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 framework is presented for performing global sensitivity analysis of model parameters associated with the Blade Element Momentum (BEM) models. 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 relatively strong model assumptions at the basis of BEM theory, the results from BEM codes can be subject to significant 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 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 likely not sufficiently accurate [4].

Recently, several papers have addressed the uncertainty in BEM model output by performing uncertainty propagation and 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.

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 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, 10]. We will employ the uncertainty quantification toolbox `UQLab` [11], which computes the Sobol indices based on a sparse polynomial chaos expansion. The main novelty of the work lies in creating a generic framework for studying parametric uncertainteis, including both model uncertainties and geometric uncertainties. This framework for sensitivity analysis is part of the so-called *WindTrue* project, in which the long-term goal is the development of calibrated BEM models. 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 [12] and IEA Task 29. The sensitivity analysis will make clear which of the model parameters are to be used in the calibration phase.

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 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` [13]. For the case of a rigid turbine, with a uniform inflow field, the main uncertainties that arise in the BEM formulation are [14]:

- 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 and typically incorporated via *a wake correction.*

In this paper we will focus on the first uncertainty – uncertainty in lift and drag coefficients – but we stress that the framework and methodology are directly applicable to other BEM model uncertainties as well. In a parallel paper, see [15], the effect of other uncertainties such as the dynamic stall model will be investigated. Next to these model uncertainties we will consider geometric uncertainties: the chord and twist distribution along the blade.

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 a limited number of control points can 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 $S(r)$ is expressed using a weighted sum of n basis functions (or B-splines):

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

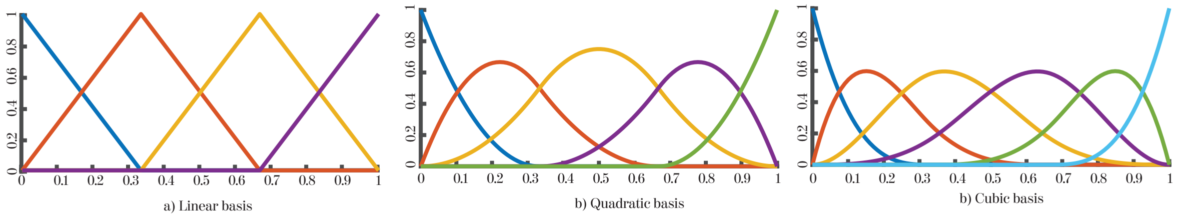
where c_i is the weight of the i-th control point and $B_{i,p}(r)$ is the value of the B-spline corresponding to the i-th control point at radial location r along the blade, with p denoting the polynomial degree. In our case, $S(r)$ represents for example the chord length. A sequence of non-decreasing parameters $X = \{r_1, r_2, ..., r_m\}$, 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 $B_{i,p}(r)$ is non-zero at r. B-splines are recursive in polynomial degree: quadratic B-splines follow from linear B-splines, cubic from quadratic B-splines etc. Starting with a B-spline of degree 0 defined as

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

higher-order B-splines follow using the recurrence relation [19]:

$$B_{i,p}(r) := \frac{r - r_i}{r_{i+p} - r_i} B_{i,p-1}(r) + \frac{r_{i+p+1} - r}{r_{i+p+1} - r_{i+1}} B_{i+1,p-1}(r), \quad p \geq 1. \tag{3.3}$$

Only a few nearby control points are active when computing the NURBS curve (3.1); we use the de Boor's algorithm [19] for efficient construction. In Fig. 1, we show linear, quadratic and cubic splines for $r \in [0, 1]$ 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 $X = \{0, 0, 1/3, 2/3, 1, 1\}$.



a) Linear basis

b) Quadratic basis

b) Cubic basis

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 $X = \{r_j\}_{j=1}^{n}$. The knots can be uniformly spaced or chosen heuristically (for advanced knot selection approaches for knot selection see [20]). Next we sample $S_{ref}(r)$ at X and also compute $B_{i,p}(r_j)$, 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:

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

where $\mathbf{S}_{ref} \in \mathbb{R}^n$ is a vector containing sampled values of $S_{ref}(r)$ 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 r_j. Once the control points are obtained, the approximate reference curve $S_n(r)$ 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.

