontinuity jumps in regression problems using deep neural networks, which automatically determines the discontinuities and trains one surrogate model that predicts only the problem's continuous response. Section 3.2 presents an introduction, architecture, and optimization of deep neural networks. Section 3.3 presents the optimization of hyperparameters in a deep neural network. The discontinuity modeling approach is explained in Section 3.4 presents a state of the art of different approaches, tests, and explains the methodology, comparing with

other regression methods in a benchmark problem and an aero-elastic load estimation. The proposed method outperforms all tested regression methods in both cases.

### 3.1 Deep learning in Wind Turbine

Deep learning (DL) proves to be an accurate and efficient alternative compared to classical modeling techniques. DL is a branch of machine learning used for supervised and unsupervised learning. DL's architecture allows handling multiple inputs and outputs in one network, giving an advantage over other machine learning algorithms to capture the behavior of high dimensional problems. However, it needs a large amount of data to train, and the number of parameters to train is more significant than all other machine learning algorithms. Nowadays, DL applications in wind turbines are increasing due to their advantage in dealing with uncertainty problems. Wind turbine uncertainties come from multiple sources as environmental conditions, material properties, and manufacturing defects. In these cases, DL is an excellent alternative to deal with these complex problems because testing a wind turbine is expensive and time-consuming. Thus DL saves human-time and effort in experiments.

Numerous DL applications are found in the literature in the latest years in many different fields, including wind turbine structure, optimization, wind power, wind speed forecasting, and others. For wind turbine's structural health monitoring (SHM), Dervilis Nikolaos [Dervilis, 2013] investigated the SHM of wind turbine blades using neural networks. Sierra-Perez et al. [Sierra-Pérez et al., 2016] introduced a hierarchical nonlinear principal component analysis method for damage diagnosis in wind turbine blades. The authors demonstrated the methodology's effectiveness based on a composite 13.5m blade's strain measurements and investigated pattern recognition for SHM. For wind energy forecasting problems, Sergio et al. [Sergio and Ludermir, 2015] investigated the hourly average wind speed in Northeastern of Brazil using a deep belief network and stacked auto-encoder: comparing with multi-layered perceptron. Khodayar and Teshnehlab [Khodayar and Teshnehlab, 2015] proposed a stacked auto-encoder neural network for ultra-short-term and short-term wind speed prediction. Liu et al. [Liu and Zhang, 2016] presents an extensive empirical study of machine learning methods for wind power predictions of seven wind farms in Ontario, Canada. They have found that the support vector machine, followed by DNN, has the best overall performance. Peng et al. [Peng et al., 2016] predicted the wind power using a Multilayer Restricted Boltzmann Machine (MRBM), a deep learning neural network with strong feature interpretation ability reducing the mean squared error compared to a standard deep neural network. Wand et al. [Wang et al., 2017] investigated convolutional neural network design for probabilistic wind power forecasting. Liu et al. [Liu et al., 2018] presents a wind speed forecasting method based on deep learning strategy using empirical wavelet transform, long short-term memory neural network, and Elman neural network. Recently, Srivastava et al. [Srivastava et al., 2019] investigated the use of the recurrent neural network for predicting the unsteady aero-elastic response of transonic pitching and plunging wing-fuel tank sloshing system. Li et al. [Li et al., 2019] investigated the prediction of unsteady flow dynamics at different Mach numbers using long short-term memory networks.

However, more DL applications in a wind turbine have not been studied intensely, representing a research opportunity to develop more methodologies to apply DL to wind turbine problems. This chapter explains how the deep neural networks work and how to optimize their hyperparameters. Also is developed an approach to predicting the maximum loads of a steady aero-elastic simulation. This approach is the first application of DL in this thesis.

## 3.2 Introduction to deep neural network

Scientists have always looked for inspiration in nature; birds inspire airplanes, whales inspire serrated-edge wind turbines. As logical, to create intelligent machines, the inspiration comes from the brain's architecture being the main idea of artificial neural networks (ANNs).

ANNs were first introduced back in 1943 by the neurophysiologist Warren McCulloch and the mathematician Walter Pitts [McCulloch and Pitts, 1943], they presented a simplified computational model explaining how biological neurons might work together to perform complex computations using propositional logic. This was the first architecture invented for ANN. Different architectures have been developed to solve problems as: recognize handwritten digits or predict the weather at a desired time or even generating a new image with the style of Leonardo da Vinci. Some of these architectures are listed bellow:

- Perceptron [Rosenblatt, 1957]
- Convolutional Neural Networks [Sermanet et al., 2012]
- Recurrent Neural Networks [Elman, 1990]
- Hopfield Network [Hopfield, 1982]
- Boltzmann Machine [Hinton et al., 1986]
- Autoencoders
- Generative Adversial Network [Goodfellow et al., 2014]

The only perceptron based architecture are described in this manuscript. The networks' main task is to predict numerical outputs from numerical inputs; in other words, supervised learning for regression problems. Also, perceptrons are used for classification problems.

### 3.2.1 Perceptron

The Perceptron is one of the simplest ANN architectures, invented by Frank Rosenblatt in 1957 [Rosenblatt, 1957]. It is based on a linear threshold unit (LTU): the inputs ( $\mathbf{x} = (x_1, x_2, x_3, \dots, x_n)$ ) and output ( $h_w(x)$ ) are now real numbers instead of binary values and

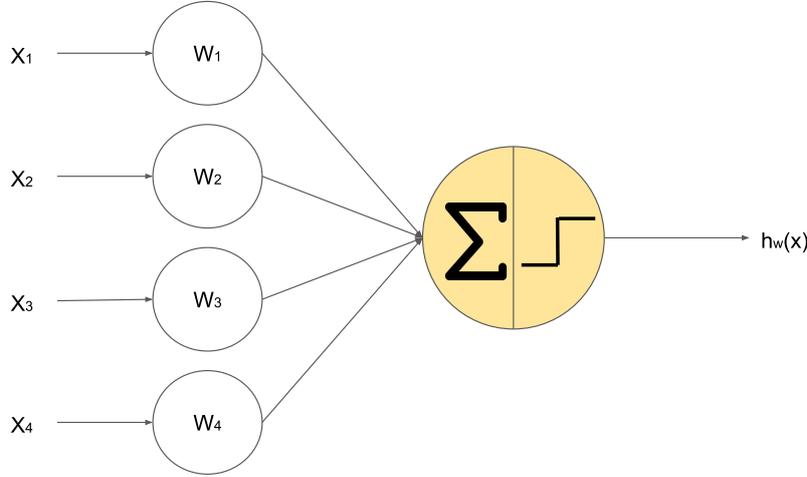


Figure 3.1: Description linear threshold unit [Géron, 2019]

each connection is associated with a weight (see figure 3.1). The LTU computes a weighted sum of its inputs connections ( $z = w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_nx_n = \mathbf{w}^T \cdot \mathbf{x}$ ).

Then applies a step function to the sum of the inputs  $z$  and gives as output the result:  $h_w(\mathbf{x}) = \text{step}(\mathbf{w}^T \cdot \mathbf{x})$ . The Heaviside step function is the most common function used.

$$\text{heaviside}(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases} \quad (3.1)$$

A Perceptron is composed of multiples LTUs with all inputs connected to each of them. The inputs connect to a pass-through neuron (input neuron), giving the same value to the output. Also, a Bias neuron is added, which always outputs the value  $x_0 = 1$  and connects to each LTUs. A Perceptron with three inputs and one output is presented in figure 3.2. This perceptron can predict a tri-dimensional problem's output value by adding all outputs in a final neuron. The same architecture can be used for classification problems without the last neuron that sums all output values from each LTU. This network can classify instances simultaneously into three different binary classes, in other words, a multioutput classifier.

However, each connection's weight is unknown and must be learned from previously know data presented to the network to train it. The Perceptron training algorithm proposed by Frank Rosenblatt was inspired by Hebb's rule [Hebb, 2005], suggesting that when a biological neuron often triggers another neuron, their connection grows stronger. That is increasing the connection weight between two neurons whenever they have the same output. Perceptrons train by feeding one training instance at a time; for each instance, it makes its predictions. For every output neuron that produced a wrong prediction, it reinforces the connection weights

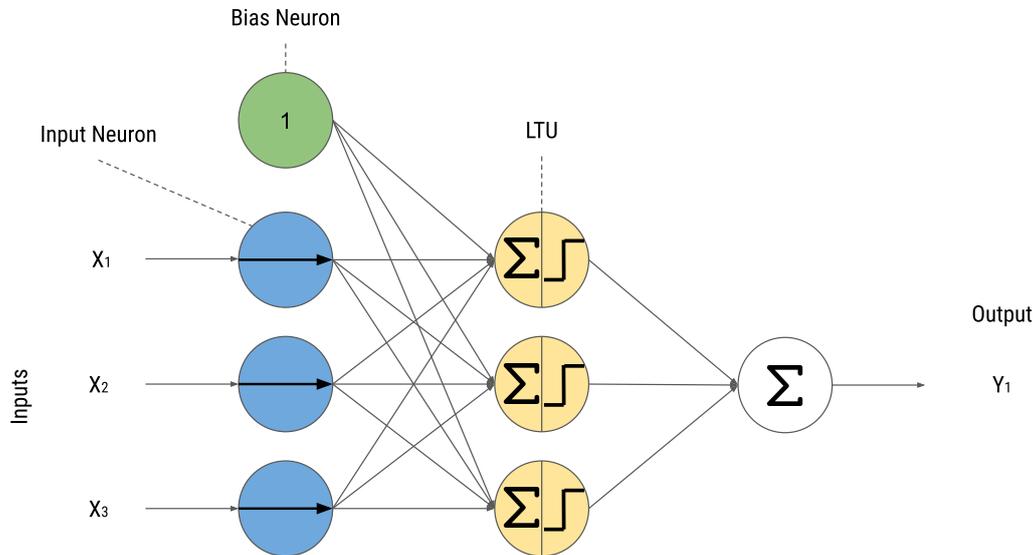


Figure 3.2: Perceptron diagram for regression [Géron, 2019].

from the inputs that contributed to the correct prediction, meaning that it considers the error made by the network.

$$w_{i,j}^{t_{i+1}} = w_{i,j}^{t_i} + \eta(\hat{y}_j - y_j)x_i \quad (3.2)$$

Where,  $w_{i,j}$  is the connection weight between the  $i$ th input neuron and the  $j$ th output neuron,  $x_i$  is the  $i$ th input value of the current training instance,  $\hat{y}_j$  is the output of the  $j$ th output neuron for the current training instance,  $y_j$  is the target output of the  $j$ th output neuron for the current training instance, and  $\eta$  is the learning rate.

The decision boundary produced per each output neuron is linear, so Perceptrons are incapable of learning complex patterns. However, some of the limitations of this architecture are solved by stacking multiple Perceptrons. The resulting ANN is called Multi-Layer Perceptron (MLP).

### 3.2.2 Multi-layer perceptron

An MLP is composed of one (pass-through) input layer, one or more layers of LTUs called hidden layers, and one final layer of LTUs called the output layer (see figure 3.3). Every layer except the output layer includes a bias neuron fully connecting each neuron to the next layer. When an ANN has two or more hidden layers, it is called a deep neural network (DNN) [Géron, 2019].

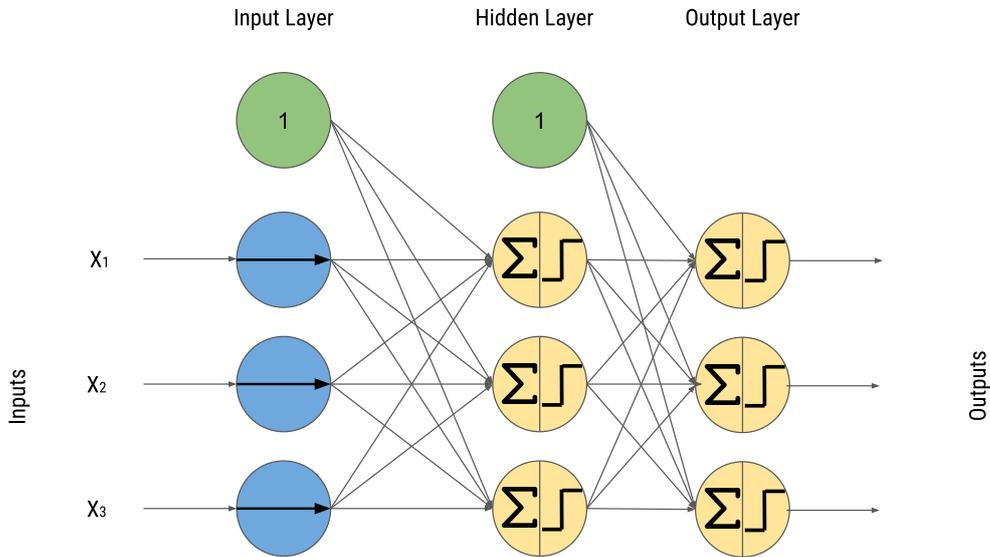


Figure 3.3: Perceptron diagram for regression [Géron, 2019].

In 1986, Rumelhart et al. [Rumelhart et al., 1985] introduced the backpropagation training algorithm, finding a way to train MLPs with success. For each training instance, the algorithm the network computes the output of every neuron in each consecutive layer (forward pass). Next, it measures the network's output error, the difference between the desired output  $y_j$  and the actual output of the network  $\hat{y}_j$ , and it computes how much each neuron in the last hidden layer contributed to each output neuron's error. This process continues to measure each neuron's error contributions in the previously hidden layer until the algorithm reaches the input layer. This reverse pass measures the error gradient efficiently across all the network connection weights by propagating the error gradient backward in the network.

In order for this algorithm to work properly, the authors made a change to the MLP's architecture, replacing the Heaviside function with the logistic function (eq. 3.3). This change was essential because the Heaviside function contains only flat segments, so there is no gradient to work with. In contrast, the logistic function has a well-defined nonzero derivative in the entire domain, allowing some progress at every step.

$$\sigma(y) = 1/(1 + \exp(-y)) \quad (3.3)$$

However, in the backpropagation algorithm, gradients often get smaller as it progresses to the lower layers. This is called the vanishing gradients problem, where training never converges to a good solution because the gradient descent update leaves the lower layer connection weights unchanged. Using a logistic activation function (see figure 3.4), when the output becomes large (positive or negative), the function has an asymptotic behavior approaching at 1 or

0, meaning that the derivative is extremely close to 0. Then, when backpropagation encounters this scenario, it has a little gradient to propagate back through the network, so there is really no variation for the lower layers. But, in 2010 Glorot and Bengio [Glorot and Bengio, 2010] introduced the weight initialization technique namely random initialization (before all weights were initialized at 0) using a normal distribution with a mean of 0 and standard deviation of  $\sigma = \sqrt{2/(n_{inputs} + n_{outputs})}$  when using logistic activation function, where  $n_{inputs}$  and  $n_{outputs}$  are the number of input and output connections for the layer whose weights are being initialized. With this strategy, the training is speeds up considerably.

The vanishing/exploding gradients' problems were in part due to the activation function. Choosing the sigmoid activation functions as biological neurons work, most people had been choosing them. Other types of activation functions behave much better in deep neural networks, avoiding the saturation parts, and being faster to compute their gradient.

### 3.2.3 Activation functions

The backpropagation algorithm could use other activation functions, instead of the logistic function. Different activation functions have been used in MLP, changing their architecture and, by consequence, the network's output prediction. Depending on the complexity of the problem, some activation function may work better than others.

The most common activation functions are:

- The hyperbolic tangent function *tanh*  
As the logistic function it has an S-shape (see figure 3.4), continuous and differentiable, but its output value ranges from  $-1$  to  $1$ , speeding up the convergence by making each layer's output more or less normalized.

$$\tanh(y) = 2\sigma(2y) - 1 \quad (3.4)$$

- The rectified linear units (ReLU)  
It is continuous but not differentiable at  $y = 0$ , the change of slope is abrupt, making the gradient descent bounce around. However, in practice, the ReLU activation function generally works better in ANNs, because it does not have a maximum output value as the other activation functions [Géron, 2019].

$$\text{ReLU}(y) = \max(0, y) \quad (3.5)$$

ReLU activation function outperforms these other activation functions. However, they have the problem known as the dying ReLUs: during the training of the network, some neurons effectively die, meaning that the ReLU function always outputs 0. If a neuron's weight gets updated and the weighted sum  $w^T \cdot x$  is negative, the neuron will start outputting 0. If the

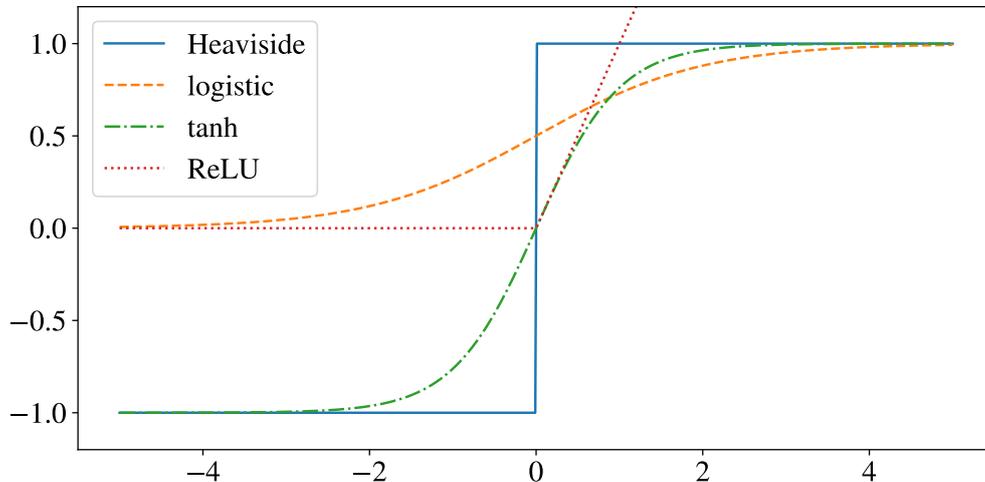


Figure 3.4: Activation functions used in LTUs [Géron, 2019].

neuron reaches this state, it is unlikely to come back since the ReLU function's gradient is 0 when its input is negative.

Some variants of the ReLU function have been developed to solve this problem, keeping the advantages of fast convergence and solving the "deads" neurons. The first variant of this activation function is the *leaky ReLU*. This function is defined as:

$$\text{LeakyReLU}_\alpha(y) = \max(\alpha y, y) \quad (3.6)$$

The hyperparameter  $\alpha$  defines the slope of the function for  $z < 0$ , and it is typically set to small values like 0.01. This small slope ensures that leaky ReLUs functions never die. Xu et al. [Xu et al., 2015] compared several variants of the ReLU activation function, and they concluded that the variants of leaky ReLU function outperform the ReLU activation function. Also, they tested different settings for  $\alpha$  between high (huge leak)  $\alpha = 0.2$  and small (small leak)  $\alpha = 0.01$ , and high leak result in a better performance than a small leak. They also evaluated the *randomized leaky ReLU* (RReLU), where  $\alpha$  is selected randomly in a given range during training, and it is fixed to an average value during testing. It also has a good performance and seemed to act as a regularizer, reducing the risk of overfitting the network in the training set. Finally, they evaluated the parametric *leaky ReLU* (PReLU), where  $\alpha$  becomes a parameter that can be modified by backpropagation like the neuron's weight. This last variation outperforms ReLU on large image datasets, but on smaller datasets, it runs the risk of overfitting the training set.

The improvement achieved by the variations of leaky ReLU function does not focus the problem of not being differentiable at  $y = 0$ . Until 2015 when Clevert et al. [Clevert et al., 2015] proposed a new activation function the *exponential linear unit* (ELU):

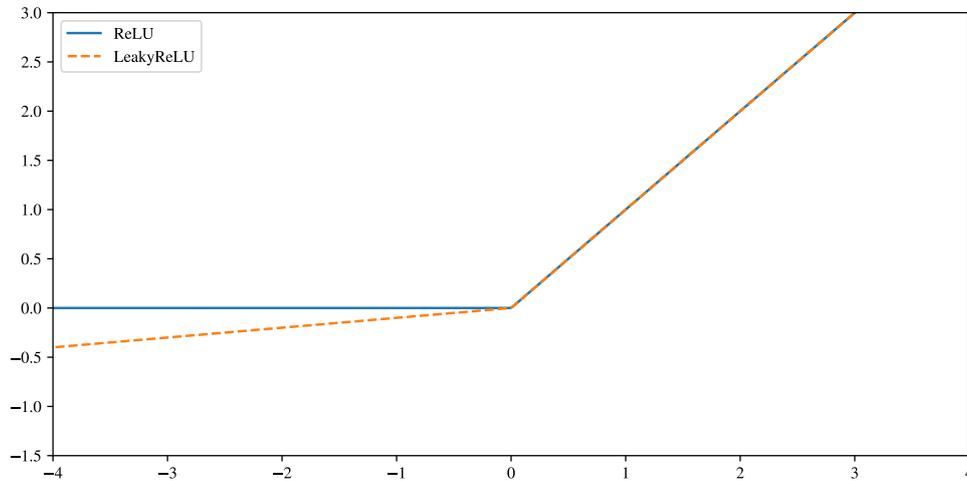


Figure 3.5: Leaky ReLU activation function,  $\alpha = 0.1$  [Géron, 2019].

$$ELU_{\alpha}(z) = \begin{cases} \alpha(\exp(z) - 1) & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases} \quad (3.7)$$

This activation function presents better performance than all the ReLU family, reducing the training time, and the neural network performed better on the test set. In figure 3.6 is represented the function with  $\alpha = 1$ . The major differences compared to ReLU functions are: First, it has a non zero gradient for  $y < 0$ , which avoids the dying units presented in ReLU, Second, it return negative values if  $y < 0$ , which allows the unit to have an average output closer to 0, and Third, the function is smooth in the entire domain, including around  $y = 0$ , which helps speed up the gradient descent since it is differentiable in  $z = 0$ .

For all the ReLU family, He et al. [He et al., 2015] provided similar strategies as Glorot and Bengio to initialize the hidden layer neurons' connection weight.

In summary, to select an activation function for the hidden layers of a deep neural network, in general,  $ELU > \text{leaky ReLU (and its variants)} > \text{ReLU} > \text{tanh} > \text{logistic}$ . If the network's training consumes high computational time, then leaky ReLU is preferred over ELU function [Géron, 2019].

### 3.2.4 Deep neural network optimizer

A large deep neural network takes a lot of time to be trained. Some strategies have been described to speed up training: using a good activation function and applying a good initialization strategy for the connection weights. Another strategy that can be useful is using

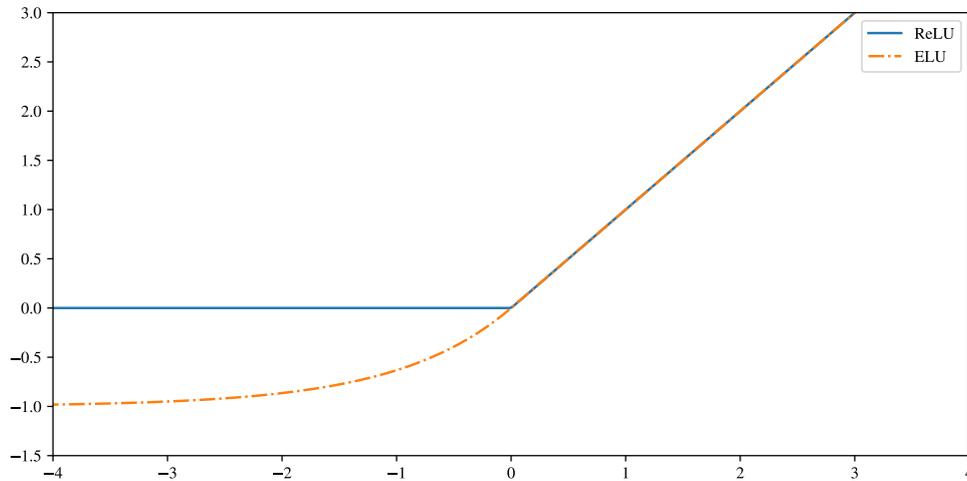


Figure 3.6: ELU activation function,  $\alpha = 1$  [Géron, 2019].

different optimizer algorithm than the Gradient Descent. Some of the most popular optimizers are presented next.

The Gradient Descent updates the weights  $\theta$  by directly subtracting the gradient of the cost function  $J(\theta)$  with regards to the weights ( $\nabla_{\theta}J(\theta)$ ) multiplied by the learning rate  $\eta$ . The equation is:

$$\theta^{i+1} = \theta^i - \eta \nabla_{\theta} J(\theta^i) \quad (3.8)$$

Selecting a low learning rate of  $\eta$  could lead to small steps and advance slowly to the optimum taking a lot of steps and consuming huge computational time. Selecting a large  $\eta$  could lead to not convergence of the algorithm, having big steps oscillating around the optimum.

To improve the optimization problem's running time, algorithms have been developed, presenting a faster convergence than Gradient Descent. Some of the most popular algorithms are presented below.

### 3.2.4.1 Momentum optimization

The momentum optimization algorithm is proposed by Polyak [Polyak, 1964], where the main idea of this method is considering the momentum that could reach a ball rolling down a slope of a surface: it starts slowly. But it will increase the velocity due to the inertial momentum until it eventually reaches terminal velocity.

Different than Gradient Descent, Momentum optimization takes into account the previous gradients. At each iteration, it adds the local gradient multiplied by the learning rate  $\eta$  to the momentum vector  $m$ , and it updates the weights by simply subtracting this momentum vector (see algorithm 1). The algorithm introduces  $\beta$  as a new hyperparameter, called momentum. To simulate a friction mechanism and create a resistance force that prevents the momentum from growing too large, which can take values between 0 (high friction) and 1 (no friction). Usually, it is set to  $\beta = 0.9$

---

**Algorithm 1** Momentum optimization algorithm

---

- 1:  $m \leftarrow \beta m + \eta \nabla_{\theta} J(\theta)$
  - 2:  $\theta \leftarrow \theta - m$
- 

This optimization algorithm with  $\beta = 0.9$  makes Momentum optimization run ten times faster than Gradient Descent, allowing Momentum optimization to get out from plateaus faster than Gradient Descent. In cases when the inputs have very different scales, the cost function will look like an elongated bowl, The Gradient Descent goes down quickly in the steep slope, but then it takes a very long time to go down the valley. In contrast, Momentum optimization will roll down the bottom of the valley faster until it reaches the optimum [Géron, 2019].

The one disadvantage of Momentum optimization is that it adds another hyperparameter to tune to the network.

### 3.2.4.2 Nesterov Accelerated Gradient (NAG)

A variant of the Momentum optimization is proposed by Nesterov [Nesterov, 1983], measuring the gradient of the cost function not at the current position, but a little ahead in the direction of the moment at  $\theta + \beta m$  (see algorithm 2).

---

**Algorithm 2** Nesterov Accelerated Gradient algorithm

---

- 1:  $m \leftarrow \beta m + \eta \nabla_{\theta} J(\theta + \beta m)$
  - 2:  $\theta \leftarrow \theta - m$
- 

This small change in the evaluation of the cost function works because, in general, the momentum vector is toward the optimum, meaning that it will be more accurate to measure the gradient a bit closer to the optimum than measuring the gradient at the local position. Adding up these small improvements, NAG is significantly faster than the regular Momentum optimization [Géron, 2019].

### 3.2.4.3 AdaGrad

Introduced by Duchi et al. [Duchi et al., 2011], the AdaGrad algorithm focuses on solving the problem of having different scales in the cost function dimensions, creating an elongated bowl. For this problem it is used Gradient Descent, it starts fast when the gradient is high and

then goes slow when the gradient is low. AdaGrad detects this problem and scales down the gradient vector along the dimension with a larger scale.

---

**Algorithm 3** AdaGrad algorithm
 

---

- 1:  $s \leftarrow s + \nabla_{\theta} J(\theta) \otimes \nabla_{\theta} J(\theta)$
  - 2:  $\theta \leftarrow \theta - \eta \nabla_{\theta} J(\theta) \oslash \sqrt{s + \epsilon}$
- 

The AdaGrad algorithm (see algorithm 3) first step accumulates the square of the gradients into the vector  $s$  (the  $\otimes$  symbol represents the element-wise multiplication). This vectorized form is equivalent to computing:

$$s_{i+1} = s_i + \left( \frac{\partial J(\theta)}{\partial \theta_i} \right)^2$$

for each element  $s_i$  of the vector  $s$ . If the cost function presents a high slope in the  $i^{\text{th}}$  dimension, then  $s_i$  will get larger and larger at each iteration.

The second step is similar to Gradient Descent, but in this case, the gradient vector is scaled down by a factor of  $\sqrt{s + \epsilon}$ , the  $\oslash$  symbol represents the element-wise division, and  $\epsilon$  is a term used to avoid division by zero. The vectorized form presented in the second step of the algorithm is equivalent to:

$$\theta_{i+1} = \theta_i - \eta \frac{\frac{\partial J(\theta)}{\partial \theta_i}}{\sqrt{s_i + \epsilon}} \quad (3.9)$$

One advantage of this method is that it requires less tuning of the learning rate hyperparameter. However, AdaGrad often performs well for simple quadratic problems, but for training neural networks, it often stops too early [Géron, 2019].

### 3.2.4.4 RMSProp

RMSProp [Tieleman and Hinton, 2012] solves the problem presented by AdaGrad that slows down a bit too fast and never converges to the global minimum. RMSProp accumulates only the gradients from the most recent iterations (as opposed to all the gradients since the beginning of training). It is done by using an exponential decay in the first step (see algorithm 4).

---

**Algorithm 4** RMSProp algorithm
 

---

- 1:  $s \leftarrow \beta s + (1 - \beta) \nabla_{\theta} J(\theta) \otimes \nabla_{\theta} J(\theta)$
  - 2:  $\theta \leftarrow \theta - \eta \nabla_{\theta} J(\theta) \oslash \sqrt{s + \epsilon}$
- 

The decay rate  $\beta$  is typically set to 0.9, it is a new hyperparameter to tune, but this default value often works well. Except for very simple problems, this optimizer almost always performs

much better than AdaGrad. It also generally performs better than Momentum optimization and NAG [Géron, 2019].

### 3.2.4.5 Adam optimization

Kingman and Ba in 2015 [Kingma and Ba, 2014] introduces the called Adam algorithm, which stands for adaptive moment estimation, combining the ideas of Momentum optimization and RMSProp. From Momentum optimization, it keeps track of an exponential decay average of the past gradients, and from RMSProp, it keeps track of an exponential decay average of past squared gradients (see algorithm 5).

---

#### Algorithm 5 Adam Optimization algorithm

---

- 1:  $m \leftarrow \beta_1 m + (1 - \beta_1) \nabla_{\theta} J(\theta)$
  - 2:  $s \leftarrow \beta_2 s + (1 - \beta_2) \nabla_{\theta} J(\theta) \otimes \nabla_{\theta} J(\theta)$
  - 3:  $m \leftarrow \frac{m}{1 - \beta_1^T}$  ▷ T is the iteration number, start at 1
  - 4:  $s \leftarrow \frac{s}{1 - \beta_2^T}$
  - 5:  $\theta \leftarrow \theta - \eta m \oslash \sqrt{s + \epsilon}$
- 

Steps 1,2, and 5 are similar to both Momentum optimization and RMSProp. The difference is found in step 1, which computes an exponentially decaying average rather than an exponentially decaying sum. Steps 3 and 4 are used to boost m and s at the beginning of the training since they are initialized at 0. Both hyperparameters are initialized to  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ . Also, like AdaGrad, Adam is an adaptive learning rate algorithm, requiring less tuning of the learning rate hyperparameter  $\eta$ . Adam outperforms all other optimization algorithms and is the one used to train all deep neural networks in this research.

### 3.2.5 Overfitting problems

Depending on the size of a deep neural network can reach thousands of parameters. Having this amount of parameters, a deep neural network can learn properly any kind of complex behavior, training with the right amount of data. But some times as the network trying to minimize the error in the training set, overfit its response. In other words, the network will predict with minimal error data that was used from training or seen already, but for new data (test set), the network performs poorly.

#### 3.2.5.1 L1 and L2 regularization

Different strategies can be used to not leave the network to overfit in the training set. The first strategy is to use regularization norms to constrains the connection weights. By adding a parameter norm to the objective function  $J(\theta)$

$$\tilde{J}(\theta; X, y) = J(\theta; X, y) + \alpha\Omega(\theta) \quad (3.10)$$

Where,  $\alpha$  is a hyperparameter that weight the contribution of the norm penalty  $\Omega$ . If  $\alpha = 0$ , no regularization is applied, and the original cost function is obtained.

Two popular regularization norms are L2 and L1. L2 regularization is also known as Tikhonov regularization, replacing the term  $\Omega(\theta)$  by:

$$\Omega(\theta) = \frac{1}{2} \|w\|_2^2 \quad (3.11)$$

Now the objective function 3.10 is written as:

$$\tilde{J}(w; X, y) = J(w; X, y) + \alpha \frac{1}{2} \|w\|_2^2 = J(w; X, y) + \frac{\alpha}{2} w^T w \quad (3.12)$$

This term forces to keep the connection weights as small as possible while fitting the training data. However, setting  $\alpha$  too high will make the weights approach zero.

The L1 regularization is similar to L2, but the function omega has a slight change. It uses the L1 norm of the weight vector instead of half the square of the L2 norms (see equation 3.13).

$$\tilde{J}(w; X, y) = J(w; X, y) + \alpha \|w\|_1 \quad (3.13)$$

This regularization tends to completely eliminate the weights of the least important features, setting them to zero. In other words, L1 regularization performs a feature selection and outputs a sparse response.

### 3.2.5.2 Dropout regularization

Dropout [Hinton et al., 2012, Srivastava et al., 2014] is a technique used to prevent the overfitting of the neural network by taking out (dropout) random neurons of the network while training. Every neuron (including input neurons but excluding output neurons) has a probability  $p$  of being temporarily ignored during a training step, but it may be active during the next step. A new hyperparameter  $p$  is introduced and is called dropout rate, and is set to 50% normally.

Dropping out random neurons at every training step forces the network to not rely on just a few input neurons. They take into account each of their input neurons. The Resulting neural network ends up being less sensitive to slight changes in the inputs; in other words, a more robust network[Géron, 2019].

### 3.2.5.3 Early stoppage

One of the most simple techniques to avoid overfitting the training set is interrupt the training when the performance of the network on the validation set reaches a minimum value or setting a number of training steps (e.g., 1000 iterations), saving this last model and using it to predict in the validation set[Géron, 2019].

## 3.3 Tuning hyperparameters

Deep neural networks are a powerful tool with too many hyperparameters to tune: activation function, number of hidden layers, neurons per layers, optimization, regularization, etc. (Not taking into account the network topology or how the neurons are connected). Depending on the selection of these hyperparameters, the network could fit properly or not the data, consuming many computational resources to train or not being able to predict with accuracy the target value. Tuning the hyperparameters is the most important task because it will define the performance of the network.

The most simple strategy to find the right hyperparameters is to use a grid search on a searching space. Then, measuring their Mean Squared Error (MSE) at each hyperparameters selecting the one that returned the minimum MSE in the validation set. This strategy is really simple to implement. But since there are many hyperparameters to tune, it will explore a tiny part of the searching space in a reasonable amount of time[Géron, 2019].

Other proposed strategies have a much better result than grid search, such as Random search [Bergstra and Bengio, 2012]. Random search selects random points uniformly distributed inside the searching space of the hyperparameters. By only making these random guesses, the number of trials is reduced compared to grid search and requires less computational time [Bergstra and Bengio, 2012].

However, if the model is extremely costly, doing random search becomes an inefficient strategy to find good hyperparameters for the neural network. This inefficiency is because first, the neural network is built, then it has to be trained, and finally, measure its performance in a test set. In this case, another technique is better to implement called Bayesian Optimization.

### 3.3.1 Bayesian optimization for hyperparameter tuning

The Bayesian optimization developed by Louppe [Louppe, 2017]construct another model of the searching space for hyperparameters. With this model is estimated how the performance varies, changing the hyperparameters. Each point (a combination of hyperparameters) evaluated is known by the model and its performance. The Bayesian optimization will then select a new combination of hyperparameters that will improve the performance the most, in a new region of the searching space that has not been explored yet or near the evaluated points.

The optimization problem has interest in solving  $x^* = \operatorname{argmin} f(x)$  under the constraints that:  $f$  is a black box for which no closed form is known nor its gradients;  $f(x)$  is expensive to evaluate; and evaluations of  $f(x)$  may be noisy.

A loop of 3 steps is performed for some iterations until the problems satisfy all the constraints.

for  $t = 1:T$

- given observations  $(x_i, y_i = f(x_i))$  for  $i = 1:t$ , build a probabilistic model for the objective  $f$ . Integrate out all possible true functions, using Gaussian process regression.
- Optimize a cheap acquisition/utility function  $u$  based on the posterior distribution for sampling the next point.  $x_{t+1} = \operatorname{argmin} u(x)$ . Exploit uncertainty to balance exploration against exploitation.
- Sample the next observation  $y_{t+1}$  at  $x_{t+1}$

Three different acquisition functions  $u(x)$  are defined by the users and they specify which sample  $x$ , should be tried next:

- Expected improvement (selected for this study):

$$-EI(x) = -\mathbf{E}[f(x) - f(x_t^+)] \quad (3.14)$$

- Lower confidence bound:

$$LCB(x) = \mu_{GP}(x) + \kappa\sigma_{GP}(x) \quad (3.15)$$

- Probability of improvement:

$$-PI(x) = -P(f(x) \geq f(x_t^+) + \kappa) \quad (3.16)$$

Where  $x_t^+$  is the best point observed so far,  $\mu_{GP}$  and  $\sigma_{GP}$  are the mean value and the standard deviation of the Gaussian process, respectively. In most cases, acquisition functions provide knobs (e.g.,  $\kappa$ ) for controlling the exploration-exploitation trade-off.

- search in regions where  $\mu_{GP}$  is high (exploitation).
- Probe regions where uncertainty  $\sigma_{GP}$  is high (exploration).

Bayesian optimization for hyperparameter tuning is presented as an algorithm by Pedersen [Pedersen, 2016], used to optimize the hyperparameters of a Convolutional Neural Network for a classification problem. This algorithm is taken as reference and then adapted for regression problems (see algorithm 6).

Both techniques, Random search, and Bayesian optimization, are performed using open source library Scikit-Optimize [Head et al., 2018] for sequential model-based optimization in Python.

**Algorithm 6** Bayesian optimization for hyperparameter tuning

- 
- 1: Definition of hyperparameter searching space
  - 2: **for**  $i = 1 : n$  **do**
  - 3:   Sample Gaussian Model for hyperparameters
  - 4:   Create DNN using hyperparameters
  - 5:   Train the DNN
  - 6:   Evaluate performance of DNN
  - 7:   Calculate Root Mean Squared Error on validation set
  - 8:   Save the model with minimum RMSE
  - 9:   Update Gaussian Model
- 

**3.3.2 Random search vs. Bayesian optimization for hyperparameters optimization**

Lets define a simple benchmark function:

$$f(x) = 20\cos(x)\sin(x) + N(0, 1) \quad (3.17)$$

where,  $N(0, 1)$  is a normal random noise added to the function with mean 0 and standard deviation 1 (see figure 3.7).

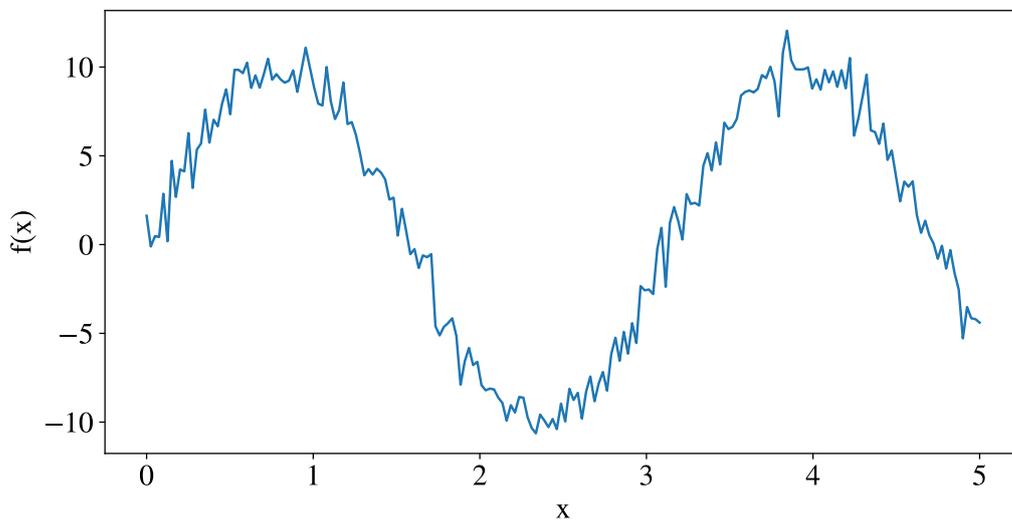


Figure 3.7: Benchmark function for training a Deep Neural Network.

This benchmark function will train a deep neural network and find the best combination of hyperparameters for this specific problem using Random search and Bayesian optimization and compare their performance. The hyperparameters to optimize will be: the learning rate, number of hidden layers, number of neurons per hidden layers (equal at every hidden layer),

and activation function (also equal for all neurons), the searching space boundary for each hyperparameter is described in table 3.1.

Table 3.1: Hyper-parameter searching space.

Hyperparameter	Range	Type
Learning rate	[1e-5,1e-3]	Real
No. hidden layers	[1, 8]	Integer
No. of nodes per layer	[1,1000]	Real
Activation function per neuron	[ReLU, LeakyReLU, logistic, tanh]	Categorical

The deep neural network model used to replace the benchmark function is a sequential fully-connected network; Adam is used as the optimizer. The data-set contains 400 points selected randomly from the function, 90% of the data is for training the network, and the other 10% for validation. For regression problems, the Mean Squared Error is the most commonly used metric (see equation 3.18). The MSE is calculated in the validation set, and the model with the minimum MSE is the best model to replace our function.

$$MSE = \frac{1}{m} \sum_{i=1}^m (\hat{y} - y)^2 \quad (3.18)$$

Both algorithms perform 30 calls, and each call, the network is trained at 100, 200, 300, 400, and 500 epochs (forward and backward pass). For the Bayesian optimization, the acquisition function is EI (Expected improvement). The initial hyperparameters are [0.02, 1, 100, *logistic*] (learning rate, number of layers, number of neurons per layer, and activation function, respectively).

Results after executing the hyperparameter optimization with both methods changing the number of epochs to train the network are presented in table 3.2. For both methods, as the number of epochs increases, the MSE is lower as expected. For Random search, the activation function most frequently found was ReLU, but Bayesian optimization was LeakyReLU. The number of hidden layers in Random search varies between 1 and 4 without an apparent behavior, but for Bayesian optimization, the number of hidden layers increases as the epochs are higher.

Table 3.2: Tuning hyperparameters using Random search and Bayesian Optimization varying the number of training epochs with a fixed number of *iteration* = 30.

Epochs	Random Search		Bayesian Optimization	
	Hyperparameters	MSE	Hyperparameters	MSE
100	[0.032543911150884904, 3, 581, 'relu']	0.007822168	[0.0019661053770823576, 2, 1000, 'relu']	0.0025471
200	[0.019567645053920467, 2, 62, 'relu']	0.003011492	[0.0009423591698125554, 4, 465, 'relu']	0.00284786
300	[0.0007949767776991855, 4, 913, 'relu']	0.002568431	[0.0011291334428399334, 4, 465, 'relu']	0.00209756
400	[0.004764743142679584, 1, 580, 'relu']	0.002536315	[0.00018701621268708032, 6, 533, 'relu']	0.00170228
500	[0.0001011533910949556, 3, 897, 'relu']	0.001911177	[0.0003521327997465734, 8, 988, 'relu']	0.00196455

Bayesian optimization tends to create a more complex DNN model. The number of epochs

increases, meaning that to reduce the MSE, they search a model that will predict with more accuracy, but without considering the training time. This problem can be solved by reducing the searching space boundaries, forcing the algorithm to find a solution with less hidden layers.

Increasing the epochs to train the DNN makes a huge impact on the computational time used. A second analysis changes the number of iteration made by the optimization algorithm, leaving the training epochs constant at 100 iterations. Results are presented in table 3.3. By increasing the number of iterations, both algorithms decreased the MSE. However, Bayesian optimization presents the lowest MSE compared to random search. With more training points, the Gaussian Process can construct a more accurate representation of the searching space model and improve the network's performance.

Table 3.3: Tuning hyperparameters using Random search and Bayesian optimization varying the number of iteration with a fixed training  $epochs = 100$ .

Random Search			Bayesian Optimization	
Iterations	Hyperparameters	MSE	Hyperparameters	MSE
11	[0.02, 1, 100, 'relu']	0.016739808	[0.02, 1, 100, 'relu']	0.01995781
30	[0.032543911150884904, 3, 581, 'relu']	0.007822168	[0.0019661053770823576, 2, 1000, 'relu']	0.0025471
50	[0.001893412804339247, 4, 715, 'relu']	0.004981555	[0.004141675453756694, 6, 341, 'relu']	0.00211608

After comparing both algorithms, Bayesian optimization has a better performance than random search [Pedersen, 2016]. However, this optimization method consumes a lot of computational time and Hardware resources. A balance between the number of iteration, training epochs, and boundaries is the best way to reduce the calculation time and find a high-performance network. Bayesian optimization algorithm tunes the hyperparameters of all the networks used in this research.

## 3.4 Discontinuity modeling for wind turbine simulated loads

### 3.4.1 Pitch control problems for steady and unsteady simulations

The simulation tools FAST is a "black box" mathematical model that calculates the response of the wind turbine represented as a beam model. This simulation tool couples the aerodynamic, hydrodynamic, energy, and control models to calculate a dynamic response in the time domain. The controller used in this study, as mentioned before, is not optimized for this specific case and is adapted from the original version created for the HAWC2 simulation tool. This controller works perfectly for FAST simulations when the wind field is uniform. However, using a steady or unsteady wind field produces a different behavior in the blade pitch control, affecting the wind turbine's generated power for different wind speeds.

This effect appears regarding the maximum aero-elastic loads extracted at one node of the beam model simulation. Figure 3.8 shows the load distribution acting in  $x$ -the direction of the blade coordinate system in the span location 1 between wind speeds from 4 to 25m/s. As the wind speed increases from cut-in to rated wind speed (from 4m/s to 11.4m/s). The

load increases almost linearly until reaching a maximum value. Then it starts to descend as the wind speed continues to increase until it reaches a jump discontinuity passed  $16m/s$ , continuing to decrease the load magnitude until reaching cut-out wind speed. This behavior in the load is due to the tuning wind turbine blade's control pitch.

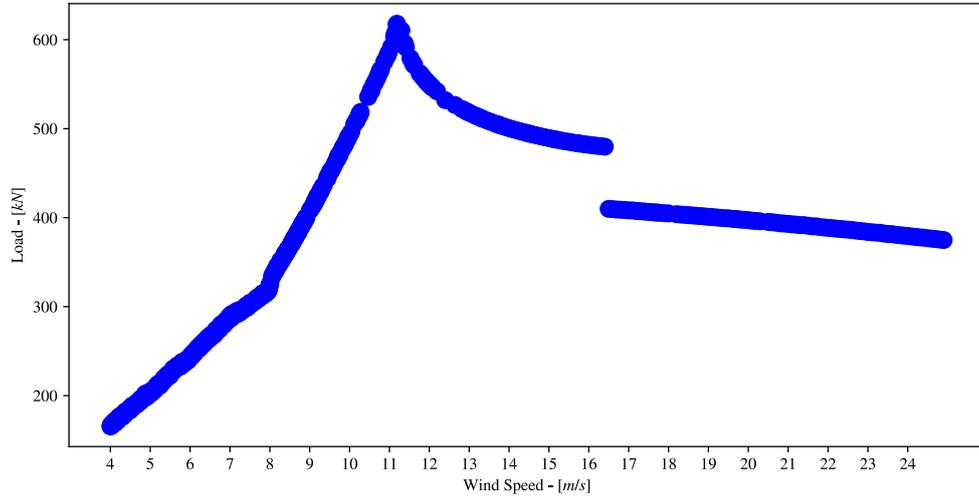


Figure 3.8: Load  $Spn1FLxb1$  (Span location 1, force in lift direction for blade 1) distribution between wind speeds from  $4m/s$  to  $25m/s$ .

The same effect appears regarding the dynamic response in an unsteady simulation. Figure 3.9 shows the wind turbine not producing energy for a mean wind speed of  $19m/s$ , meaning that it is not rotating at a velocity of  $9.6rpm$ . The controller sets the wind turbine to an angle of  $82^\circ$ , whether the wind surpasses the cut-out speed. For the entire aero-elastic simulation, the wind turbine has this constant pitch angle, but it has no breaks acting in the rotor, meaning that the wind turbine continues to rotate but at a lower  $rpm$ .

These presented cases produce a jump discontinuity in the wind turbine blade response, making the model's prediction more complex. Surrogate models could approximate this black box model, reducing the computational time to predict a new output. However, as the model contains different behaviors per region, one surrogate model may not accurately predict its discontinuity.

To solve the problem of jump discontinuities, Bettebghor et al. [Bettebghor et al., 2011] trained surrogate models in different regions and combined their response using Gaussian mixture models. Also, to select the regions to train, the authors used the unsupervised learning algorithm Expectation-maximization clustering. The performance of the surrogate model is increased using this model. However, the number of surrogate models increases with the number of discontinuities, and this method predicts values inside a jump in the original model.

Niutta et al. [Niutta et al., 2018] presents a different approach by identifying the disconti-

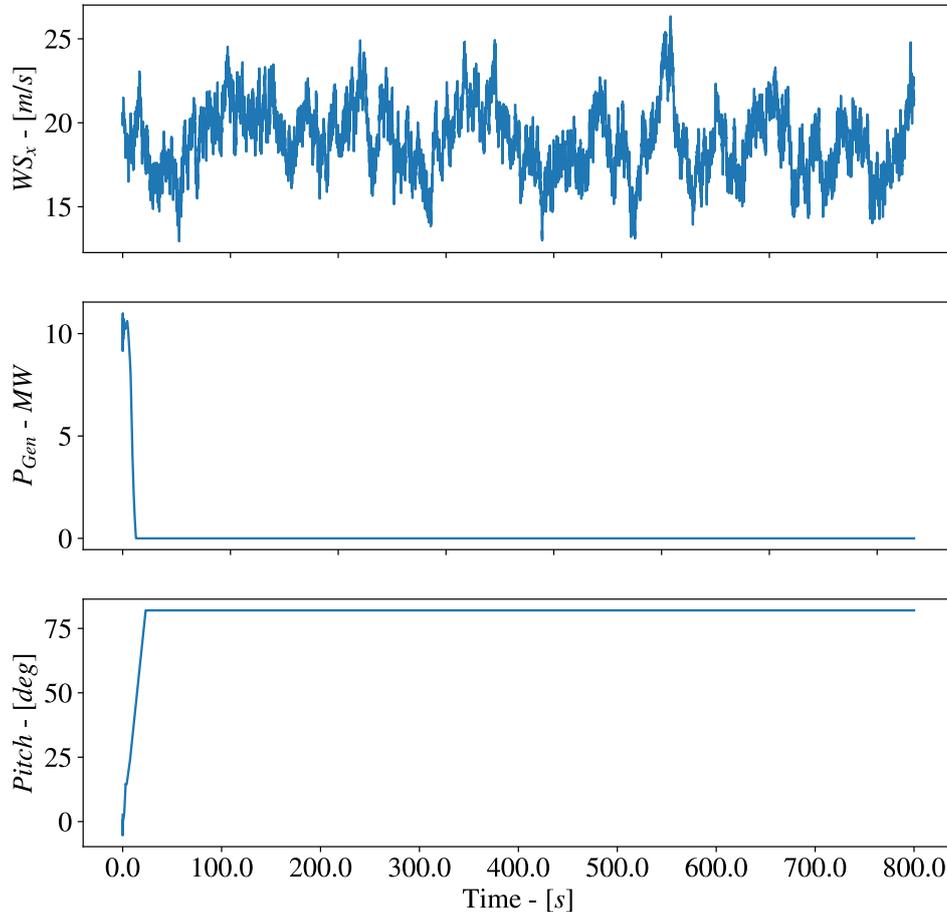


Figure 3.9: Aero elastic simulation for 19m/s normal turbulence wind speed in X-direction, power generated and pitch angle of the blade (from up to down respectively).

nities using a support vector machine (SVM) for classification and then using local surrogate models as Gaussian Process (Kriging) and a 2nd order regression. This method improves the surrogate models' global performance by not predicting values between discontinuities or jumps from the original model. However, many surrogate models have to be trained in each local region.

Another author to propose a different approach to solve this problem was Chen et al. [Chen et al., 2018] in his work of "Rational Neural Networks for Approximating Jump Discontinuities of Graph Convolution Operator". The author integrates rational approximation with a neural network to approximate jump discontinuities in two theoretical and real-world scenarios. This method solves the problem of using multiples surrogate models by using one

neural network. However, around the jump discontinuity, the neural network predicts wrong values that do not exist in the original mathematical model.

