
# Computing the power profiles for an Airborne Wind Energy system based on large-scale wind data

Renewable Energy 162 (2020) 766–778

Contents lists available at ScienceDirect

Renewable Energy

journal homepage: www.elsevier.com/locate/renene

# Computing the power profiles for an Airborne Wind Energy system based on large-scale wind data

E.C. Malz [a, *], V. Verendel [b], S. Gros [c]

[a] Department of Electrical Engineering, Chalmers University of Technology, Göteborg, Sweden
[b] Department of Computer Science and Engineering, Chalmers University of Technology, Göteborg, Sweden
[c] Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway

ARTICLE INFO

*Article history:*
Received 19 February 2020
Received in revised form
23 May 2020
Accepted 11 June 2020
Available online 13 August 2020

*Keywords:*
Airborne wind energy
Optimal control
Big data
Homotopy path strategy
Machine learning

ABSTRACT

Airborne Wind Energy (AWE) is a new power technology that harvests wind energy at high altitudes using tethered wings. Studying the power potential of the system at a given location requires evaluating the local power production profile of the AWE system. As the optimal operational AWE system altitude depends on complex trade-offs, a commonly used technique is to formulate the power production computation as an Optimal Control Problem (OCP). In order to obtain an annual power production profile, this OCP has to be solved sequentially for the wind data for each time point. This can be computationally costly due to the highly nonlinear and complex AWE system model. This paper proposes a method how to reduce the computational effort when using an OCP for power computations of large-scale wind data. The method is based on homotopy-path-following strategies, which make use of the similarities between successively solved OCPs. Additionally, different machine learning regression models are evaluated to accurately predict the power production in the case of very large data sets. The methods are illustrated by computing a three-month power profile for an AWE drag-mode system. A significant reduction in computation time is observed, while maintaining good accuracy.



## 1. Introduction

Airborne Wind Energy (AWE) is a new type of wind-harvesting energy technology, attracting research interest in academia and industry. The technology produces electricity using tethered wings that fly in a cross-wind fashion. There are two main technologies currently under development, pumping-mode and drag-mode systems. In the pumping-mode technology, the tethered wing generates power by reeling out a tether from a ground-based winch connected to an electric generator. When reaching the maximum tether length, the wing is reeled back by using a fraction of the energy generated during the reel-out phase. Drag-mode systems instead generate power via turbines mounted on the tethered wings, transmitting the produced electricity via the tether to the ground station.

In order to estimate the large-scale potential of AWE, previous work has evaluated the available annual wind resource at high altitudes [1,2]. This theoretical potential is reduced when the physical limitations of the AWE system are properly taken into account along with the efficiency of the conversion from wind to electric energy. This conversion efficiency is affected by the tether drag [3] and the availability of the AWE during severe weather conditions [4]. A realistic assessment of the large-scale AWE potential must account for these effects. However, unlike traditional wind turbines, AWE systems have no machine power curves that describe the instantaneous power as a function of the current wind speed [5]. Instead, the power output of an AWE system depends on the flight trajectory and the specific wind speeds at the current flight altitude.

Previous work has taken the energy conversion into account and estimated the annual power production of an AWE pumping-mode system [6,7]. In Ref. [6], Loyd's analytical model [8] is used to estimate the power and generate a set of power curves. For the power curves, the ratios of the tether forces during traction and retraction are varied, along with the ratios of tether speeds, while other variables are kept constant. In Ref. [7], a point-mass model is used for the AWE system, describing the system dynamics in more detail than in Ref. [6]. To avoid the complexity of controlling a dynamic

* Corresponding author.
*E-mail addresses:* elenama@chalmers.se (E.C. Malz), vive@chalmers.se (V. Verendel), sebastien.gros@ntnu.no (S. Gros).

https://doi.org/10.1016/j.renene.2020.06.056

point-mass model, the flight trajectory is prescribed and discretized. Assuming a quasi-steady state, the power is then optimized for each time instant with respect to wing speed, tether speed, tether force, and roll angle. The work of [7] offers a fast estimation of the power production, but the assumption of a quasi-steady state limits the accuracy of the estimation. Another limitation of the model, as noted in Refs. [7], is the inability to ascertain choices for different optimal operational altitudes that depend on the location as well as on the different system designs. In order to alleviate these limitations, the power can be estimated more accurately by formulating the power maximization as a genuine optimal control problem (OCP), fully accounting for the system dynamics. The approach of using an OCP for computing the power production is a commonly used technique among the community. The concept is originally introduced in Ref. [9] and has been applied in several work [10–15]. The underlying system models used for the optimization range from simple and low-order to detailed and highly complex. In this study, the OCP is solely used for trajectory planning and the power computation, rather than investigating the control strategy of the wing. The latter is not focus of this study and needs to be analyzed separately.

For traditional wind turbines, the relevant wind field is limited to around the height of the nacelle, i.e., about 100 m–180 m, depending on the type of turbine. AWE systems aim for winds at higher altitudes, typically up to 500 m, and tap into a larger vertical wind field ranging from 50 m to 500 m during their operation. In order to estimate the wind speeds at the height of a wind turbine, the logarithmic law or the power law is often used [16]. However, real wind profiles can differ significantly from these models, and vary strongly in time [17]. Moreover, for high-altitude winds, the assumption of a logarithmic wind profile is not valid [2]. Hence, for detailed AWE analyses, such as an estimate of the annual power generation, the logarithmic approximation might not be sufficient.

Both [6,7], used real wind-speed measurements at a specific location. Such detailed measurements are not necessarily available at a high spatial and temporal resolution in conjunction with a large-scale assessment. This limitation is considered in Ref. [6], which proposes the use of statistical wind-speed distributions as an alternative to detailed data. However, analyses of large-scale systems, studying regional dynamics of power generation and demand, require representative hourly power profiles for all involved technologies, cf [18]. In Refs. [14], the detailed temporal evolution of wind data is used to model and compare the power production profiles of traditional wind turbines and drag-mode AWE systems. Here, we present the method applied there.

Research is looking into computing the AWE power production based on real wind profiles in order to investigate the performance of this new renewable energy technology. At the same time, the concept of using optimal control for computing the power production of AWE is common, but presents a rather complex and computational tedious approach. Hence, this paper aims to combine these two targets and proposes a methodology for computing the power production profiles of an AWE system via an OCP. The methodology allows to implement large-scale wind data into an OCP while keeping the computational complexity at a low level. As a result, optimal control can be used as a tool to evaluate the power production for wind data with a high spatial and temporal resolution in order to assess the large-scale potential of the technology.

In order to achieve this, first realistic wind profiles are acquired from MERRA wind data (NASA), which are available at a high temporal and spatial resolution [19]. Then, the wind profiles for each time and location point are implemented *individually* into the OCP for computing the instantaneous maximal power production. The OCP is discretized into a Nonlinear Program (NLP) and solved with a primal-dual interior-point solver. Solving this NLP is computationally costly, due to the fairly high complexity of the dynamics of the AWE system.

The OCP is discretized into a Nonlinear Program (NLP) and solved with a primal-dual interior-point solver. Solving the NLP is computationally costly, due to the fairly high complexity of the dynamics of the AWE system. As the wind profiles are treated one by one, the NLP must be solved for a large number of times and evaluating the annual energy yield becomes a tedious and time-consuming task. As also noted in Ref. [7], the complexity associated with solving an OCP is problematic. In order to overcome this problem and reduce the computation time, this paper investigates and proposes an approach based on continuation and homotopy methods [20]. These techniques are commonly used in parametric optimization studies and have been applied to a number of problems. The basic idea is to find a homotopy path among the optimal solutions to facilitate the work of the solver. In Ref. [21], a homotopy-based nonlinear interior-point method is used to solve nonlinear model predictive control problems. In Ref. [9], a homotopy strategy is proposed to find an initial guess for a non-convex highly nonlinear program of the optimization of an AWE system. This strategy is applied in several studies addressing AWE optimization, cf. [15,22,23], and is adopted here for the first initialization of the problem. This strategy is necessary to obtain the initial guess, but the computation is costly. Ideally, solving a large family of NLPs to obtain power production profiles would not require this initialization for each individual problem. Therefore, this paper presents a specific homotopy-following-path algorithm (HPbd) that allows for solving a whole family of NLPs with a single initialization. To illustrate the algorithm, a drag-mode AWE system is considered.

If the annual power production for a very large region with a high spatio-temporal resolution is needed, the number of NLPs to be solved will still be extremely high. In order to circumvent this, we propose machine-learning techniques for training a regression model on the solutions of the optimizations, so that the power production of an AWE system can be predicted based on the wind conditions. Regression models of various types have been used in previous work to predict the power generated by traditional wind turbines [24–26]. However, this method has not yet been applied to predict the power generated by AWE systems. Here, we evaluate several regression models in order to find the one that is best suited to predict the optimal power production.

The paper is organized as follows. Section 2.1 presents the power-optimizing OCP and the underlying AWE model. Section 2.2 details the wind-data processing and implementation in the NLP. Section 2.3 briefly describes primal-dual interior-point methods. The continuation method and homotopy-path-following method are presented in Section 2.4. Section 2.5 describes the implementation of the proposed method in the current case. Section 2.6 estimates the optimal power production using a regression model. Section 3 presents the results and discusses accuracy and computation time measurements, and Section 4 concludes.

## 2. Background

This section first briefly presents the mathematical model representing the AWE system along with the OCP. Here, a drag-mode system is considered, but the proposed method also applies to pumping-mode systems. Next, the raw wind data are processed to yield NLP parameter inputs. This is followed by a short background on solving a Nonlinear Program (NLP) with the primal-dual interior-point method. In section 2.4, we discuss the application of the homotopy-path-following method in NLPs and describe the actual implementation. Finally, we propose a strategy based on machine learning using regression models to treat extremely large wind

data.

### 2.1. Description of the optimal control problem (OCP)

In order to compute the average power production of an AWE system, we formulate a mathematical model describing the dynamics of a wing flying in a cross-wind trajectory. This paper considers a drag-mode system following a circular flight path. The dynamics of the wing is described as a body with six degrees of freedom using high-index Differential Algebraic Equations (DAEs). The tether is modeled as a rigid rod with mass and drag, neglecting the tether dynamics. The motion of the wing is driven by the aerodynamic forces and moments, which depend on the instantaneous wing position and relative wind speed.

