Updates on the OpenFAST Lidar Simulator

Updates on the OpenFAST Lidar Simulator

Feng Guo[1,4], David Schlipf[1], Hailong Zhu[2], Andy Platt[3], Po Wen Cheng[4], Florian Thomas[5]

[1]Wind Energy Technology Institute, Flensburg University of Applied Sciences, 24943 Flensburg, Germany
[2]Nanjing Movelaser Co., Ltd, 210038, Nanjing City, Jiangsu Province, China
[3]National Renewable Energy Laboratory (NREL), 15013 Denver West Parkway, Golden, Colorado 80401, USA
[4]Stuttgart Wind Energy, Stuttgart University, 70569 Stuttgart, Germany
[5] sowento GmbH, Stuttgart, Germany

E-mail: feng.guo@hs-flensburg.de
Keywords: Lidar Simulation, Wind Turbine, OpenFAST, Lidar-Assisted Control

Abstract. Lidar systems are able to measure the wind speed remotely by detecting the aerosol movement caused by wind. A nacelle-based lidar system scanning the wind in front of a wind turbine can provide a preview of the incoming wind before the wind interacts with the turbine. Implementing a realistic lidar simulator into the wind turbine aero-elastic simulation tool can be beneficial for various wind energy related fields, such as lidar-assisted control, load validation, and load monitoring. In previous work, a lidar simulation module has been integrated into the open-source aero-elastic simulation tool OpenFAST, covering already lidar characteristics like different scan patterns, the volume averaging along the beam, and the coupling with the nacelle motion due to turbine tower dynamics. This paper focuses on adding further features to the lidar simulation module of OpenFAST to make the lidar simulation more realistic: the evolving turbulence, the blade blockage effect, and the adjustable data availability. Further, the wind preview quality with the new features is assessed.

1. Introduction

This work focuses on integrating three new features into the lidar simulation module[1] of the aero-elastic simulation tool OpenFAST[2] to make the coupled simulation more realistic, as shown by Figure 1.

The first feature is the evolution of turbulent wind. In the previous version, the lidar system is scanning the exact same wind field which is also used for the aero-elastic simulation of the wind turbine, assuming Taylor's frozen turbulence theory [23]. As a result, there is only a shift in time between the lidar-measured wind and the turbine-experienced wind. Specifically, when assessing the benefits of lidar-assisted turbine control, this assumption tends to overestimate the benefit brought by lidar. In reality, the turbulence is evolving as it propagates towards the turbine [7], thus the turbine will not be exposed to the identical disturbances as that measured by the lidar in advance. So, wind evolution should be considered in order to provide more accurate and realistic simulation of a lidar system [14].

[1] commit 829511a on 13 March 2020,https://github.com/sowentoDavidSchlipf/openfast/tree/f/lidarsim
[2] last access on 26 March 2022, https://openfast.readthedocs.io/en/main/

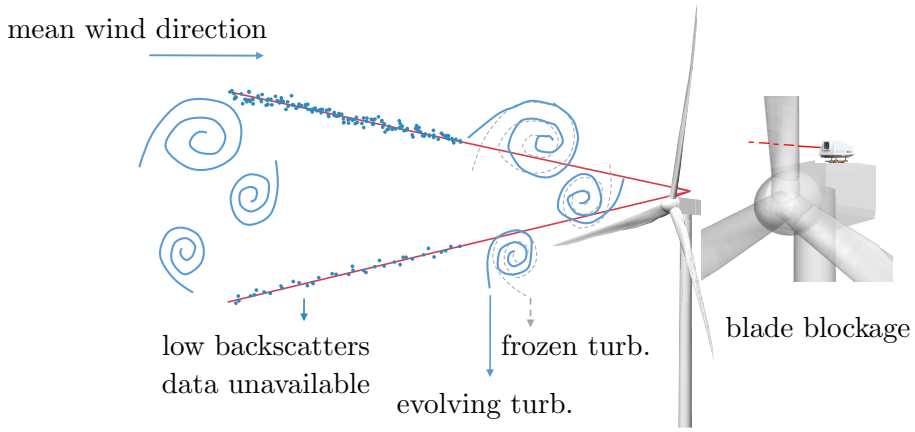

Figure 1. A sketch of the new features that are updated in the OpenFAST lidar simulation module. Note that the frozen turbulence remains unchanged from upstream to downstream. A typical four beam lidar is shown, the right chart shows a situation where one beam (dashed lines) is blocked by a blade.