A new approach for approximating models with jump discontinuity using deep neural network is presented in this research. As done by the other approaches to measure the performance of the proposed approach, it is compared against state of art regression surrogate models:

- Linear Regression (LR)
- Polynomial Regression (PR) [Stigler, 1974]
- LASSO [Tibshirani, 1996]
- Epsilon-Support Vector Regression (SVR) [Vapnik et al., 1997]
- Ridge Regression (RR) [Ng, 2004]
- Regression Deep Neural Network (RDNN)
- Gaussian Process Regression with Radial Basis Function kernel (GP)

### 3.4.2 Automated Discontinuity Modeling Deep Neural Network (ADM-DNN)

The proposed approach identifies the jump discontinuities by performing a classification of the different regions using a Gaussian Mixture Model and trains a neural network, including the inputs parameters and classification values.

#### 3.4.2.1 Gaussian mixture model for discontinuity modeling

A Gaussian mixture model (GMM) is a probabilistic model that assumes that the instances or features were generated from a mixture of several Gaussian distributions with unknown parameters. GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm.

GMM's equation is a weighted sum of  $N$  components Gaussian densities given by:

$$p(x|w_i, \mu_i, \Sigma_i) = \sum_{i=1}^N w_i g(x|\mu_i, \Sigma_i) \quad (3.19)$$

where  $x$  is a D-dimensional continuous-valued data vector,  $w_i$ ,  $i=1, \dots, N$ , are the mixture weights and  $g(x|\mu_i, \Sigma_i)$ , are the component Gaussian densities. Each component density is a D-variate Gaussian function of the vector form:

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)\right) \quad (3.20)$$

with mean vector  $\mu_i$  and covariance matrix  $\Sigma_i$ . Depending on the assumptions of the covariance matrix, GMM can have several variants. The  $\Sigma_i$  can be a full or diagonal matrix or even having the same covariance matrix for all components. In this case, the covariance matrix is full.

Once the training vector and GMM configuration are decided, the parameters has to be estimated. There are different techniques for estimating the parameters of a GMM [McLachlan and Basford, 1988]. From all this techniques the most popular and well-established method is maximize the likelihood of the GMM. For a sequence of T training vectors  $X = \{x_1, x_2, \dots, x_T\}$ , the GMM likelihood, assuming independence between the vectors, can be written as,

$$p(X|w_i, \mu_i, \Sigma_i) = \prod_{t=1}^T p(x_t|w_i, \mu_i, \Sigma_i) \quad (3.21)$$

To estimate the maximum likelihood parameters is used the iteratively algorithm Expectation-Maximization (EM) [Dempster et al., 1977], it begins with an initial model with parameters  $(w_i, \mu_i, \Sigma_i)$ , to estimate a new model with parameters  $(\hat{w}_i, \hat{\mu}_i, \hat{\Sigma}_i)$ , such that  $p(X|\hat{w}_i, \hat{\mu}_i, \hat{\Sigma}_i) \geq p(X|w_i, \mu_i, \Sigma_i)$ . The new model becomes the initial model for the next iteration until a stopping criteria is reached.

### 3.4.2.2 Input parameter classification

Once the GMM algorithm is trained and has classified the regions, it labels the data points by the number of clusters defined. The number of clusters is equal to the number of discontinuities plus one.

After the data is labeled, this new vector is added as a feature or input parameter increasing the input space domain. Finally, the neural network is trained using these new features (see algorithm 7) by adding this new feature of labeled data points as input parameter constrains the DNN to not predict in between the jump discontinuity.

---

#### Algorithm 7 Automatic Discontinuity Modeling using DNN Algorithm

---

- 1: Train Gaussian Mixture Model
  - 2: Create new vector  $\bar{x}$  with classification label
  - 3: Concatenate  $\bar{x}$  with  $x$
  - 4: Train Deep Neural Network using new  $x$
-

### 3.4.2.3 Benchmark example: comparison study

Let's consider the function:

$$f(x) = a \cdot \text{sign}(x) \quad (3.22)$$

where,  $a$  is a constant of value 10 and  $\text{sign}(x)$  returns the sign of the value  $x$ . This function is in  $\mathbb{R}$  and has a discontinuity jump at  $x = 0$  (see figure 3.10).

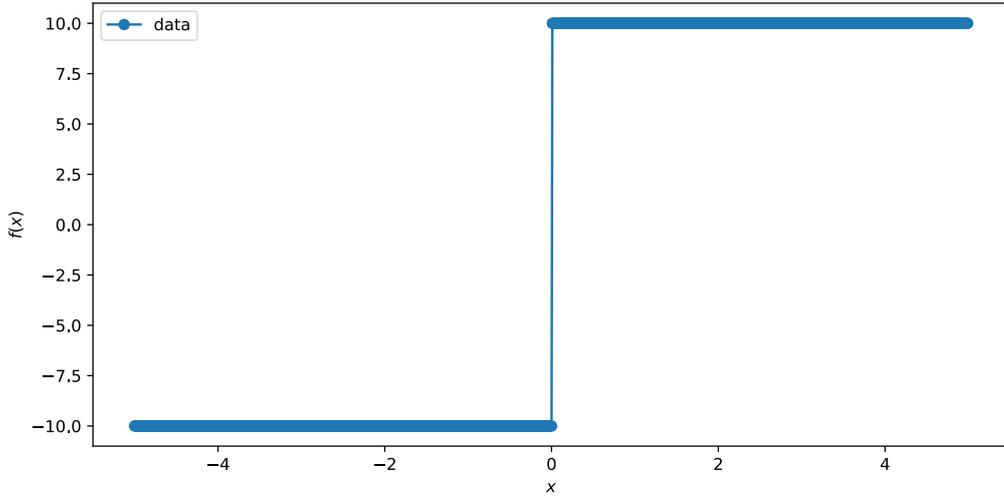


Figure 3.10: Benchmark function:  $10\text{sign}(x)$

First, is used GMM algorithm to cluster the data points (see figure 3.11). The GMM has found two cluster with parameters :

$$\mu_1 = (-2.75, -10), \quad \Sigma_1 = \begin{bmatrix} 2.0625 & 0 \\ 0 & 1e-06 \end{bmatrix}$$

$$\mu_2 = (2.75, 10), \quad \Sigma_2 = \begin{bmatrix} 2.0625 & 0 \\ 0 & 1e-06 \end{bmatrix}$$

The mean value of each cluster corresponds to the center of the data. The covariance matrix tells us that in  $x$  direction is found the data dispersion, for all other direction is almost zero, meaning that the GMM is an ellipsis wider in  $x$  direction than  $y$  direction with zero correlation between  $x$  and  $y$ , corresponding with the function model.

Once the model is classified into two cluster, each data point in the vector  $x$  is classified. A new vector  $\bar{x}$  is created by labeling the vector  $x$  as 0 or 1, depending on the probability of being inside each cluster. Then, both vectors  $x$  and  $\bar{x}$  are concatenated creating a  $\mathbb{R}^2$  input space. the first component has the coordinate and the second component the classification label, as shown bellow:

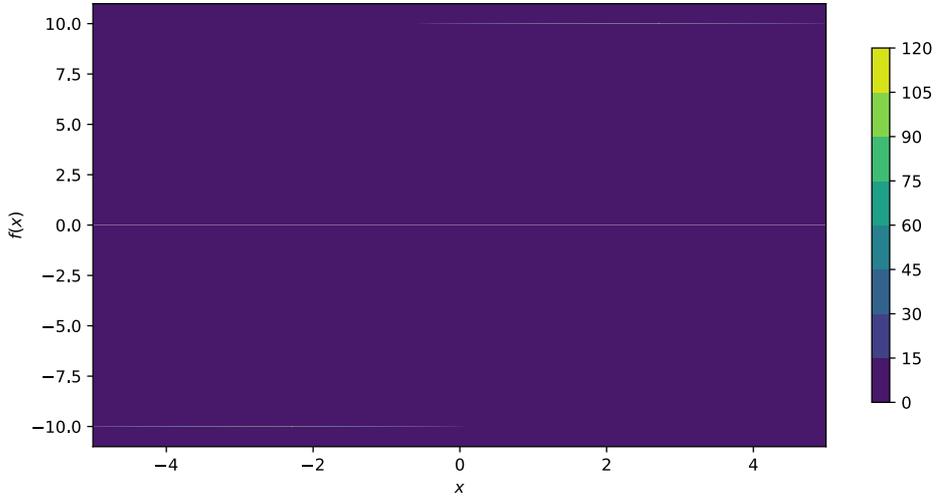


Figure 3.11: Gaussian mixture cluster classification.

$$\hat{x} = ((-5.0, 0), (-4.9, 0), \dots, (4.9, 1), (5.0, 1))$$

The neural network is trained using this new input space with a uniformly distributed sample space between  $(-5,5)$  with a step of 0.1. The resulting prediction is then compared with some state of art machine learning methods. The prediction results are compared using the MSE and  $r^2$  metrics as criteria to determine each method's performance.

$$r^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \quad (3.23)$$

Where  $y_i$  is the known value,  $\hat{y}_i$  is the predicted value, and  $\bar{y}$  is the mean value of the known data.

Table 3.4: Regression comparison of a benchmark function. MSE represents the mean squared error and  $r^2$  represents the coefficient of determination.

Method	MSE	$r^2$
Linear Regression	25.002160	0.749977
Ridge Regression	25.002336	0.749976
LASSO	25.124482	0.748754
Polynomial Regression	9.796450	0.902035
SVR	11.268276	0.887317
GP(RBF)	1.313957	0.986860
DNN	1.139421	0.988606
ADM-DNN	<b>0.000165</b>	<b>0.999998</b>

As shown in figure 3.12, the ADM-DNN method is the most accurate method among

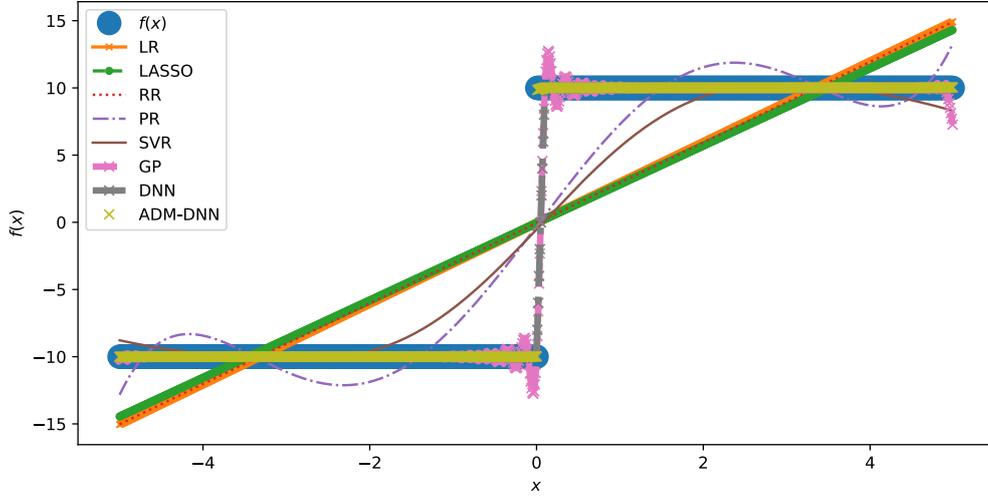


Figure 3.12: Comparison of regression methods predicting a  $f(x) = 10\text{sign}(x)$

all others. All other methods predict a continuous response along with the entire domain, some of them presenting a good performance as DNN and Gaussian process with a Radial Basis function as the kernel. However, they predict values inside the jump discontinuity. This behavior is the contrary case of ADM-DNN; inside the jump discontinuity, no values are predicted, decreasing the MSE to  $1.65e-4$  from a 1.14 (see table 3.4) of his counterpart DNN method using the original inputs parameters.

#### 3.4.2.4 Application of ADM-DNN to wind turbine blade response

This proposed method's mechanical application case is on the maximum aero-elastic loads, resulting in a steady simulation. The fast tool generates the wind turbine blade response by simulating a grid of wind speed between  $[4, 25]m/s$  with a step of  $0.02m/s$  giving a total of 1050 simulations. The network is composed of two hidden layers, each of them having 300 neurons. Figure 3.13 shows a comparison between the state of the art of machine learning regression techniques. As can be seen, ADM-DNN has a better performance compared to all other methods, dealing with the jump discontinuity (All other methods presents a continuous response). A reduction on the mean squared error about 1.53 and having a  $r^2$  closest to one (see table 3.5) shows the applicability of this methodology to mechanical problems outperforming all others regression approaches.

The application problem used only has one discontinuity jump, meaning there are only 2 continuous zones. For problems with more discontinuities the proposed methodology will work because it only depends on the classification of the discontinuities using the Gaussian mixture model. To ensure a good prediction, it is recommended to verify that the clusters are representative of the data and have located the discontinuity point. Also this methodology

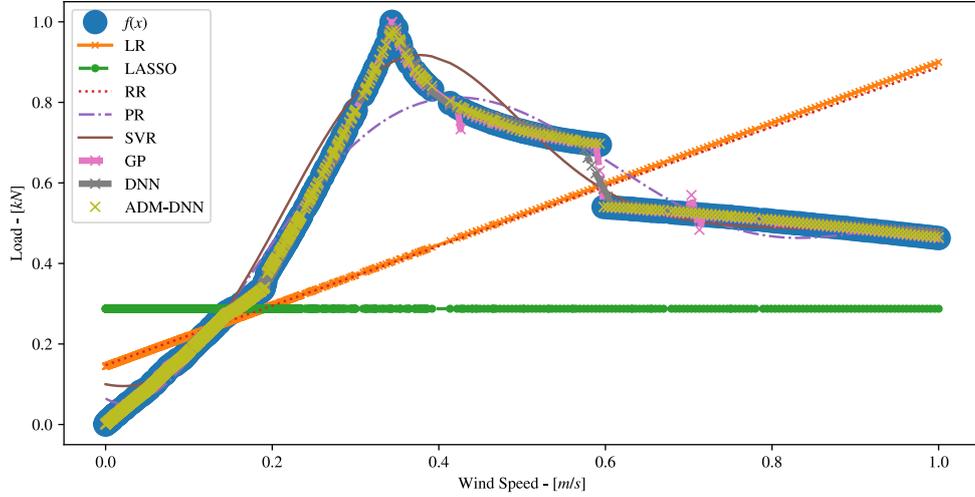


Figure 3.13: Comparison of regression methods predicting an aero-elastic load  $Spn1FLxb1$  on a wind turbine blade.

Table 3.5: Comparison of state of arts regression methods to predict discontinuity in loads generated using aero-elastic simulation tools.

Method	MSE	r2
Linear Regression	2,7511E-02	4,8917E-01
Ridge Regression	2,7552E-02	4,8841E-01
LASSO	5,3856E-02	-5,4704E-06
Polynomial Regression	1,8292E-03	9,6604E-01
SVR	2,3669E-03	9,5605E-01
GP(RBF)	1,0934E-05	9,9980E-01
DNN	3,6024E-05	9,9933E-01
ADM-DNN	<b>7,1429E-06</b>	<b>9,9987E-01</b>

has been tested with low noise signals, in the case of high noise signals with discontinuities there is needed to test more clustering methods to classify the continuous zones.

### 3.5 Conclusions

The proposed approach to model jump discontinuity shows a better performance compared to other regression machine learning models. The case of maximum aero-elastic loads under steady environmental conditions the proposed approach predicts more accurately the output loads by not predicting values in between discontinuity contrary to state of art regression machine learning models. Moreover, only one surrogate model needs to be trained to capture the complete behavior including discontinuities. Showing the applicability of this approach to

mechanical problems.

The ADM-DNN is the method selected to replace a model when a prediction problem has jump discontinuities in their response as presented in the next chapter used in a global sensitivity analysis. Also, all neural network hyperparameters are optimized using the Bayesian optimization method.

# Sensitivity of LAMs for wind turbine using surrogate models

## Sommaire

<b>4.1</b>	<b>Introduction to sensitivity analysis</b>	<b>72</b>
4.1.1	Screening analysis	73
4.1.2	Global sensitivity analysis	77
<b>4.2</b>	<b>Sensitivity analysis in wind turbine</b>	<b>81</b>
<b>4.3</b>	<b>Sensitivity analysis of high dimensional models using surrogate models</b>	<b>83</b>
4.3.1	Inputs parameters for surrogate models	83
4.3.2	Surrogate model 1: aero-elastic loads prediction	89
4.3.3	LAMs for blade 2D shell model	91
4.3.4	Output parameter for global sensitivity	93
4.3.5	Surrogate model 2: stress prediction at blade root	94
4.3.6	Sobol sensitivity analysis using DNN stress	95
4.3.7	Comparison of LAM's sensitivity	100
4.3.8	Morris Analysis: LAM 2D shell FEM	100
<b>4.4</b>	<b>Conclusions</b>	<b>103</b>

Load Application Methods (LAM) are used to transfer the loads obtained from an aero-elastic simulation to a blade shell FEM, replicating the same behavior in both simulations. However, to estimate the wind turbine blade's fatigue life or reliability, authors describe the LAM that distributes the loads as pressure distribution has a more realistic behavior in terms of ultimate damage distribution or fatigue damage along the blade. However, there is no criterion based on the LAM's sensitivity that considers the sensitivity introduced to the LAM's original problem. Also, as the aero-elastic simulation gives a time-history response, LAMs are used for each response time step to generate a time history stress distribution on the 2D shell FEM. This transfer from the 1D model to the 2D FEM increases the total computational calculation time, resulting in a high time-consuming process to implement. This chapter presents the global sensitivity analysis of using load application methods to estimate the stress distribution in a shell FEA of a composite wind turbine blade using deep neural networks. Decreasing the computational time using deep learning surrogate model to replace the 2D shell FEM to predict the stress. In Section 4.1 is presented different methodologies to calculate the

sensitivity as screening and Sobol index analysis. A state of the art of sensitivity analysis in a wind turbine is presented in Section 4.2. Section 4.3 is explained as a new methodology to calculate the global sensitivity of high dimensional problems, applied in the sensitivity of load application methods. Based on deep neural networks to replace the aero-elastic simulations and the wind turbine shell FEM, creating a two-stage neural network to estimate the Von Mises stress and calculate the Sobol indexes for different load application methods. The proposed methodology calculates the global sensitivity using the Sobol method and is compared against a Morris analysis using the blade 2D FEM, presenting good agreement between both analyses.

## 4.1 Introduction to sensitivity analysis

Sensitivity analysis is the study of how uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input [Saltelli, 2002]. This analysis has numerous purposes: measuring the robustness of a model under uncertainties, understanding of the interaction of input and output variables in a model, input variables reduction by identifying the input variables that cause more uncertainty in the output.

The sensitivity analysis can be categorized into three main groups: local, screening, or global analysis. First, local sensitivity analysis is used only when the model analysis does not require a crucial computational time to execute. This type of analysis gives information on the model's behavior only in a reference point neighborhood. It bases on the estimation of partial derivatives of each input parameter. The derivative-based approach has the attraction of being very efficient in computer time. However, it is inefficient in terms of the analyst's time. One has to intervene in the computer program, insert ad hot coding, and efficiently perform it.

Nevertheless, a derivative-based approach's fatal limitation is that it is unwarranted when the model input is uncertain and has unknown linearity. The derivatives only give information at the point analyzed and did not provide an exploration of the rest of the input domain space [Saltelli et al., 2008]. Second, screening sensitivity analysis is used mostly to identify the input parameters with less influence in the model output. The screening analysis provides qualitative measures of sensitivity. In other words, it arranges the input parameters based on the measure of their importance without quantification. Screening analysis explores the input space in more points than the derivatives approach, staying numerically efficient. However, there is no exploration in the entire domain space. This method can handle problems with a high number of input parameters (from dozen to 100). Also, they are used before applying global sensitivity analysis to reduce the number of inputs considered [Saltelli et al., 2008]. Finally, global sensitivity analysis studies the effect of the input parameters in the model considering the uncertainties all over its domain, and not only at a point neighborhood. The idea of global analysis is to quantify the contribution of the relative variability of each input parameter over the variability of the model output. In other words, it estimates the contribution of each input parameter over the mathematical variance over the response of the model.

For this reason, they are often called Analysis of Variance (ANOVA). Global analysis is computationally expensive compared to all other methods because they use Monte Carlo simulation to explore the input space requiring thousand or millions of simulations. As this research aims to explore the input space of our models, the two last methods are explained in more detail.

#### 4.1.1 Screening analysis

One of the most popular methods is Morris method [Morris, 1991] for sensitivity analysis (i.e., screening analysis) or one-step-at-a-time method (OAT), is used to quantify the variance effect of each input parameter in the output parameter. Also, screening methods are often used as a ranking method for input parameters by their importance order. This characteristic is intriguing in cases where the model analyzed has an actual number of inputs parameters. The application of this study can be considered a pre-sensitivity analysis. Identifying the input parameter with less influence and reducing the number of parameters considered in the next sensitivity analysis. In this context, one of the methods most developed is the Morris method. The main idea is to determine among all input parameters which affect: (a) negligible, (b) linear and additive, or (c) nonlinear or interaction with other input parameters. This ranking or classification is based on the notation of an elementary effect (explained below). This classification is not performed directly from the elementary effect but from its statistics (i.e., mean and standard deviation) calculated from many samples. In fact, for a given input parameter, a high value of it mean elementary effect shows the importance of this parameter over the model output. In contrast, a high value of the elementary effect's standard deviation shows strong interactions between input parameters (see figure 4.1).

The method is very simple, consider a model with  $k$  independent input factors  $X_i, i = 1, 2, \dots, k$ , which varies across  $p$  levels. The input space is discretized in  $p$ -level grid  $\Omega$ . To quantify this effect, Morris has defined an elementary effect ( $EE$ ) associated with the input parameter  $X_i$  defined as:

$$EE_i = \frac{f(x_1, \dots, x_{i-1}, x_i + \Delta, x_{i+1}, \dots, x_n) - f(x)}{\Delta} \quad (4.1)$$

where,  $p$  is the number of levels,  $\Delta$  is a value in  $1/(p-1), \dots, 1 - 1/(p-1)$ ,  $X = (x_1, x_2, \dots, x_k)$  in any selected value in  $\Omega$  such that the transformed points  $(X + e_i \Delta)$  is still in  $\Omega$  for each index  $i = 1, 2, \dots, k$  and  $e_i$  is a vector of zeros but with a unit as its  $i$ th component.

The distribution of elementary effects associated with the  $i$ th input parameter is obtained by randomly sampling different  $X$  from  $\Omega$  and is denoted by  $F_i$  (i.e.  $EE_i F_i$ ). The  $F_i$  distribution is finite and, if  $p$  is even and  $\Delta$  is chosen to be equal to  $p/(2(p-1))$ , the number of elements of  $F_i$  is  $p^{k-1}[p - \Delta(p-1)]$ .

Realizing many random selections of point  $X_i$ , is obtained for each input parameter a finite distribution of  $EE_i$ . To characterize, in a statistical context, the distribution of  $EE_i$  is

calculated his mean and standard deviation values:

$$\mu_i = \frac{1}{r} \sum_j^r EE_i^j \quad (4.2)$$

$$\sigma_i = \sqrt{\frac{1}{r} \sum_j^r (EE_i^j - \mu_i)^2} \quad (4.3)$$

Where  $r$  is the number of elements in the distribution of  $F_i$ .

The mean value  $\mu$  estimates the global influence of the input parameters on the output. The standard deviation  $\sigma$  estimates the ensemble of the input's effects, whether there are nonlinear or interactions with other parameters. In other words, a high value of  $\mu_i$  indicates that the input parameters have a significant overall influence on the output parameter. On the other hand, a high value of the standard deviation of  $\sigma_i$  indicates that the input parameter  $X_i$  has interactions with other parameters  $X_j$ . Campolongo et al. [Campolongo et al., 2007] proposed replacing the use of  $\mu$  with  $\mu^*$ , which defines the estimation of the absolute mean values of the elementary effects denoted with  $G_i$  (i.e.,  $|EE_i| G_i$ ).

$$\mu_i^* = \frac{1}{r} \sum_j^r |EE_i^j| \quad (4.4)$$

The use of  $\mu^*$  solves the problem of type II errors (failing to identify a parameter with considerable influence on the model), to which the original measure  $\mu$  is vulnerable. This type of error occurs when the distribution  $F_i$  contains both positive and negative elements. In these cases, some effects may cancel each other out when computing  $\mu$ , thus producing a low mean value even for an important parameter. To avoid type II errors, Morris [Morris, 1991] recommended considering the values of  $\mu$  and  $\sigma$  simultaneously, since a parameter with elementary effects of different signs would have a low value of  $\mu$  but a considerable value of  $\sigma$ . A graphical representation in the  $(\mu, \sigma)$  plane allows a better interpretation of results by taking into account at the same time, two sensitivity measures (see figure 4.1).

The method described has a total number of calls equal to  $N_s = 2rk$ , each elementary effect  $EE_i$  needs to evaluate the model two times. To measure the efficiency of the experimental procedure used, Morris [Morris, 1991] introduces a measure called economy, equal to the number of elementary effects it produces divided by the number of the model experimental runs. The economy of the described experiment before is  $rk/2rk = 1/2$ .

To reduce the cost of the calculation (i.e., the number of calls to the model), Morris [Morris, 1991] has proposed a more economical procedure that builds  $r$  trajectories of  $(k + 1)$  points in the input space, each providing  $k$  elementary effects, one per input parameter, for a total of  $r(k + 1)$  sample points. The procedure is based on constructing a matrix denoted  $\mathbf{B}^*$  of  $k + 1$  rows and  $k$  columns, where each row represents a realization of vector  $x$ . The matrix

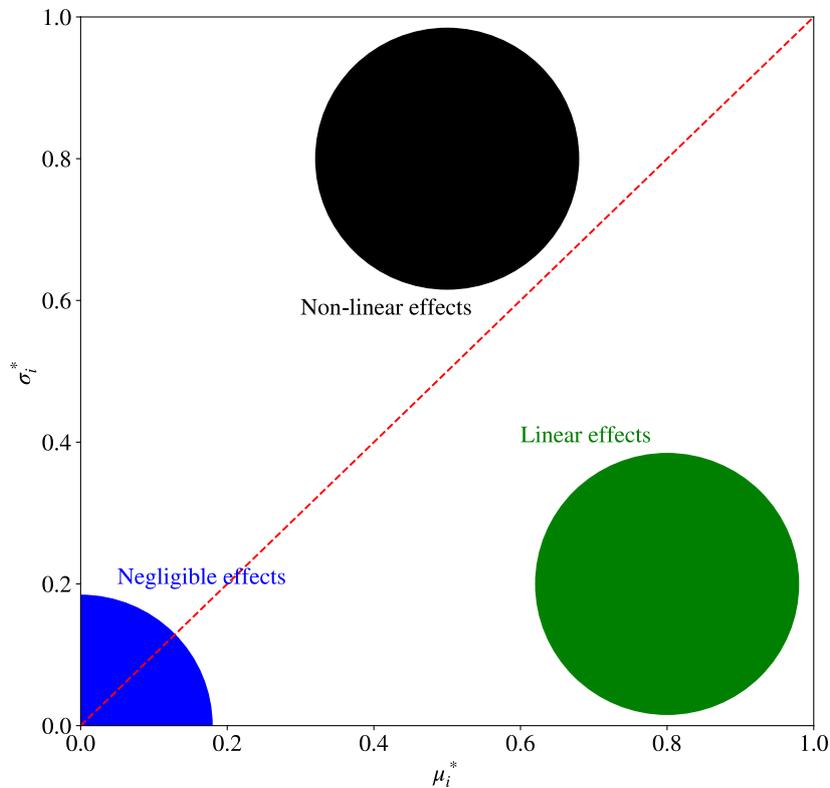


Figure 4.1: Morris method classification of input parameters.

$\mathbf{B}^*$ , has the following property: for every column  $j = 1, 2, \dots, k$ , only two rows of matrix  $\mathbf{B}^*$  have different components. Due to this property, the  $k + 1$  rows of matrix  $\mathbf{B}^*$ , represents  $k + 1$  different points of vector  $x$ , resulting  $k + 1$  values of the model response allowing to calculate  $k$  elementary effect (i.e. one elementary effect for each input parameter). This new procedure's economy is  $k/k + 1$ , which is better than the original procedure. To create the matrix  $\mathbf{B}^*$ , Morris [Morris, 1991] has proposed that is more suitable to select an even  $p$ -level and a  $\Delta = p/2(p - 1)$ , ensuring that each  $p^{k-1}[p - \Delta(p - 1)] = p^k/2$  associated elementary effect to an input parameter has the same probability to be selected.

A scheme to generate trajectories with the required properties explained before is as follows. To build  $\mathbf{B}^*$ , the first step is the selection of a matrix  $\mathbf{B}$ , whose dimensions are  $(k + 1) \times k$ , with elements that are 0's and 1's and the key property that every column index  $j$ ,  $j = 1, \dots, k$ , there are two rows of  $\mathbf{B}$  that differ only in the  $j$ th entry. In this case  $\mathbf{B}$  is strictly lower triangular matrix of 1's:

$$B = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}.$$

Then, the matrix  $\mathbf{B}'$ , given by

$$\mathbf{B}' = \mathbf{J}_{k+1,k} \mathbf{x}^* + \Delta \mathbf{B},$$

Where  $\mathbf{J}_{k+1,k}$  is a  $(k+1) \times k$  matrix of 1's and  $\mathbf{x}^*$  is a randomly chosen base value of  $X$ , is a potential candidate for the desired design matrix. However, it has the limitation that the  $k$ th elementary effect it produces would not be randomly selected.

A randomized version of the sampling matrix is given by

$$\mathbf{B}^* = (\mathbf{J}_{k+1,1} \mathbf{x}^* + (\Delta/2)[(2\mathbf{B} - \mathbf{J}_{k+1,k})\mathbf{D}^* + \mathbf{J}_{k+1,k}])\mathbf{P}^*, \quad (4.5)$$

Where  $\mathbf{D}^*$  is a  $k$ -dimensional diagonal matrix in which each element is either +1 or -1 with equal probability, and  $\mathbf{P}^*$  is a  $k$ -by- $k$  random permutation matrix in which each row contains one element equal to 1, all others are 0, and no two columns have 1's in the same position. Read row by row,  $\mathbf{P}^*$  gives the order in which factors are moved;  $\mathbf{D}^*$  states whether the factors will increase or decrease their value along the trajectory.  $\mathbf{B}^*$  provides one elementary effect per input, which is randomly selected. The following algorithm summarized the sampling procedure proposed by Morris to rank the inputs parameters of a model.

---

**Algorithm 8** Morris sensitivity analysis algorithm

---

- 1: Initialize:  $p, k, \Delta$
  - 2:  $i \leftarrow 1$
  - 3: **while**  $i \leq k$  **do**
  - 4:   Random selection of point  $x^{i,*}$   $\triangleright x_k \in \{0, 1/(p-1), \dots, 1-\Delta\}$
  - 5:   Build matrix  $\mathbf{B}^{1,*}$
  - 6:   Model evaluation at points  $\mathbf{B}^{1,*}$
  - 7:   Calculate the elementary effect  $EE_i$
  - 8:    $i \leftarrow i + 1$
  - 9: Calculate elementary effect statistics  $\mu$  and  $\sigma$
- 

In general, Morris's approach is valuable, but it may become problematic in large models with multiple outputs [Campolongo et al., 2007]. On the other hand,  $\mu^*$  is a practical and concise measure to use, especially when there are several output variables. Moreover, in contrast,  $\mu$  and  $\mu^*$  has the advantage that it can be adjusted to work with a group of parameters, i.e., to produce an overall sensitivity measure relative to a group [Saltelli et al., 2008]. Campolongo [Campolongo et al., 2007] have also shown that  $\mu^*$  is a good proxy of the total sensitivity index  $S_T$  discussed in next (4.1.2).

In summary, The number of calls of this method is efficient compared to more demanding methods for global sensitivity analysis because the number of simulation required is linear to the number the model input parameters. Since the screening analysis goal is to rank the influence of the input parameters in a sub-set of the whole input domain space, this method only qualifies each input parameter's sensitivity in three states: linear effect, nonlinear effect, or neglected effect. In other words, it is not clear what contribution has one variable to the output parameter, or how the input parameters interact between them, how many of them interact. For these reasons, a more demanding sensitivity analysis must be used to analyze the input parameters in their entire domain, giving a more accurate description of each parameter's influence and interactions.

## 4.1.2 Global sensitivity analysis

### 4.1.2.1 Variance decomposition

The sensitivity analysis based on variance study aims to determine the contribution of each input random variable  $X_i$  in the variance  $V[Y]$  of the response  $Y$ . Based on this idea, Saltelli [Saltelli et al., 2010] have shown that this variance can be decomposed in the way:

$$V[Y] = \sum V_i + \sum V_{ij} + \sum V_{ijk} + \dots + \sum V_{ijk\dots N} \quad (4.6)$$

where  $V_i, V_{ij}, V_{ijk}, \dots, V_{(ijk\dots N)}$  are called partial variance:

$$V_i = [E[Y|X_i = x_{i*}]] \quad (4.7)$$

$$V_{ij} = V[E[Y|X_i = x_{i*}, X_j = x_{j*}]] \quad (4.8)$$

$$V_{ijk} = V[E[Y|X_i = x_{i*}, X_j = x_{j*}, X_k = x_{k*}]] \quad (4.9)$$

with  $E[Y|X_i = x_{i*}]$  the expected value of the random variable  $Y$  given the parameter  $X_i$  is deterministic (i.e fixed to a particular value  $x_{i*}$ ). It is often called conditional expectation of first order. Having frozen one potential source of variation ( $X_i$ ), the resulting variance  $V_i$  will be less than the corresponding total or unconditional variance  $V[Y]$ . Therefore,  $V_i$  could be used to measure the relative importance of  $X_i$ , reasoning that the smaller  $V_i$ , the greater the influence of  $X_i$ . Likewise,  $E[Y|X_i = x_{i*}, X_j = x_{j*}, X_k = x_{k*}]$  is the conditional expectation of  $k$ th order, evaluated considering  $k$  input parameters as deterministic.

The partial variance  $V_i$  measures the contribution of the randomness associated with the input parameter  $X_i$  in the response total variance  $Y$ , whereas the superior order variances measure the interaction contribution between the different input parameters.

## 4.1.2.2 Sensitivity Sobol indexes

To provide a straightforward Monte Carlo-based implementation of the concept, capable of computing sensitivity measures for arbitrary groups of parameters. Given a square integrable function  $f$  over  $\Omega^N$ , the  $N$ -dimensional unit hypercube,

$$\Omega^N = (X|0 \leq x_i \leq 1; i = 1, \dots, N) \quad (4.10)$$

Sobol [Sobol, 1993] has proposed the decomposition of the function  $f$  that represents the model as a sum of functions of increasing dimensions. In other words, It is composed of a constant term  $f_0$ , uni-dimensional  $f(x_i)$ , bi-dimensional  $f(x_i, x_j)$  and so on. This representation is written as:

$$f(\mathbf{x}) = f_0 + \sum_{i=1}^N f_i(x_i) + \sum_{i=1}^N \sum_{j=1}^N f_i(x_i, x_j) + \dots + f_{1,2,3,\dots,N}(x_1, x_2, x_3, \dots, x_N) \quad (4.11)$$

Sobol [Sobol, 1993] proved, for uniform variables  $[0, 1]$ , this representation exist, if and only if the term  $f_0$  is constant and the integral of the high order terms  $f_{1,2,\dots,s}$  with respect to one parameter  $x_{i_k}$  is zero:

$$\int_0^1 f_{1,2,\dots,s}(\mathbf{x}) dx_{i_k} = 0, \text{ for } 1 \leq k \leq s \quad (4.12)$$

Therefore, all the terms  $f_{1,2,\dots,s}$  from decomposition defined in equation 4.11 are orthogonal. If  $(i_1, \dots, i_s) \neq (j_1, \dots, j_l)$ , then:

$$\int_{[0,1]^N} f_{i_1,\dots,i_s}(\mathbf{x}) f_{j_1,\dots,j_l}(\mathbf{x}) d(\mathbf{x}) = 0 \quad (4.13)$$

These terms can be calculated by the evaluation of the multidimensional integrals, as:

$$f_0 = \int_{[0,1]^N} f(\mathbf{x}) d(\mathbf{x}) \quad (4.14)$$

$$f_1 = \int_{[0,1]^{N-1}} f(\mathbf{x}) d(\mathbf{x}_{-i}) - f_0 \quad (4.15)$$

$$f_2 = \int_{[0,1]^{N-2}} f(\mathbf{x}) d(\mathbf{x}_{-ij}) - f_i(x_i) - f_j(x_j) - f_0 \quad (4.16)$$

where,  $d(\mathbf{x}_{-i})$  and  $d(\mathbf{x}_{-ij})$  indicate that the integral is done with respect to all the variables except with respect to  $x_i$  or with respect to  $x_i$  and  $x_j$  respectively. The high-order terms can be calculated using the same procedure.

The equations 4.14 to 4.16 correspond respectively to conditional expectation of order 0, 1 and 2. They can be written as:

$$E[Y] = f_0 \quad (4.17)$$

$$E[Y|X_i = x_i^*] = f_i(x_i) - f_0 \quad (4.18)$$

$$E[Y|X_i = x_i^*, X_j = x_j^*] = f_{ij}(x_i, x_j) - f_0 \quad (4.19)$$

The same logic can be used for higher orders.

By definition, the variance of a random variable is:

$$V[Y] = \int_{[0,1]^N} f^2(\mathbf{x})d(\mathbf{x}) - f_0^2 \quad (4.20)$$

Replacing  $f$  in equation 4.20 and developing as equation 4.11, is obtained a decomposition of the variance  $V[Y]$  as defined in equation 4.6. By identification, partial variances  $V_i$ ,  $V_{ij}$  are defined respectively as:

$$V_i = V[f_i] = \int_{[0,1]} f_i^2(x_i)dx_i \quad (4.21)$$

$$V_{ij} = V[f_{ij}] = \int_{[0,1]^2} f_{ij}^2(x_i, x_j)dx_i dx_j \quad (4.22)$$

Generally, the partial variance  $V_{i_1, \dots, i_s}$  correspond to the term  $f_{i_1, \dots, i_s}$  such as  $1 \leq i_1 < \dots < i_s \leq N$  and  $s = 1, \dots, N$ , is obtained by:

$$V_{i_1, \dots, i_s} = V[f_{i_1, \dots, i_s}] = \int_{[0,1]^s} f_{i_1, \dots, i_s}^2(x_{i_1}, \dots, x_{i_s})dx_{i_1} \dots dx_{i_s} \quad (4.23)$$

Normalizing this partial variances with respect to the total variance  $V[Y]$ , Sobol [Sobol, 1993] has introduced a set of measures of sensitivity for different orders, called Sobol sensitivity indexes. The first order Sobol index is defined by:

$$S_i = \frac{V_i}{V[Y]} \quad (4.24)$$

Furthermore, it represents the main effect contribution of the input parameter  $X_i$  in the variable variance of  $Y$ . This measure is also called importance measure or correlation ratio.

The Sobol second order index is defined by:

$$S_{ij} = \frac{V_{ij}}{V[Y]} \quad (4.25)$$

Measuring the contribution of the interaction between input parameters  $X_i$  and  $X_j$  with the variance of the output variable  $Y$ . In other words, the sensitivity  $Y$  that is not taken into account in the effect of parameters  $X_i$ ,  $X_j$  and  $X_k$  alone, neither in interactions two by two.

The Sobol index for higher orders are obtained in the same way. In general, the Sobol indices  $S_{i_1, \dots, i_s}$  of order  $s$ , are defined by:

$$S_{i_1, \dots, i_s} = \frac{V_{i_1, \dots, i_s}}{V[Y]} \quad (4.26)$$

As all Sobol indexes are positive and their sum is equal to one:

$$\sum_i S_i + \sum_{i < j} S_{ij} + \sum_{i < j < k} S_{ijk} + \dots + S_{1, 2, \dots, N} = 1 \quad (4.27)$$

The Sobol indices are easy to interpret, indexes with values closed to one; the variability of an input parameter has a significant effect over the variability of the output parameter.

For a model with  $N$  input parameters, the number of evaluations for the Sobol index of a different order is equal to  $2^N - 1$ . If  $N$  is enormous, the number of Sobol index increases exponentially. In consequence, the computational time due to model evaluation is not possible. In the case of models where the interaction between parameters does not affect the output, this problem does not arise since the indices of first-order Sobol alone can quantify each parameter's effect on the model response. On the other hand, when the interactions between the parameters have an influence on the response of the model, this problem arises seriously.

To solve this particular problem, Homma et Saltelli [Homma and Saltelli, 1996] have proposed a new sensitivity measure, called the total sensitivity index  $S_T$ . It is the sum of the principal relative effect of the input variable  $X_i$  and the interaction effects in which this parameter appears. In other words, for the input parameter  $X_i$ , the total sensitivity index corresponds to the sum of all indexes that interfere with the index  $i$ .

$$S_{T_i} = S_i + \sum S_{ij} + \dots + S_{ijk\dots n} = 1 - \frac{V[E[Y|\mathbf{X}_{-i}]]}{V[Y]} \quad (4.28)$$

where  $\mathbf{X}_{-i}$  is a vector that represents all the parameters except the parameter  $X_i$ . Referring to the law of total variance, the variance of the output variable  $Y$  of the model is:

$$V[Y] = V[E[Y|X_{-i}]] + E[V[Y|X_{-i}]] \quad (4.29)$$

where  $V[E[Y|X_{-i}]]$  and  $E[V[Y|X_{-i}]]$  are the "unexplained" and the "explained" components of the variance respectively.

Replacing  $V[E[Y|X_{-i}]]$  in equation 4.28, the total index  $S_T$  can be rewritten as:

$$S_{T_i} = \frac{E[V[Y|\mathbf{X}_{-i}]]}{V[Y]} \quad (4.30)$$

The measure  $E[V[Y|\mathbf{X}_{-i}]]$  is the expected value of the conditional variance  $V[E[Y|X_{-i}]]$  and can be estimated by Monte-Carlo simulations. The evaluation of total index  $S_T$  is more economical in terms of model runs concerning the classical procedure, which calculates the different high order Sobol indices. The number of estimations is equal to  $M(N + 1)$ , where  $M$  is the number of simulations used to calculate the first-order Sobol index. Calculating the first order and total order indices, the total cost is equal to  $M(N + 2)$ .

Campolongo [Campolongo et al., 2007] has found a relationship between Morris and Sobol analysis, showing that  $\mu^*$  is a good proxy of the total sensitivity index  $S_T$  discussed before.  $S_{T_i}$  is used when the goal is that of identifying non-influential parameters in a model (rather than prioritizing the most influential ones).  $\mu^*$  is an adequate substitute for the total index when the computational cost of  $S_T$  is unaffordable [Saltelli et al., 2008].

## 4.2 Sensitivity analysis in wind turbine

Wind turbine sensitivity analysis is performed by using different methodologies depending on the complexity of the design problem. Global sensitivity analysis based on Monte Carlo simulation (MCS) for the wind turbine is used in low-dimensional problems with a high computational cost model [Ziegler et al., 2015] or for high dimensional problems with a low computational cost model [Tran and Smith, 2018], which means that large samples can be produced with a moderate computational cost. However, for problems with high-dimensional and high computational cost problem modeling, these methods are unaffordable. Also, sensitivity analysis is performed in a grid of values selected by the user as performed by Geng et al. [Geng et al., 2018] for airfoil aerodynamics, contrary to MCS, the model is only evaluated in some points of the entire domain resulting in a cheaper computational cost. However, the sensitivity effect is not entirely captured with this type of methodology.

Another method for sensitivity is Morris's "screening" analysis [Morris, 1991] or One-at-a-Time (OAT) used to qualify the sensitivity effects of the input parameters on the output parameter. This method is evaluated in a subset of the entire domain of the input parameters and only varying one variable per iteration; an example of this method in wind turbine foundation fatigue loads by Velarde et al. [Velarde et al., 2019]. The results obtained are easy to understand, and as the number of the iteration is linear to the number of input parameters is frequently used for high-dimensional problems. However, the Morris method does not estimate the interaction between variables quantitatively; in other words, it is unknown which variables correlate them and their sensitivity effect value quantity. For this reason, the Morris method is used as a pre-step to reduce the number of input variables and subsequently use variance-based sensitivity analysis; Hübler et al. presents an example of this methodology. [Hübler et al., 2017] for global sensitivity analysis of offshore wind turbines based on aero-elastic time-domain simulations starting with more than 50 parameters for screening analysis and reducing it to near 20 parameters for variance-based sensitivity analysis.

Nevertheless, for some problems as the input parameters may have interactions between each other, ignoring some inputs after the reduction, their inner interaction will not be measured while using the variance-based sensitivity analysis. There are different variance-based sensitivity analysis as Sobol method [Sobol, 1993] and Fourier amplitude sensitivity test [Cukier et al., 1973, Cukier et al., 1975, Cukier et al., 1978]. That quantifies each input parameter's sensitivity effect and their interactions by calculating some indexes of different degrees, representing the effect on the output variance produced by one or any combination of input variables. Compared to screening analysis, these methods are evaluated in the entire domain of the inputs parameters, generally using MCS, making their computational cost expensive. Some examples of these methods have been used on the blade geometry variables' sensitivity on the wind turbine performance by Echevarria et al. [Echeverría et al., 2017] using 21 input parameters. Other authors use a surrogate model to replace their original model and estimate the global sensitivity analysis based on this new model, reducing the model evaluation's computational time. McKay et al. [McKay et al., 2014] studied the global sensitivity of wind turbine power output based on experimental data and only considering eight input parameters to train their surrogate model. As this problem has low dimension input parameters and data could be obtained from experimentation, their computational cost is less expensive than problems where the data are generated using aero-elastic simulation tools. The sensitivity analysis has to be performed in high dimensional input parameters (more than 50 variables).

This study presents a new methodology to calculate the global sensitivity of high dimensional problems without reducing the number of inputs parameters and using computational expensive calculation time aero-elastic simulations and finite element analysis for the wind turbine blade model. This methodology and its application in structural mechanics is developed bellow.

### 4.3 Sensitivity analysis of high dimensional models using surrogate models

As explained in chapter 2, Load Application Methods (LAM) are used to transfer the loads obtained from an aero-elastic simulation to a blade shell FEM, replicating the same behavior in both simulations. However, to estimate the wind turbine blade's fatigue life or reliability, authors describe the LAM that distributes the loads as pressure distribution has a more realistic behavior in terms of ultimate damage distribution or fatigue damage along the blade [Caous et al., 2018, Hu et al., 2016b]. However, there is no criterion based on the LAM's sensitivity that considers the sensitivity introduced to the original problem by the LAM itself. Also, as the aero-elastic simulation gives a time-history response, LAMs are used for each response time step to generate a time history stress distribution on the 2D shell FEM. This transfer from 1D model to 2D FEM increases the total computational calculation time, resulting in a high time-consuming process to implement.

To measure the sensitivity of the LAMs, the author proposed a methodology based on surrogate models to decrease the high computational cost of the model evaluation and manage a high input dimension model without reducing the input space. First, an aero-elastic simulation is performed using a simplified model of the wind turbine blade that output the resultant loads acting in different nodes of the blade. Adding the material distribution as input variables and the loads from aero-elastic simulations is created a 2D shell FEM of the blade. This study transfer of loads with three different LAM: rigid body element (RBE), 4 Nodes (4NO), and pressure distribution (PD) method by Caous [Caous et al., 2018] (see chapter 2). The output from this FE analysis is the stress distribution of the wind turbine blade. Finally, a global sensitivity analysis evaluates all input parameters: the aero-elastic loads and the thickness of composite material distribution and output of the blade's stresses. As explained, executing this strategy is unaffordable because of the two high time-consuming models, FAST and blade 2D shell FEM. In this case, both simulations model are replaced with a surrogate model, as shown in Figure 4.2. With the increasing development of machine learning methods in structural engineering [Salehi and Burgueno, 2018], deep neural network (DNN) is the selected surrogate model technique to replace the wind turbine blade shell FEM model, and FAST simulation tool.

#### 4.3.1 Inputs parameters for surrogate models

To create the surrogate model to predict the blade root stress (Surrogate model 2 or High dimensional DNN), their inputs parameter space must be defined. This study considers only two types of uncertainties. Due to the wind conditions acting on the blade and uncertainties in the distribution of composite materials along the blade. This study classifies two groups of input variables: aero-elastic loads and material composite thickness. It is well known by the author that other sources of uncertainty could be introduced in the model as uncertainties by manufacturing defects, the orientation of the fibers, uncertainties on the mechanical properties, uncertainty in the long term loads, uncertainty introduced by the surrogate model

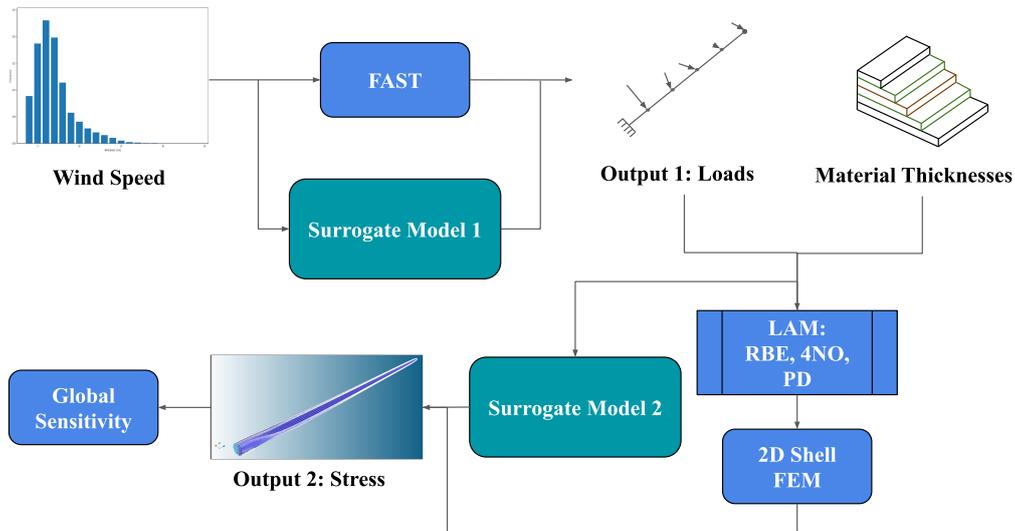


Figure 4.2: Simulation methodology to calculate and compare the Sobol global sensitivity analysis of LAMs: 4 nodes (4NO), rigid body elements (RBE) and pressure distribution Caous method (PD).

itself, and others. Complicating the model but having the closest behavior to reality. However, the authors only consider these two uncertainties variables to evaluate and validate the methodology proposed.

#### 4.3.1.1 Maximum aero-elastic loads: steady simulation

These loads are obtained at ten different locations (gages) in the blade span-wise direction (see figure 4.3) by extracting the maximum value after 10 minutes simulation using FAST code from a uniform and steady wind speeds between  $4$  and  $25m/s$ . The value is extracted after 300 simulated seconds after the beginning of the simulation to ensure that the response is stationary for each wind speed, and the pitch control of the blade has obtained their setpoint value (see figure 4.4). As nine gages are selected (excluding blade root gage in the cases LAM RBE and 4NO), and at each gage, the loads act in 6 DOF (i.e., degree of freedom) of the blade coordinate system. It considers 54 aero-elastic loads (3 forces and three moments) as inputs parameters for surrogate model 2 or DNN Stress.

To identify each load is adopted the following nomenclature. First, is specified the span location of the load along the blade. Second, the load can be the resultant force or a moment acting at a span location. Third, the blade axis direction of the loads;  $y$  axis points towards the trailing edge of the blade and parallel with the chord line;  $z$  axis is points along the pitch axis towards the tip of the blade, and the  $x$  axis is orthogonal with the  $y$  and  $z$  axis such that they form a right-handed coordinate system. Moreover, finally, all of them are extracted for



Figure 4.3: Wind turbine blade cross section in the span-wise direction equally located at the gages from FAST simulation.

one blade  $b1$ . For example,  $Spn2FLyb1$  means, span location 2: force acting in the y-direction of blade 1.

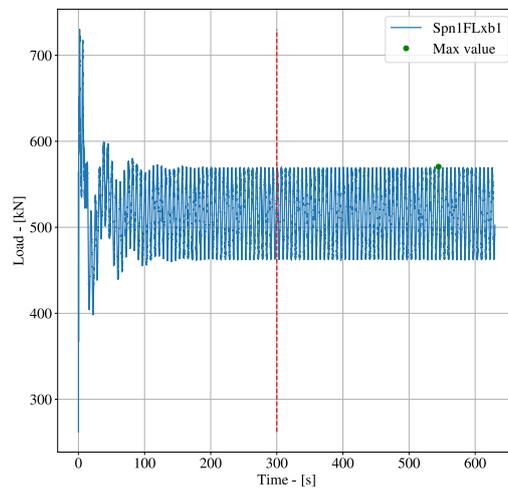


Figure 4.4: Load  $Spn1FLxb1$  (Span location 1, force in lift direction for blade 1) maximum value extraction for a steady simulation at  $11m/s$ .

A FAST simulation must be executed for different wind speed to calculate their maximum value to generate these loads. Since these simulations use steady wind conditions, and only the maximum at the stationary phase. This process can be replaced by a surrogate model

(Surrogate Model 1: Low dimensional or DNN Loads) that will predict 54 maximum loads acting on the blade given the desired wind speed.