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 $\tilde{S}(r)$:

$$\tilde{S}(r) = \sum_{i=1}^{n} \tilde{c}_i B_{i,p}(r) \qquad \text{where} \qquad \tilde{c}_i = c_i(1 + \Delta_i), \tag{3.5}$$

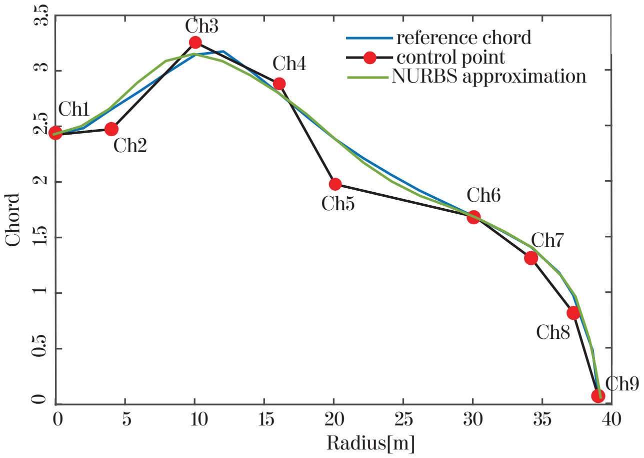


Figure 2: Parametrization of chord using 9 control points and second-order B-splines.

and the perturbation is randomly sampled as $\Delta_i \sim \mathcal{U}[-0.05, 0.05]$. This equation is applicable to both the twist- and chord distribution, where $\tilde{c}_i$ can denote either the value of the twist control points, Tw_i, or of the chord control points, Ch_i. For the lift- and drag polars, we use the following parametrization:

$$\tilde{S}(r, \alpha) = \sum_{i=1}^{n} \tilde{c}_i(\alpha) B_{i,p}(r) \qquad \text{where} \qquad \tilde{c}_i(\alpha) = c_{\mathrm{ref},i}(\alpha)(1 + \Delta_i), \tag{3.6}$$

where $c_i(\alpha)$ can denote either the lift coefficient Cl_i or drag coefficient Cd_i, and the subscript ref denotes the reference curve.

4. Global sensitivity analysis

The objective of sensitivity analysis is to quantify the relative significance of individual inputs (or their combination) and how variations in input values affect the output of interest. Sensitivity analysis techniques can be classified as local and global [21]. 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 around the 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 behaviour 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 the aeroelastic models considered in this work.

4.1. Sobol analysis

We employ the 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. Consider a generic BEM model $f(\mathbf{z})$, depending on uncertain parameters $\mathbf{z}$ and returning a quantity of interest Y (e.g. power, loads, moments):

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

where $\mathbf{z} = [z_1, z_2, ..., z_M] \in \mathcal{D}_{\mathbf{z}} \in \mathbb{R}^M$ is a vector of random variables (parameters). For ease of notation, we assume these input parameters to be uniformly distributed i.e. $z_i \sim \mathcal{U}(0, 1)$ and the support of the input set is $\mathcal{D}_{\mathbf{z}} = [0,1]^M$ where M is the total number of input parameters. The Sobol decomposition is based on the following hierarchical representation [21]:

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

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

$$f_0 = \int_{\mathcal{D}_{\mathbf{z}}} f(\mathbf{z}) d\mathbf{z}, \qquad f_i(z_i) = \int_{\mathcal{D}_{\mathbf{z}}{}^{M-1}} f(\mathbf{z}) d\mathbf{z}_{\sim\{i\}} - f_0, \tag{4.3}$$

$$f_{ij}(z_i, z_j) = \int_{\mathcal{D}_{\mathbf{z}}{}^{M-2}} f(\mathbf{z}) d\mathbf{z}_{\sim\{i,j\}} - f_i(z_i) - f_j(z_j) - f_0, \tag{4.4}$$

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 $\{i, j\}$. Higher-order terms are interpreted in a similar manner. Note that (4.2) is only valid for independent input parameters. We consider a second-order expansion of (4.2), which is generally sufficient for the purpose of this article.

As an example, Y can represent the power output of a wind turbine, with random inputs windspeed (z_1), wind standard deviation (z_2) and rotational speed (z_3). f_0 then represents the mean power output, $f_1(z_1)$ represent the independent contribution of windspeed on the power and $f_{1,2}(z_1, z_2)$ quantifies the interactions of wind speed and wind standard deviation on power.

