Review of motion-correction method for floating lidar measurements

Department of
Wind Energy
I-Report

Alfredo Peña

DTU Wind Energy Report-I-1635

January 2024

DTU Wind Energy
Department of Wind Energy

Document Information

Project	Co-financed research between Energinet and DTU Wind and Energy Systems
Document	Review of motion-correction method for floating lidar measurements
Type	Internal Report
Number	DTU Wind Energy Report-I-1635
Year	2024
Contractor	Technical University of Denmark (DTU)
Authors	Alfredo Peña
Contact	Mail: aldi@dtu.dk

Published by:
DTU, Department of Wind and Energy Systems
Frederiksborgvej 399, Building 118, 4000 Roskilde Denmark
www.vindenergi.dtu.dk

Preface

As part of the cooperation agreement on co-financed research between Energinet Eltransmission A/S (Energinet) and DTU Wind and Energy Systems (DTU Wind), whose objective is the study of correction methods to lidar turbulence measurements so that they agree with industry-accepted measurements of turbulence, DTU Wind first task is the review of the motion-correction method for floating lidar measurements that Fugro has developed to correct buoy lidar turbulence measurements. Fugro used this method to correct lidar turbulence measurements performed at several positions in the North and Baltic Seas for Energinet as part of the Danish government plans with regards to the Danish energy islands.

Roskilde, 22-01-2024

Alfredo Peña, Senior Scientist, DTU Wind

Reviewed by:

Jakob Mann, Professor and Head of Section, DTU Wind

Contents

1 Introduction 5

2 Framework to analyze the effect of motion on buoy lidar turbulence 6

2.1 Impact of motion 8

3 Conclusion 11

References 12

Appendix A 13

Appendix B 14

1 Introduction

This review focuses on the method employed and developed by Fugro to correct for the effect of the buoy motion on a particular floating lidar system based on the ZX 300M lidar, which is a continuous wave (CW) lidar instrument. Fugro describes the method in the reports "Motion correction of turbulence intensity. WP1: North Sea pre-deployment verification tests" and "Motion correction of turbulence intensity. WP3: Baltic Sea pre-deployment verification tests". In both reports, which are hereafter referred to as *motion-correction reports*, the method is exactly the same. The motion correction of data is based on post-processing of the motion signals and does not account for surge and sway as these motions were not recorded. Appendix A provides the description of the method from the *motion-correction reports*.

The main assumption in Fugro's method is that the standard deviation of the horizontal velocity fluctuations, which are measured by the buoy lidar, i.e., the result of the motion-impacted fluctuations combined with the fluctuations caused by atmospheric turbulence, is the linear sum of those, i.e., the sum of the standard deviation of the motion-induced fluctuations and that of the turbulence fluctuations. This assumption is fair for the case in which these two sources of velocity fluctuations, from turbulence and motion, are uncorrelated. How fair is this assumption? This is a difficult question to answer. As en example, Peña et al. [2022] showed that the buoy lidar will measure more horizontal velocity variance than a fixed lidar measuring at the same height when accounting for the lidar probe volume (recall that for a CW lidar the probe volume increases quadratically with measurement range). This difference depends on both the turbulence characteristics and the motion characteristics. As far as the *motion-correction reports* describe, the probe volume is not accounted for in Fugro's method. Further, radial velocities measured within the lidar scanning pattern will be 'contaminated' differently from the velocity components once motion takes place compared to the degree of contamination they will experience for a fixed lidar.

In addition, as far as the *motion-correction reports* describe, Fugro's method heavily relies on the synchronisation of the measured signals from both the lidar and the motion data. This is perhaps the core of the data post-processing in Fugro's method as it determines the impact of a particular motion on the lidar measured output. We cannot review how good/accurate the procedure developed by Fugro for this particular dataset is, since we do not have the code/algorithm they used for synchronisation/timing of the signals but in Sect. 2.1, we show how important synchronisation can be for motion correction.

As mentioned, surge and sway were not recorded and so Fugro's method does not account for the effect of these motions. Our understanding (from an online meeting where Fugro, Energinet and DTU Wind participated) is that Fugro has found that these two motions do not have a strong impact on the motion-induced fluctuations, since they had deployments where these signals were indeed measured in the past; Fugro's method did not result in significant differences when they accounted for these motions compared to the method used for correcting turbulence estimates from the lidars used by Energinet. Here it is important to note that our previous work [Peña et al., 2022] showed, e.g., that surge can significantly increase the fluctuations measured by a buoy lidar: for surge velocities of 0.45 m s^{-1} with a sinusoidal signal with periodicity of 4 s, the buoy lidar measured up to 10% more along-wind variance than a fixed lidar. In Sect. 2.1, we show more insights with regards to the impact of surge and sway on buoy lidar turbulence.