#### 4.3.1.2 Composite material thickness distribution

The wind turbine used in this research is the reference 10MW WT from DTU [Bak et al., 2013]. However, the material distribution along the blade was simplified to work with a less complicated model (refer to chapter 2). After making all these changes in the spatial material distribution, the total material thickness input variables are 24 (6 shear webs, nine trailing panels, and nine leading panels).

The nomenclature used to identify each thickness is: first, specify the cross-sections; second, define the composite material; and finally, specify the region in the span-wise direction. For example, *LP\_QQ1\_3* means QQ1 composite material used in the leading panel at region 3 in the blade's span direction.

#### 4.3.1.3 Input parameter domain

For LAM RBE and 4NO, the total number of inputs parameters is 78, with 24 are composite material thickness, and 54 are aero-elastic loads (forces and moments) distributed in the blade span-wise direction. A uniform probability distribution at different ranges is selected for each thickness with a variation of 10% from their mean value (see tables 2.3, 2.4, 2.5) presenting the mean values of each thickness). A uniform distribution is used to give more probability to the variation of thicknesses along the blade, however it is not taken into account uncertainties in the manufacturing defects, fiber orientation or mechanical properties, the composite material are consider homogeneous in the simulations. Concerning the aero-elastic loads, is used the surrogate model 1 or DNN Loads to predict them. Lower and upper boundaries for all loads and moments are presented in table 4.1.

In LAM PD, the input parameters have to be defined differently from the other methods. This method corrects the pressure distribution along the blade in different steps, differentiating the aero-elastic loads and inertial loads. The input parameters are selected as the quantities that correct the pressure distribution. The aerodynamic correction selected as input parameters: the TJ Knill lift coefficient  $C$ , the difference of the resultant forces in X-direction between the beam aero-elastic model and the shell FEM  $DFX$ , and the difference of the resultant forces in Y-direction between the beam aero-elastic model and the shell FEM  $DFY$ . For inertial loads corrections are selected, the acceleration correction in X-direction  $AX$  and the acceleration correction in Y-direction  $AY$  (see chapter 2). Each of these quantities is extracted from 10 gages along the blade span as the other methods, resulting in a total of 50 input parameters for LAM PD. Adding these parameters to the 24 material thickness parameters, there is a total of 74 input variables.

To extract each parameter's boundaries values, the maximum wind speeds after 300 seconds

### 4.3. Sensitivity analysis of high dimensional models using surrogate models 87

Table 4.1: Lower and upper boundary for maximum steady aero-elastic loads for wind speed between  $4m/s$  and  $25m/s$ .

Forces	Lower Boundary	Upper Boundary	Moments	Lower Boundary	Upper Boundary
	[ $kN$ ]	[ $kN$ ]		[ $kNm$ ]	[ $kNm$ ]
Spn1FLxb1	162.1	615.5	Spn1MLxb1	691.9	14650.0
Spn1FLyb1	102.1	426.5	Spn1MLyb1	6929.0	30630.0
Spn1FLzb1	415.4	1747.0	Spn1MLzb1	-216.7	9.306
Spn2FLxb1	222.7	725.4	Spn2MLxb1	-8370.0	3345.0
Spn2FLyb1	175.8	583.4	Spn2MLyb1	5245.0	27070.0
Spn2FLzb1	343.8	1616.0	Spn2MLzb1	-209.0	1.669
Spn3FLxb1	192.4	635.8	Spn3MLxb1	-263.0	4886.0
Spn3FLyb1	39.52	341.2	Spn3MLyb1	4556.0	23060.0
Spn3FLzb1	272.1	1430.0	Spn3MLzb1	-176.7	-6.005
Spn4FLxb1	157.5	593.7	Spn4MLxb1	-169.6	4717.0
Spn4FLyb1	31.33	260.0	Spn4MLyb1	3626.0	19800.0
Spn4FLzb1	228.8	1290.0	Spn4MLzb1	-152.1	-5.448
Spn5FLxb1	109.1	507.2	Spn5MLxb1	37.5	3257.0
Spn5FLyb1	25.85	162.9	Spn5MLyb1	2046.0	13350.0
Spn5FLzb1	146.0	958.6	Spn5MLzb1	-105.7	-4.35
Spn6FLxb1	55.94	363.7	Spn6MLxb1	48.71	1489.0
Spn6FLyb1	15.68	87.17	Spn6MLyb1	764.7	6428.0
Spn6FLzb1	76.03	581.5	Spn6MLzb1	-73.39	-2.036
Spn7FLxb1	21.51	204.5	Spn7MLxb1	18.64	465.0
Spn7FLyb1	6.854	37.09	Spn7MLyb1	210.1	2073.0
Spn7FLzb1	30.87	269.6	Spn7MLzb1	-46.9	-0.6398
Spn8FLxb1	7.533	75.36	Spn8MLxb1	3.348	75.09
Spn8FLyb1	2.478	13.21	Spn8MLyb1	33.38	336.0
Spn8FLzb1	10.67	101.1	Spn8MLzb1	-16.03	-0.1125
Spn9FLxb1	0.2011	3.459	Spn9MLxb1	0.009084	0.1636
Spn9FLyb1	0.08318	0.4892	Spn9MLyb1	0.06526	1.291
Spn9FLzb1	0.3552	3.545	Spn9MLzb1	-0.5307	-0.002515

are extracted from a 10 minutes FAST simulation for integer values between 4 and 25  $m/s$ . At these points of maximum, wind speed is used the LAM PD to created the pressure distribution on the blade shell FEM. After each correction step, maximum and minimum values obtained estimates the boundaries of each parameter's domain. In Table 4.2 are presented all boundaries for each parameter.

Table 4.2: Lower and upper boundary for LAM PD input parameter.

Aerodynamic	Lower Boundary	Upper Boundary	Inertial	Lower Boundary	Upper Boundary
C0	-3.368509	24.176223	AX0	-105.0009	331.4767
C1	-3.368509	24.176223	AX1	-180.0015	568.2458
C2	-112.05878	5.433464	AX2	-884.635	443.0037
C3	-3.445245	46.991469	AX3	-516.4531	1095.643
C4	-37.796476	2.834009	AX4	-322.1667	121.1666
C5	-2.44472	37.255213	AX5	-118.0659	28.28159
C6	-26.237441	1.220251	AX6	-78.74321	295.0361
C7	-0.711944	23.541814	AX7	-316.7837	73.80315
C8	-9.407377	0.250142	AX8	-1197.934	434.8546
C9	-0.010792	0.767064	AX9	-2543.008	7451.595
DFX0	-355791.902084	473263.673336	AY0	-28.40739	277.1644
DFX1	-440608.986341	367762.970092	AY1	-48.69837	475.139
DFX2	-515250.997642	179050.887345	AY2	-891.5786	151.866
DFX3	-303970.564528	236151.182101	AY3	-201.0471	1099.18
DFX4	-233913.614991	218456.614617	AY4	-317.1137	57.21071
DFX5	-134209.981053	147713.943587	AY5	-112.4886	23.36589
DFX6	-41400.928272	95469.442316	AY6	-58.42417	285.4324
DFX7	-18058.178784	36789.483522	AY7	-300.5045	66.44489
DFX8	-4484.623867	16366.00083	AY8	-1167.492	209.1005
DFX9	-3930.281194	4488.128521	AY9	-1342.672	7270.382
DFY0	7218.059077	397259.535114			
DFY1	-434529.665061	650037.60218			
DFY2	-251539.191076	571963.960417			
DFY3	-242072.240823	557112.48932			
DFY4	-194423.920695	474083.946064			
DFY5	-116080.975215	462909.81976			
DFY6	-65719.258813	285909.28482			
DFY7	-16949.46591	185261.08947			
DFY8	-14917.773067	72688.463153			
DFY9	-5614.305239	8602.720734			

### 4.3.2 Surrogate model 1: aero-elastic loads prediction

In LAMs RBE and 4NO, the input domain is a grid of wind speed between  $[4, 25]m/s$  with a step of  $0.02m/s$  giving a total of 1050 simulations. After extracting the maximum value for each simulation, each load has a behavior concerning the wind speed.

The methodology developed for modeling discontinuities, ADM-DNN (refer to chapter 3), is used to predict all 54 loads given the wind speed due to the jump discontinuity that these loads presents. The network is composed of two hidden layers, each of them having 300 neurons.

After the model is validated, it uses a probability density function (PDF) of wind speed to determine the PDF of the corresponding steady aero-elastic loads. This last one will be used as input parameters of the second surrogate model to predict the wind turbine blade's stresses.

To use a real scenario, measurements of wind speed during the last three years (2017-2019) were extracted from open-data Engie Renewable project at La Haute Borne, Grand Est, France [Engie, 2019]. This data is collected for a 2MW wind turbine at a hub height of  $80m$ . The wind turbine used in this study has a hub height of  $119m$  and cut-in and cut-out wind speeds of  $(4, 25) m/s$  respectively. In order to use this measurements, wind speed is translated to DTU hub height using a power law [Touma, 1977] and Hellmann's exponent coefficient  $q = 0.27$  for a condition of unstable air above human inhabited areas [Kaltschmitt et al., 2007]:

$$\bar{u}(z) = \bar{u}(z_{ref}) \left( \frac{z}{z_{ref}} \right)^q \quad (4.31)$$

The resultant predicted loads under each wind speed are obtained using 40,924 wind speed data point. Figure 4.5a shows the wind speed histogram, and figure 4.5b shows the predicted load at one span location acting in the lift direction of the blade.

In LAM PD, there is no need to construct these surrogate models because it does not use the loads directly from FAST as input parameters, different to LAM RBE and 4NO, the parameter used is extracted after applying the method itself. The input parameters are distributed uniformly between their boundary with 1000 samples for each parameter assumed independent. It is assumed a uniform distribution for the parameters, to train the surrogate model 2 or DNN stress.

#### 4.3.2.1 Validation of DNN Loads

For the first surrogate model, the input parameter is the wind speed, and the outputs parameters are the aero-elastic simulated loads. After training the surrogate model, the validation set uses different new points, giving a  $MSE = 1.82e - 05$ , resulting in a good correlation between the predicted and simulated steady load (see figure 4.6).

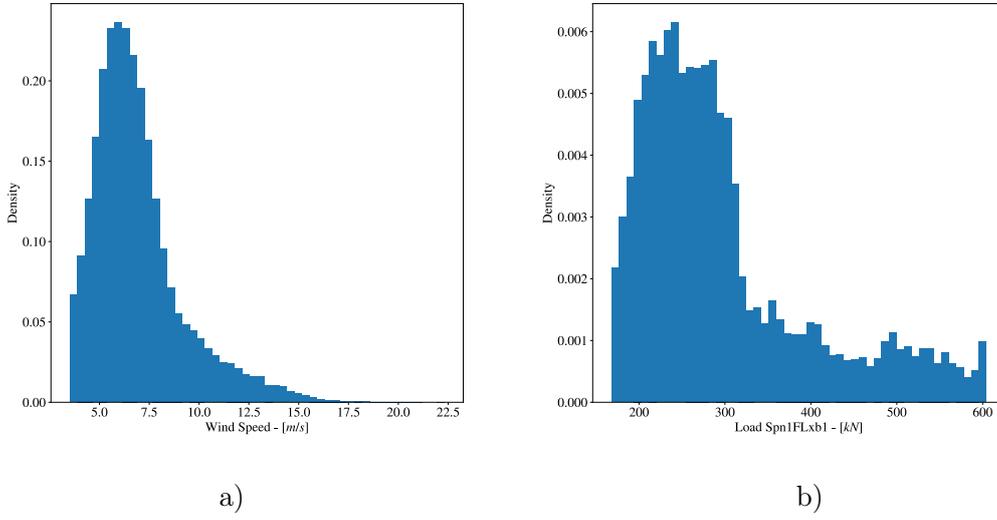


Figure 4.5: a) Wind speed histogram at wind turbine hub height and b) prediction of load histogram at span location 1 acting in X-direction.

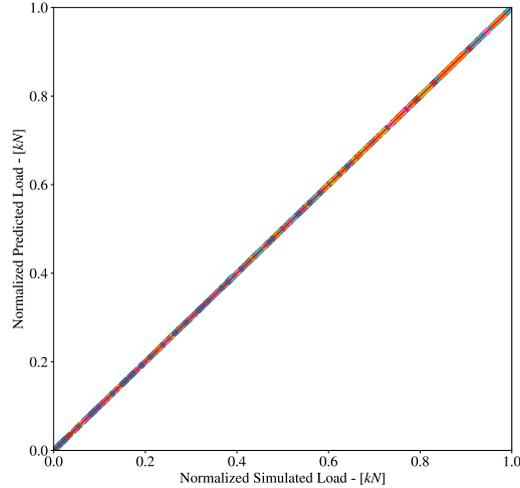


Figure 4.6: Validation of all predicted aero-elastic loads.

As the same model predict all loads (forces and moments), the relative error of each output is plotted in figure 4.7, the maximum relative error is less than  $e_{Spn5Mz} = 0.08$  or 8%. All loads have an excellent fitting (less than 10% of relative error).

After predicting all loads using surrogate model 1, an empirical cumulative density function (ECDF) can be generated for each load and then use the inverse transforming sample [Devroye, 2006] to generate samples with the same probability distribution. To generate a

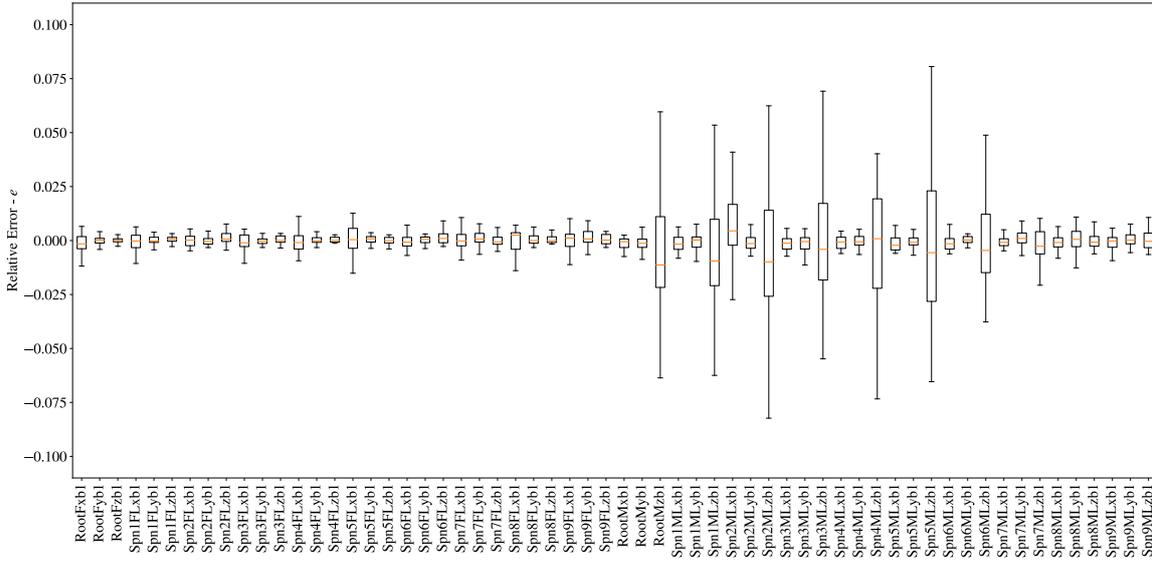


Figure 4.7: Relative error of all predicted aero-elastic loads.

new sample for each load is defined 54 uniform random variable  $(0, 1)$ . The inverse of the ECDF generates random aero-elastic variables corresponding to each load PDF. As shown in figure 4.8, 2000 random points per each input parameter are generated and will be used to train surrogate model 2, which will be used to predict the von misses stress at the blade root. Finally, a global sensitivity will be performed using the prediction of surrogate model 2, taking as input parameters the 54 aero-elastic and 24 material thickness (a total of 78 parameters).

### 4.3.3 LAMs for blade 2D shell model

As the loads are extracted from an aero-elastic simulation code, the blade is modeled by 1D beam FEM, meaning that the extraction of loads is the resultant forces and moments acting in one node of the 1D model. To transfer the 1D load distribution from the beam FEM to a 3D load distribution to be applied to a shell FEM, Caous [Caous et al., 2018] has classified the methods reviewed in the literature into four groups, as listed below:

- Group 1: Application of loads by sections and on one point of each section.
- Group 2: Application of loads by sections but physical distribution on sections.
- Group 3: Continuous application on the blade of an oriented surface load (pressure oriented in a specific direction).
- Group 4: Dissociation of inertial and aerodynamic loads with an application of an acceleration field and pressure distribution across the whole blade.

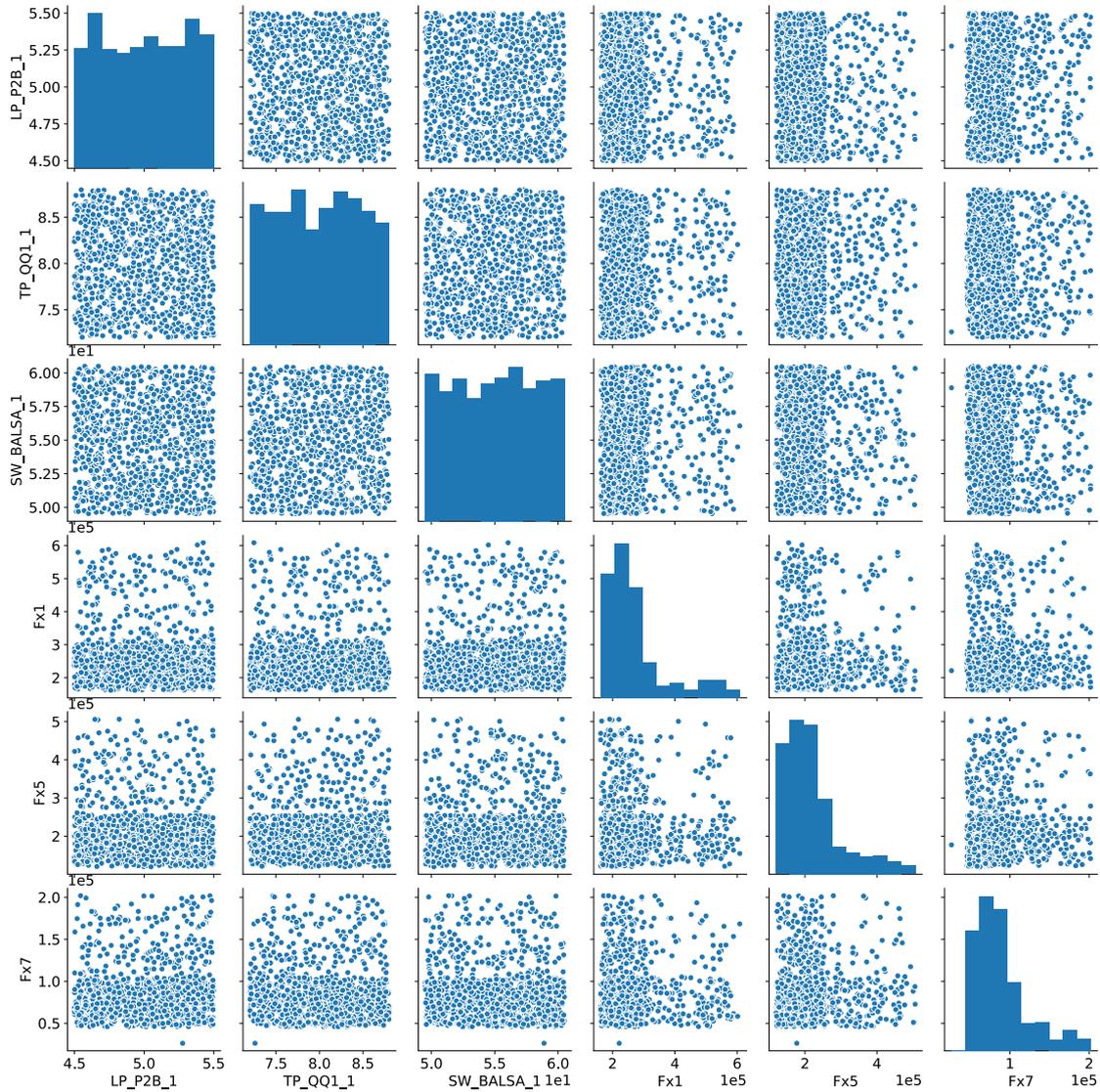


Figure 4.8: Input parameters domain for surrogate model 2, material thickness and aero-elastic loads.

This study compares three LAM: one from the group 1, applying the loads at nine span-wise sections using rigid body elements (RBE), other from the group 2, applying the loads at four nodes per section assuming simplified phenomenological laws [Caous and Valette, 2014], and lastly, one from the group 4, state of the art Caous method creating a pressure distribution across the whole blade [Caous et al., 2018], for further information about this method see the chapter 2.

Lastly, the 1D beam model resulting loads are transferred to the 3D model. A shell finite element model of the blade is created to apply the forces (see figure 4.9). Code-Aster [de France, ] is used to perform the finite element analysis extracting the stress of each node of the blade at the local coordinate system. The  $X$  axis is the principal direction,  $Y$  is the

second direction, and the  $Z$  axis is normal to each shell element pointing to the inside of the blade surface.

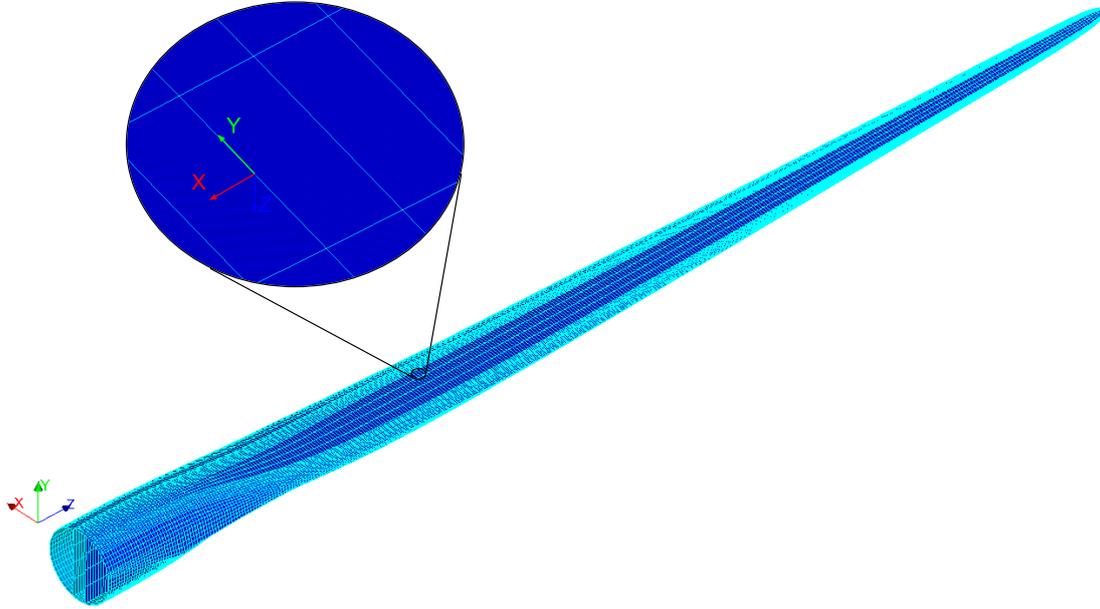


Figure 4.9: 10MW wind turbine blade and shell local coordinate system.

#### 4.3.4 Output parameter for global sensitivity

The blade 2D shell model has many layers of composite material. From each layer can be extracted three levels of stress: inferior, medium, and superior. Considering that the shell model is discretized in 31,648 linear shell elements with 30,613 nodes (see mesh in figure 4.9), the total number of output is more than 400,000. Analyze each output requires a lot of computational calculation time. To make this analysis possible, as found by Hu [Hu et al., 2016b], maximum stresses are located either at the top or bottom layers of the composite laminate. Both LAM methods introduce stress concentration at the applied load's location; only nodes at the blade root are analyzed. This case uses a node located at the top of the root section, bottom layer, and inferior level stress to calculate the Von Mises stress with a general plane stress state considering the stress acting in their local coordinate system.

$$\sigma_{vm} = \sqrt{(\sigma_{11}^2 - \sigma_{11}\sigma_{22} + \sigma_{22}^2 + 3\sigma_{12}^2)} \quad (4.32)$$

Subsequently, Both LAMs (4NO and RBE) will be used to generate the data-set to train and validate the surrogate model 2 (DNN Stress), used to predict the output value without running the blade 2D shell finite element simulation, reducing its computational time.

### 4.3.5 Surrogate model 2: stress prediction at blade root

Due to the high number of inputs parameters (78 variables and 74 variables for LAM RBE and 4NO and LAM PD, respectively), the surrogate model selected for this problem is a deep neural network (DNN). For training the DNN Stress, all inputs and output parameters were scaled between (0, 1) to normalize all parameters magnitude, using 90% of the data to train the model, and the left 10% for validating the model prediction. The DNN is a fully connected network with four hidden layers; each has 250, 250, 200, 200 neurons. Also, all layers have a rectified linear unit (Relu) activation function. For this regression problem, the weight and bias for the DNN are optimized using Adam optimizer [Kingma and Ba, 2014] to minimize the mean squared error of the difference between prediction and real value.

#### 4.3.5.1 Validation DNN stress: high input dimension

Three different networks are used for each case, one for LAM 4NO, other one for LAM RBE as shown in figure 4.10 and the last one for LAM PD shown in figure 4.11 . In all DNNs, the normalized predicted stresses  $\hat{\sigma}_{11}$ ,  $\hat{\sigma}_{22}$  and  $\hat{\sigma}_{12}$  are well correlated with their respective simulated stress. The mean squared error (MSE) and  $R^2$  metrics for each DNN :  $MSE_{RBE} = 0.0016$  and  $MSE_{4NO} = 0.0011$  and  $MSE_{PD} = 0.0028$ ,  $R^2_{RBE} = (0.94, 0.89, 0.94)$ ,  $R^2_{4NO} = (0.97, 0.96, 0.97)$  and  $R^2_{PD} = (0.89, 0.88, 0.96)$  for  $\hat{\sigma}_{xx}$ ,  $\hat{\sigma}_{yy}$  and  $\hat{\sigma}_{zz}$ , representing a good fit for each output by only using 1800 training points for RBE and 4NO and 900 for LAM PD.

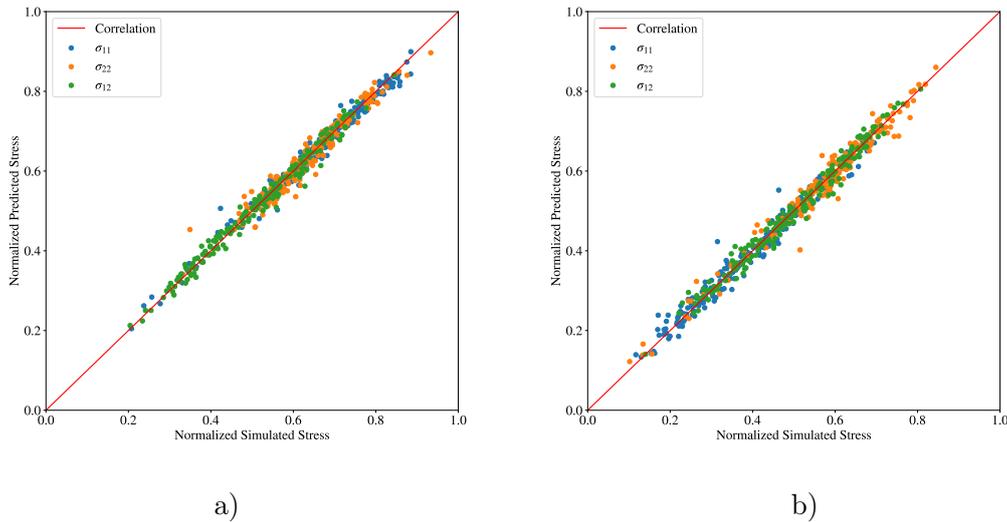


Figure 4.10: Validation of predicted and simulated stress for LAM a) RBE and b) 4 Nodes.

Having all DNN trained and validated, they can perform the Sobol Index sensitivity analysis of both LAMs on the blade root's Von Mises stress.

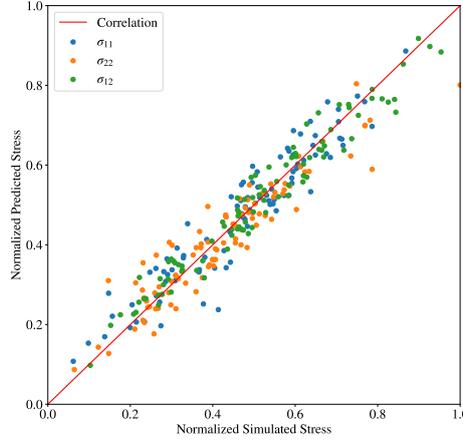


Figure 4.11: Validation of predicted and simulated stress for LAM PD.

### 4.3.6 Sobol sensitivity analysis using DNN stress

#### 4.3.6.1 First order Sobol index: LAMs 4 nodes and RBE

The results showing the Sobol sensitivity analysis are presented next. These results are obtained after generating  $3 \times 10^6$  samples using Saltelli's sample method. The first order indexes for each variable are shown in figure 4.12, the input parameters that have the highest value are forces ( $S_{pn6FLxb1}, S_{pn7FLxb1}, S_{pn8FLxb1}, S_{pn9FLxb1}$ ) and moments ( $S_{pn6MLyb1}, S_{pn7MLyb1}, S_{pn8MLyb1}$ ) for both methods, which are applied between the center and the tip of the blade and acting in edge-wise direction, also only for RBE, moments ( $S_{pn7Mzb1}, S_{pn8MLzb1}$ ) presents linear interaction with the output.

#### 4.3.6.2 First order Sobol index: LAM PD

The result obtained after generating  $3 \times 10^6$  samples using Saltelli's method and transferring the loads using LAM PD is shown in figure 4.13. The parameters that present the highest sensitivity are ( $C5, C6, DFY6, DFY7, DFY8, DFY9$ ). The location of these parameters corrects the loads acting in the blade Y-direction (lift) are near the tip of the wind turbine blade. For all other parameters used in this analysis, their sensitivity is closed to zero. The maximum value is found in the parameter  $DFY8 = 0.4086$ , more significant than all values found for each LAM tested.

Comparing all LAMs, PD shows a more significant first order  $S1$  interaction with the output than RBE and four nodes methods. Meaning that aero-elastic loads applied using PD have a more linear interaction with this output than applying the loads using RBE and 4 Nodes LAM. Nevertheless, only seeing  $S1$  cannot be concluded that one method is less

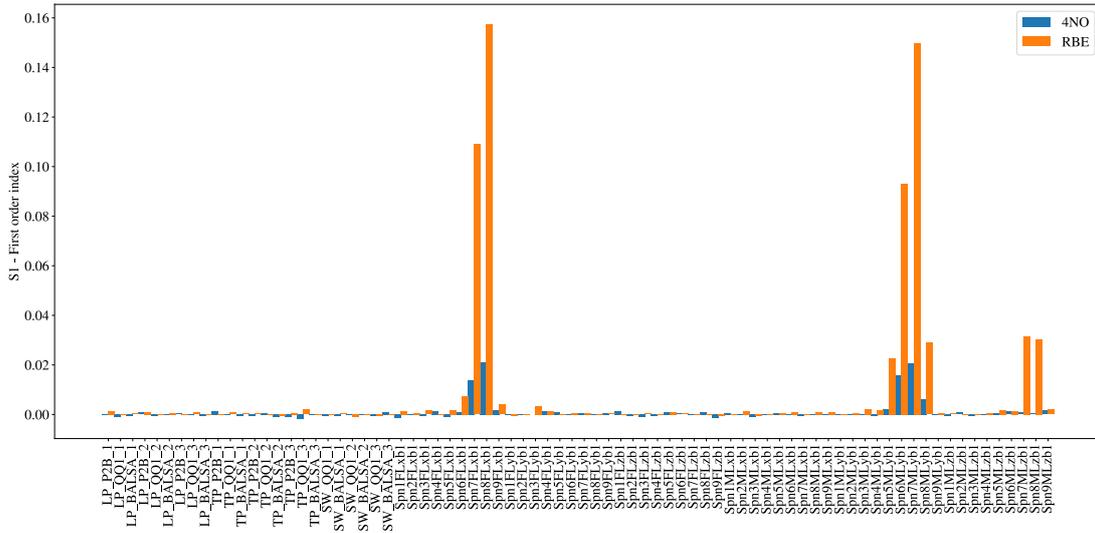


Figure 4.12: Comparison of first order Sobol index for RBE and 4NO methods to estimate the detailed stress distribution in a blade shell FEM.

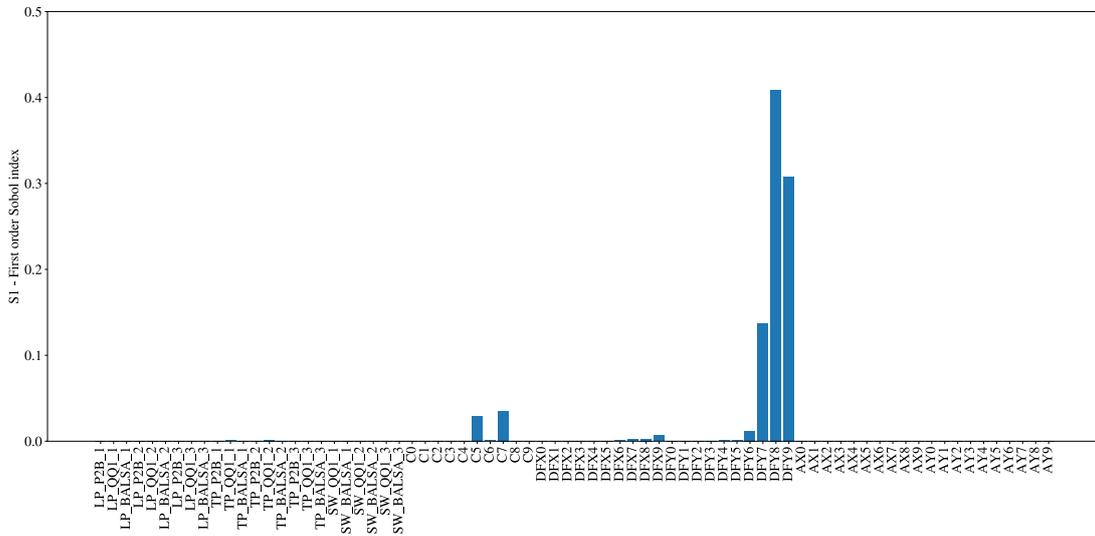


Figure 4.13: First order Sobol index using LAM PD to estimate the detailed stress distribution in a blade shell FEM.

sensitive than others. For these reasons, the next step is to analyze the total order *ST* Sobol index to calculate interactions between variables and their effect in the output.



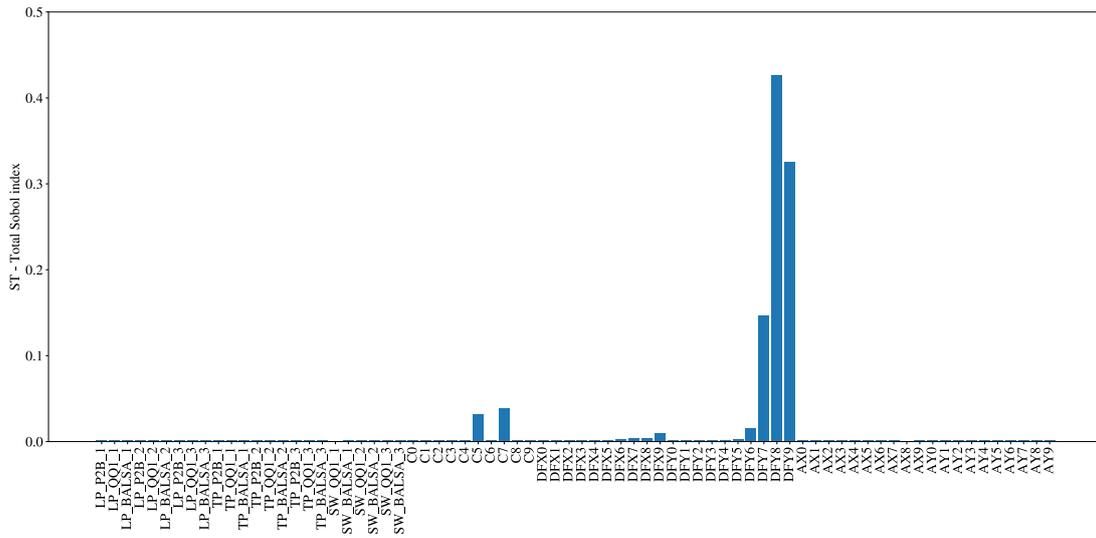


Figure 4.15: Total order Sobol index using LAM PD to estimate the detailed stress distribution in a blade shell FEM.

ted in figure 4.16. The highest values are presented in the interaction between these loads (*Spn7FLxb1*, *Spn8FLxb1*, *Spn6MLyb1*, *Spn7MLyb1*). Some of these loads also present a low interaction with input parameters with a low first and total index order.

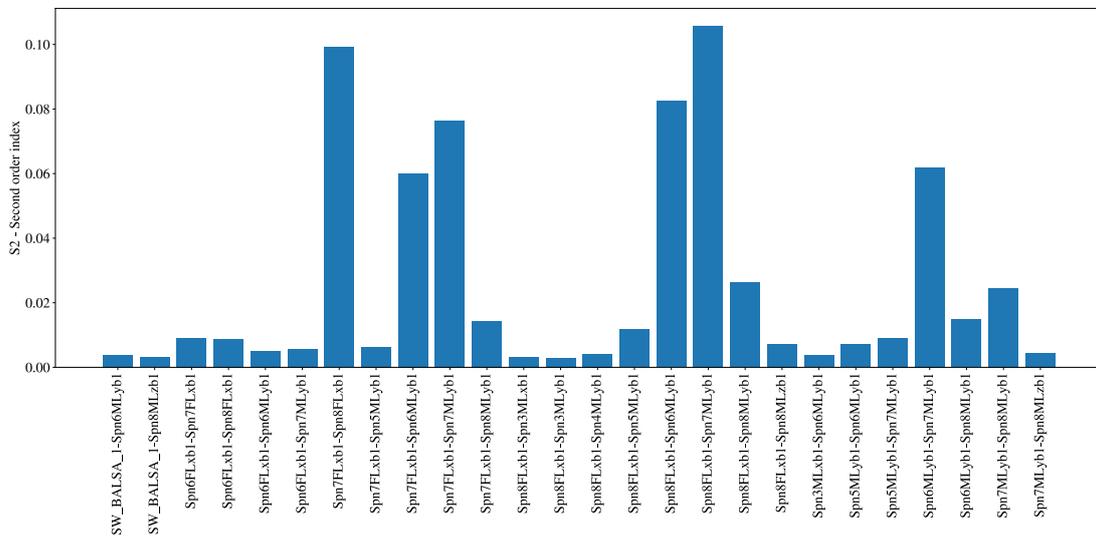


Figure 4.16: Second order Sobol index for Von Mises stress at blade root using LAM 4 node.

To see the second-order indexes for LAM RBE, these are presented in figure 4.17. Only the interactions greater than 0.003 are plotted, presenting a high interaction between loads *Spn6FLxb1*, *Spn7FLxb1*, *Spn8FLxb1*, *Spn6MLyb1*, *Spn7MLyb1*, *Spn8MLyb1*. Using this method, the interaction occurs between variables that have a high *S1* Sobol index.

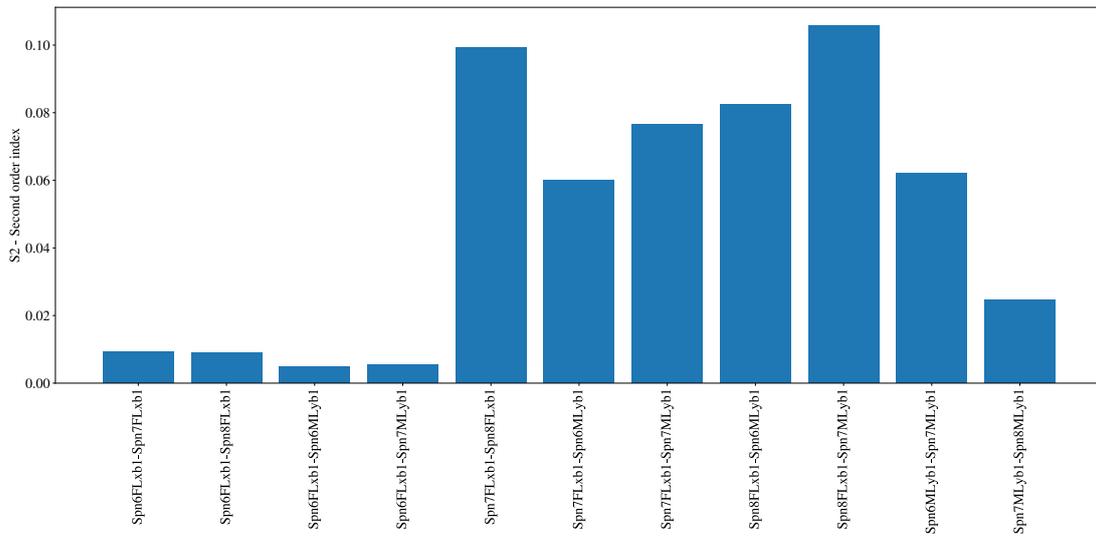


Figure 4.17: Second order Sobol index for Von Mises stress at blade root using LAM RBE.

Finally, the second-order indexes for LAM PD are presented in figure 4.18. Only the interactions greater than 0.002 are plotted due to the low Sobol index value, presenting a high interaction between loads  $DFY8, DFY9$ , and  $AX$  and  $AY$  at different locations. The maximum value of  $S2$  is  $DFY8 - DFY9 = 0.00842$ , which is a short interaction compared to LAM RBE and 4NO. These results show the low non-linear interaction, showing that only the linear interaction of the parameter and the output is significant.

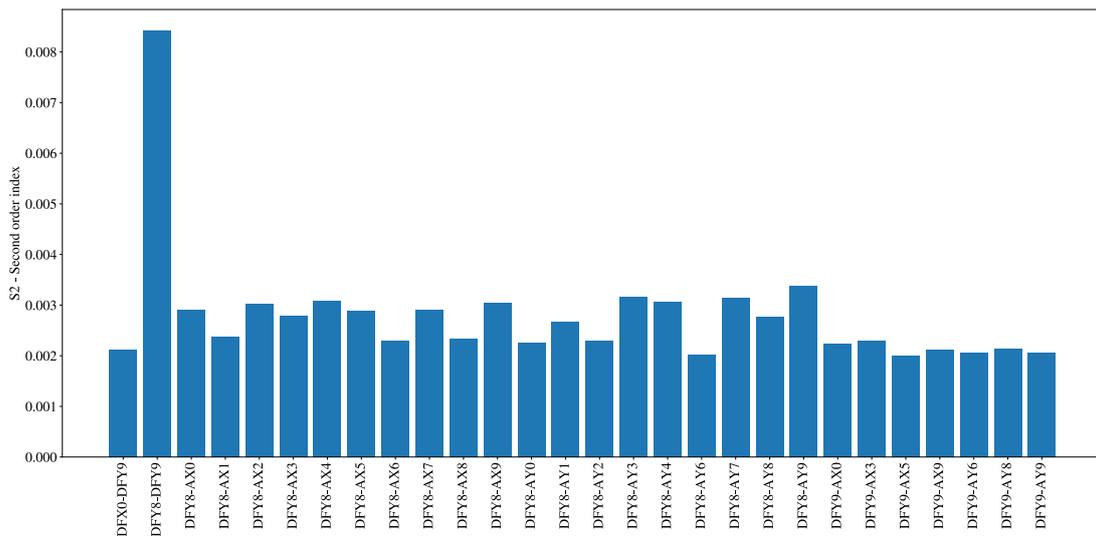


Figure 4.18: Second order Sobol index for Von Mises stress at blade root using LAM PD.

### 4.3.7 Comparison of LAM's sensitivity

Regarding the first-order Sobol index LAM PD has the highest value, then LAM RBE, and lastly, LAM 4NO.

LAM 4NO presents a high value in higher-order interaction between variables than LAM RBE and LAM PD; this is due to how each method applies the loads. LAM RBE limits the degrees of freedom of each section that aero-elastic loads are applied. This method can give a higher linear interaction between input and output parameters. Stronger linear interaction than RBE is found using LAM PD, which applies a pressure distribution across the whole blade. Differently to LAM 4 nodes that leave each section free to be deformed. The loads are applied distributed in 4 points of the cross-section, resulting in an interaction between the loads applied towards the output parameter.

This study presents only  $S_2$  Sobol index. Other higher-order interactions between input parameters are missing. They could be estimated in the  $ST$  index as the  $S_2$  for LAM 4 nodes has more high-order interactions with inputs parameters that do not present a significant  $S_1$  index. A surrogate model could be created with the most sensible variables (reducing the number of inputs parameters) and used to calculate the wind turbine blade's reliability. As these variables interact with other variables, the total effect will be lost, and the surrogate model will add uncertainty. Unlike LAMs RBE and PD, these last ones can be represented with a surrogate model with only the most sensible parameters because all interaction occurs between these variables. For reliability analysis or estimation of the fatigue life, this representation will not introduce uncertainty by itself; only the problem's uncertainty will be present.

LAM PD is the method that introduces less sensitivity overall because it presents the less non-linear interaction between variables compared to RBE and 4NO (RBE has the second-lowest sensitivity). In other words, the method more suitable to transfer the loads from an aero-elastic beam simulation to a blade shell model from a sensitivity point of view is LAM PD, then RBE, and lastly, LAM 4NO. These results are similar to the one found by Caous [Caous et al., 2018], comparing Puck's damage using these LAMs, having LAM PD as the method with the better damage distribution along the blade.

### 4.3.8 Morris Analysis: LAM 2D shell FEM

To validate the results found from Sobol analysis with a surrogate model for high dimension problem, the ideal way is to analyze the Sobol sensitivity of the original model (i.e., blade 2D shell model), but this model is costly to run and to realize three millions of simulations is unaffordable. In this case, as found by Campolongo [Campolongo et al., 2007], there is a relationship between  $\mu^*$  and  $S_T$ , a Morris "screening" analysis is performed in the 2D shell FEM of the wind turbine blade, and will be used as validation of the results obtained by Sobol analysis using surrogate models. In this analysis, all loads variables have an upper and lower boundary with a  $\pm 10\%$  of their mean value and are performed using 1000 samples for each

input parameter. Subsequently, each sample is used to launch the 2D shell FE analysis that gives the resultant stress on the blade root. Using these outputs, the Morris analysis gives the results shown in figure 4.19. Both results are normalized between (0,1) to compare then on the same scale.

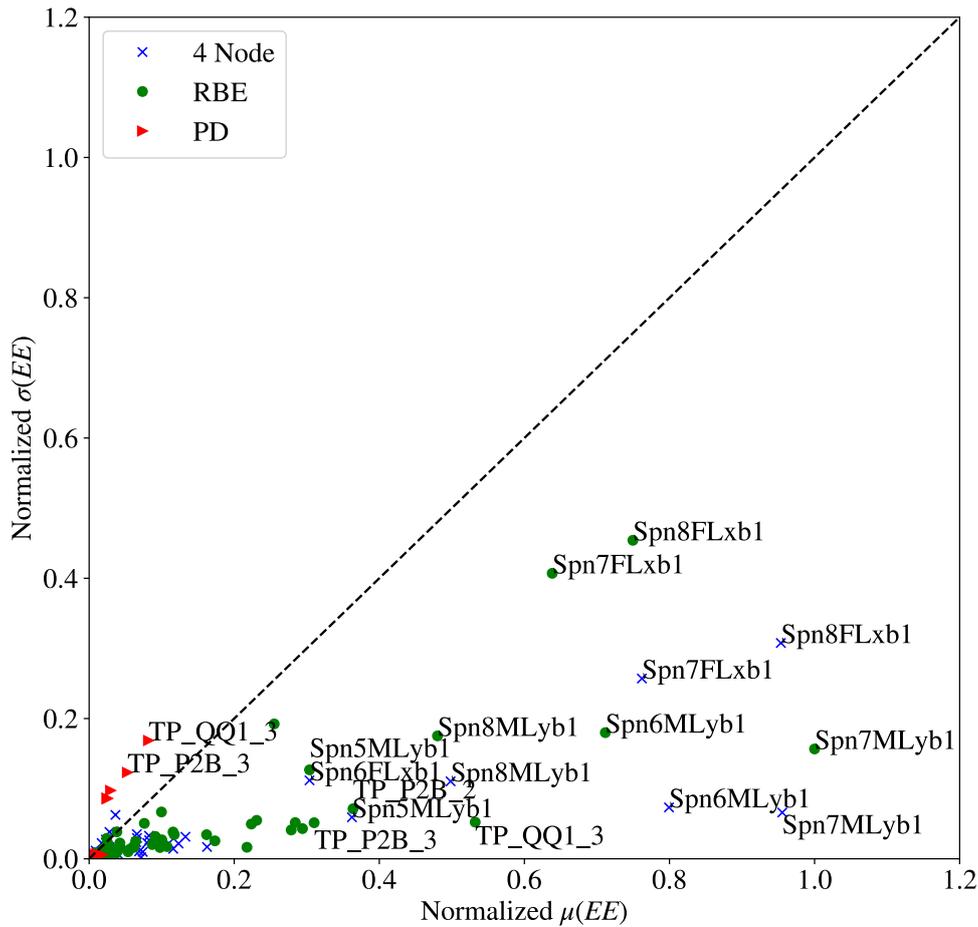


Figure 4.19: Morris sensitivity analysis for LAM: RBE, 4 nodes and PD.

As can be seen for methods RBE and 4NO, the most sensible variables are [ $Spn7FLxb1$ ,  $Spn8FLxb1$ ,  $Spn6MLyb1$ ,  $Spn7MLyb1$ ,  $Spn8MLyb1$ ] with a linear effect and for LAM PD the most sensible variables are [ $TP_{QQ1_3}$ ,  $TP_{P2B_3}$ ]. Other variables present a less linear effect, and most of them, their effect can be neglected. The fact that Morris analysis is a quantitative method is performed in a subset of the input domain. Only one input variable is changed at each time. The full sensitivity effect of each parameter cannot be estimated. For LAM PD at this sub-domain, the load parameters do not introduce sensitivity, only material parameters, showing in the Morris analysis less sensitivity than the other two methods (see figure 4.20).

The interaction between two or more variables cannot be captured by this analysis, and these interactions could help us to understand better the relationship between variables that affect the sensitivity of the models and select the adequate LAM to estimate the fatigue life of the wind turbine blade.

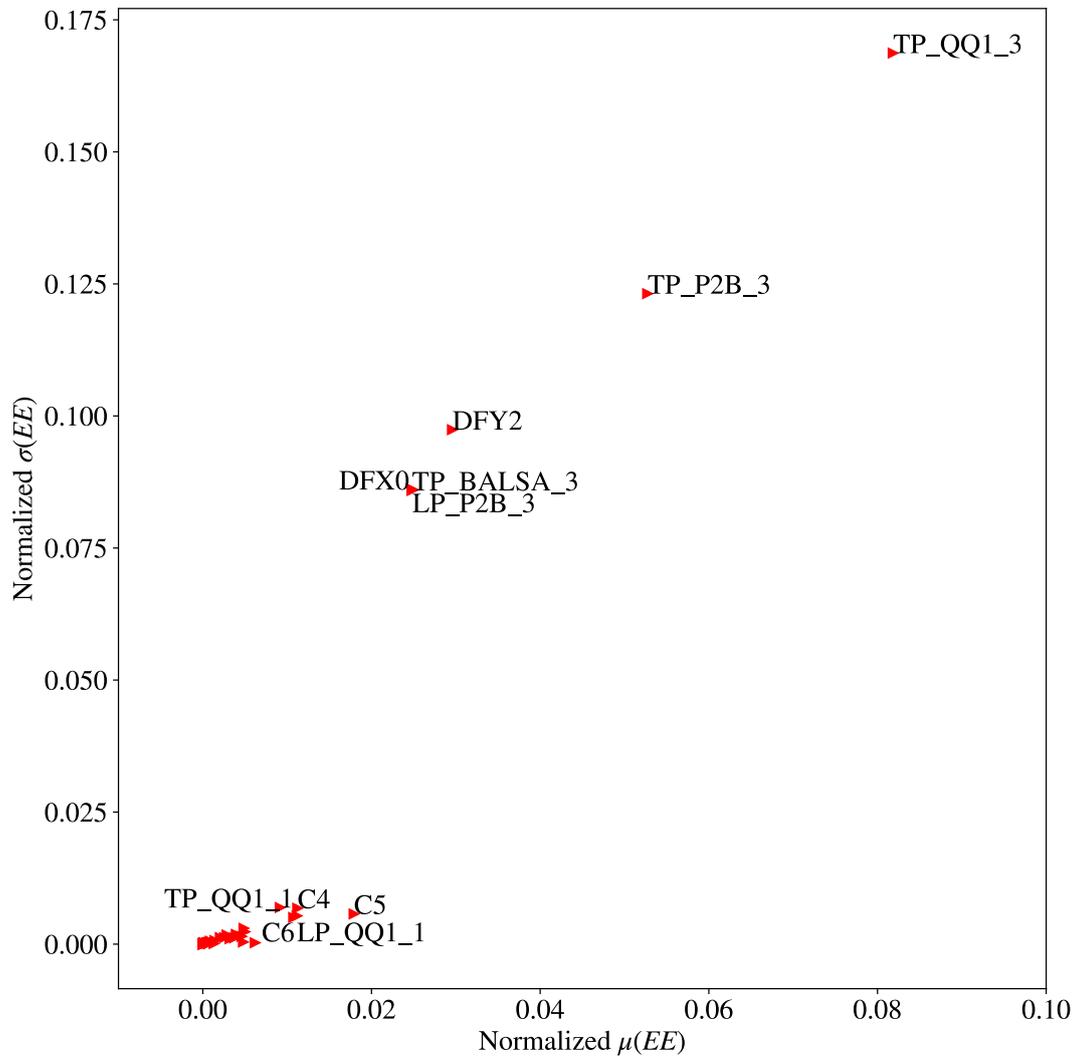


Figure 4.20: Zoom of Morris sensitivity analysis for LAM PD.