The second new feature is the blade blockage effect. A forward-looking nacelle lidar is usually installed behind the rotor. The lidar beam can be blocked by the turbine blades during operation and the blades draw additional peaks in the Doppler spectrum for the lidar, resulting in wrong LOS measurements [4]. For high wind speeds, the spectral peaks caused by the blade can be distinguished by the lidar system because they appear in different frequency ranges compared to the peaks drawn by wind. Also, the carrier-to-noise ratio(CNR) values are usually low when the lidar beam is blocked by the blade. The blade blockage has an impact on the quality of the derived signals, because some of the desired measurements can not be obtained. Especially for assessing lidar-assisted control performance, this blockage effect should be included to verify the lidar data processing algorithms.

The third updated feature is the adjustable data availability. Wind lidars are based on the Doppler effect and they are designed to provide line-of-sight (LOS) measurements for all the specified measuring range gates [8]. The availability defines the percentage of usable data retrieved by a lidar over the total measurement attempts [3]. In the previous version of the lidar simulator, full availability was assumed. However, the availability can drop when the back-scattered power is not able to form a distinguishable Doppler spectrum [3]. The availability of LOS measurements is usually quantified by the CNR of the laser signal, which depends on aerosol back-scatter, humidity and precipitation in the atmosphere [1]. Measurements with CNR values below a certain threshold will be treated as invalid. When the data availability drops, special treatment needs to be carried out to interpolate the missing lidar data to avoid interference with important turbine structural modes [20]. The changeable data availability should be taken into consideration for simulating lidar since the lidar is subjected to various atmospheric conditions in a real application.

The rest of this paper first introduces the methods to simulate the three features mentioned above sequentially. Then the wind preview quality using the updated realistic simulator is assessed and the conclusions are drawn.

2. Integrating the Evolving Turbulence

This section introduces the evolving turbulence generation tool *evoTurb* and explains the integration of the tool into the OpenFAST lidar simulation.

2.1. evoTurb: the turbulence evolution tool

Prior to this work, an open-source tool *evoTurb* [6] has been developed, which is written in both *Python* and *Matlab* languages[3]. The main function of *evoTurb* is to generate a 4D stochastic turbulent wind field, enabling simulating the evolution of turbulence from upstream to downstream positions. One advantage of this tool is that it uses the existing 3D turbulence simulation tools TurbSim (initially developed on the basis of the method proposed by [24]) and Mann turbulence generator (based on the Mann [15] uniform shear model) to produce the 4D turbulent wind field.

Apart from the typical parameters already required in 3D turbulence simulation, *evoTurb* only requires a definition of longitudinal coherence (the decorrelation of turbulence by longitudinal spatial separation) and the interested upstream positions. Currently, only the Bladed-style wind field is supported by *evoTurb* and the updated lidar simulator in this work. *evoTurb* writes out a 3D turbulent wind field in ".wnd" format used for the simulation of the aero-elastic response of the turbine at the position of the rotor, and other 3D upstream wind fields in ".evo" format used for the simulation of lidar measurements. The main benefit of writing the wind field separately is that the simulation results at the rotor position without LAC can be performed independently. If the users specifies another lidar upstream measurement distance or longitudinal coherence, only the ".evo" format file needs to be updated. Apart from the wind field data, a ".sum" file will also be written which includes some additional information about the wind field, e.g. the hub-height mean wind speed and the turbulence intensity.

2.2. Lidar measurement simulation

To simulate the lidar measurement, the identical algorithm developed by the previous simulator is used, where the wind components u,v and w are projected onto the beam direction to find the single point line-of-sight (LOS) wind speed. Then a weighted sum of several single-point LOS speeds along a beam is calculated to approximate the weighting effect caused by the lidar probe volume. In order to calculate the LOS speed at upstream positions, the upstream wind fields are imported into the *lidarsim* module of the code. In the previous version, the same wind field which was used for the simulation of the wind turbine aerodynamics was also used for simulating the lidar system assuming the Taylor's Frozen Turbulence hypothesis.

3. Simulating the Blade Blockage Effect

In this section, the algorithm used to detect the blade blockage status and the implementation in the OpenFAST code frame work will be presented.

3.1. Blade blockage detection

Forward-looking nacelle lidar systems are usually installed behind the rotor. In practice, the lidar measurements from behind the rotor are affected by the blade sections close to the root, where the blade geometry is mostly transitioning from a circle to the thickest airfoils. It is possible to derive the detailed geometry of these sections if the airfoil coordinates, the twist angles, and the chord lengths are provided. In OpenFAST, the above information is usually defined for several representative blade element nodes [13]. To detect the blockage caused by the blades, we approximate the problem by simplifying the blade geometry as two triangles, whose 3D Cartesian coordinates can be derived from the information carried by the blade element nodes. After the coordinates of the two triangles are obtained, the ray-triangle intersection algorithm by Tomas Möller and Ben Trumbore [16] is used to detect the blockage status.