The mathematical model is detailed in Ref. [27], which considers a pumping-mode AWE system. However most of the formulations are similar for the drag-mode model. The main differences are the tether model and power production model, along with some of the wing parameters. Here, we choose a rigid 666 kW wing with a wingspan of 28 m. Other parameters can be found in our work [14], which uses the same system model. The model proposed in Ref. [27] assumes a straight tether, where the length is controlled via the tether acceleration $\ddot{l}$. In the drag-mode system, the tether length is fixed to $l = 500$ m with the tether acceleration set to zero. The power is generated via on-board turbines, modeled as

$$P = F_{\text{prop}} \mathbf{v}_{\text{app}} \frac{\mathbf{v}_{\text{app}}}{||\mathbf{v}_{\text{app}}||} \eta, \tag{1}$$

where $F_{\text{prop}}$ is a scalar control input describing the braking force of the propellers, $\mathbf{v}_{\text{app}}$ is the instantaneous relative wind velocity and $\eta$ the efficiency of the on-board propellers, assumed constant here. The braking force $F_{\text{prop}}$ is an additional drag force added to the aerodynamic wing and tether drag. The aerodynamic model is defined as in Ref. [27].

The average power generation can then be computed by solving an optimal control problem to maximize the average value of (1) over a periodic trajectory of the system, with boundary conditions and a free duration $T > 0$. The OCP is defined as

$$\min_{\mathbf{x},z,\mathbf{u},T} \quad -\frac{1}{T}\int_0^T P(\mathbf{x},\mathbf{p},T) + \ell_{\text{Reg}}(\mathbf{u})\, \mathrm{d}t \tag{2a}$$

$$\text{s.t.} \quad \mathbf{F}(\dot{\mathbf{x}},\mathbf{x},z,\mathbf{u},\mathbf{p},T) = 0, \tag{2b}$$

$$c(\mathbf{x}(0),\mathbf{x}(T)) = 0, \tag{2c}$$

$$\mathbf{h}(\mathbf{x},z,\mathbf{u},\mathbf{p},T) \leq 0, \tag{2d}$$

where function $\mathbf{F}$ gathers the equations of the dynamic AWE system model in the form of implicit DAEs. The decision variables $\mathbf{x} \in \mathbb{R}^{23}$, $z \in \mathbb{R}$ gather the differential and (scalar) algebraic states, respectively, $\mathbf{u} \in \mathbb{R}^4$ gathers the control inputs acting on the system, namely the change of the surface deflections and the change in the drag force $F_{\text{prop}}$. Vector $\mathbf{p} \in \mathbb{R}^{2\times 10}$ gathers the wind profile parameters, detailed below in Section 2.2. The optimization variable $T \in \mathbb{R}$ denotes the orbit time for one power cycle. Constraint (2c) enforces the periodicity of the trajectories, and (2d) gathers the actuator and operational limitations of the AWE system. The cost functional (2a) provides the average power generation by computing the electrical energy $P(\mathbf{x},\mathbf{p},T)$ produced by one flight trajectory, and dividing it by the trajectory time $T$. A small quadratic regularization term $\ell_{\text{Reg}}(\mathbf{u})$ is applied on the control inputs $\mathbf{u}$ in order penalize them and regularize the problem [9].

### 2.2. Wind data extraction and processing

Computing the power generated over a certain timespan, e.g., over the course of a year, requires accurate wind data. In this paper, the data are taken from the Modern-Era Retrospective analysis for Research and Applications version 2 (MERRA-2) [19], an atmospheric reanalysis provided by NASA. The reanalysis is four-dimensional, covering space and time coordinates $(lat, lon, z, t)$, where $lat, lon$ correspond to latitude and longitude and $z$ pressure. Two-dimensional wind-speed vectors are available for each grid point $lat, lon, z, t$. The wind-speed vectors are given in the common meteorological wind directions $u$ and $v$, i.e., as east/west and north/south. In this study only horizontal wind-speeds are considered.

For this paper, two different MERRA subsets of wind data are combined in order to cover both low and high altitudes. For both datasets, the spatial resolution of the grid is $0.625^\circ \times 0.5^\circ$ longitude-by-latitude. The low altitude winds are given by the hourly data set from Ref. [28], including the wind vectors $u$ and $v$ at the vertical altitudes 2, 10, and 50 m, respectively. The high-altitude winds are taken from the averaged three-hourly data set from Ref. [29], sampled vertically at 72 pressure levels, from surface pressure down to 0.0100 hPa, corresponding to an altitude of about 70 km. In this study, the maximal system operation height is set to 500 m, which means that only the ten bottom altitude levels up to 835 hPa are relevant. The low- and high-altitude datasets are combined at each grid and time point to form the wind vectors with a three-hourly resolution.

In order for the wind data to be implemented into the NLP, the points have to be represented as a sufficiently smooth function. Thus, the wind speed $u$,$v$ components are converted into the Cartesian reference frame for each time instance, with the main wind direction always aligned to the $x$-axis and interpolated to get the wind conditions as a function of the altitude. The alignment can be done under the assumption that the AWE system can operate in any direction and the transition energy needed between different directions is negligible. Further, it is assumed that the three-hourly-averaged winds are constantly available during the 3 h and the corresponding average power is produced during the whole time interval. In the end, for each of the $n_{lev} = 10$ altitude levels considered, the wind is represented by two parameters: i) the main wind velocity, which is the norm of the $u, v$ wind speed vectors, and ii) the wind direction as a deviation from the main direction ($x$-axis).

Fig. 1 shows the wind data processing for an illustrative wind profile. The processing of the $u, v$ wind vectors in the Euclidean plane between 0 and $2\pi$ (= East) is shown on the left. The parameter $\bar{d}_{\text{ts}}$ denotes the main wind direction at time $t_s$, and $d_{\text{ts}}$ denotes the deviation from this main wind vector at each altitude level. In the AWE model, these two parameters are transferred to the Cartesian coordinate frame. The average wind direction $\bar{d}_{\text{ts}}$ is defined to be along the $x$-axis, and the deviation $d_{\text{ts}}$ enters as an orthogonal wind vector on the $y$-axis. Thus, the wind data are implemented as a multi-dimensional parameter $\mathbf{p}_{\text{w}} \in \mathbb{R}^{2\times 10}$ in the NLP.

On the right, Fig. 1 illustrates the wind vectors in the Cartesian coordinate frame as implemented in the model.

In the model (2b), the wind conditions have to be defined over the whole altitude range, and the numerical tools used to solve (2b) operate best if all functions in (2b) are at least twice continuously differentiable. Hence, a smooth interpolation of the vertical wind data is proposed. In the following, a Lagrange polynomial function

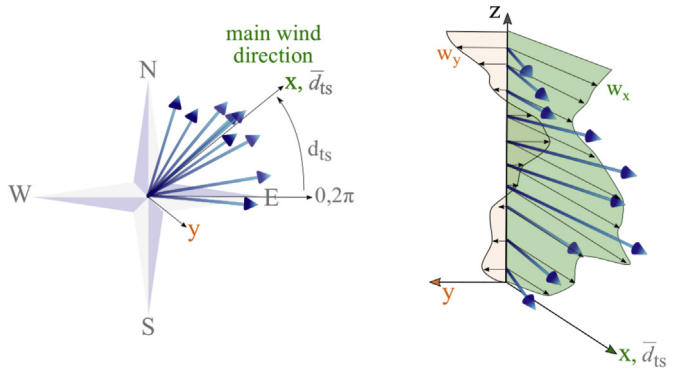



**Fig. 1.** Illustration of wind-data processing. Wind data are given as $u$ and $v$ components for a single time point for each of the ten altitude levels. The concepts main wind direction and angular deviation are illustrated in the Euclidean plane, for each altitude (left). In the model, the main wind direction $\bar{d}_{ts}$ (here, illustratively north-east) is aligned with the $x$-axis of the earth frame. The wind speeds are then expressed as a $\mathbf{p}_x$ wind component, and the deviations are represented as orthogonal vectors in the y-axis $\mathbf{p}_y$ (right).

is chosen for the interpolation, which goes through each data point. In order to prevent unrealistic interpolations due to over-fitting the wind data, a regularization to the polynomial is added. For the $x$ and $y$ wind parameters, Lagrange polynomials $P_{\mathscr{L},x}$ and $P_{\mathscr{L},y}$ are given as:

$$P_{\mathscr{L}}(z) = \sum_{i=0}^{n_{lev}=10} \theta_i \cdot \prod_{\substack{k=1 \\ k\neq i}} \frac{z - z_k}{z_i - z_k} \tag{3a}$$

$$\boldsymbol{\theta} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \quad \frac{1}{2}\|P_{\mathscr{L}}(\mathbf{z}, \boldsymbol{\theta}) - \mathbf{p}_{\mathbf{w}}\|^2 + k \cdot \left| \frac{\partial^2 P_{\mathscr{L}}(\mathbf{z}, \boldsymbol{\theta})}{\partial \mathbf{z}^2} \right|^2, \tag{3b}$$

where $\mathbf{p}_w = \{\mathbf{p}_x, \mathbf{p}_y\}$ and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_x, \boldsymbol{\theta}_y\}$ hold the parameters related to $x$ or $y$ components. Eq. (3b) computes the parameters $\theta_i$, such that they are close to the actual wind speed data at height $z$, but at the same time limit the second derivative of the polynomial of Eq. (3a) with the weighting factor $k$. The weighting factor is chosen such that the difference between the resulting polynomial parameters and the original data is minimal.

An example of interpolation of the $x$-wind component can be seen in Fig. 2, in which four different wind profiles are plotted. The illustration shows that the standard Lagrangian polynomial (dashed line), that goes directly through the original wind data points $\mathbf{p}_x$ are heavily over-fitted. As a contrary, the regularized Lagrangian polynomial shows a much more realistic wind shear and it goes also directly through each wind data point. Since the difference is barely visible we can state that $\boldsymbol{\theta}_x \approx \mathbf{p}_x$. Given that the wind data are also modeled and based on assumption the authors assume that this modification of the wind data is negligible. Alternative techniques, e.g., cubic B-splines, can be used.