#### 4.3.8.1 Validation of Sobol sensitivity analysis

Morris method is a qualitative method calculated using the blade shell FEM, and the Sobol method is a quantitative method performed using a surrogate model of the blade shell FEM; they estimate the results in different ways. However, their sensitivity results present the same input variables to be the most sensible for  $\mu^*$  and  $S_T$ , meaning that the estimation made by the global sensitivity approach based on DNN is comparable to the obtained by using the

2D blade shell FEM Morris analysis. If the Sobol analysis is performed using the 2D shell blade FEM, the CPU calculation time will increase dramatically. Using the 2D shell FEM to perform 1000 iterations FEA for Morris analysis, it takes near 72 hours, and for the Sobol analysis was used  $3 \times 10^6$  iterations, meaning that it takes 216,000 hours. This time could be reduced by using distributed processing and a more robust and modern CPU. Compared to training the DNN, it only takes 35-45 seconds, and for the Sobol analysis, it takes less than 5 minutes and shows a great advantage for high dimensional problems as the one treated in this study.

The next chapter analyzes the fatigue damage of the composite wind turbine blade, in this case, is used by LAM PD and LAM 4NO to compare their effectiveness in the fatigue damage estimation.

## 4.4 Conclusions

The methodology proposed for calculating the global sensitivity using a deep neural network as the surrogate model is applied to a mechanical case of load application methods sensitivity. After trained and validated, both surrogate models presented a low mean squared error in low and high-dimension input parameters. This methodology deals with the quantitative sensitivity of high-dimensional problems, representing an advantage over qualitative methods using the same amount of samples. The result obtained using the deep learning based sensitivity approach is validated using a qualitative Morris sensitivity analysis using the wind turbine blade's shell finite element model. This methodology could be used for other wind turbine structure parts as the jacket and tower.



# Fatigue life estimation of wind turbine blades using DNN

## Sommaire

<b>5.1</b>	<b>Aero-elastic unsteady simulation</b>	<b>106</b>
5.1.1	Environmental conditions	106
5.1.2	FAST simulation with filtered response	108
<b>5.2</b>	<b>LAM wind turbine blade 2D shell FEM for stress analysis</b>	<b>110</b>
5.2.1	Stress analysis: LAM 4NO	111
5.2.2	Stress analysis: LAM PD	112
<b>5.3</b>	<b>Multi-axial fatigue criteria for composite materials</b>	<b>115</b>
5.3.1	S-N curves model	117
5.3.2	Constant Life Diagrams - CLD	117
5.3.3	Fatigue damage accumulation	124
<b>5.4</b>	<b>DNN: 10-minutes damage</b>	<b>124</b>
5.4.1	Input domain	124
5.4.2	DNN description	126
5.4.3	DNN damage: hyperparameters optimization	126
5.4.4	DNN: 10-Minutes damage prediction	127
5.4.5	Blade damage distribution	131
<b>5.5</b>	<b>20 years fatigue damage estimation using DNN</b>	<b>138</b>
5.5.1	20 years fatigue damage results for LAM PD	140
<b>5.6</b>	<b>Probabilistic fatigue life estimation for LAM PD</b>	<b>143</b>
<b>5.7</b>	<b>Conclusions</b>	<b>145</b>

The estimation of the fatigue damage of a wind turbine blade depends on how the loads on the wind turbine blade are applied and the fatigue damage criteria. Some authors use the loads directly from aero-elastic simulations and calculate a damage equivalent load that estimates the damage at one node of the beam 1D blade model, which is not a detailed analysis of the blade. Others create a blade 2D shell FEM but apply the loads as a constant pressure equivalent to a maximum moment, which leads to the evaluation of one direction of the blade. Others create a more realistic pressure distribution using computational fluids dynamics models. However, these models do not consider the response of the blade pitch

control or the dynamic behavior of the blade. On the other hand, fatigue criteria based on an energy approach for composite materials are not accurate; for this kind of material, criteria based on phenomenological events are more suitable.

This chapter presents a fatigue damage procedure for a composite wind turbine blade using a deep neural network to estimate 10 minutes of damage and the probabilistic estimation of the fatigue life until failure, which includes: wind field simulation, coupled aerodynamic, elastic, and electrical, and control (servo) analysis, filter aero-elastic loads, stress analysis by finite element analysis, and fatigue damage evaluation based on deep neural network surrogate model considering a multi-axial non-proportional stress state. Using the proposed methodology, a 10 minutes fatigue damage is determined by a 10 minutes aero-elastic simulation assuming a normal turbulence model for the environmental conditions and used to estimate the probabilistic fatigue life. In Section 5.1 is presented the aero-elastic simulations for unsteady conditions of a composite wind turbine blade. The stress distribution on the wind turbine blade shell FEM using load application methods: 4NO and PD are presented in Section 5.2. In Section 5.3 is presented the methodology to calculate the fatigue damage using multi-axial criteria for non-proportional loading using constant life diagrams for composite materials. Section 5.4 explains the methodology to estimate 10 minutes of fatigue damage using a deep neural network. Also, the estimation of 10 minutes of damage for different wind turbine blade locations is compared to the fatigue damage obtained by using different load application methods. Section 5.5 presents the fatigue damage of the blade for 20 years for different composite plies, and Section 5.6 presents the estimation of the fatigue life of the composite wind turbine blade. The node with the maximum fatigue damage follows a normal distribution for its fatigue life.

Figure 5.1 presents the methodology to estimates the fatigue damage in 10 minutes followed in this study.

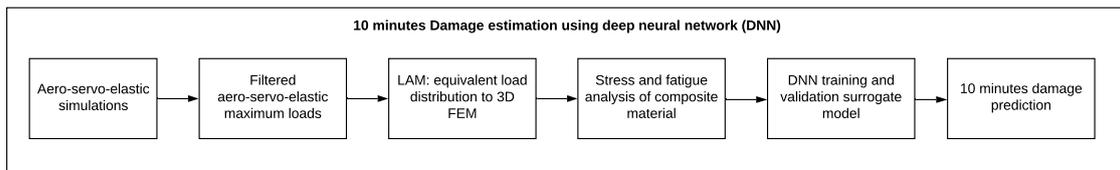


Figure 5.1: Methodology proposed for 10 minutes fatigue damage estimation.

## 5.1 Aero-elastic unsteady simulation

### 5.1.1 Environmental conditions

Measurements of wind speed during the last three years were extracted from the open-data Engie Renewable project at la Haute Borne, Grand Est, France [Engie, 2019]. This data is collected for a 2MW wind turbine at a hub height of 80m. The wind turbine used in this study

has a hub height of  $119m$  and cut-in and cut-out wind speeds of  $[4, 25]$   $m/s$ , respectively. The same translation is done as in the sensitivity analysis (see chapter 4).

The dependency between wind speed ( $WS$ ) and the standard deviation of the wind speed ( $\sigma_1$ ) is defined in the Normal Turbulence Model described in the IEC 61400-1 [Standard et al., 2005]. In our study case is used a reference ambient turbulence intensity of a site Class 1A:  $TI_{ref} = 0.16$ . This dependency is given by the local statistical moments of  $\sigma_1$  as:  $\mathbf{E}(\sigma_1|WS) = TI_{ref}(0.75WS + 3.8)$  and  $\mathbf{V}(\sigma_1|WS) = (1.4TI_{ref})^2$ .

The parameters of the  $\sigma_1$  density probability distribution are given in the following equations as a function of  $WS$ .

$$\sigma_{\sigma_1} = \left[ \ln \left( \frac{\mathbf{V}(\sigma_1|WS)}{\mathbf{E}^2(\sigma_1|WS)} + 1 \right) \right]^{1/2} \quad (5.1)$$

$$\mu_{\sigma_1} = \ln[\mathbf{E}(\sigma_1|WS)] - \frac{\sigma_{\sigma_1}^2}{2} \quad (5.2)$$

The correlation between  $WS$  and mean shear exponent  $\alpha$  is based on the simplified joint distribution defined by Dimitrov et al. [Dimitrov et al., 2015].

$$\mu_{\alpha} = 0.088(\ln(WS) - 1) \quad (5.3)$$

$$\sigma_{\alpha} = (1/WS) \quad (5.4)$$

In table 5.1 is summarized all inputs parameters for the fatigue analysis with their corresponding probability density function (PDF) and their parameters.

Table 5.1: Wind turbine surrogate model inputs parameters.

Input Parameter	PDF	PDF Parameter
10-min mean hub height wind speed	Weibull	$\mathbf{E}(WS) = 7.28m/s$
Std. of the wind speed during the 10-min simulation	Lognormal	$\mu_{\sigma_1}(WS), \sigma_{\sigma_1}(WS)$
10-min mean shear exponent	Normal	$\mu_{\alpha}(WS), \sigma_{\alpha}(WS)$

The 10-minute turbulence intensity is defined as the ratio of the standard deviation of wind speed to the mean wind speed, determined from measured data of wind speed in 10 minutes [Commission et al., 2019] as

$$I_{10} = \frac{\sigma_{10}}{WS_{10}} \quad (5.5)$$

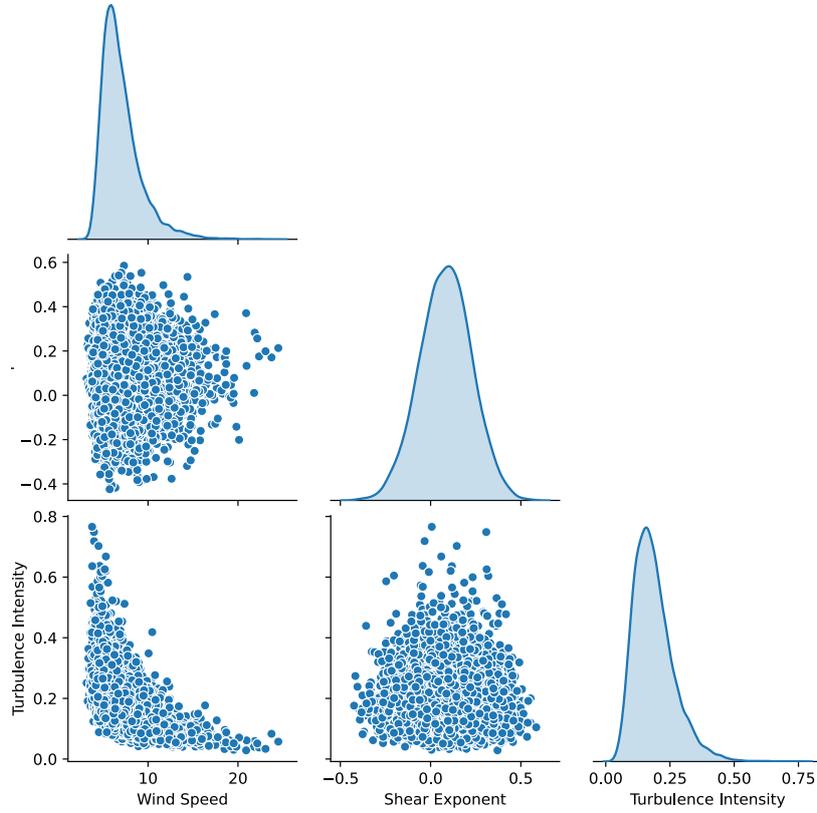


Figure 5.2: environmental condition parameters distribution.

As found in the sensitivity analysis (see chapter 4) the most sensible variables are found in the short term wind loads applied on the blade, rather than the material thicknesses. In this case, for the fatigue analysis only uncertainties on the wind loads are considered. Other studies [Slot et al., 2020] take into account variation in multiple variables of the wind condition as: mean wind speed, standard deviation of the wind speed, shear exponent, wind direction, miss-align angle. This study only considers uncertainties in the 10 minutes mean wind speed to decrease the complexity of the problem and also reduce the sample size of simulations, decreasing the computational time. This simplification could lead to an under estimation of the fatigue damage of the wind turbine, however it is considered as an initial step to evaluate and validate the proposed methodology. For the generation of the 3D wind field the turbulence intensity and the shear exponent are constant defined in the IEC standard [Standard et al., 2005].

### 5.1.2 FAST simulation with filtered response

The environmental conditions described before are inputs to FAST [Jonkman and Buhl Jr, 2005] and an effective 10 mins unsteady study (i.e., neglecting the simulation start due to abrupt response) with a time step  $\Delta t = 0.015s$  (see figure 5.3) is launched for wind speeds between  $[4, 25]m/s$  generated using inverse method sampling from the measured wind speed. A tur-

bulent simulation using the design load case (DLC) 1.2 for normal turbulence model (NTM) [Standard et al., 2005] is made using a full-field flow that contains a proper spatio-temporal turbulent velocity field generated using TurbSim [Jonkman and Buhl Jr, 2006] with an IEC Kaimal spectral model and a coherence function for IEC spectral models (refer to chapter 2). Afterward, the generated wind field is used as a FAST input to simulate the turbulent inflow environment that incorporates many of the important fluid dynamic features known to adversely affect turbine aero-elastic response and loading. The number of simulations realized is 1 per each integer wind speed in the range  $[4, 25)m/s$  (i.e., not including  $25m/s$ ), resulting in a total of 21 FAST simulations.

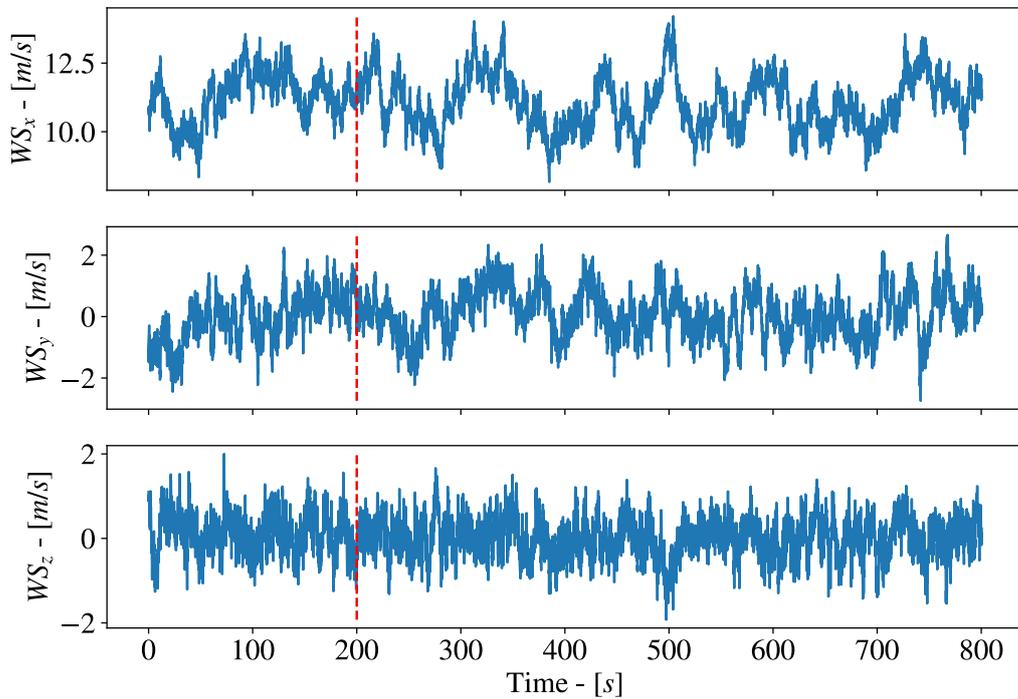


Figure 5.3: Turbsim wind field generation for wind speed  $11m/s$  and normal turbulence, DLC 1.2.

Subsequently, a time history response of the forces and moments acting on ten nodes of the blade is extracted as outputs from the FAST model, resulting in 60 outputs variables.

After 200 simulated seconds, these loads are re-sampled to reduce the time step to  $\Delta t = 0.63s$  of the resulting time history for each wind speed (see figure 5.4). Then, it applies each time instant to the 2D FEM blade to create a high fidelity response resulting in 21,000 finite element analysis.

The filter used to re-sample the aero-elastic response is based on the Discrete Fourier Transformation (DFT) [Arfken and Weber, 1999, Weisstein, 2002]. The DFT is the equivalent

of the continuous Fourier Transformation (FT) for signals only knowing at  $N$  instants separated by sample times  $T$ , a finite number of data points. The FT of an original signal,  $f(t)$ , would be

$$F(j\omega) = \int_{-\text{inf}}^{\text{inf}} f(t)e^{-j\omega t} dt \quad (5.6)$$

Considering  $N$  samples denoted as  $f(0), f(1), \dots, f(k), f(N - 1)$ . Each sample  $f(k)$  could be regarded as an impulse having area  $f(k)$ . Then, since the integrand exists only at the sample points the DFT could be defined as

$$F(j\omega) = \sum_{k=0}^{N-1} f(k)e^{-j\omega kT} \quad (5.7)$$

This method is used to down-sample the original signal by assuming is periodical, changing the sample space from  $dt$  to  $dt \cdot \text{length}(T)/M$ , where  $M$  is the new sample number. After re-sampling the loads, there is little reduction in the maximum and minimum value of the output, also the low frequencies of the signal are eliminated, which will underestimate the total damage produced by this loads. However, this reduction is not important for the magnitude of the loads and as only the low frequencies are eliminated the signal response is well represented for the fatigue analysis. The filtered signal is used for further analysis.

## 5.2 LAM wind turbine blade 2D shell FEM for stress analysis

Two approaches are selected to analyze the stress on the blade 2D shell model: from group 2, LAM 4 Nodes (LAM 4NO), and group 4, the Caous method for pressure distribution (LAM PD). Both methods transfer the aero-elastic loads of time history to the blade 2D shell model. The 2D shell FEM is used for each time step to calculate the blade's stress time history.

The open-source FEA *Code\_Aster* calculates the stresses by applying the load conditions from the aero-elastic simulation at each time instant after re-sampling the load-time history. Shell DST elements are used to calculate the plane stress components. It is fixed six degrees of freedom of the blade root boundary in the FEA.

In this study, the stress distribution is calculated at layer positions, indicating specific locations through the laminate thickness. Each layer has three different positions to extract the stress: superior, middle, and inferior (SUP, MID, INF, respectively). Meaning that for a node in the shell FEM is associated multiples layer position starting from the surface of the blade shell towards the blade center.

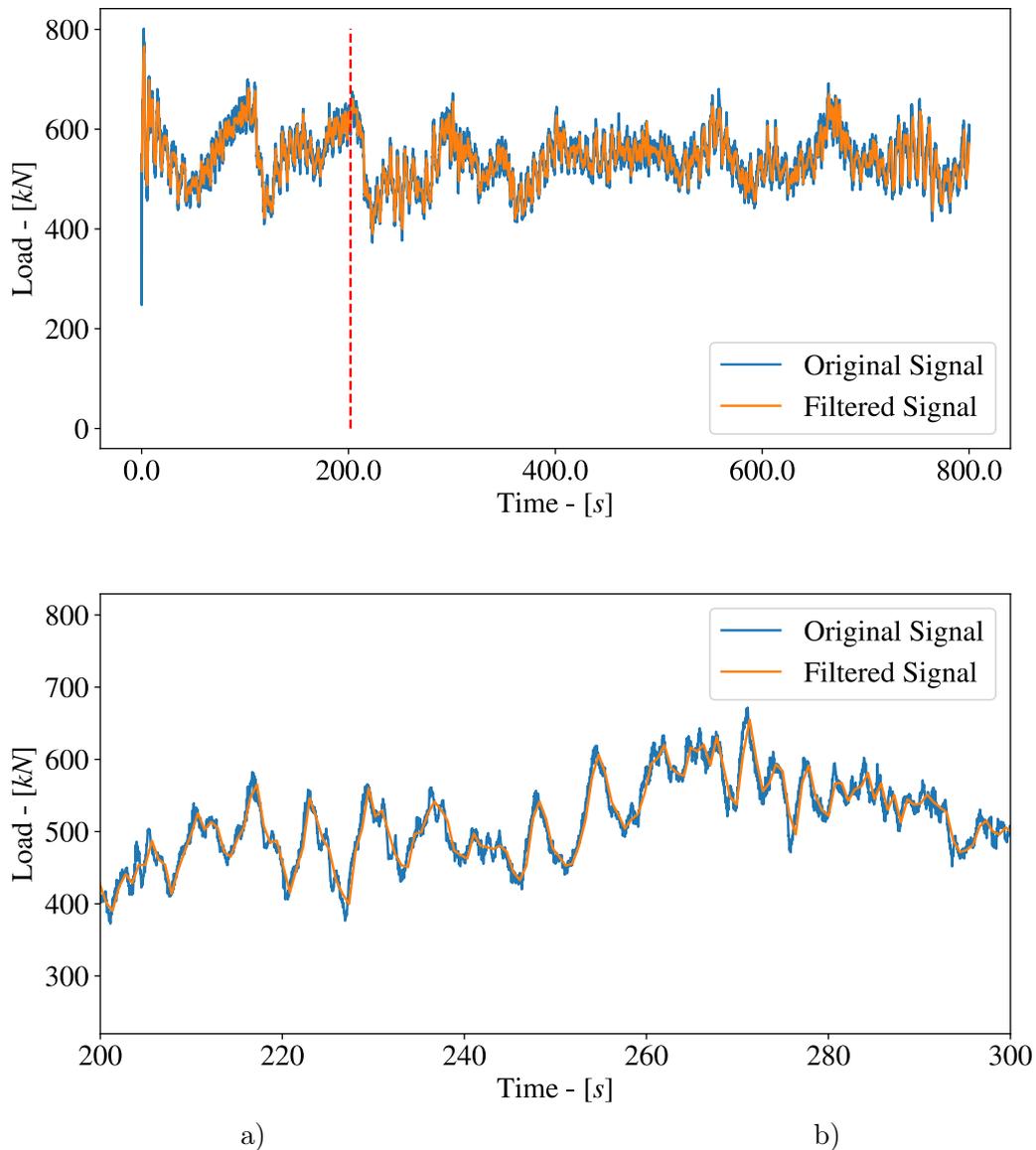


Figure 5.4: FAST load output time history for *Spn1FLxb1* at  $11\text{m/s}$  with a time step of  $0.015\text{s}$  and re-sampled to a time step of  $0.63\text{s}$ , and a zoom between 200 and 300 seconds.

### 5.2.1 Stress analysis: LAM 4NO

As an example, the longitudinal  $\sigma_{11}$  stress distribution through node 606 of the blade root (see figure 5.5 using LAM 4NO approach is presented in figure 5.6, the stress distribution is analyzed for different wind speed. It is found that the stress is piece-wise linear varied at each ply of the laminate thickness, which is consistent with the classical laminate theory of composite materials. Also, almost all wind speed, the maximum stress at positions located in the second and fourth ply, has QQ1 as material (see figure 5.6a). However, for some wind

speed, the stress distribution has strong variations at each ply, regarding from a middle point (i.e., Third layer, MID position). In this case, no assumption could be made to select a layer where the stress is always maximum needs to analyze all layers of positions (see figure 5.6b).

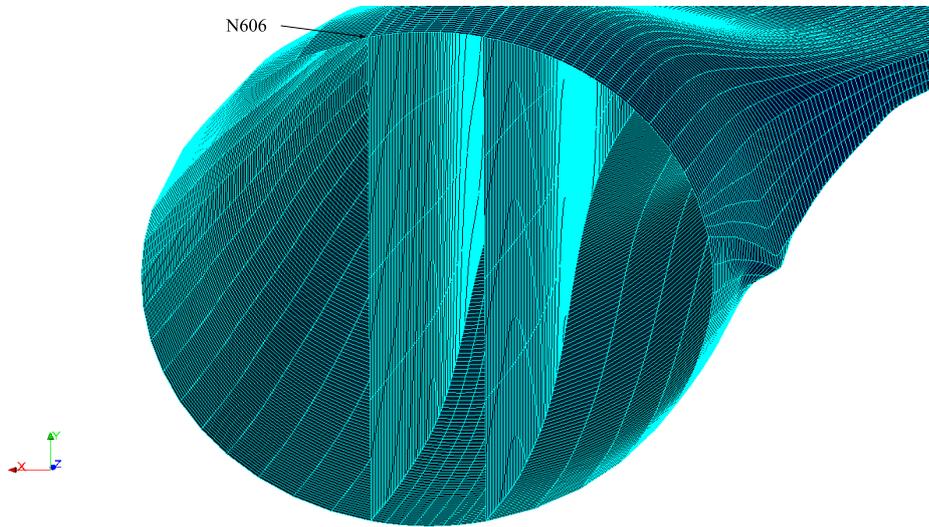


Figure 5.5: Location node 606 on blade root.

In the same node of the blade root is illustrates the stress time history at the second ply (QQ1 material) at an inferior level (see figure 5.7). As can be seen, for each of the directions in the shell's local coordinate system, the stresses magnitudes are different and changing through time, presenting a complex multi-axial stress behavior. However, they have an harmonic response at each direction. This response is mainly due to the rotation of the blade because as the wind speed signal was filtered some high frequencies that produce turbulence in the model are neglected, creating a less turbulent wind field. To have a fully turbulent field is recommended to use the original wind speed signals but it will increase the computational time.

### 5.2.2 Stress analysis: LAM PD

At the same node is analyzed the  $\sigma_{11}$  stress distribution using the LAM PD approach for different wind speeds. In LAM 4NO, the stress is piece-wise linear varied at each ply of the laminate thickness. Moreover, the absolute maximum stress location is always at the second ply at the INF position (QQ1 material) for all wind speed.

Also, as LAM 4NO, LAM PD presents a similar behavior. However, its stress magnitude is smaller. The maximum stress for both LAM is found acting in the transversal direction of the local coordinate system  $\sigma_{22}$ .

Both methods are consistent with the classical laminate theory of composite material. However, the stress distribution is different in magnitude and for some wind speed also in variation through the laminate thickness. These differences may lead to different fatigue

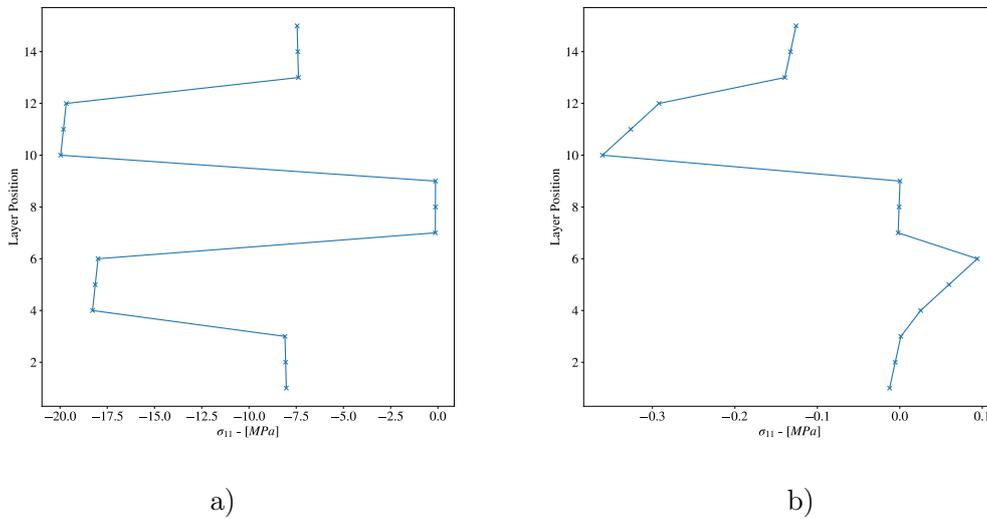


Figure 5.6: Longitudinal  $\sigma_{11}$  stress distribution through laminate thickness at node 606 using LAM 4NO for a) wind speed 11m/s and b) wind speed 16m/s.

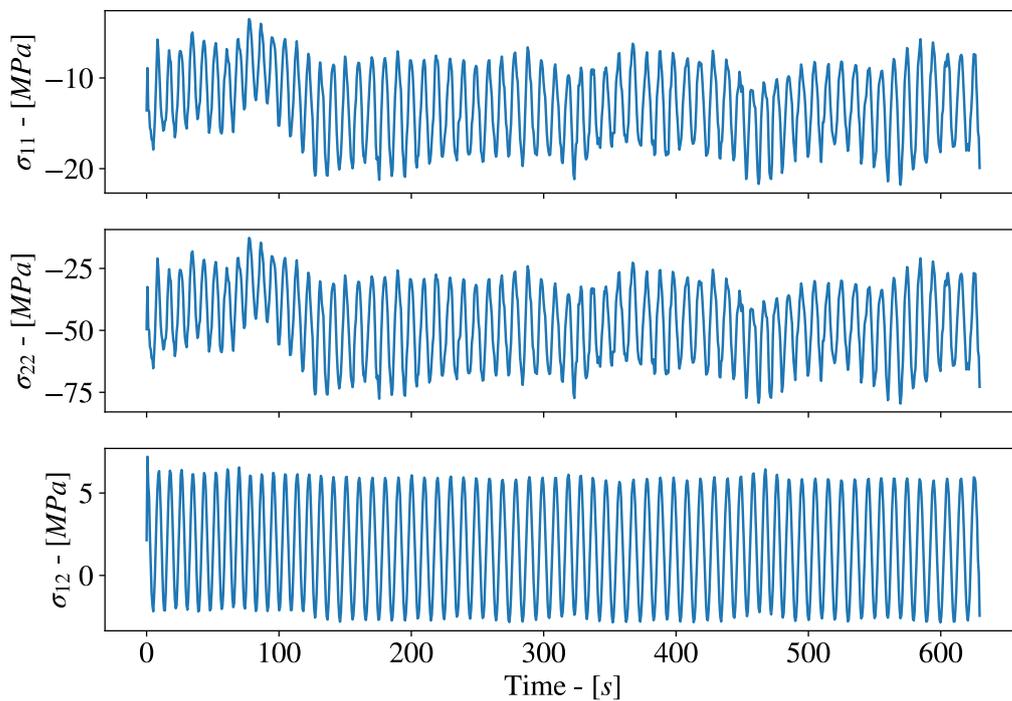


Figure 5.7: Stress history at node 606, second ply, inferior position at wind speed 11m/s using LAM 4NO.

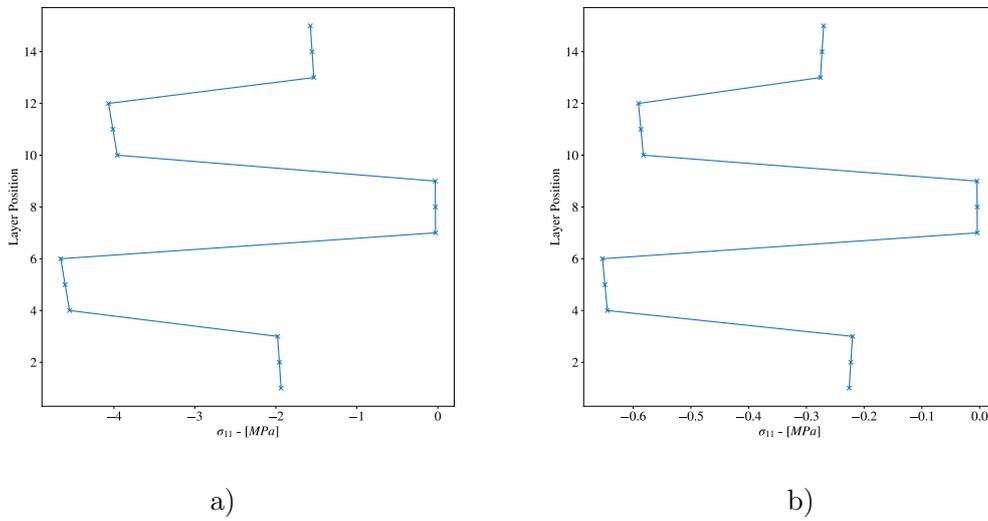


Figure 5.8: Stress distribution through laminate thickness at node 606 using LAM PD for a) wind speed  $11m/s$  and b) wind speed  $16m/s$ .

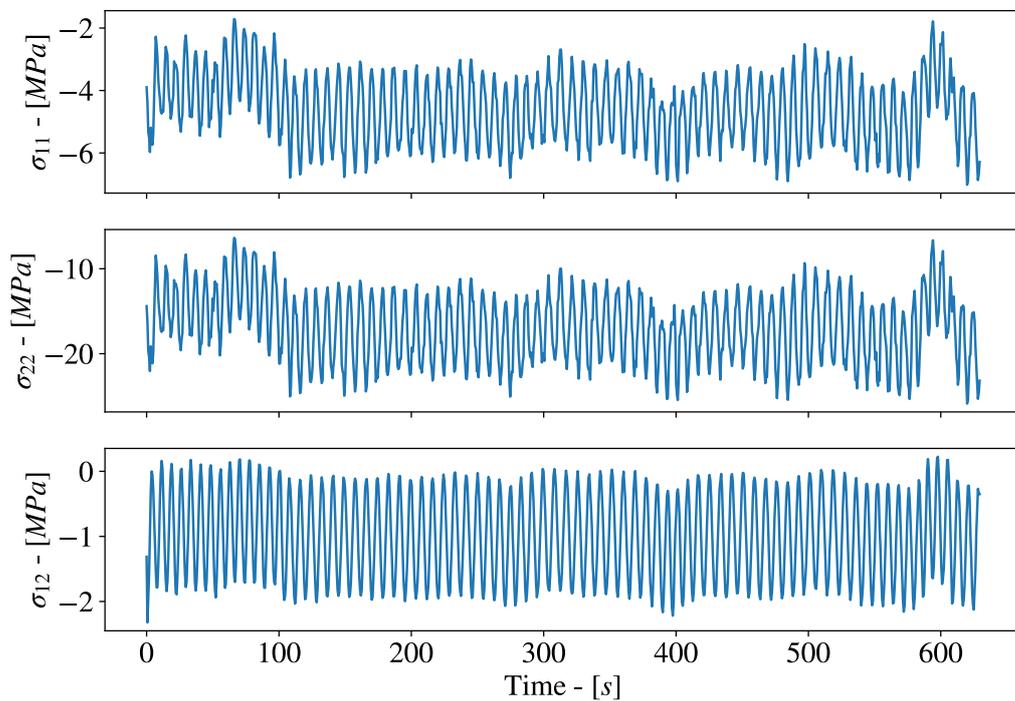


Figure 5.9: Stress history at node 606, second ply, inferior position at wind speed  $11m/s$  using LAM PD.

damage values depending on the approach selected.

The resulting stress can be extracted at each node and each position of every ply (3 positions per ply), these results in a total of more than 450,000 output values per time instant. In this case, to reduce the size of the output domain is only analyzed the inferior position of the two first plies of 4 nodes per cross-section. The forces are applied using LAM 4NO, as these are the position where stress concentration can be found by applying the loads directly.

### 5.3 Multi-axial fatigue criteria for composite materials

The stress time history obtained in section 5.2 shows that the blade bears non-proportional multi-axial complex stress states of variable amplitude and mean, which is similar to the same result obtained by [Hu et al., 2016b]. Stress history of longitudinal normal stress  $\sigma_{11}$ , transverse normal stress  $\sigma_{22}$  and shear stress  $\sigma_{12}$  of blade root node 606, are shown in figure 5.5. Nussbaumer et al. [Nussbaumer et al., 2012] explained the differences between proportional and non-proportional cyclic loads. Proportional stresses usually result from a single loading, varying with time, acting on the structure. All multi-axial stresses vary proportional to the loading and proportionally to each other, which is also true concerning their ranges and mean values. On the other hand, non-proportional stresses usually result from the action of at least two loadings that vary non-proportionally with time in a different manner. Also, They can result from one constant combined with one moving load. The non-proportional stresses in wind turbine blades are because the wind load, gravity load, and centrifugal load vary non-proportionally with time [Hu et al., 2016a].

To count cycles for the non-proportional multi-axial complex stress states, the realization of 900 seconds using the FAST simulation is executed. After neglecting the first 200 seconds, the next 600 seconds (10 min of an effective simulation) are used to count peaks and valley for each stress component. A range-mean counting method described by ASTM Committee [ASTM, 2003] is applied to count all the half cycles, allowing a cycle-by-cycle fatigue analysis. The stress path from one stress state to the next following stress state is counted as a one-half cycle. As an example, in figure 5.10 is shown a time history of loads units to explain how the range-mean counting works. The most common method to count cycles for wind turbine fatigue is Rainflow, however this approach breaks the sequence of cycles creating a new signal with equivalent damage. In this study is keep the original sequence of cycles to determine the damage suffer by the wind turbine at each moment and the range-mean approach could solve this problem.

This method defines a range as the difference between two successive reversals. The range is positive when a valley is followed by a peak (i.e., A-B) and negative when a valley is followed by a valley (i.e., D-E), counting both positive and negatives ranges. If only positive or negative ranges are counted, then each is counted as one cycle. If both positive and negative are counted, then each is counted as a one-half cycle. Also, considering their mean value at every range, the results are presented in a matrix (see table 5.2).

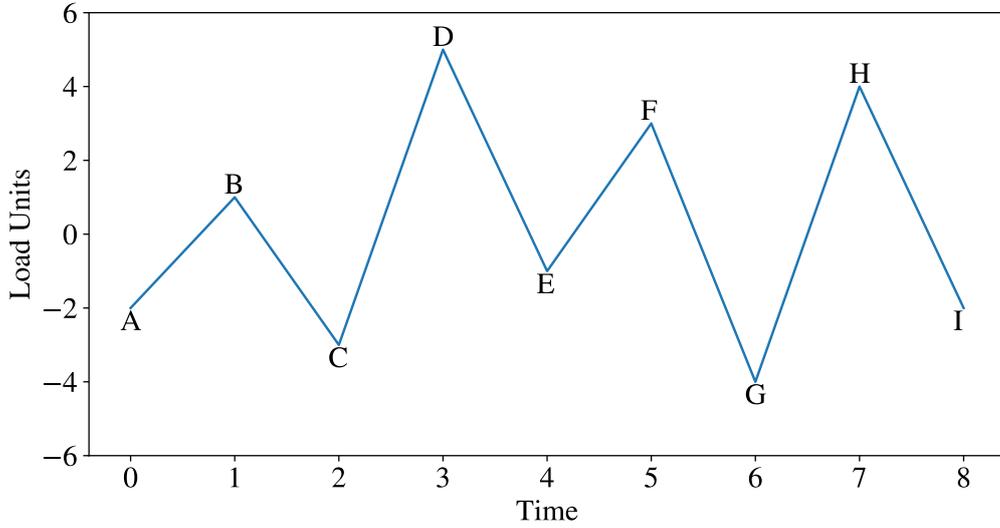


Figure 5.10: Range-mean cycle counting example.

Table 5.2: Example Range-Mean Counting

Range Units	Mean Units								
	-2	-1.5	-1	-0.5	0	0.5	1	1.5	2
10	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-
8	-	-	-	-	0.5	-	0.5	-	-
7	-	-	-	0.5	-	-	-	-	-
6	-	-	-	-	-	-	0.5	-	0.5
5	-	-	-	-	-	-	-	-	-
4	-	-	0.5	-	-	-	0.5	-	-
3	-	-	-	0.5	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-	-	-

According to the Tsai-Hill criterion [Jones, 1998], a multi-axial fatigue damage index [Liu and Mahadevan, 2005] caused in a half cycle under a stress amplitude level  $\sigma_{11}^i$ ,  $\sigma_{22}^i$  and  $\sigma_{12}^i$  is computed as:

$$D^i = 0.5 \sqrt{\frac{1}{(N_{11}^i)^2} + \frac{1}{(N_{22}^i)^2} + \frac{1}{(N_{12}^i)^2} + \frac{1}{N_{11}^i N_{22}^i}} \quad (5.8)$$

where  $N_{11}^i$ ,  $N_{22}^i$  and  $N_{12}^i$  are the number of allowable cycles under pure stress components  $\sigma_{11}^i$ ,  $\sigma_{22}^i$  and  $\sigma_{12}^i$ , respectively. The coefficient 0.5 indicates the half cycle. This procedure to calculate fatigue damage is the same as used by Hu et al. [Hu et al., 2016a].

### 5.3.1 S-N curves model

The S-N curves or Wöhler diagram [Wöhler, 1870] are used to represent the fatigue properties of structures in terms of a damage accumulation law. In this representation, the number of cycles to failure is given as a function of the stress cycle's amplitude. In its typical form, the S-N fatigue life model [E73910, 2015] is provided by the relation:

$$N_{ij} = a_{ij} S_{ij}^{b_{ij}} \quad (5.9)$$

where  $S_{ij}$  is the stress amplitude,  $N_{ij}$  is the number of cycles to failure,  $a_{ij}$  and  $b_{ij}$  are fatigue strength coefficients corresponding to a stress ratio  $R$ . Taking logarithm of both sides and rearranging terms, is obtained

$$\log(N_{ij}) = \log(a_{ij}) + b_{ij} \log(S_{ij}) \quad (5.10)$$

the above equation describes a linear relation between  $\log(N_{ij})$  and  $\log(S_{ij})$ , with  $\log(a_{ij})$  as the intercept and  $b_{ij}$  as the slope. This relationship describes very well the fatigue life of materials over a wide range of stress amplitudes, except when the stress amplitude approaches the static strength of the specimen, or the fatigue threshold level (the endurance limit) [Dimitrov, 2013]. In this study is only considered stress amplitudes that produce cycles to failure of  $N = 1e2$ . The stress ratio  $R$  is equal to the ratio of the minimum cyclic stress to the maximum cyclic.

$$R = \frac{\sigma_{min}}{\sigma_{max}} \quad (5.11)$$

ASTM standards [E73910, 2015] suggest to treat statistically the fatigue strength coefficient used to construct the S-N curve to represent a 95% survival probability with a confidence interval of 95%. The S-N curve of material QQ1 in the longitudinal direction at a stress ratio of  $-0.5$  is shown in figure 5.11, retaining the lower bound confidence interval to calculate the cycles to failure at any stress amplitude for all composite materials.

However, S-N curves only capture the failure behavior at a constant ratio  $R$ , which is used for constant amplitude stress. In this case, the stress amplitude is variable through time, meaning that failure cycles at different ratios must be estimated. In this case, is used a constant life diagrams [Vassilopoulos and Nijssen, 2010].

### 5.3.2 Constant Life Diagrams - CLD

The constant life diagram (CLD) is a model used to predict the number of cycles to failure as a function of cycle amplitude  $S$  and mean stress values by interpolating between S-N curves obtained from constant amplitude with different load ratios. CLD for composite materials

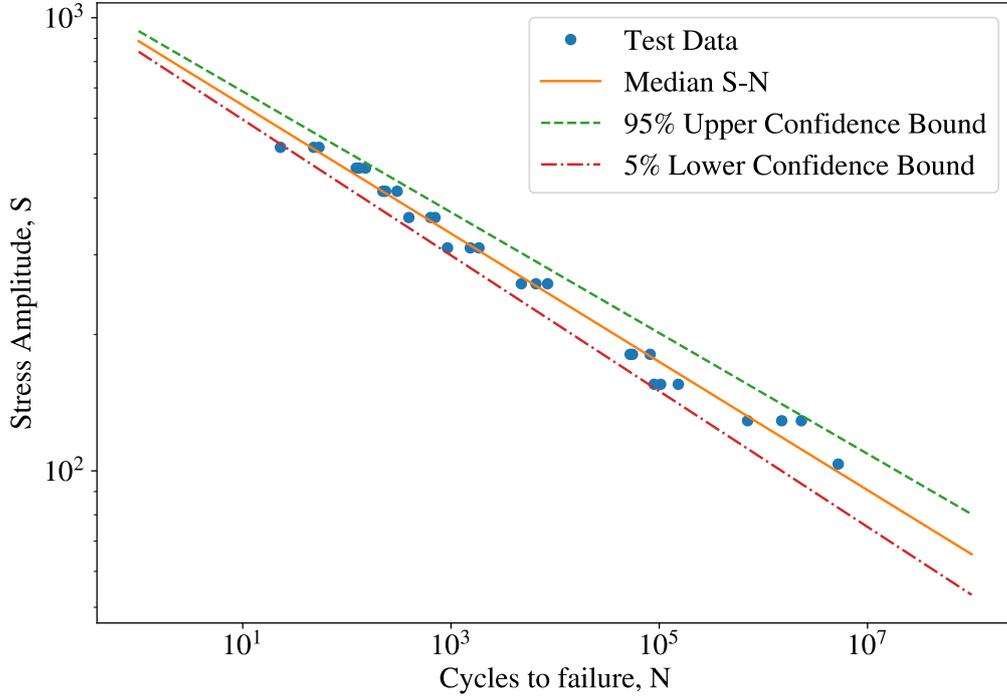


Figure 5.11: 95% confidence band for the S-N curve of QQ1 in longitudinal direction with  $R = -0.5$ .

QQ1 and P2B for longitudinal and transversal direction are constructed using fatigue test data extracted from SNL/MSU/DOE composite material database [Mandell and Samborsky, 2016]. This test include maximum stress, minimum stress and cycles to failure for each material tested at ratios  $R = [-2, -1, -0.5, 0.1, 0.5, 0.7, 10]$ . The amplitude stress and mean stress are calculated using the equations:

$$S_{ij} = \frac{\sigma_{max} - \sigma_{min}}{2} \quad (5.12)$$

$$\sigma_{ij}^m = \frac{\sigma_{max} + \sigma_{min}}{2} \quad (5.13)$$

The constant life diagram is constructed from the 95% lower bound S-N curves. Vasilopoulos [Vasilopoulos et al., 2010] studied the influence of the CLD formulation and concluded after analyzing different approaches that piece-wise linear is the most accurate of the approaches when using a reasonable number of S-N curves ( $> 3$ ). The piece-wise linear CLDs of QQ1 and P2B are illustrated in figures 5.12, 5.13, 5.14 and 5.15. The range of a number of cycles to failure goes from  $10^2$  to  $10^8$ .

To generate the CLD for each composite material QQ1 and P2B, the module `scipy optimize` [Jones et al., 01] estimates the parameters of a function given the data using a non-linear least squares to fit the function. This is used to fit the S-N curves for each tested ratio at longitudinal and transverse direction for both materials. Subsequently is created a piece-wise linear function for both materials in both directions using bi-dimensional interpolation tool from `scipy` [Jones et al., 01].

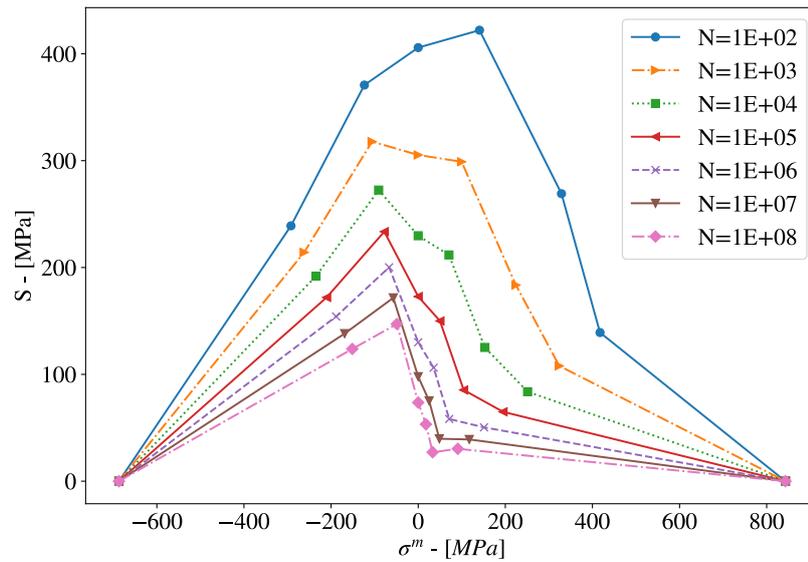


Figure 5.12: Constant life diagram of QQ1 in longitudinal direction using lower bound 95% S-N curves.

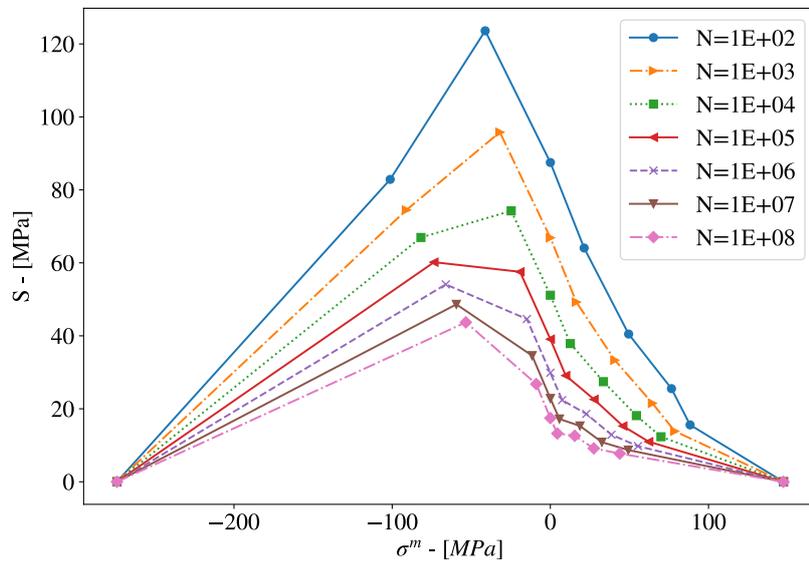


Figure 5.13: Constant life diagram of QQ1 in transversal direction using lower bound 95% S-N curves.

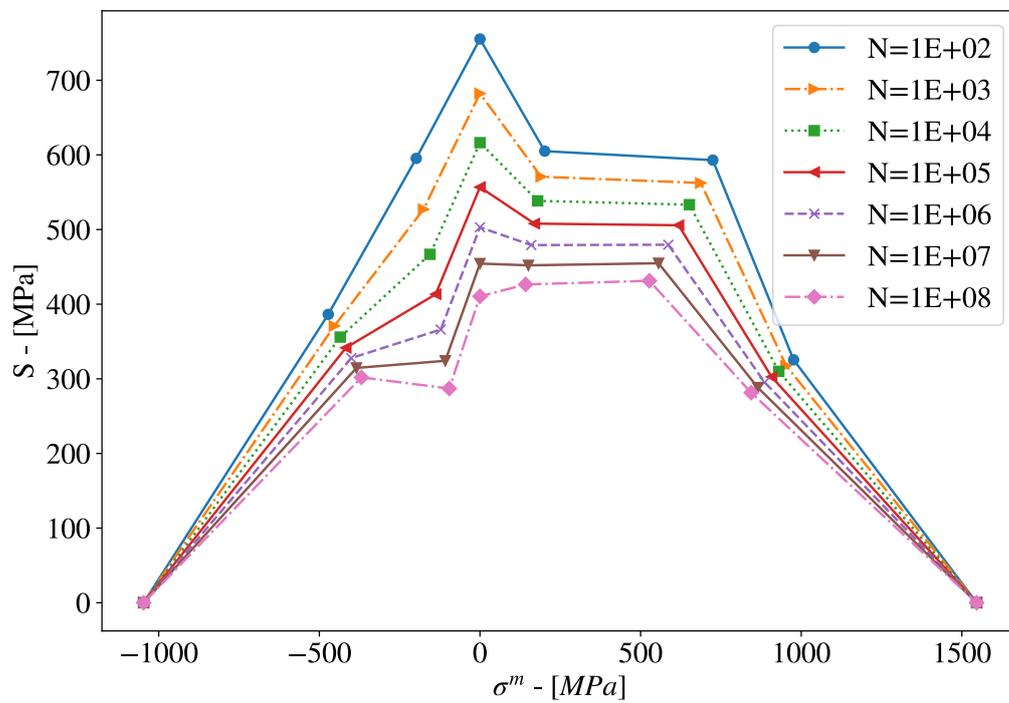


Figure 5.14: Constant life diagram of P2B in longitudinal direction using lower bound 95% S-N curves.