Overall, in our opinion correcting buoy lidar turbulence due to the buoy's motion is necessary if these measurements are used to characterise turbulence and determine turbulence-related site conditions. One should not characterise turbulence using buoy lidar turbulence measurements that are not motion compensated/corrected. Also, any type of correction will provide a better estimate of turbulence by the lidars (even without accounting for some degrees of motion), if and only if the motion and the scanning signals of the lidar and the buoy are timely and adequately synchronised. Therefore, we believe that if the correction performed by Fugro has the latter characteristics with regards to signal processing/synchronization, their motion-corrected turbulence estimates will be closer to turbulence estimates from a fixed lidar compared to those that are not motion-corrected. However, the uncertainty of the correction cannot be assessed from the information we have from Fugro.

2 Framework to analyze the effect of motion on buoy lidar turbulence

In Appendix B, we provide a Matlab code [The MathWorks Inc., 2022] with the numerical framework to analyze the effect of buoy motion on lidar turbulence estimates. It is similar to that described in Peña et al. [2022] and the idea is to virtually 'immerse' a lidar within an atmospheric turbulence box. An atmospheric turbulence box is a synthetic realisation of simulated atmospheric turbulence with specific turbulence characteristics. One can use different turbulence models to generate such synthetic turbulence fields. In our case, we created for this short analysis a single turbulence box using the Mann spectral turbulence model [Mann, 1994]. The Mann model contains three parameters besides the wavenumber vector $\boldsymbol{k}$; the dissipation rate of turbulence $\alpha\epsilon^{2/3}$, the turbulence length scale L, and the turbulence anisotropy Γ.

The turbulence box has a size of $(L_x, L_y, L_z) = (18000, 128, 128)$ m with a number of grid points $(n_x, n_y, n_z) = (8192, 64, 64)$, where x, y, and z are the along-wind, transverse, and vertical axes, respectively. The Mann turbulence parameters used to generate the box are $\alpha\epsilon^{2/3} = 0.05$ m$^{4/3}$ s^{-2}, $L = 61$ m, and $\Gamma = 3.2$. With these turbulence parameters, the target velocity covariances are $[\langle u'u'\rangle, \langle v'v'\rangle, \langle w'w'\rangle, \langle u'w'\rangle] = [1.342, 0.796, 0.495, -0.359]$ m^2 s^{-2}.

For a fixed lidar conically scanning the atmosphere such as the ZX 300M lidar, the theoretical expressions for the velocity spectra are slightly complex. Following Sathe et al. [2011], the w-velocity spectrum is given by

$$F_w^l(k_1) = \left(\frac{1}{\cos^2\phi}\right)\widehat{T_f}(k_1)\iint \Phi_{ij}(\boldsymbol{k})\alpha_i(\boldsymbol{k})\alpha_j^*(\boldsymbol{k})dk_2dk_3, \tag{1}$$

where Φ_{ij} is the spectral velocity tensor (this is what the Mann model represents), α is a spectral weighting function (* means complex conjugation), ϕ the lidar half-opening angle (30° for the ZX 300M lidar), and $\widehat{T_f}$ a spectral transfer function accounting for the low-pass filter effect due to the time the lidar takes to scan the cone t_s. This effect can be modelled by a rectangular filter with a length scale $L_f = Ut_s$ as

$$\widehat{T_f}(k_1) = \text{sinc}^2\left(\frac{k_1 L_f}{2}\right). \tag{2}$$

For the u and v velocities, the spectrum is similar to that in Eqn. (1) but the term $\cos^2\phi$ in the denominator should be replaced by $\sin^2\phi$ and the weighting function α by β and γ, respectively. For a CW lidar, these weighting functions are

$$\alpha_i(\boldsymbol{k}) = \frac{1}{2\pi}\int_0^{2\pi} n_i(\theta)\exp\left(1id_f\boldsymbol{k}\cdot\boldsymbol{n}(\theta)\right)\exp\left(-z_R|\boldsymbol{k}\cdot\boldsymbol{n}(\theta)|\right)d\theta, \tag{3}$$

$$\beta_i(\boldsymbol{k}) = \frac{1}{\pi}\int_0^{2\pi} \cos\theta n_i(\theta)\exp\left(1id_f\boldsymbol{k}\cdot\boldsymbol{n}(\theta)\right)\exp\left(-z_R|\boldsymbol{k}\cdot\boldsymbol{n}(\theta)|\right)d\theta, \tag{4}$$

$$\gamma_i(\boldsymbol{k}) = \frac{1}{\pi}\int_0^{2\pi} \sin\theta n_i(\theta)\exp\left(1id_f\boldsymbol{k}\cdot\boldsymbol{n}(\theta)\right)\exp\left(-z_R|\boldsymbol{k}\cdot\boldsymbol{n}(\theta)|\right)d\theta, \tag{5}$$

where $\boldsymbol{n}(\phi,\theta) = (\cos\theta\sin\phi, \sin\theta\sin\phi, \cos\phi)$ is the unit vector describing the lidar scanning pattern with θ being the azimuthal positions (50 in the case of the ZX lidar), d_f the focused distance, and z_R is the Rayleigh length [Sonnenschein and Horrigan, 1971] that characterizes the length of the probe volume and is estimated as

$$z_R = \frac{\lambda d_f^2}{\pi a_0^2}, \tag{6}$$

where λ is the laser wavelength and a_0 the effective beam radius at the output lens.

