Received: 18 April 2023 | Revised: 26 July 2023 | Accepted: 26 September 2023

DOI: 10.1002/we.2875

RESEARCH ARTICLE

WILEY

Synthetic turbulence modelling for offshore wind farm engineering models using coherence aggregation

Valentin Chabaud

SINTEF Energy Research, Trondheim, Norway

Correspondence
Valentin Chabaud, SINTEF Energy Research, NO-7465, Trondheim, Norway.
Email: valentin.chabaud@sintef.no

Funding information
European Commission, Grant/Award Number: 851207; Norges Forskningsråd, Grant/Award Numbers: 304229, 321954

Abstract

Turbulent wind fields are known to be a major driver for structural loads and power fluctuations on offshore wind turbines. At the single-turbine scale, there exist well-established design standards based on wind spectra and coherence functions calibrated from years of measurements, which are used to generate multiple 10-min wind field realisations known as *synthetic turbulence boxes*, themselves used as input to turbine-scale aero-hydro-servo elastic codes. These methods are however not directly applicable at farm scale. When analysing the dynamics of large offshore wind farms, measurements reveal the importance of large, low-frequency turbulent vortices for power fluctuations and hence for wind farm control and grid integration. Also, farm-scale wind fields are needed as input to farm-scale aero-servo-elastic codes for the modelling of wake dynamics, affecting structural loads. These new concerns motivate an upgrade in the original turbine-scale wind field representation: (1) spectral models need to be based on farm-scale measurements, (2) the frozen-turbulence assumption merging temporal and along-wind coherence must be lifted, (3) simplifications are needed to reduce the number of degrees of freedom as the domain becomes excessively large. This paper suggests models and algorithms for *aggregated* farm-wide corrrelated synthetic turbulence generation—lumping the wind field into space-averaged quantities—adapted to the aero-hydro-servo elastic modelling of large offshore wind farms. Starting from the work of Sørensen et al. in the early 2000s for grid integration purposes, methods for structural load modelling (through wake meandering and high-resolution wind field reconstruction) are introduced. Implementation and efficiency matters involving mathematical subtleties are then presented. Finally, numerical experiments are carried out to (1) verify the approach and implementation against a state-of-the-art point-based—as opposite to aggregated—synthetic turbulence generation code and (2) illustrate the benefit of turbulence aggregation for the modelling of large offshore wind farms.

KEYWORDS
aggregated spectral modelling, synthetic turbulence, wind farm simulations

1 | INTRODUCTION

As offshore wind energy technology becomes more and more mature, plant size has evolved from arrays of a handful of turbines to 50–100+ turbines for newly installed and planned farms. To further reduce costs and make the technology truly competitive, integrated system-level modelling is identified as a major challenge.[1] However, while engineering aero-hydro-servo elastic (AHSE) models have been successfully developed at the turbine level, the farm level presents difficulties that have not yet been satisfactorily addressed. One is linked to atmospheric modelling, which is another recognised major challenge.[1] While high-fidelity large eddy simulation (LES) solvers are available, their excessive computational cost does not qualify them as engineering models. The current engineering approach as suggested in the IEC standard is to split the modelling into ambient wind on the one side and disturbed wake flow on the other side, later summed using the superposition principle.[2,3]

Leading-edge engineering codes for structural modelling such as DTU's HAWC2 or DNV's Bladed focus on single turbines, precomputing both the ambient and wake flows prior to simulation. In practice, a so-called *turbulence box* is generated whose slices correspond to wind input for each simulation time step. While this sequential approach is efficient for structural load analysis, it cannot handle feedback effects from turbines to flow occurring at runtime—for example, through wind farm control. A notable recent advance has been achieved through the development of NREL's FAST.Farm, where the whole farm is simulated simultaneously with each turbine running its own AHSE model in parallel.[4,5] This gives the possibility to incorporate farm-wide flow dynamics and controls. However, while farm-wide wake flow dynamics are indeed captured, inherent limitations in ambient wind modelling inhibit the practical inclusion of farm-wide turbulence. Extending the traditional methods generating fine-meshed turbulent boxes is an option when considering an array of a handful of turbines[6] but becomes infeasible when looking at a utility scale farm: the computational cost increases cubically with the number of points, becoming excessively large as the box would need to (1) cover the entire farm area instead of one single rotor and (2) be augmented along an additional dimension as variations in time and in space in the along-wind direction can no longer be interchanged (i.e., the so-called Taylor's frozen-turbulence assumption is no longer valid for the large inter-turbine distances, as the turbulence at a downstream point is not simply linked to an upstream point by a time delay). The alternatives are (1) LES-based modelling of ambient flow and (2) independent generation of turbulent boxes for each turbine.[4] The first option, although cheaper than LES simulations with wind turbines present, is still too impractical to be qualified for mid-fidelity engineering modelling. The second option is inherently wrong for the purpose of this paper to model farm-wide turbulence.

Perhaps the most visible consequence of farm-wide turbulence is fluctuations in power production arising from large coherent turbulence patterns across several turbines, if not the entire farm. Towards more accurate models for grid integration purposes, this led Sørensen and Vigueras-Rodriguez's research group to develop simplified wind turbine and farm models using farm-level turbulence modelling.[7–10] As part of this development, farm-level turbulence was characterised, showing the significance of large, low-frequency turbulent vortices in existing offshore farms,[9,10] arising from mesoscale diurnal variations depending on atmospheric stability.[11] Further, noticing that small, high-frequency vortices—although important for loads—do not significantly affect rotor dynamics and hence power output (the large rotor inertia and swept area acting as low-pass filters), the idea of coherence aggregation was developed.[7,8] As only rotor-averaged quantities are of interest, the averaging operation is performed beforehand in the frequency domain instead of during simulation in the time domain. This bypasses the need of modelling turbulence on a fine grid in a first stage only to, in turn, average it out, thus reducing complexity tremendously. However, the simplified farm models for which coherence aggregation has been originally developed are not comprehensive AHSE models able to capture the dynamic response of the turbines, nor do they include disturbed (wake) flow.

This paper presents the theoretical background enabling the adaption of coherence aggregation for practical use in farm-level AHSE simulation software packages (among other applications). The novelty is twofold. First, the paper addresses specific challenges linked to a major difference with the work of Sørensen: aggregated turbulence needs not only to be modelled at each turbine but also downstream of turbines where it drives wake meandering (in practice, aggregated turbulence needs to be modelled on a regularly spaced grid discretising the entire farm area). Second and perhaps most importantly, a method for constrained turbulence without information loss is suggested, to (1) construct a wind field over the entire farm area knowing the wind field at the turbines and (2) reconstruct a full-field high-resolution wind field at the turbines given an aggregated realisation (i.e., reverse the aggregation process), thus providing a method for generating correlated high-resolution turbulent boxes across the farm.

The paper is organised as follows: The Veers method for frequency-domain synthetic turbulence generation for single turbines is first introduced in Section 2. It is then progressively extended to farm-wide applications in Section 3 through coherence aggregation, lifting the frozen-turbulence assumption, and full-field reconstruction. Specific challenges linked to the synthetic generation of aggregated quantities between closely spaced points are then addressed in Section 4. Section 5 presents a numerical verification against full-field realisations from a state-of-the-art code and illustrates the efficiency of the various concepts and algorithms suggested in the paper.

1.1 | A note on scales

The correlation of atmospheric turbulence decays exponentially with both distance and frequency (see Equation 2), meaning that low-frequency vortices are large and high-frequency vortices are small. From the turbine point of view, vortices larger than a couple of times the rotor diameter

correspond to a timescale of approximately 10 min. They may be seen as quasi-static variations, explaining why 10-min simulations are often used to characterise loads.[3,12] However, from the farm point of view, vortices that cover the entire area of a utility-scale farm (100 turbines) typically have periods of several hours. In between, vortices may be just large enough to influence clusters of turbines but not the entire farm. This results in power fluctuations and changes in the disturbed wake flow, hence driving the wind conditions at the minutes timescale. This separation of length- and timescales suggests a nested approach, where farm level feeds into turbine level. This paper focuses on the farm level but also offers the possibility to jointly model the two levels simultaneously through long (1 h) simulations capturing farm-wide effects while still featuring high resolution at the turbine scale.

2 | THE VEERS METHOD

Joint random generation of multiple correlated stationary processes from their spectral representation dates back to the work of Shinozuka[13] and has become a standard method used in a variety of applications where multiple stochastic inputs are correlated in space, such as seismic vibrations or wind.[14] Veers[15] adapted it to wind energy, initiating the current approach of precomputing turbulent boxes propagated through the rotor at simulation time under Taylor's frozen-turbulence assumption. The method relies on Cholesky factorisation and random phases to come up with a turbulent box realisation that satisfies spectral properties fitted from field measurements.

2.1 | Notations and coordinate system

The North-West-Upward (NWU) coordinate system with wind direction pointing northward (standard for AHSE codes) is used. Coordinates and axes are noted (x,y,z) and corresponding wind velocity components (u,v,w). The slash symbol / used in subscripts switches between components for a more compact notation, not to be confused with the divider operator always represented in fraction format.

2.2 | Spectral representation of turbulence

Turbulence is first characterised through its power spectral density (PSD), describing how the energy carried by wind velocity at a specific point in space is distributed as function of frequency. The second spectral property characterising turbulence is the real part of the coherence (denoted coherence in the rest of the paper), characterising by how much the wind velocities u_1 and u_2 at two specific points in space are correlated with each other:

$$\gamma_{u_1u_2} = \frac{|S_{u_2u_1}|}{\sqrt{S_{u_1u_1}S_{u_2u_2}}} = \gamma_{u_2u_1}\left(\boldsymbol{p}, f, \vec{r}_{12}\right) \tag{1}$$

with $S_{u_2u_1}$ the—in general—complex-valued cross spectral density (CSD), $S_{u_iu_i}$ the PSD at point i, $\boldsymbol{p}$ a set of parameters characterising the wind condition (mean wind speed, direction, atmospheric stability, turbulence intensity, etc.), f the frequency and $\vec{r}_{12}$ the position vector from point 1 to 2 (in the Veers method, these are located on the same rotor plane). This coherence of atmospheric turbulence mathematically defines the spatio-temporally correlated flow patterns known as turbulent *vortices*, while the power spectral density represents their strength.

Engineering AHSE models benefit from well-defined standards providing spectral models to model turbulence at the turbine/minutes scale. At the farm/hours scale, however, turbulence is highly site-dependent and meteorological studies using field measurements and high-fidelity simulations have not led yet to a consensus that may be used to define a standard engineering procedure. At offshore locations—and also on land[16]—the so-called spectral gap, showing no turbulence for periods 10 min to 4 h with least content at 1 h, has been widely questioned. Worth mentioning are the works of Larsén at al.,[17–19] Cheynet et al.,[11] and Vigueras-Rodriguez et al.,[9,10] who suggested spectral models at larger scales. It should be noted, however, that these models rely on the assumption of stationarity, which may be questionable and needs careful calibration and parametrisation with respect to atmospheric conditions.

Common to turbine and farm scales, a generic exponential model for the coherence function reads:

$$\gamma_{u_1u_2}\left(\boldsymbol{p}, f, \vec{r}_{12}\right) = exp\left(-\left\|\mathbf{a}\otimes\vec{r}_{12}\right\|\sqrt{\left(\frac{f}{u_\infty}\right)^2 + \left(\frac{\left\|\mathbf{b}\otimes\vec{r}_{12}\right\|}{\left\|\vec{r}_{12}\right\|}\right)^2}\right) \tag{2}$$

where u_∞ is the mean wind speed, $\mathbf{a}$ is a vector of decay factors for each direction, and $\mathbf{b}$ a vector of offsets forcing coherence to decrease with distance at zero frequency. A starting point may be the model of Vigueras-Rodriguez et al[10] for the farm-level coherence, namely,

$$\mathbf{a}_{\text{farm}} = [a_{\text{long}} \quad a_{\text{lat}} \quad N/A]^T \text{ with } a_{\text{lat}} = c_1 \frac{u_\infty}{\left\|\vec{r}_{12}\right\|} + c_2 \tag{3a}$$

$$\mathbf{b}_{\text{farm}} = \mathbf{0}_{3\times 1} \tag{3b}$$

where a_{long}, c_1, and c_2 are empirical coefficients and $\mathbf{0}$ is the zero matrix/vector. For physical consistency and numerical efficiency (see Section 4.3.3), the boundary between turbine and farm scales should be continuous and the coherence should decrease with distance and frequency. The following suggestion matches these requirements by introducing a transition distance d_{b_y} at the boundary while putting more focus on farm-level coherence:

$$\mathbf{a} = [a_{\text{long}} \quad a_{lat} \quad \mathbf{a}_{\text{turb}_z}]^T \text{ with } a_{\text{lat}} = \begin{cases} \mathbf{a}_{\text{turb}_y} & \left\|\vec{r}_{12}\right\| \le d_{b_y} \\ c_1 \dfrac{u_\infty}{\left\|\vec{r}_{12}\right\|} + c_2 & \left\|\vec{r}_{12}\right\| \ge d_{b_y} \end{cases}; \; d_{b_y} = \frac{c_1 u_\infty}{\mathbf{a}_{\text{turb}_y} - c_2} \tag{4a}$$

$$\mathbf{b} = \mathbf{0}_{3\times 1} \tag{4b}$$

which removes the offset represented by $\mathbf{b}$, arguing it has become obsolete as it used to model low-frequency effects at the turbine level, now overridden by the farm-level coherence model. $\mathbf{a}_{\text{turb}}(\mathbf{p}) = \left[N/A \quad \mathbf{a}_{\text{turb}_y}(\mathbf{p}) \quad \mathbf{a}_{\text{turb}_z}(\mathbf{p})\right]^T$ are the turbine-level decay factors, given by standards[3,12] or site-specific models.[11,20]