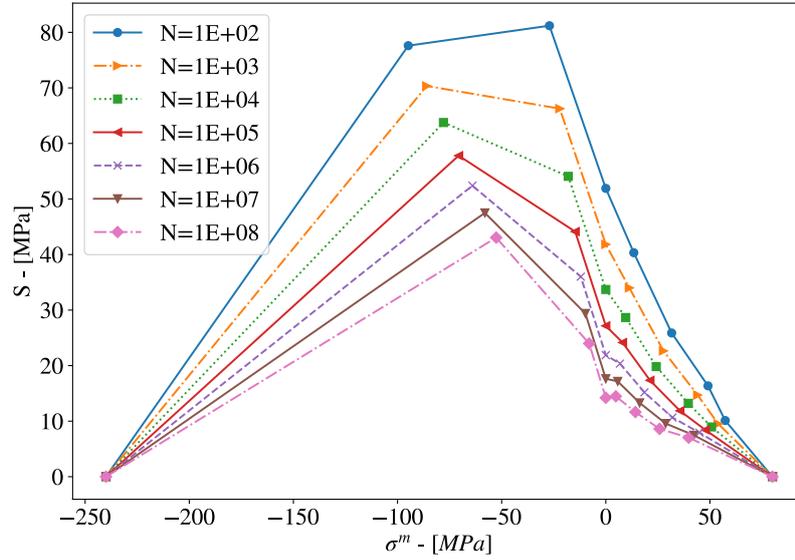


Figure 5.15: Constant life diagram of P2B in transversal direction using lower bound 95% S-N curves.

For CLD QQ1 in the longitudinal direction (figure 5.12), the damage done by a cycle with some amplitude is susceptible to the mean stress at reversed loading R-values. Tension is much more damaging than compression at high cycles, much less so at low cycles. The transverse QQ1 CLD (figure 5.13) is distorted toward higher strength and fatigue resistance in compression, as is typical for the transverse direction of composites. These results may be used to predict matrix cracking in blades. CLD P2B in the longitudinal direction (figure 5.14) reflects a similar ratio of compression to tensile strength compared with fiberglass QQ1 but improves fatigue resistance at all R values. The CLD P2B in transversal direction (figure 5.15) is similar in shape to that for fiberglass material QQ1 transversal, also tested in the transverse direction. As noted earlier, QQ1T has higher strength values due to the different contents of plies in various directions and the higher transverse modulus for glass versus carbon [Mandell et al., 2010].

To calculate the number of failure cycles in the shear direction is needed, the CLD in shear direction for QQ1 and P2B. However, fatigue test data is not available in the composite material database [Mandell et al., 2010]. To estimate missing test data for shear fatigue strength of composite materials, Philippidis and Vassilopoulos [Philippidis and Vassilopoulos, 2002] shows that shear fatigue strength values are calculated as  $1/2.2$  of the fatigue strength of a flat coupon cut off-axis at  $45^\circ$  and loaded uniaxially, adequately fit most of the experimental data. Liu and Mahadevan [Liu and Mahadevan, 2005] considered the average value of the S-N curve slopes corresponding to the longitudinal and transverse directions tests as the slope of the S-N shear fatigue curve. Hu et al. [Hu et al., 2016a] approximated the fatigue shear strength by dividing the average fatigue strength in the longitudinal and transverse direction by a constant. The calculated constants for QQ1 and P2B are 3.4628 and 5.3157, respectively. This

last approximation is used to estimate the shear CLD of QQ1 and P2B, illustrates in figure 5.16 and figure 5.17.

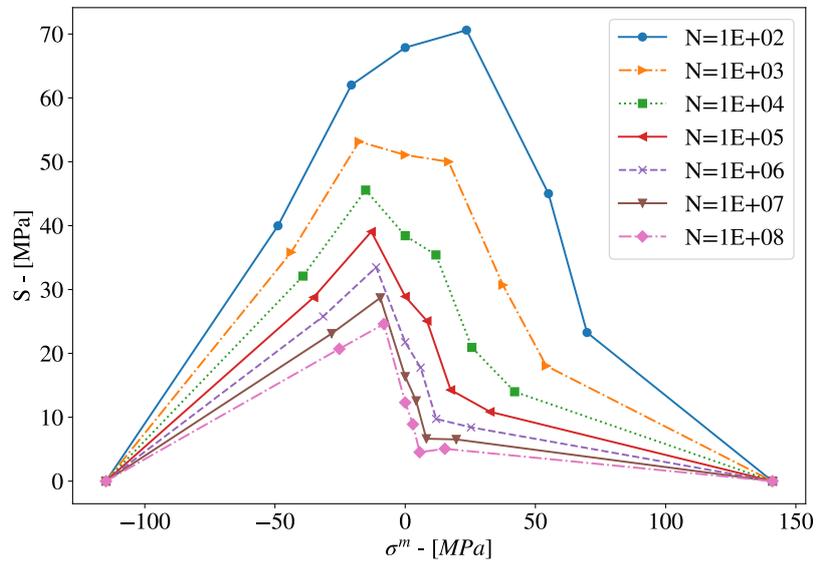


Figure 5.16: Constant life diagram of QQ1 in shear direction using lower bound 95% S-N curves.

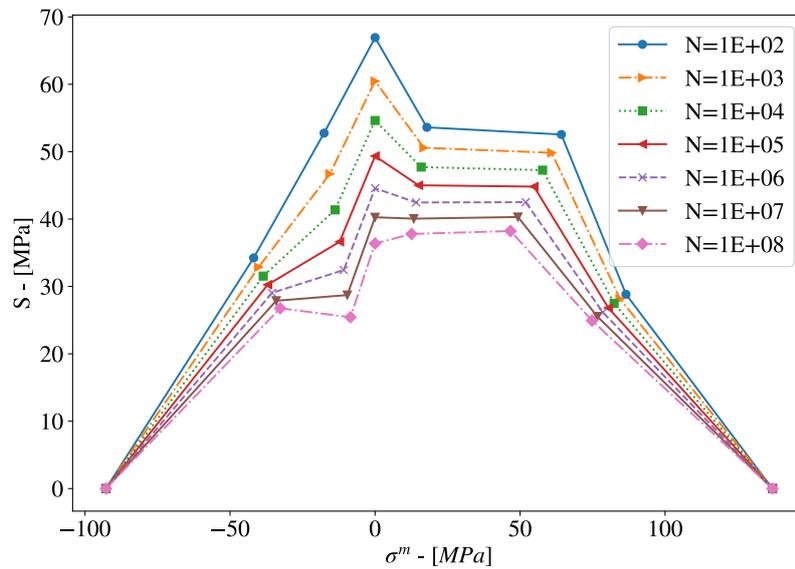


Figure 5.17: Constant life diagram of P2B in shear direction using lower bound 95% S-N curves.

### 5.3.3 Fatigue damage accumulation

To continue the fatigue analysis is necessary to select the estimation of the fatigue damage accumulation, which is non-linear under variable amplitude non-proportional multi-axial loading. A comparative study of non-linear damage accumulation in stochastic fatigue FRP (fiber-reinforced plastics) laminates by Sarkani [Sarkani et al., 2001] presents different non-linear methods versus a linear method and all of them predicted fatigue lives that were comparable to those predicted by the linear damage accumulation rule. Miner's rule [Miner, 1945], the most used method to predict fatigue damage of wind turbine blades, assumes a linear damage accumulation [Guideline and Lloyd, 2010]. The fatigue damage in 10 minutes simulation is calculated as

$$D_{10} = \sum_{i=1}^n D^i = 0.5 \sum_{i=1}^n \sqrt{\frac{1}{(N_{11}^i)^2} + \frac{1}{(N_{22}^i)^2} + \frac{1}{(N_{12}^i)^2} + \frac{1}{N_{11}^i N_{22}^i}} \quad (5.14)$$

The study uses this linear model for damage accumulation.

## 5.4 DNN: 10-minutes damage

### 5.4.1 Input domain

The inputs parameters to generate the environmental conditions are the 10 minutes mean wind speed, 10 minutes turbulence intensity, and 10 minutes mean shear exponent. Only one simulation per wind speed between  $(4, 25)m/s$  was performed, resulting in a total of 21 stress time history used to calculate the damage caused in a half cycle  $D^i$  per wind speed  $WS$ , for a unique value of  $WS$  multiple values of  $D^i$  are correlated as shown in figure 5.18(a). As can be seen, if a neural network is used to predict the damage, it only predicts a mean value per wind speed, resulting in a bad prediction of  $D^i$ . In order to predict the 10 min damage  $D_{10}$  is added as the second input parameter, the empirical cumulative distribution function (ECDF) of  $D^i$  per each wind speed ordered as a increasing function, using python package statsmodel [Seabold and Perktold, 2010]. Adding this second input parameter, the damage prediction for  $D_{10}$  is estimated by selecting the desired wind speed and generating a uniform distribution function  $[0, 1]$ , of size equal to the number of half-cycles in a 10 min stress history (i.e., count of ranges of stress).

The ECDF is a step function with a step size equal to  $1/n_{obs}$  where  $n_{obs}$  is the number of observed data points. Its value at any specified value of the measured variable is the fraction of observations of the measured variable that are less than or equal to the specified value.

Let  $X_1 \dots X_n$  be a random sample from a distribution  $F$  on the real line. The empirical cumulative distribution function is defined by Van [Van der Vaart, 2000] as:

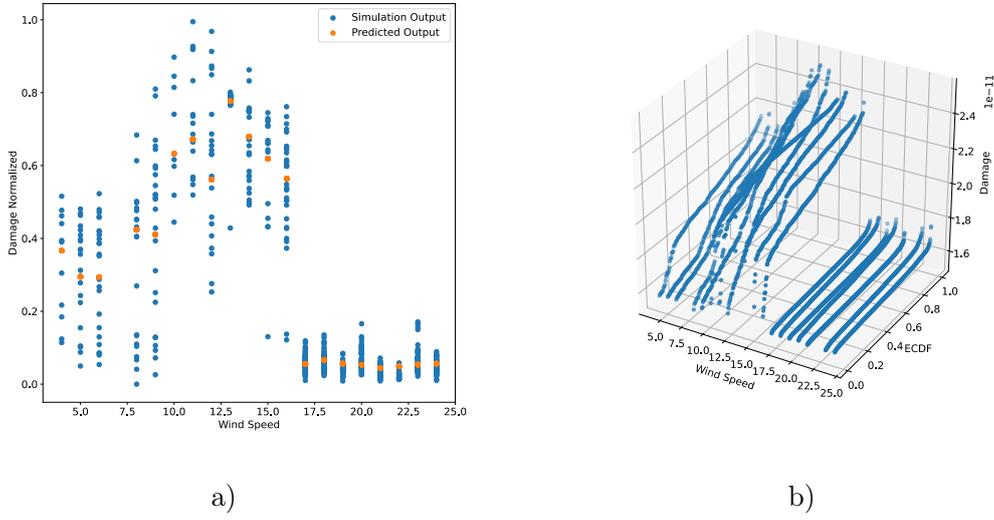


Figure 5.18: a) Damage distribution  $D^i$  per wind speed and b) ECDF per wind speed of Damage using LAM 4NO at Node 506.

$$\bar{F}_n(t) = \frac{1}{n} \sum_{i=1}^n 1\{X_i \leq t\} \quad (5.15)$$

where  $1\{X_i \leq t\}$  is the indicator of the event  $\{X_i \leq t\}$  for a fixed  $t$  is a Bernoulli random variable with parameter  $p = F(t)$ ; hence  $n\bar{F}_n(t)$  is a binomial random variable with mean  $nF(t)$  and variance  $nF(t)(1 - F(t))$ . This implies that  $\bar{F}_n(t)$  is an unbiased estimator for  $F(t)$ .

The use of ECDF in this problem is motivated from the idea of the inverse transform method [Devroye, 2006], the theorem implies that, If  $F$  is a continuous distribution function on  $\mathbb{R}$  with inverse  $F^{-1}$  defined by

$$F^{-1}(u) = \inf x : F(x) = u, 0 < u < 1$$

If  $U$  is a uniform  $[0, 1]$  random variable, then  $F^{-1}(U)$  has distribution function  $F$ . Also, if  $X$  has distribution function  $F$ , then  $F(X)$  is uniformly distributed on  $[0, 1]$ . This theorem can be used to generate random variables with an arbitrary continuous distribution function  $F$  provided that  $F^{-1}$  is explicitly known.

As shown in figure 5.18 sorting the damage from lowest to highest, each value of  $D_i$  per wind speed has a corresponding value of  $\bar{F}$ , which gives the network a better representation of the behavior of the damage.

### 5.4.2 DNN description

For training the DNN, all inputs and output parameters were scaled between (0,1) to normalize all parameters magnitude. The literature suggests only to normalize the input parameters if their scales have huge magnitude differences that can lead to retard the convergence of the optimization algorithm for the hyper-parameters [Géron, 2019]. However, in this case, the output is normalized because it is close to 0 (i.e., magnitude 1e-9), and the DNN will always predict 0. After normalization, the data is shuffled, taking 90% of the data to train the model and the left 10% for validating the model prediction. The DNN is a sequential, fully connected network having a total of 6 hidden layers; each of them has 296 neurons. Also, all layers have a rectified linear unit (Relu) activation function and a learning rate of 0.0012. For this regression problem, the weight and bias for the DNN are optimized using Adam optimizer to minimize the mean squared error of the difference between prediction and real value.

These hyper-parameters are the result of a tuning optimization explained next.

### 5.4.3 DNN damage: hyperparameters optimization

To begin the optimization problem, the hyper-parameters space needs to be defined. The DNN topology is a sequential, fully connected network, and the hyper-parameters that will be optimized are presented in table 5.3. Only these hyper-parameters are selected to be changed by the optimization problem. It could be added to the number of nodes or the activation function different for each layer. However, these add more dimensions to our problem, and it would need more calculation time to explore/exploit the whole hyper-parameter space.

Table 5.3: Hyper-parameter searching space for DNN Damage.

Hyper-parameter	Range	Type
Learning rate	[1e-2,1e-5]	Real
No. hidden layers	[4, 10]	Integer
No. of nodes per layer	[1,1000]	Real
Activation function per layer	[ReLU, LeakyReLU, Sigmoid]	Categorical

After defining our DNN model, the optimization algorithm search for the parameters that minimize the validation mean squared error of the DNN for 30 calls. At each call, the DNN trains for 100 epochs. When the best result is founded, the algorithm is relaunched, but this time the initial hyper-parameters are the best result from the previous analysis. If there is no change in the validation mean squared error, these hyper-parameters are selected to train the DNN. They ensure the lowest error in the validation set in less training iterations.

In figure 5.19 (a) is shown the first run of the hyper-parameters optimization and in figure 5.19 (b) the second run to ensure that no other minimum was found. As a result, it is found that the best learning rate, number of hidden layers, number of nodes per layer, and activation function are [0.001, 6, 296, *ReLU*], respectively.

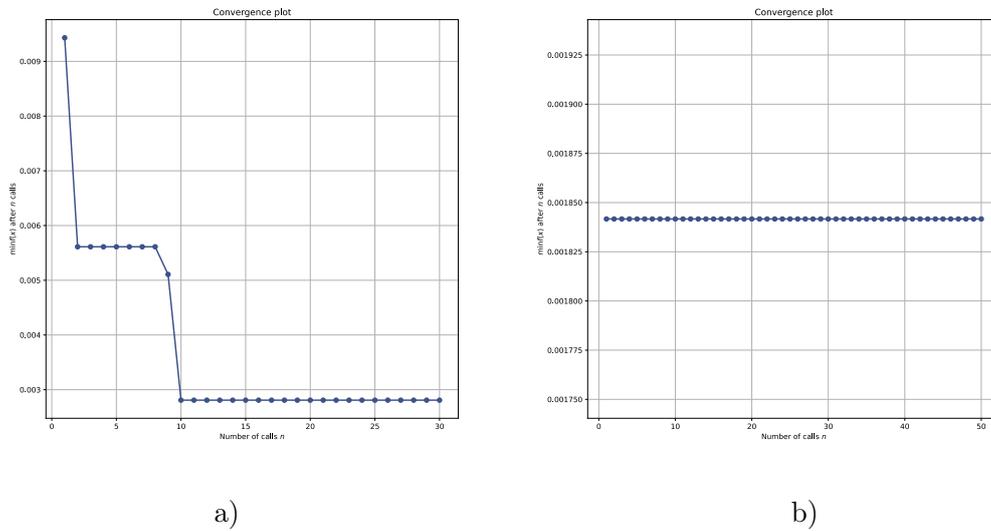


Figure 5.19: Convergence curve of hyper-parameter optimization a) first run b) second run starting from optimum of first run.

Training only 100 epochs per call is not enough for the DNN to learn the pattern in the training data; the results founded gives the configuration that has the minimum error at this amount of epochs. Using at least 1000 epochs per call to train the DNN, a lower RMSE could be achieved, and these results could change the hyper-parameters values.

Subsequently, the DNN is built with the optimized hyper-parameters and trained. The stopping criteria were a maximum iteration of 3000 epochs or an absolute error of validation mean squared error between two subsequent epochs.

#### 5.4.4 DNN: 10-Minutes damage prediction

After finding the optimized hyper-parameters for the DNN, the training phase starts, then the neural networks are validated. Ten neural networks per LAM approach are trained to predict the damage of all levels of composite material QQ1 and P2B at ten nodes where the LAM 4NO applies the maximum load (i.e.,  $F_i^C, i = x, y, z$ ), where may occur the maximum stress concentration. At each node are found four layers of composite material (excluding Balsa wood, not analyzed) and per each layer three levels (INF, MID, SUP), resulting in a total of 12 damage output per neural network. As shown in table 5.4, DNN for both LAMs presents an excellent fit to the simulated damage with an MSE between  $1.38E - 4$  to  $6.10E - 7$ .

This prediction is made by training the DNNs using only one aero-elastic simulation that generates a time history response. This history response depends on a random signal generation. For this simulation, the random seed was fixed, meaning that for the same environmental conditions and a different random seed, the wind field time history will result differently.

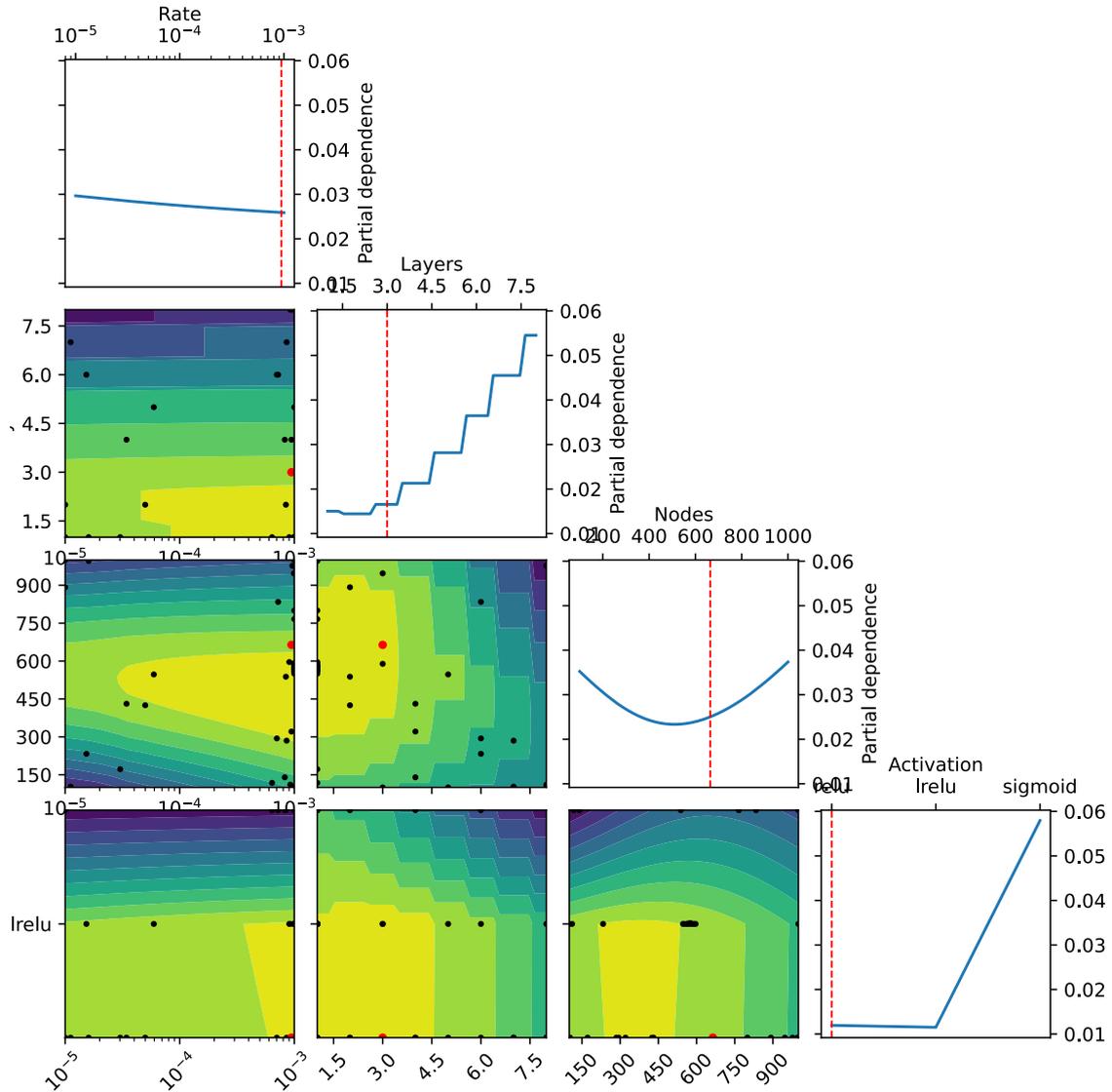


Figure 5.20: Hyper-parameters space search points for DNN 10 Minute Damage.

To test these DNNs, a new 3D wind field is generated using a different seed number selected randomly and used to run the aero-elastic simulation to generate the time history of the wind turbine blade’s behavior. Subsequently transferred to the blade shell FEM using only LAM 4NO and calculated the 10 minutes damage at the blade nodes where load  $F_i^C$  is applied per each wind speed. Only LAM 4NO is used as a test method because it is less time consuming than LAM PD, about six times faster.

In table 5.5 is presented the average error of the 10 minutes damage prediction of DNN, to calculate this error is used the following equation:

Table 5.4: MSE of DNNs Damage using LAM 4NO and LAM PD.

Z Coordinate – [m]	Node	LAM 4NO MSE	LAM PD MSE
2,8	N606	1,38E-04	2,12E-04
4,8	N603	3,60E-05	3,49E-04
11,658	N595	5,65E-05	4,32E-04
19,267	N586	6,91E-05	3,50E-04
24,156	N580	2,25E-04	2,00E-04
35,519	N567	1,92E-05	8,39E-06
51,257	N549	7,95E-05	6,10E-07
66,516	N531	6,76E-05	3,72E-05
79,034	N516	1,69E-04	2,61E-05
89,166	N506	7,03E-05	1,12E-04

$$e_{avg} = \frac{1}{M} \sum_{i=1}^M \frac{abs(D_{10}^{NS}(i) - \hat{D}_{10}(i))}{D_{10}^{NS}(i)} * 100\% \quad (5.16)$$

Where  $\hat{D}_{10}$  is the 10-minute damage predicted by the DNN,  $D_{10}^{NS}$  is the damage calculated using LAM 4NO 2D shell FEM, and  $i$  is the number of nodes where the damage is extracted on the blade, in this case, is extracted the damage at the inferior level of the corresponding layer. These  $D_{10}^{NS}$  is used to test the response of the trained DNN to predict  $\hat{D}_{10}$ . In this case is neglected the prediction on nodes 595 and 586 due to the high-stress concentration presented, inducing a significant amount of error, and only are considered all other eight nodes, where the stress level is homogeneous (i.e.,  $M = 8$ ). The maximum relative percent error was found at wind speed 22  $m/s$  for P2B and 12 $m/s$  for QQ1 composite material. The error is less than 14%, meaning that our DNN can accurately predict the accumulated damage at 10-minutes for different time history wind speed by using one time history to train the network.

By only training these networks with one time history, they can generate the half-cycle damage of different time history, having the same environmental conditions. These results are obtained by using DNN LAM 4NO. It assumes that for LAM PD, the results will be similar because there is no stress concentration. The damage distribution is much better than using LAM 4NO, which will result in a better prediction of the 10 minutes fatigue damage.

Another test performed with this network was to predict missing wind speed in the training set. In this case the network is trained without a given wind speed and then is used to predict the damage at this exact wind speed. For this test is used the LAM PD at first and second composite layer (P2B and QQ1, respectively) at the inferior level to predict the damage at the blade root node  $N606$ . Results are presented in table 5.6 where is illustrated, the simulated damage at the missing or skipped wind speed, the damage predicted by the network and the relative error in percentage per each wind speed. To obtain each value the network is

Table 5.5: Average relative error for LAM 4NO 10 minutes damage prediction testing using a different seed number for generation of the 3D wind.

Wind Speed - [ $m/s$ ]	10 Minutes Damage Error	
	P2B	QQ1
4	4.746%	5.205%
5	6.868%	4.507%
6	6.735%	3.320%
7	3.621%	3.156%
8	5.466%	6.372%
9	9.128%	6.635%
10	1.642%	6.058%
11	3.005%	8.042%
12	7.021%	<b>8.783%</b>
13	2.065%	7.119%
14	2.020%	6.148%
15	4.684%	7.415%
16	2.624%	4.274%
17	4.094%	1.486%
18	3.753%	2.009%
19	3.972%	3.024%
20	4.676%	3.596%
21	5.510%	0.890%
22	<b>13.199%</b>	8.568%
23	5.044%	1.204%
24	4.406%	0.964%

re-trained without the wind speed analyzed. The layer P2B shows the greater relative error at missing wind speed  $12m/s$  which is closed to the rated wind speed, and all other wind speed has an error lower than 5%. The layer QQ1 shows the greater error at missing wind speeds 16 and  $17m/s$ , this is due to the pitch control behavior that turns the blade between these wind speeds creating a change in the behavior of the blade, however for the others wind speed the error is lower than 10% in the worst case. These results shows that this methodology can also predict damage values in between wind speeds without being simulated, predicting this values with an acceptable relative error. In this case, the discontinuity problem is not tackled by the ADM-DNN approach developed in chapter 3 because the fatigue damage prediction is not a continuous function of the wind speed. But it is only evaluated at the simulated wind speed, meaning that the discontinuity does not affect the response given by the neural network.

Table 5.6: 10 minutes damage prediction for a missing wind speed using LAM PD at blade root node for P2B and QQ1 layers at inferior level.

Missing Wind Speed - [m/s]	P2B			QQ1		
	Damage simulated	Damage Predicted	Error - [%]	Damage simulated	Damage Predicted	Error - [%]
5	2.08368e-09	2.09732e-09	0.65436	4.22951e-07	4.36328e-07	3.16275
6	1.92903e-09	1.92261e-09	0.33294	4.15519e-07	4.05184e-07	2.48729
7	2.11990e-09	2.11008e-09	0.463124	4.63673e-07	4.61568e-07	0.45392
8	1.77269e-09	1.80708e-09	1.94036	3.74536e-07	3.92229e-07	4.72390
9	2.28704e-09	2.22766e-09	2.59631	4.58177e-07	4.43849e-07	3.12728
10	2.11294e-09	2.15022e-09	1.76401	4.14194e-07	4.40371e-07	6.32009
11	2.33139e-09	2.37313e-09	1.79065	4.64537e-07	4.48154e-07	3.52689
12	2.80975e-09	2.68793e-09	<b>4.33584</b>	5.10204e-07	4.89535e-07	4.05105
13	3.02754e-09	3.13410e-09	3.51975	4.82082e-07	5.10898e-07	5.97732
14	2.87157e-09	2.79734e-09	2.58521	4.58418e-07	4.49535e-07	1.93773
15	2.72282e-09	2.76014e-09	1.37074	4.29057e-07	4.24736e-07	1.00709
16	3.16152e-09	3.22820e-09	2.10891	5.17371e-07	8.15514e-07	<b>57.62636</b>
17	7.708697e-09	7.53175e-09	2.29536	2.95702e-06	1.76823e-06	<b>40.20226</b>
18	8.60202e-09	8.55422e-09	0.555669	3.16936e-06	3.45968e-06	9.16016
19	5.94937e-09	6.07155e-09	2.05369	2.32489e-06	2.17627e-06	6.39247
20	7.45447e-09	7.32073e-09	1.79410	2.71307e-06	2.85448e-06	5.21216
21	3.98562e-09	4.04078e-09	1.38397	1.54475e-06	1.5078e-06	2.39087
22	1.08506e-09	1.11895e-09	3.12348	4.19524e-07	4.10399e-07	2.17493
23	7.22244e-09	7.08789e-09	1.86292	2.72399e-06	2.76120e-06	1.36613

### 5.4.5 Blade damage distribution

To determine the effect of using different LAMs, the blade's damage distribution is extracted at ten nodes, one per each cross-section where the LAM 4NO applied the loads directly (see figure 5.21). These nodes are the ones where the forces  $F_i^C$  are applied because it has a greater amplitude according to the physical law distribution. As the load is applied directly to the node, it may introduce stress concentration. Also, as there are many layers of composite materials, LAM PD results are extracted only on the two first layers (P2B and QQ1, respectively) at their inferior level because these layers at this position presented the absolute maximum stress if compared with their symmetrical layer position. These layers are the two closest to the blade shell.

First, the resulting 10 minutes damage accumulation obtained using LAM 4NO in the first layer (P2B material) is illustrated in figure 5.22. The first thing to notice is that the damage along the blade Z-coordinate from blade root (i.e., section 1) to section 3 increases drastically, the high damage produced at section 3, node 595,  $Z = 11.658$  has a magnitude order of  $D_{10} = 1e - 6$ . After this position, the damage decreases to magnitudes  $D_{10} = 1e - 8$  for the rest of the wind turbine, making node at S3, the location with the highest probability of failing. As a function of the wind speed, for almost all P2B layer at each node, the damage starts low and increases proportionally as the wind speed increases to reach the rated wind speed. After this wind speed, the damage increases drastically for some nodes at (S1, S2, S3, S4, S5, S9). At the other locations, the damage increases. The damage distribution is illustrated separately in appendix A for a better comprehension of the damage distribution.

Results obtained at the second layer (QQ1 material) using LAM 4NO are shown in Figure 5.23. In this case, the maximum damage is located in node 586 of section 4  $Z = 19.267$  with

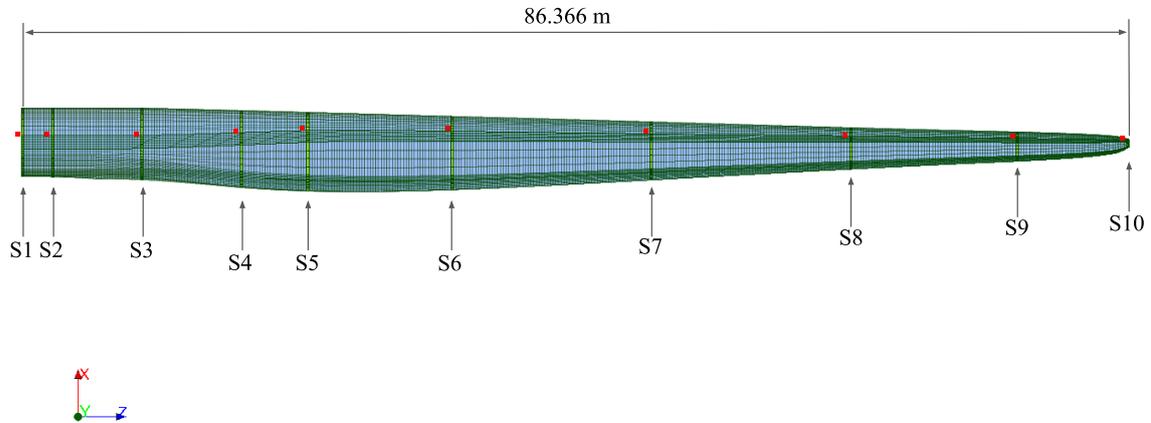


Figure 5.21: Blade nodes (red dots) where the damage is calculated using LAMs 4NO and PD.

a magnitude of  $D_{10} = 0.001$ , the highest of all layers, the second maximum damage is found in node 595 at section 3 with a damaging magnitude of  $D_{10} = 0.0001$ . All other damages have a magnitude of  $D_{10} = 1e - 6$  greater than the damage found in P2B. In other words, the layers with QQ1 composite material will fail faster than P2B using this stacking sequence and thicknesses. Regarding the damage as a function of the wind speed, it is noticed that for almost all nodes, the damage increases at wind speed greater than 16, having a similar value for other wind speed. Except for sections 3 and 4, where the maximum damage is found at  $12m/s$  (close to the wind turbine blade's rated wind speed).

Second, the resulting 10 minutes damage accumulation obtained using LAM PD in the first layer (P2B material) is presented in figure 5.24. Using this LAM, the maximum damage is located at node 586 of section 4 with a damaging magnitude of  $D_{10} = 1e - 6$ ; all other nodes have a lower damage with magnitudes between  $[1e - 9, 1e - 8]$ . The damage as a function of the wind speed has almost a constant behavior for all nodes at different wind speeds.

In the case of the second layer (QQ1 material) using LAM PD, results are shown in figure 5.25. The damage distribution along the blade z-coordinate has the same damage amplitude  $D_{10} = 1e - 6$  for all nodes, which is higher than the P2B layer for all sections. Regarding its distribution as a function of wind speed for all nodes, there is an increase in the damage after wind speed passes  $17m/s$  for all blade locations.

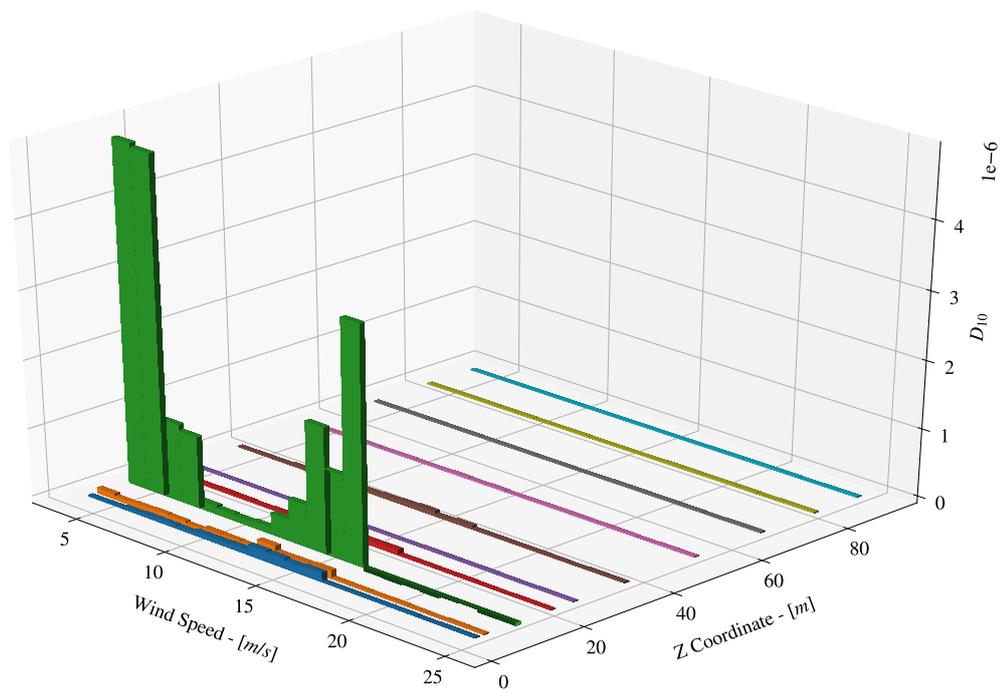


Figure 5.22: 10 minutes damage distribution along the blade Z coordinate as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO.

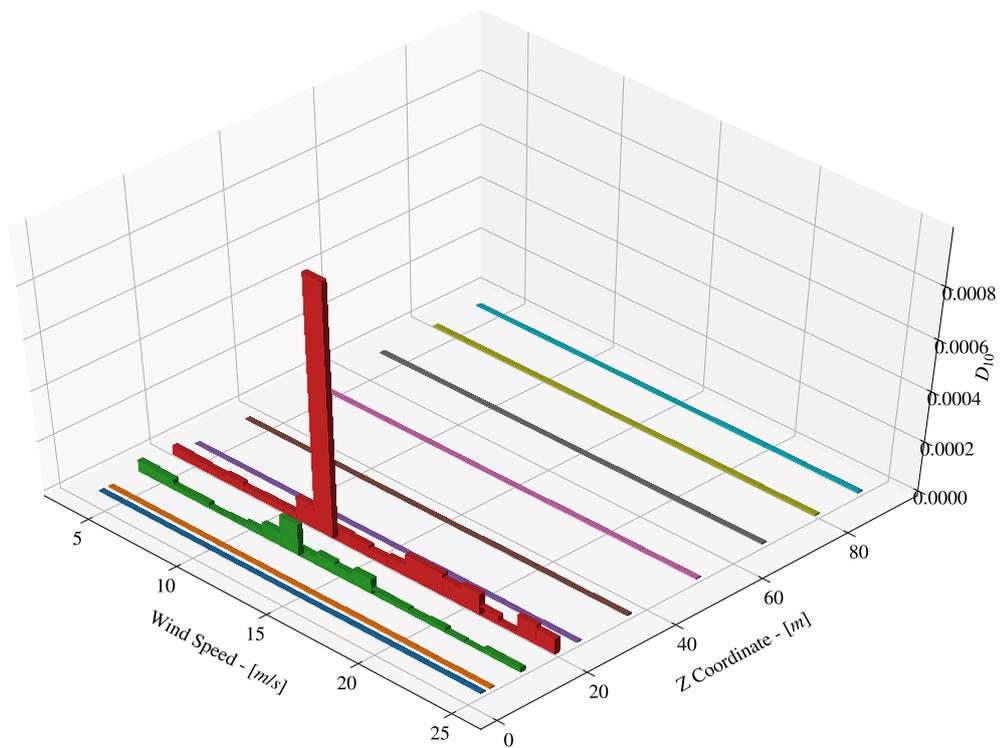


Figure 5.23: 10 minutes damage distribution along the blade Z coordinate as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO.

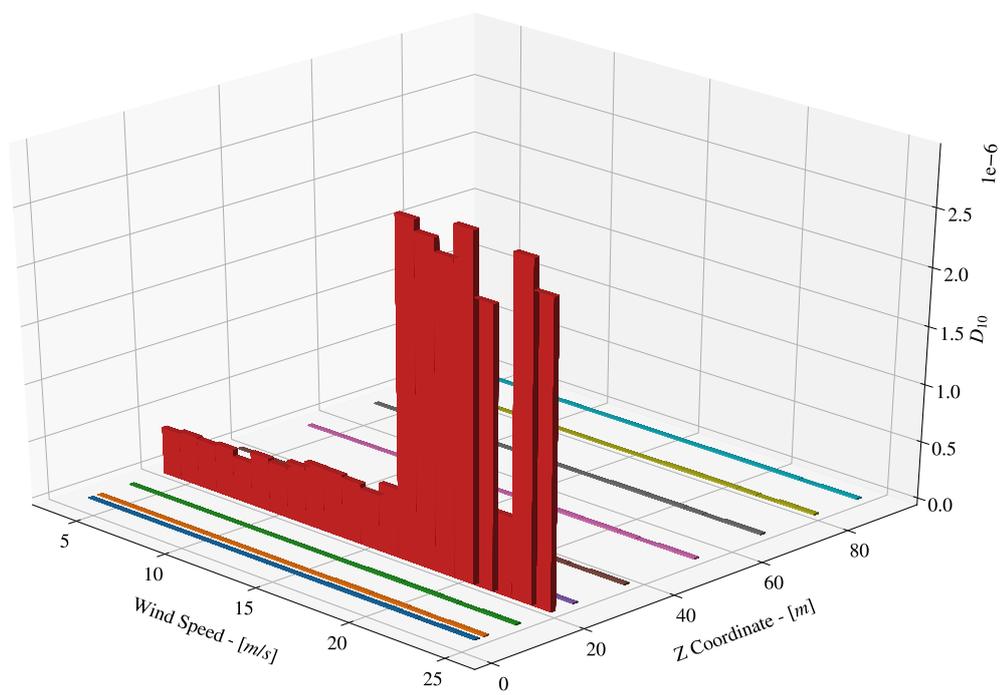


Figure 5.24: 10 minutes damage distribution along the blade Z coordinate as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

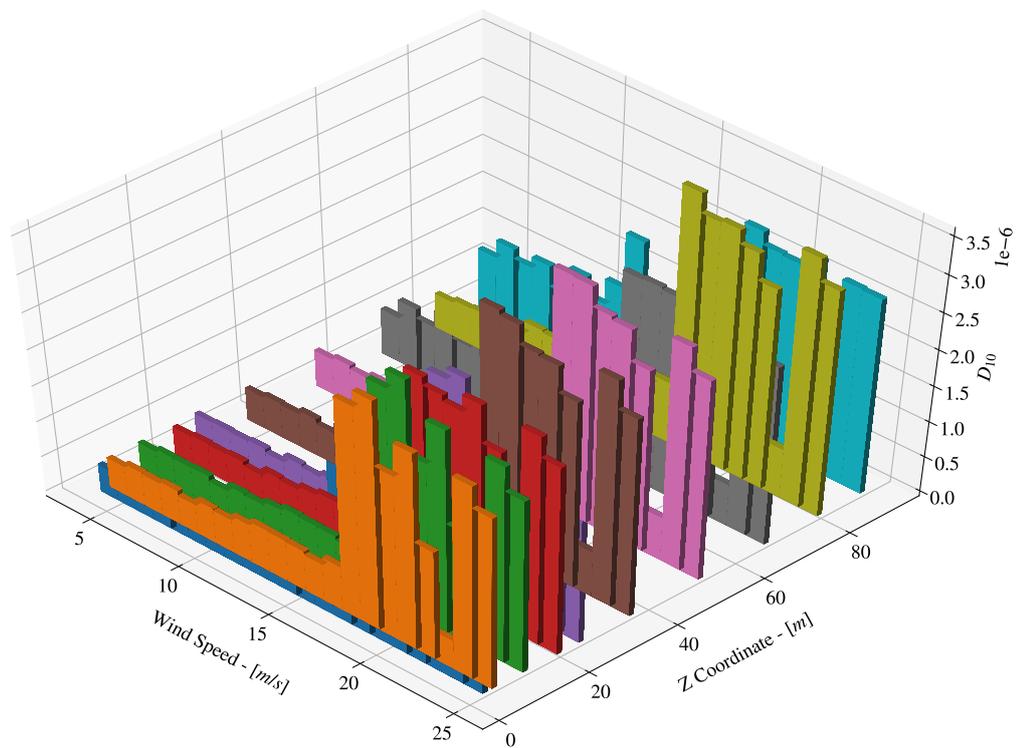


Figure 5.25: 10 minutes damage distribution along the blade Z coordinate as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

Comparing both methods LAM, The overall damage is higher when the loads are projected using LAM 4NO than when using LAM PD, which is similar to the results found by [Caous et al., 2018] when compared the methods using a Puck's damage criteria [Puck et al., 2002]. For layer P2B, LAM PD presents a more constant damage distribution as a function of the wind speed than LAM 4NO, which varies considerably with the wind speed. Also, LAM PD estimates the highest damage located in section 4, when LAM 4NO estimates the most significant damage in section 3. On the other hand, layer QQ1 using LAM 4NO presents a higher magnitude value in sections 3 and 4 than LAM PD, about 120 and 1000 times higher, which means that in these locations, LAM 4NO has stress concentrations. For the other nodes, the stress magnitude is similar, and also, both LAM presents the same behavior of increases of the damage to wind speed greater  $17m/s$ .

Depending on the method used to transfer the loads, different damage magnitude are estimated and can lead to miss calculations. LAM 4NO is a more straightforward method to implement than LAM PD; computationally, it is less expensive, about six times faster, but for damage, analysis is not worth it due to the overestimation of the damage.

Regarding the results obtained using LAM PD, for all nodes, there is an increase of damage after wind speed of  $17m/s$ . As mentioned in Chapter 3 Regarding the aero-elastic simulation results for  $19m/s$  using a wind 3D field with normal turbulence the wind turbine is not producing energy. The controller is set to an angle of  $82^\circ$ . For the entire aero-elastic simulation, the wind turbine has this constant pitch angle, but it has no breaks acting in the rotor, meaning that the wind turbine continues to rotate but at a lower *rpm*. This condition creates constant oscillations in the edge-wise direction. On the nodes analyzed on the local coordinate system, this direction corresponds to the composite materials' transversal direction, which is the weakest of both directions. For this reason, the damage is more significant when the wind speed reaches these values.

After analyzing these results, wind turbine blades have to be analyzed in more than one situation or design load cases (DLC) specified by the certification guides [Guideline and Lloyd, 2010]. To determine the damage distribution under different situations to ensure that the wind turbine blade will withstand these conditions. Also, this methodology has some limitations, first it can not be used to predict one exact value, because when is introduced the ECDF as the second input variable, each damage  $D^i$  correspond to a value of accumulated probability and by generating an uniform random variable it cannot be selected the correct one, however when is regarded the distribution of damage this method predicts this distribution and in the presented case the sum of  $D^i$  represents the 10 minutes damage. Another limitation correspond to the data used to train the network, with more data the surrogate model could predict more variations in the damage, in other words, more data, better predictions.

## 5.5 20 years fatigue damage estimation using DNN

The 20-year fatigue damage is calculated using surrogate model developed before to estimate 10 minutes damage Eq. 5.14, assuming that the 10 minutes fatigue damage is constant through the entire lifespan.

$$D_{20 \text{ years}}^i = 20 \times 52560 \times D_{10}^i \quad (5.17)$$

where  $D_{10}^i$  is the 10 minutes damage per wind speed  $i = [4, 24]$ , 52560 transform the 10 minutes damage to one-year damage and 20 estimates the 20 years damage. The 20 years fatigue damage should be smaller than 1; otherwise, the blade has failed at this point. Using this equation, the damage is calculated per wind speed, assuming that this one is constant throughout the lifespan. To correct this assumption and create a more realistic case, the damage calculated per each wind speed is proportional to a weight representing the proportion of wind speed acting in a year. In this case, the wind speed during 2017 at la Haute Borne, Grand Est, France [Engie, 2019], and translated to the hub height of DTU wind turbine is assumed to be acting for the entire wind turbine life. To Equation 5.17 is added a weight constant depending on the wind speed proportion as

$$D_{20 \text{ years}}^i = 20 \times 52560 \times w_{ws}^i D_{10}^i \quad (5.18)$$

Finally, the accumulated 20 year damage is the sum of the damage per wind speed.

$$D_{20 \text{ years}} = \sum_{i=4}^{24} D_{20 \text{ years}}^i \quad (5.19)$$

To wind speed weight,  $w_{ws}^i$  are estimated at each wind speed, dividing the number of times that a selected range of wind speed by the total number of times winds speed measurements. Table 5.7 shows that the weight per wind speed range; the highest proportion of wind speed is between 4 and 8m/s; after this wind speed, the proportion decreases.

The 10 minutes fatigue damage DNN was tested to predict the damage of a new time history of the wind 3D field for the same environmental conditions and presented a maximum error of 5%, meaning that this network can simulate different time history distribution of damage per sample. The number of 10-minute damage samples generated per wind speed is 10,000 each representing a different wind field time history. To generate a sample, the DNN needs a wind speed value between  $[4, 25]m/s$  and a uniform variable distribution  $[0, 1]$  with a size equal to the number of damage cycles  $D^i$  per wind speed. Each generation of the uniform variable is independent. Using Eq. 5.18 and Monte Carlo Simulation (MCS) [Rubinstein and Kroese, 1981] is generated each sample to estimate the probabilistic fatigue damage in 20 years.

Table 5.7: Wind speed proportion distribution over a measured year at la haute borne, France

Wind Speed	Weight
$3.5 \geq WS < 4.5$	0.08844
$4.5 \geq WS < 5.5$	0.18730
$5.5 \geq WS < 6.5$	0.23074
$6.5 \geq WS < 7.5$	0.19845
$7.5 \geq WS < 8.5$	0.11395
$8.5 \geq WS < 9.5$	0.05738
$9.5 \geq WS < 10.5$	0.04038
$10.5 \geq WS < 11.5$	0.02800
$11.5 \geq WS < 12.5$	0.02037
$12.5 \geq WS < 13.5$	0.01521
$13.5 \geq WS < 14.5$	0.01005
$14.5 \geq WS < 15.5$	0.00511
$15.5 \geq WS < 16.5$	0.00240
$16.5 \geq WS < 17.5$	0.00110
$17.5 \geq WS < 18.5$	0.00056
$18.5 \geq WS < 19.5$	0.00032
$19.5 \geq WS < 20.5$	0.00012
$20.5 \geq WS < 21.5$	0.00005
$21.5 \geq WS < 22.5$	0.00005
$22.5 \geq WS < 23.5$	0.000002
$23.5 \geq WS < 24.5$	0.000001

One point to clarify is that the uncertainty of the laminate thickness due to manufacturing procedures has not been considered to calculate the fatigue life estimation. Only the uncertainty due to the generation of the time history wind 3D field at an environmental condition is considered for the probabilistic fatigue damage in 20 years.

The 20 years fatigue damage analysis of the wind turbine DTU 10MW with a simplified composite material distribution is carried out using LAM PD. The wind turbine blade has a total of 30613 nodes, where 5575 nodes belong to the three shear webs on the blade and the rest of the nodes to the leading and trailing panels. The laminate in the shear webs comprises three layers and the leading and trailing panels by five layers, each remembering that each layer can be extracted three levels of stress (inferior, middle, superior). The total number of points to evaluate the fatigue damage is equal to 425745.

In this study, only ten nodes along the blade are used to estimate the 20 years fatigue damage LAM PD. The nodes selected are the ones where the force  $F_i^C$  is applied when using LAM 4NO. At this position is applied the forces with a higher magnitude as defined by its physical distribution of loads per section (see chapter 2), meaning that in this point may occur the greater stress concentration due to the application of the load directly to the nodes. However LAM 4NO is not used in the continuation of this damage analysis because of the high level of damage presented for 10 minutes damage.

Using LAM PD, the absolute maximum value of stress in a level for different nodes are found in the inferior level of QQ1 composite material stacked in the second layer. For this reason is selected only the two first layers of the laminate (P2B and QQ1 respectively) at the inferior level to analyze their 20 years fatigue damage, resulting in a total of 20 points to evaluate.

The number of 10 minutes damage generated using 10 minutes DNN damage is 10000 per wind speed, resulting in a total of 210000 realization; as explained earlier, each realization represents a different time history wind field for the same environmental condition.

The computational time to generate 1000 FEM evaluations per each wind speed using LAM 4NO is about three days calculating each wind speed in parallel computation. Using LAM PD for the same number of FEM evaluations, it takes 21 days. However, using these simulation data to train the DNNs only takes less than 40 seconds and less than 6 minutes to generate the 210000 realization, making the fatigue life distribution estimation computational affordable when using a DNN surrogate model. This time may increase if all points are used to estimate the fatigue life distribution, but it will continue to be computationally affordable.

### 5.5.1 20 years fatigue damage results for LAM PD

A probabilistic estimation of 20 years fatigue damage is performed in the wind turbine by using LAM PD. First, the damage at 20 years per wind speed is illustrated in figure 5.26 for node 606 in the first layer P2B at the inferior level. In this case, the higher damages are founded at low wind speed, where there is more probability of occurrence then its descend as the wind speed increases. Also, at wind speeds between 11 and 13m/s, there is a slight increase in the accumulated damage.

The 20 year accumulated fatigue damage per location along the blade z-direction is shown in figure 5.27. The maximum accumulated damage is found near the blade tip also some other node in the middle [516, 549]. The standard deviation for almost all locations is similar, presenting a low variance from the mean value. After the node, 516,  $Z = 79.034$ , the damage values found are not retained in the following calculations because the blade shell model is provided by DTU [Bak et al., 2013] is not closed at the blade tip. The analysis made in the DTU report considers the blade open, saying that this geometry does not impact the results. However, applying the aero-elastic loads as pressure distribution creates a high-stress level due to the open geometry. This layer's accumulated damage does not reach a failure condition in 20 years life span.