### 2.3. Background on solving an NLP with the primal-dual interior-point method

In order to solve the formulated OCP (2), the problem is discretized with the direct collocation method into $n = 40$ time intervals with $d = 3$ collocation points [30]. The resulting NLP

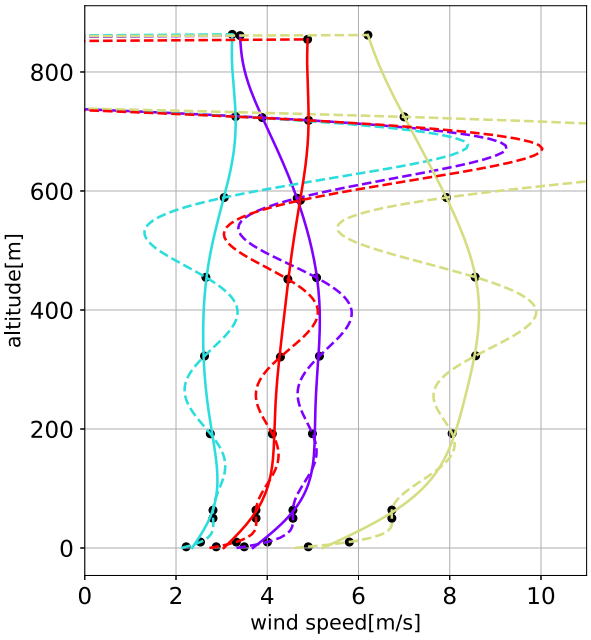



**Fig. 2.** Wind speed profiles in their main wind direction as polynomial functions of the altitude, illustrated for four different time instances. The black dots in a profile represent the wind parameters $\mathbf{p}_x$. The dashed lines illustrate the Lagrange interpolation ($k = 0$) using those points as polynomial parameters. The respective solid lines represent the smoothed Lagrange polynomial $P_{\mathscr{L}}$ with respect to the second derivative ($k = 10^4$). Despite the smoothing, the new Lagrangian parameters are $\boldsymbol{\theta}_x \approx \mathbf{p}_x$.

$$\min_{\mathbf{w}} \quad \Phi(\mathbf{w}, \mathbf{p}) \tag{4a}$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{w}, \mathbf{p}) = 0 \tag{4b}$$

$$\mathbf{h}(\mathbf{w}, \mathbf{p}) \leq 0, \tag{4c}$$

is implemented in Python within the nonlinear optimization and optimal control framework CasADi [31]. The cost function (4a) is minimized with respect to the decision variables $\mathbf{w} \in \mathbb{R}^{2103}$ and the parameters $\mathbf{p} \in \mathbb{R}^{2710}$. Equations (4b) and (4c) present the equality and the inequality constraints, respectively.

The solver used in this work is IPOPT, a primal-dual interior-point (PDIP) algorithm tailored to large-scale NLPs [32]. The solver algorithm seeks solutions to (4) via solving the relaxed KKT (Karush-Kuhn-Tucker) conditions [30]:

$$\nabla_{\mathbf{w}}\Phi(\mathbf{w}, \mathbf{p}) + \nabla_{\mathbf{w}}\mathbf{g}(\mathbf{w}, \mathbf{p})\boldsymbol{\lambda} + \nabla_{\mathbf{w}}\mathbf{h}(\mathbf{w}, \mathbf{p})\boldsymbol{\mu} = 0 \tag{5a}$$

$$\mathbf{g}(\mathbf{w}, \mathbf{p}) = 0 \tag{5b}$$

$$\mathbf{h}(\mathbf{w}, \mathbf{p}) + \mathbf{s} = 0 \tag{5c}$$

$$\operatorname{diag}(\mathbf{s})\boldsymbol{\mu} - \tau\mathbf{1} = 0 \tag{5d}$$

under the condition $\mathbf{s}, \boldsymbol{\mu} > 0$, where $\tau$ is the relaxation or barrier parameter, $\mathbf{s}$ the slack variable, and $\{\boldsymbol{\lambda}, \boldsymbol{\mu}\}$ the dual variables [32]. For $\tau = 0$, the classical KKT conditions associated with (4) are recovered, such that the complementary slackness conditions (5d) combined with (5c) become

$$\operatorname{diag}(\mathbf{s})\boldsymbol{\mu} = -\boldsymbol{\mu}^{\top}\mathbf{h}(\mathbf{w}, \mathbf{p}) = 0. \tag{6}$$

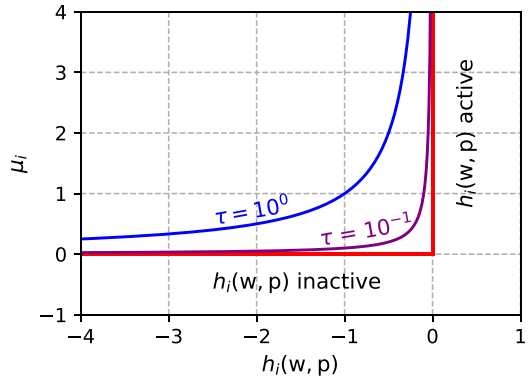



**Fig. 3.** Solution manifold of the original complementary slackness conditions $\mu_i h_i(\mathbf{w}, \mathbf{p}) = 0$ plotted in red, and the relaxed conditions $\mu_i h_i(\mathbf{w}, \mathbf{p}) + \tau = 0$ in blue and purple for different values of $\tau$. The smaller the value of $\tau$, the more accurate the solution. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

For $\tau > 0$, the solution of (5) approximates the primal-dual solution of the original problem on the order of $\tau$. The non-smooth complementary slackness conditions $\boldsymbol{\mu}^\top \mathbf{h}(\mathbf{w}, \mathbf{p}) = 0$ are smoothed in (5d) via the relaxation parameter $\tau$. The manifold of the complementary slackness conditions (5d) is visualized in Fig. 3.

In order to compute the solutions to the relaxed KKT conditions (5) efficiently, a Newton iteration is typically used, starting from an initial guess for $\mathbf{w}$, $\lambda$, $\boldsymbol{\mu}$, $\mathbf{s}$ [20]. Solvers like IPOPT solve (5) for large values of $\tau$ first and then decrease $\tau$ in a homotopy strategy towards very small values of $\tau$ [32].

In order for a solution point $\mathbf{w}^\star$ to be a local optimum, the KKT conditions (5) have to be fulfilled and a regularity condition called Linear Independence Constraint Qualification (LICQ) has to be satisfied, as well as the Second-Order Sufficient Condition (SOSC) of optimality. LICQ requires that, at $\mathbf{w}^\star$, the gradients of the equality constraints $\nabla \mathbf{g}^\star$ and the gradients of the active inequality constraints $\nabla \mathbf{h}^\star$ are linearly independent. The SOSC is fulfilled for a minimum if

$$\mathbf{d}^\top \nabla^2 \mathscr{L} \mathbf{d} > 0, \quad \text{for any feasible} \quad \mathbf{d} \neq 0, \tag{7}$$

i.e., the Hessian of the Lagrangian is positive in all feasible directions $\mathbf{d}$, defined by the joint null-spaces of the Jacobian of the equality constraints and strictly active inequality constraints. If the NLP (4) satisfies SOSC and LICQ, and if all constraints are strictly (in) active, the accuracy of the solution for (5) and the one for (4) is on the order of $||\mathbf{w}_\tau^\star - \mathbf{w}^\star|| = \mathscr{O}(\tau)$ [32], where $\mathbf{w}_\tau^\star$ denotes the approximate solution.

### 2.4. Homotopy-path-following methods

The NLP resulting from the highly nonlinear periodic OCP (2) of an AWE system is rather complex to solve. Successful convergence to an optimal solution requires special solution approaches. One issue is that periodic OCPs that involve high-index DAE formulations and non-singular representations of the SO(3) Lie groups, as in (2), can cause severe difficulties for Newton-based solvers [33]. Indeed, the resulting NLP may fail the LICQ and SOSC conditions. This typically yields a poorly conditioned linear algebra in the neighborhood of the solution and jeopardizes the convergence of the Newton iterations. In order to ensure that LICQ and SOSC hold at the optimal solution, the formulation of the problem has to be modified in simple but counterintuitive ways. This problem and its solution are further detailed in Ref. [33].

The other issue relates to the high nonlinearity of the DAEs describing the dynamics of the AWE system. Solving the NLP requires an appropriate initial guess. In order to generate this initial guess, a *homotopy-path-following method* is introduced, which allows the gradual transformation of the OCP from a fairly simple problem into the original highly nonlinear one. A simpler OCP is solved first, and its solution serves as the initial guess for the next slightly modified problem until the originally complex problem is recovered. This procedure is detailed in Ref. [23] and in the following referred to as Homotopy-Path-initialization (HPinit).

In this paper, the complex OCP delivering the optimal trajectories of the AWE system is solved for a very large set of wind parameters $\mathbf{p} \subseteq \mathscr{P}$. The two issues detailed previously are tackled by implementing the solutions proposed in Refs. [23,33]. However, deploying the above-mentioned HPinit for every individual parameter in the set $\mathscr{P}$ would be very computationally inefficient. As an alternative, in order to process a sequence of parametric NLPs, an overlying homotopy path among the family of problems can be of help. This homotopy-path-following method for performing parameter studies might also be known as the *continuation method*, which is detailed in, e.g., Ref. [20]. Below, we cover the mathematical background of the homotopy-path-following method, tailored to reduce the computational burden of using an OCP formulation to compute the annual power production curve of an AWE system.

For the NLP (4), satisfying SOSC and LICQ for all $\mathbf{p} \subseteq \mathscr{P}$, the parametric primal-dual solutions provided by (5), labeled as

$$S_\tau(\mathbf{p}) = \{\mathbf{w}_\tau^\star(\mathbf{p}),\ \lambda_\tau^\star(\mathbf{p}),\ \boldsymbol{\mu}_\tau^\star(\mathbf{p}),\ \mathbf{s}_\tau^\star(\mathbf{p})\} \tag{8}$$

and implicitly defined by the relaxed KKT conditions (5), are smooth for all $\tau > 0$, and converge to piecewise-smooth functions as $\tau \to 0$ [34]. If LICQ and SOSC hold for all $\mathbf{p}$ in $\mathscr{P}$, then the homotopy path for $S_\tau(\mathbf{p})$ is unique, and for $\tau > 0$, the first order approximation

$$S_\tau(\mathbf{p} + \alpha \Delta \mathbf{p}) = S_\tau(\mathbf{p}) + \alpha \frac{\partial S_\tau(\mathbf{p})}{\partial \mathbf{p}} \Delta \mathbf{p} + \mathscr{O}\left(\alpha^2\right) \tag{9}$$

holds in a neighborhood of $\mathbf{p}$ ($\alpha = 0$). The size of the neighborhood where (9) holds decreases with the value of $\tau$. The sensitivities $\frac{\partial S_\tau(\mathbf{p})}{\partial \mathbf{p}}$ can be obtained inexpensively as a by-product of the Newton iterations using the implicit function theorem on the KKT conditions (5), i.e.,

$$\frac{\partial S_\tau(\mathbf{p})}{\partial \mathbf{p}} = -\frac{\partial \mathbf{R}(S_\tau(\mathbf{p}), \mathbf{p})}{\partial \mathbf{z}}^{-1} \frac{\partial \mathbf{R}(S_\tau(\mathbf{p}), \mathbf{p})}{\partial \mathbf{p}}, \tag{10}$$

where $\mathbf{R}(\mathbf{z}, \mathbf{p})$ collects the KKT conditions (5) and $\mathbf{z} = \{\mathbf{w}, \lambda, \boldsymbol{\mu}, \mathbf{s}\}$ collects the primal-dual and slack variables involved in (5).