2.3 | Random Gaussian process generation

Extending from two to n points, the symmetric $n \times n$ coherence matrix $\boldsymbol{\Gamma}$ reads

$$\boldsymbol{\Gamma} = \begin{bmatrix} 1 & \dots & \gamma_{u_1 u_n} \\ \vdots & \ddots & \vdots \\ \gamma_{u_n u_1} & \dots & 1 \end{bmatrix} = \begin{bmatrix} 1 & \dots & \dfrac{S_{u_1 u_n}}{S_u} \\ \vdots & \ddots & \vdots \\ \dfrac{S_{u_n u_1}}{S_u} & \dots & 1 \end{bmatrix} \tag{5}$$

where it has been assumed that the process is ergodic, meaning the power spectral density S_u is the same for all points (this assumption is not necessary but will be used throughout the paper for the sake of simplicity). The cross spectral matrix $\boldsymbol{S}$ relates to the Fourier transform $\boldsymbol{U}$ of the wind velocity timeseries we want to generate through the outer product:

$$\mathbf{S}(\mathbf{p},\mathbf{f}) = \boldsymbol{\Gamma}(\mathbf{p},f) S_u(\boldsymbol{p},f) = \mathbf{U}\mathbf{U}^H \tag{6}$$

with ^{H} being the Hermitian (transpose and complex conjugate) operator. The goal is then to find a realisation of $\boldsymbol{U}(\boldsymbol{p},f)$ that satisfies (6), for each frequency for a given wind condition. The timeseries are then obtained from the inverse Fourier transform. Writing element j of $\mathbf{U}$ in complex modulus and argument form $\mathbf{U}_j = |\mathbf{U}_j| e^{i\angle \mathbf{U}_j}$, it appears that while $|\mathbf{U}_j| = \sqrt{S_u}$ is trivial, solving for the complex argument involves an overdetermined system of $\frac{n(n-1)}{2}$ equations with full rank for n unknowns. There is therefore no deterministic solution to (6), but it can be solved in the stochastic sense using the average $\left\langle \mathbf{U}\mathbf{U}^H \right\rangle$ over frequencies and realisations (i.e., simulations) instead of $\mathbf{U}\mathbf{U}^H$:

$$\left\langle \boldsymbol{U}\boldsymbol{U}^H \right\rangle = \boldsymbol{\Gamma} S_u \tag{7}$$

If $\boldsymbol{\Gamma}$ can be expressed as a product of matrix square roots $\boldsymbol{\Gamma} = \boldsymbol{A}\boldsymbol{A}^H$, a solution to (7) is

$$\mathbf{U} = \sqrt{S_u}\mathbf{A}\boldsymbol{\Phi} \tag{8}$$

where $\boldsymbol{\Phi} = \left[e^{i\phi_1} \ldots e^{i\phi_n}\right]^T$ is a phase angle vector, the phases ϕ_i being random and independent such that $\left\langle \boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{H}} \right\rangle = \mathbf{I}_n$, the identity matrix. A particular square root of $\boldsymbol{\Gamma}$ that is convenient to use is the lower-triangular Cholesky factor **L**, obtained by Cholesky factorisation. The complete set of solutions may be obtained by multiplication of **L** with the space of orthogonal matrices.

2.4 | Taylor's frozen-turbulence assumption

Conceptualising the turbulence input as a three-dimensional "box" is made possible thanks to the confusion between space and time. Time variations and along-wind spatial variations advected with the mean wind speed are assumed equivalent. This is known as Taylor's frozen-turbulence assumption (or hypothesis). For the Veers model, this assumption is not necessary *in theory*, but in practice not making it would require one 3D turbulence box per time step (or conceptually a 4D box) covering all along-wind locations where wind will or could be evaluated. For single turbines, modelling spatial variations in the wind field along the x axis is only needed to encompass the relatively small rotor or blade displacements, so the frozen-turbulence assumption is reasonable. When larger distances are of interest, the assumption becomes questionable. This has been recently addressed in the context of lidar-based control of wind turbines, where the wind field is measured upstream of wind turbines and significantly evolves before hitting the turbine. A 4D upgrade augmenting the original 3D box has been recently suggested by Chen et al,[21] making use of a two-step Cholesky decomposition to tackle the otherwise dramatic increase in computational cost arising from the curse of dimensionality.

For large farms however, neither the original 3D nor the 4D upgrade is appropriate. Computational resources would still be a limitation when using the 4D version, and the two-step Cholesky decomposition relies on the assumption of decoupled coherence in the longitudinal and lateral directions, which conflicts with the farm-wide model of Vigueras-Rodriguez et al.[10] Keeping the frozen-turbulence assumption would be equivalent to assuming that downstream turbines experience the same ambient wind fluctuations as upstream turbines, simply delayed by the advection time. This would obviously lead to incorrect power fluctuations as vortices do undergo changes along the way.

Mathematically, this delay is not a time but a phase delay described by the imaginary part of the coherence between two points as a function of along-wind distance. The coherence becomes complex-valued, and its argument reads

$$\angle \boldsymbol{\Gamma}_{\mathrm{ij}} = \boldsymbol{\Theta}_{\mathrm{ij}} = \exp\left(-2\pi \mathrm{f} \frac{\vec{e}_{\mathrm{x}}^{\mathrm{T}} \vec{r}_{\mathrm{ij}}}{\overline{\overline{u}}_{\mathrm{ij}}}\right) \tag{9}$$

with $\vec{e}_x$ being the unitary vector in along-wind direction and $\overline{\overline{u}}_{ij}$ the equivalent advection wind speed between points *i* and *j*. This makes the random process generation method described in Section 2.2 more complicated. While Sørensen et al[8] suggested an ad-hoc method, further simplification is possible by realising that if $\overline{\overline{u}}_{ij}$ is constant, the phase angle matrix $\boldsymbol{\Theta}$ might be directly transferred to the coherence matrix and its square root,[14] namely,

$$\mathbf{L} = |\mathbf{L}| \odot \boldsymbol{\Theta} \tag{10a}$$

$$|\boldsymbol{\Gamma}| = |\mathbf{L}||\mathbf{L}|^T \tag{10b}$$

$$\boldsymbol{\Gamma} = |\boldsymbol{\Gamma}| \odot \boldsymbol{\Theta} \tag{10c}$$

with $|\boldsymbol{\Gamma}|$ being the modulus of the now complex $\boldsymbol{\Gamma}$ and $\odot$ the Hadamard (elementwise) product. The solution (8) may then simply be updated to

$$\boldsymbol{U} = \sqrt{S_u}\left(\mathbf{L} \odot \boldsymbol{\Theta}\right)\boldsymbol{\Phi} \tag{11}$$

A simplistic constant model for $\overline{\overline{u}}_{ij}$ is given by Vigueras-Rodriguez et al[10] based on data from large Danish offshore wind farms as

$$\overline{\overline{u}}_{ij} = \kappa u_\infty \tag{12}$$

with u_∞ being the undisturbed mean wind speed and κ a constant fitted to $\kappa \approx 0.85$, which may be interpreted as an approximate inclusion of wake velocity deficit and atmospheric boundary layer or blockage effects. Ideally, $\overline{\overline{u}}_{ij}$ should encompass the effect of time- and space-varying wake effects on the advection speed. However, the importance of this effect in a mid-fidelity setting is an active topic of discussion (FAST.Farm includes it, while many other farm simulation tools implementing the Dynamic Wake Meandering method do not), and a constant needs to be assumed here in order to enable the above-presented decoupling between modulus and phase.

3 | COHERENCE AGGREGATION

As a result of Section 2.3, the modulus and argument of the complex coherence matrix may be dissociated. In the following, coherence refers only to the modulus part, the argument being characterised by (9) and (12).

3.1 | Spatial averaging

As mentioned in Section 2.3, a 4D turbulence box solving (11) for the entire farm at each time step would be needed. Since the Cholesky factorisation has a computational cost of $\mathcal{O}(n^3)$, this approach would quickly prove infeasible if point-based velocities were to be calculated with the same resolution as for single turbines. However, space-averaged values are sufficient for the purpose of farm-level flow modelling. Therefore, the averaging operation may be performed in the frequency domain beforehand instead of through costly Gaussian process generation and subsequent time-domain averaging in farm simulations.

Spatial averaging lumps turbulence at the centre of elementary averaging domains discretising the farm domain with a much coarser resolution than if point-based values were used, hence greatly reducing the number of degrees of freedom. Note that using a coarse resolution with a point-based representation—although widely used in current practice for convenience—leads to undersampling errors.

Naming Ω_i the averaging domain corresponding to element i, $\bar{u}_\Omega$ the time-domain average of u over the element, and $\mathcal{F}\{\}$ the Fourier transform operator, the weighted aggregated coherence $\overline{\gamma}_{\Omega_{u_1u_2}}$ between two elements is derived by averaging the point-based coherence between all pairs of points, adapting the work of Sørensen[7]:

$$\overline{\gamma}_{\Omega_{u_1u_2}}\left(\boldsymbol{p},f,\vec{r}_{0_1 0_2}\right)=\frac{\int_{\Omega_1}\int_{\Omega_2}\gamma_{u_1u_2}\left(\boldsymbol{p},f,\vec{r}_{12}\right)\psi_{\Omega_1}\left(\vec{r}_{0_1 1}\right)\psi_{\Omega_2}\left(\vec{r}_{0_2 2}\right)d\Omega_1 d\Omega_2}{\int_{\Omega_1}\int_{\Omega_2}\psi_{\Omega_1}\left(\vec{r}_{0_1 1}\right)\psi_{\Omega_2}\left(\vec{r}_{0_2 2}\right)d\Omega_1 d\Omega_2} \tag{13}$$

with $\vec{r}_{ij}$ being a position vector from i to j, 0_i being the centre of element i, and $\vec{r}_{12}$ reading $\vec{r}_{12}=\vec{r}_{0_1 0_2}+\vec{r}_{0_2 2}-\vec{r}_{0_1 1}$. See Figure 1 for an illustration. ψ_{Ω_i} is a weighting function associated with element i, which can represent the following:

- deterministic spatial variations of the wind field affecting the amplitude of the total wind speed—including turbulent fluctuations— such as vertical wind profile from ground friction, tower shadow, or wake-induced velocity deficit profile(s) from upstream turbine(s);
- mapping functions from axial wind speed to other stochastic quantities, such as linear wind shears or aerodynamic loads.

The choice of weighting function depends on what process is not to be modelled later in the time domain in turbine or farm simulations and should be accounted for when generating the equivalent wind field in the frequency domain. In practice, $\psi_{\Omega_i}=1$ is used to simplify the aggregation process and leave as much as possible to the time domain.*

The PSD of the aggregated wind speed is obtained by averaging on the same element (i.e., $\Omega_1=\Omega_2$, $0_1=0_2$):

$$S_{\bar{u}_\Omega}(\boldsymbol{p},f)=\mathcal{F}\left\{\frac{\int_{\Omega_1}\psi_\Omega\left(\vec{r}_{0_1 1}\right)u\left(\boldsymbol{p},t,\vec{r}_{0_1 1}\right)d\Omega_1}{\int_{\Omega_1}\psi_\Omega\left(\vec{r}_{0_1 1}\right)d\Omega_1}\right\}^2=\overline{\gamma}_{\Omega_{u_1u_2}}(\boldsymbol{p},f,0)S_u(\boldsymbol{p},f)=|H_{\Omega_u}(\boldsymbol{p},f)|^2S_u(\boldsymbol{p},f) \tag{14}$$

where H_{Ω_u} is the resulting transfer function, called the *admittance* function. The coherence between aggregated quantities $\gamma_{\bar{u}_{\Omega_1}\bar{u}_{\Omega_2}}$ may then be derived from the aggregated coherence between point-based quantities $\overline{\gamma}_{\Omega_{u_1u_2}}$ (through the definition of coherence from PSD and cross-spectrum as in (1)), yielding

$$\gamma_{\bar{u}_{\Omega_1}\bar{u}_{\Omega_2}}\left(\boldsymbol{p},f,\vec{r}_{0_1 0_2}\right)=\frac{\overline{\gamma}_{\Omega_{u_1u_2}}\left(\boldsymbol{p},f,\vec{r}_{0_1 0_2}\right)}{|H_{\Omega_{u1}}(\boldsymbol{p},f)||H_{\Omega_{u2}}(\boldsymbol{p},f)|} \tag{15}$$

*The weighting function for rotor averaging may incorporate the radial distribution of thrust or power coefficients along the blade, giving thrust and power-equivalent wind speeds. In practice, however, it appears that unitary weighting is sufficient to map wind speed to aerodynamic loads, hence no distinction between thrust and power is necessary, nor between cell- and rotor-averaging, as far as the weighting function is concerned.

