A global sensitivity analysis of model uncertainty in aeroelastic wind turbine models

P. Kumar[1], B. Sanderse[1], K. Boorsma[2], M. Caboni[2]

E-mail: b.sanderse@cwi.nl
[1] Centrum Wiskunde & Informatica (CWI), Amsterdam, The Netherlands
[2] TNO, Westerduinweg 3, Petten, The Netherlands

Abstract. A global sensitivity analysis of model parameters associated with the Blade Element Momentum (BEM) models is performed. The Sobol indices based on a sparse polynomial expansion are used as a measure of global sensitivities. The sensitivity analysis workflow is developed using the uncertainty quantification toolbox `UQLab` that is integrated with the ECN's `Aero-Module` aeroelastic code. Sensitivity studies are performed on the NM80 wind turbine model from the DANAERO project.

1. Introduction

Aeroelastic models such as the Blade Element Momentum (BEM) models [1] play a critical role in the design, development, and optimization of modern wind turbines. A large number of BEM models have been developed to predict turbine responses such as the structural loads and power output [2].

As a consequence of the strong model assumptions at the basis of BEM theory, the results from BEM codes can be subject to significant inaccuracies or uncertainties. For example, the effect of sheared inflow [3] is not naturally accounted for in the theory and needs to be incorporated via correction terms. Other major model uncertainties in BEM models are for example the time constant in dynamic stall models, the wake correction factor, the tip loss model parameter, and the lift- and drag-polars used to compute local aerodynamic forces. Especially for increasing turbine sizes, these model parameters are foreseen to be not sufficiently accurate [4]. In other words, the uncertainty associated with the output of currently employed BEM models is rather large.

Recently, several papers have addressed the uncertainty in BEM model output by performing forward uncertainty propagation or sensitivity studies, e.g. [5, 6, 7, 8, 9]. In these studies, the focus in mainly on uncertainties in the external conditions (wind parameters) and/or uncertainty in the turbine specification (geometric parameters). Apart from understanding how uncertainty in the output is related to the different uncertain inputs, such sensitivity studies are very useful to reduce the number of parameters as needed for example in design optimization [5]. However, the uncertainty in BEM model output as caused by uncertainty in the model formulation itself, e.g. through the values chosen for model parameters, has been given little attention (exceptions being the effect of aerodynamic properties studied in [10, 6]).

The goal of this work is to perform a systematic assessment of the uncertainty in model parameters in BEM models. We approach this by performing a global sensitivity analysis based

this is also what we do

on the Sobol expansion approach, which decomposes the total variance of the quantity of interest (model output) into contributions from individual parameters and their combinations, similar to [5, 7, 11]. We will employ the uncertainty quantification toolbox `UQLab` [12], which computes the Sobol indices based on a sparse polynomial chaos expansion.

The main novelty of the work lies in the study of the effect of model uncertainties rather than external conditions or geometric uncertainties. As such, the current sensitivity analysis is part of the so-called WindTrue project, in which the long-term goal is the development of calibrated BEM models, that possess a quantified level of uncertainty. Anticipating on this goal, we are using the NM80 wind turbine model from the Tjaereborg wind farm, for which extensive measurement data is available through the DanAero project [13] and IEA Task 29. The measurement data will enable the calibration step, and our sensitivity analysis will make clear which of the model parameters should be used in this calibration step.

This paper is structured as follows. First, in section 2 we give a short description of the BEM model and associated geometric and model uncertainties. In section 3 the parameterization of the uncertainties is described, and in section 4 the global sensitivity analysis methodology. Section 5 discusses the results of the sensitivity analysis for the NM80 turbine test case, and conclusions follow in section 6.

2. Aeroelastic model uncertainties

State-of-the-art wind-turbine models simulate the aeroelastic behavior of wind turbines by combining the concept of momentum conservation of the flow (BEM theory) with the equations of motion for the structure, possibly extended with the hydrodynamics of the sea and control algorithms [2]. In this work, we only concentrate on the first aspect, namely the prediction of flow and blade forces as given by the BEM method. The particular BEM code that we use is ECN's `Aero-Module` [14].

For the case of a rigid turbine, with a uniform inflow field, the main model uncertainties that arise in the BEM formulation are the following [15]:

- The use of (2D) *airfoil polars*; uncertainty arises not only because the actual flow along the blade is 3D, but also because the 2D polars itself can be inaccurate, either when obtained from measurements or 2D codes like `XFOIL`;
- The assumption of locally 2D flow along the blade is also invalid at the tip; this is typically corrected via *a tip loss model*;
- Three-dimensional flow effects also cause stall delay at the root; this is typically modeled via *a dynamic stall model* and associated time constant;
- Yawed inflow is not naturally included in BEM and typically incorporated via *a wake correction*.