The method of computing the consecutive solution $S_\tau(\mathbf{p}_+)$ by means of the first order predictor (9) is called the *tangent continuation method*. The value of $\alpha$ then serves as a scaling parameter for the distance $\Delta \mathbf{p}$ between $\mathbf{p}$ and the consecutive parameter $\mathbf{p}_+$, i.e., $\alpha = [0, 1]$. An algorithm can be constructed to evaluate $S_\tau(\mathbf{p}_+)$ if the solution $S_\tau(\mathbf{p})$ is available:

**Algorithm:** PDIP homotopy-path-following

**Input:** $\mathbf{p}_+$, $\mathbf{p}$, $S_\tau(\mathbf{p})$, $\Delta\alpha > 0$
Set $\alpha = 0$, and $\Delta\mathbf{p} = \mathbf{p}_+ - \mathbf{p}$
**while** $\alpha < 1$ **do**
  Update $\alpha_+ \leftarrow \min\{1, \alpha + \Delta\alpha\}$
  Compute:

$$S_\tau^{\text{init}} = S_\tau(\mathbf{p}) + (\alpha_+ - \alpha)\frac{\partial S_\tau(\mathbf{p})}{\partial \mathbf{p}}\Delta\mathbf{p} \tag{11}$$

  Set $\mathbf{p} \leftarrow \mathbf{p} + (\alpha_+ - \alpha)\Delta\mathbf{p}$, and $\alpha \leftarrow \alpha_+$
  Use $S_\tau^{\text{init}}$ as a initial guess for solving $S_\tau(\mathbf{p})$
**return** $= S_\tau(\mathbf{p}_+) = S_\tau(\mathbf{p})$

The *PDIP homotopy-path-following* algorithm is guaranteed to converge for $\Delta\alpha$ small enough and if all intermediate problems fulfill SOSC and LICQ. The admissible step size $\Delta\alpha$ decreases if $\tau$ decreases or if $||\mathbf{p}_+ - \mathbf{p}||$ increases. A small $\Delta\alpha$, does not substantially affect the performance of the algorithm. Clearly, the closer $\mathbf{p}_+$ is to $\mathbf{p}$, the faster the algorithm returns $S_\tau(\mathbf{p}_+)$, as the step size $\Delta\alpha$ can be increased if the distance $||\mathbf{p}_+ - \mathbf{p}||$ decreases. Typically, for sufficiently small $||\mathbf{p}_+ - \mathbf{p}||$, the choice $\Delta\alpha = 1$ is possible. These characteristics are true for the parameters in this study as the wind data are fairly dense. In practice, it also can be preferable to simplify the update (11) to $S_\tau^{\text{init}}$=$S_\tau(\mathbf{p})$, which yields the simpler classical continuation method. The solutions for the parameter set $\mathscr{P}$ are computed in this study using the simpler classical continuation method, i.e., with the initial guess $S_\tau^{\text{init}} = S_\tau(\mathbf{p})$. However, if there are gaps in the wind data, the general PDIP path-following algorithm may be advantageous or the parameters may have to be clustered to obtain a dense data set within each cluster. In order to exploit a homotopy-path-following method in the context of interior-point methods, the relaxation parameter of the PDIP solver IPOPT should be kept fixed. Additionally, for accurate solutions to the OCP (4), a small value of $\tau$ ought to be chosen. However, a small value of $\tau$ also typically requires a small $\Delta\alpha \ll 1$, impeding the PDIP path-following algorithm. There are two options for circumventing this issue:

1. Run the PDIP homotopy-path-following algorithm using a "large" $\tau = \tau_0$ and compute refined solutions $S_{\tau_1}(\mathbf{p}_+)$ with $\tau_1$ small at the end of the algorithm.
2. Use a fixed value of $\tau$ that delivers an accuracy of the solutions $S_\tau$ sufficient for practical purposes, while allowing the PDIP homotopy-path-following algorithm to perform decently.

The first option presents a more refined solution to the OCPs and is used in other work [21]. However, given that the data contain uncertainties and that the AWE model includes assumptions and simplifications, we here assume that a fixed value of $\tau$ is sufficient for the aim of the study.

In general, the optimal sorting strategy for the parameters cannot be defined without first solving the NLPs. In the context of the predictor-based method (11), a general way of assessing the distance between the current solution $S_\tau(\mathbf{p})$ corresponding to some given parameters $\mathbf{p}$ and another solution $S_\tau(\mathbf{p}_+)$ is via assessing the residual of the KKT conditions evaluated at the guess provided by (11). More specifically, one can use as a distance between $\mathbf{p}$ and $\mathbf{p}_+$

$$\left|\left|\mathbf{R}\left(S_\tau(\mathbf{p}) + \frac{\partial S_\tau(\mathbf{p})}{\partial \mathbf{p}}(\mathbf{p}_+ - \mathbf{p}), \mathbf{p}_+\right)\right|\right|, \tag{12}$$

which requires evaluating the KKT conditions (5) for (at least) the data points of neighboring $\mathbf{p}$. Neighboring $\mathbf{p}$ can, e.g., correspond to a small Euclidean distance between the parameters $||\mathbf{p} - \mathbf{p}_+||$.

When no tangential predictor is used, i.e., when $S_\tau^{\text{init}} = S_\tau(\mathbf{p})$, a cheaper measurement of the distance between the latest wind profile $\mathbf{p}$ and another wind profile $\mathbf{p}_+$ can be constructed using:

$$||\mathbf{R}(S_\tau(\mathbf{p}), \mathbf{p}_+)|| = \left|\left|\left.\frac{\partial \mathbf{R}}{\partial \mathbf{p}}\right|_{S_\tau(\mathbf{p}),\mathbf{p}}(\mathbf{p}_+ - \mathbf{p})\right|\right| + \mathscr{O}\left(||\mathbf{p}_+ - \mathbf{p}||^2\right). \tag{13}$$

Evaluating the left-hand side of (13) requires a possibly expensive evaluation of the KKT conditions (5) at a new parameter $\mathbf{p}_+$, while evaluating the first term in the right-hand side of (13) simply requires a matrix multiplication, which is arguably adequate for deployment within a large-scale ordering algorithm.

### 2.5. Implementation of method

In this study, three months of wind data at a three-hourly averaged resolution from September 1, 2017, to November 30, 2017, from the Gothenburg archipelago in Sweden, is used as an illustrative parameter set $\mathscr{P}$, yielding $n_{\text{ts}} = 728$ parametric NLPs to be solved. The individual NLPs describe the power optimization of a drag-mode AWE system for one vertical wind profile, which is defined via the wind coefficients collected in the parameter $\mathbf{p} \subseteq \mathscr{P}$. The parameter $\mathbf{p}$, which holds the coefficients representing the wind profile for a single time instance, is multidimensional, including the wind speed and the deviation from the main direction for each of the $n_{\text{lev}} = 10$ altitude levels, i.e., $\mathbf{p}_{\text{w}} = \{\mathbf{p}_{\text{x}}, \mathbf{p}_{\text{y}}\} \in \mathbb{R}^{2\times n_{\text{lev}}}$ (see Section 2.2). In the model, the parameters form a polynomial wind profile for each time instance. Drag-mode systems have a constant tether length and thus a rather consistent and limited variation in average operational altitude. As a result, the wind speed at the average operational altitude correlates relatively well to the resulting power output and is hence a determining factor of the optimal solution. Consequently, the wind speed at the average operational height of the drag-mode AWE system is used as an intuitive sorting parameter for the wind parameter set $\mathscr{P}$, in order to obtain a small distance between $S_\tau(\mathbf{p})$ and $S_\tau(\mathbf{p}_+)$. The average operational altitude for the studied system is $\sim 300\,m$, which is extracted from previous results of the OCPs. Note that in the case treated in this paper, a reasonable ordering of the parameters could be defined without applying the technique described at the end of Section 2.4. However, the techniques described are generally applicable and may be helpful for more complex AWE problems, e.g., where the flight altitude is more variable.

The three-hourly-averaged wind data typically do not change drastically from one time instance $\mathbf{p}_t$ to the next $\mathbf{p}_{t+1}$, so a chronological ordering may already be a natural sorting of the data for a

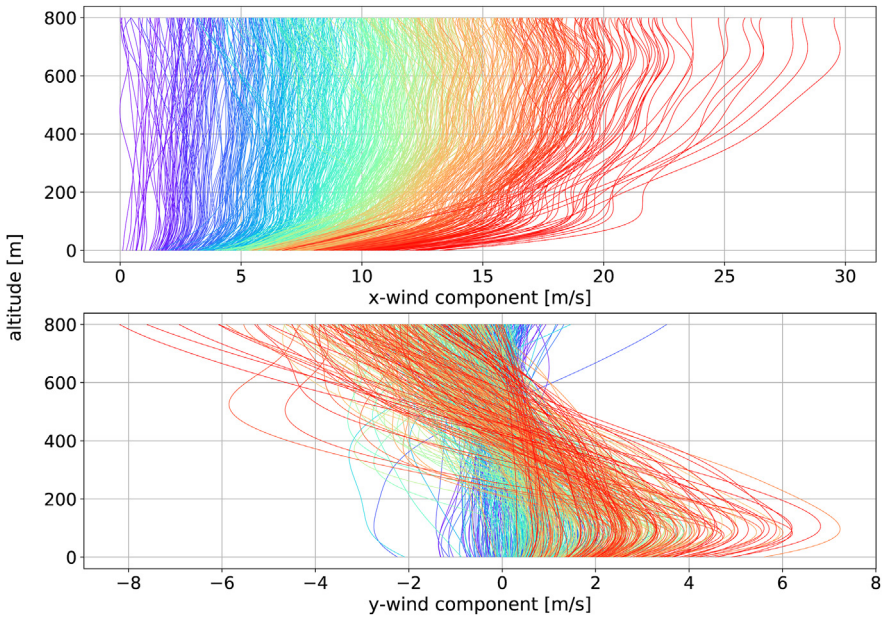



**Fig. 4.** Wind speed profiles as functions of the altitude for the Gothenburg archipelago. Top: Wind speeds in the individual profile's main wind direction, corresponding to the model's $x$-axis. Bottom: Orthogonal parts of the wind speed profiles, collecting the wind speeds that deviate from the individual profile's main wind direction. Wind profiles in top figure are sorted by wind speed at the average operational height of 300 m, which is the wind speed labeled on the $x$-axis. The polynomial profile curves result from the Lagrange interpolation (Eq. (3)) described in Section 2.2, with each curve based on $n_{\text{lev}} = 10$ data points at different altitudes. The colors are for a clearer illustration and to show that the high wind speed profiles (red) are linked to the wide ranging y-wind component values, i.e. great changes in wind direction over the altitude range. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

path-following method. However, wind profiles reoccur and patterns repeat during the course of a year, hence a reordering can be beneficial.