Second, the damage at 20 years per wind speed is illustrated in figure 5.26 for node 606 in the second layer QQ1 at the inferior level. In this case, the damage distribution is similar to the first layer P2B, but the confidence interval at low wind speeds is greater for QQ1. Also, the damage magnitude for QQ1 is greater than P2B in all cases, meaning that this layer of composite material QQ1 will fail faster than P2B. However, in this case, after 8m/s, the damage continues to decrease, and only in node 506 (blade tip), there is a slight increase in damage at 12m/s. As said before, All damage distribution per wind speed can be found in

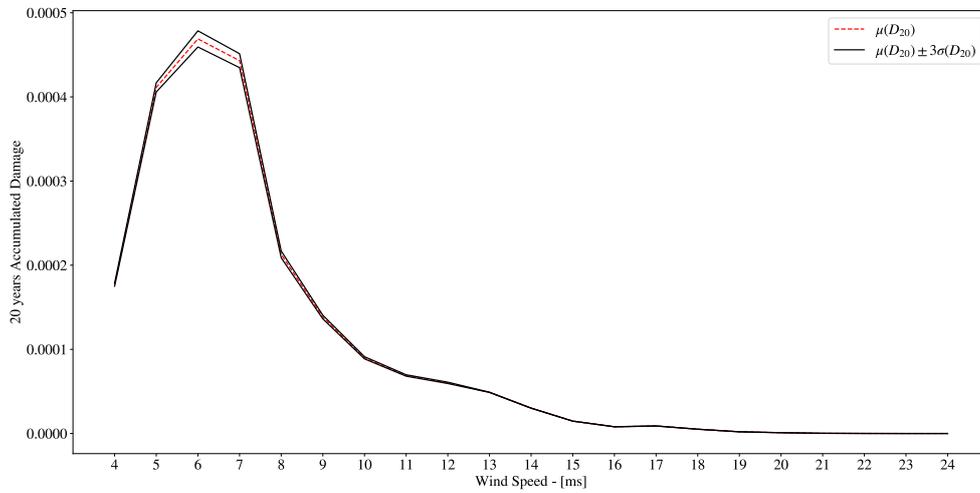


Figure 5.26: 20 years damage distribution at node 606, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

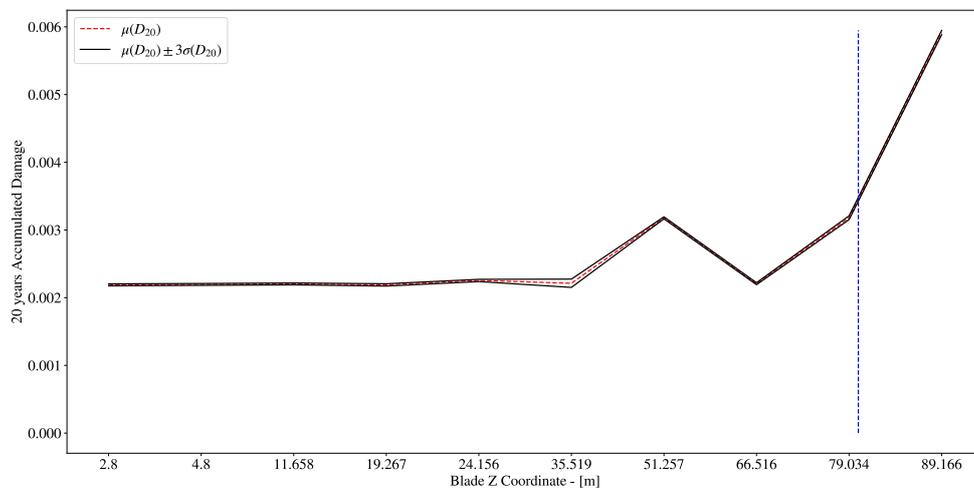


Figure 5.27: 20 years damage distribution along the blade in Z-direction at first layer (P2B), inferior level by using LAM PD.

appendix B.

In figure 5.29 is presented the accumulated damage at 20 years per location along the blade. As can be noticed, it presents a similar distribution as P2B but with higher accumulated damage. Similar to P2B, the blade tip is not considered in this analysis. In this case, all other examined locations, none of them reached a failure condition before 20 years.

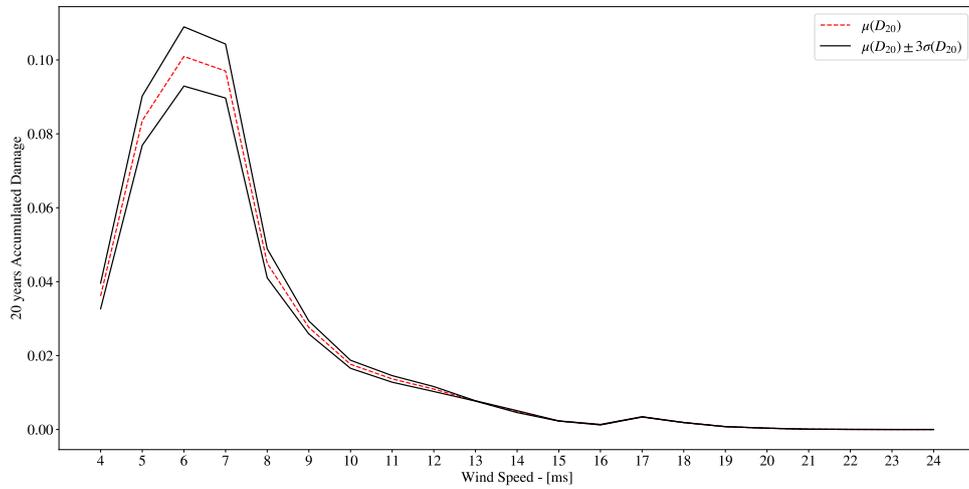


Figure 5.28: 20 years damage distribution at node 606, section 1 of the blade as a function of the wind speed at first layer (QQ1), inferior level by using LAM PD.

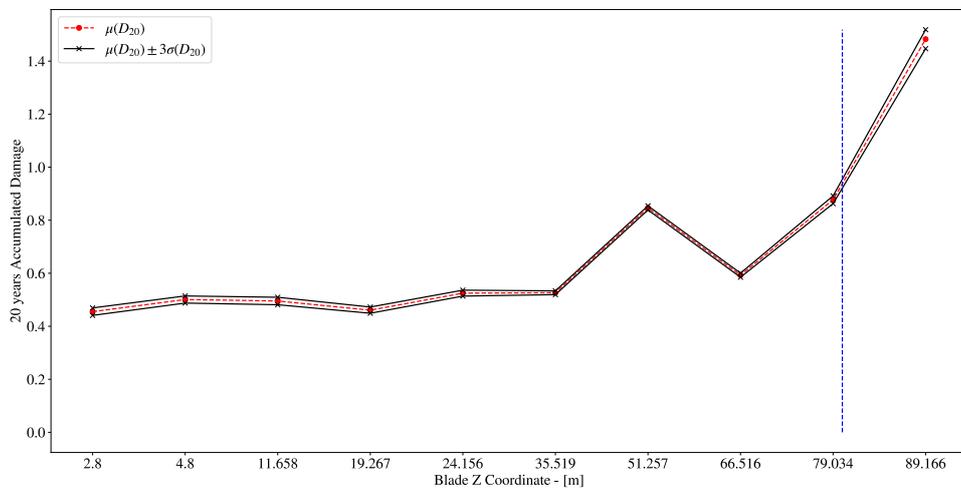


Figure 5.29: 20 years damage distribution along the blade in Z-direction at first layer (QQ1), inferior level by using LAM PD.

Comparing both composite layers, QQ1 presents a greater damage overall nodes than P2B. Also, the node with the highest damage is found for both method to be positioned at  $Z = 51.257m$  of the blade length. For further analysis is only considered this node at layer QQ1 because all other nodes have a lower damage and this one will be the first one to fail. To determine the time to failure a probabilistic estimation of the fatigue life is analyzed next.

## 5.6 Probabilistic fatigue life estimation for LAM PD

After analyzing the accumulated damage distribution at 20 years along the blade, the probabilistic fatigue damage life until can be calculated using equation 5.20, similar as done by Hu et al. [Hu et al., 2016a]:

$$L_{fatigue} = \frac{20}{D_{20}} \quad (5.20)$$

Once the lifespan distribution to failure is calculated, a Kolmogorov-Smirnov test [Massey Jr, 1951] is performed to determine the best fit probability distribution function (PDF). Statistics SciPy library [Jones et al., 01 ] is used to perform this test overall defined PDF on the library. The KS-test return two metrics, the test statistic and the p-value. The test statistic is the maximum absolute difference between the empirical cumulative distribution function (CDF) calculated from x and the hypothesized CDF. The p-value is the probability of observing a test statistic as extreme or more extreme than the observed value under the null hypothesis. Small values of p-value cast doubt on the validity of the null hypothesis. In figure 5.30 is shown the KS-test result for the lifetime to failure distribution of node 549 of material QQ1. In this case, the best fit is obtained using a Weibull-exponential distribution. However, the differences between the best PDFs found by the KS-test are minimal meaning that all of this PDF can represent our data, as presented in table 5.8. The best fit are found for a normal distribution, An exponentially modified Normal continuous (Exponnorm), An exponentiated Weibull continuous (Exponweib) and a log-normal distribution (see [Jones et al., 01 ] for the explanaiton of these continuous distributions).

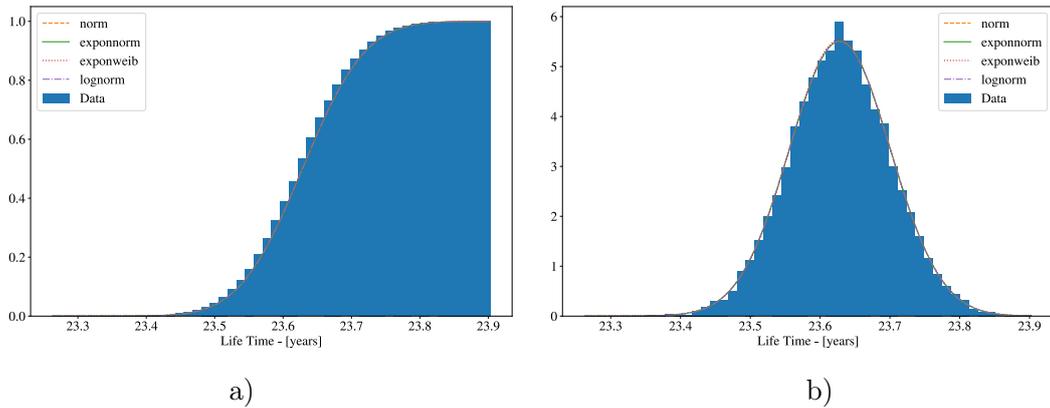


Figure 5.30: Kolmogorov-Smirnov test for fatigue life estimation distribution of node 549 at position  $Z = 51.257m$  and material QQ1 a) CDF and b) PDF.

To prove that the probabilistic fatigue life follows a Normal distribution a less conservative approach is tested. The Anderson-Darling test [Stephens, 1974, Stephens, 1976] is used to test if a sample of data comes from a specific distribution. It is a modification of the KS-test and gives more weight to the tails of the distribution than does the KS-test. The KS-test is distribution free in the sense that the critical values do not depend on the specific distribution

Table 5.8: Kolmogorov-Smirnov goodness fit test for fatigue life estimation distribution of QQ1 node at  $Z = 51.257$ 

PDF	Parameters				Statistics	P-value
Normal	23.6283133	0.07242199	-	-	<b>0.0062011</b>	<b>0.83658789</b>
Exponnormal	0.03073069	23.6260887	0.07238776	-	0.00619846	0.83696514
Exponweib	13.0963707	124.440019	-0.07963451	23.5016492	<b>0.00517517</b>	<b>0.951623</b>
Log-Normal	0.00305343	-0.08881627	23.7170217	-	0.00584435	0.88419696

being tested. This test returns a statistic that is compared to different critical values at different significance levels. If the statistics value is greater than a certain critical value the hypothesis that the tested data comes from the selected PDF is rejected. Table 5.9 shows the results of performing Anderson Darling test assuming that the data comes from a normal distribution. The statistic value is lower than the critical value at 1% significance level (also all others values), meaning that the hypothesis that our data comes from a normal distribution is true for this significance level.

Table 5.9: Anderson-Darling test for fatigue life estimation distribution of QQ1 node at  $Z = 51.257m$ 

PDF	Statistics	Critical Values	Significance Levels
Normal	<b>0.2507</b>	0.576	15
		0.656	10
		0.787	5
		0.918	2.5
		1.092	1

Once is proven that the fatigue life for composite layer QQ1 at position  $Z = 51.257m$  follows a Normal distribution, the central limit theorem is used to estimate the fatigue life corresponding to a significance level of 5%.

The fatigue life to failure of this point is equal to 23.509 years which is greater to the 20 years goal. Meaning that the wind turbine blade does not fail after 20 years of continuous usage at this point where is located the maximum fatigue damage.

This simplified material composite wind turbine blade probabilistic fatigue life estimation methodology using LAM PD could be used for other wind turbine components as the tower and the jacket when a detailed analysis is required using shell FEA. Also, the material thickness values used need to be optimized to reduce the mass and wind turbine blade cost. For example, material P2B has a long life until failure meaning that a better distribution of this material in the wind turbine could be found.

## 5.7 Conclusions

The stress distribution throughout the composite laminate differs in magnitude and direction depending the load application method used to apply the loads. The same result is found for the fatigue damage for 10 minutes. LAM PD is the method that presents the best results for stress and damage distribution along the blade. The deep neural network used to predict this 10 damage can predict a new 10 minutes damage from another wind field time history with an error less than 14% for P2B and 9% for QQ1 by only using one time history as training data. Validating the applicability and efficiency of the proposed methodology. The fatigue life distribution of the node with the maximum damage is a normal distribution and the estimated life with a significance level of 5% is 23.51 years, meaning that the blade does not fail before 20 years.



# Conclusions

---

In this chapter, the conclusions of the thesis and future recommendations are presented. Section 6.1 presents conclusions about the load application methods for wind turbine blades, the discontinuity modeling using deep neural network, the fatigue analysis procedure for composite wind turbine blades using aero-elastic simulations, and the probabilistic fatigue life estimation of the wind turbine blade. Future recommendations are provided in Section 6.2.

## 6.1 Conclusions

A fatigue analysis procedure for composite wind turbine blades, including unsteady wind 3D field simulation, the coupled dynamic response joining aerodynamic models, control and electrical system (servo), and structural (elastic) dynamics models to enable nonlinear simulation in the time domain. Detailed stress analysis in wind turbine blade shell FEM and fatigue damage evolution is proposed in this thesis. The wind field simulation applies Veer's method to simulate 10 minutes wind field based on a 10 minutes mean wind speed  $WS_{10}$ , 10 minutes turbulence intensity  $I_{10}$ , and 10 minutes mean shear exponent  $\alpha$ . Using the simulated wind field realization, an aero-elastic-servo simulation using beam FEM is carried out in a simplified version of the 10 MW DTU wind turbine with a simpler composite material distribution to calculate the time history of loads and displacements on the wind turbine blade. These time history responses are transferred from the beam FEM to a shell FEM of the wind turbine blade using the called load application method. These methods try to recreate the same behavior found in the beam FEM in the shell FEM by different strategies, The one used in this thesis recreate a pressure distribution along the blade correcting the aerodynamic loads and the inertial loads of the wind turbine blade. Also, this load application method is compared with another method with a different approach to analyze in detail the stress distribution of a composite wind turbine blade, including the stress per layers of composite laminate at different wind speeds. A fatigue failure criterion is used to consider the non-proportionality multi-axial stress states with variable amplitude and mean value. Finally, 10 minutes of fatigue damage at levels per layers of laminate nodes are calculated using the fatigue damage criterion. This study also includes a detailed composite laminate schedule and the structural properties comparison with the reference 10MW model and fatigue constant life diagrams for the composite material used in the wind turbine blade model extracted from the real fatigue test database. This 10 minutes fatigue damage is replaced by a surrogate model based on deep neural networks, validated and tested to estimate the 10 minutes damage for a different

wind 3D field simulation, which has been used in the probabilistic estimation of the fatigue life of composite wind turbine blades.

A load application methods comparison in the stress distribution over a composite wind turbine blade shell FEM presents that methods that apply the aero-elastic loads using nodal forces or rigid body elements introduce a certain level of stress concentration in the zone where the loads are applied. Contrary to methods that not only apply the loads as pressure distribution but make a difference in aerodynamic loads and inertial loads, recreating a more accurate and realistic load case as founded similar to various authors [Caous et al., 2018, Bottasso et al., 2014].

However, load application methods are only compared in the stress or damage distribution along the blade but they are not studied to measure the uncertainty introduced by the load application methodology itself. A sensitivity study is performed in a composite wind turbine blade by using different load application methods: rigid body elements (RBE) 4 nodes loads distribution (4NO) and pressure distribution Caous method (PD). To perform these sensitivity studies a two-stage surrogate model methodology is developed to handle the high number of input parameters that each load application method presents. First, a surrogate model based on a deep neural network is constructed to predict the maximum aero-elastic simulation loads for a steady wind condition. For this surrogate model, a methodology is proposed in this thesis, to deal with discontinuity jumps in the model response. Compared with other state of art regression methods, the one proposed outcores the other methods having a lower MSE and higher R2 metrics by not predicting a continuous response as most of the other methods and also by training one surrogate model for the entire domain. This surrogate model has a different relative error per output load prediction, with a maximum error of 8% presenting a good fitting of the outputs parameters. This surrogate model is used to quantify the uncertainties produced in the aero-elastic simulation loads generating the probability density function for each load based on measured wind speed distribution for one year. Second, these aero-elastic loads distributions and uniformly distributed material thicknesses of the composite wind turbine blade are used as input parameters (78 in total, 54 aero-elastic loads and 24 thicknesses) only for LAM 4NO and RBE, In the case of LAM PD, the number of input parameters is 74, where 50 parameters are selected from correction variables of the method itself (for this method the first surrogate model is not used). These loads are transferred to the blade 2D shell finite element model using a load application method, allowing a detailed stress distribution on the wind turbine blade. Subsequently, it is extracted the Von Mises plane stress state at the blade root as an output parameter for the sensitivity analysis. Due to the high computational cost of performing a global sensitivity analysis using the blade 2D shell model, a surrogate model based on a deep neural network was constructed to predict the Von Mises plane stress considering all input parameters. A surrogate model was built for two load application methods and then used to estimate the global sensitivity using the Sobol index method. Comparing both Sobol sensitivity analysis can be concluded that:

- LAM PD is the method that introduces less sensitivity, having a great linear interaction and a low high-order interaction than all other load methods.

- Forces and moments between the middle and blade tip acting in the edge-wise direction has the most sensible effect on output Von Mises stress at the blade root.
- LAM 4 Nodes presents a lower first order  $S1$  indexes value than load method RBE and PD, but load method 4 Nodes has a higher total order  $ST$  indexes with high order interactions with input parameters that do not present high sensitivity index, contrary to load method RBE and PD, that all high-order interactions occur between variables that have a high sensitivity index.

To validate the results obtained using the two-stage surrogate models, a Morris sensitivity analysis is performed without using surrogate models. In this case, is used directly to the blade 2D shell finite element model to estimate the Von Mises stress. Comparing these results, both sensitivity methods present the same input parameters that produce a high sensitivity  $ST$  Sobol index and  $\mu^*$  for Morris analysis, validating the results obtained using surrogate models. Comparing the execution time of Morris analysis and two-stages surrogate model global sensitivity, considering the training of the network and Sobol index analysis is 720 times faster (without considering the data generation for training). Also, this methodology can be applied to other structural components of the wind turbine as a tower or jacket to calculate their global sensitivity. These findings confirm that the load application method used to transfer the wind load variation plays a critical role in the blade stress estimation and uncertainties in wind load have a higher impact than uncertainties in material thicknesses and must be considered in the fatigue life estimation of the wind turbine blade.

For the fatigue analysis, load application methods: 4 nodes distribution and pressure distribution Caous method (PD) are compared, to show the impact of selecting a load application method in the fatigue life estimation of a wind turbine blade. First, the stress distribution varies depending on the load method used. load method PD presents a more consistent stress distribution along with the blade composite laminate, where symmetrical ply positions with the same material present a similar behavior (both in compression or tension) but with different stress magnitude. On the other hand, load method 4 nodes, at points where the loads are applied, symmetrical ply positions present different behavior (one tension and another compression) resulting in a more complex stress state, also the stress magnitude using load method 4 node is much higher than load method PD. All these changes are due to the stress concentration of applying the loads directly to the node. Using the stress time history, the fatigue damage can be calculated for a 10 minutes aero-elastic simulation. Then, to predict 10 minutes of fatigue damage using different wind field distribution is used as a surrogate model based on a deep neural network. This network predicts takes as input value the mean 10 minutes wind speed and a random variable uniformly distributed that represents the number of ranges counted in the stress time history, to determine the 10 minutes damage distribution. This network is validated by predicting fatigue damage calculated for another wind field with a maximum 13.2% error in a ply with P2B composite material and 8.8% error in QQ1 ply, resulting in a good prediction considering that training is only used one wind field distribution at one wind environmental condition. The fatigue damage is predicted only in nodes where the loads are applied directly to the node, to measure the impact in the most critical zones of the load application method. Load method PD, for all positions along the blade span direction,

presents lower and realistic fatigue damage compared to LAM 4NO. Load method PD presents the higher damage at position  $Z = 19.267$  for ply of P2B composite material, however, for QQ1 ply the damage along the blade span direction presents a similar magnitude. At position  $Z = 19.267$  the ratio factor between QQ1 and P2B is 1.01 and for all other positions is about 300 meaning that this method does not produce stress concentration and these differences are due to geometrical and material properties. Another important aspect is founded by looking at the fatigue damage as a function of the wind speed, for load application method PD, in ply QQ1, there is an increase in the fatigue damage after wind speed  $17m/s$  due to the pitch control of the blade, after reaching this wind velocity and having a normal turbulence environmental condition, the control put the wind turbine at a pitch angle of  $82^\circ$ , does not apply breaks in the rotation bearing and does not produce energy, creating a more dangerous stress state in the blade where the shear webs do not stiff the blade. In the case of load method 4 nodes, this increase in fatigue damage is not observed, due to the assumptions in the physical load distributions where the lift in point C must be twice the lift force in point A to ensure a pressure center near the shear webs, which in this state is not the case. This finding shows the critical impact of selecting a load application method and also adding the pitch control of the wind turbine blade introduces a realistic behavior, leading to more conservative fatigue damage. This result shows the impact of using different load application methods. The response obtained using 4 nodes methods is less conservative than load method PD, resulting in a more damaged blade. Using 4 nodes method for a wind turbine blade optimization may lead to a more heavy and expensive blade, because the life estimation needed to be increased more than load method PD, and more material will be added to mitigate the stress concentration induced by the method.

Using this deep neural network for predicting 10 minutes damage for different wind field time history, a 20 years fatigue damage is calculated assuming a linear fatigue damage extrapolation but adding a weight coefficient as a function of the wind speed. This weight is calculated from a wind speed distribution measured in 1 entire year and assumed that for each year during 20 years life span the same wind speed distribution is the same. For both load application methods, the maximum 20 years fatigue damage is produced at low wind speeds as these have a greater probability of occurrence during the 20-year life span than high wind speeds, similar results are found by Hu [Hu et al., 2016a].

A probabilistic fatigue life estimation using the sampling-based method Monte Carlo simulation is performed by generating 10,000 samples per wind speed resulting in a total of 210,000 samples per each loading method. Then from the MCS is used two tests to determine the corresponding probability density function of the node with the maximum fatigue damage: the Kolmogorov-Smirnov goodness test and the Anderson-Darling test. Both tests accept the hypothesis that fatigue life distribution is a Normal distribution for these nodes. Using the central limit theorem, the resulting fatigue life to failure is 23.51 years with a significance level of 5%, in other words, the blade does not fail before 20 years of continuous energy production. The developed fatigue and fatigue life estimation could apply to other wind turbine components as a tower and jacket.

## 6.2 Recommendations for future works

Some areas can be improved in the future in the procedures developed in this thesis. First, the load application method of pressure distribution developed by Caous could be implemented in another software for finite element analysis where the inertial forces could be applied as suggested by the author as acceleration fields and not by using distributed forced proportional to the surface area, as applied in this research using Code\_Aster. Also, as the pitch control of the blade can present a non-producing energy behavior, the pressure coefficient at a greater angle of attack could be calculated used 2D software Xfoil. In this study, the maximum angle calculated is  $\pm 32^\circ$  and after this angle, the pressure coefficient is assumed to be this last one. For future research, the blade pitch control must be optimized for this wind turbine blade to ensure an energy production simulation for all wind speeds between cut-in and cut-out. Second, the global sensitivity analysis methodology presented in this thesis is validated using a Morris analysis, for future research is recommended to calculate the global sensitivity using the blade 2D shell finite element model to compare and validate the Sobol index values obtained. One method may be using GPU solver for finite element analysis to carry out parallel computation at high performance. The sensitivity analysis was performed by assuming steady environmental conditions, for future research an unsteady study could produce different results introducing more uncertainty due to the wind loads. Third, the surrogate models used for the fatigue analysis were only trained using one unsteady environmental condition and one wind field time history. This surrogate presented a good prediction for the amount of data trained, is suggested for future research to increase the number of samples for different environmental conditions and different wind field time history, to create a surrogate model that can predict more wind loads conditions and be used for reliability analysis. These recommendations could be order by their time to implement, in this case first the sensitivity analysis related, then the load application method related and finally the fatigue analysis related because this last one depends on the others.

For future works, the methodology developed to calculate the probabilistic fatigue life distribution could be used for reliability analysis and then a reliability-based optimization design (RBDO) for the wind turbine blade, to minimize the total cost or weight of the blade, or both. A more complex composite material distribution along the blade as used by the reference DTU 10MW wind turbine could be used for the optimization problem or substituting the discrete composite material distribution for a continuous spline function to represent the thickness distribution along the blade. This composite material distribution would lead to a more realistic wind turbine blade optimized design.



# Damage Distribution per Layer

---

## A.1 LAM 4NO 10 minutes Damage P2B Layer: Inferior Level

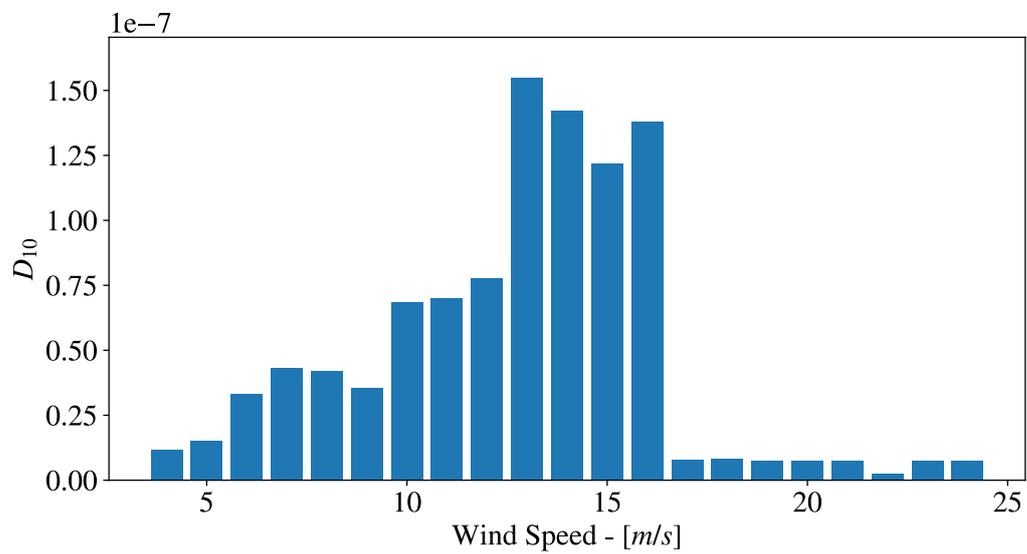


Figure A.1: 10 minutes damage distribution at node 606, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO.

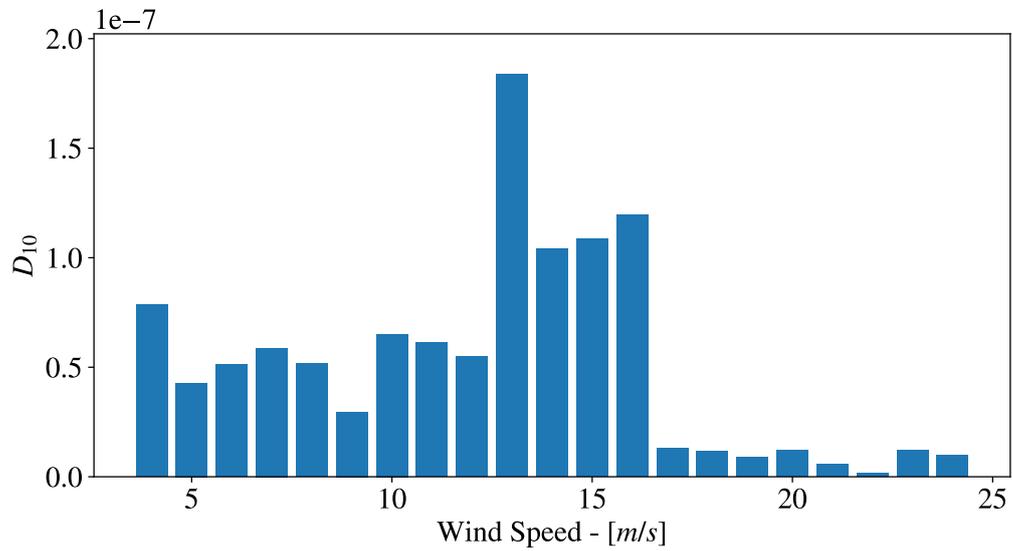


Figure A.2: 10 minutes damage distribution at node 603, section 2 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO.

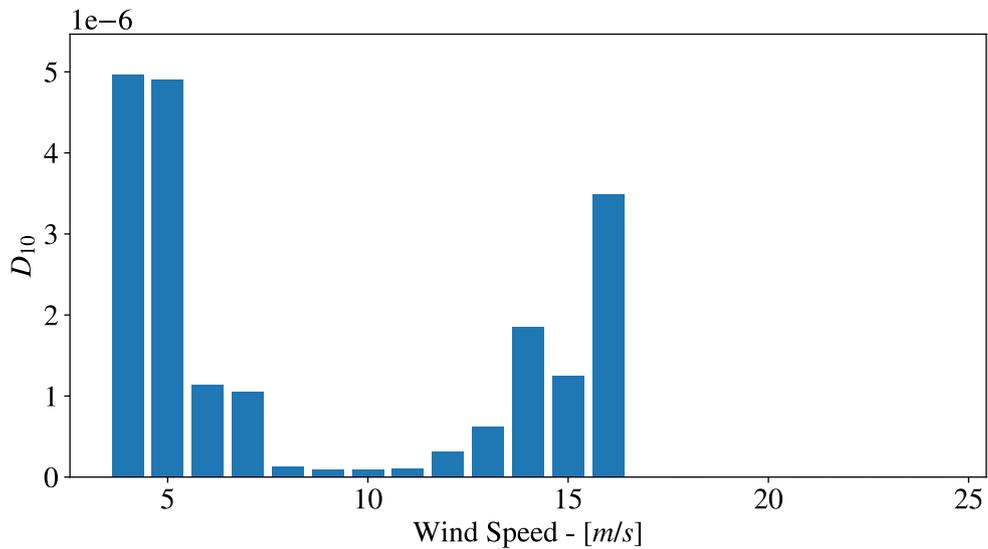


Figure A.3: 10 minutes damage distribution at node 595, section 3 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO.

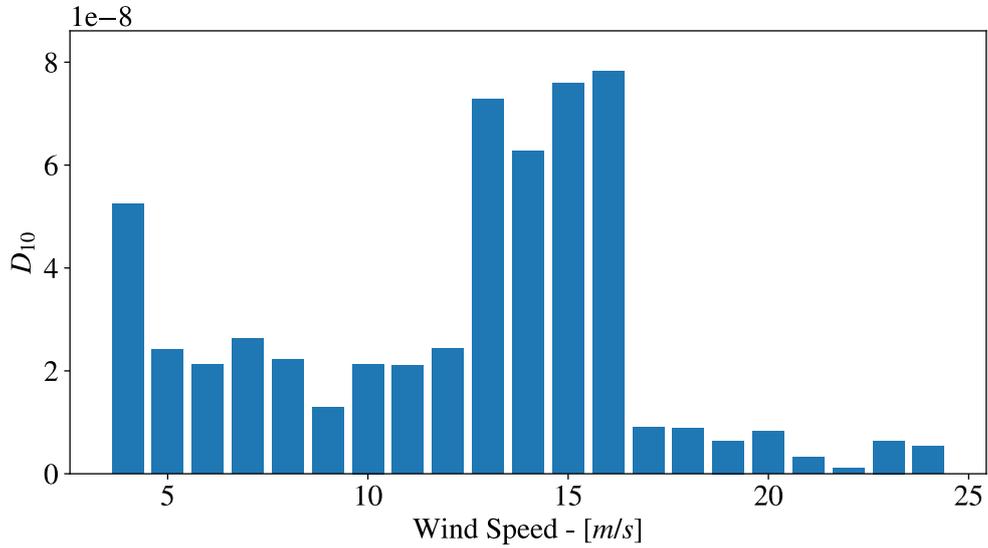


Figure A.4: 10 minutes damage distribution at node 586, section 4 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO.

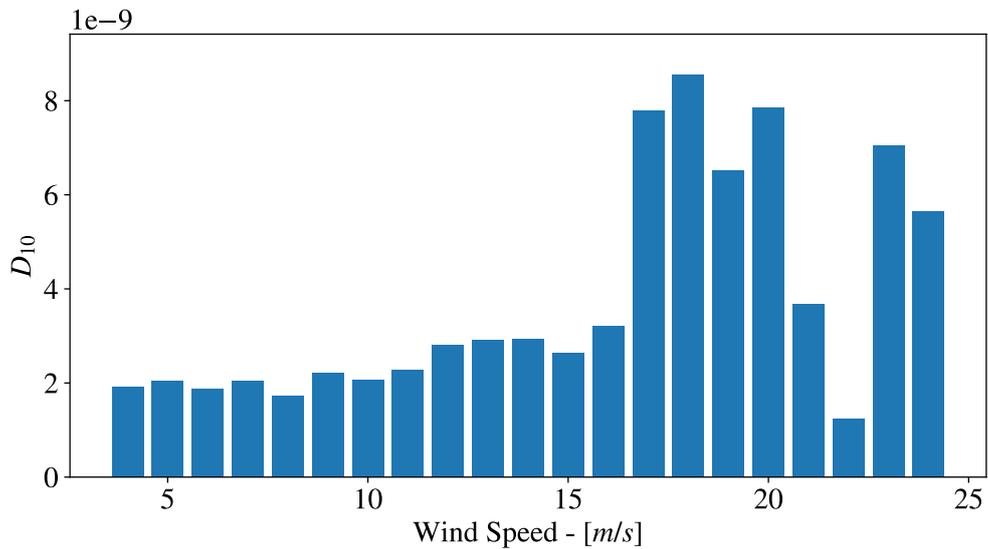


Figure A.5: 10 minutes damage distribution at node 580, section 5 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO.

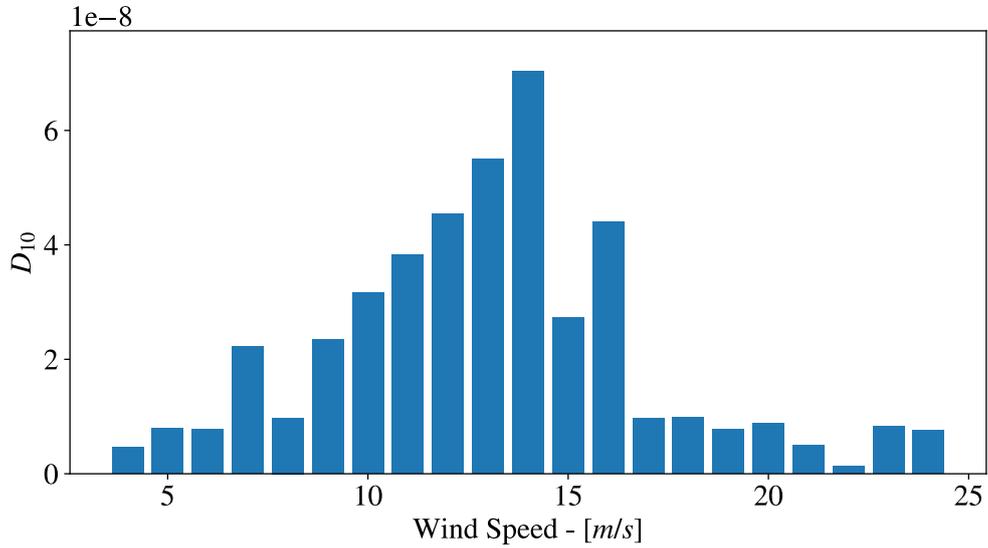


Figure A.6: 10 minutes damage distribution at node 567, section 6 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO.

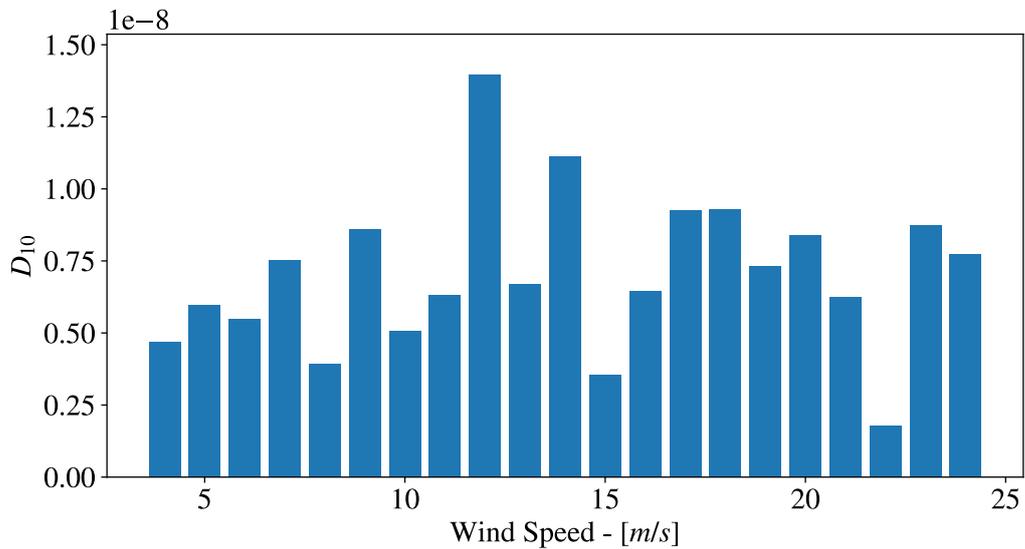


Figure A.7: 10 minutes damage distribution at node 549, section 7 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO.

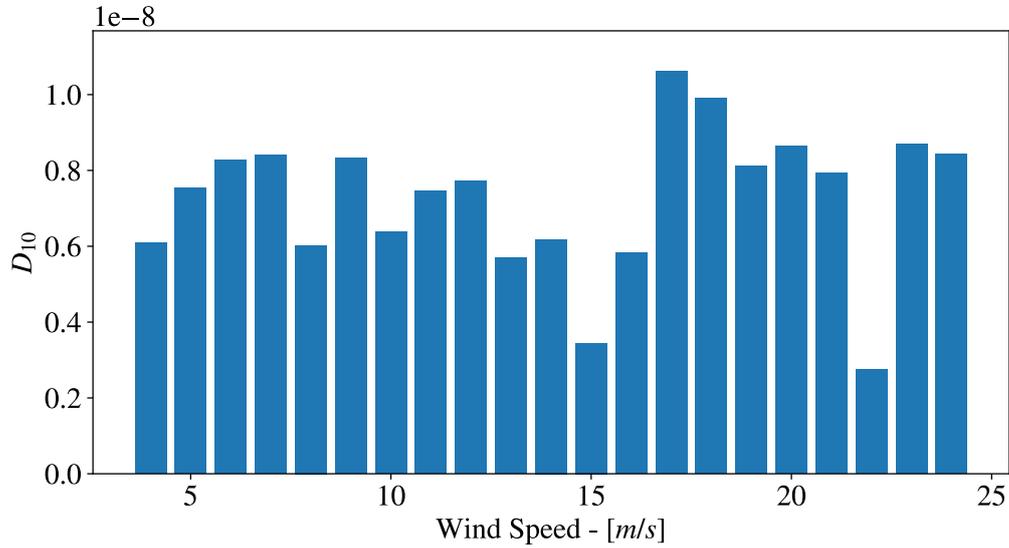


Figure A.8: 10 minutes damage distribution at node 531, section 8 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO.

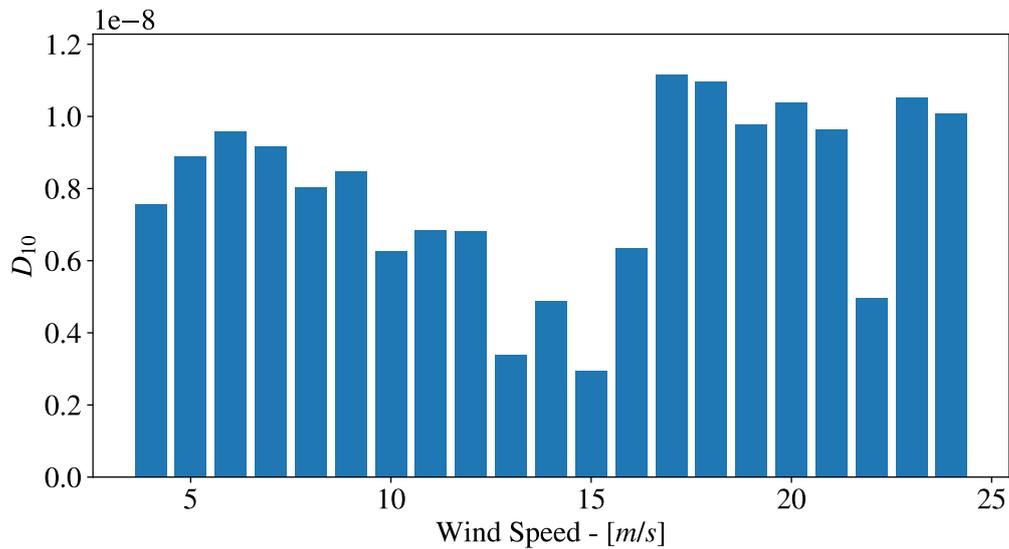


Figure A.9: 10 minutes damage distribution at node 516, section 9 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO.

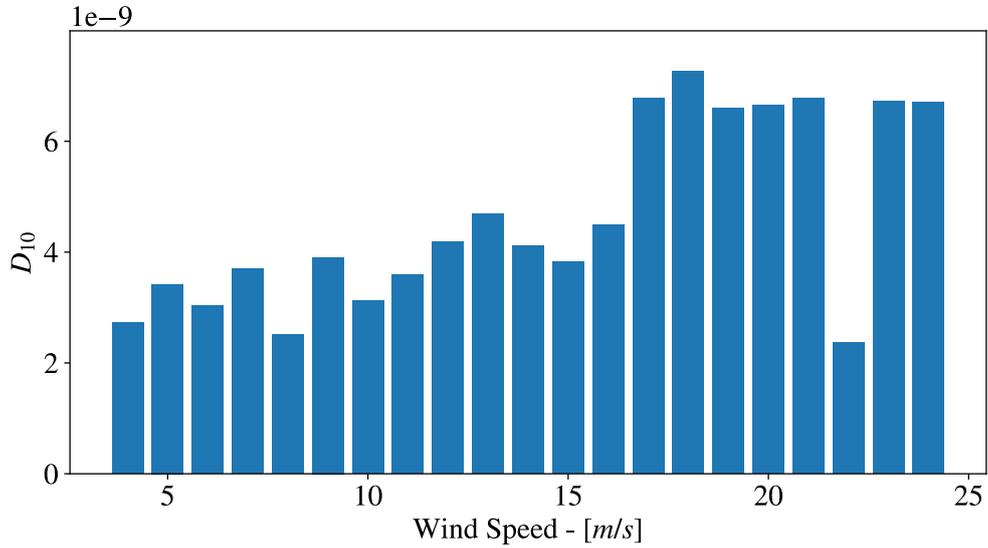


Figure A.10: 10 minutes damage distribution at node 506, section 10 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO.

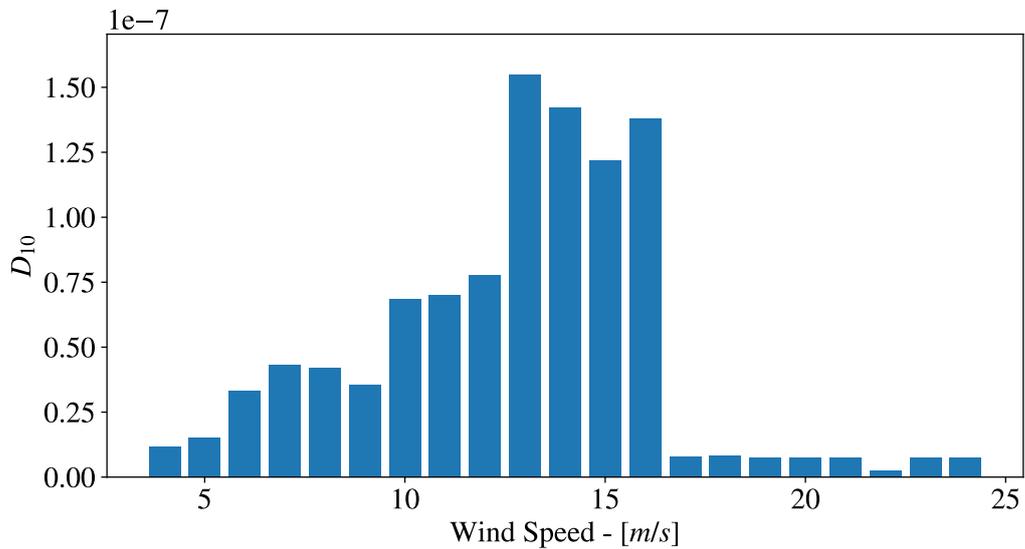


Figure A.11: 10 minutes damage distribution at node 606, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM 4NO.

## A.2 LAM 4NO 10 minutes Damage QQ1 Layer: Inferior Level

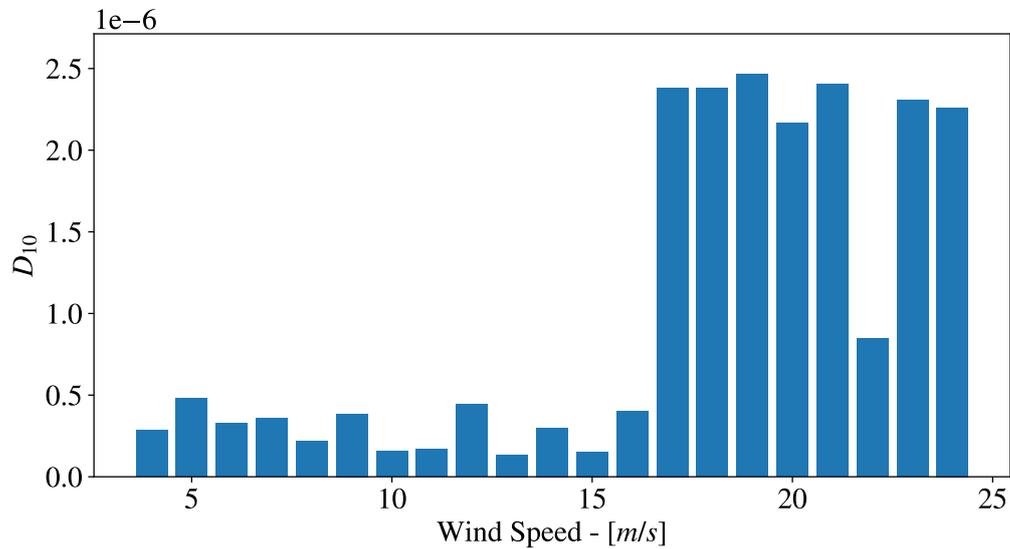


Figure A.12: 10 minutes damage distribution at node 606, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO.

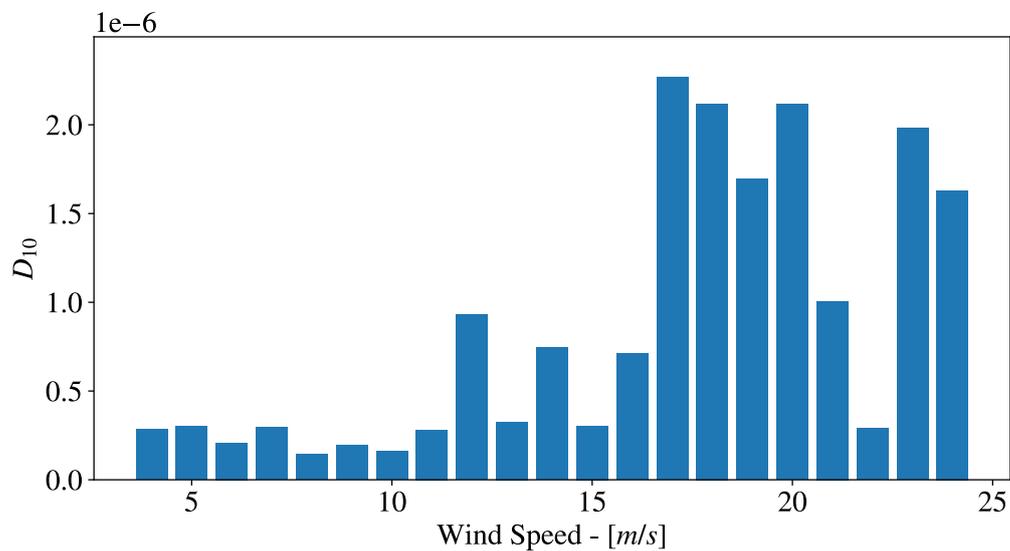


Figure A.13: 10 minutes damage distribution at node 603, section 2 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO.

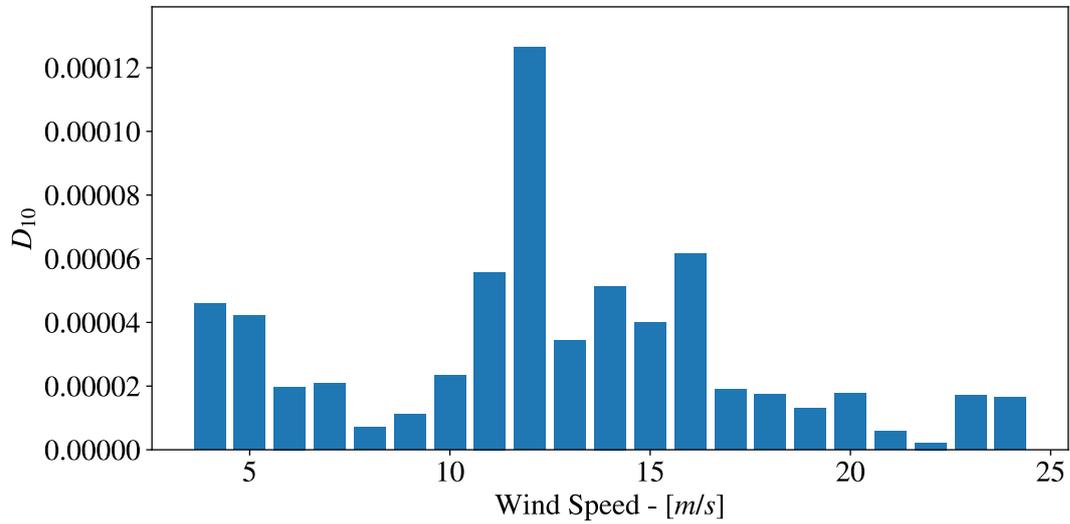


Figure A.14: 10 minutes damage distribution at node 595, section 3 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO.

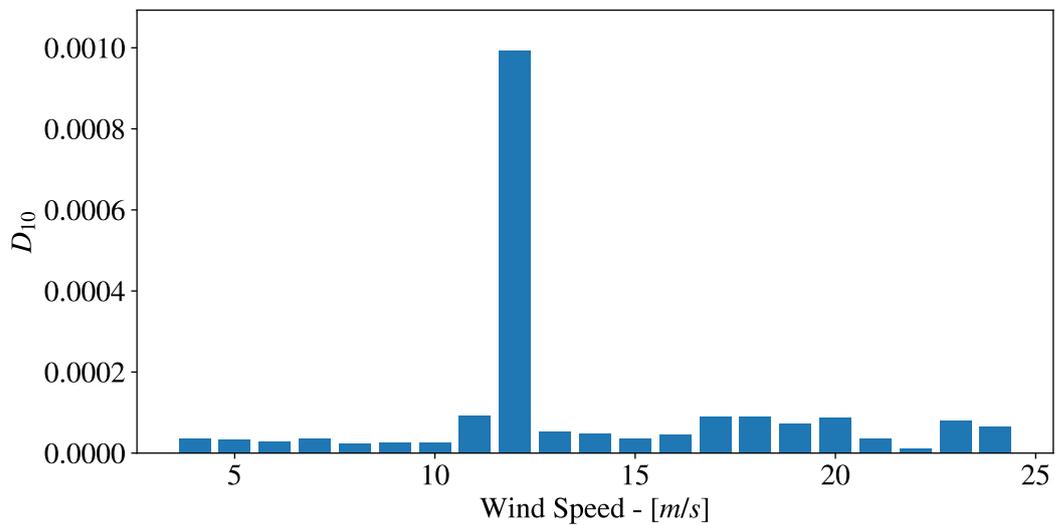


Figure A.15: 10 minutes damage distribution at node 586, section 4 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO.