Next to these model uncertainties we will consider the chord and twist distribution along the blade to be uncertain (geometric uncertainties).

3. Parametrization of uncertain inputs

The chord, twist, lift coefficient, and drag coefficient are functions of the radial distance along the blade, leading to a very high-dimensional number of uncertain parameters. In order to reduce this number, we parametrize the variation along the blade by using Non-Uniform Rational Basis Splines (NURBS) [16]. In this way only a limited number of control points is needed to approximate a large variety of curves, so that the resulting number of uncertain parameters is relatively small. NURBS have already been exploited in several aerodynamics applications, for instance, design optimization of wind-turbine blades, e.g. [17, 18], and to parametrize chord and twist curves [5]. In the following, we briefly outline the procedure used to parametrize a curve using NURBS.

3.1. NURBS representation

A NURBS curve $S(x)$ is expressed using a weighted sum of n basis functions (or B-splines):

$$S(x) = \sum_{i=1}^{n} c_i B_{i,p}(x), \tag{3.1}$$

where c_i is the weight of the i-th control point and $B_{i,p}(x)$ is the value of the B-spline corresponding to the i-th control point at x, with the subscript p denoting its polynomial degree. In our case, $S(x)$ represent for example the chord as a function of the radial location along the blade. A sequence of non-decreasing parameters $X = \{x_1, x_2, ..., x_m\}$, called the *knot vector*, determines the domain in which any control point is active. A control point c_i is said to be active at a location when the corresponding B-spline $B_{i,p}(x)$ is non-zero at x. B-splines are recursive in polynomial degree, thus we can derive quadratic B-splines using linear B-splines, cubic from quadratic B-splines and so on. Starting with a B-spline of degree 0 defined as

$$B_{i,0}(x) := \begin{cases} 1 & x_i \leq x < x_{i+1}, \quad i = 1, 2, ..., m, \\ 0 & \text{elsewhere}, \end{cases} \tag{3.2}$$

we can derive higher-order B-splines using the recurrence relation [19]:

$$B_{i,p}(x) := \frac{x - x_i}{x_{i+p} - x_i} B_{i,p-1}(x) + \frac{x_{i+p+1} - x}{x_{i+p+1} - x_{i+1}} B_{i+1,p-1}(x), \quad p \geq 1. \tag{3.3}$$

Note that with increasing p the interval in which a B-spline is non-zero also become larger. For a curve of polynomial degree p, there are $p + 1$ active control points at any location x. For example, for a linear curve ($p = 1$) only two adjacent control points are active at any location. Therefore, only a few nearby control points are active when computing the NURBS curve (3.1). These B-splines can be constructed efficiently using the de Boor's algorithm [19]. To compute high-order B-splines in (3.3), we need to pad the knot vector X by repeating the first and last elements p times each. In Fig. 1, we show linear, quadratic and cubic splines for $x \in [0, 1]$ with original knot vector $X = \{0, 1/3, 2/3, 1\}$, padded accordingly with the curve order, for e.g., for linear B-splines, we use the knot vector $X = \{0, 0, 1/3, 2/3, 1, 1\}$.

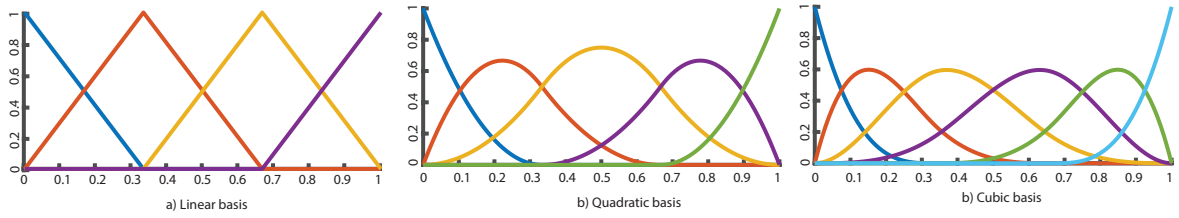


Figure 1: Examples of B-splines with original knot vector $X = \{0, 1/3, 2/3, 1\}$.

In order to parameterize a given curve, e.g. the blade reference chord, we choose a degree p and a number of knots $X = \{x_j\}_{j=1}^{n}$. The knots can be uniformly spaced or chosen heuristically (for advanced approaches for knot selection see[20, 21]). Next we sample $S_{ref}(x)$ at X and also compute $B_{i,p}(x_j)$, for $i, j = 1, 2, ..., n$. We then formulate a linear system and solve for the set of control points $\mathbf{c} = \{c_i\}_{i=1}^{n}$ as:

$$\mathbf{Bc} = \mathbf{S}_{ref}, \tag{3.4}$$

where $\mathbf{S}_{ref} \in \mathbb{R}^n$ is a vector containing sampled values of $S_{ref}(x)$ and $\mathbf{B} \in \mathbb{R}^{n \times n}$ is a matrix with the j-th row consisting of values of n B-splines sampled at location x_j. Once the control points are obtained, the approximate reference curve $S_n(x)$ follows from (3.1). Fig. 2 shows a parameterization of the chord curve using $n = 9$ control points and second-order ($p = 2$) B-splines.