Fig. 4 shows the wind profiles sorted by the absolute wind speed at 300 m (average operational altitude). Note the dense wind data and the gradual transition from low to high main wind speeds in the top plot, and the corresponding orthogonal wind speeds in the bottom plot. The figure also shows that, for altitudes above 100 m, the logarithmic approximation of the profile is not always valid, with wind maxima occurring at various altitudes.

After sorting, the wind data may not yield a dense data set. If random wind data samples or data from locations with particular weather phenomena are used, the wind data set can appear less smooth. As a result, larger gaps can be found in the data, and the sorting technique might not be sufficient for obtaining an overall dense data set. If this is the case, and the difference $\|\mathbf{p}_+ - \mathbf{p}\|$ grows too large, the more genuine method of tangent continuation has to be applied, i.e., steps with the scaling parameter $\alpha$ have to be added between the individual parameters $\mathbf{p}$ and $\mathbf{p}_+$. Otherwise, if the classical continuation strategy is to be used, another strategy is to first cluster the wind data into characteristic profile shapes before sorting by average wind speed. This assigns the parameters to different clusters and the homotopy-path-following method is then applied and initialized individually in each cluster. For demonstration purposes, the clustering method is tested with the parameter set $\mathscr{P}$ used in this study. The data set is divided into seven clusters according to two criteria: i) the mean wind speed and ii) the span of total variation in wind direction over the ten altitude levels, $\max(d_{ts}) - \min(d_{ts})$, at a single time instance. The k-means clustering method is used to create the different clusters. For each cluster, the optimal solution of the NLP (4) for a simple logarithmic wind profile with similar winds as the wind data in the cluster yields the first initial guess $S^{\text{init}}_{\tau,0}$. Due to the characteristics of the specific data set used here, the clustering results in a representation with almost the same order of parameters as when sorted by the wind speed at average operational height. Therefore, in what follows, the analysis is performed with the simple parameter sorting. As mentioned above, the circumstances can be different in other locations.

So far, we have covered the background on solving the NLP (4) and on the homotopy-path-following method for parametric analyses in Sections 2.1, 2.3, and 2.4. We have also presented the study-specific wind parameters and how they are handled in the NLPs. Below, we present the entire algorithm that applies the homotopy-path-following method to solve the family of NLPs, with the wind-parameter set $\mathscr{P}$. The classical continuation method is applied. The relaxation parameter $\tau$ is fixed to $\tau_0$, a value between $10^{-3}$ and $10^{-8}$, and the parameters are sorted by the wind speeds at 300 m altitude. For brevity, we call the algorithm the Homotopy Path for big data (HPbd):

**Algorithm:** Homotopy Path for big data (HPbd)

**Input:** $\mathcal{P}$, $S_0$
Set $\tau = \tau_0$
Set $S^{\text{init}} = S_0$
**for** $t$ *in* $\mathcal{T}$ **do**
  Use $S^{\text{init}}$ as a initial guess for solving $S_\tau(\mathbf{p}_t)$
  Update $S^{\text{init}} \leftarrow S_\tau(\mathbf{p}_t)$
**return** $S_\tau(\mathcal{P})$

The solution $S_0$ is the initial guess at the solution of the NLP, and $\mathscr{T}$ is the set of ordered time steps referring to the wind speeds from high to low, as shown in Fig. 4 from right to left. The first initial guess $S_0$ holds the optimal solution for a high wind-speed profile. It is generated in advance using HPinit to ensure the convergence to the solution of the highly nonlinear program. Fig. 5 shows a schematic of the entire procedure including the initialization with HPinit and the parameter computation with HPbd.

The case shown has multiple clusters. Each cluster requires a good initial guess $S_0$ for the respective wind cluster before HPbd is applied. After finalizing all parameters in one cluster, the next cluster gets initialized. If the parameter set $\mathscr{P}$ is not clustered, $c_k$ only exists for k = 1.

It is important to note that for low wind speeds, the convergence tends to be problematic as power production is physically impossible below a certain wind speed, which implies that the solver generates solutions with meaningless trajectories. A persistent failure in finding a meaningful solution is classified as a situation where the AWE system should idle, i.e., the wing should be landed, and the power output is zero. HPbd is applied to solve the OCP for the wind parameter set $\mathscr{P}$ with 728 time steps. As mentioned above, the original chronologically ordered data might already serve as a well-sorted parameter set, but a re-ordering can further reduce the number of overall iterations in the solver and ensure a successful homotopy-path-following implementation.

In order to obtain an estimation of the computation times, the time for solving the NLP with HPinit is compared to the time needed when applying HPbd. Additionally, using HPbd, the original chronological ordering of the parameters is compared to the sorting strategy, where the ordering is based on the average wind speed at the operational altitude of 300 m. A first comparison is performed at a fixed $\tau = 10^{-4}$. The resulting difference in computational performance is reported. In a second comparison, HPbd is applied to solve the sequential OCPs using different fixed values for τ. The solution accuracy and the computation times are compared. At low wind speeds, the solver is likely to struggle. In order to obtain a fair comparison between the different relaxation parameters τ, this comparison is performed on the upper half of the sorted wind data, corresponding to greater wind speeds.

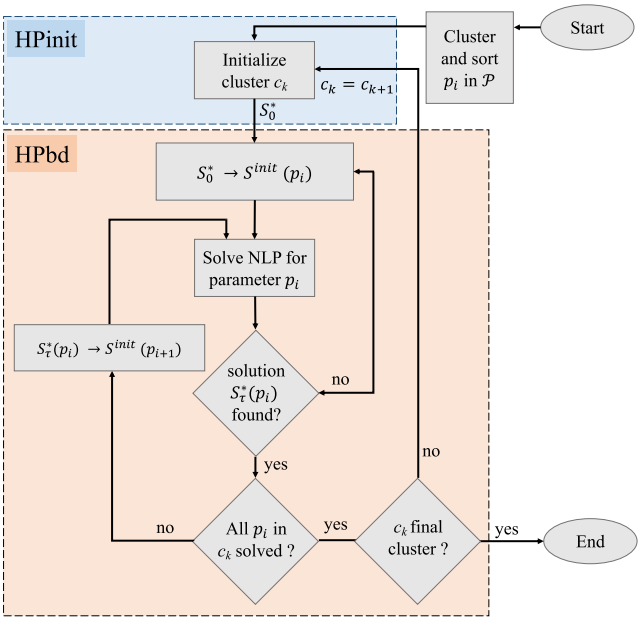



**Fig. 5.** Schematic of the entire procedure of applying the two homotopy-path-following algorithms to initialize and solve the NLPs sequentially for the clustered and sorted parameters. In this analysis no clustering is applied, so $c_k$ only exists for k = 1.

### 2.6. Estimation based on machine learning using regression models

For the goal of assessing the potential of large-scale AWE development, a large number of geographic locations has to be analyzed. Applying the proposed homotopy-path-following method as implemented in HPbd already provides a reasonably short computation time. However, this OCP approach to computing the potential grows tedious if the analyzed region is large, e.g., when comprising more than tens or hundreds of sites. One option, then, is to estimate the optional power production with a function approximation from statistics and machine learning.

In order to obtain the power production for a very large number of sites, we thus propose a machine learning approach that uses regression models. The idea is that if a number of power production profiles are available—computed with the previously presented optimization method—they can be used to learn a function that mimics the optimization method with reasonable accuracy. Obviously, some accuracy is sacrificed, but a great deal of computational speed could be gained in return. In order to find a regression model that predicts the optimal average power production based on the wind-feature parameters $\mathbf{p}_w$ well, different regression models are cross-validated and compared. Cross-validation is used to analyze the accuracy of the models by holding out parts of the data in the training phase for testing how the model generalizes to previously unseen data [35]. The different regression models are chosen with an increasing and heterogeneous capability to fit nonlinear functional relationships: starting with linear regression models, moving on to a k-nearest neighbor (k-NN) method, a support vector regression (SVR) model, and a decision-tree-method (random forest) model. In order to establish a baseline, the linear regression models used are expected to perform less optimally due to the nonlinearity of the optimization. By establishing a baseline, we can evaluate if and how well different models (suitable for nonlinear relationships) improve performance when predicting average power for new geographic locations.

The different linear regression models are implemented based on the minimization of the general function

$$\|y - \mathbf{X}\boldsymbol{\beta}\|^2 + \alpha_1\|\boldsymbol{\beta}\|_1 + \alpha_2\|\boldsymbol{\beta}\|^2,$$

where $\mathbf{X}$ contains the input variables (here: wind features), $\boldsymbol{\beta}$ the regression coefficients and $y$ the response variable (here: power output). The coefficients $\alpha_1, \alpha_2$ are chosen dependent on the linear regression method. The standard linear regression is in fact the least squares method, since the coefficients $\alpha_1, \alpha_2 = 0$, i.e., the two norms L1 and L2 are not included. The ridge regression adds a shrinkage of the coefficients by adding the L2 norm, i.e, $\alpha_1 = 0$, $\alpha_2 > 0$, in order to provide regularization. The lasso (least absolute shrinkage and selection operator) regression includes the L1 norm to eliminate unnecessary variables, i.e, $\alpha_1 > 0$, $\alpha_2 = 0$, while the elastic net regression combines the norms such that $\alpha_1, \alpha_2 > 0$. More background information can be found in Refs. [36,37]. The k-nearest neighbor model interpolates the data locally by weighting the neighboring data point dependent on the (for instance) Euclidean distance. Thus, the closest neighbors contribute the most to the solution [38]. The support vector regression (SVR) tries to find a hyperplane that represents all data with no error greater than a parameter ε [38]. Finally, the random forest regression model includes multiple decision trees that independently evaluate sampled

**Table 1**
Regression models that are evaluated on their performance predicting the average power output of the AWE system based on the wind speed data. The hyperparameters that are optimized via a grid search for the individual models are listed.

| regression model | | optimized hyperparameters |
|---|---|---|
| Linear regression models | linear regression | – |
| | lasso | $\alpha_1$ |
| | ridge | $\alpha_2$ |
| | elastic-net | $\alpha_1$, $\alpha_2$, $L_1$-$L_2$-ratio |
| Instance-based | Support vector regression | $C$-penalty, kernel function |
| | k-nearest neighbor (k-NN) | $n_{\text{neighbors}}$, weights |
| Decision trees | random forest | $n_{\text{trees}}$, min. sample{leaf, split}, max. depth |

parts of the data and then average the decisions to avoid overfitting or bias [39]. The tested regression models and the hyperparameters that are optimized during the evaluation are listed in Table 1.