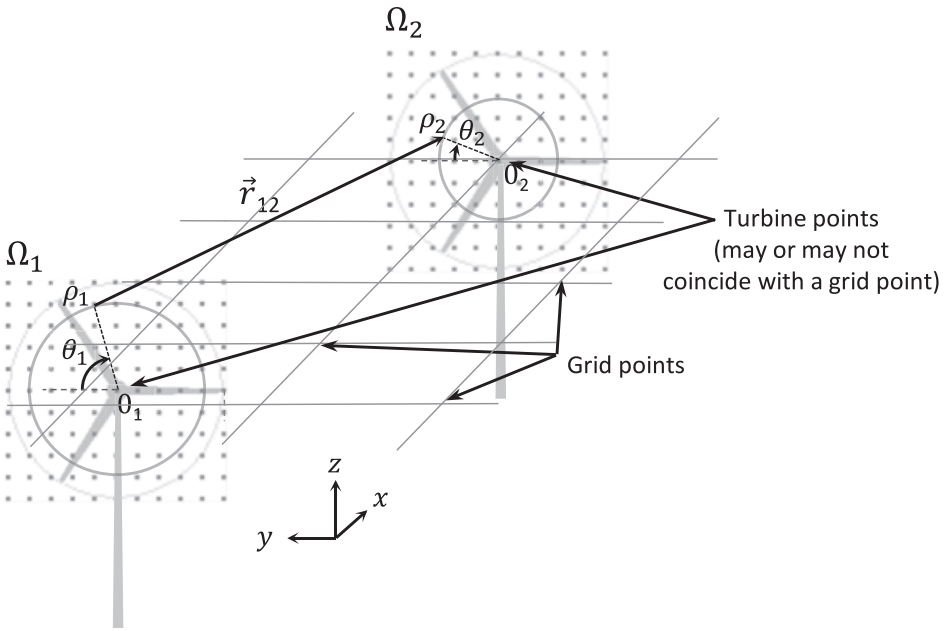

FIGURE 1 Schematic of the spatial averaging concept, including coordinate system and definition of turbine and grid points (see Section 3.2.1). The figure illustrates rotor-based averaging between turbine points, grid points, and cell-based averaging following a similar concept.

where $H_{\Omega_{u_1}}$ and $H_{\Omega_{u_2}}$ are kept distinct on purpose as they may use distinct weighting functions.

The choice of the integral domain Ω depends on whether one works with *cell-averaged* or *rotor-averaged* quantities. Rotor averaging is appropriate to model load-equivalent wind speeds (see footnote* for choice of weighting function) and is performed over a disk of the size of the rotor. Wake dynamics as modelled in the Dynamic Wake Meandering model are also driven by disc-averaged fluctuations, but an additional filter is used to remove the effect of small vortices.[2] Improvements of the model in FAST.Farm also make use of averaged fluctuations to update the eddy viscosity responsible for wake recovery and are crucial in velocity deficit modelling.[4] As this deals with volumetric quantities, 3D averaging using volume (triple) integrals over cuboid cells may be conceptually more appropriate in this case although this is open to discussion.

Note that the solution given by (8) depends on the choice of integral domain. Simulating both cell-averaged and rotor-averaged quantities would require solving (8) twice while ensuring consistency between the two solutions. This is addressed by using cascaded simulations, presented in Section 3.2.1.

In summary, the space-averaged turbulence may be characterised by aggregated spectral quantities given by (14) and (15), involving a sextuple or quadruple integral (each cell integral being triple or double depending on whether we look at cell-averaged or rotor-averaged quantities) of the point-based coherence function. Realisations may then be obtained by the Veers method described in Section 2.

3.1.1 | Extension to other components

The term *quantities* has been used on purpose, as not only the axial wind speed u can be modelled using Veers method and coherence aggregation. Tangential velocities v and w may also be obtained in a similar way.† Harmonics corresponding to rotational sampling of the wind field by one single or all three blades (the so-called 1p and 3p frequencies) may also be represented by azimuthal expansion.[8] 1p variations on the blade have a physical interpretation, as they correspond to linear wind shears (horizontal and vertical).

If the coherence between components appears to be significant and of interest (it is for instance known that u and w are correlated at the turbine level[11]; at the farm level, correlation between u, v and w will have consequences on wake meandering), the coherence matrix $\boldsymbol{\Gamma}$ may be augmented to simultaneously solve for all correlated components.

†Note that IEC standards specify zero coherence in tangential directions, resulting in zero aggregated PSD. This is nonphysical, and updated models found in the literature should be used.

3.1.2 | A note on the Mann model

Like the Veers method, the Mann model[22] is a standard method used to model turbulence for medium-fidelity wind turbine simulations. It uses a space-based rather than time-based spectral representation, using wave numbers instead of frequencies. The concept of coherence is replaced by spectral tensors to describe the correlation between vortices, which is arguably closer to physics and less reliant on calibration from wind measurements. It also has the advantage of not relying on the stationarity assumption, enabling a consistent inclusion of transients like gusts (although the Veers method may also be adapted for this purpose[23]). However, the Mann model is not appropriate for our purpose because (1) turbulence aggregation is ill-defined as spatial averaging conflicts with the space-based spectral representation, (2) the domain size would have to be excessively large in order to deal with the low wave numbers (long wavelengths) corresponding to the low-frequency fluctuations we are interested in, and (3) lifting the frozen-turbulence assumption is challenging as it is intrinsic to the model. Another final argument of practical nature is that, unlike the Veers method, the Mann model does not benefit from the same flexible open-source implementation.

3.2 | Constrained turbulence and cascaded construction around turbine points

This section presents a novel solution for constrained turbulence (conditional random process) generation using the Moore–Penrose matrix pseudo-inverse, enabling the cascaded, correlated generation of wind speeds at the farm and turbine levels without loss of information.

3.2.1 | Farm domain

As presented in Section (1), farm-wide correlation in turbulent wind fields is a driver for power fluctuations, and also affects wake dynamics. Power fluctuations are mostly driven by ambient wind fluctuations at the turbines, with wakes having a secondary effect. A first objective is therefore to get rotor-averaged turbulent wind fields *at the turbines* (around turbine hubs), with the right correlation between them. The corresponding entries in $\mathbf{U},\mathbf{S},\boldsymbol{\Gamma},\mathbf{L},\boldsymbol{\Theta},\boldsymbol{\Phi}$ are thereafter called *turbine points*. A second objective is to model aggregated turbulence in the interturbine domain, on a regularly spaced grid (see Section 4.1) that will be interpolated in the calculation of wake dynamics. Corresponding points are referred to as *grid points*. See Figure 1 for an illustration. It is desirable to treat turbine and grid points separately, first because we may only be interested in turbine points to get power fluctuations, and second because measuring wind fluctuations at the turbines is a lot easier than between them, leading to different validation / calibration approaches.

As mentioned earlier, the solution given by (11) is only valid in a stochastic sense and the order in which the different points come have an influence on the realisation, as a property of Cholesky factorisation. This leads to bad reproducibility if the turbine layout or the grid size changes. Turbine points may be put first in the Cholesky factorisation—this is the method used in the tool PyConTurb to include user-defined points at the turbine level[23,24]—to ensure independence from grid points (turbine points are realised first, and grid points adapted to yield the right correlation). This is however inconvenient: The short distance between turbine points and their respective nearest neighbour grid points will lead to challenges in the calculation of aggregated coherence and decomposition of the aggregated coherence matrix (i.e., positive-definiteness issues, see Section 4.3). Coping with this issue would involve: (1) the use of a nonregularly spaced grid, inducing computational issues as the number of points increases, and (2) mixing grid and turbine points, thus forcing the use of a single aggregation method (rotor- or cell-based, weighting, etc.) for physically different quantities (as discussed in Section 3.1). This paper suggests a solution that circumvents these issues by first simulating turbine points and then constructing regular grid points around them in a consistent (read here *coherent*) manner in a second stage. A prerequisite is that grid resolution is fine enough for the realisations at turbine and grid points to be linked through linear interpolation (discussed in Section 4.1).

In the following, only u will be considered for simplicity, as the extension to other components is straightforward. Let n_T be the number of turbine points, small in comparison to the number of grid points n_G. In a first stage, the Fourier transform $\mathbf{U}_T$ of u at turbine points is obtained using the method described in the previous sections. Here we are interested in load-equivalent quantities so rotor-based aggregation is used. In a second stage, we are looking for a realisation of the Fourier transform $\mathbf{U}_G$ of u at grid points (this time using cell-based aggregation), that is consistent with the realisation at turbine points, that is, satisfying

$$\mathbf{U}_T = \mathbf{P}_T\mathbf{U}_G = \frac{|H_T|}{|H_G|}\sqrt{S_u}\mathbf{P}_T(\mathbf{L}_G \odot \boldsymbol{\Theta}_G)\hat{\boldsymbol{\Phi}}_G = \mathbf{G}_G\hat{\boldsymbol{\Phi}}_G \tag{16}$$

where $\mathbf{G}_G = \frac{|H_T|}{|H_G|}\sqrt{S_u}\mathbf{P}_T(\mathbf{L}_G \odot \boldsymbol{\Theta}_G)$ is the solution (11) at grid points corrected by the ratio of the transfer functions H specific to the aggregation methods used (as given by (14)) and multiplied by $\mathbf{P}_T$, an $n_T \times n_G$ interpolation matrix linearly mapping grid points to turbine points, reading

$$\mathbf{P}_T = \mathbf{P}_{T_x} \otimes \mathbf{P}_{T_y} \tag{17a}$$

$$\mathbf{P}_{T_{\xi_{ij}}} = \begin{cases} 1-\alpha_i & \text{for} j = \left\lfloor \frac{\xi_i}{\Delta_\xi} \right\rfloor \\ \alpha_i & \text{for} j = \left\lfloor \frac{\xi_i}{\Delta_\xi} \right\rfloor + 1 \\ 0 & \text{else} \end{cases} \text{ with } \alpha_i = \frac{\xi_i}{\Delta_\xi} - \left\lfloor \frac{\xi_i}{\Delta_\xi} \right\rfloor, \text{ for} \xi \in (x,y) \tag{17b}$$

with $\otimes$ being the Kronecker product, (x_i, y_i) the coordinates of turbine i and $\lfloor\rfloor$ the rounding to nearest smaller integer operator. We are then looking for a solution to $\mathbf{U}_G$ that would satisfy both $\left\langle \mathbf{U}_G \mathbf{U}_G^H \right\rangle = \boldsymbol{\Gamma}_G S_u$ and (16), through a careful choice of the phase angle vector $\hat{\boldsymbol{\Phi}}_G$.

Satisfying (16) amounts to solving an underdetermined system of equations whose set of solutions yields

$$\hat{\boldsymbol{\Phi}}_G = \mathbf{G}_G^+ \mathbf{U}_T + (\mathbf{I}_{n_G} - \mathbf{G}_G^+ \mathbf{G}_G)\boldsymbol{\Phi}_G = \mathbf{G}_G^+ \mathbf{U}_T + \mathbf{G}_G^- \boldsymbol{\Phi}_G \tag{18}$$

with $\mathbf{G}_G^+$ being the $n_G \times n_T$ Moore–Penrose pseudo-inverse[25] of $\mathbf{G}_G$, reading

$$\mathbf{G}_G^+ = \mathbf{G}_G^H \left(\mathbf{G}_G \mathbf{G}_G^H \right)^{-1} \tag{19}$$

and $\mathbf{G}_G^- = \mathbf{I}_{n_G} - \mathbf{G}_G^+ \mathbf{G}_G$ the associated $n_G \times n_G$ nullspace matrix. $\boldsymbol{\Phi}_G \in \mathbb{R}^{n_G}$ is an arbitrary vector.