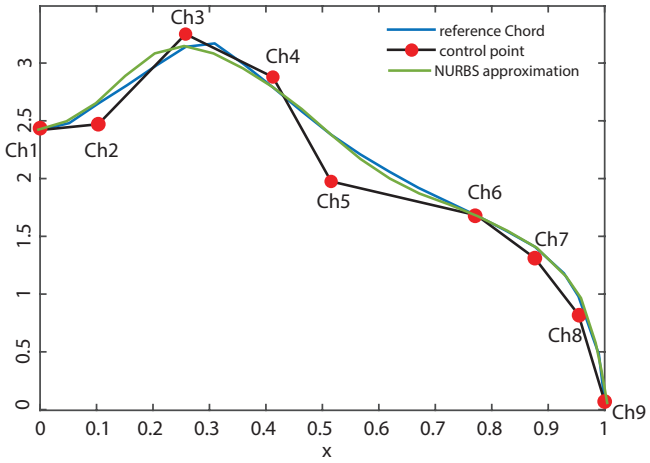


Figure 2: Parametrization of chord using 9 control points and second-order B-splines.

3.2. Perturbing the NURBS curves

Given the reference values of $\mathbf{c}$, it becomes straightforward to generate perturbed curves: we sample a uniform random variable to perturb the baseline control points and use these to compute a perturbed curve $\tilde{S}(x)$:

$$\tilde{S}(x) = \sum_{i=1}^{n} \tilde{c}_i B_{i,p}(x), \tag{3.5}$$

where

$$\tilde{c}_i = c_i(1 + \Delta_i), \tag{3.6}$$

and the perturbation is randomly sampled as $\Delta_i \sim \mathcal{U}[-0.05, 0.05]$.

3.3. Model uncertainties

The above framework is directly applicable to express the geometric uncertainty in twist- and chord distribution, with c denoting either the value of the twist control points or of the chord control points. For model uncertainties, i.e. lift- and drag polars,

Insert mathematical expression

4. Global sensitivity analysis

The objective of sensitivity analysis is to quantify the relative significance of individual inputs (or in combination) and how variations in input values affect the output of interest. In engineering, sensitivity analysis can be employed for several reasons: to determine the stability and robustness of a computational model for input parameters, for simplification of stochastic models by fixing the insensitive parameters, and to guide data acquisition campaigns and experimental design to refine the data on sensitive parameters. Sensitivity analysis techniques can be classified as local and global methods, see [22]. In a local sensitivity analysis, individual parameters are perturbed around their nominal values allowing for the description of output variability only in a small neighbourhood of nominal input values. Although, local approaches are widely employed due to their ease of implementation and low computational cost, they are unable to quantify the global behavior of nonlinearly parametrized models such as aeroelastic models. Global sensitivity approaches, on the other hand, consider the entire range of input values to compute output sensitivities. Therefore, a global sensitivity analysis is more suitable for aeroelastic models considered in this work.

4.1. Sobol analysis

We employ variance-based Sobol decomposition to perform global sensitivity analysis. This approach allows for quantification of the relative importance of the input parameters on a scale of $[0, 1]$ known as *Sobol indices*. In a Sobol analysis, we express the total variance of the output in term of contributions from individual parameters and their combinations. For the ease of exposition, let us consider a nonlinear model:

$$Y = f(\mathbf{z}), \tag{4.1}$$

where $\mathbf{z} = [z_1, z_2, ..., z_M] \in \mathcal{D}_\mathbf{z} \in \mathbb{R}^M$ is a vector of random inputs. For simplicity, we assume input parameters are uniformly distributed i.e. $z_i \sim \mathcal{U}(0, 1)$ and the support of input set is $\mathcal{D}_\mathbf{z} = [0, 1]^M$ where M is the total number input parameters. The Sobol decomposition is based on the hierarchical representation [22]:

$$f(z_1, z_2, ..., z_M) = f_0 + \sum_{i=1}^{M} f_i(z_i) + \sum_{1 \le i < j \le M} f_{ij}(z_i, z_j) + ... + f_{1,2,...,M}(z_1, z_2, ..., z_M), \tag{4.2}$$

where the zeroth-order function f_0 is the mean response of f, the first-order univariate functions $f_i(z_i)$ quantify independent contributions due to each random input, second-order functions $f_{i,j}(z_i, z_j)$ represent the effect of interaction between z_i and z_j on the response f. Higher-order terms are interpreted in a similar manner. Note that the (4.2) is only valid for independent input parameters.