The models are trained and compared with a nested k-fold cross validation to evaluate the performance of different models on out-of-sample prediction for unseen sites. In the nested k-fold cross validation, the data set is split into k subsets in order to obtain a segmentation into k different train and test splits. Some locations are left out of the training to leave them aside for testing. The training data are then further partitioned into splits for hyperparameter optimization. Finally, a set of model hyperparameters are chosen that generalize best on average. In this paper, model generalization is performed using the $R^2$ metric. This validation has an advantage over the single training and testing method, as it reduces the bias in the evaluation and gives a more truthful picture of the model performance to generalize to new data [40].

In this study, k = 10 folds are used on the data of 15 geographic locations (43,800 data points). The hyperparameters listed in Table 1 are optimized via a grid search for each model. The range of the individual grids is set to common reasonable values. The resulting best-fitting regression model is then trained on the data of 15 different European locations and tested with the data of the three-month wind data $\mathscr{P}$ used previously for the homotopy-path-following algorithm, HPbd. The predicted power production from the regression model is then compared to the results of the optimization, and the computation time is provided.

## 3. Results and discussion

In this section, the proposed homotopy-path-following method for processing big data (HPbd) is applied to solve a family of parametric NLPs (4) for the parameter set $\mathscr{P}\in\mathbb{R}^{728\times 20}$ presented in Section 2.2. The computation yields the average power production for Gothenburg in a three-hourly resolution over a three-month period. Further, different regression models are 10-fold cross-validated. The most suitable model is used to predict the power production profile based on the wind data $\mathscr{P}$. The predicted power profile is then compared with the solution of the OCP. The results are followed by a discussion.

### 3.1. Results of the homotopy-path-following method

At the very start, in order to initialize the algorithm HPbd, a first solution $S_0$, serving as initial guess, needs to be computed with the help of the initializing algorithm, HPinit. With HPinit, computing one parameter $\mathbf{p}$ takes approximately 21 $s$ on a standard computer. A computation of the whole family of NLPs by applying HPinit to each parameter $\mathbf{p}\in\mathscr{P}$ would correspondingly result in a very long computation time. In order to reduce this computation time for remaining parameters in a big wind data set $\mathscr{P}$, the family of parametric NLPs can be computed with the algorithm HPbd described in Section 2.5. Solving the $n_{ts}$ NLPs with the parameters in their original chronological order results in an average computation time of $\sim$ 1.58 $s/\mathbf{p}$. This delivers a significant reduction in computation time. If the parameter set is additionally sorted by the wind speeds at 300 m altitude before solving, the computation time is further reduced to $\sim$ 1.03 $s/\mathbf{p}$ . Hence, HPbd, in association with an appropriate sorting technique, offers a reasonable computation time. Table 2 summarizes the computation times using HPbd for the two different sorting approaches compared to using HPinit to solve for all parameters in the set. The shorter computational time stems from an overall reduction in the number of iterations required to solve the various NLPs.

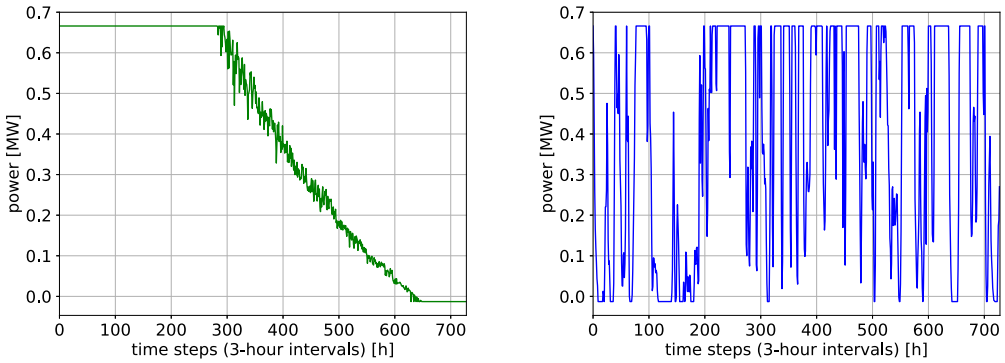
**Table 2**
Computational time needed to solve the $n_{\text{ts}} = 728$ NLPs. Computation times using the algorithm HPbd, with parameters either sorted by main wind speed at the average operational altitude of 300 m or in their original chronological order; or using the algorithm HPinit for all parameters. The relaxation parameter in the solver is fixed at $\tau = 10^{-4}$. HPbd is thirteen times faster per parameter than HPinit, and sorting the data shortens the computation time even more.

| | HPbd | | HPinit |
|---|---|---|---|
| parameter order | — **Case A** — **by wind speed** | — **Case B** — **chronological** | **chronological** |
| time/parameter [$s/\mathbf{p}$] | 1.03 | 1.58 | 21.01 |
| total time [$s$] | 750 | 1150 | ($\sim$ 15300) |

Solving the family of parametric NLPs with HPbd yields a solution set $\boldsymbol{S}_\tau\in\mathbb{R}^{n_{ts}}$. Each individual solution $S_\tau$ contains the average power production $P$ of the AWE system. The subscript $\tau$ denotes the solutions obtained with the fixed barrier parameter $\tau$.

As described above, the reason for sorting the parameters is to obtain a homotopy path along similar optimal solutions. In Case A, the parameters are sorted by the wind speeds at the operational altitude in order to result in similar optimal solutions for the average power production $P$ for the consecutive problems. In Case B, the parameters are left unsorted, following the variability of the wind availability. Fig. 6 shows the resulting power P in the same order as the parameters have been solved by the algorithm. Fig. 6a shows Case A, which has sorted parameters. The effect of the sorting is obvious in the reduced distance between a solution $P(\mathbf{p})$ and consecutive solution $P(\mathbf{p}_+)$. For comparison, Fig. 6b shows Case B. The varying wind pattern is clearly reflected in the consecutive parameters and the resulting fluctuating power output. Fig. 6 shows why the homotopy-path-following algorithm reduces the computation time as per Table 2.

Performing the computations at a barrier parameter $\tau$ above very small values yields a trade-off between the solution accuracy and the computation time. Hence, the adequate value is not obvious. For this reason, we next use different fixed values of $\tau$ to




(a) NLPs solved sorted by wind (b) NLPs solved sorted by time

**Fig. 6.** Power generation profile **P** of a drag-mode AWE system with a maximum capacity of 666 kW, computed by applying HPbd to the NLPs. On the left, the power is shown in the order of the sorted parameters of Case A. On the right, the power is shown in the original chronological order of the parameters, Case B. The reduced distance between the solutions $S_\tau(\mathbf{p})$ and $S_\tau(\mathbf{p}_+)$ in Case A is obvious.

**Table 3**
Results of applying HPbd for different values of fixed $\tau = [10^{-2}, \cdots, 10^{-8}]$. Computation time [$s$] and the accuracy error of the computed average power of the AWE system $\mathbf{P}_\tau$ are shown. The error is measured with respect to the reference power profile $\mathbf{P}_{-8}$ computed for $\tau = 10^{-8}$ as (14).

| $\tau$ | computation time [s] | accuracy error $\varepsilon_\tau$ [−] |
|---|---|---|
| $10^{-2}$ | 554 | $5.8 \cdot 10^{-4}$ |
| $10^{-3}$ | 571 | $5.5 \cdot 10^{-5}$ |
| $10^{-4}$ | 584 | $5.5 \cdot 10^{-6}$ |
| $10^{-5}$ | 782 | $8.1 \cdot 10^{-7}$ |
| $10^{-6}$ | 737 | $3.3 \cdot 10^{-7}$ |
| $10^{-7}$ | 750 | $1.4 \cdot 10^{-7}$ |
| $10^{-8}$ | 841 | – |

analyze the computation time and solution accuracy. For this analysis, the parameters are implemented as in Case A, sorted by wind speed at the operational altitude. When solving the OCP for low wind speeds, the power optimization problem might not be feasible, and the solver can run into conversion problems. As this aspect is not relevant for the study on the accuracy of the barrier parameter, wind parameters corresponding to very low wind speeds are left out.

The NLPs are solved for $\tau = \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}\}$. The solutions for $\tau = 10^{-8}$ serve as reference for computing the average absolute error of the accuracy as

$$\varepsilon_\tau = \frac{1}{n_{\text{ts}}} \sum_{t=1}^{n_{\text{ts}}} \left| P_{\tau,i} - P_{-8,i} \right|_1, \tag{14}$$

where $P_{-8,i}$ is the power production at $t = i$ solved for $\tau = 10^{-8}$, and $P_{\tau,i}$ the power production at $t = i$ solved for other values of $\tau$. The resulting computation times and errors are presented in Table 3.

Computation time trends up as $\tau$ decreases. This is due to the increasing nonlinearity of the relaxed KKT conditions (5) as $\tau$ decreases, which deteriorates the performance of the algorithm. With a small barrier parameter $\tau$, managing changes in the complementary slackness condition is more difficult because the solution manifold is less smooth, see Fig. 3. This becomes especially relevant if there are many active inequality constraints in the NLP. In this solved family of NLPs, the maximal power constraint $P \leq P_{max}$, maximal tether force constraint $F_{tet} \leq F_{tet,max}$ and minimal altitude

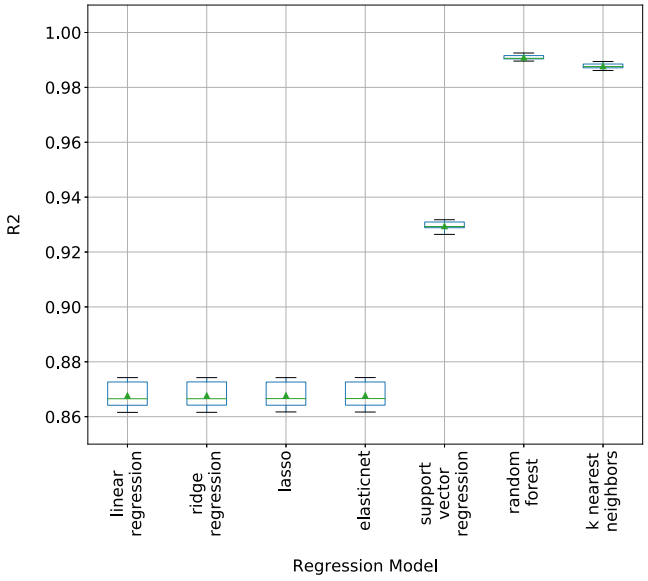



**Fig. 7.** Coefficient of determination value $R^2$ of the tested regression models. Each model is 10-fold cross validated and the hyper-parameters are optimized. The decision tree method of the random forest model presents the highest $R^2$ value of 0.997 and thus presents the best suitable method for this study case.

constraint $h \geq h_{min}$, for instance, are activated in a large number of cases. The accuracy error becomes smaller as the value of $\tau$ decreases. Considering modeling errors and data uncertainties, the errors are all relatively small for the choice of $\tau$ tested here. However, the error can become significant for $\tau \leq 10^{-2}$. In this study, $\tau = 10^{-3}$ or $\tau = 10^{-4}$ offer a short computation time and small error and seem to be the best choices. This observation may, however, not apply to other types of AWE system models. A fast approach to estimating the power yield has been presented in Ref. [7]. However, the approach assumes a logarithmic wind profile and limits the operational range to a prescribed trajectory. Here, we instead take detailed wind speed profiles into account and optimize the trajectory for the individual profile. The results of the NLPs confirm that for maximal power production, the average flight altitude and thus the trajectory are adjusted based on the current wind profile. Thus, the computation via OCP offers insight on

varying optimal operational altitudes at different locations.