As mentioned, the theoretical expressions above are not simple to compute and are valid for the fixed CW lidar only. However, we can demonstrate that our framework can be used to analyze the effect of buoy motion on floating lidars by immersing a fixed lidar within the turbulence box, scanning the velocity fluctuations of the box with the lidar, and comparing the spectra/velocity variances with the theoretical expressions. This is shown in Fig. 1 where we illustrate both the velocity spectra computed from the theoretical expressions and those computed by scanning the turbulence box along the x-axis at 51 heights of the turbulence box with the lidar. For simplicity, we have assume that $z_R = 0$ m, i.e., no probe volume (the framework in Appendix B does not include probe volume effects) and that we scan at all positions along the x-axis of the turbulence box (i.e., 8192). The lidar scans at 62 m; therefore, the lidar is 'out of the box' nearly half of the scanned heights, although the scanning pattern is well inside. We have also computed, both theoretically and with the turbulence box framework, the velocity spectra of an ideal anemometer (sonic) in the middle of the circular scan of the fixed lidar.

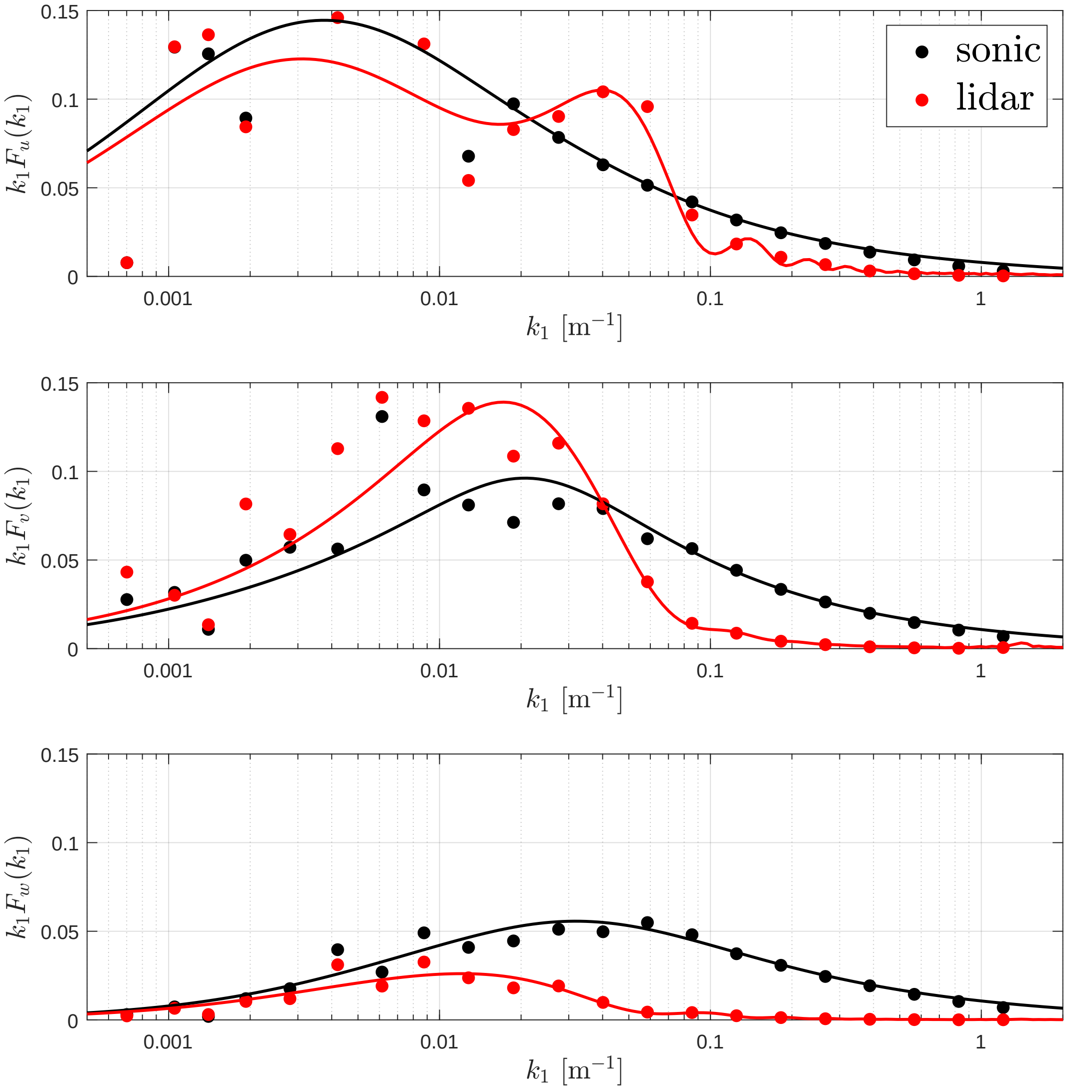

Figure 1: Premultipled velocity spectra (u-top, v-middle, and w-bottom) for and ideal sonic anemometer and a ZX 300 lidar measuring at a height of 62 m with $\alpha\epsilon^{2/3} = 0.05$ m$^{4/3}$ s^{-2}, $L = 61$ m, and $\Gamma = 3.2$. Markers show the spectra derived from the turbulence box framework and solid lines show the spectra from the theoretical computations

A good match of the velocity spectra, as shown in Fig. 1, provides us confidence in the analysis of the turbulence measured by lidars using the turbulence boxes' approach. Recall that the variance of the velocity components is the area below the velocity spectra in Fig. 1.