Formally, the zeroth-, first- and second-order terms are defined as:

$$f_0 = \int_{\mathcal{D}_\mathbf{z}} f(\mathbf{z}) d\mathbf{z}, \tag{4.3}$$

$$f_i(z_i) = \int_{{\mathcal{D}_\mathbf{z}}^{M-1}} f(\mathbf{z}) d\mathbf{z}_{\sim\{i\}} - f_0, \tag{4.4}$$

$$f_{ij}(z_i, z_j) = \int_{{\mathcal{D}_\mathbf{z}}^{M-2}} f(\mathbf{z}) d\mathbf{z}_{\sim\{i,j\}} - f_i(z_i) - f_j(z_j) - f_0, \tag{4.5}$$

where ${\mathcal{D}_\mathbf{z}}^{M-1} = [0, 1]^{M-1}$ and the notation $\mathbf{z}_{\sim\{i,j\}}$ denotes the vector having all the components of $\mathbf{z}$ except in the set $\{i, j\}$. We only consider second-order expansion of (4.2), as for a number of applications, the second-order expansion is usually sufficient and higher-order terms have negligible effect on the output response. In the case of wind turbine, suppose y represent the power output that depends on say 3 random inputs: windspeed (z_1), wind standard deviation (z_2) and RPM (z_3), then f_0 represents the mean power output considering all the three random inputs, $f_1(z_1)$ represent the independent contribution of windspeed on the power and $f_{1,2}(z_1, z_2)$ quantify the interactions of wind speed and wind standard deviation on power. Further, the total variance of $f(\mathbf{z})$ is defined as:

$$D = \mathrm{Var}[f(\mathbf{z})] = \int_{\mathcal{D}_\mathbf{z}} f^2(\mathbf{z}) d\mathbf{z} - f_0^2 \tag{4.6}$$

As the expansion in (4.2) is not unique, some orthogonality conditions must be imposed [23, 24] which allows to express the total variance as

$$D = \sum_{i=1}^{M} D_i + \sum_{1 \le i < j \le M} D_{ij}, \tag{4.7}$$

where the first- and second-order partial variances are defined as

$$D_i = \int_0^1 f_i^2(z_i)dz_i, \tag{4.8}$$

$$D_{ij} = \int_0^1 \int_0^1 f_{ij}^2(z_i, z_j)dz_i dz_j. \tag{4.9}$$

The first- and second-order Sobol indices are then computed as

$$S_i = \frac{D_i}{D}, \quad S_{ij} = \frac{D_{ij}}{D}, \quad i, j = 1, 2, ..., M, \tag{4.10}$$

and the sum of all indices satisfy

$$\sum_{i=1}^{M} S_i + \sum_{1 \le i < j \le M} S_{ij} = 1. \tag{4.11}$$

Finally, the total effect of the parameter z_i on the output response Y is quantified using the total sensitivity indices measure

$$S_{T_i} = S_i + \sum_{j=1}^{M} S_{ij}. \tag{4.12}$$

The total sensitivity indices can be interpreted as an importance measure for the parameter z_i, therefore a large S_{T_i} implies that z_i has a strong influence on Y. The computation of S_{T_i} requires the approximation of partial variances defined in (4.8) - (4.9). Using the Monte Carlo method for computing these variance can be prohibitive for computationally expensive models Y. Polynomial Chaos Expansions (PCE) based approaches provides a more efficient alternative for sensitivity analysis. As shown in [25], one can compute the Sobol indices analytically by post-processing the PCE coefficients.

4.2. PCE based Sobol indices computation

The PCE approximation $f^K(\mathbf{z})$ of the computational model Y can be defined as a weighted sum of multivariate polynomials in $\mathbf{z}$ [22]

$$Y = f(\mathbf{z}) \approx f^K(\mathbf{z}) = \sum_{|\mathbf{k}|=0}^{K} w_{\mathbf{k}} \Psi_{\mathbf{k}}(\mathbf{z}), \tag{4.13}$$

where $\mathbf{k} \in \mathbb{N}_0^M$ is a M-dimensional multi-index with the magnitude $|\mathbf{k}| = k_1 + k_2 + ... + k_M$ and $\Psi_{\mathbf{k}}(\mathbf{z})$ multivariate polynomial is computed using a tensor product of univariate polynomials $\psi_{k_i}^{(i)}(z_i)$

$$\Psi_{\mathbf{k}}(\mathbf{z}) := \prod_{i=1}^{M} \psi_{k_i}^{(i)}(z_i). \tag{4.14}$$

where k_i is the order of univariate polynomial. The multivariate polynomials $\Psi_{\mathbf{k}}(\mathbf{z})$ is orthogonal with respect to the joint probability distribution of random vector $\mathbf{z}$. The choice of the univariate orthogonal polynomial ψ_{k_i} depends on the type of random variable, z_i. For example, for uniform random variable the Legendre family of polynomials is used, see [26] for details.