Satisfying $\left\langle \mathbf{U}_G \mathbf{U}_G^H \right\rangle = \boldsymbol{\Gamma}_G S_u$ comes down to satisfying

$$\left\langle \hat{\boldsymbol{\Phi}}_G \hat{\boldsymbol{\Phi}}_G^H \right\rangle = \mathbf{I}_{n_G} \tag{20}$$

which is achieved by choosing $\boldsymbol{\Phi}_G$ as a random phase vector as in (11), that is, reading

$$\boldsymbol{\Phi}_G = \left[e^{i\phi_1} \ldots e^{i\phi_n} \right]^T \tag{21}$$

with ϕ_i being random and independent. This may be proved using $\left\langle \boldsymbol{\Phi}_G \mathbf{U}_T^H \right\rangle = \left\langle \mathbf{U}_T \boldsymbol{\Phi}_G^H \right\rangle^H = \mathbf{0}_{n_G \times n_T}$, $\left\langle \boldsymbol{\Phi}_G \boldsymbol{\Phi}_G^H \right\rangle = \mathbf{I}_{n_G}$ and $\left\langle \mathbf{U}_T \mathbf{U}_T^H \right\rangle = \mathbf{G}_G \mathbf{G}_G^H$ yielding

$$\left\langle \hat{\boldsymbol{\Phi}}_G \hat{\boldsymbol{\Phi}}_G^H \right\rangle = \mathbf{G}_G^+ \left\langle \mathbf{U}_T \mathbf{U}_T^H \right\rangle \mathbf{G}_G^{+H} + \mathbf{G}_G^+ \left\langle \mathbf{U}_T \boldsymbol{\Phi}_G^H \right\rangle \mathbf{G}_G^{-H} + \mathbf{G}_G^- \left\langle \boldsymbol{\Phi}_G \mathbf{U}_T^H \right\rangle \mathbf{G}_G^{+H} + \mathbf{G}_G^- \left\langle \boldsymbol{\Phi}_G \boldsymbol{\Phi}_G^H \right\rangle \mathbf{G}_G^{-H} = \mathbf{G}_G^+ \mathbf{G}_G \mathbf{G}_G^H \mathbf{G}_G^{+H} + \mathbf{G}_G^- \mathbf{G}_G^{-H} \tag{22}$$

then using succeedingly the properties of the pseudo-inverse and nullpsace $\left(\mathbf{G}_G^+ \mathbf{G}_G \right)^H = \mathbf{G}_G^+ \mathbf{G}_G$, $\mathbf{G}_G \mathbf{G}_G^+ = \mathbf{I}_{n_T}$ and $\mathbf{G}_G^- \mathbf{G}_G^{-^H} = \mathbf{G}_G^-$:

$$\left\langle \hat{\boldsymbol{\Phi}}_G \hat{\boldsymbol{\Phi}}_G^H \right\rangle = \mathbf{G}_G^+ \mathbf{G}_G \left(\mathbf{G}_G^+ \mathbf{G}_G \right)^H + \mathbf{G}_G^- = \mathbf{G}_G^+ \mathbf{G}_G + \mathbf{I}_{n_G} - \mathbf{G}_G^+ \mathbf{G}_G = \mathbf{I}_{n_G} \tag{23}$$

which proves the consistency of the method for providing a *constrained yet random* realisation for $\mathbf{U}_G$ based on $\mathbf{U}_T$.

Note 1: Decoupling moduli and arguments not being possible this time, $\mathbf{G}_G, \mathbf{G}_G^+$ and $\mathbf{G}_G^-$ are complex-valued.

Note 2: Unlike $\boldsymbol{\Phi}_G$, $\hat{\boldsymbol{\Phi}}_G$ is not necessarily unitary (its components typically have nonunity modulus).

Note 3: $\mathbf{U}_T$ (or parts of) may also be imported from external sources (observed from SCADA data or aggregated from turbine-level turbulence boxes) with appropriate signal processing.

3.2.2 | Turbine domain

Using a similar approach, it is possible to reconstruct a high-resolution turbulence box consistent with the aggregated value $\mathbf{U}_T$ around each turbine. This closes the loop, enabling the possibility to obtain the turbine/minutes and farm/hours scales in an integrated, correlated manner. Let n_H be the number of points in the turbulence box. We are looking for a high-resolution wind field $\mathbf{U}_H$ that relates to $\mathbf{U}_{T_i}$ at turbine point i through weighted integrals (i.e., sums), namely,

$$\boldsymbol{\Psi}\mathbf{U}_H = \mathbf{U}_{T_i} = \sqrt{S_u}\boldsymbol{\Psi}\mathbf{L}_H\hat{\boldsymbol{\Phi}}_H = \mathbf{G}_H\hat{\boldsymbol{\Phi}}_H \tag{24}$$

with $\boldsymbol{\Psi}$ being the weighted averaging matrix which, when only the u component is simulated at the farm level, may be a $1 \times n_H$ vector reading:

$$\begin{cases} \boldsymbol{\Psi}_{1i} = \dfrac{1}{\sum \boldsymbol{\Psi}_{1i}} & r_0 \le r \le R \\ \boldsymbol{\Psi}_{1i} = 0 & r < r_0, r > R \end{cases} ; \; i = 1 \text{to} n_H \tag{25}$$

with r being the radial position of the point and r_0,R the blade root and tip radii respectively. This may be augmented in a straightforward manner with other components and corresponding weights. The solution is then similar to that of the previous section, namely,

$$\hat{\boldsymbol{\Phi}}_H = \mathbf{G}_H^{+}\mathbf{U}_T + \left(\mathbf{I}_{n_H} - \mathbf{G}_H^{+}\mathbf{G}_H\right)\boldsymbol{\Phi}_H \tag{26a}$$

$$\boldsymbol{\Phi}_H = \left[e^{i\phi_1} \ldots e^{i\phi_{n_H}}\right]^T \text{ with } \begin{cases} \langle \phi_i, \phi_j \rangle = 1 & i = j \\ \langle \phi_i, \phi_j \rangle = 0 & i \neq j \end{cases} \tag{26b}$$

Note that accordingly to the Veers method with frozen-turbulence assumption, $\mathbf{G}_\mathrm{H}$ and $\mathbf{G}_\mathrm{H}^{+}$ are real-valued.

4 | IMPLEMENTATION

The method described in the previous sections readily provides as such a mid-fidelity solution to model farm-wide correlated synthetic turbulence. However, for practical engineering use, a decent computational time is required. The method suffers from the so-called curse of dimensionality, with computational cost increasing nonlinearly with complexity. Without model order reduction measures, the number of grid points n_G increases proportionally with the farm area. The number of coherence aggregation operations, however, increases with $\mathcal{O}(\mathrm{n}_\mathrm{G}^2)$ as does the size of the coherence matrix $\boldsymbol{\Gamma}$, and the cost of the random generation process increases with $\mathcal{O}(\mathrm{n}_\mathrm{G}^3)$. Considering a utility-scale farm of 100 turbines with standard spacing and grid resolution (see Section 4.1), about 5000 points are needed. The computational time of a 1-h simulation (to capture farm-scale effects) with standard step size (see Section 4.1) on a standard personal computer would be on the order of days. This being deemed incompatible with an engineering solution, efforts have been made to drastically reduce the computational time.

Section 4.1 presents how to limit the number of grid points, while Section 4.2 aims to decrease the cost of the aggregation operation itself. Finally, Section 4.3 presents challenges and solutions related to Cholesky factorisation greatly affecting the computational cost.

4.1 | Farm domain discretisation

The number of points $n_G = \mathrm{n}_{G_x}\mathrm{n}_{G_y}\mathrm{n}_{G_z}$ derives directly from the domain size and grid resolution $\boldsymbol{\Delta}$. The latter should therefore be chosen as large as possible, still capturing the relevant physics accurately. The dynamic wake meandering model averages the velocity over the wake diameter in the cross-wind plane,[4] so the y and z resolutions should be of comparable size. The definition of wake diameter being unclear in literature,[4] using the rotor diameter D seems a reasonable assumption for $\boldsymbol{\Delta}_z$. As a result, there may only be need for one single aggregated point in the vertical dimension (i.e., $\mathrm{n}_{G_z} = 1$) to characterise turbulence (note that this is independent to the modelling of the wind profile which may still use high resolution), and this is assumed in the rest of the paper. This simplifies implementation as the turbulence box is again reduced from 4D to 3D (like the turbine level, this time not by using the frozen-turbulence assumption to remove x but by using aggregation to remove z). The choice of $\boldsymbol{\Delta}_x$ should be consistent with the distance between two wake planes, that is, the advection during one time step. This implies choosing a step size Δ_t, which could be based on the largest frequency of significance in aggregated wind speed quantities. However, arguing that aggregation filters out vortices smaller than one rotor diameter suggests the simpler yet pertinent choice of $\boldsymbol{\Delta}_x = D$ and $\Delta_\mathrm{t} = \frac{\boldsymbol{\Delta}_x}{u_\infty}$. Finally, $\boldsymbol{\Delta}_y = D$ could also be used, resulting in cubic cells. However, better consistency with the anisotropic nature of coherence (as $\mathbf{a}_x$ is significantly lower than $\mathbf{a}_y$ in (2)[10]) resulting in better numerical consistency and efficiency (see Section 4.3.1) suggests using $\boldsymbol{\Delta}_y \approx \frac{\mathbf{a}_x}{\mathbf{a}_y}\boldsymbol{\Delta}_x$.

Assuming ergodicity, the coherence between two points depends only on their *relative* positions. To avoid redundant operations, the aggregated coherence may then be precomputed at regularly-spaced locations covering the entire farm area with a specified grid resolution $\boldsymbol{\Delta}$, resulting in a $\mathrm{n}_{G_x} \times \mathrm{n}_{G_y}$ look-up table matrix $\boldsymbol{\Sigma}$, thereafter referred to as the kernel matrix (see Section 4.3.3). The much larger $n_G \times n_G$ coherence matrix $\boldsymbol{\Gamma}$ is then built using the nearest neighbour points in $\boldsymbol{\Sigma}$, namely,

$$\boldsymbol{\Gamma}_{ij} = \boldsymbol{\Sigma}_{\left\lfloor \frac{j-1}{n_{G_x}} \right\rfloor - \left\lfloor \frac{i-1}{n_{G_x}} \right\rfloor + 1, \left(j - n_{G_x}\left\lfloor \frac{j-1}{n_{G_x}} \right\rfloor\right) - \left(i - n_{G_x}\left\lfloor \frac{i-1}{n_{G_x}} \right\rfloor\right) + 1} \tag{27}$$

All frequency components $\mathbf{U}_G$ should be expressed at the same locations (i.e., grid points) prior to inverse Fourier transform. However, if accurate enough, coarser grid resolutions may be used and mapped back to the common specified grid by interpolation. The effective resolution $\tilde{\boldsymbol{\Delta}}(f)$ and number of points $\tilde{n}_G(f) = \tilde{n}_{G_x}(f)\tilde{n}_{G_y}(f)$ may then be adapted to accuracy requirements for each frequency component. An adaption strategy based on the rate of change of the kernel matrix is suggested as

$$\tilde{n}_{G_\xi} = \left\lceil \frac{\left\| \delta \boldsymbol{\Sigma}_\xi \right\|}{\epsilon_\gamma} \right\rceil, \ \xi \in (x, y) \tag{28}$$

with ϵ_γ being a specified tolerance on coherence errors, $\lceil\,\rceil$ the rounding to nearest greater integer operator, and $\delta\boldsymbol{\Sigma}_\xi$ the $(n_{G_x} - 1) \times n_{G_y}$ (for $\xi = x$) or $n_{G_x} \times (n_{G_y} - 1)$ (for $\xi = x$) difference matrices defined as $\delta\boldsymbol{\Sigma}_{x_{ij}} = \boldsymbol{\Sigma}_{(i+1)j} - \boldsymbol{\Sigma}_{ij}$, and $\delta\boldsymbol{\Sigma}_{y_{ij}} = \boldsymbol{\Sigma}_{i(j+1)} - \boldsymbol{\Sigma}_{ij}$. Any matrix norm such as infinite (maximum) or Frobenius (root mean square) may be used. The reduced-order and original coherence matrices $\tilde{\boldsymbol{\Gamma}}$ and $\boldsymbol{\Gamma}$ are then linked through interpolation matrices from full to reduced grid $\mathbf{P}_x(f)$ and $\mathbf{P}_y(f)$ and their reduced-to-full-grid counterparts $\mathbf{P}_x^+(f)$ and $\mathbf{P}_y^+(f)$, both constructed analytically based on weighting of neighbouring points (similar to (17b), details are not shown for the sake of conciseness):

$$\begin{aligned} \tilde{\boldsymbol{\Gamma}} &= (\mathbf{P}_x \otimes \mathbf{P}_y)\boldsymbol{\Gamma}(\mathbf{P}_x \otimes \mathbf{P}_y)^T \\ \boldsymbol{\Gamma} &\approx \left(\mathbf{P}_x^+ \otimes \mathbf{P}_y^+\right)\tilde{\boldsymbol{\Gamma}}\left(\mathbf{P}_x^+ \otimes \mathbf{P}_y^+\right)^T \end{aligned} \tag{29}$$

Wind velocities are then found using (11) for the reduced grid and then interpolated back to the original grid, namely,

$$\boldsymbol{U} \approx \sqrt{S_u}\left(\mathbf{P}_x^+ \otimes \mathbf{P}_y^+\right)\left(\tilde{\mathbf{L}} \bigodot \tilde{\boldsymbol{\Theta}}\right)(\mathbf{P}_x \otimes \mathbf{P}_y)\boldsymbol{\Phi} \tag{30}$$