As shown by Figure 2, c_1 and c_2 denote the aerodynamic centers of the two airfoil cross sections, l_1 and l_2 correspond to the leading edges of the airfoils, and t_1 and t_2 are the trailing

[3] last access on 26 March 2022, `https://github.com/SWE-UniStuttgart/evoTurb`

edges. The lidar beam originates from b_0 and measures at b_2. The coordinates of the two triangles ($\triangle t_1 l_1 t_2$ and $\triangle l_1 l_2 t_2$) are our targets. In order to determine the blade elements that are possible to block the lidar beam, the position of the point b_1, that lies on both the lidar beam and on the rotational plane, is first calculated. Note that the rotational plane is the one perpendicular to the rotational axis in Figure 2. The coordinate of b_1 can be computed by

$$\boldsymbol{b_1} = \boldsymbol{b_0} + \overrightarrow{b_0 b_2} \frac{\overrightarrow{b_0 o} \cdot \overrightarrow{\boldsymbol{n}_\mathbf{r}}}{\overrightarrow{b_0 b_2} \cdot \overrightarrow{\boldsymbol{n}_\mathbf{r}}}, \tag{1}$$

where "$\cdot$" denotes the dot product, $\boldsymbol{b_0}$ is the coordinate vector of point b_0, $\overrightarrow{\boldsymbol{n}_\mathbf{r}}$ is the normal vector of the rotational plane, and $\overrightarrow{b_0 b_2}$, $\overrightarrow{b_0 o}$ are vectors defined by the specific start and end points. After that, the radial distance $|\overrightarrow{ob_1}|$ is compared with the radial distances ($|\overrightarrow{oc_1}|$ and $|\overrightarrow{oc_2}|$) from the blade element nodes to the rotor rotational center o. If $|\overrightarrow{oc_1}| < |\overrightarrow{ob_1}| < |\overrightarrow{oc_2}|$, then the two blade nodes are chosen to detect the blockage in the next step. By this approach, we reduce the computational effort by avoiding calculation of blockage for all the blade element nodes.

In the next step, the leading edge coordinates for l_1, l_2 and the trailing edge coordinates for t_1 and t_2 are obtained by

$$\boldsymbol{l_i} = \boldsymbol{c_i} - 0.25 L_i \overrightarrow{\boldsymbol{n}_\mathbf{b}} \quad \text{and} \quad \boldsymbol{t_i} = \boldsymbol{c_i} + 0.75 L_i \overrightarrow{\boldsymbol{n}_\mathbf{b}} \tag{2}$$

with L_i the chord length of the ith the blade element node, $\boldsymbol{c_i}$ the coordinate of the ith aerodynamic center, and $\overrightarrow{\boldsymbol{n}_\mathbf{b}}$ the orientation vector that aligns with the chord length direction. Here it is assumed that the aerodynamic center is located at 25% of the chord line. After $\boldsymbol{l_i}$ and $\boldsymbol{t_i}$ are calculated, the blockage status is finally determined by calculating the geometric relationship whether the lidar beam ray $\overrightarrow{b_0 b_2}$ intersects with triangles $\triangle t_1 l_1 t_2$ and $\triangle l_1 l_2 t_2$. If the ray has an intersection that lies in one of the two triangles, the blade blockage status satisfies and vice versa. The details of the ray-triangle intersection algorithm can be found in [16].

3.2. Verification of implementation

In order to verify the code implementation, a lidar system is simulated that is mounted on the nacelle of the IEA 3.4MW reference wind turbine [5]. The lidar system is assumed to have one beam pointing towards the longitudinal direction (x axis). A constant rated rotor speed is assumed in the OpenFAST simulation. Figure 3 shows a polar scatter plot between the azimuth angle of the rotor and the blockage status. It is clear that the blockage happens at three ranges of azimuth angles, corresponding to the blockage by three blades.

4. Simulating the Lidar Measurement Availability

In this section, we perform statistical spectral analysis on the CNR measurement of a pulsed lidar system. Then the implementation of data availability in OpenFAST will be introduced.

4.1. Background

Previously, a measurement availability model has been proposed by [3], which simulates the availability over the measurement distance recorded for a certain period. The model provides a good reference to select measurement distance to give higher availability. In aero-elastic simulation, however, the time series of availability status is required. Thus, we chose to derive a novel spectral model for the CNR signal based on the lidar measurement. The model aims to produce CNR time series that has similar auto-correlation and availability statistics compared to the measurement data.