### 3.2. Results of the regression model approach

In order to reduce the computation time of the AWE power estimation further for larger wind data sets, we propose the use of a regression model as function approximation. In total, seven different regression models, listed in Table 1, are evaluated via a 10-fold cross validation. The training data are obtained by applying HPbd to 15 geographic locations in Europe using a fixed barrier parameter, $\tau = 10^{-4}$. For each fold, the individual hyperparameters of the models are optimized via the grid search technique by a nested folding based on the training data. The models are evaluated in terms of the value of the coefficient of determination ($R^2$), shown in Fig. 7. A $R^2$-value of 1 would present the ideal case of a perfectly accurate model.

As the figure shows, the linear regression models seem to perform the worst at predicting the optimal power production. The more flexible models, SVR, random forest, and the k-nearest neighbor regression model, perform well, though the random forest model stands out with the best $R^2$-value. The following sets were tested as hyperparameters for the random forest method: $n_{\text{trees}} = [10, 100, 200]$, minimum sample leaf $= [1,3]$, and the optimal minimum sample split $= [2,4]$. The optimal hyperparameters used for Fig. 7 are $n_{\text{trees}} = 200$, minimum samples required per leaf $= 1$, and minimum samples required for split $= 2$.

In the following, the performance of the random forest model is tested by training it on 15 locations and then testing it on a new unseen location, the previously used three-month wind data $\mathscr{P}$ of Gothenburg, shown in Fig. 6. Fig. 8 shows predicted power profile of the test site in green and the corresponding power profile obtained from the NLPs in blue.

As can be seen, the power production predicted by the model matches the value computed by the NLPs fairly well. Computing the mean absolute error as (14) with the data plotted in blue as a base case results in $\varepsilon = 4{\cdot}10^{-3}$. With the regression model, the computation time for the $n_{ts} = 728$ wind data points is in total $0.067s$, so $\sim 1{\cdot}10^{-4}\ s/\mathbf{p}$. This is four orders of magnitude faster than computing the power production with HPbd. Taken together, this shows that with the random forest regression model it is possible to increase the speed of the power computation drastically, while maintaining high accuracy.

### 3.3. Discussion

In order to obtain a reasonable computation time for solving the NLPs that estimate the power production of an AWE system for a large spatio-temporal data set, a homotopy-path-flowing method is proposed. With the algorithm HPbd (Homotopy Path big data), an AWE power production profile can be computed for a big data set without initiating the complex NLP for every new parameter in the data set. By applying the homotopy-path-following algorithm, the average computation time per parameter is significantly reduced from initially 21.01 $s/\mathbf{p}$ to 1.58 $s/\mathbf{p}$ for an unsorted chronological ordering of the parameters. By additionally sorting the parameters by the average wind speed at the operational height before implementing HPbd, the computation time is reduced by another 30%. If a random forest regression model is trained on a few NLP solutions and the power production profile of a site is predicted, the computation time per parameter is sped up enormously to $\sim 10^{-4}\ s/\mathbf{p}$. However, this computation time per parameter comes with prior work, as the regression model is obviously no complete workaround to the computation of the NLPs. Before being able to use the random forest model to estimate the power, 10 to 20 sites must first be solved with HPbd to obtain a training set for the regression model. The greater the wind-data variance and the greater the solution accuracy sought, the more data need to be used for the training. As this regression model setup is indeed time costly, for any planned case the usefulness of the regression-model approach should be evaluated. Furthermore, the power production computation with the regression model provides no information on operational altitudes or other model variables.

The choice of what parameter sorting to use with HPbd depends on the study case. In the current study, the drag-mode AWE system has a relatively constant average flight altitude, which makes the choice of simply sorting the parameters by the wind speed at that altitude a very effective approach. For AWE systems with greater variation in operational altitude, the notion of average wind speed at the operating altitude may not apply. Thus, the basic wind data sorting strategy applied here may not be effective when using HPbd for other types of AWE systems. In those cases, more sophisticated sorting algorithms based on the general ordering technique presented in Section 2.4 may be necessary. When there are larger gaps in the wind data, a clustering of the wind profiles might be beneficial. The clustering of wind profiles into multiple classes generates

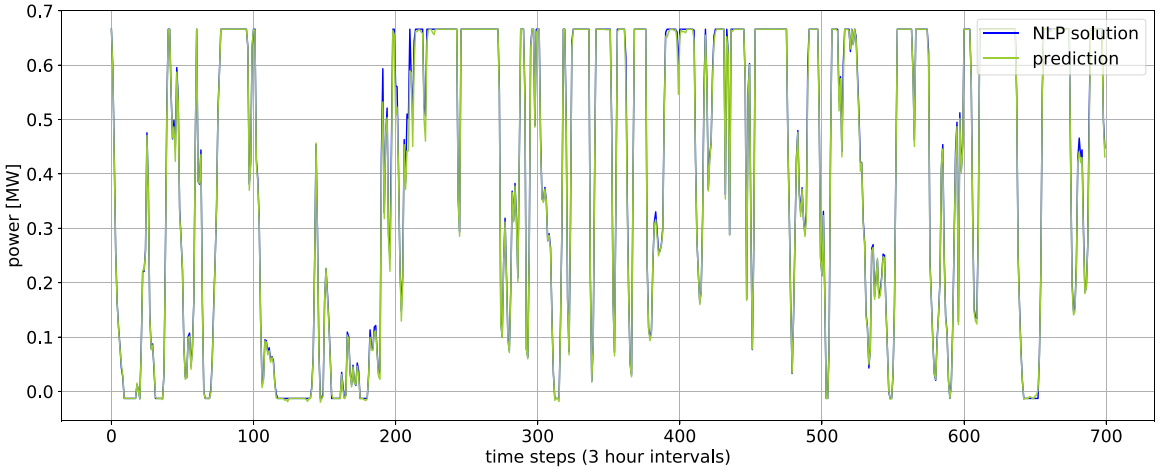



**Fig. 8.** Prediction of the power profile at an unseen site with the random forest regression model. Note, that for this system the power production is physically constrained at 666 kW. Power profile prediction based on the Gothenburg three-month, three-hourly-resolved wind data in green. The corresponding optimal solutions of the NLPs are shown in blue. The mean absolute error results in $\varepsilon = 4{\cdot}10^{-3}$. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

individual dense subsets of data, which can then be handled separately by HPbd. In this study, the initial wind data present a dense data set without significant gaps, see Fig. 4, and thus clustering does not reduce the computation cost of the algorithm in this case.

By lowering the barrier parameter $\tau$, the computation time is increased and the accuracy error decreased. This tendency could be different for other AWE system models, in which a higher number of inequality constraints are active at the solution. If more constraints are active, the effect of the nonlinearity complementary slackness conditions becomes more severe for low $\tau$, and the computation time increases accordingly. Also, the actual computation time is highly dependent on the type of computer used.

In general, all the errors in accuracy for the different NLP solutions are very small. If the regression model is used to estimate the power production, accuracy is sacrificed for a shorter computation time. The accuracy of the regression model ought to improve with the size of the training set, which requires more NLPs to be solved in the first place. However, even with large training sets, the regression model might not be able to represent the solutions of the OCP with exact accuracy. The average error of the regression model with respect to the NLP solution is fairly good, with $\varepsilon = 4 \cdot 10^{-3}$. Fortunately, for a large-scale power production estimate, the comparably very small mismatch might not be a problematic issue as model approximations and the wind data reanalysis already present a source of inaccuracy. Either way, the presented methods are helpful to obtain a detailed estimate of the available AWE system power production with low computational power.

The overall aim of this work is to propose a methodology that simplifies the use of the common used optimal control strategy for extensive power generation analyses. The use of a OCP for the power computation is indeed complex and computational tedious. There exist simpler and faster methods to compute the power production using real wind data, as presented in Refs. [6,7]. However, it has been shown that using OCP results into accurate power approximations [10,27]. In order to get an accurate estimation of the power generation for a performance evaluation of the technology, a detailed model is indispensable. By formulating an OCP to solve for the maximal power production of a point-mass model, fewer limitations are introduced, and it allows for the evaluation of more accurate vertical wind profiles instead of assuming a logarithmic wind shear. Additional information, such as the ideal operational height for specific regions can be analyzed with the results of the OCPs. Indeed, the accuracy gain might be decreased when applying a regression model to analyze larger areas. If the goal is to maximize the accuracy of the power production estimation, future work should compare the approach presented here to the approach presented in Ref. [7].

## 4. Conclusion

In order to estimate the large-scale potential of AWE systems, large-scale spatial and temporal wind data need to be evaluated without neglecting the transformation of available wind energy to electrical energy. In order to compute the maximum average power production of a drag-mode AWE system under a specific wind condition, the use of optimal control is a commonly used method. It provides an accurate approximation of the power production and the possibility of modeling the dynamics of the AWE system in a realistic way. However, a power computation in larger scale with this method becomes tedious as the corresponding parametric NLP is fairly complex to solve and needs appropriate initialization. This paper proposes a method to lower the computational cost in order to enable the use of optimal control for a large scale power investigation, where good approximations of the system dynamics are required. The method can process big wind data by efficiently computing a large family of NLPs using a homotopy-path-following method. The path-following method is most efficient if the studied parameters are sorted such that the distance of the sequential optimal solutions $S(\mathbf{p})$ and $S(\mathbf{p}_+)$ is minimal. In order to exploit the method in the context of primal-dual interior-point problems, the barrier parameter $\tau$ ought to be kept fixed in the solver.