4.2 | Aggregation

4.2.1 | Semianalytical integration

The cell-based sextuple integrals in Section 3.1 are excessively costly. Fortunately, through a change of coordinates, the inner integrals may be solved analytically, leaving only a triple integral to be solved numerically. The method is adapted from the derivation of the Mann model,[22] noticing that coherence depends only on the relative position between cells through $\vec{r}_{12}$, not on the individual positions of each cell. Let the elementary integral domain Ω be a volume V representing a cell of dimensions given by vector $\boldsymbol{\Delta}$. Let $\mathbf{r}$ and $\mathbf{s}$ be

$$\mathbf{r} = \vec{r}_{0_2 2} - \vec{r}_{0_1 1}; \ \ \mathbf{s} = \vec{r}_{0_2 2} + \vec{r}_{0_1 1} \tag{31}$$

Then, change variable in (13) and rewrite the integrals over cells as integrals over an infinite volume where parts that lie outside the cell are made zero, namely,

$$\overline{\gamma}_{V_{u_2 u_1}} = \frac{\int_\infty \int_\infty \gamma_{u_2 u_1}\left(\boldsymbol{p}, f, \mathbf{r} + \vec{r}_{0_1 0_2}\right)\psi_{V_{1_N}}\left(\frac{\mathbf{s}-\mathbf{r}}{2}\right)\psi_{V_{2_N}}\left(\frac{\mathbf{s}+\mathbf{r}}{2}\right)I_1\left(\frac{\mathbf{s}-\mathbf{r}}{2}\right)I_2\left(\frac{\mathbf{s}+\mathbf{r}}{2}\right)|\det(\mathbf{J})|\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{s}}{\int_\infty \int_\infty \psi_{V_{1_D}}\left(\frac{\mathbf{s}-\mathbf{r}}{2}\right)\psi_{V_{2_D}}\left(\frac{\mathbf{s}+\mathbf{r}}{2}\right)I_1\left(\frac{\mathbf{s}-\mathbf{r}}{2}\right)I_2\left(\frac{\mathbf{s}+\mathbf{r}}{2}\right)|\det(\mathbf{J})|\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{s}}; \quad \begin{cases} I_k(\mathbf{z}) = 1 & |\mathbf{z}_i| \le \boldsymbol{\Delta}_i \\ I_k(\mathbf{z}) = 0 & |\mathbf{z}_i| > \boldsymbol{\Delta}_i \end{cases} \quad k = 1,2; i = 1,2,3; \mathbf{z} \in \mathbb{R}^3 \tag{32}$$

where subscripts N, D refer to numerator and denominator, respectively. $\mathbf{J} = \frac{\partial\left[\vec{r}_{0_1 1} \ \ \vec{r}_{0_2 2}\right]^T}{\partial\left[\mathbf{r} \ \ \mathbf{s}\right]^T}$ is the Jacobian between old and new variables, yielding $|\det(\mathbf{J})| = \frac{1}{8}$. Integrating on $\mathbf{s}$ yields

$$\overline{\gamma}_{V_{u_2 u_1}} = \frac{\int_{-\boldsymbol{\Delta}}^{+\boldsymbol{\Delta}} \gamma_{u_2 u_1}\left(\boldsymbol{p}, f, \mathbf{r} + \vec{r}_{0_1 0_2}\right)\overline{\psi}_V \mathrm{d}\mathbf{r}}{\int_{-\boldsymbol{\Delta}}^{+\boldsymbol{\Delta}} \overline{\psi}_V \mathrm{d}\mathbf{r}} \tag{33}$$

with $\overline{\psi}_V$ being analytically integrated weights, reading for a unitary weighting function:

$$\overline{\psi}_V = \prod_{\ell=1}^{3} \Delta_\ell - |\mathbf{r}_\ell| \tag{34}$$

Note that the denominators in (32) and (33) may be integrated fully analytically, without need for coordinate change. However, for numerical accuracy, it is desirable that discretisation errors are present in both numerator and denominator and semianalytical integration should be applied to both.

Rotor-based averaging

A similar method may be used to reduce rotor-based quadruple integrals on a disc A in polar coordinates. In this case, $\mathbf{r} = \rho_\mathbf{r}[0 \;\; \cos\theta_\mathbf{r} \sin\theta_\mathbf{r}]^T$, $\mathbf{s} = \rho_\mathbf{s}[0 \;\; \cos\theta_\mathbf{s} \sin\theta_\mathbf{s}]^T$, and the aggregated coherence yields:

$$\overline{\gamma}_{A_{u_2u_1}} = \frac{\int_0^{2\pi}\int_0^{2R} \gamma_{u_2u_1}\left(\mathbf{p}, f, \mathbf{r} + \vec{r}_{0_1 0_2}\right)\overline{\psi}_A \rho_\mathbf{r} d\boldsymbol{\rho}_\mathbf{r} d\theta_\mathbf{r}}{\int_0^{2\pi}\int_0^{2R} \overline{\psi}_A \rho_\mathbf{r} d\boldsymbol{\rho}_\mathbf{r} d\theta_\mathbf{r}} \tag{35a}$$

$$\overline{\psi}_A = \int_0^{2\pi}\int_0^{\rho_{s_{max}}} \psi_{A_1}\left(\frac{\mathbf{s}-\mathbf{r}}{2}\right)\psi_{A_2}\left(\frac{\mathbf{s}+\mathbf{r}}{2}\right)\rho_\mathbf{s} d\boldsymbol{\rho}_\mathbf{s} d\theta_\mathbf{s} \tag{35b}$$

with R being the disc (i.e., rotor) radius and $\rho_{\mathbf{s}_{\max}}$ the upper bound for radial integration on $\mathbf{s}$ yielding

$$\rho_{\mathbf{s}_{max}} = \sqrt{4 - \rho_\mathbf{r}^2 \sin^2(\theta_\mathbf{r} - \theta_\mathbf{s})} - \rho_\mathbf{r}|\cos(\theta_\mathbf{r} - \theta_\mathbf{s})| \tag{36}$$

In practice, only the innermost integral in (35a) (i.e., radial in $\mathbf{s}$) may admit an analytical solution, provided that the weighting function is analytical and simple enough. Computational benefit may therefore only be seen when the number of averaging points becomes large. Another alternative for PSD calculations (i.e., elements 1 and 2 are one and the same) used in Sørensen et al[7] is to keep the original coordinates but assume $\gamma_{u_2u_1}$ is only dependent on the difference between azimuthal angles, again allowing for solving for one inner integral analytically.

4.2.2 | Numerical integration

The remainder of the integral is computed numerically through discretisation with a sufficient number of points, which may be chosen in an adaptive manner based on the tolerance ϵ_γ. One single point will typically be sufficient for low frequencies and large distances, Sørensen et al. even assumed that turbine-to-turbine aggregated coherence was equal to point-based.[8,10] For short distances (neighbouring grid points) and mid frequencies, however, the nonlinearity of coherence leads to significant errors when approximated to the integral domain's centre value and hence requires a finer mesh. At high frequencies, coherence between points may be neglected.

4.3 | Nearest correlation matrix

Random process generation using Cholesky factorisation as presented in Section 2.2 is ill-defined if the coherence matrix $\boldsymbol{\Gamma}$ (or its reduced-order version $\tilde{\boldsymbol{\Gamma}}$) is not positive-definite (abbreviated p.-d.). This may occur when the aggregated coherence function generating the matrix is not a p.-d. kernel (see Section 4.3.2). This appears to be an issue for dense farm-level coherence matrices (both point-based and aggregated, see Section 4.3.2), and may be a—if not the—major driver for computational cost. Two approaches may be considered to ensure p.-d. ness:

- Top-down approach, acting on the final matrix: non-intrusive, optimal in terms of induced errors, but not scalable; presented in Section 4.3.1
- Bottom-up approach, acting on the coherence function generating the matrix: intrusive (model-dependent), suboptimal, but scalable; presented in Section 4.3.2

It should be noted that making the matrix p.-d. will introduce a discrepancy from the original positive-indefinite coherence matrix. However, it should be kept in mind that the latter has been generated by a coherence function that is itself uncertain. Actually, as explained in Section 4.3.2, an exact model of the underlying correlated physical processes should always generate a p.-d. coherence matrix. Therefore, the introduced discrepancy may actually be seen as a correction rather than an error. This issue should be solved with increased knowledge in farm-wide coherence. A quantification of the underlying uncertainty is not actual at this point, the focus being on including farm-wide correlation *as much as we can* versus not doing it at all.

4.3.1 | The alternate projection algorithm

Finding the nearest p.-d. matrix is a common problem when working with real data. The nearest correlation matrix (NCM) algorithm was suggested by Higham[26] based on the alternate projection method of Dykstra.[27] It is based on artificially forcing p.-d.ness by replacing the negative eigenvalues of the matrix by slightly positive ones (in the following, uniformly equal to λ_+) through a projection onto the p.-d. space P_{PD} involving the eigenvalue-eigenvector decomposition. This alters the matrix's diagonal, which is then forced to unity in a second stage through another projection P_U. Doing so creates negative eigenvalues, which should then be replaced by positive ones, and so on. After n_{NCM} iterations and starting with $\hat{\boldsymbol{\Gamma}}_0 = \tilde{\Gamma}$ and $\Delta\hat{\Gamma}_{PD_0} = \mathbf{0}_{n_G \times n_G}$, the algorithm reads:

$$\begin{cases} \hat{\boldsymbol{\Gamma}}_{2n_{NCM}+1} & = \mathrm{P_{PD}}\left\{\hat{\boldsymbol{\Gamma}}_{2n_{NCM}} - \boldsymbol{\Delta}\hat{\boldsymbol{\Gamma}}_{\mathrm{PD}_{n_{NCM}-1}}\right\} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T + \lambda\mathbf{I}_{n_G} \\ & \quad with\, \hat{\boldsymbol{\Gamma}}_{2n_{NCM}} - \boldsymbol{\Delta}\hat{\boldsymbol{\Gamma}}_{PD_{n_{NCM}-1}} - \lambda\mathbf{I}_{n_G} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T;\ \boldsymbol{\Lambda}_+ = \max(\boldsymbol{\Lambda}_{ii}, 0), i = 1 \mathrm{to} n_G \\ \boldsymbol{\Delta}\hat{\boldsymbol{\Gamma}}_{\mathrm{PD}_{n_{NCM}}} & = \mathrm{P_{PD}}\left\{\hat{\boldsymbol{\Gamma}}_{2n_{NCM}+1} - \boldsymbol{\Delta}\hat{\boldsymbol{\Gamma}}_{\mathrm{PD}_{n_{NCM}-1}}\right\} - \hat{\boldsymbol{\Gamma}}_{2n_{NCM}+1} + \boldsymbol{\Delta}\hat{\boldsymbol{\Gamma}}_{PD_{n_{NCM}-1}} \\ \hat{\boldsymbol{\Gamma}}_{2n_{NCM}+2} & = \mathrm{P_U}\left\{\hat{\boldsymbol{\Gamma}}_{2n_{NCM}+1}\right\} = \hat{\boldsymbol{\Gamma}}_{2n_{NCM}+1} - \mathrm{Diag}\left(\hat{\boldsymbol{\Gamma}}_{2n_{NCM}+1}\right) + \mathbf{I}_{n_G} \end{cases} \tag{37}$$

with $\hat{\boldsymbol{\Gamma}}_{2n_{NCM}}$ being the current NCM candidate and $\boldsymbol{\Delta}\hat{\boldsymbol{\Gamma}}_{\mathrm{PD}_{n_{NCM}}}$ the Dykstra correction providing integral action ensuring convergence towards a p.-d. matrix that minimises the error with the original matrix in the least-square (Frobenius norm) sense.[26]

Each iteration involves a full eigen decomposition, whose cost is, although also $\mathcal{O}(n_G^3)$, at least one order of magnitude higher than Cholesky factorisation. Keeping $\boldsymbol{\Gamma}$ p.-d. for as many frequency components as possible is then crucial to reduce the overall cost. A first measure is to choose $\boldsymbol{\Delta}$ so that it decays evenly in x and y directions, hence the suggestion in Section 4.1. In practice, experience shows that a couple of iterations is sufficient, and the maximum (infinite norm) error is seldom higher than ϵ_γ.

The NCM algorithm is also a major driver for memory usage, as several temporary matrices of order n_G need to be stored simultaneously.

4.3.2 | Shrinking anisotropy of farm-level coherence

The alternate projections algorithm is appropriate for the first stage of Section 3.2.1, which involves only turbine points and hence relatively small matrices with higher need for minimising errors. For the second stage involving many more points and only affecting wake dynamics, it is desirable to trade accuracy for computational efficiency. This section presents a bottom-up alternative, starting from the point-based coherence model and using the analogy with Gaussian process models[28] where a kernel is used to generate the covariance matrix of a given quantity evaluated/measured at different points. In our case, $\boldsymbol{\Sigma}$ introduced in Section 4.1 may be seen as the kernel matrix of aggregated turbulence, used to generate the coherence matrix $\boldsymbol{\Gamma}$, directly related to the covariance matrix used in Gaussian process models through integration over frequency.