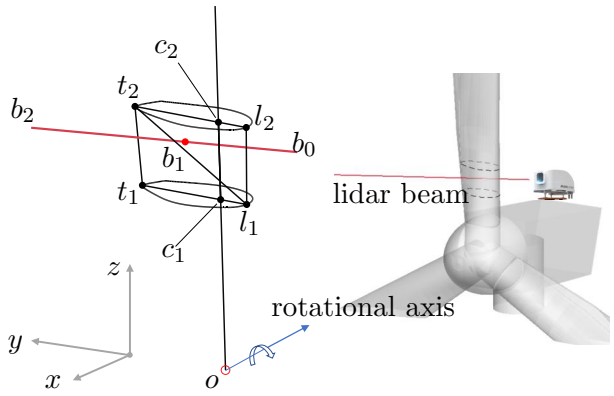

Figure 2. A geometrical sketch of the blade blockage detection algorithm. The lidar figure courtesy of Movelaser.

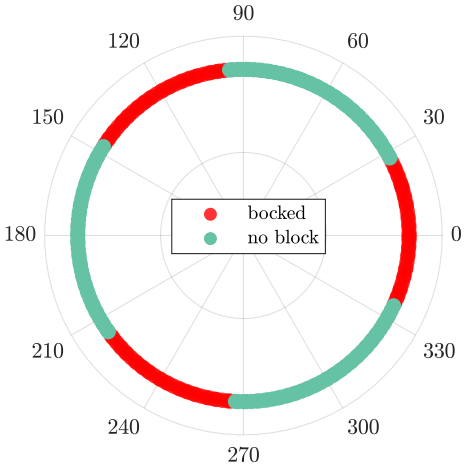

Figure 3. The polar scatter plot of the blade blockage status from OpenFAST simulation.

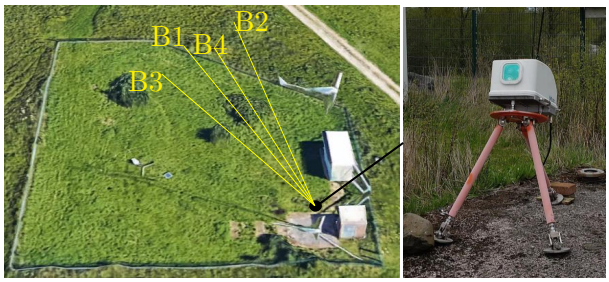

Figure 4. Overview of the lidar measurement site. B1: beam number 1.

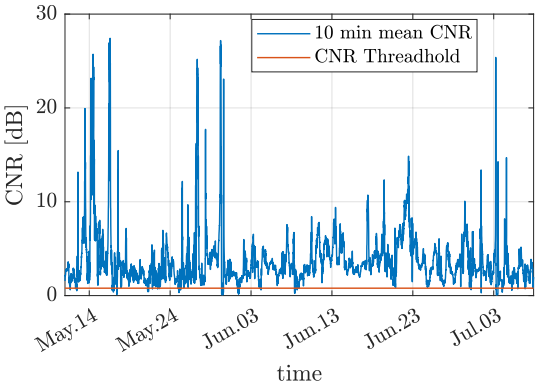

Figure 5. 10min mean CNR of beam 1.

Table 1. Scan configurations for the Molas lidar system.

number of beams	4	measurement distance	50-200 m
beam azimuth-angles ϕ [°]	15.0, 15.0, -15.0, -15.0	full scan time	1 s
beam elevation-angles β [°]	12.5, -12.5, -12.5, 12.5	pulse width at half maximum	30 m

The measurement was performed at the campus of Flensburg University of Applied Sciences in northern Germany, using the four beam pulsed lidar (Molas NL) produced by Movelaser, as shown by Figure 4. The data obtained from 10 May to 9 July 2021 was used for our analysis, during which the lidar was positioned towards the atmosphere and there were no obstacles in the LOS direction.

4.2. The spectral analysis of CNR

To observe the power spectral density of the CNR signals, we start by checking the stationarity of the CNR time series. The measurement data are divided into continuous 10 min samples, because 10 min is a typical time interval used for atmospheric turbulence for wind energy application [12]. As shown by Figure 5, the 10 min mean CNR shows some spikes on some days when the mean CNR values are quite high, so it can not be considered as stationary [2] in general. However, it can also been seen from Figure 5 that the samples with the mean CNR below the threshold are more stationary. Thus, only the CNR 10 min samples whose 10 min mean value is below 0.79 dB are chosen for the spectral estimation, and the spectrum is obtained by taking the sample mean spectrum from all the low CNR intervals. The rectangular windows (size of 512 data points) are applied to each 10 min CNR sample. Before applying the Fourier transform, the mean values are first removed from the signals and outliers (5 times larger than the mean CNR) are removed and replaced using zero padding. As mentioned previously, the lidar measurement is distinguished as unavailable when the CNR value is below a certain threshold (0.79 dB for the used lidar). The reason for analyzing the spectrum using low CNR samples is to emphasize on situations with low CNR, since situations with high CNR will not result in unavailable data.