This paper demonstrates the HPbd algorithm by solving a family of NLPs for a wind data set of $n_{ts} = 728$ wind profiles. Different values of the fixed barrier parameter $\tau$ are compared and a $\tau$ in the range of $10^{-4} - 10^{-3}$ is found to offer a high solution accuracy for a short computation time. The computation time with HPbd is tested against applying the pure initiating HPinit method. With a fixed barrier parameter $\tau = 10^{-4}$ and the ordering of the parameters by the wind speed at the average operational height of the drag-mode AWE system (300 m) a significant reduction in computation time (from 21.01 to 1.03 s per parameter) is observed.

For extremely large-scale temporal and spatial wind data, we propose a machine learning approach using a regression model. Different regression models are evaluated in order to find the model that performs best in mimicking the OCP and predicting the optimal power production based on the available wind data. With an $R^2$-value of 0.997, the random forest model seems to be most accurate. Testing the model on the same $n_{ts} = 728$ wind profiles used previously and comparing the predictions to the NLP solutions results in an average absolute error of $4 \cdot 10^{-3}$. The computation time for the whole data only takes $\sim 0.067s$, i.e., this regression method estimates power profiles four orders of magnitude faster than the HPbd method.

Finally, the proposed methodology is a suitable tool for studies on potential power production, which are set to scale up in both the volume of data and the number of geographic locations. To the best of the authors' knowledge, this is the first study to consider realistic wind shears other than the logarithmic profile assumption for power profile estimations using optimal control. This study is focused on drag-mode systems, but the method will be tested on other AWE systems in future work.

## CRediT authorship contribution statement

**E.C. Malz:** Conceptualization, Methodology, Software, Investigation, Writing - original draft. **V. Verendel:** Methodology, Software, Writing - review & editing. **S. Gros:** Conceptualization, Methodology, Supervision, Writing - review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

This work is supported by the project AWESCO (H2020-ITN-642682) funded by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 642682. The authors would also like to thank for the support from Chalmers Area of Advance Energy and ICT as well as Paulina Essunger for comments to the manuscript.

## References

[1] P. Bechtle, M. Schelbergen, R. Schmehl, U. Zillermann, S. Watson, Airborne wind energy resource analysis, Renew. Energy 141 (2019) 1103–1116 [Online]. Available: http://www.10.1016/j.renene.2019.03.118.

[2] C.L. Archer, L. Delle Monache, D. Rife, Airborne wind energy: optimal locations and variability, Renew. Energy 64 (2014) 180–186 [Online]. Available: http://www.10.1016/j.renene.2013.10.044.

[3] S. Costello, C. Costello, G. François, F. Bonvon, Analysis of the maximum efficiency of kite-power systems, J. Renew. Sustain. Energy 7 (5) (2015) [Online]. Available: http://www.10.1063/1.4931111.

[4] S. Kambouris, The Flying Electric Generator: Evaluating the Claims of a Largely Ignored Proposal for Generating Electricity from High-Altitude Winds, Msc Thesis, University of Melbourne, 2016 [Online]. Available: http://hdl.handle.net/11343/91085.

[5] J. Manewell, J. McGowan, A. Rogers, Wind Energy Explained: Theory, Design and Application, Wiley, 2009.

[6] J.P.R. Machado, Estimation of Energy Production in Aerial Systems of Wind Energy, Msc Thesis, University of Porto, 2019 [Online]. Available: https://www.researchgate.net/publication/335168596_Estimation_of_energy_production_in_aerial_systems_of_Wind_Energy.

[7] M. Ranneberg, D. Wölfle, A. Bormann, P. Rohde, F. Breipohl, I. Bastigkeit, Fast power curve and yield estimation of pumping airborne wind energy systems, in: R. Schmehl (Ed.), In *Airborne Wind Energy: Advances In Technology Development And Research*, Springer, 2018, pp. 623–641.

[8] M.L. Loyd, Crosswind kite power, J. Energy 4 (Jun. 1980) 106–111 [Online]. Available: http://www.10.2514/3.48021.

[9] G. Horn, S. Gros, M. Diehl, Numerical trajectory optimization for airborne wind energy systems described by high fidelity aircraft models, in: U. Ahrens, M. Diehl, R. Schmehl (Eds.), Airborne Wind Energy, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 205–218.

[10] G. Licitra, J. Koenemann, P. Williams, G. Ruiterkamp, R. Horn, M. Diehl, Viability assessment of a rigid wing airborne wind energy pumping system, in: Proceedings Of the 2017 21st International Conference On Process Control, PC 2017, No. Proceedings Of the 2017 21st International Conference On Process Control, PC 2017, Ampyx Power B.V, 2017, pp. 452–458 [Online]. Available: http://proxy.lib.chalmers.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-85027503592&site=eds-live&scope=site.

[11] G. Licitra, J. Koenemann, A. Bürger, P. Williams, R. Ruiterkamp, M. Diehl, Performance assessment of a rigid wing airborne wind energy pumping system, Energy 173 (2019) 569–585 [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360544219302592.

[12] J. Koenemann, P. Williams, S. Sieberling, M. Diehl, Modeling of an airborne wind energy system with a flexible tether model for the optimization of landing trajectories, IFAC-PapersOnLine 50 (1) (2017) 11 944–11 950, 20th IFAC World Congress. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896317315227.

[13] J. Stuyts, G. Horn, W. Vandermeulen, J. Driesen, M. Diehl, Effect of the electrical energy conversion on optimal cycles for pumping airborne wind energy, IEEE Transactions on Sustainable Energy 6 (1) (2015) 2–10.

[14] E. Malz, F. Hedenus, L. Göransson, V. Verendel, S. Gros, Drag-mode airborne wind energy vs. wind turbines: an analysis of power production, variability and geography, Energy 193 (2020) 116765 [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360544219324600.

[15] R. Leuthold, J. De Schutter, E.C. Malz, G. Licitra, S. Gros, M. Diehl, Operational regions of a multi-kite awe system, in: 2018 European Control Conference (ECC), 2018, pp. 52–57 [Online]. Available: http://www.10.23919/ECC.2018.8550199.

[16] L. Landberg, Meteorology for Wind Energy: an Introduction, first ed., Wiley, 2016.

[17] R.B. Stull, An Introduction to Boundary Layer Meteorology, Kluwer Academics Publishers, 1988.

[18] L. Göransson, J. Goop, M. Odenberger, F. Johnsson, Impact of thermal plant cycling on the cost-optimal composition of a regional electricity generation system, Appl. Energy 197 (2017) 230–240 [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261917304142.

[19] R. Gelaro, et al., The modern-era retrospective analysis for research and applications, version 2 (merra-2), J. Clim. 30 (14) (2017) 5419–5454.

[20] P. Deuflhard, Newton Methods for Nonlinear Problems, Springer, 2004.

[21] A. Zanelli, R. Quirynen, J. Jerez, M. Diehl, ", A Homotopy-Based Nonlinear Interior-point Method for NMPC, *IFAC PapersOnLine*, 2017 [Online]. Available: http://www.10.1016/j.ifacol.2017.08.2175.

[22] E.C. Malz, M. Zanon, S. Gros, A quantification of the performance loss of power averaging in airborne wind energy farms, in: 2018 European Control Conference (ECC), 2018, pp. 58–63 [Online]. Available: http://www.10.23919/ECC.2018.8550357.

[23] S. Gros, M. Zanon, M. Diehl, A relaxation strategy for the optimization of airborne wind energy systems, in: 2013 European Control Conference (ECC), 2013, pp. 1011–1016 [Online]. Available: http://www.10.23919/ECC.2013.6669670.

[24] M. Mehrjoo, M.J. Jozani, M. Pawlak, Wind turbine power curve modeling for reliable power prediction using monotonic regression, Renew. Energy 147 (2020) 214–222 [Online]. Available: http://www.10.1016/j.renene.2019.08.060.

[25] J. Heinermann, O. Kramer, Machine learning ensembles for wind power prediction, Renew. Energy 89 (2016) 671–679.

[26] M. Marčiukaitis, I. Žutautaitė, L. Martišauskas, B. Jokšas, G. Gecevičius, A. Sfetsos, Non-linear regression model for wind turbine power curve, Renew. Energy 113 (2017) 732–741 [Online]. Available: http://www.10.1016/j.renene.2017.06.039.

[27] E.C. Malz, J. Koenemann, S. Sieberling, S. Gros, A Reference Model for Airborne Wind Energy Systems for Optimization and Control, *Renewable Energy*, 2019.

[28] Global modeling and assimilation office gmao, *merra-2tavgl_2d_slv_nx:2d,1-hourly*,time-averaged,single-level,assimilation,single-level Diagnostics v5.12.4, Greenbelt, Md, usa, Goddard Earth Sciences Data and Information Services Center (Ges Disc), 2015 accessed: [30.08.3018].

[29] Global modeling and assimilation office gmao, *merra-2tavg3_3d_asm_nv:3d,3-Hourly*,time-Averaged,model-Level,assimilation,assimilated Meteorological Fields v5.12.4, Greenbelt, Md, usa, Goddard Earth Sciences Data and Information Services Center (Ges Disc), 2015 accessed: [30.08.3018]. [Online]. Available: https://disc.gsfc.nasa.gov/datasets/M2T3NVASM_V5.12.4/summary.

[30] L. Biegler, Nonlinear Programming, Society for Industrial and Applied Mathematics, 2010.

[31] J.A.E. Andersson, J. Gillis, G. Horn, J.B. Rawlings, M. Diehl, "CasADi – A Software Framework for Nonlinear Optimization and Optimal Control, " *Mathematical Programming Computation*, 2018 [Online]. Available: http://www.10.1007/s12532-018-0139-4.

[32] A. Wächter, L.T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Math. Program. 106 (1) (2006) 25–57 [Online]. Available: http://www.10.1007/s10107-004-0559-y.

[33] S. Gros, M. Zanon, Numerical periodic optimal control in the presence of invariants, IEEE Trans. Automat. Contr. (2017) 2818–2832 [Online]. Available: http://www.10.1109/TAC.2017.2772039.

[34] J. Nocedal, S. Wright, Numerical Optimization, Springer, 2006.

[35] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning with Applications in R, Springer, 2017.

[36] R. Tibshirani, Regression shrinkage and selection via the lasso, J. Roy. Stat. Soc. B 58 (1) (1996) 267–288 [Online]. Available: http://www.10.1111/j.2517-6161.1996.tb02080.x.

[37] A.E. Hoerl, R.W. Kennard, Ridge regression: applications to non-orthogonal problems, Technometrics 12 (1) (1970) 69–82 [Online]. Available: http://www.10.1080/00401706.1970.10488635.

[38] T. Hastie, R. Tibshirani, J.H. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2009.

[39] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32 [Online]. Available: http://www.10.1023/A:1010933404324.

[40] G.C. Cawley, N.L. Talbot, On over-fitting in model selection and subsequent selection bias in performance evaluation, J. Mach. Learn. Res. 11 (2010) 2079–2107.