2.1 Impact of motion

To simulate the floating ZX 300M lidars used by Energinet more closely using our framework, we keep the measuring height at 62 m (due to the limits of the turbulence box) but scan every 17 s, which is about the time used by the Fugro lidar system. Our turbulence box is made for studying a velocity field advected within a 30-min period. Therefore, the lidar is set to scan the box at 106 positions along the x-axis (three times those that the lidar will perform within a 10-min period scanning every 17 s). To increase the significance of the computation, we measure at different heights within the turbulence box but with the same focus distance so we virtually move the lidar vertically once all positions along the x-axis are done, similarly as in Sect. 2. This results in 51 vertical levels, i.e., 51 velocity variances. The statistics therefore are the ensemble average (from 51 estimates) of the ratio between the horizontal velocity variance measured by the floating lidar to that measured by the lidar without motion. For simplicity, we do not consider the effect of the probe volume in the turbulence box.

We first evaluate the impact of motion by assuming the pitch is given as

$$p = A_p \sin\left(2\pi/T_p + \varphi_p\right), \tag{7}$$

where A_p is the amplitude, T_p the period, and φ_p the phase of the pitch motion. To have an idea of the values that the Fugro floating lidars experience, we analyze both pitch and roll motions measured by the WS180 lidar (`motion_WS180atLEG`). Figure 2 illustrates histograms of the amplitudes and periods experienced by that particular lidar: pitch and roll have mean amplitudes of 4.96° and 4.65°, while their mean periods are 2.71 and 2.79 s, respectively.

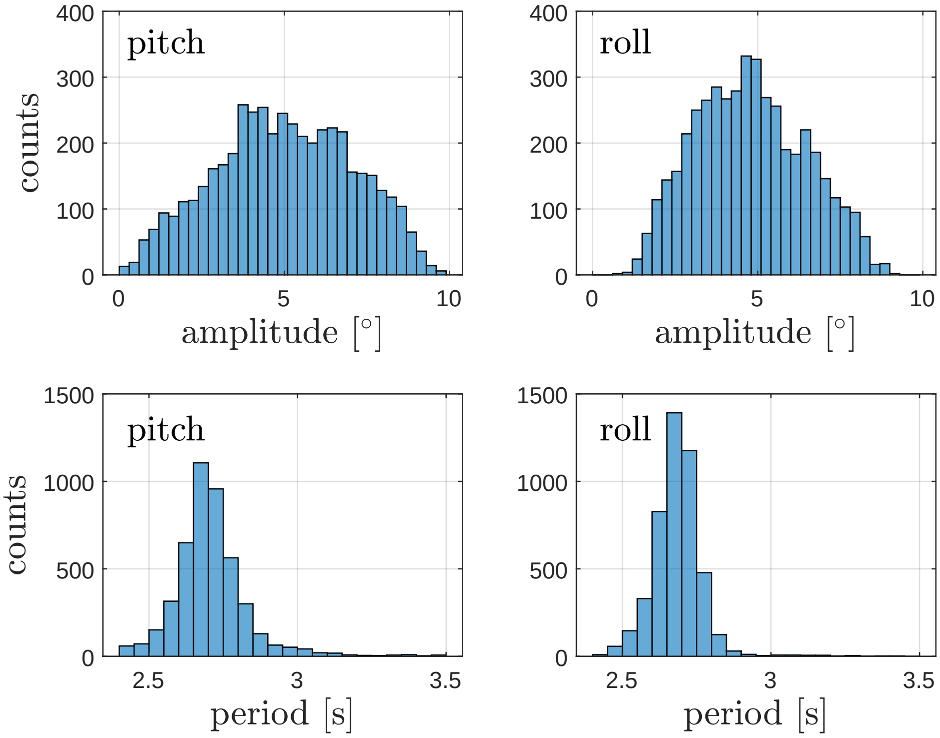

Figure 2: Amplitudes (top) and periods (bottom) of the pitch (left) and roll (right) motions for the WS180 lidar