The spectra estimation results are shown by Figure 6, where the dashed lines show the spectra estimated from the data from the four lidar beams (gate distance 190 m). Also, an empirical model is proposed to fit the spectrum based on the measurement. Here only the model for beam 1 (B1) is shown to have better visibility in the figure, because the models fitted from other beams are similar and will be mutually overlapped. The empirical model is expressed as

$$S_{\mathrm{CNR}}(f) = a(1 - 10^{-df})^{-1} + e \tag{3}$$

with three empirical parameters: a, d and e. The model is fitted to the estimated spectrum using a least square fit. [17] gives a detail description of the CNR sources, which implies that the higher frequency components of CNR are related to the instrument, while the lower frequency parts are related to the atmospheric conditions. As shown by Figure 6, the spectra is transitioning to the white noise spectrum (constant over frequencies) after 0.1 Hz, where the spectra are more composited by the noise by the physical characteristics of the optical and electrical instruments. The coherence between CNRs of the four beams are also compared, as shown by Figure 8. Except for beams 1 and 2 that measure at the same higher height above the ground, which have some correlation in the lower frequency range, the other pairs show weak correlations in the analyzed frequency range.

To verify the proposed spectral model, the CNR time series are generated by applying the Inverse Fast Fourier Transform to the spectrum [10]. After the fluctuation part of the time series is generated, the mean value is added and then the same CNR threshold by the lidar manufacturer is used to distinguish the availability. Since the spectral model has a high similarity with that of white noise, a white noise spectrum with a constant value of $0.023\,\mathrm{dB}^2/\mathrm{Hz}$ is also used to simulate the CNR time series for comparison. The scatter plot of the 10 min availability and the 10 min mean CNR is shown by Figure 7. It can be seen that the proposed spectral model results in very good agreement compared to the measurement data. Also, the CNR simulated using a white noise spectrum also shows nice agreement. There is only a small difference between the result by spectral model and by white noise spectrum.

4.3. Data availability implementation in OpenFAST

Considering the fact that the data availability can be site or lidar specific, it will be much more flexible to simulate the data availability externally (not within OpenFAST). So, the data availability module is implemented to read external files that contain the data availability time series. Apart from the time series, the sampling time, total time, and number of lidar measurement positions are required in the file. This information will be read into OpenFAST

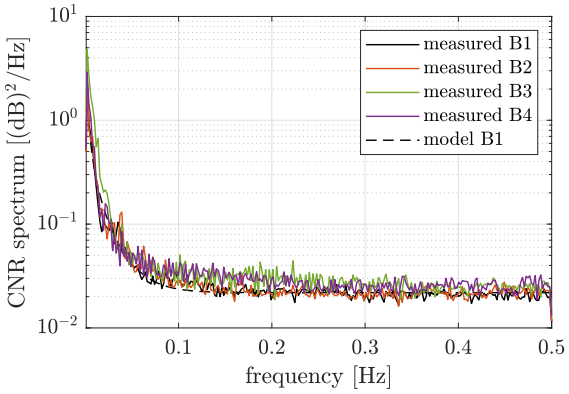

Figure 6. The CNR spectra estimated by measurements (gate 190m)and by models.

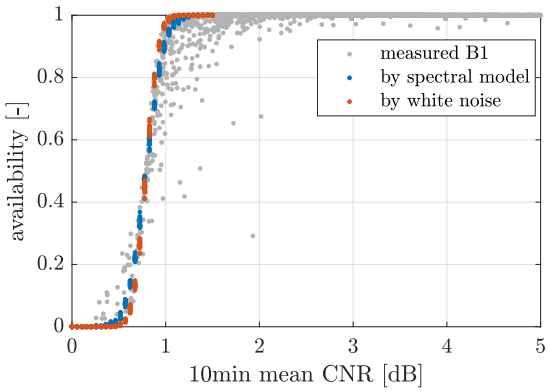

Figure 7. Comparison of the availability by spectral model and measurement.