Sobol indices are based on a variance decomposition [22]. The definition of the variance is:

$$D = \mathrm{Var}[f(\mathbf{z})] = \int_{\mathcal{D}_{\mathbf{z}}} f^2(\mathbf{z}) d\mathbf{z} - f_0^2, \tag{4.5}$$

which is rewritten as

$$D = \sum_{i=1}^{M} D_i + \sum_{1 \leq i < j \leq M} D_{ij}, \tag{4.6}$$

where the first- and second-order partial variances are defined via decomposition (4.2) as

$$D_i = \int_0^1 f_i^2(z_i) dz_i, \qquad D_{ij} = \int_0^1 \int_0^1 f_{ij}^2(z_i, z_j) dz_i dz_j. \tag{4.7}$$

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

$$S_i = \frac{D_i}{D}, \quad S_{ij} = \frac{D_{ij}}{D}, \quad i, j = 1, 2, ..., M. \tag{4.8}$$

The total effect of z_i on the output Y can be quantified using the total sensitivity index $S_{T_i} = S_i + \sum_{j=1}^{M} S_{ij}$. 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 (4.7). Using the Monte Carlo method for computing these variances can be prohibitive for computationally expensive models Y. On the other hand, Polynomial Chaos Expansions (PCE) based approaches provide a more efficient alternative for sensitivity analysis, and will be used here. As shown in [23], one can compute the Sobol indices analytically by post-processing the PCE coefficients, whose computation will be detailed next.

4.2. PCE-based Sobol indices computation

A PCE approximation $f^K(\mathbf{z})$ to the computational model $Y = f(\mathbf{z})$ can be defined as a weighted sum of multivariate polynomials in $\mathbf{z}$ [21]:

$$Y = f(\mathbf{z}) \approx f^K(\mathbf{z}) = \sum_{|\mathbf{k}|=0}^{K} w_{\mathbf{k}} \Psi_{\mathbf{k}}(\mathbf{z}), \tag{4.9}$$

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

$$\Psi_{\mathbf{k}}(\mathbf{z}) := \prod_{i=1}^{M} \psi_{k_i}^{(i)}(z_i), \tag{4.10}$$

where k_i is the order of the univariate polynomial. The choice of univariate orthogonal polynomial ψ_{k_i} depends on the type of the random variable z_i. For example, for uniform random variables the Legendre family of polynomials is used. The total number of terms K in (4.9) depends on the truncation scheme; in a standard truncation scheme all polynomials in M random variable with order less than or equal to p are used. In this work, a more advanced truncation scheme known as the hyperbolic truncation scheme is used [24].

The PCE coefficients $w_{\mathbf{k}}$ corresponding to $\Psi_{\mathbf{k}}(\mathbf{z})$ can be computed using a pseudo-spectral approach by using quadrature methods. For a small number of dimensions this is an efficient approach. However, in the current work we will encounter more than 10 dimensions and quadrature methods become too expensive as they suffer strongly from the curse of dimensionality. We therefore focus on methods in which the sampling points can be chosen more freely and adaptively: Ordinary Least Squares (OLS) and Least-Angle Regression (LARS).

4.2.1. Ordinary Least Squares The PCE expansion (4.9) can be expressed in terms of the following vector-matrix product

$$Y = f^K(\mathbf{z}) + \varepsilon_K = \sum_{|\mathbf{k}|=0}^{K} w_{\mathbf{k}} \Psi_{\mathbf{k}}(\mathbf{z}) + \varepsilon_K = \mathbf{w}^T \boldsymbol{\Psi}(\mathbf{z}) + \varepsilon_K, \tag{4.11}$$

where $\mathbf{w} = \{w_0, w_1, ..., w_K\}^T$ is the coefficient vector, $\boldsymbol{\Psi}(\mathbf{z}) = \{\Psi_0(\mathbf{z}), \Psi_1(\mathbf{z}), ..., \Psi_K(\mathbf{z})\}^T$ 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:

$$\overline{\mathbf{w}} = \arg\min_{\mathbf{w}} \mathbb{E}\left[\left(\mathbf{w}^T \boldsymbol{\Psi}(\mathbf{z}) - f(\mathbf{z})\right)^2\right]. \tag{4.12}$$

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:

$$\overline{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}, \tag{4.13}$$

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, 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 most important. Therefore, the least squares form in (4.12) can be regularized such that the minimization results in a low-order sparse PCE model, as discussed next.