We start our analysis by fixing a pitch motion period at $T_p = 3$ s, assuming the lidar scans every 17 s, and no phase $\varphi_p = 0°$. Figure 3 illustrates the results for a number of pitch amplitudes covering the range of measured mean amplitudes in Fig. 2. For a pitch amplitude of 5°, which is close to the mean of the observations, this ideal pitch motion increases the horizontal variance of the floating ZX 300 lidar by 5% compared to that of a fixed ZX 300 lidar. For the largest pitch amplitude simulated (10°), the floating lidar variance is more than 20% the value of the fixed lidar. Although we do not know the levels of amplitude of translation motions such as surge or sway, with the framework we can also explore the effect of surge. With a surge motion with a form similar to that in Eqn. 7, lidar scanning every 17 s, and surge periods of 3 s with zero phase, we also show the impact of a range of surge amplitudes on the variance ratio in Fig. 3. As illustrated, similar ratios are found compared to those for pitch motion for a range of surge amplitudes between 0 and 0.8 m s^{-1}.

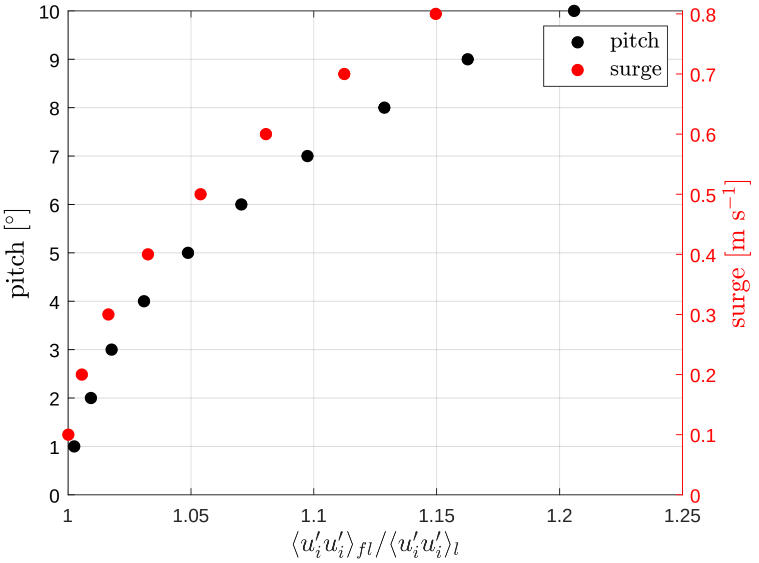

Figure 3: Dependency of the ratio of the horizontal velocity variance of the floating lidar to that of the fixed lidar on both pitch and surge amplitudes. See text for details

The results in Fig. 3 assume that the pitch/surge motion, with a simple sinusoidal behaviour, are timed with the lidar scanning. To show the impact of unsynchronized lidar/motion signals on the horizontal velocity variance measured by a buoy lidar, we can first add phases to the motion signal and compute the ratio of the horizontal velocity variance as we did for the study of the pitch/surge amplitude. Figure 4 shows the results of adding a phase to a pitch motion with the form in Eqn. (7) ($A_p = 5°$ and $T_p = 3$ s) within the range 0–180°. The figure shows two sets of results: one assumes the lidar scans every 17 s and the other that the scan is made every 18 s. For the 17-s case and within the range of phases simulated, the buoy lidar measures ≈3–5% more variance than the fixed lidar. The behavior of the variance ratio with increasing phase changes with no clear pattern. For the 18-s case, we can get variance ratios much closer to one and the pattern with phase appears symmetric since the time of the lidar scans (18 s) is multiple of the motion period (3 s).