The total number of terms K in (4.13) can be based on a standard truncation scheme where we use all polynomials in M random variable with order less than or equal to p. More

advanced truncation schemes such as the maximum interaction or a hyperbolic scheme can also be employed depending on the application, see [27].

The PCE coefficient $w_{\mathbf{k}}$ corresponding to $\Psi_{\mathbf{k}}(\mathbf{z})$ can be computed using the projection approach:

$$w_{\mathbf{k}} = \mathbb{E}[\Psi_{\mathbf{k}}(\mathbf{z}) \cdot f^K(\mathbf{z})]. \tag{4.15}$$

The above expectation can be computed using a quadrature based integration methods such as the Gaussian quadrature, sparse quadrature rules (e.g. Smolyak's tensor grid), etc. Quadrature based integration methods suffer from curse of dimensionality where the number of integration points rises exponentially with an increase in the number of dimensions. In the current work where we have a fixed computational budget, methods that minimizes the approximation error for a given number of samples are more practical. Next, we briefly describe two such methods to compute PCE coefficients based on a fixed number of model evaluations.

4.2.1. Ordinary Least Square Instead of using projection based method one can also use least squares based approaches to compute coefficients [28]. The PCE expansion (4.13) can be expressed as a vector-matrix product

$$Y = f^K(\mathbf{z}) + \varepsilon_K = \sum_{|\mathbf{k}|=0}^{K} w_{\mathbf{k}} \Psi_{\mathbf{k}}(\mathbf{z}) + \varepsilon_K = \mathbf{w}^T \mathbf{\Psi}(\mathbf{z}) + \varepsilon_K, \tag{4.16}$$

where $\mathbf{w} = \{w_0, w_1, ..., w_K\}^T$ is the coefficient vector, $\mathbf{\Psi}(\mathbf{z}) = \{\Psi_0(\mathbf{z}), \Psi_1(\mathbf{z}), ..., \Psi_K(\mathbf{z})\}^T$ is the matrix containing multivariate polynomials in $\mathbf{z}$ and ε_K is the approximation error. The least-squares minimization problem can then be formulated as:

$$\overline{\mathbf{w}} = \arg\min \mathbb{E}\left[\left(\mathbf{w}^T \mathbf{\Psi}(\mathbf{z}) - f(\mathbf{z})\right)^2\right] \tag{4.17}$$

The coefficient vector $\overline{\mathbf{w}}$ can be computed using the Ordinary Least Squares (OLS) method. Given N model evaluations $\mathbf{Y} = \{f(\mathbf{z}_1), f(\mathbf{z}_2), ..., f(\mathbf{z}_N)\}$, we can obtain the coefficients as:

$$\overline{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}, \tag{4.18}$$

where $\mathbf{X} \in \mathbb{R}^{N \times K}$ is the design matrix where $\mathbf{X}_{i,j} = \Psi_j(\mathbf{z}_i)$, $i = 1, 2, ..., N$ and $j = 0, 1, 2, ..., K$. Although the OLS method allows to compute the coefficients using a limited number of model evaluations, one of the main drawback of this approach is that it may result in a PCE model that consists of high-order interaction terms that may lead to overfitting. For many engineering problems, low-order interactions between the input variables are more important. Therefore, the least squares form in (4.17) can be regularized such that the minimization results is a low-order sparse PCE model.

4.2.2. Least Angle Regression Least angle regression (LARS) algorithm [29] is one of the several methods can be applied to compute sparse PCE models [27, 30]. The key idea is to find the best set of polynomials from a given candidate set that minimizes the regularized form of (4.17):

$$\overline{\mathbf{w}} = \arg\min \mathbb{E}\left[\left(\mathbf{w}^T \mathbf{\Psi}(\mathbf{z}) - f(\mathbf{z})\right)^2\right] + \lambda ||\mathbf{w}||_1. \tag{4.19}$$

The LARS algorithm in the context of PCE can be summarized as follows. We start with all PCE coefficients for the candidate polynomials set to zero, and find the polynomial, say $\Psi_1(\mathbf{w})$ most correlated with the model evaluations. This polynomial enters the active set. Next, we take the largest step in the direction of $\Psi_1(\mathbf{w})$ until some other polynomial $\Psi_2(\mathbf{w})$ from the candidate

set, has as much correlation with the current residual. The second polynomial is now also added to the active set. In the next step, we move both the coefficients in the active set towards their least-square values until a third polynomial from candidate set shows as much correlation as the current PCE model based on two polynomials show with the model evaluations. The number of iteration is given by the minimum of $\{K, N\}$. After every iteration a posteriori error, for instance, Leave-One-Out (LOO) cross-validation error is computed. The active set with the smallest LOO is then chosen as the best sparse PCE model.