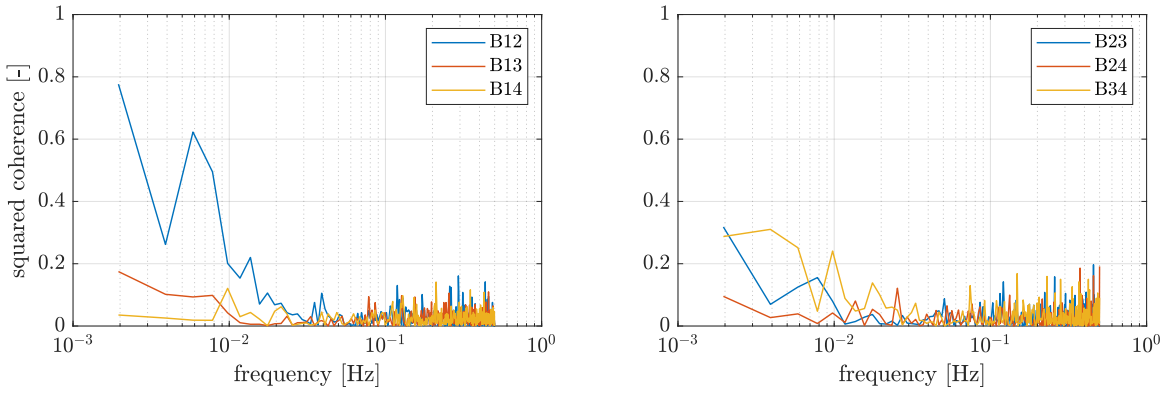

Figure 8. The CNR coherence of different beams (gate 190m) using low mean CNR samples.

and a check for the suitability with the lidar simulation configuration will be performed, e.g. it will be checked if the number of measurement positions agrees or if the time series have the correct time length.

At each simulated lidar measurement, the availability of the specific measured position will be checked. If the availability is true, the normal LOS will be simulated. Otherwise, the LOS speed will be returned as a error value of -99, which is a distinguishable number for the lidar data processing and it does not require additional allocation of variables in OpenFAST.

5. Simulation with the Updated Lidar Simulator

In this section, we will first give a short review on the lidar measurement coherence for wind turbine control. Then different simulation setups will be discussed: (1) consider wind evolution only; (2) consider both wind evolution and blade blockage; and (3) consider wind evolution; blade blockage and data availability. Lastly, the results of the measurement coherence will be presented.

5.1. Lidar measurement coherence/wind preview quality

We use the typical definition of rotor effective wind speed (REWS) for control purposes, which is the mean longitudinal wind component over the rotor swept area [18, 11, 21, 9]. The REWS

at rotor position u_{R} and the one estimated by lidar u_{L} can be calculated by [9]:

$$u_{\mathrm{R}}(t) = \frac{1}{n}\sum_{i=1}^{n} u_i(t) \text{ and } u_{\mathrm{L}}(t) = \frac{1}{x_{\mathrm{n}}}\frac{1}{m}\sum_{j=1}^{m} v_{\mathrm{los},j}(t), \tag{4}$$

with u_i is the ith u component in the discrete rotor swept area and $v_{\mathrm{los},j}$ is the jth LOS measurement, n is the total number of grid points in the rotor swept area and m is the total number of lidar measurements. The constant x_{n} is the first element of the unit vector between the longitudinal direction and the lidar beam direction, which can be simply calculated knowing the beam geometry (see e.g. [18], [19] for more details).

The coherence (magnitude squared [2]) between u_{R} and u_{L} defines the correlation between two signals in the frequency-domain, and it is ≤ 1 in all frequency bins. For a certain frequency, a higher coherence means the two signals are more correlated at this frequency. An over all higher coherence means the lidar wind preview quality is better since measurement is more correlated with the actual disturbance. For a certain turbulence model, the measurement coherence can be derived analytically without considering the disturbances such as nacelle motion, blade blockage, and data availability. Details can be found from [18, 11].

5.2. Lidar data processing algorithm

In Equation (4), the lidar estimated REWS is averaged by the LOS of different positions that are measured at the same time. However, for the pulsed lidar we considered in simulation, the measurement is performed from one position to another position, meaning they are not sampled at the same time. Also, due to the blade blockage and the availability related to CNR values, the measurement is not always available for one position. Thus, we used a data buffer based processing algorithm.

The data buffer has a size equal to the number of measurement positions and is initialized to be "not available" for all measurement positions. At a certain time, if successful new measurement data is available for one position, the new data will replace the old data. Otherwise, when the data is not available due to blade blockage or low CNR, the old data will be kept for that position. The maximum time to keep a data is designed to be 2 s for now, however further investigation on the memory time should be investigated. If the memory time exceeds the designed maximum, the measurement status at the corresponding position will be set as not available. For each time, the processed u_{L} will be calculated only using the available positions in the data buffer.

5.3. Simulation Setup

In all the simulations, the lidar is assumed to have the trajectory shown in Table 1 and be installed on the nacelle of the IEA 3.4 MW reference turbine [5]. The single measurement distance at 120 m is considered.