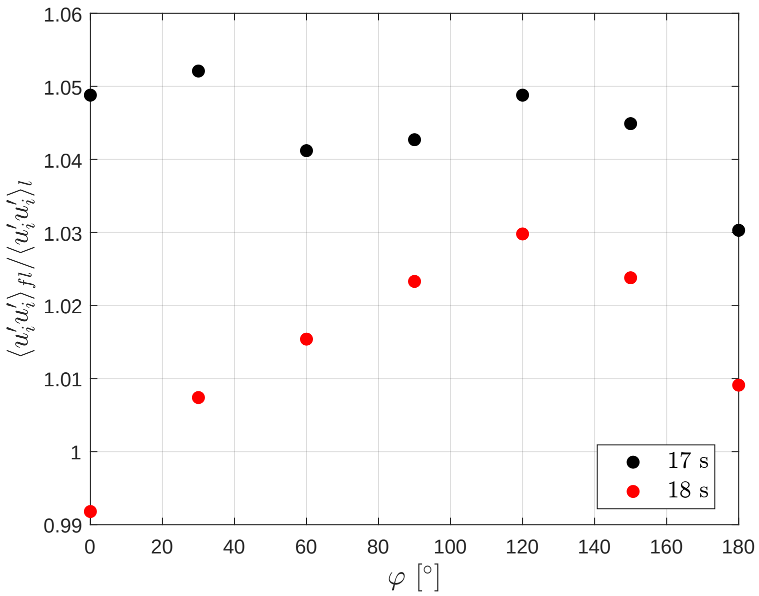

Figure 4: Dependency of the ratio of the horizontal velocity variance of the floating lidar to that of the fixed lidar on both phase of pitch motion and lidar scan frequency. See text for details

3 Conclusion

Correction of buoy lidar turbulence due to the buoy's motion is necessary if these measurements are used to characterise turbulence. As discussed, post-processing of the floating lidar turbulence measurements using the motion signals will provide a better estimate of turbulence even without accounting for all degrees of motion, if the signals of motion and lidar retrievals are timely and adequately synchronised.

Note that a preliminary analysis of Fugro's motion-corrected turbulence measurements, when compared to the reference turbulence measurements from a fixed lidar unit (`WS170atLEG_data.csv`), revealed that Fugro's algorithm overcorrects turbulence measurements at low turbulence values. This results in non-physical negative turbulence intensity values at low turbulence levels. Figure 5 shows this problem for one of Fugro's unit. The uncorrected (unc) turbulence intensity measures are, as expected, always positive and higher than the motion-compensated (com) values. The uncorrected (unc) turbulence intensities are overall higher than the fixed lidar (ref) turbulence intensity measures, whereas the motion-compensated (com) values appear less positively biased compared to the fixed lidar (ref) turbulence intensities.

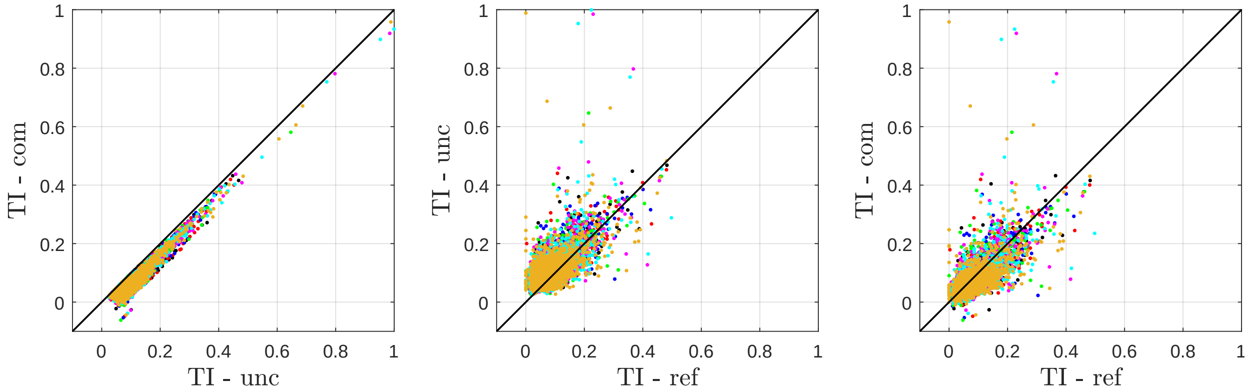

Figure 5: Scatter plots of the turbulence intensity (TI) measures from the motion-compensated data (com), the uncorrected TI values (unc), and the reference TI values (ref) from a fixed lidar. Markers show the estimated TI values from the measurements (different colors indicate different measurement heights) and the solid lines show a 1:1 line. See other details in the text

References

J. Mann. The spatial structure of neutral atmospheric surface-layer turbulence. *J. Fluid Mech.*, 273:141–168, 1994.

A. Peña, J. Mann, N. Angelou, and A. Jacobsen. A motion-correction method for turbulence estimates from floating lidars. *Remote Sens.*, 14:6065, 2022.

A. Sathe, J. Mann, J. Gottschall, and M. S. Courtney. Can wind lidars measure turbulence? *J. Atmos. Ocean. Technol.*, 28:853–868, 2011.