One of the main advantage of the LARS algorithm is that it works well in cases when $K >> N$, i.e. the number of polynomials in the candidate set is much more larger than the number of model evaluations.

4.3. Sensitivity analysis with discrete random variable

5. Results

5.1. NM80

As a reference wind turbine, we consider the 2MW NM80 turbine from the DANAERO project [13] with a blade radius of 38.8 m. The data for lift (Cl) and drag (Cd) polars are available at four locations along the blade radius at 11.87 m, 17.82 m, 28.97 m, and 35.53 m. The lift and drag variables at these sections are numbered Cl1 - Cl4 and Cd1 - Cd4 respectively. The reference value of polars is obtained from the wind-tunnel experiment with 3D corrections. Random samples of chord, twist, lift- and drag-polars are obtained by perturbing the control points with a uniformly distributed random variable.

There are two ways to obtain perturbed curves either by independently perturbing each of the control points (local perturbation) or using the same random number to perturb all the control points (global perturbation). We use local perturbation for chord control points Ch3 - Ch7 and twist control points Tw2 - Tw7 control points, see Fig. 3 (a) - (b), respectively. Each control point are perturbed using a uniform random number with bounds given by $\pm 5\%$ of the magnitude of a control point. For lift and drag curves, all control points are perturbed using the same uniform random number resulting in samples of the lift and drag curves as shown in Fig. 3 (c) - (d), respectively. For simplicity, we only consider uniform distributions for perturbing the control points, however, we can also use other probability distributions for this purpose. Also note that when using a uniform distribution with bounds given by the relative magnitude of control points, we tend to obtain small perturbations for control points with values close to zero.

5.1.1. Verification of GSA method We perform a convergence study for OLS and LARS to determine which method converges fastest with increasing dimension. For this, we use the parametrized chord from Fig. 3(a) with 9 control points. In Fig. 4, we show the convergence behavior of the two methods with increasing number of uncertain inputs, consisting of 4, 6 and 8 control points and for different number of model evaluations. We see that both methods scales well with increasing dimensions, however, PCE_LARS exhibits faster convergence than PCE_OLS in high dimensions.

5.1.2. Sensitivity analysis results In Fig. 5, we show the total order Sobol indices as a measure of the sensitivity of the different geometric and model parameters on the power output. In this case, we observe the important result that the uncertainty in model parameters is at least as important as the uncertainty in geometric parameters. Furthermore, within the geometric parameters, the chord distribution shows a significantly higher sensitivity compared to the twist variables