4.2.2. Least Angle Regression The LARS algorithm [25] is one of the methods that can be applied to compute sparse PCE models [24]. The key idea is to find the best set of polynomials from a given candidate set that minimizes the regularized form of (4.12):

$$\overline{\mathbf{w}} = \arg\min_{\mathbf{w}} \mathbb{E}\left[\left(\mathbf{w}^T\mathbf{\Psi}(\mathbf{z}) - f(\mathbf{z})\right)^2\right] + \lambda||\mathbf{w}||_1. \tag{4.14}$$

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 the candidate set shows as much correlation as the current PCE model based on two polynomials shows with the model evaluations. The number of iterations is given by the minimum of $\{K, N\}$. After every iteration an a-posteriori error, for instance the 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 larger than the number of model evaluations.

5. Results

5.1. Operating conditions

As a reference wind turbine, we consider the 2MW NM80 onshore turbine from the DANAERO project [12] with a blade radius of 38.8 m and a hub height of 57.19 m with the mean wind speed at hub height set as 6.1m/s. Furthermore, the RPM is set at 12.3/min., the blade pitch angle at 0.15 deg and the turbine yaw angle is zero.

5.2. Choice of uncertain inputs

The data for lift (C_l) and drag (C_d) 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 these 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: independently perturbing each of the control points (local perturbation) or using a single 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. The twist and chord are perturbed following equation (3.5), whereas the lift and drag are perturbed according to equation (3.6). Examples of the resulting lift and drag curves are 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.

5.3. Verification of sensitivity analysis method

We perform a convergence study for both OLS and LARS in order to determine which method converges fastest with increasing number of uncertain parameters and number of model runs.

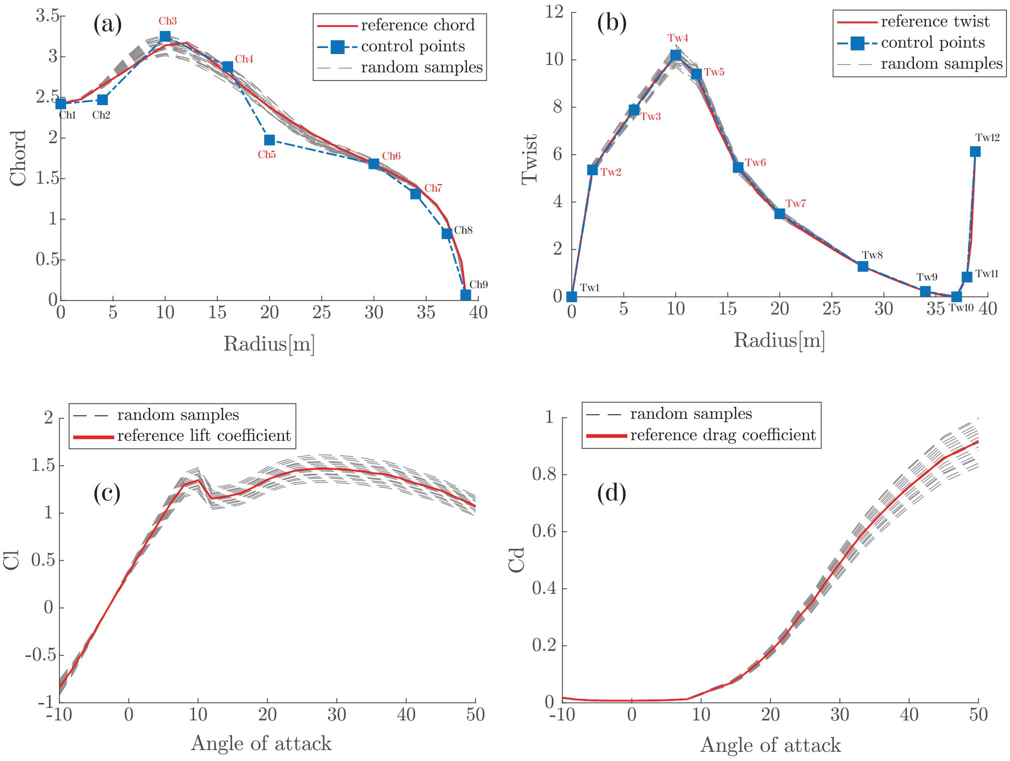
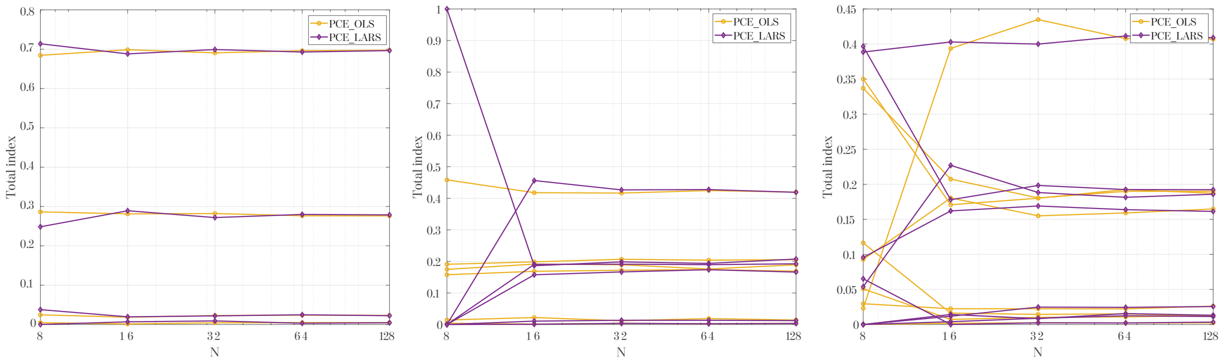