Take u as an example (straightforwardly extendable to v and w). Kernel theory tells that $\boldsymbol{\Gamma}_{uu}$ is p.-d. whenever $\boldsymbol{\Sigma}_{uu}$ is p.-d., which is in turn granted if the generating function (15) of $\boldsymbol{\Sigma}_{uu}$ is itself p.-d. For point-based coherence as used in turbine-level turbulence box modelling, this amounts to (2) p.-d. For aggregated coherence, it is trivial to show that p.-d.ness is preserved in the aggregation process, as the latter is merely a combination of sum of p.-d. kernels, translation of p.-d. kernels and multiplication by positive constants (best seen using the semianalytically integrated form of the aggregated coherence in (35a)) which all preserve p.-d.ness.[29] This is however valid only if the same number of averaging points is used for all kernel entries.

As a consequence, p.-d.ness of the aggregated coherence matrix is granted whenever the point-based coherence function belongs to or is constructed from well-known p.-d. kernel families, with two particularly relevant families in our case:

- Completely monotone radial basis functions[30] of the form $\gamma = f\left(\left\|\vec{r}_{12}\right\|^2\right)$ with $(-1)^n \frac{\mathrm{d}^n f}{\mathrm{d}\left\|\vec{r}_{12}\right\|^n} > 0\ \forall n \in \mathbb{N}$, that is, the kernel is only dependent on the norm of the position vector (i.e., distance) squared $\left\|\vec{r}_{12}\right\|^2$ and the corresponding derivatives are positive for any differentiation order.

This occurs when **a** and **b** are isotropic and positive ($\mathbf{a} = a\left(\left\|\vec{r}_{12}\right\|\right)[111]^T$, $\mathbf{b} = b\left(\left\|\vec{r}_{12}\right\|\right)[111]^T, a,b \in \mathbb{R}^+$) but not necessarily independent on $\left\|\vec{r}_{12}\right\|$.

- Anisotropic powered exponential kernels of the form $\gamma = \exp\left(-\left\|\mathbf{a} \otimes \vec{r}_{12}\right\| \frac{f}{u_\infty}\right)$. This occurs when $\mathbf{b} = \mathbf{0}_{3\times1}$ and all components of **a** are positive and independent on the position vector, but not necessarily isotropic.

While turbine-level coherence models have been built with p.-d.ness in mind and belong to one of these categories (typically the first one), the model from Vigueras-Rodriguez et al. described by (3a) falls in between the two and its p.-d.ness is hence not guaranteed, although close. The bottom-up approach to the NCM problem consists then in bringing the coherence function closer to one of the two p.-d. kernel families until the matrix becomes p.d., thus guaranteeing p.-d.ness in a suboptimal intrusive manner but with little computational effort.

Keeping in mind that this is only actual for the second stage mostly important for wake meandering and that longitudinal coherence has been found not to affect meandering to a large extent,[6] choice is made to preserve the distance-dependent lateral decay factor and rather help p.-d.ness by reducing anisotropy (i.e., moving towards the first of the two above-introduced kernels) by modifying the longitudinal decay factor, namely,

$$\widetilde{a_{long}} = \alpha a_{long} + (1-\alpha) a_{lat} \tag{38}$$

with $0 \le \alpha \le 1$. This method is referred to as *shrinking* [31] (originally for matrices and not kernels); α is the shrinking coefficient that we seek to maximise with p.-d.ness as a constraint.

4.3.3 | Kernel pregeneration

Independent from the approach, it is possible to pregenerate p.-d. aggregated coherence kernels for a specific wind condition/spectral model in a first simulation, to speed up later computations. p.-d. matrices may then be directly generated for other random variable realisations (i.e., seeds, enabling efficient generation of samples for, e.g., Monte Carlo simulations), turbine layouts, grid sizes (provided the precomputed kernel covers the entire farm area), or time discretisations and durations (provided the precomputed kernel matches frequency components). Using the bottom-up approach, this is straightforwardly done by aggregating the shrinked point-based kernel of Section 4.3.2. The top-down approach implies further considerations presented in the following.

With (27) describing its generation process from $\boldsymbol{\Sigma}_{uu}$, $\boldsymbol{\Gamma}_{uu}$ is a block-Toeplitz matrix whose blocks are themselves Toeplitz—that is, a *Toeplitz-block-Toeplitz* matrix—reading

$$\boldsymbol{\Gamma}_{uu} = \begin{bmatrix} \boldsymbol{\Gamma}_{\mathrm{uu}_{x_0}} & \boldsymbol{\Gamma}^T_{\mathrm{uu}_{x_1}} & \dots & \boldsymbol{\Gamma}^T_{\mathrm{uu}_{x_{n_{G_x}}}} \\ \boldsymbol{\Gamma}_{\mathrm{uu}_{x_1}} & \boldsymbol{\Gamma}_{\mathrm{uu}_{x_0}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{\Gamma}^T_{\mathrm{uu}_{x_1}} \\ \boldsymbol{\Gamma}_{\mathrm{uu}_{x_{n_{G_x}}}} & \dots & \boldsymbol{\Gamma}_{\mathrm{uu}_{x_1}} & \boldsymbol{\Gamma}_{\mathrm{uu}_{x_0}} \end{bmatrix}, \boldsymbol{\Gamma}_{\mathrm{uu}_{x_j}} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathrm{uu}_{i1}} & \boldsymbol{\Sigma}_{\mathrm{uu}_{i1}} & \dots & \boldsymbol{\Sigma}_{\mathrm{uu}_{i n_{G_y}}} \\ \boldsymbol{\Sigma}_{\mathrm{uu}_{i2}} & \boldsymbol{\Sigma}_{\mathrm{uu}_{i1}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{\Sigma}_{\mathrm{uu}_{i2}} \\ \boldsymbol{\Sigma}_{\mathrm{uu}_{i n_{G_y}}} & \dots & \boldsymbol{\Sigma}_{\mathrm{uu}_{i2}} & \boldsymbol{\Sigma}_{\mathrm{uu}_{i1}} \end{bmatrix} \tag{39}$$

where grid points are listed in column-major order. Hence, inversely, the nearest p.-d. kernel $\hat{\boldsymbol{\Sigma}}_{\mathrm{uu}}$ may be retrieved from the *nearest positive-definite Toeplitz-block-Toeplitz unit-diagonal* matrix of $\boldsymbol{\Gamma}_{uu}$. This implies adding into the alternate projection algorithm a projection P_{TBT} onto the Toeplitz-block-Toeplitz space, which is simply an averaging of diagonals (adapted from Grigoriadis et al.[32]):

$$P_{TBT}\{\boldsymbol{\Sigma}_{uu}\}_{ij} = \frac{1}{(n_{G_x}-i+1)(n_{G_y}-j+1)} \sum_{k=1}^{n_{G_x}-i+1} \sum_{\ell=1}^{n_{G_y}-j+1} \hat{\boldsymbol{\Gamma}}_{\mathrm{uu}_{(i+k-2)n_{G_y}+j+\ell-1,(k-1)n_{G_y}+\ell}} \tag{40}$$

where indexing of $\hat{\boldsymbol{\Gamma}}$ refers to matrix elements and not iterations. Follows a reconstruction of $\hat{\boldsymbol{\Gamma}}_{\mathrm{uu}}$ using (27) or (39). Although overlooked by Grigoriadis et al. in their paper, it can be shown that $\mathrm{P_{TBT}}$ does not need to be attributed a Dykstra correction[27]—like $\mathrm{P_U}$, but unlike $\mathrm{P_{PD}}$—as it is linear (the projection of a sum equals the sum of projections of its elements).

5 | NUMERICAL EXPERIMENTS

The goal of this section is to illustrate turbulence aggregation and the associated concepts and methods presented in this paper. It shows the consistency and efficiency of the approach in a qualitative manner, while detailed validation against higher-fidelity farm simulations and measurements being more related to calibration are left as further work.

5.1 | Turbine scale

In order to validate the aggregation concept, a comparison with turbine-scale simulations has been performed through an academic case study. The domain is as illustrated in Figure 2:

- on the one hand three adjacent (closely-spaced) turbine points simulated using the model presented in this paper
- on the other hand a point-based turbulence box covering the same area, simulated in NREL's TurbSim[20] synthetic turbulence simulation code used a reference. Aggregation—that is, space averaging—is then performed in the time domain over each of the three virtual rotor areas.

The Kaimal spectrum and corresponding coherence model defined by IEC standard 61400-1 ed.3 with turbulence class B have been used in the two simulations, for a wind speed of 10 m/s, and the DTU 10MW[33] reference wind turbine rotor to define hub height and rotor area for averaging. No farm-level turbulence model has been used (this will result in somewhat lower correlation between rotors). Ten seeds of each simulation are considered, and mean spectral quantities (PSD and coherence) over all seeds are looked at for comparison, in addition to theoretical aggregated values as defined by (13) to (15). Figure 3A shows point-based versus aggregated PSDs for the u component, with horizontal linear wind shear s_z included for further illustration of the consistency of wind field reconstruction. Figure 3B shows the coherence between two adjacent rotors, matching the theoretical value well.

Figure 4 illustrates the concept of turbulence box reconstruction from aggregated values, using a modified version of NREL's TurbSim able to sync aggregated values over multiple adjacent rotors. Figure 4A show timeseries of aggregated values for the parent (aggregated solution) and child (point-based reconstruction). The small discrepancy there may be attributed to discretisation errors, the resolution being rather coarse to cope with the large domain while maintaining computational efficiency. Figure 4B shows the PSD of a rotationally sampled wind field, averaging over the blades with a weighting function proportional to radial location (consistently with actuator-disc aggregation in polar coordinates as done by Sørensen et al.[7]) and taking rotor speed as constant. The good match between regular (fully random) and synced turbulence box generation shows the ability of the latter to consistently reconstruct high-resolution wind fields while matching predefined aggregated values, and generally speaking the soundness of the suggested method for constrained synthetic turbulence generation.

5.2 | Farm scale

A case study is performed on the TotalControl reference wind power plant with 32 DTU 10MW (rotor diameter of 178.3 m) turbines placed in a staggered pattern with 5D spacing,[34] in the same wind conditions as in Section 5.1. The grid resolution is 200 m in the longitudinal direction, and 80 m in the lateral direction (it may be deduced that turbine and grid points do not coincide). The farm-scale spectral model of Vigueras-Rodriguez et al[9,10] with original parameter values for $(a_{\text{long}}, c_1, c_2, \kappa)$ is chosen here for its simplicity and validation against power measurements from multiple Danish offshore wind farms. Multiple 1-h realisations are generated, playing with a selection of parameters for illustration of the model's capabilities regarding power fluctuations (turbine points) and wake meandering (grid points).

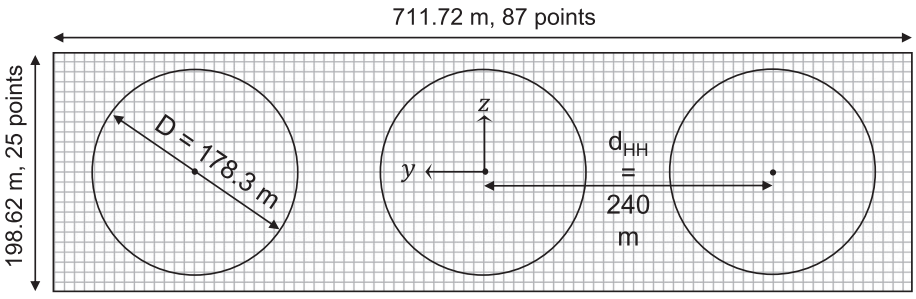

FIGURE 2 Domain used for verification against point-based Gaussian process synthetic turbulence generation.