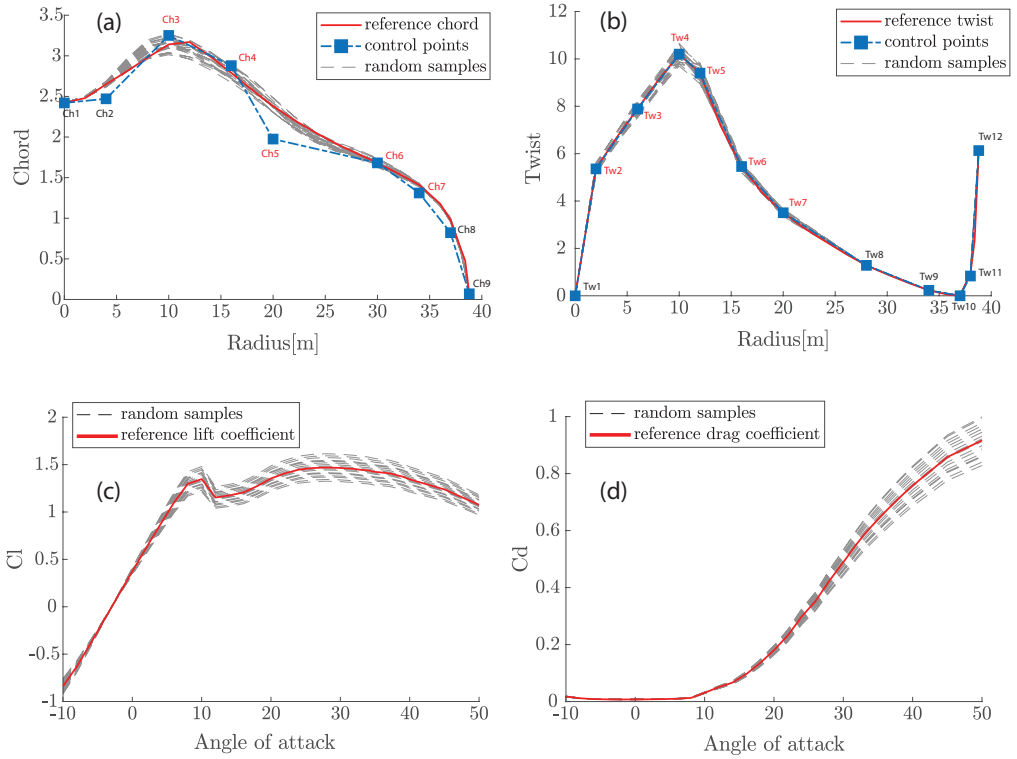
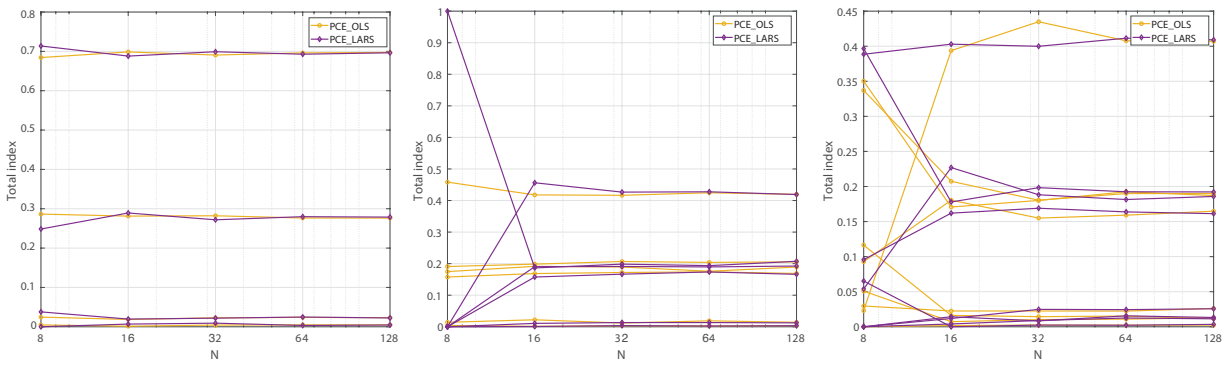


Figure 3: Random realization of chord, twist, lift- (Cl2) and drag-polars (Cd2).



(a) $M = 4, Ch1 - Ch4$ (b) $M = 6, Ch1 - Ch6$ (c) $M = 8, Ch1 - Ch8$

Figure 4: Convergence of PCE_OLS and PCE_LARS with increasing number of uncertain inputs M and samples N.

5.2. AVATAR

6. Conclusions

We have shown how Sobol indices computed using sparse adaptive polynomial expansion can be used for high-dimensional global sensitivity analysis of both geometric and model uncertainties.

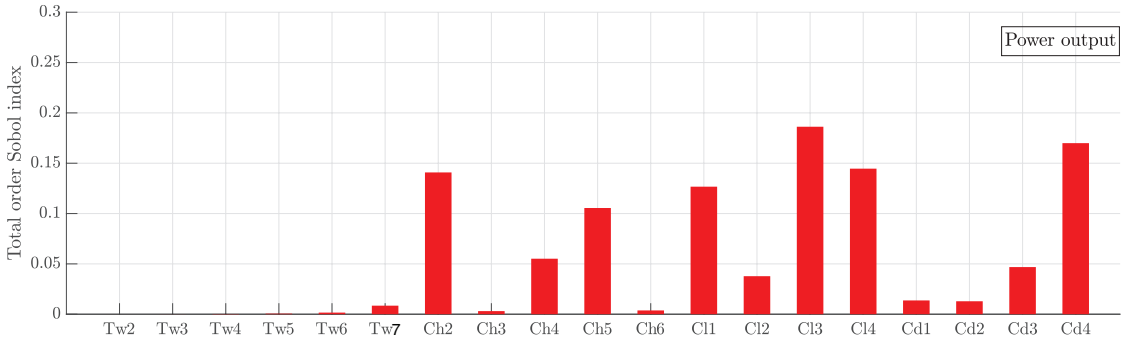


Figure 5

The identified sensitive model parameters will be utilized in future work to develop calibrated BEM models with built-in uncertainty estimates as part of the *WindTrue* project, and is connected to IEA Task 29.