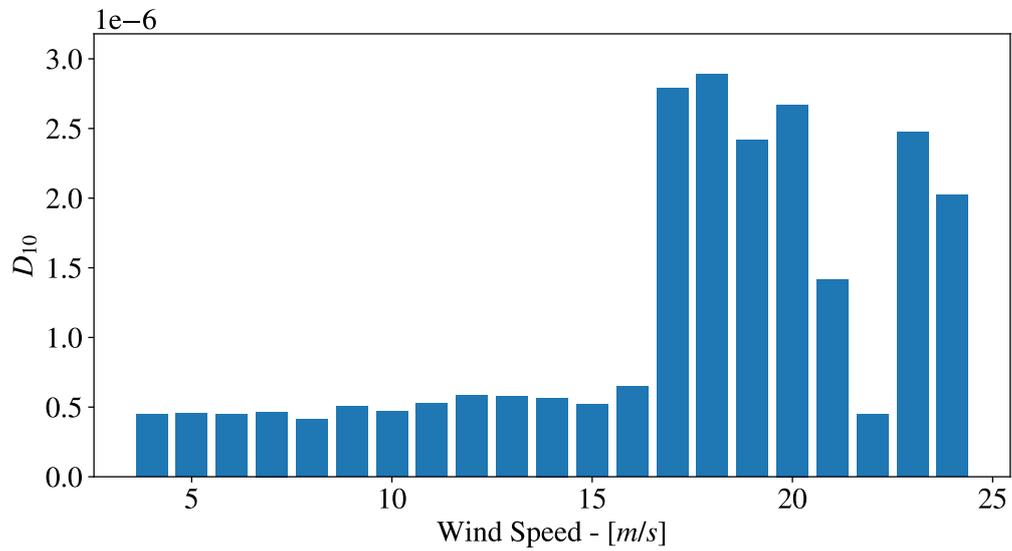


Figure A.16: 10 minutes damage distribution at node 580, section 5 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO.

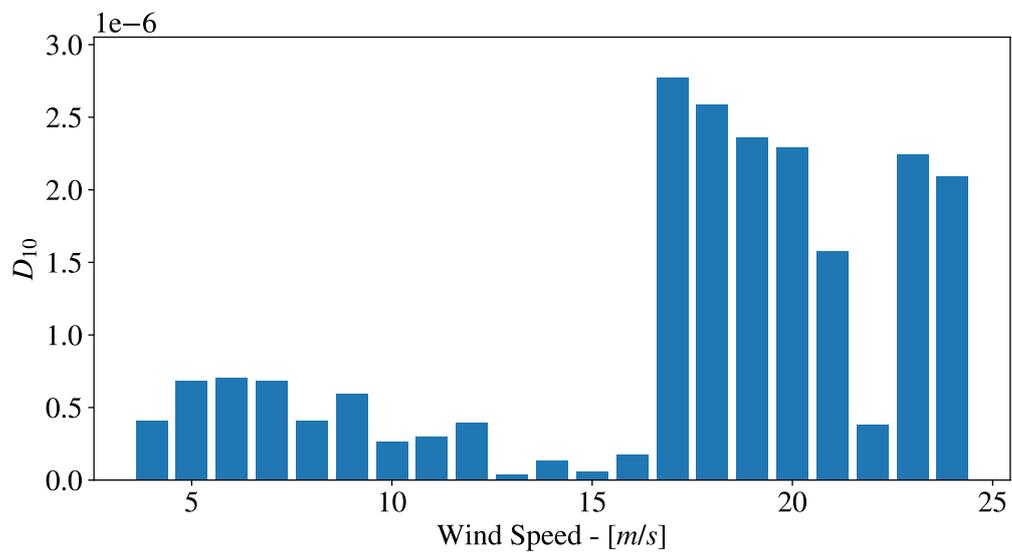


Figure A.17: 10 minutes damage distribution at node 567, section 6 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO.

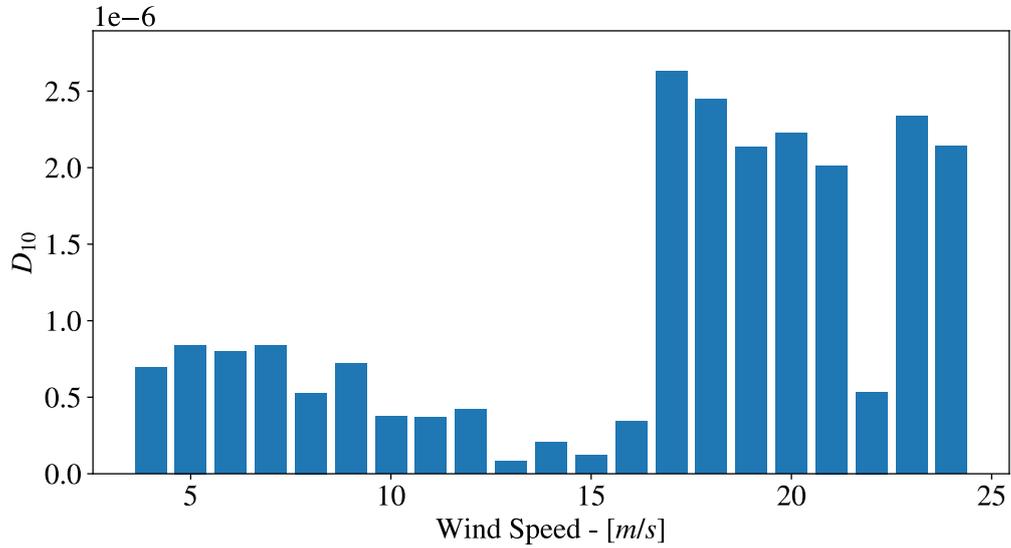


Figure A.18: 10 minutes damage distribution at node 549, section 7 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO.

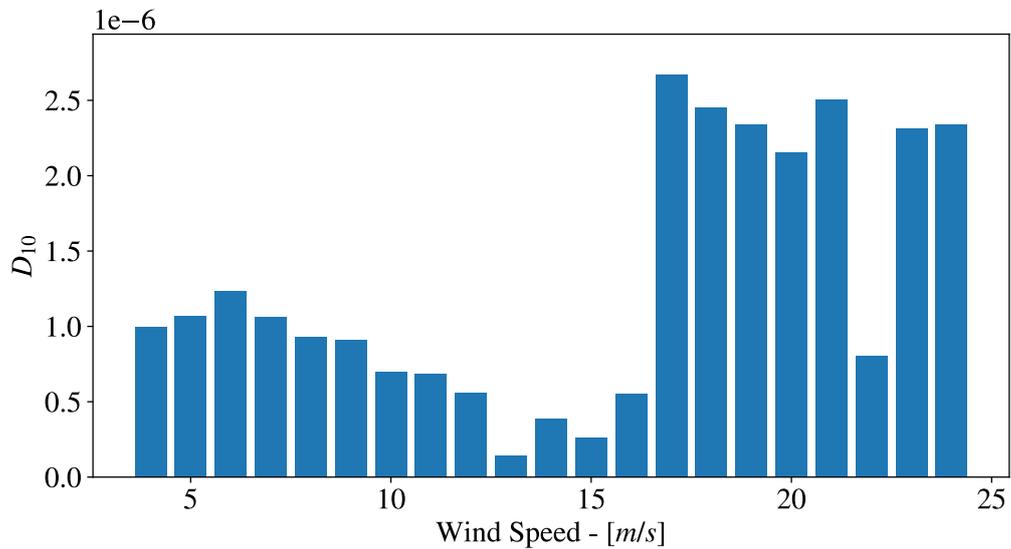


Figure A.19: 10 minutes damage distribution at node 531, section 8 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO.

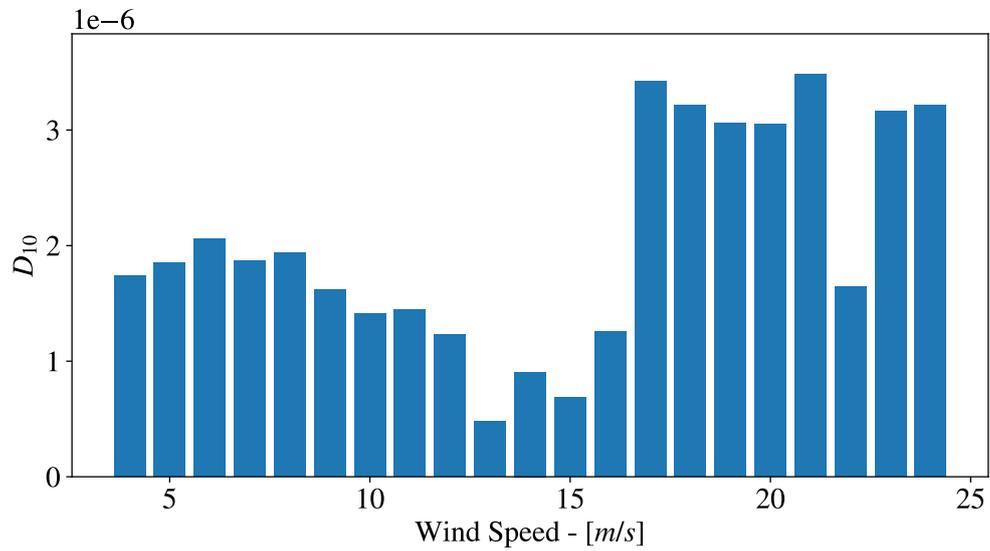


Figure A.20: 10 minutes damage distribution at node 516, section 9 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO.

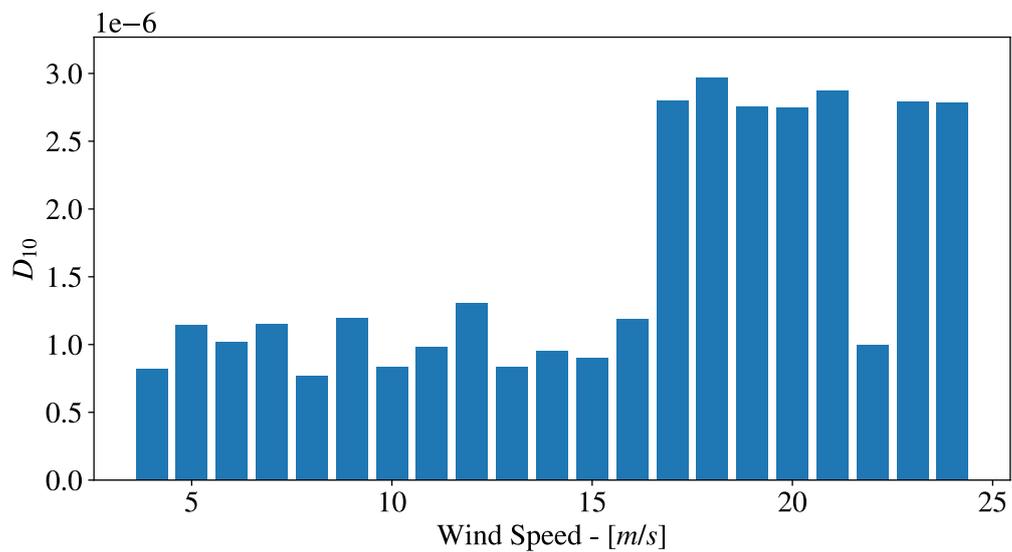


Figure A.21: 10 minutes damage distribution at node 506, section 10 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO.

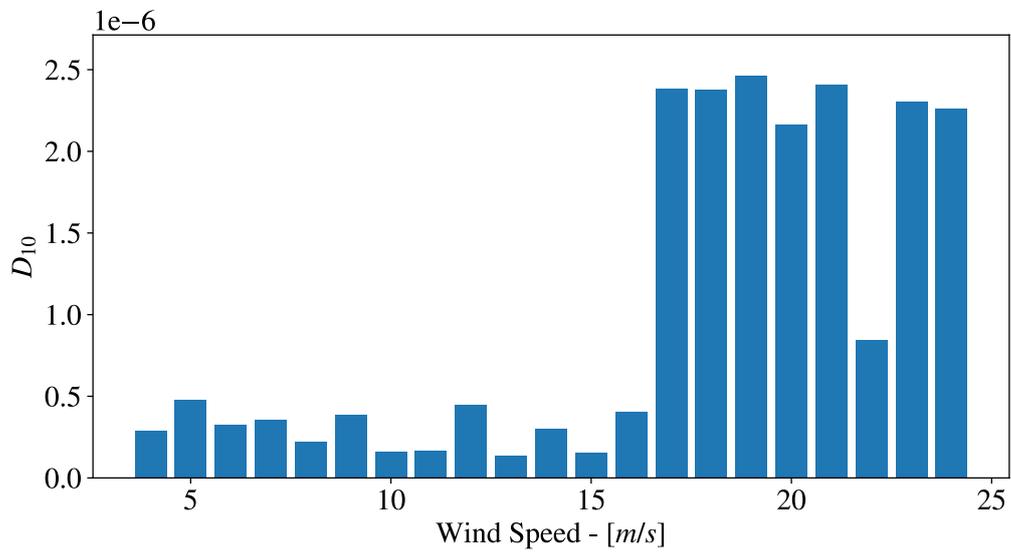


Figure A.22: 10 minutes damage distribution at node 606, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM 4NO.

A.3 LAM PD 10 minutes Damage P2B Layer: Inferior Level

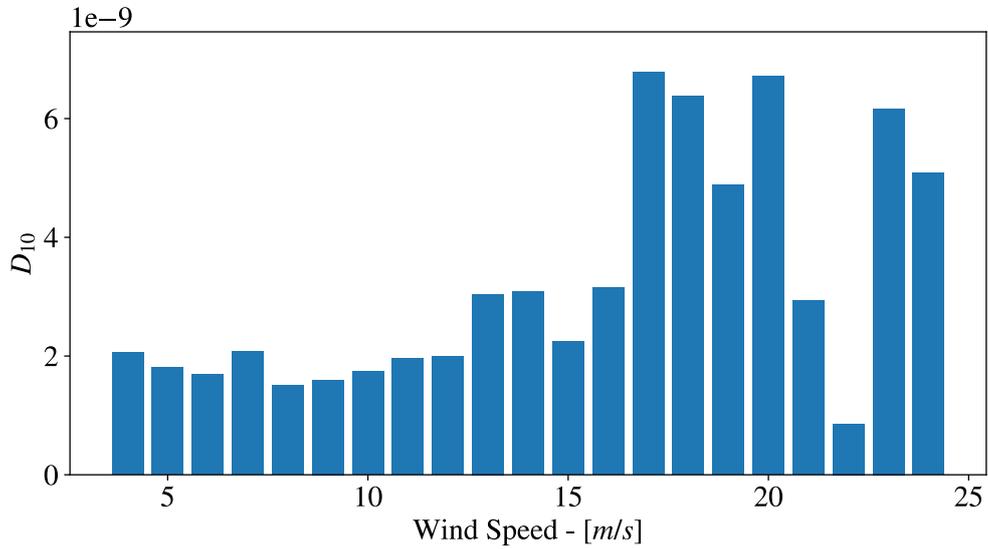


Figure A.23: 10 minutes damage distribution at node 606, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

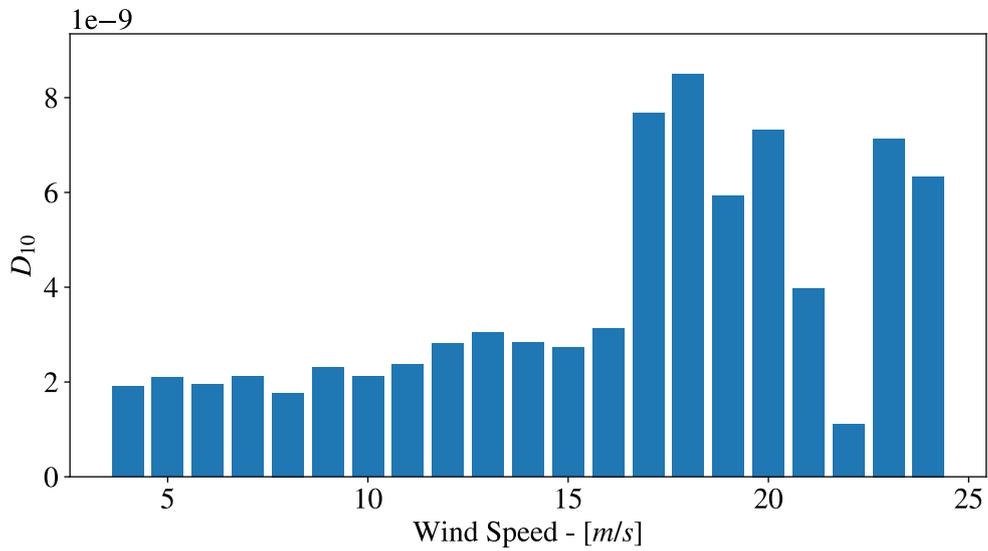


Figure A.24: 10 minutes damage distribution at node 603, section 2 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

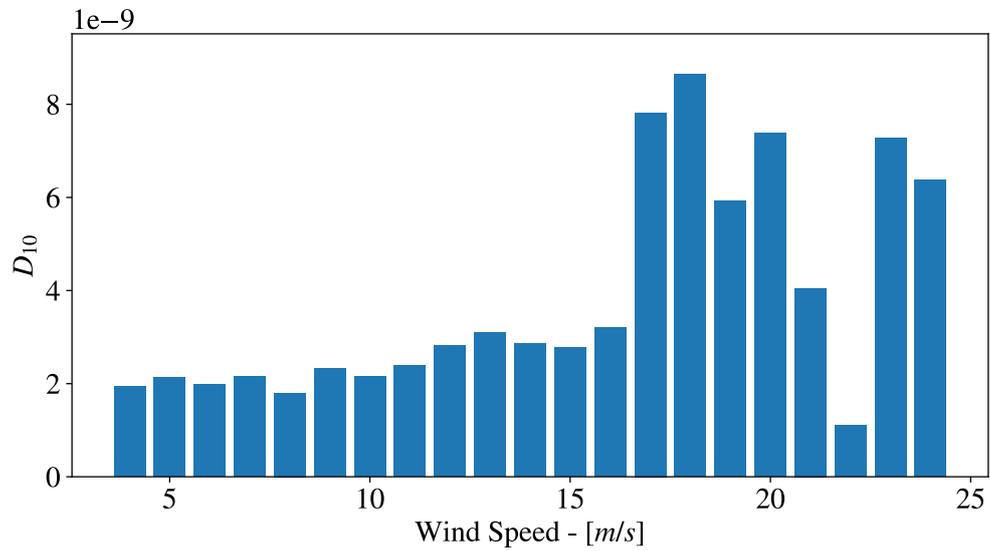


Figure A.25: 10 minutes damage distribution at node 595, section 3 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

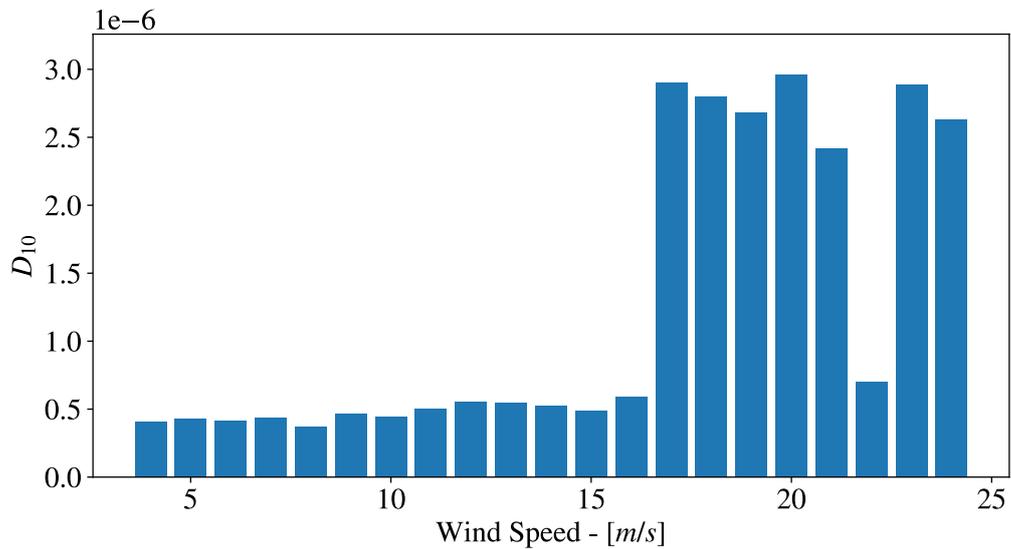


Figure A.26: 10 minutes damage distribution at node 586, section 4 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

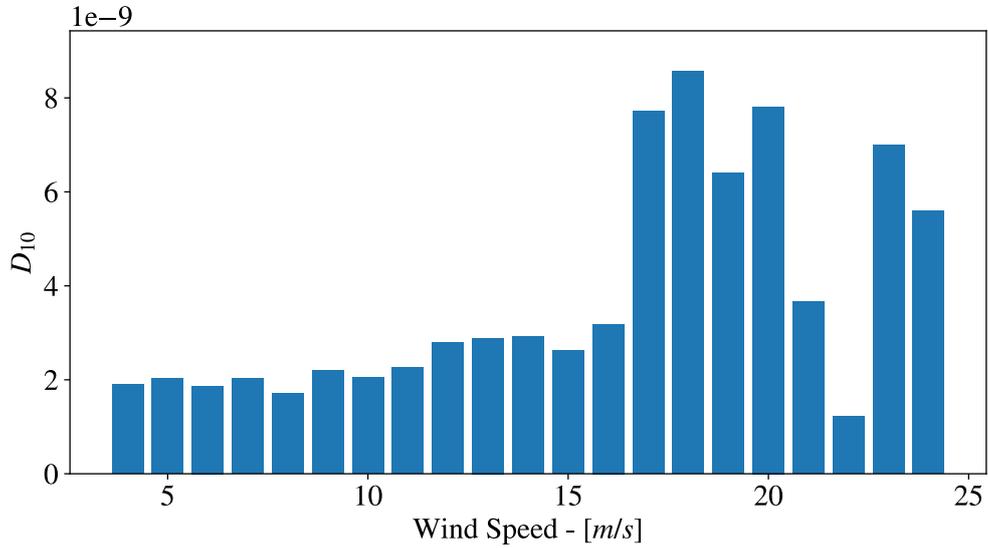


Figure A.27: 10 minutes damage distribution at node 580, section 5 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

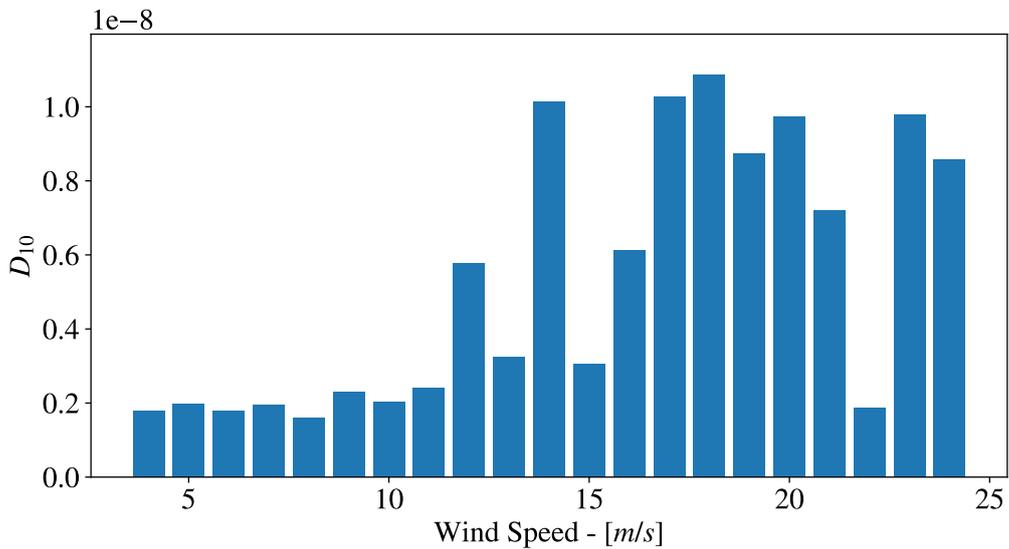


Figure A.28: 10 minutes damage distribution at node 567, section 6 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

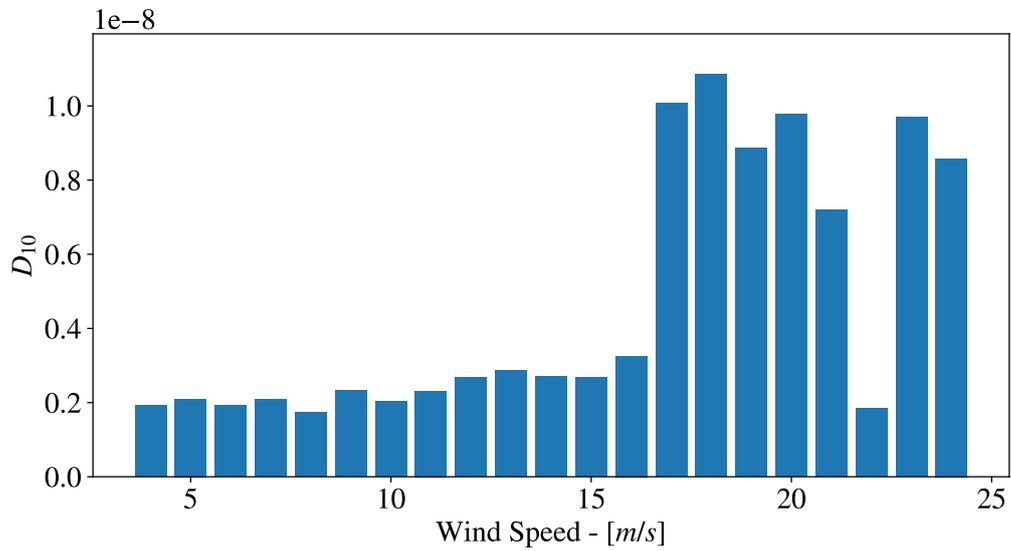


Figure A.29: 10 minutes damage distribution at node 549, section 7 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

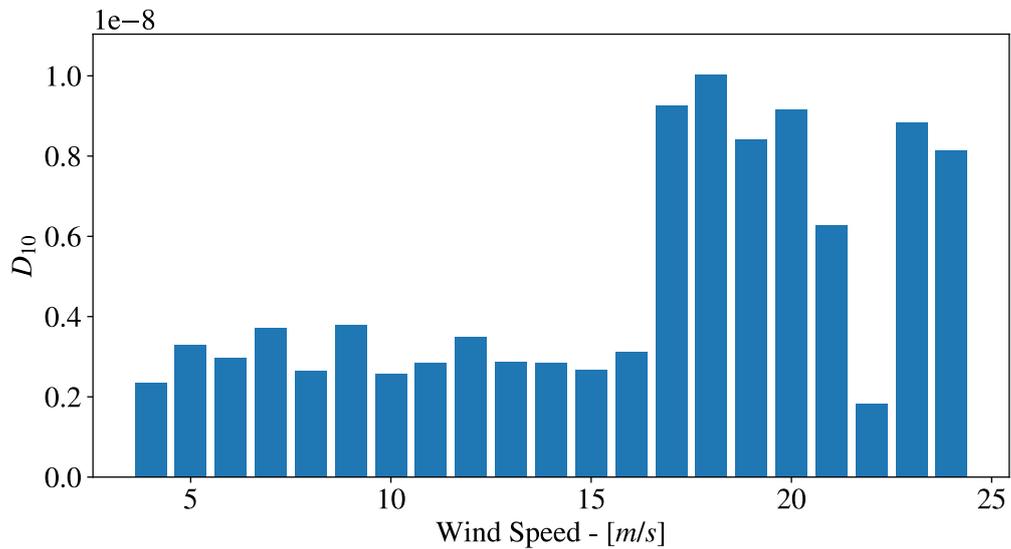


Figure A.30: 10 minutes damage distribution at node 531, section 8 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

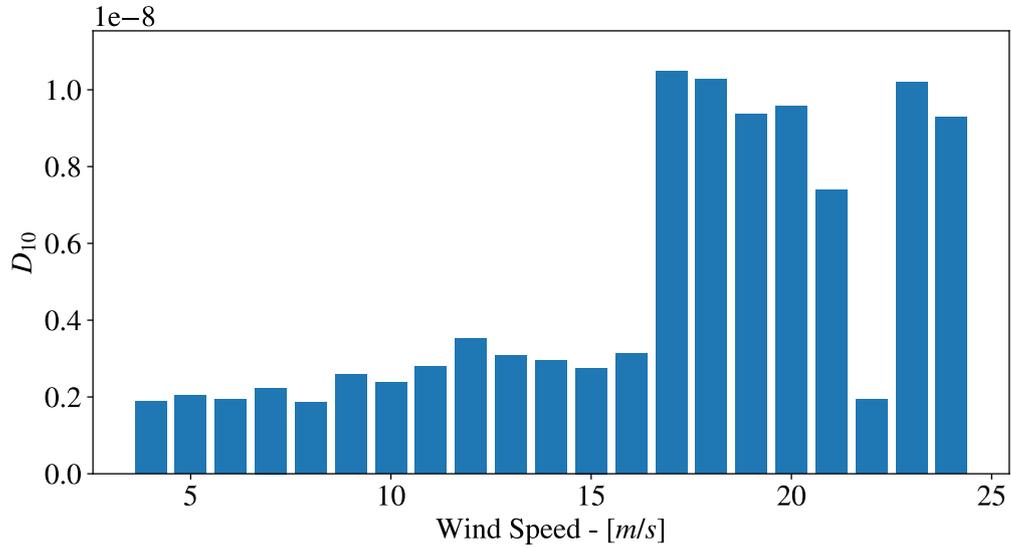


Figure A.31: 10 minutes damage distribution at node 516, section 9 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

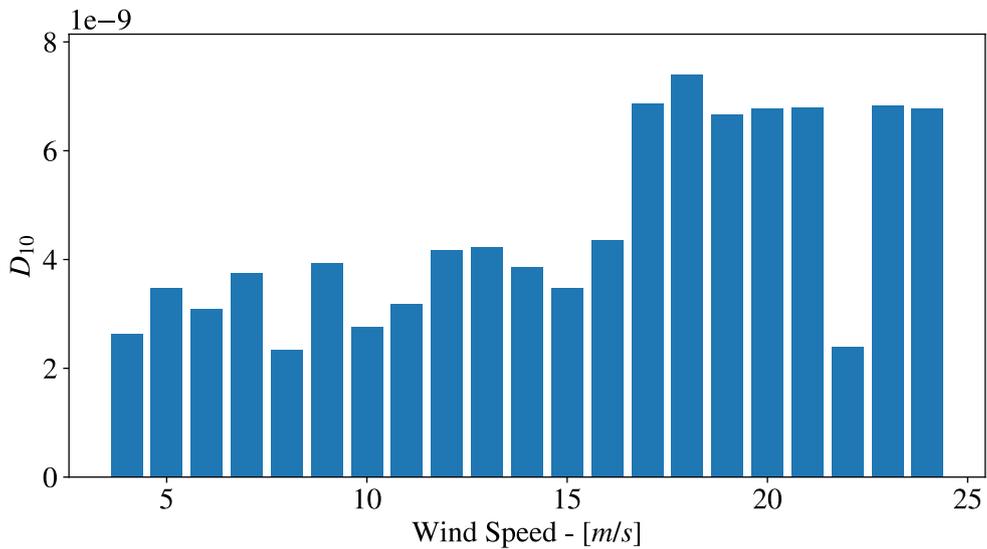


Figure A.32: 10 minutes damage distribution at node 506, section 10 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

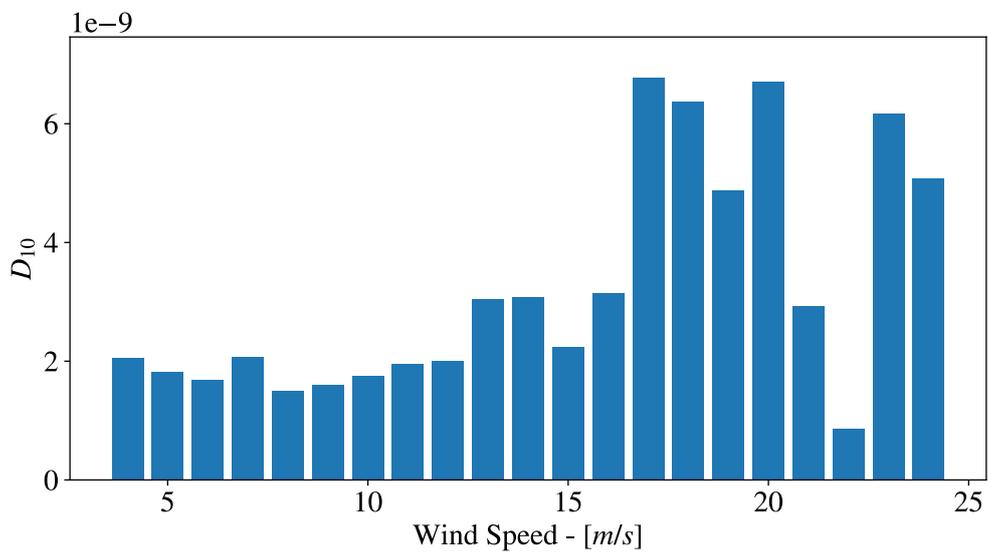


Figure A.33: 10 minutes damage distribution at node 606, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

A.4 LAM PD 10 minutes Damage QQ1 Layer: Inferior Level

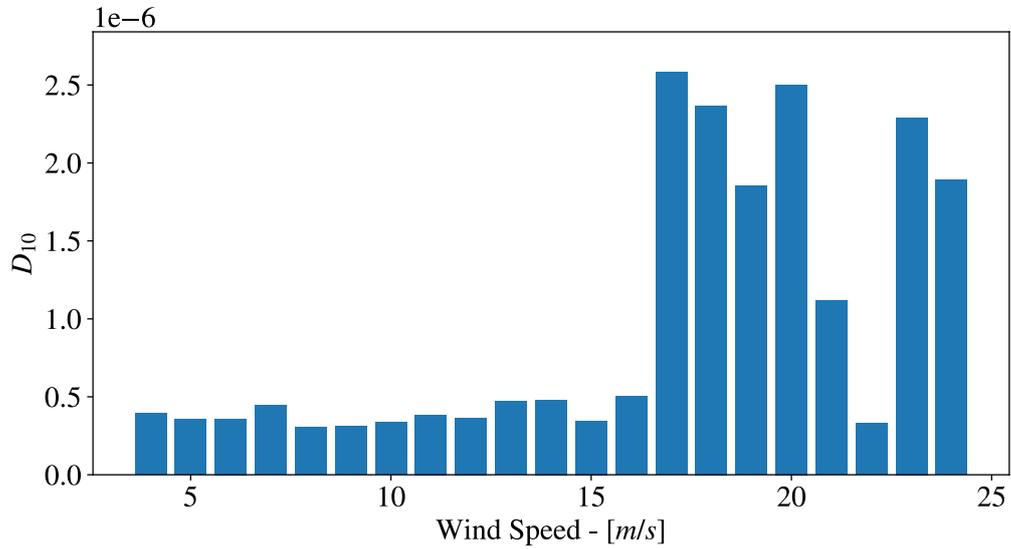


Figure A.34: 10 minutes damage distribution at node 606, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

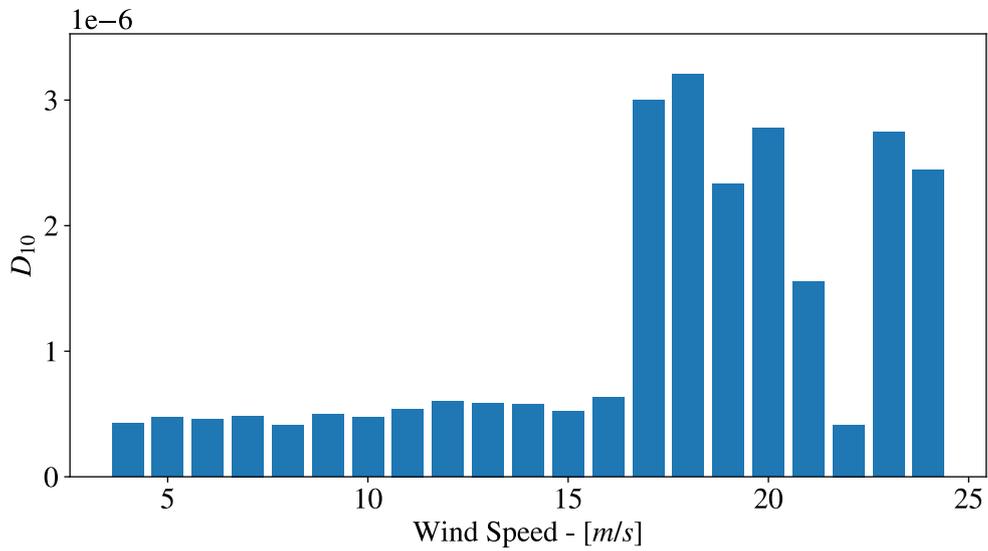


Figure A.35: 10 minutes damage distribution at node 603, section 2 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

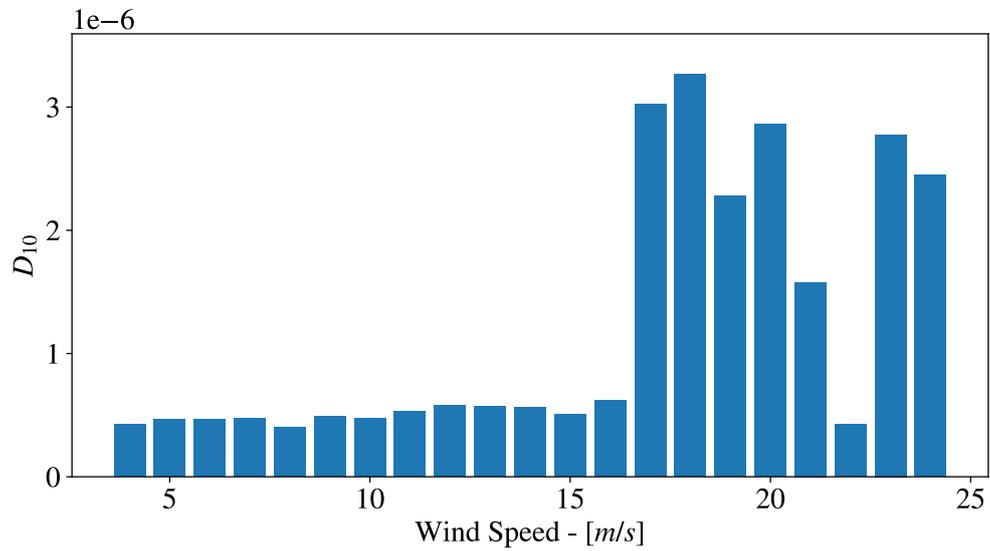


Figure A.36: 10 minutes damage distribution at node 595, section 3 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

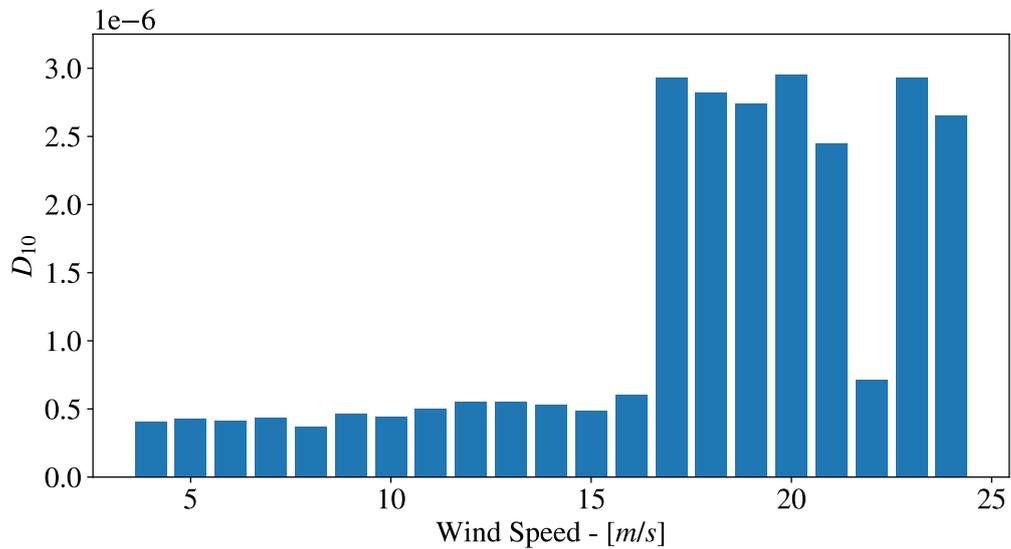


Figure A.37: 10 minutes damage distribution at node 586, section 4 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

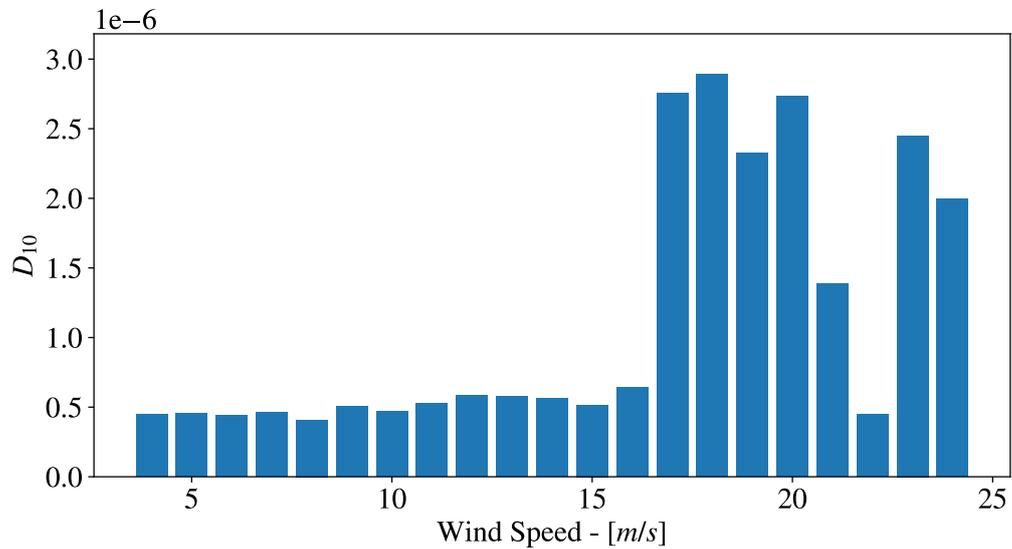


Figure A.38: 10 minutes damage distribution at node 580, section 5 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

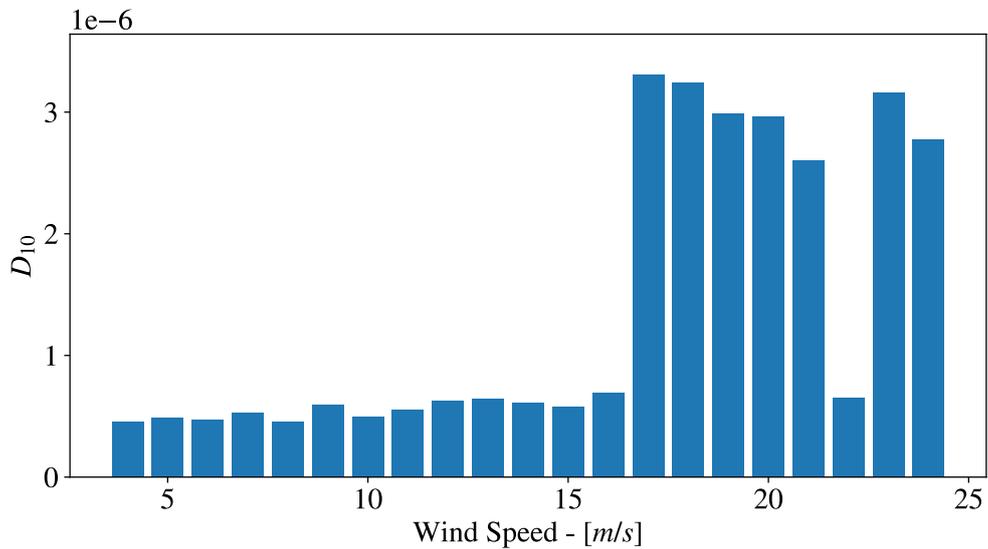


Figure A.39: 10 minutes damage distribution at node 567, section 6 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

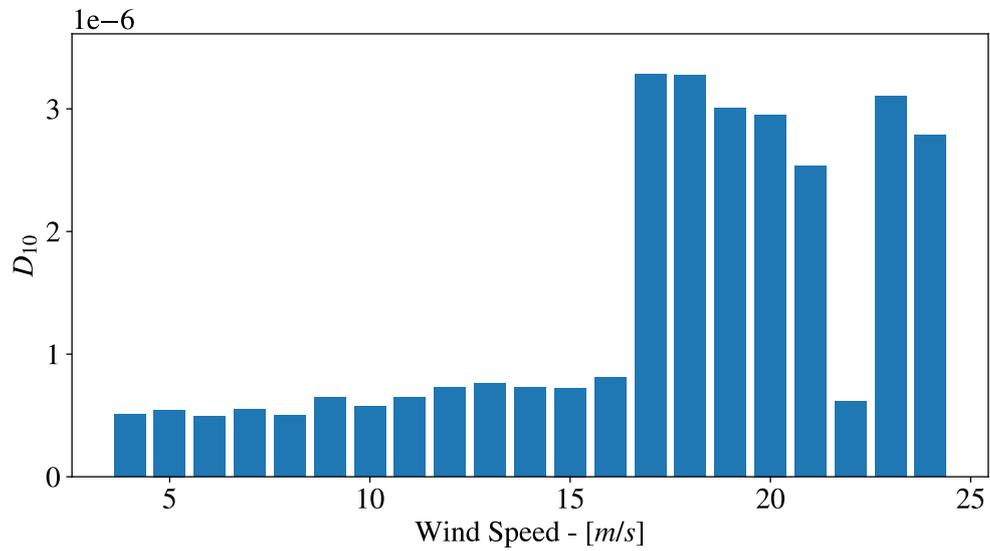


Figure A.40: 10 minutes damage distribution at node 549, section 7 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

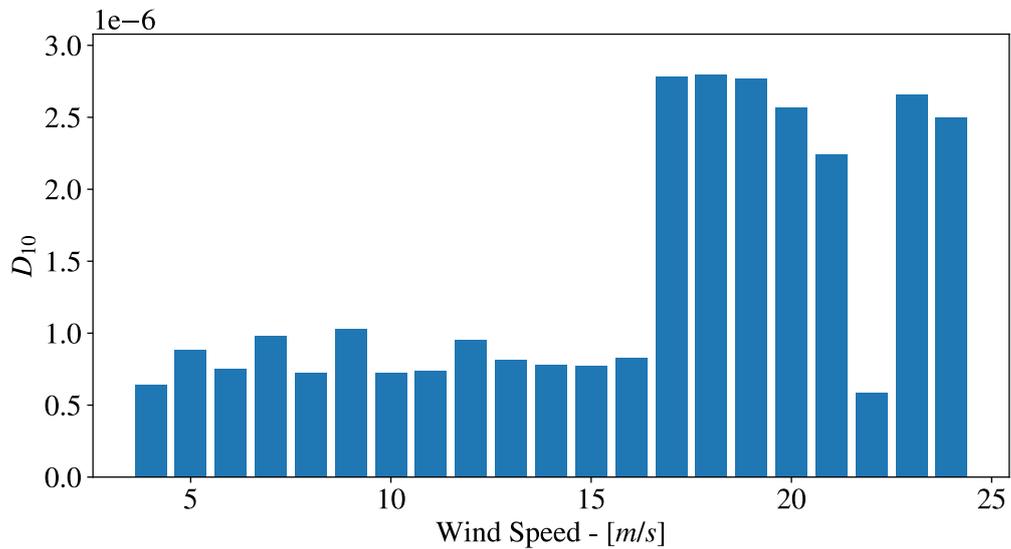


Figure A.41: 10 minutes damage distribution at node 531, section 8 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

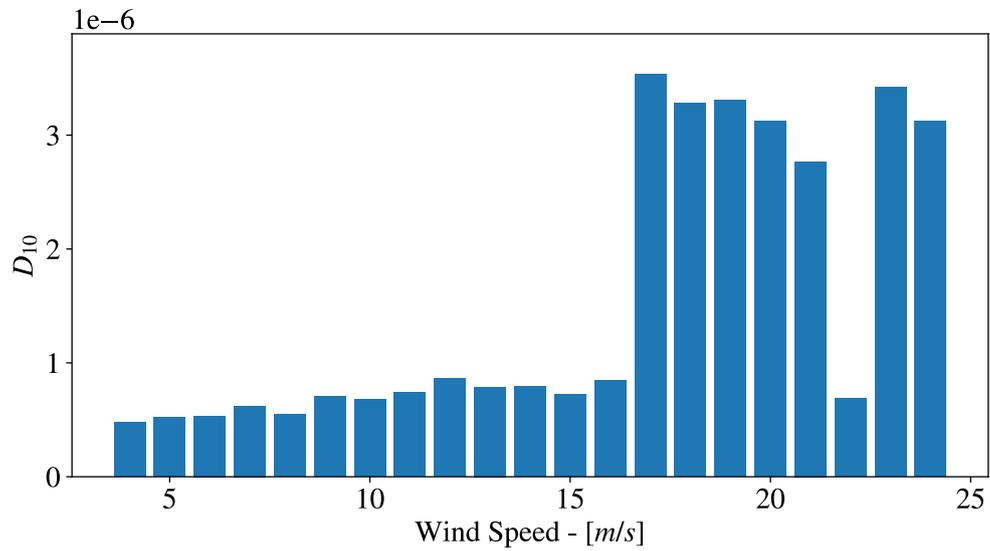


Figure A.42: 10 minutes damage distribution at node 516, section 9 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

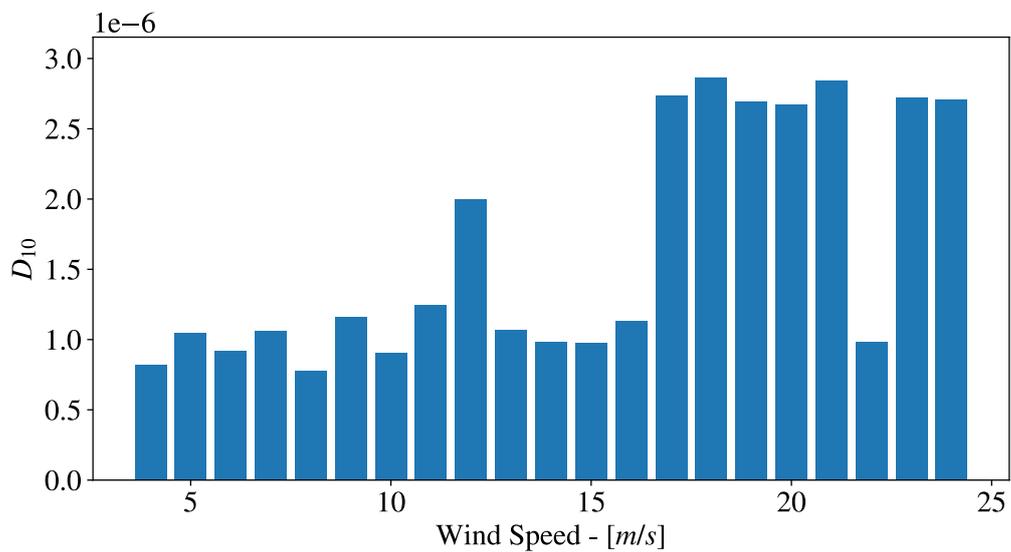


Figure A.43: 10 minutes damage distribution at node 506, section 10 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.