The IEC Kaimal turbulence model [12] is applied in this work (Class A). The turbulence has a reference wind speed of 16 m/s and a reference turbulence intensity of 16%. In the case of considering wind evolution, the 4D turbulence field is generated using *evoTurb*. The longitudinal coherence model applied is based on [22] with the coherence decay parameters $a = 2.5$ and $b = 0$, corresponding to a medium evolution scenario [7]. Three different mean CNR scenarios: 0.5 dB, 0.7 dB, and 0.9 dB are defined to study the impact of availability on lidar measurement coherence. For each lidar measured position, the CNR time series is generated by both the spectral model and the white noise spectrum (see Section 4.2 for the method), then the threshold of 0.79 dB is applied to obtain the availability time series. As discussed in Section 4.2, the CNR values of different lidar measurement positions show weak coherence. Thus, we assume the simulated

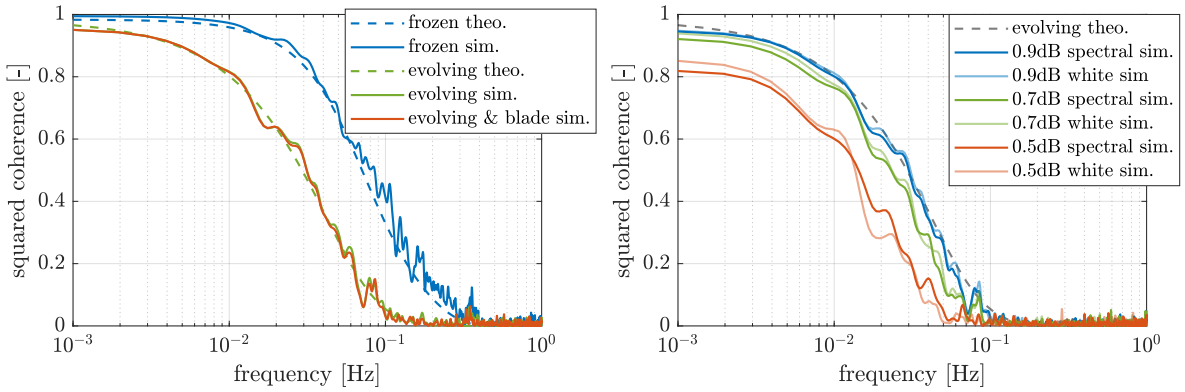

Figure 9. The coherence between u_{R} and u_{L} under different simulation setups. "theo." denotes the theorectical coherence and "sim." denotes the ensemble mean from simulations. "white": white noise spectrum, "spectral": the fitted spectral model. Left: impact of evolving turbulence and blade blockage. Right: impact of different data availabilities on top of evolving turbulence and blade blockage.

CNR time series of different positions are stochastically independent. Also, the CNR time series is assumed to be independent of turbulence.

For each setup, 12 simulations with different random seed numbers for turbulence and CNR time series are performed.

5.4. Result and Discussion

The coherence comparison is shown by Figure 9. In the left plot the good agreement between the simulation and the theoretical curves for frozen and evolution cases verifies the code implementation of the wind evolution module. Also it shows, the coherence is overestimated when the frozen theory is made compared with the wind evolution case. Besides, the high similarity between the blue and orange dashed lines demonstrates the negligible impact of the blade blockage on the coherence. The right plot shows that the coherence decrease when there is low data availability (represented by the low mean CNR value). However, the coherence still maintains an acceptable level for control application even when the mean CNR is below 0.5 dB. This might be caused by the fact that all measurement positions seldom become unavailable over a long time. Comparing the spectral based CNR and white noise based CNR curves, the tiny difference also implies the white noise spectrum can be used to simulate data availability time series.

6. Conclusions and Further Work

In this work, three new features, namely the evolving turbulence, the blade blockage effect, and the adjustable data availability are added to the existing OpenFAST lidar simulator to make it more realistic. These new features are supportive for performing simulations when these phenomena are of especial interest, for example for the certification of lidar-assisted wind turbine control.

The wind evolution module is updated in OpenFAST, which can be easily interacted with an existing open-source 4D turbulence generation tool *evoTurb*. As for the blade blockage effect, an easy and robust algorithm relying on ray-triangle intersection detection is deployed. A data availability module has been developed in OpenFAST to read in availability time series. Statistical spectral analysis for lidar CNR values has been carried out using real lidar measurements. Based on our investigation, it can be concluded that the low data availability

event can be simulated with a white noise spectrum. Analysis regarding the coherence between rotor effective wind speed and the lidar estimated one has been carried out using the updated lidar simulator. Based on the simulation, the blade blockage has negligible impact on the coherence, while both the wind evolution and the low availability can reduce the coherence.

In the future, the coherence between CNR signals from different measurement positions will be further studied. The filtering of lidar estimated rotor effective wind speed for turbine control will be further investigated under different wind evolution conditions and various low CNR intervals.