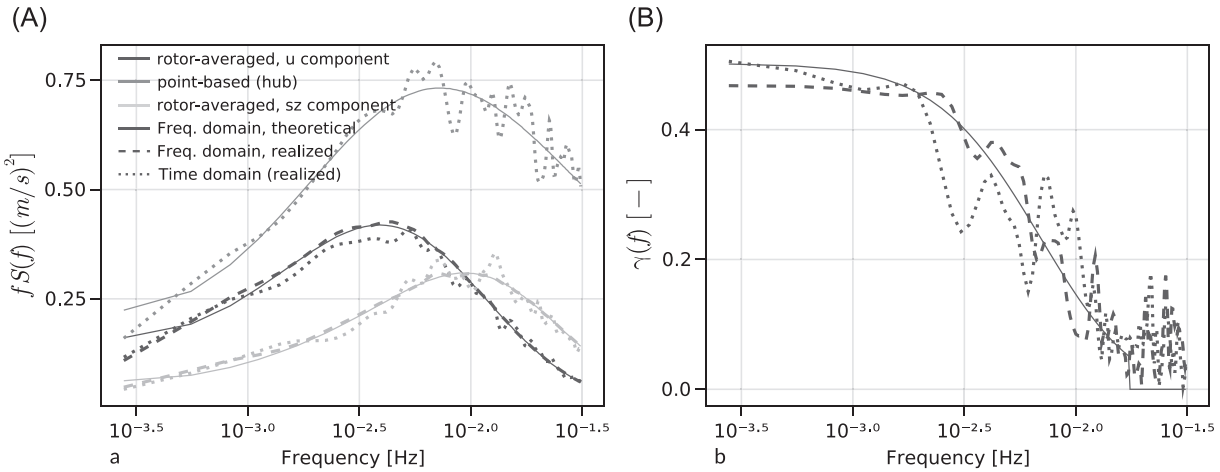

FIGURE 3 Spectrum and coherence, point-based versus aggregated and theoretical versus realised. "Freq. domain theoretical"—referring to frequency-domain averaging, represented by thin continuous lines—is the result of the aggregation process in (14) and (15). The maximum frequency used to compute coherence is 0.02 Hz (theoretical values drop to zero passed this value). "Freq. domain realised" is the average over realisations obtained by (8). "Time domain realised" is the result of space- and seed-averaging of constrained turbine-level simulations using (24) to (26b). Note that point-based values are not realised because they are undefined when using frequency-domain averaging.

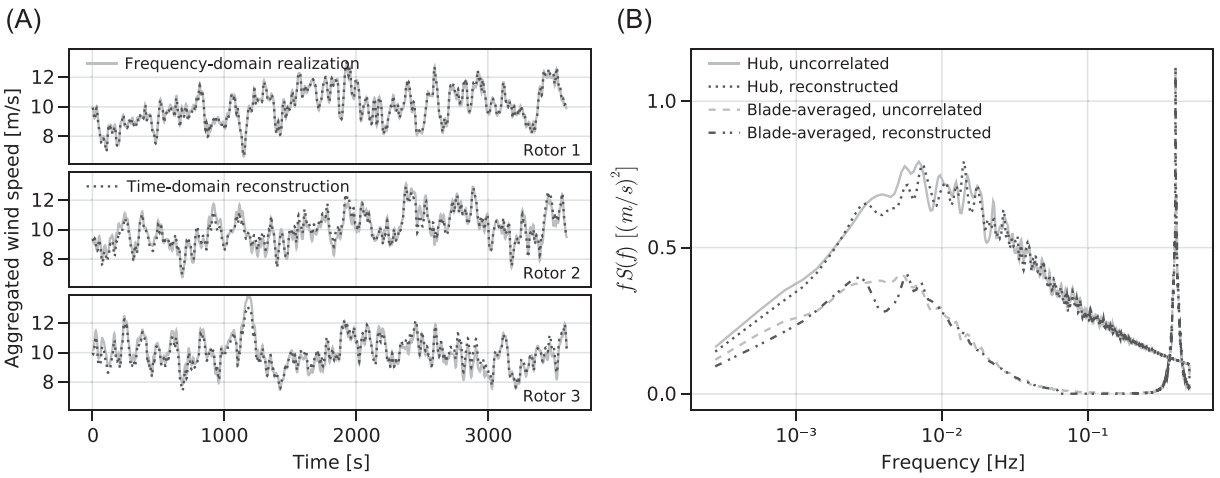

FIGURE 4 Timeseries and spectra illustrating point-based reconstruction from aggregated values. Plot a show the comparison between the original aggregated values from (11) and the space average of the reconstructed field using (24) to (26b). Plot b shows the consistency of reconstructed fields with respect to uncorrelated fields (original version of TurbSim) used a reference, through hub and blade-averaged values.

5.2.1 | Aggregated coherence

Figure 5A shows the low-pass filtering effect of aggregation on the wind spectrum and the additional low-frequency content suggested by Vigueras-Rodriguez et al.[9] Figure 5B,C shows the effect of aggregating coherence versus the original point-based model of Vigueras-Rodriguez et al.[10] It is seen that aggregated coherence is larger than point-based, due to the nonlinearity of coherence: parts of the averaging domains that are closer to each other than the distance between the domains' center points will have larger coherence than the center-to-center—that is, point-based—value, while parts that are further away will have lower coherence. The former effect being more pronounced than the latter, the average value will be in total larger than the center-to-center one.

This effect increases significantly with frequency. When considering only turbine points as done by Sørensen et al., that is, distances above five diameters—in the current case about 1 km—approximating aggregated coherence to point-based may be reasonable. For grid points however, aggregated values should be used—though the impact of inaccuracies on top of the existing uncertainty on wake dynamics may be questioned (see Section 5.2.3).

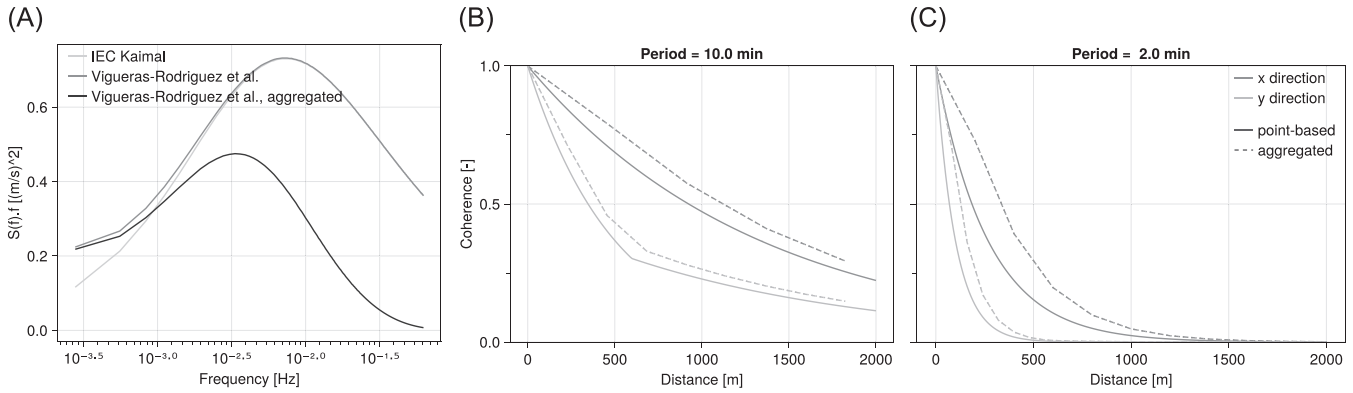

FIGURE 5 Aggregated and point-based wind spectra with various models as function of frequency (plot A); aggregated versus point-based coherence functions as function of distance in the longitudinal and lateral directions (plots B and C), for two different periods.

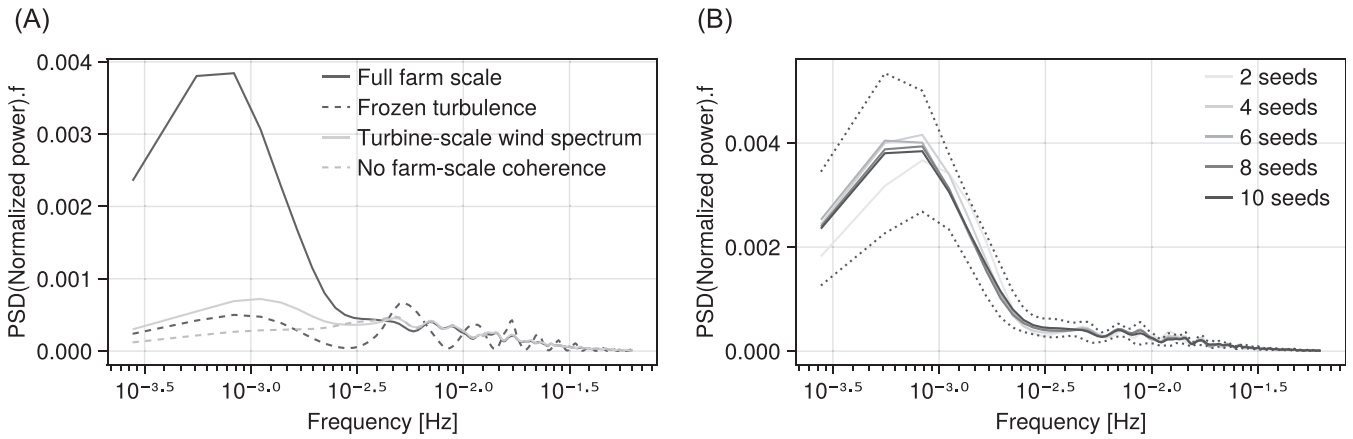

FIGURE 6 Power fluctuations.

5.2.2 | Power fluctuations

The power is modelled using a simplistic representation of the wind turbine controller and a power coefficient:

$$P = \min\left(C_{p_{max}} \frac{1}{2} \rho A (U_{mean} + u)^3, P_{rtd}\right) \tag{41}$$

with P_{rtd} being the rated power and $C_{P_{max}}$ the maximum power coefficient. Contributions from individual turbine are then summed up and normalised by installed capacity before computing PSD and averaging over realisations. Figure 6A shows the effect of farm-scale modelling features (i.e., farm-scale wind spectrum model, farm-scale coherence function model[‡] and lifting the frozen-turbulence assumption). It is seen that missing any of the farm-scale features leads to underestimation of low-frequency fluctuations (periods larger than 5 min). Perhaps even more interesting are the frequency-domain oscillations observed when using the frozen-turbulence assumption, which are actually as expected since the simplistic time-delay correlation leads to fluctuations between upwind and downwind turbines being more or less in phase depending on frequency. At this stage, a conclusion would be that as far as power fluctuations are concerned, it is better to use no coherence model than any coherence model that uses the frozen-turbulence assumption.

Figure 6B aims at showing the importance of running Monte Carlo simulations based on multiple stochastic realisations, just like turbine-level analyses use multiple 10-min high-resolution realisations. The dotted line in the figure correspond to ± the standard deviations over 10 realisations, representing the uncertainty that may be expected from studies based on a single simulation, which is commonly the case when costly LES is used.

[‡]Using standard turbine-scale coherence models at farm scale yields in practice zero coherence between turbines due to the large decay factor.

5.2.3 | Wake meandering

While the frozen-turbulence assumption has been deemed valid for the modelling of wake meandering by Shaler et al,[6] significant differences were observed in wake meandering pathlines when compared to LES and were found heavily dependent on the calibration process. While this is still hypothetical at this stage and remains to be proven, lifting the frozen-turbulence assumption by use of aggregated synthetic turbulence generation is believed to provide a more solid representation closer to physics and hence with a lesser need for calibration and filtering. Figure 7 shows alongwind spatial variations of the lateral wind speed in the wake of turbine 1, obtained by taking values along a line parallel to the X axis for a particular realisation. Figure 8 shows pathlines in the XY plane of parcels entering the domain for three realisations, computed using a simple transport equation in Lagrangian coordinates similar to that used for wake meandering. The starting point coincides with the domain's upwind boundary and phase angles are identical between cases, so they all share the same initial conditions. As expected from Figure 7, frozen turbulence removes a large part of wind fluctuations so the pathlines show less amplitude and are smoother. Not using farm scale-specific models for the wind spectrum and coherence function also yields differences from the fully coherent farm-scale baseline, which relate more to model calibration, while the significantly different trends obtained with the frozen-turbulence assumption put its validity into question.

5.2.4 | Efficiency

For the above-described case study, generating 10 seeds instead of 1 increased CPU time—not to be confused with real time, as the calculation is partly multi-threaded on a machine with multiple cores—by a factor of 4. Considering a farm of four times the size in terms of area with 128 turbines increased CPU time by a factor of 13. This illustrates the curse of dimensionality, stemming from the coherence matrix's building and decomposition phases. These phases are common to all realisations, so the computational speed increases less than linearly with the number of seeds, and its relative effect becomes smaller as farm size increases.