[1] Burton T, Jenkins N, Sharpe D and Bossanyin E 2001 *Wind Energy Handbook, Second Edition* (John Wiley & Sons) ISBN 9781119992714

[2] Vorpahl F, Strobel M, Jonkman J M, Larsen T J, Passon P and Nichols J 2013 *Wind Energy* **17** 519–547 ISSN 10954244 URL `http://doi.wiley.com/10.1002/we.1588`

[3] Madsen H A, Riziotis V, Zahle F, Hansen M, Snel H, Grasso F, Larsen T, Politis E and Rasmussen F 2012 *Wind Energy* **15** 63–81 ISSN 10954244 URL `http://doi.wiley.com/10.1002/we.493`

[4] Sayed M, Klein L, Lutz T and Krämer E 2019 *Renewable Energy* **140** 304–318 ISSN 18790682

[5] Echeverría F, Mallor F and San Miguel U 2017 *Wind Energy* **20** 1601–1616 ISSN 10954244 URL `http://doi.wiley.com/10.1002/we.2111`

[6] Matthäus D, Bortolotti P, Loganathan J and Bottasso C L 2017 *35th Wind Energy Symposium, AIAA SciTech Forum, 9-13 January 2017, Grapevine, Texas* (Reston, Virginia: American Institute of Aeronautics and Astronautics) ISBN 978-1-62410-456-5 URL `http://arc.aiaa.org/doi/10.2514/6.2017-1849`

[7] Murcia J P, Réthoré P E, Dimitrov N, Natarajan A, Sørensen J D, Graf P and Kim T 2018 *Renewable Energy* **119** 910–922 ISSN 09601481 URL `http://dx.doi.org/10.1016/j.renene.2017.07.070https://linkinghub.elsevier.com/retrieve/pii/S0960148117306985`

[8] Robertson A, Sethuraman L and Jonkman J M *Assessment of Wind Parameter Sensitivity on Extreme and Fatigue Wind Turbine Loads*

[9] van den Bos L M M, Bierbooms W A A M, Alexandre A, Sanderse B and van Bussel G J W 2019 1–18 (*Preprint* `1904.07021`) URL `http://arxiv.org/abs/1904.07021`

[10] Bortolotti P, Canet H, Bottasso C L and Loganathan J 2019 *Wind Energy Science* **4** 397–406 ISSN 23667451

[11] Rinker J M 2016 *Journal of Physics: Conference Series* **753** ISSN 17426596

[12] Marelli S and Sudret B *UQLab: A Framework for Uncertainty Quantification in Matlab* pp 2554–2563

[13] Troldborg N, Bak C, Madsen H A and Skrzypinski W 2013 Danaero mw: Final report Tech. Rep. DTU Wind Energy E-0027

[14] Boorsma K, Grasso F and Holierhoek J 2012 Enhanced approach for simulation of rotor aerodynamic loads Tech. Rep. ECN-M-12-003

[15] Hansen A C and Butterfield C P 1993 *Annual Review of Fluid Mechanics* **25** 115–149

[16] Rogers D F 2000 *An introduction to NURBS: with historical perspective* (Elsevier)

[17] Bottasso C L, Croce A, Sartori L and Grasso F 2014 *Journal of Physics: Conference Series* **524** ISSN 17426596

[18] Ribeiro A F, Awruch A M and Gomes H M 2012 *Applied Mathematical Modelling* **36** 4898–4907 ISSN 0307904X

[19] de Boor C 1978 *Applied Mathematical Sciences*

[20] 2005 *Computer-Aided Design* **37** 791 – 797 ISSN 0010-4485 cAD '04 Special Issue: Modelling and Geometry Representations for CAD

[21] Razdan A 1999 *Tempe, AZ: Arizona State University* **1** 8

[22] Smith R C 2013 *Uncertainty Quantification: Theory, Implementation, and Applications* (Philadelphia, PA, USA: Society for Industrial and Applied Mathematics) ISBN 161197321X, 9781611973211

[23] Rabitz H and Aliş Ö F 1999 *Journal of Mathematical Chemistry* **25** 197–233 ISSN 1572-8897

[24] Sobol I 2001 *Mathematics and Computers in Simulation* **55** 271 – 280 ISSN 0378-4754 the Second IMACS Seminar on Monte Carlo Methods URL http://www.sciencedirect.com/science/article/pii/S0378475400002706
[25] Sudret B 2008 *Reliability Engineering & System Safety* **93** 964 – 979 ISSN 0951-8320 bayesian Networks in Dependability
[26] Xiu D and Karniadakis G E 2002 *SIAM Journal on Scientific Computing* **24** 619–644
[27] Blatman G 2009 *Ph.D. thesis, Université Blaise Pascal, Clermont-Ferrand, France*
[28] Berveiller M, Sudret B and Lemaire M 2006 *European Journal of Computational Mechanics* **15** 81–92
[29] Efron B, Hastie T, Johnstone I and Tibshirani R 2004 *Ann. Statist.* **32** 407–499 URL https://doi.org/10.1214/009053604000000067
[30] Blatman G and Sudret B 2011 *Journal of Computational Physics* **230** 2345 – 2367 ISSN 0021-9991