Figure 3: Random realization of chord, twist, lift- (Cl2) and drag-polars (Cd2).

For this purpose we only include uncertainty in the chord and set the power production as the quantity of interest. The NURBS representation is defined by 9 control points, of which a subset is taken uncertain (see Fig. 3(a)). In Fig. 4, we show for both OLS and LARS the convergence behaviour of the total Sobol indices of the power output with increasing number of model evaluations N (runs of the AeroModule), and with increasing number of uncertain inputs M (4, 6 and 8 control points). We see that both methods scale well with increasing dimensions, however, LARS exhibits faster convergence than OLS in high dimensions due to its adaptive nature, and will be the method of choice in the subsequent tests. For both OLS and LARS, we start with a small N and double it with every iteration until converged values of Sobol indices are obtained.

5.4. Sensitivity analysis results

In Fig. 5 and 6, we show the total order Sobol indices as a measure of the sensitivity of the different geometric and model parameters on the power output and the axial force, respectively, computed using LARS with $M = 19$ uncertain parameters chosen from Fig. 3. We found that $N = 1024$ model evaluations were sufficient to achieve converged indices. In this case, we observe 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 than the twist variables. Lastly, we observe that the axial force is basically only sensitive to the lift coefficient at the outward blade sections, whereas the power is sensitive to both lift and drag coefficients, also at the inward section of the blade.



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

Figure 4: Convergence of OLS and LARS with increasing number of uncertain inputs M and model evaluations N.

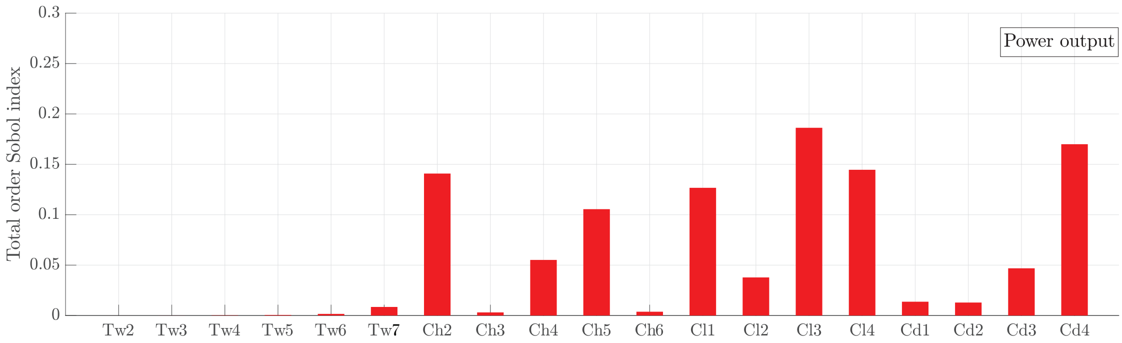


Figure 5: Sensitivity of power output towards geometric and model uncertainties.