Figure 9A shows the effect of the NCM algorithm and tolerance (ϵ_γ in Sections 4.1 and 4.2.2, also used to defined convergence criteria in Section 4.3) on the processing of the coherence matrix (NCM algorithm + Cholesky decomposition) for the first frequency components. Decreas-

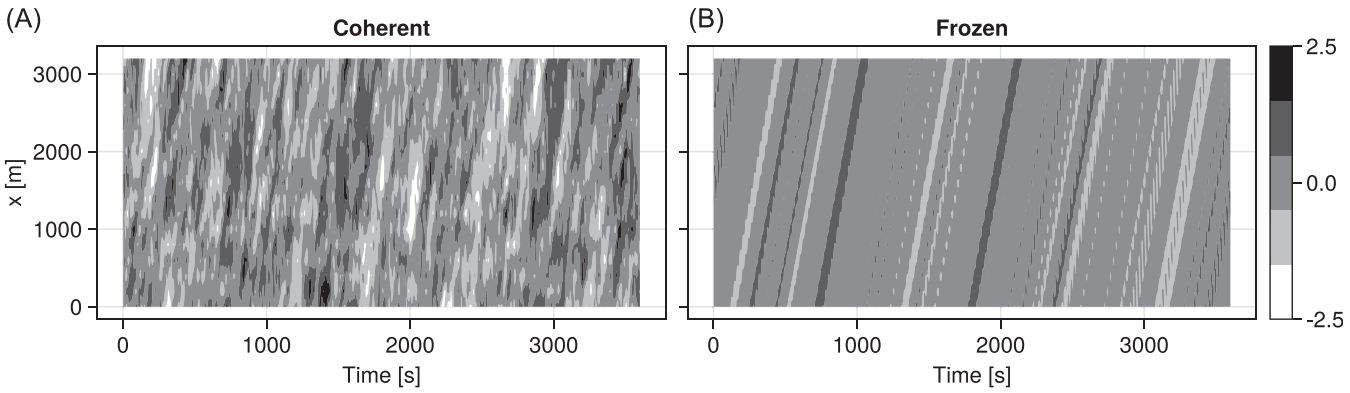

FIGURE 7 Slice along X axis ($Y = 0$) of a sample aggregated wind field realisation, lateral component.

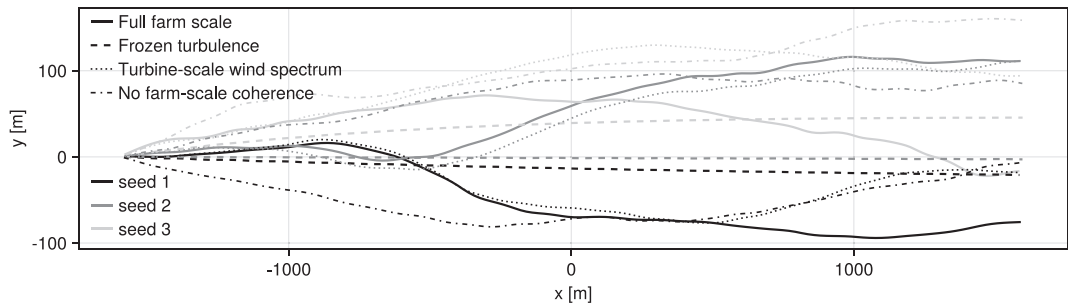

FIGURE 8 Samples of horizontal meandering pathlines for three different realisations using four different models (fully featured, missing longitudinal coherence, missing farm-scale spectrum model, and missing farm-scale coherence model).

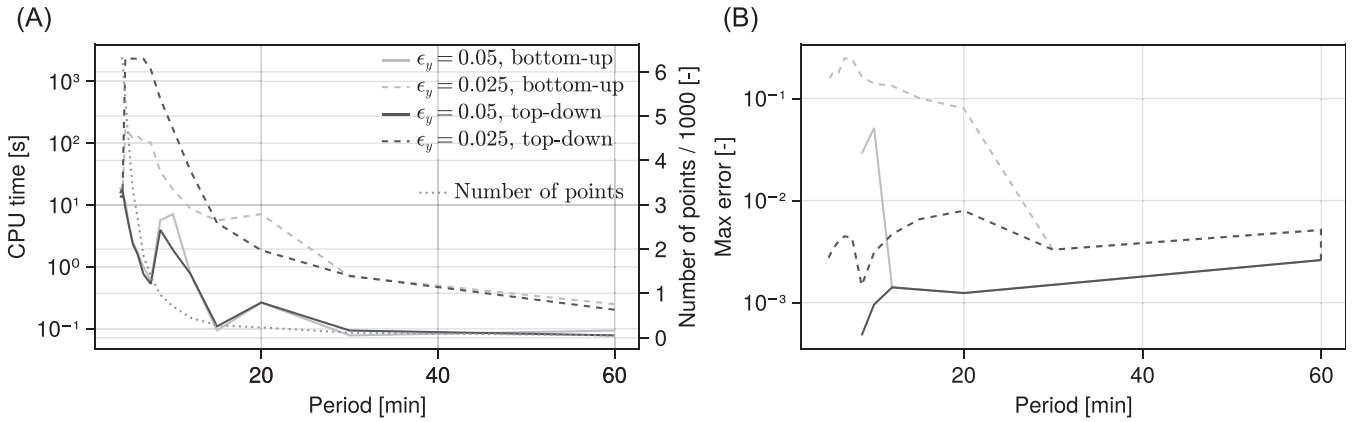

FIGURE 9 CPU times, grid size, and NCM-induced error as function of tolerance and NCM algorithm.

ing tolerance in the 128-turbine farm yields denser adaptive grids enabling illustrating the effect of grid size. While top-down NCM is slightly faster for smaller matrices (as it does not imply reaggregating coherence), its efficiency becomes critically poor as the number of points increases. Looking at the maximum error (infinite norm of the difference between the original non-positive-definite matrix and the NCM) in Figure 9B, the top-down NCM algorithm yields values two order of magnitudes lower than its bottom-up counterpart. Although the maximum error induced by the bottom-up algorithm may seem large (up to 0.25), the effect on results is better represented through the average weighted absolute error defined as $\frac{\sum\sum|\gamma_{original}-\gamma_{NCM}|\gamma_{original}}{n_G^2}$, which is about five times smaller.

6 | CONCLUSION

A method for farm-wide correlated synthetic turbulence generation based on Gaussian process generation has been suggested. It is based on the concept of aggregated turbulence and more specifically aggregated coherence, which characterises large turbulent vortices responsible for global wind turbines loads, wake dynamics and farm-wide power fluctuations. The method consists in averaging coherence in space in the frequency domain prior to Gaussian process generation (i.e., translating to time domain through inverse Fourier transform). This enables a drastic reduction in the number of degrees of freedom and hence, among others, lifting the frozen-turbulence assumption used in point-based representations. Major improvements to the original method suggested by P. Sørensen's research group at Risø/DTU are (1) the modelling of turbulence not only at but also between turbines, as input to farm simulation codes using the dynamic wake meandering model, and (2) the reconstruction of turbine-scale point-based (opposite to aggregated) high-resolution wind fields in a correlated manner. Key enabling mathematical subtleties for practical implementation are thoroughly presented. The model has been verified against a state-of-the-art point-based counterpart at small scale. Its superiority to turbine-level models for the modelling of power fluctuations and potentially wake dynamics in large wind farms is then illustrated.

This paper focuses on the theoretical background behind the method rather than on its use and validation. More work is needed to validate the methods against measurements of power fluctuations from recent large wind farms, updating the work of Sørensen et al. Deeper understanding and validation should also be obtained by comparison with LES simulations and hindcast weather data. This will require a calibration of underlying spectral models and a tuning of parameters such as grid resolution.

ACKNOWLEDGEMENTS

NREL is gratefully acknowledged for the open-source development of the FAST ecosystem for which—among other uses—the present method is designed, and on which it is based in its implementation. This work has been financed by the European Commission through the WATEREYE project (grant nr. 851207) and the Research Council of Norway through the CONWIND project (grant nr. 304229) and FME NorthWind research centre (grant nr. 321954).

DATA AVAILABILITY STATEMENT

The open-source code and input files used to generate the data that support the findings are available from the corresponding author upon reasonable request.

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1002/we.2875.

ORCID

Valentin Chabaud https://orcid.org/0000-0002-2628-2072

REFERENCES

1. Veers P, Dykes K, Lantz E, et al. Grand challenges in the science of wind energy. *Science*. 2019;366(6464).
2. Madsen HA, Larsen GC, Larsen TJ, Troldborg N, Mikkelsen R. Calibration and validation of the dynamic wake meandering model for implementation in an aeroelastic code. *J Solar Energy Eng*. 2010;132(4):41014.
3. IEC 61400-3-1. Wind energy generation systems—part 3-1: design requirements for fixed offshore wind turbines. In: International Electrotechnical Commission, Geneva; 2019.
4. Jonkman JM, Shaler K. *Fast. Farm User's Guide and Theory Manual*: National Renewable Energy Laboratory; 2018.
5. Openfast. https://github.com/OpenFAST/openfast
6. Shaler K, Jonkman J, Doubrawa Moreira P, Hamilton N. *Fast. Farm Response to Varying Wind Inflow Techniques*. Boulder, CO (USA): National Renewable Energy Lab.(NREL); 2019.
7. Sørensen P, Hansen AD, Rosas PAC. Wind models for simulation of power fluctuations from wind farms. *J Wind Eng Ind Aerodyn*. 2002;90(12-15):1381-1402.
8. Sørensen P, Cutululis NA, Vigueras-Rodríguez A, et al. Modelling of power fluctuations from large offshore wind farms. *Wind Energy: Int J Progress Appl Wind Power Convers Technol*. 2008;11(1):29-43.
9. Vigueras-Rodríguez A, Sørensen P, Cutululis NA, Viedma A, Donovan MH. Wind model for low frequency power fluctuations in offshore wind farms. *Wind Energy*. 2010;13(5):471-482.
10. Vigueras-Rodríguez A, Sørensen P, Viedma A, Donovan MH, Lázaro EG. Spectral coherence model for power fluctuations in a wind farm. *J Wind Eng Ind Aerodyn*. 2012;102:14-21.
11. Cheynet E, Jakobsen JB, Reuder J. Velocity spectra and coherence estimates in the marine atmospheric boundary layer. *Bound-Layer Meteorol*. 2018;169(3):429-460.
12. Burton T, Jenkins N, Sharpe D, Bossanyi E. *Wind Energy Handbook*: Wiley; 2013.
13. Shinozuka M, Jan C-M. Digital simulation of random processes and its applications. *J Sound Vib*. 1972;25(1):111-128.
14. Huang G, Liao H, Li M. New formulation of Cholesky decomposition and applications in stochastic simulation. *Probabilistic Eng Mech*. 2013;34:40-47.
15. Veers PS. Three-Dimensional Wind Simulation, Albuquerque, NM (USA), Sandia National Labs.; 1988.
16. Apt J. The spectrum of power from wind turbines. *J Power Sources*. 2007;169(2):369-374.
17. Larsén XG, Vincent C, Larsen S. Spectral structure of mesoscale winds over the water. *Q J R Meteorol Soc*. 2013;139(672):685-700.
18. Larsén XG, Larsen SE, Petersen EL, Mikkelsen TK. A model for the spectrum of the lateral velocity component from mesoscale to microscale and its application to wind-direction variation. *Bound-Layer Meteorol*. 2021;178(3):415-434.
19. Göçmen T, Larsén XG, Imberger M. The effects of open cellular convection on wind farm operation and wakes. In: Journal of physics: Conference series, Vol. 1618 IOP Publishing; 2020:62014.
20. Jonkman BJ. Turbsim user's guide: Version 1.50. 2009.
21. Chen Y, Guo F, Schlipf D, Cheng PW. Four-dimensional wind field generation for the aeroelastic simulation of wind turbines with lidars; 2022.
22. Mann J. Wind field simulation. *Probabilistic Eng Mech*. 1998;13(4):269-282.
23. Nielsen M, Larsen GC, Hansen KS. Simulation of inhomogeneous, non-stationary and non-Gaussian turbulent winds. In: Journal of physics: Conference series, Vol. 75 IOP Publishing; 2007:12060.
24. Rinker JM. Pyconturb: an open-source constrained turbulence generator. In: Journal of physics: Conference series, Vol. 1037 IOP Publishing; 2018:62032.
25. Petersen KB, Pedersen MS. The matrix cookbook; 2012.
26. Higham NJ. Computing the nearest correlation matrix: a problem from finance. *IMA J Numer Anal*. 2002;22(3):329-343.
27. Boyle JP, Dykstra RL. A method for finding projections onto the intersection of convex sets in hilbert spaces. *Advances in Order Restricted Statistical Inference*: Springer; 1986:28-47.
28. Williams CKI, Rasmussen CE. Gaussian processes for regression; 1996.
29. Fasshauer GE. Positive definite kernels: past, present and future. *Dolomites Res Notes Approx*. 2011;4:21-63.
30. Schoenberg IJ. Metric spaces and completely monotone functions. *Ann Math*. 1938;39:811-841.
31. Higham NJ, Strabic N, Sego V. Restoring definiteness via shrinking, with an application to correlation matrices with a fixed block. *SIAM Rev*. 2016;58(2):245-263.
32. Grigoriadis KM, Frazho AE, Skelton RE. Application of alternating convex projection methods for computation of positive Toeplitz matrices. *IEEE Trans Signal Process*. 1994;42(7):1873-1875.
33. Bak C, Zahle F, Bitsche R, et al. The DTU 10-MW reference wind turbine. Danish wind power research 2013; 2013.
34. Andresen S, Madariaga A, Merz K, Meyers J, Munters W, Rodriguez C. Reference wind power plant. TotalControl Deliverable D1.3; 2018.

How to cite this article: Chabaud V. Synthetic turbulence modelling for offshore wind farm engineering models using coherence aggregation. *Wind Energy*. 2023;1-20. doi:10.1002/we.2875