C. M. Sonnenschein and F. A. Horrigan. Signal-to-noise relationships for coaxial systems that heterodyne backscatter from the atmosphere. *Appl. Opt.*, 10:1600–1604, 1971.

The MathWorks Inc. Matlab version: 9.13.0.2080170 (r2022b) update 1, 2022. URL https://www.mathworks.com.

Appendix A:

The motion-correction of Fugro uses data processing and consists of the following steps, which are directly and literally extracted from the *motion-correction reports*:

1. Assign accurate timestamps to Wavesense motion data by synchronization with DGPS yaw and pitch data
2. Find correct timing of first beam of each velocity-azimuth display (VAD) scan
3. Calculate 10-minute mean wind speed and direction including 180 deg ambiguity correction
4. Generate synthetic line-of-sight data from known beam geometry, timing and mean wind data, without turbulence
5. Perturbate synthetic line-of-sight data using motion data for a series of possible temporal offsets between LiDAR and motion timing
6. Reconstruct three-dimensional wind vectors from the synthetic motion-perturbed line-of-sight data
7. Subtract wind speed fluctuations of the synthetic wind vector data from the lidar1hz data and find temporal offset with the lowest resulting wind speed variance
8. Calculate the standard deviation of horizontal wind speed fluctuations by subtracting the motion-perturbed values from the measured values, assuming that they are statistically uncorrelated
9. Average TI reduction over all measurement elevations and one hour to smoothen the motion-compensated TI results

Appendix B: Matlab code to simulate the horizontal velocity variance of the buoy and a fixed ZX 300 lidar on a turbulence box