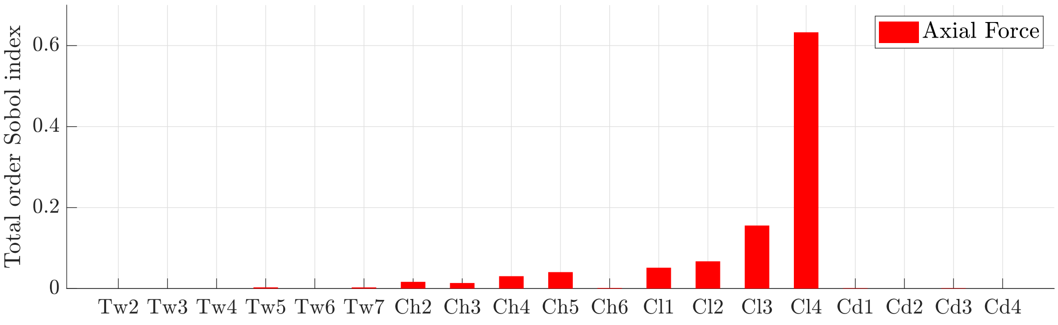


Figure 6: Sensitivity of axial force towards geometric and model uncertainties.

6. Conclusions

In this paper we have developed a framework for global sensitivity analysis using Sobol indices computed using sparse adaptive polynomial chaos expansions. The framework can be used for high-dimensional global sensitivity analysis of both geometric and model uncertainties in BEM models. As such, the proposed framework is generic and can deal with many types of (parametric) uncertainties, of which a few were investigated in this paper: twist, chord, lift coefficient and drag coefficient. By using NURBS parameterizations of chord, twist, lift and drag as a function of the blade radius, the number of uncertainties could be limited to 19. The results highlight amongst others the importance of the lift coefficient, especially for the axial force prediction. The identified sensitivities will be utilized in future work to develop calibrated BEM models with built-in uncertainty estimates, as part of the *WindTrue* project, and used to better understand the differences between BEM models and measurement data as studied in IEA Task 29.

References

[1] Burton T, Jenkins N, Sharpe D and Bossanyi E 2001 *Wind Energy Handbook* (John Wiley & Sons)

[2] Vorpahl F, Strobel M, Jonkman J M, Larsen T J, Passon P and Nichols J 2013 *Wind Energy* **17** 519–547

[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

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

[5] Echeverría F, Mallor F and San Miguel U 2017 *Wind Energy* **20** 1601–1616

[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*

[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

[8] Robertson A, Sethuraman L, Jonkman J M and Quick J 2018 *NREL/CP-5000-70445*

[9] van den Bos L M M, Bierbooms W A A M, Alexandre A, Sanderse B and van Bussel G J W 2020 *Wind Energy* **1** 1–15

[10] Rinker J M 2016 *Journal of Physics: Conference Series* **753**

[11] Marelli S and Sudret B 2014 *UQLab: A Framework for Uncertainty Quantification in Matlab*

[12] Troldborg N, Bak C, Madsen H A and Skrzypinski W 2013 DANAERO MW: Final Report Tech. Rep. DTU Wind Energy E-0027

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

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

[15] Caboni M, Carrion M, Rodriguez C, Schepers G and Boorsma K 2020 *Journal of Physics: Conference Series* **this volume**

[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**

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

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

[20] Li W, Xu S, Zhao G and Goh L P 2005 *CAD Computer Aided Design* **37** 791–797

[21] Smith R C 2013 *Uncertainty Quantification: Theory, Implementation, and Applications* (SIAM)

[22] Sobol I 2001 *Mathematics and Computers in Simulation* **55** 271 – 280

[23] Sudret B 2008 *Reliability Engineering & System Safety* **93** 964 – 979

[24] Blatman G 2009 *Ph.D. thesis, Université Blaise Pascal, Clermont-Ferrand, France*

[25] Efron B, Hastie T, Johnstone I and Tibshirani R 2004 *Ann. Statist.* **32** 407–499