Data Availability

The recent version of OpenFAST with the new lidar simulator can be accessed by `https://github.com/fengguoFUAS/OpenFAST3.0_LidSim`.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 858358 (LIKE – Lidar Knowledge Europe).

References

[1] Matthew L Aitken, Michael E Rhodes, and Julie K Lundquist. Performance of a wind-profiling lidar in the region of wind turbine rotor disks. *Journal of Atmospheric and Oceanic Technology*, 29(3):347–355, 2012.

[2] Julius S. Bendat and Allan G. Piersol. *Random data; analysis and measurement procedures*. John Wiley & Sons, New York, USA, 1971.

[3] Matthieu Boquet, Philippe Royer, Jean-Pierre Cariou, Mehdi Machta, and Matthieu Valla. Simulation of doppler lidar measurement range and data availability. *Journal of Atmospheric and Oceanic Technology*, 33(5):977–987, 2016.

[4] Antoine Borraccino, David Schlipf, Florian Haizmann, and Rozenn Wagner. Wind field reconstruction from nacelle-mounted lidar short-range measurements. *Wind Energy Science*, 2(1):269–283, 2017.

[5] Pietro Bortolotti, Helena Canet Tarres, Katherine Dykes, Karl Merz, Latha Sethuraman, David Verelst, and Frederik Zahle. IEA wind task 37 on systems engineering in wind energy – WP2.1 reference wind turbines. Technical report, International Energy Agency, 2019.

[6] Y. Chen, F. Guo, D. Schlipf, and P. W. Cheng. 4d wind field generation for the aeroelastic simulation of wind turbines with lidars. *Wind Energy Science Discussions*, 2021:1–31, 2021.

[7] Yiyin Chen, David Schlipf, and Po Wen Cheng. Parameterization of wind evolution using lidar. *Wind Energy Science*, 6(1):61–91, 2021.

[8] Samuel Davoust, Alban Jehu, Michael Bouillet, Mathieu Bardon, Benoist Vercherin, Andrew Scholbrock, Paul Fleming, and Alan Wright. Assessment and optimization of lidar measurement availability for wind turbine control. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2014.

[9] Feng Guo and David Schlipf. Lidar wind preview quality estimation for wind turbine control. In *2021 American Control Conference (ACC)*, pages 552–557, 2021.

[10] Feng Guo and David Schlipf. A spectral model of grid frequency for assessing the impact of inertia response on wind turbine dynamics. *Energies*, 14(9), 2021.

[11] Dominique Held and Jakob Mann. Comparison of methods to derive radial wind speed from a continuous-wave coherent lidar doppler spectrum. *Atmospheric Measurement Techniques*, 11:6339–6350, 11 2018.

[12] IEC 61400-1. Wind turbines - Part 1: Design requirements, 2005.

[13] J. Jonkman and M. L. Buhl. FAST user's guide. Technical Report EL-500-38230, NREL, 2005.

[14] Jason Laks, Eric Simley, and Lucy Pao. A spectral model for evaluating the effect of wind evolution on wind turbine preview control. In *2013 American Control Conference*, pages 3673–3679. IEEE, 2013.

[15] Jakob Mann. The spatial structure of neutral atmospheric surface-layer turbulence. *Journal of fluid mechanics*, 273:141–168, 1994.

[16] Tomas Möller and Ben Trumbore. Fast, minimum storage ray-triangle intersection. *Journal of graphics tools*, 2(1):21–28, 1997.

[17] Alfredo Peña, C Bay Hasager, and Julia Lange. Remote sensing for wind energy. 2013.

[18] David Schlipf. *Lidar-Assisted Control Concepts for Wind Turbines.* Dissertation, University of Stuttgart, 2015.
[19] David Schlipf, Feng Guo, and Steffen Raach. Lidar-based estimation of turbulence intensity for controller scheduling. *Journal of Physics: Conference Series*, 1618:032053, sep 2020.
[20] David Schlipf, Steffen Raach, and Florian Haizmann. Control system for controlling a turbine, method for controlling a turbine, and wind turbine, Patent EP3717770B1, 2018.
[21] Eric Simley, Holger Fürst, Florian Haizmann, and David Schlipf. Optimizing lidars for wind turbine control applications – results from the iea wind task 32 workshop. *Remote Sensing*, 10(863), 2018.
[22] Eric Simley and Lucy Pao. A longitudinal spatial coherence model for wind evolution based on large-eddy simulation. In *2015 American Control Conference (ACC)*, pages 3708–3714. IEEE, 2015.
[23] Geoffrey Ingram Taylor. The spectrum of turbulence. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, 164(919):476–490, 1938.
[24] Paul S Veers. Three-dimensional wind simulation. Technical report, Sandia National Labs., Albuquerque, NM (USA), 1988.