# 20 years Damage per Wind Speed

## B.1 LAM PD 20 years Damage P2B Layer: Inferior Level

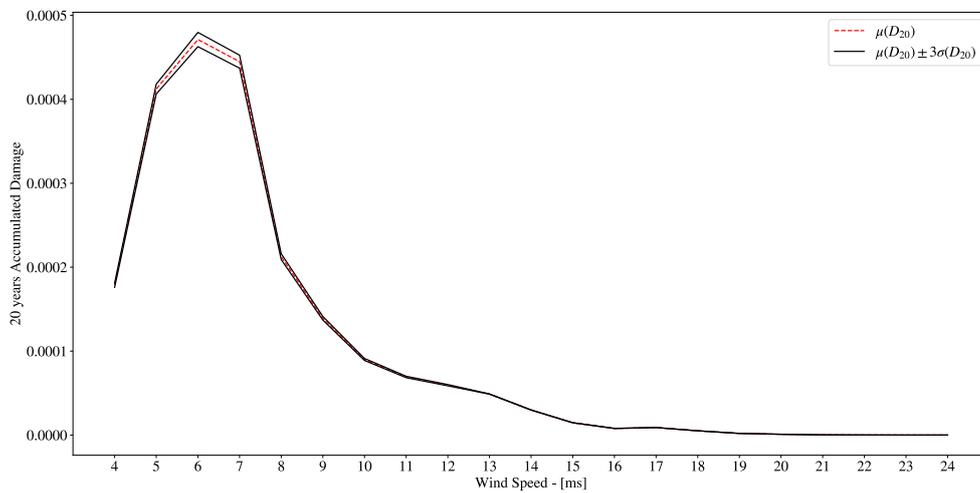


Figure B.1: 20 years damage distribution at node 603, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

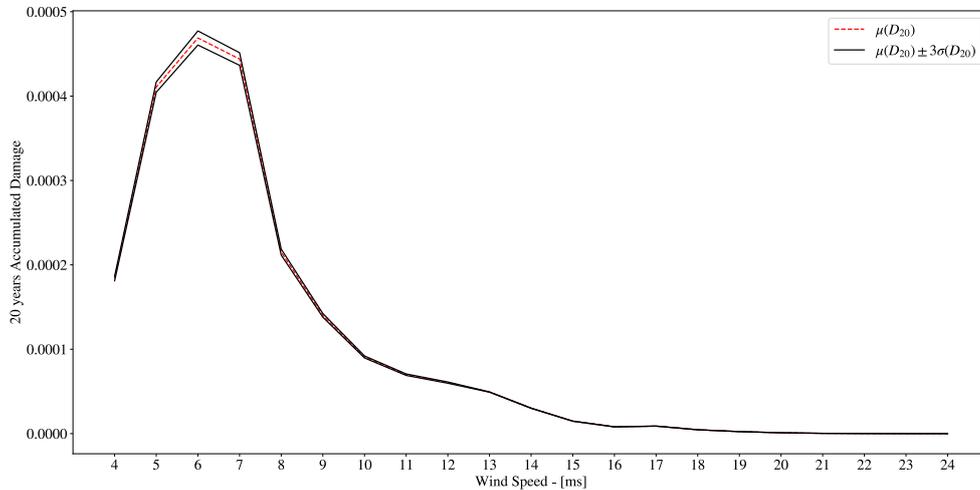


Figure B.2: 20 years damage distribution at node 595, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

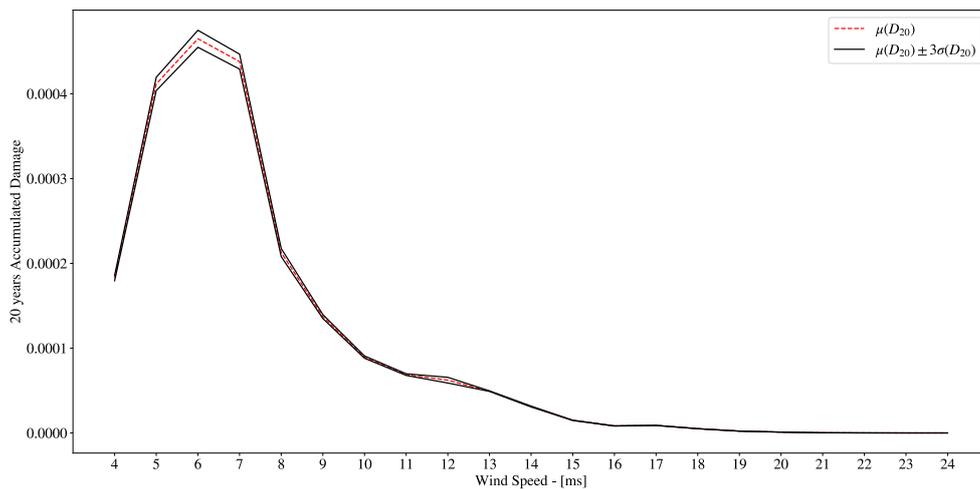


Figure B.3: 20 years damage distribution at node 586, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

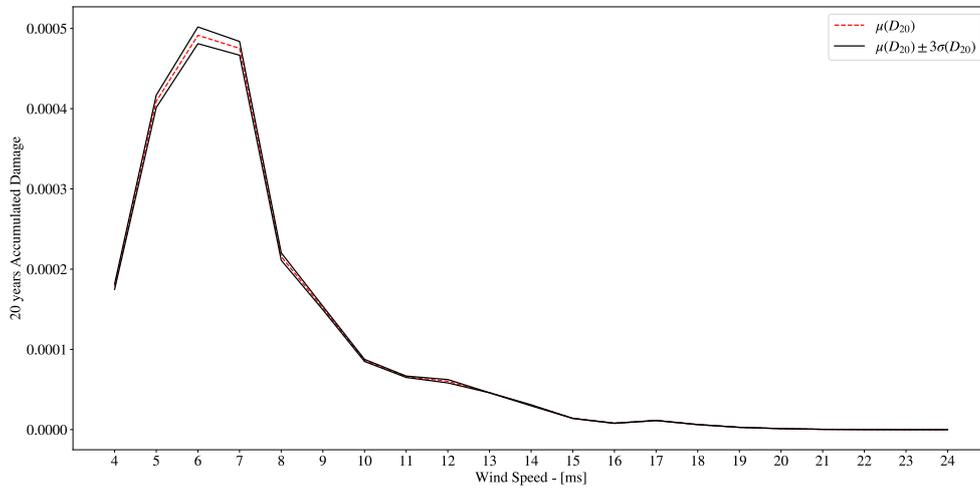


Figure B.4: 20 years damage distribution at node 580, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

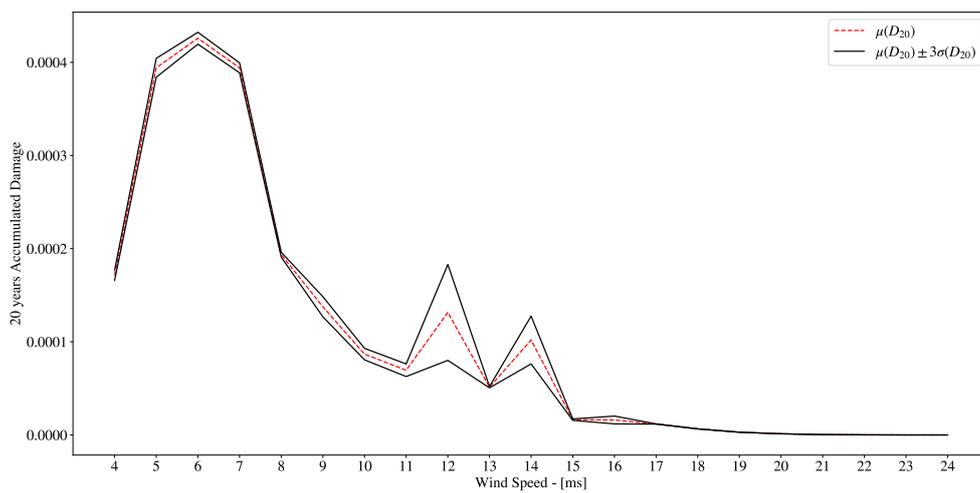


Figure B.5: 20 years damage distribution at node 567, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

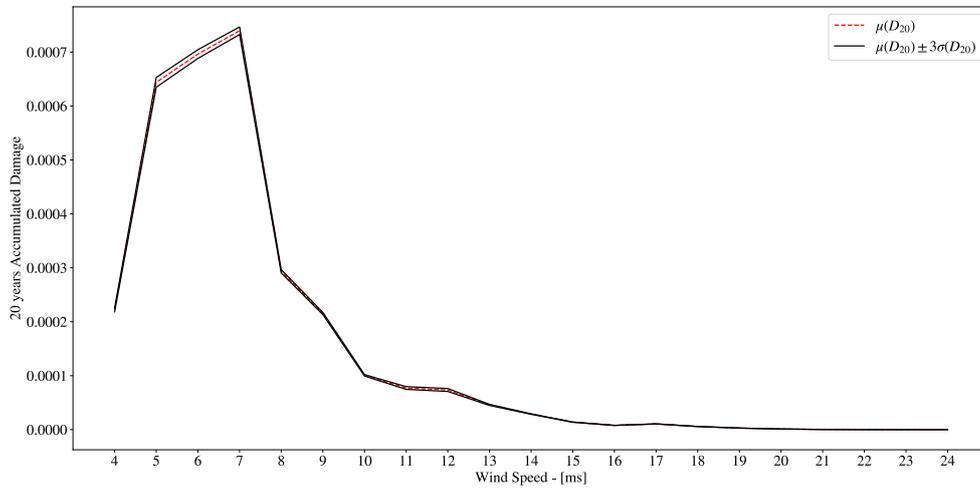


Figure B.6: 20 years damage distribution at node 549, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

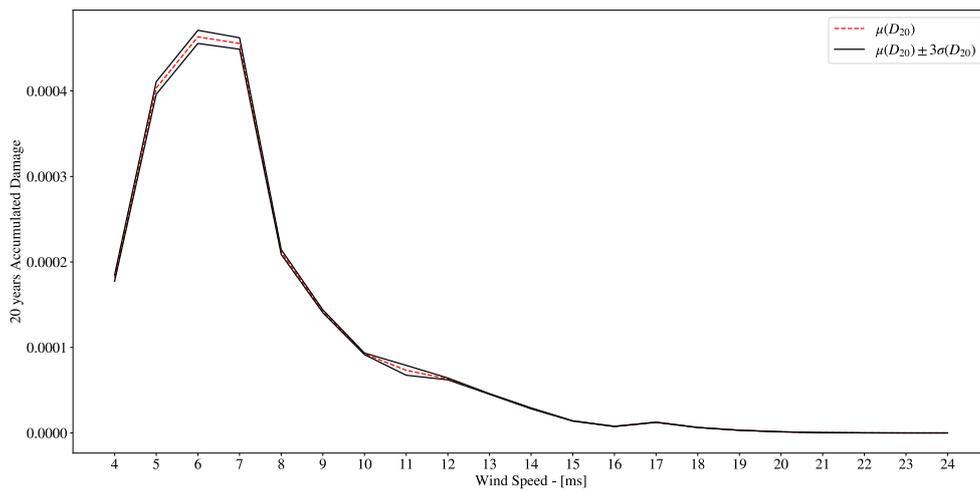


Figure B.7: 20 years damage distribution at node 531, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

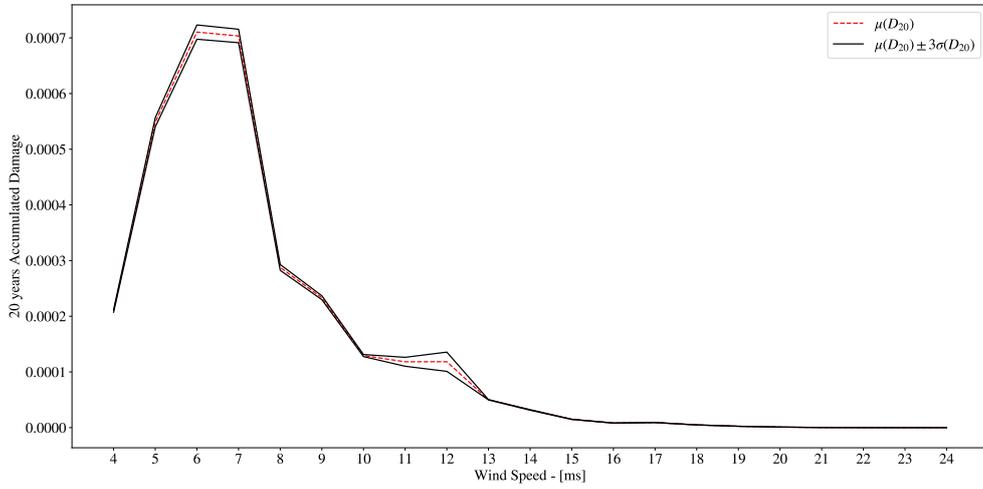


Figure B.8: 20 years damage distribution at node 516, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

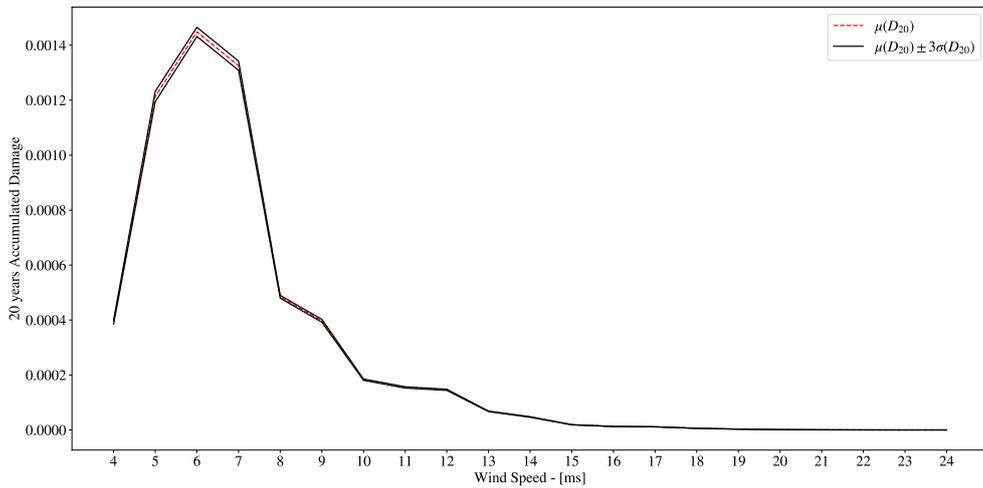


Figure B.9: 20 years damage distribution at node 506, section 1 of the blade as a function of the wind speed at first layer (P2B), inferior level by using LAM PD.

## B.2 LAM PD 20 years Damage QQ1 Layer: Inferior Level

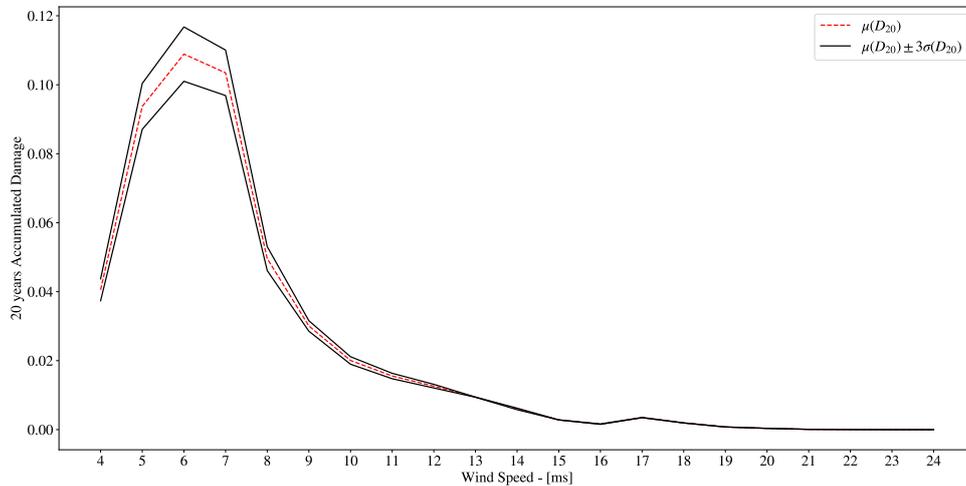


Figure B.10: 20 years damage distribution at node 603, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

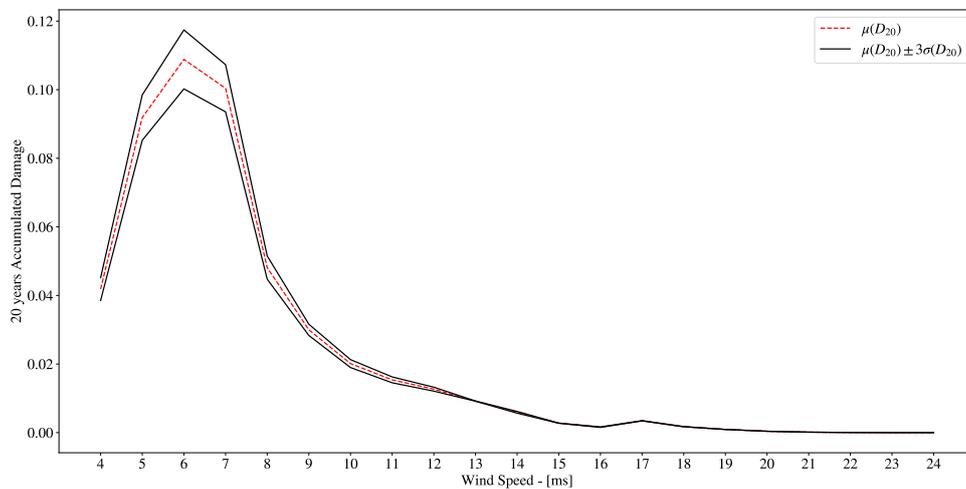


Figure B.11: 20 years damage distribution at node 595, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

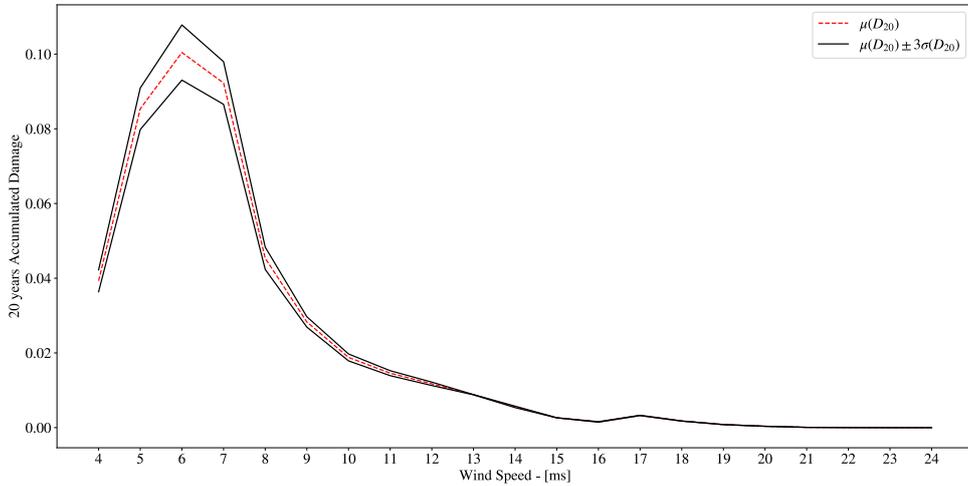


Figure B.12: 20 years damage distribution at node 586, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

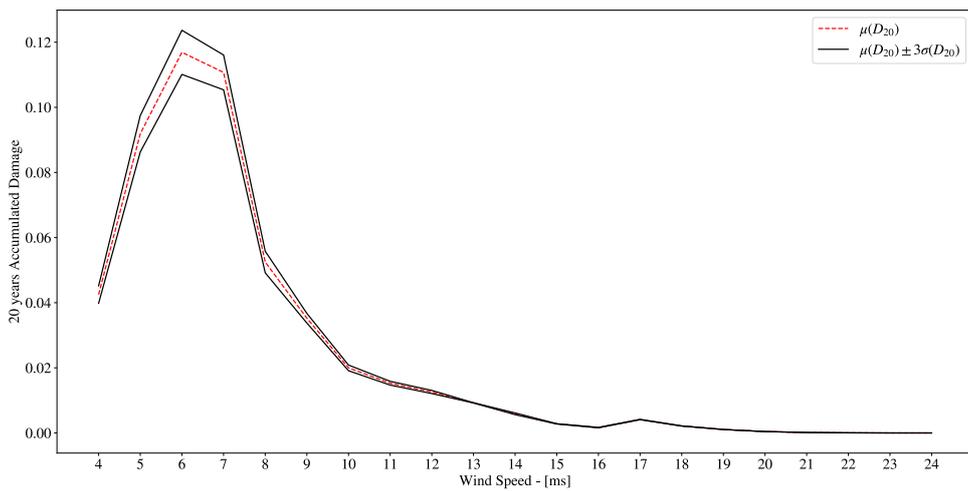


Figure B.13: 20 years damage distribution at node 580, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

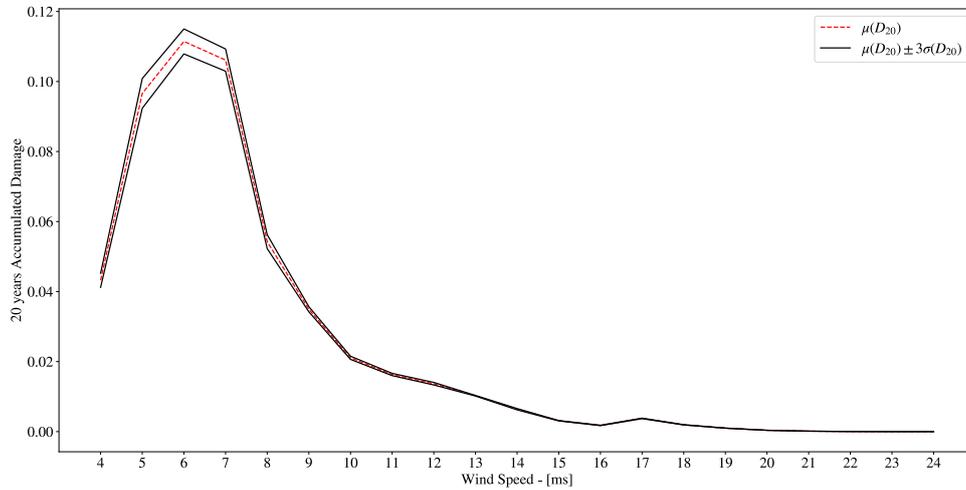


Figure B.14: 20 years damage distribution at node 567, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

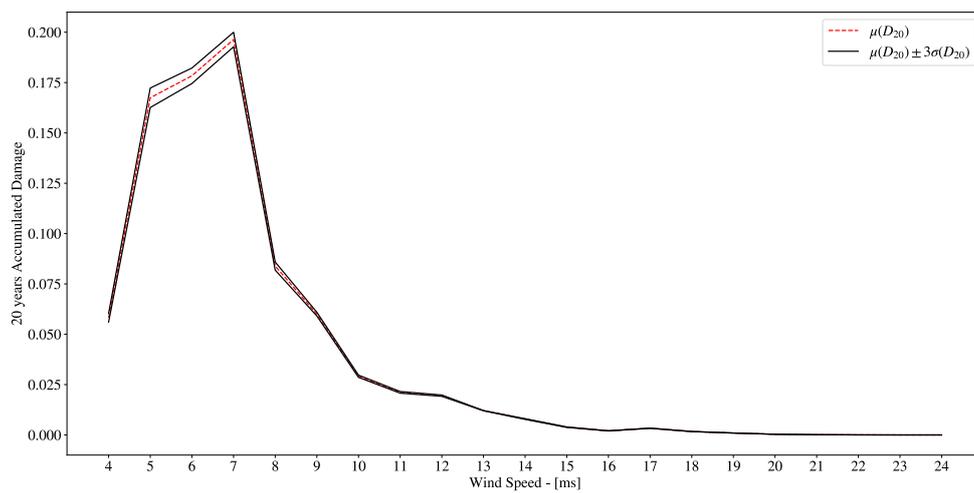


Figure B.15: 20 years damage distribution at node 549, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

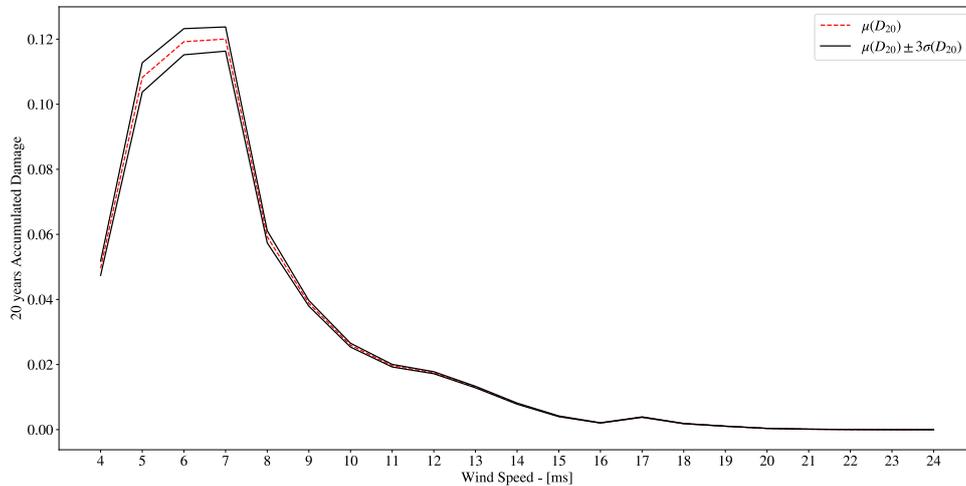


Figure B.16: 20 years damage distribution at node 531, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

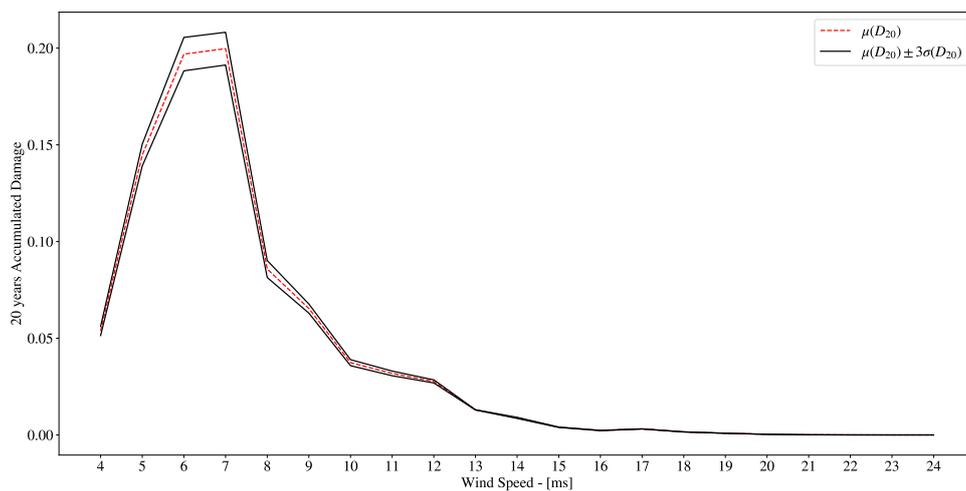


Figure B.17: 20 years damage distribution at node 516, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

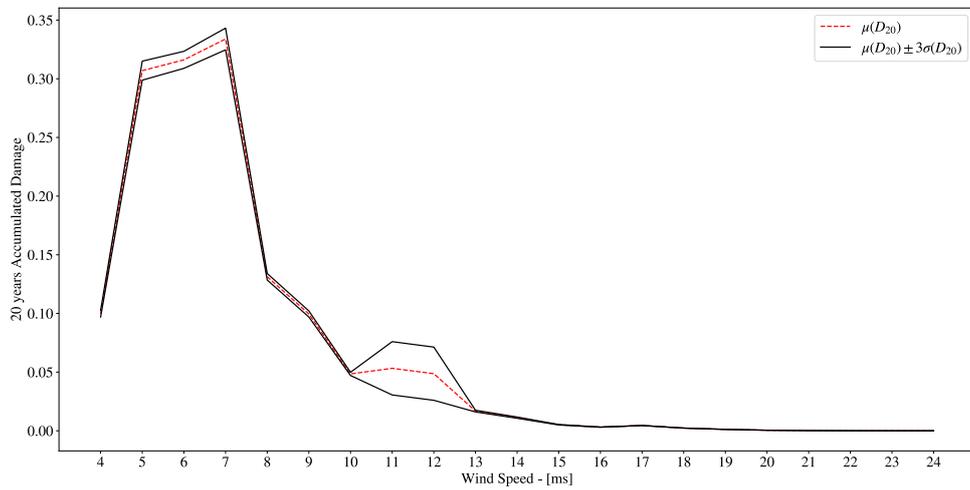


Figure B.18: 20 years damage distribution at node 506, section 1 of the blade as a function of the wind speed at second layer (QQ1), inferior level by using LAM PD.

# Bibliography

- [Arfken and Weber, 1999] Arfken, G. B. and Weber, H. J. (1999). Mathematical methods for physicists.
- [ASTM, 2003] ASTM, A. E. (2003). 1049—standard practices for cycle counting in fatigue analysis. *West Conshohocken (PA)*.
- [Bak et al., 2013] 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. (2013). The dtu 10-mw reference wind turbine. In *Danish Wind Power Research 2013*.
- [Bauchau et al., 2001] Bauchau, O. A., Bottasso, C. L., and Nikishkov, Y. G. (2001). Modeling rotorcraft dynamics with finite element multibody procedures. *Mathematical and Computer Modelling*, 33(10-11):1113–1137.
- [Berggreen et al., 2007] Berggreen, C., Branner, K., Jensen, J. F., and Schultz, J. P. (2007). Application and analysis of sandwich elements in the primary structure of large wind turbine blades. *Journal of Sandwich Structures & Materials*, 9(6):525–552.
- [Bergstra and Bengio, 2012] Bergstra, J. and Bengio, Y. (2012). Random search for hyperparameter optimization. *The Journal of Machine Learning Research*, 13(1):281–305.
- [Bettebghor et al., 2011] Bettebghor, D., Bartoli, N., Grihon, S., Morlier, J., and Samuelides, M. (2011). Surrogate modeling approximation using a mixture of experts based on em joint estimation. *Structural and multidisciplinary optimization*, 43(2):243–259.
- [Bir, 2005] Bir, G. (2005). User’s guide to bmodes (software for computing rotating beam-coupled modes). Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States).
- [Bir, 2006] Bir, G. S. (2006). User’s guide to precomp (pre-processor for computing composite blade properties). Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States).
- [Bitsche, 2012] Bitsche, R. (2012). Airfoil2becas: A preprocessor for the cross-section analysis software becas. *Technical University of Denmark*.
- [Blasques, 2012] Blasques, J. P. A. A. (2012). User’s manual for becas: A cross section analysis tool for anisotropic and inhomogeneous beam sections of arbitrary geometry.
- [Bottasso, 2006] Bottasso, C. (2006). Croce a. *Cp-Lambda: User’s Manual*, Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano, 2016.
- [Bottasso et al., 2014] Bottasso, C. L., Campagnolo, F., Croce, A., Dilli, S., Gualdoni, F., and Nielsen, M. B. (2014). Structural optimization of wind turbine rotor blades by multilevel sectional/multibody/3d-fem analysis. *Multibody System Dynamics*, 32(1):87–116.

- [Branner et al., 2007] Branner, K., Berring, P., Berggreen, C., and Knudsen, H. W. (2007). Torsional performance of wind turbine blades—part ii: Numerical validation. In *16th International Conference on Composite Materials, Anonymous*, pages 8–13.
- [Branner et al., 2012] Branner, K., Blasques, J. P. A. A., Kim, T., Fedorov, V., Berring, P., Bitsche, R., and Berggreen, C. (2012). Anisotropic beam model for analysis and design of passive controlled wind turbine blades.
- [Campolongo et al., 2007] Campolongo, F., Cariboni, J., and Saltelli, A. (2007). An effective screening design for sensitivity analysis of large models. *Environmental modelling & software*, 22(10):1509–1518.
- [Caous et al., 2018] Caous, D., Lavauzelle, N., Valette, J., and Wahl, J.-C. (2018). Load application method for shell finite element model of wind turbine blade. *Wind Engineering*.
- [Caous and Valette, 2014] Caous, D. and Valette, J. (2014). Methodology for g1 blade assessment. Technical report, TENSYL, La Rochelle.
- [Chamis, 1983] Chamis, C. C. (1983). Simplified composite micromechanics equations for hygral, thermal and mechanical properties.
- [Chen et al., 2013] Chen, J., Wang, Q., Shen, W. Z., Pang, X., Li, S., and Guo, X. (2013). Structural optimization study of composite wind turbine blade. *Materials & Design*, 46:247–255.
- [Chen et al., 2018] Chen, Z., Chen, F., Lai, R., Zhang, X., and Lu, C.-T. (2018). Rational neural networks for approximating jump discontinuities of graph convolution operator. *arXiv preprint arXiv:1808.10073*.
- [Clevert et al., 2015] Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.
- [Commission et al., 2019] Commission, I. E. et al. (2019). Iec 61400-1. *Wind Turbines—Part, 1*.
- [Cukier et al., 1973] Cukier, R., Fortuin, C., Shuler, K. E., Petschek, A., and Schaibly, J. (1973). Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. i theory. *The Journal of chemical physics*, 59(8):3873–3878.
- [Cukier et al., 1978] Cukier, R., Levine, H., and Shuler, K. (1978). Nonlinear sensitivity analysis of multiparameter model systems. *Journal of computational physics*, 26(1):1–42.
- [Cukier et al., 1975] Cukier, R., Schaibly, J., and Shuler, K. E. (1975). Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. iii. analysis of the approximations. *The Journal of Chemical Physics*, 63(3):1140–1149.
- [de France, ] de France, E. Finite element *code\_aster*, analysis of structures and thermomechanics for studies and research, year = 1989–2017. Open source on [www.code-aster.org](http://www.code-aster.org).

- [Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.
- [Dervilis, 2013] Dervilis, N. (2013). *A machine learning approach to structural health monitoring with a view towards wind turbines*. PhD thesis, University of Sheffield.
- [Devroye, 2006] Devroye, L. (2006). Nonuniform random variate generation. *Handbooks in operations research and management science*, 13:83–121.
- [Dimitrov et al., 2015] Dimitrov, N., Natarajan, A., and Kelly, M. (2015). Model of wind shear conditional on turbulence and its impact on wind turbine loads. *Wind Energy*, 18(11):1917–1931.
- [Dimitrov, 2013] Dimitrov, N. K. (2013). Structural reliability of wind turbine blades: Design methods and evaluation.
- [Drela, 1989] Drela, M. (1989). Xfoil: An analysis and design system for low reynolds number airfoils. In *Low Reynolds number aerodynamics*, pages 1–12. Springer.
- [Duchi et al., 2011] Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).
- [E73910, 2015] E73910, A. (2015). Standard practice for statistical analysis of linear or linearized stress-life (sn) and strain-life ( $\epsilon$ -n) fatigue data.
- [Echeverría et al., 2017] Echeverría, F., Mallor, F., and San Miguel, U. (2017). Global sensitivity analysis of the blade geometry variables on the wind turbine performance. *Wind Energy*, 20(9):1601–1616.
- [Elman, 1990] Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.
- [Engie, 2019] Engie (2019). La haute borne data (2017-2020). <https://opendata-renewables.engie.com/explore/index>. Accessed October 22, 2019.
- [Forcier and Joncas, 2012] Forcier, L.-C. and Joncas, S. (2012). Development of a structural optimization strategy for the design of next generation large thermoplastic wind turbine blades. *Structural and Multidisciplinary Optimization*, 45(6):889–906.
- [Geng et al., 2018] Geng, F., Kalkman, I., Suiker, A., and Blocken, B. (2018). Sensitivity analysis of airfoil aerodynamics during pitching motion at a reynolds number of  $1.35 \times 10^5$ . *Journal of Wind Engineering and Industrial Aerodynamics*, 183:315–332.
- [Géron, 2019] Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O’Reilly Media.
- [Giavotto et al., 1983] Giavotto, V., Borri, M., Mantegazza, P., Ghiringhelli, G., Carmaschi, V., Maffioli, G., and Mussi, F. (1983). Anisotropic beam theory and applications. *Computers & Structures*, 16(1-4):403–413.

- [GL, 2015] GL, L. B. D. (2015). Qualification of innovative floating substructures for 10mw wind turbines and water depths greater than 50m.
- [Glorot and Bengio, 2010] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.
- [Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.
- [Griffith and Ashwill, 2011] Griffith, D. T. and Ashwill, T. D. (2011). The sandia 100-meter all-glass baseline wind turbine blade: Snl100-00. *Sandia National Laboratories, Albuquerque, Report No. SAND2011-3779*, page 67.
- [Grujicic et al., 2010] Grujicic, M., Arakere, G., Subramanian, E., Sellappan, V., Vallejo, A., and Ozen, M. (2010). Structural-response analysis, fatigue-life prediction, and material selection for 1 mw horizontal-axis wind-turbine blades. *Journal of materials engineering and performance*, 19(6):790–801.
- [Guideline and Lloyd, 2010] Guideline, G. and Lloyd, G. (2010). Guideline for the certification of wind turbines. *Hamburg: Germanischer Lloyd Wind Energie Gmb H*.
- [Hansen and Henriksen, 2013] Hansen, M. H. and Henriksen, L. C. (2013). Basic dtu wind energy controller.
- [Haselbach et al., 2016] Haselbach, P. U., Bitsche, R., and Branner, K. (2016). The effect of delaminations on local buckling in wind turbine blades. *Renewable Energy*, 85:295–305.
- [Hashin, 1980] Hashin, Z. (1980). Failure criteria for unidirectional fiber composites.
- [Hashin, 1981] Hashin, Z. (1981). Fatigue failure criteria for unidirectional fiber composites.
- [He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.
- [Head et al., 2018] Head, T., MechCoder, G. L., Shcherbatyi, I., et al. (2018). scikit-optimize/scikit-optimize: v0. 5.2.
- [Hebb, 2005] Hebb, D. O. (2005). *The organization of behavior: A neuropsychological theory*. Psychology Press.
- [Hinton et al., 1986] Hinton, G. E., Sejnowski, T. J., et al. (1986). Learning and relearning in boltzmann machines. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1(282-317):2.
- [Hinton et al., 2012] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

- [Højstrup, 1982] Højstrup, J. (1982). Velocity spectra in the unstable planetary boundary layer. *Journal of the Atmospheric Sciences*, 39(10):2239–2248.
- [Homma and Saltelli, 1996] Homma, T. and Saltelli, A. (1996). Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering & System Safety*, 52(1):1–17.
- [Hopfield, 1982] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558.
- [Hu et al., 2016a] Hu, W., Choi, K., and Cho, H. (2016a). Reliability-based design optimization of wind turbine blades for fatigue life under dynamic wind load uncertainty. *Structural and Multidisciplinary Optimization*, 54(4):953–970.
- [Hu et al., 2016b] Hu, W., Choi, K., Zhupanska, O., and Buchholz, J. H. (2016b). Integrating variable wind load, aerodynamic, and structural analyses towards accurate fatigue life prediction in composite wind turbine blades. *Structural and Multidisciplinary Optimization*, 53(3):375–394.
- [Hu et al., 2012] Hu, W., Han, I., Park, S.-C., and Choi, D.-H. (2012). Multi-objective structural optimization of a hawt composite blade based on ultimate limit state analysis. *Journal of mechanical science and technology*, 26(1):129–135.
- [Hübler et al., 2017] Hübler, C., Gebhardt, C. G., and Rolfes, R. (2017). Hierarchical four-step global sensitivity analysis of offshore wind turbines based on aeroelastic time domain simulations. *Renewable Energy*, 111:878–891.
- [Jones et al., 01 ] Jones, E., Oliphant, T., Peterson, P., et al. (2001–). SciPy: Open source scientific tools for Python.
- [Jones, 1998] Jones, R. M. (1998). *Mechanics of composite materials*. CRC press.
- [Jonkman and Buhl Jr, 2006] Jonkman, B. J. and Buhl Jr, M. L. (2006). Turbsim user’s guide. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States).
- [Jonkman et al., 2009] Jonkman, J., Butterfield, S., Musial, W., and Scott, G. (2009). Definition of a 5-mw reference wind turbine for offshore system development. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States).
- [Jonkman and Buhl Jr, 2005] Jonkman, J. M. and Buhl Jr, M. L. (2005). Fast user’s guide—updated august 2005. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States).
- [Kaltschmitt et al., 2007] Kaltschmitt, M., Streicher, W., and Wiese, A. (2007). *Renewable energy: technology, economics and environment*. Springer Science & Business Media.

- [Kelley et al., 2002] Kelley, N., Hand, M., Larwood, S., and McKenna, E. (2002). The nrel large-scale turbine inflow and response experiment: preliminary results. In *Wind Energy Symposium*, volume 7476, pages 412–426.
- [Kelley et al., 2004] Kelley, N., Shirazi, M., Jager, D., Wilde, S., Adams, J., Buhl, M., Sullivan, P., and Patton, E. (2004). Lamar low-level jet program interim report. Technical report, National Renewable Energy Lab., Golden, CO.(US).
- [Khodayar and Teshnehlab, 2015] Khodayar, M. and Teshnehlab, M. (2015). Robust deep neural network for wind speed prediction. In *2015 4th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, pages 1–5. IEEE.
- [Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [Knill, 2005] Knill, T. J. (2005). The application of aeroelastic analysis output load distributions to finite element models of wind. *Wind Engineering*, 29(2):153–168.
- [Larsen and Hansen, 2007] Larsen, T. J. and Hansen, A. M. (2007). How 2 hawc2, the user’s manual.
- [Li et al., 2019] Li, K., Kou, J., and Zhang, W. (2019). Deep neural network for unsteady aerodynamic and aeroelastic modeling across multiple mach numbers. *Nonlinear Dynamics*, 96(3):2157–2177.
- [Lindgaard and Lund, 2010] Lindgaard, E. and Lund, E. (2010). Nonlinear buckling optimization of composite structures. *Computer methods in applied mechanics and engineering*, 199(37-40):2319–2330.
- [Liu et al., 2018] Liu, H., Mi, X.-w., and Li, Y.-f. (2018). Wind speed forecasting method based on deep learning strategy using empirical wavelet transform, long short term memory neural network and elman neural network. *Energy conversion and management*, 156:498–514.
- [Liu and Mahadevan, 2005] Liu, Y. and Mahadevan, S. (2005). Probabilistic fatigue life prediction of multidirectional composite laminates. *Composite Structures*, 69(1):11–19.
- [Liu and Mahadevan, 2007] Liu, Y. and Mahadevan, S. (2007). A unified multiaxial fatigue damage model for isotropic and anisotropic materials. *International Journal of Fatigue*, 29(2):347–359.
- [Liu and Zhang, 2016] Liu, Y. and Zhang, H. (2016). An empirical study on machine learning models for wind power predictions. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 758–763. IEEE.
- [Louppe, 2017] Louppe, G. (2017). Bayesian optimisation with scikit-optimize.
- [Lund and Stegmann, 2005] Lund, E. and Stegmann, J. (2005). On structural optimization of composite shell structures using a discrete constitutive parametrization. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 8(1):109–124.

- [Mandell and Samborsky, 2016] Mandell, J. and Samborsky, D. (2016). Snl/msu/doe composite material fatigue database mechanical properties of composite materials for wind turbine blades version 25.0. *Montana State University, Bozeman*.
- [Mandell et al., 2010] Mandell, J. F., Samborsky, D. D., Agastra, P., Sears, A. T., and Wilson, T. J. (2010). Analysis of snl/msu/doe fatigue database trends for wind turbine blade materials. *Sandia Contractor Report, To Be Published (draft available on [www.coe.montana.edu/composites/](http://www.coe.montana.edu/composites/))*.
- [Massey Jr, 1951] Massey Jr, F. J. (1951). The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78.
- [McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.
- [McKay et al., 2014] McKay, P. M., Carriveau, R., Ting, D. S.-K., and Johrendt, J. L. (2014). Global sensitivity analysis of wind turbine power output. *Wind Energy*, 17(7):983–995.
- [McKittrick et al., 2001] McKittrick, L. R., Cairns, D. S., Mandell, J., Combs, D. C., Rabern, D. A., and Van Luchene, R. D. (2001). Analysis of a composite blade design for the aoc 15/50 wind turbine using a finite element model. *SAND2001-1441. Sandia National Laboratories Contractor Report*.
- [McLachlan and Basford, 1988] McLachlan, G. and Basford, K. (1988). Mixture models marcel dekker. *New York*.
- [Miner, 1945] Miner, M. (1945). Cumulative damage in fatigue. *Journal of Applied Mechanics*, 12(3):159–164.
- [Morató et al., 2019] Morató, A., Sriramula, S., and Krishnan, N. (2019). Kriging models for aero-elastic simulations and reliability analysis of offshore wind turbine support structures. *Ships and Offshore Structures*, 14(6):545–558.
- [Moriarty and Hansen, 2005] Moriarty, P. J. and Hansen, A. C. (2005). Aerodyn theory manual. Technical report, National Renewable Energy Lab., Golden, CO (US).
- [Morris, 1991] Morris, M. D. (1991). Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2):161–174.
- [Nastran, 2004] Nastran, M. (2004). Basic dynamic analysis user’s guide. *MSC. Software Corporation. US A*.
- [Nesterov, 1983] Nesterov, Y. (1983). A method for unconstrained convex minimization problem with the rate of convergence  $O(1/k^2)$ . In *Doklady an ussr*, volume 269, pages 543–547.
- [Ng, 2004] Ng, A. Y. (2004). Feature selection,  $l_1$  vs.  $l_2$  regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78.

- [Niutta et al., 2018] Niutta, C. B., Wehrle, E., Duddeck, F., and Belingardi, G. (2018). Surrogate modeling in design optimization of structures with discontinuous responses. *Structural and Multidisciplinary Optimization*, 57(5):1857–1869.
- [Nussbaumer et al., 2012] Nussbaumer, A., Borges, L., and Davaine, L. (2012). *Fatigue design of steel and composite structures: Eurocode 3: Design of steel structures, part 1-9 fatigue; Eurocode 4: Design of composite steel and concrete structures*. John Wiley & Sons.
- [Olesen et al., 1984] Olesen, H. R., Larsen, S. E., and Højstrup, J. (1984). Modelling velocity spectra in the lower part of the planetary boundary layer. *Boundary-Layer Meteorology*, 29(3):285–312.
- [París and Jackson, 2001] París, F. and Jackson, K. E. (2001). A study of failure criteria of fibrous composite materials.
- [Pedersen, 2016] Pedersen, M. E. H. (2016). Tensorflow tutorials.
- [Peng et al., 2016] Peng, X., Xiong, L., Wen, J., Xu, Y., Fan, W., Feng, S., and Wang, B. (2016). A very short term wind power prediction approach based on multilayer restricted boltzmann machine. In *2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, pages 2409–2413. IEEE.
- [Petroni et al., 2011] Petroni, G., de Nicola, C., Quagliarella, D., Witteveen, J., and Iaccarino, G. (2011). Wind turbine performance analysis under uncertainty. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, page 544.
- [Philippidis and Vassilopoulos, 2002] Philippidis, T. P. and Vassilopoulos, A. P. (2002). Complex stress state effect on fatigue life of grp laminates. part ii, theoretical formulation. *International journal of fatigue*, 24(8):825–830.
- [Polyak, 1964] Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17.
- [Puck et al., 2002] Puck, A., Kopp, J., and Knops, M. (2002). Guidelines for the determination of the parameters in puck’s action plane strength criterion. *Composites Science and Technology*, 62(3):371–378.
- [Rosenblatt, 1957] Rosenblatt, F. (1957). *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory.
- [Roshko, 1954] Roshko, A. (1954). On the drag and shedding frequency of two-dimensional bluff bodies.
- [Roshko, 1961] Roshko, A. (1961). Experiments on the flow past a circular cylinder at very high reynolds number. *Journal of fluid mechanics*, 10(3):345–356.
- [Rubinstein and Kroese, 1981] Rubinstein, R. Y. and Kroese, D. (1981). Simulation and the monte carlo method. john wiley&sons. *Inc. Publication*.

- [Rumelhart et al., 1985] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.
- [Salehi and Burgueno, 2018] Salehi, H. and Burgueno, R. (2018). Emerging artificial intelligence methods in structural engineering. *Engineering structures*, 171:170–189.
- [Saltelli, 2002] Saltelli, A. (2002). Sensitivity analysis for importance assessment. *Risk analysis*, 22(3):579–590.
- [Saltelli et al., 2010] Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., and Tarantola, S. (2010). Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. *Computer Physics Communications*, 181(2):259–270.
- [Saltelli et al., 2008] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S. (2008). *Global sensitivity analysis: the primer*. John Wiley & Sons.
- [Santner et al., 2003] Santner, T. J., Williams, B. J., Notz, W., and Williams, B. J. (2003). *The design and analysis of computer experiments*, volume 1. Springer.
- [Sarkani et al., 2001] Sarkani, S., Michaelov, G., Kihl, D. P., and Bonanni, D. L. (2001). Comparative study of nonlinear damage accumulation models in stochastic fatigue of frp laminates. *Journal of Structural Engineering*, 127(3):314–322.
- [Seabold and Perktold, 2010] Seabold, S. and Perktold, J. (2010). statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.
- [Sergio and Ludermir, 2015] Sergio, A. T. and Ludermir, T. B. (2015). Deep learning for wind speed forecasting in northeastern region of brazil. In *2015 Brazilian Conference on Intelligent Systems (BRACIS)*, pages 322–327. IEEE.
- [Sermanet et al., 2012] Sermanet, P., Chintala, S., and LeCun, Y. (2012). Convolutional neural networks applied to house numbers digit classification. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3288–3291. IEEE.
- [Shinozuka and Jan, 1972] Shinozuka, M. and Jan, C.-M. (1972). Digital simulation of random processes and its applications. *Journal of sound and vibration*, 25(1):111–128.
- [Shokrieh and Rafiee, 2006] Shokrieh, M. M. and Rafiee, R. (2006). Simulation of fatigue failure in a full composite wind turbine blade. *Composite structures*, 74(3):332–342.
- [Sierra-Pérez et al., 2016] Sierra-Pérez, J., Torres-Arredondo, M. A., and Güemes, A. (2016). Damage and nonlinearities detection in wind turbine blades based on strain field pattern recognition. fbg, obr and strain gauges comparison. *Composite Structures*, 135:156–166.
- [Slot et al., 2020] Slot, R. M., Sørensen, J. D., Sudret, B., Svenningsen, L., and Thøgersen, M. L. (2020). Surrogate model uncertainty in wind turbine reliability assessment. *Renewable Energy*, 151:1150–1162.

- [Sobol, 1993] Sobol, I. M. (1993). Sensitivity estimates for nonlinear mathematical models. *Mathematical modelling and computational experiments*, 1(4):407–414.
- [Sørensen et al., 2014] Sørensen, S. N., Sørensen, R., and Lund, E. (2014). Dmto—a method for discrete material and thickness optimization of laminated composite structures. *Structural and Multidisciplinary Optimization*, 50(1):25–47.
- [Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.
- [Srivastava et al., 2019] Srivastava, S., Damodaran, M., and Khoo, B. C. (2019). Machine learning surrogates for predicting response of an aero-structural-sloshing system. *arXiv preprint arXiv:1911.10043*.
- [Standard et al., 2005] Standard, D. et al. (2005). Wind turbines-part 1: Design requirements. *IEC61400-1, Copenhagen*.
- [Stephens, 1974] Stephens, M. A. (1974). Edf statistics for goodness of fit and some comparisons. *Journal of the American statistical Association*, 69(347):730–737.
- [Stephens, 1976] Stephens, M. A. (1976). Asymptotic results for goodness-of-fit statistics with unknown parameters. *The Annals of Statistics*, pages 357–369.
- [Stigler, 1974] Stigler, S. M. (1974). Gergonne’s 1815 paper on the design and analysis of polynomial regression experiments. *Historia Mathematica*, 1(4):431–439.
- [Sudret, 2014] Sudret, B. (2014). Polynomial chaos expansions and stochastic finite element methods. *Risk and reliability in geotechnical engineering*, pages 265–300.
- [Teixeira et al., 2017] Teixeira, R., O’Connor, A., Nogal, M., Krishnan, N., and Nichols, J. (2017). Analysis of the design of experiments of offshore wind turbine fatigue reliability design with kriging surfaces. *Procedia Structural Integrity*, 5:951–958.
- [Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.
- [Tieleman and Hinton, 2012] Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31.
- [Touma, 1977] Touma, J. S. (1977). Dependence of the wind profile power law on stability for various locations. *Journal of the Air Pollution Control Association*, 27(9):863–866.
- [Tran and Smith, 2018] Tran, T. T. and Smith, A. D. (2018). Incorporating performance-based global sensitivity and uncertainty analysis into lcoe calculations for emerging renewable energy technologies. *Applied energy*, 216:157–171.
- [Van der Vaart, 2000] Van der Vaart, A. W. (2000). *Asymptotic statistics*, volume 3. Cambridge university press.

- [Vapnik et al., 1997] Vapnik, V., Golowich, S. E., and Smola, A. J. (1997). Support vector method for function approximation, regression estimation and signal processing. In *Advances in neural information processing systems*, pages 281–287.
- [Vassilopoulos and Nijssen, 2010] Vassilopoulos, A. and Nijssen, R. (2010). Fatigue life prediction of composite materials under realistic loading conditions (variable amplitude loading). In *Fatigue life prediction of composites and composite structures*, pages 293–333. Elsevier.
- [Vassilopoulos et al., 2010] Vassilopoulos, A. P., Manshadi, B. D., and Keller, T. (2010). Influence of the constant life diagram formulation on the fatigue life prediction of composite materials. *International journal of fatigue*, 32(4):659–669.
- [Veers, 1988] Veers, P. S. (1988). Three-dimensional wind simulation. Technical report, Sandia National Labs., Albuquerque, NM (USA).
- [Velarde et al., 2019] Velarde, J., Kramhøft, C., and Sørensen, J. D. (2019). Global sensitivity analysis of offshore wind turbine foundation fatigue loads. *Renewable Energy*, 140:177–189.
- [Wang et al., 2017] Wang, H.-z., Li, G.-q., Wang, G.-b., Peng, J.-c., Jiang, H., and Liu, Y.-t. (2017). Deep learning based ensemble approach for probabilistic wind power forecasting. *Applied energy*, 188:56–70.
- [Weisstein, 2002] Weisstein, E. W. (2002). Discrete fourier transform. <https://mathworld.wolfram.com/>.
- [Wöhler, 1870] Wöhler, A. (1870). *Über die festigkeitsversuche mit eisen und stahl*. Ernst & Korn.
- [Xu et al., 2015] Xu, B., Wang, N., Chen, T., and Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.
- [Yu et al., 2002] Yu, W., Hodges, D. H., Volovoi, V., and Cesnik, C. E. (2002). On timoshenko-like modeling of initially curved and twisted composite beams. *International Journal of Solids and Structures*, 39(19):5101–5121.
- [Zhao et al., 2011] Zhao, L., Choi, K., and Lee, I. (2011). Metamodeling method using dynamic kriging for design optimization. *AIAA journal*, 49(9):2034–2046.
- [Ziegler et al., 2015] Ziegler, L., Voormeeren, S., Schafhirt, S., and Muskulus, M. (2015). Sensitivity of wave fatigue loads on offshore wind turbines under varying site conditions. *Energy Procedia*, 80:193–200.