Contact the report author in case turbulence boxes are needed to run the code.

```
clc
clear

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%--- VAD ZX lidar configuration
res=360/50;                 %%%--- resolution of the VAD in deg
phi=30;                     %%%--- half-opening angle in deg
theta=(0:res:360-res)';     %%%--- azimuthal angles in deg
el=(90-phi);                %%%--- elevation angle in deg
%%%--- dimensions of the turbulence box
Nx=8192;
Ny=64;
Nz=64;
Lx=18000;
Ly=128;
Lz=128;
%%%--- physical properties
z0=0.0002;  %%%--- roughness lenght in m
kappa=0.4;  %%%--- von Karman constant
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%--- domain setup (x is the along wind, v across and z vertical directions)
dx=(Lx/Nx);
x=0:dx:Lx-dx;
dy=Ly/Ny;
dz=Lz/Nz;
y=-Ly/2+dy:dy:Ly/2;
z=0:dz:Lz-dz;

%%%--- parameter to vary: heights to explore in the turbulence box
%%%--- the 10 deg is the maximum rotation accounted for
h_exp=62; %%%--- height in m above the lidar unit for exploration
h_m=(-h_exp+h_exp*sind(10)/cosd(30):dz:z(end)-(h_exp+h_exp*sind(10)/cosd(30)));
%%%--- lidar coordinates
y_lo=0;

%%%--- this is the time the lidar revisits the same height
dt_l=17;%1800/8192;  %% 17 s for the ZX lidars used by Fugro
%%%--- the "velocity" of the turbulence box assuming is made for a field
%%%--- advected within a 30-min period
vel_l=(x(end)-x(1))/1800;
dx_l=vel_l*dt_l;  %%%--- separation of the lidar along the turbulence box

%%%--- vector of the positions in x-axis of the lidar within the box
x_l=x(1)+h_exp*sind(10)+tand(30)*h_exp:dx_l:x(end)-(h_exp*sind(10)+tand(30)*h_exp);
%%%--- sonic coordinates
y_m=y_lo;
x_m=x_l;

%%%--- 3D mesh grid
[Z,Y,X]=ndgrid(z,y,x);

%%%--- unit vectors of the VAD
n_z=[cosd(el)*cosd(theta) sind(theta)*cosd(el) sind(el)*ones(length(theta),1)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%--- reading turbulence boxes

filename='test1.u';
fileID=fopen(filename,'r');
sizeu=Ny*Nz*Nx;
```

```
u=fread(fileID,sizeu,'float32');
fclose(fileID);
u=reshape(u,Nz,Ny,Nx);
filename='test1.v';
fileID=fopen(filename,'r');
sizeu=Ny*Nz*Nx;
v=fread(fileID,sizeu,'float32');
fclose(fileID);
v=reshape(v,Ny,Nz,Nx);
filename='test1.w';
fileID=fopen(filename,'r');
sizeu=Ny*Nz*Nx;
w=fread(fileID,sizeu,'float32');
fclose(fileID);
w=reshape(w,Ny,Nz,Nx);

%%%--- assuming some shear by using the roughness and mean wind speed of the box
ustar=kappa*vel_l/log(z(end)./z0);
Uz=(ustar/kappa)*log(z/z0);
for i=1:Nz
    u(i,:,:)=u(i,:,:)+Uz(i);
end

%%%--- interpolation functions for u, v, and w
Fu=griddedInterpolant(Z,Y,X,u);
Fv=griddedInterpolant(Z,Y,X,v);
Fw=griddedInterpolant(Z,Y,X,w);

%%%--- initializing arrays with statistics
var_ZX=NaN*ones(1,length(h_m));
var_ZX2=NaN*ones(1,length(h_m));
var_mast=NaN*ones(1,length(h_m));

%%%--- time_fast is a full time vector of 50 Hz covering 1800 s
time_fast=0:1/50:1800;
%%%--- time of the first beam scan per position
time_l=x_l./vel_l;

%%%--- simulating pitch (rotation)
Tp=3; % in s
Ap=5; % in deg
phase=0*pi/180;
pitch=Ap*sin((2*pi/Tp)*time_fast+phase);
%%%--- simulating roll(rotation)
Tp=3; % in s
Ap=0.0; % in deg
roll=Ap*cos((2*pi/Tp)*time_fast+phase);
%%%--- simulating yaw (rotation)
Tp=4; % in s
Ap=0.0; % in deg
yaw=Ap*cos((2*pi/Tp)*time_fast);
%%%--- simulating surge (velocity)
Tp=3; % in s
Ap=0.0; % in m/s
surge=Ap*sin((2*pi/Tp)*time_fast);
%%%--- simulating heave (velocity)
Tp=4; % in s
Ap=0.0; % in m/s
heave=Ap*cos((2*pi/Tp)*time_fast);

%%%--- time scanning over one height;
t_s=(0:1/50:1-1/50);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%--- loop to go through all heights
for k=1:length(h_m)
    disp(k)

    %%%--- lidar coordinates
    z_lo=h_m(k);

    %%%--- focused distance in m
    f_d=h_exp/cosd(phi);

    %%%--- height of the sonic
    z_m=h_m(k)+h_exp;

    %%%--- loop to go throughout the turbulence box (actually through all lidar positions ...
        along the box)
    options=optimset('Display','off');
    u_ZX=NaN*ones(length(x_l),1);
    v_ZX=NaN*ones(length(x_l),1);
    w_ZX=NaN*ones(length(x_l),1);
    U_ZX=NaN*ones(length(x_l),1);
    u_ZX2=NaN*ones(length(x_l),1);
    v_ZX2=NaN*ones(length(x_l),1);
    w_ZX2=NaN*ones(length(x_l),1);
    U_ZX2=NaN*ones(length(x_l),1);
    u_m=NaN*ones(length(x_l),1);
    v_m=NaN*ones(length(x_l),1);
    w_m=NaN*ones(length(x_l),1);
    vr_ZX=NaN*ones(length(x_l),50);
    vr_ZX2=NaN*ones(length(x_l),50);
    %%%--- displacement in the x-coordinate
    for i=1:length(x_l)
        %disp(i)

        %%%--- mast positions
        pos_mast=[z_m y_m x_l(i)];

        %%%--- mapping velocities at the mast positions
        u_m(i,1)=Fu(pos_mast);
        v_m(i,1)=Fv(pos_mast);
        w_m(i,1)=Fw(pos_mast);

        %%%--- new position of the beams
        pos_beams=[z_lo+n_z(:,3)*f_d y_lo+n_z(:,2)*f_d  (x_l(i)+vel_l*t_s)'+n_z(:,1)*f_d];
        beams_time=time_l(i)+t_s;

        %%%--- mapping velocities at the lidar positions
        u_l=Fu(pos_beams);
        v_l=Fv(pos_beams);
        w_l=Fw(pos_beams);

        %%%--- ZephIR calculations
        vr_ZX(i,:)=(u_l(:,1).*n_z(:,1)+v_l(:,1).*n_z(:,2)+w_l(:,1).*n_z(:,3))';
        fit=lsqcurvefit(@(x,X) abs(x(1)*cosd(X-x(2))+x(3)),...
            [0 0 0],theta',abs(vr_ZX(i,:)),[],[],options);
        U_ZX(i,1)=fit(1,1)/sind(phi);

        %%%--- going through each beam
        pos_beams2=NaN*ones(50,3);
        u_l2=NaN*ones(50,1);
        v_l2=NaN*ones(50,1);
        w_l2=NaN*ones(50,1);
        n_z2=NaN*ones(50,3);
        pitch_int=interp1(time_fast,pitch,beams_time);
        roll_int=interp1(time_fast,roll,beams_time);
        yaw_int=interp1(time_fast,yaw,beams_time);
        surge_int=interp1(time_fast,surge,beams_time);
        heave_int=interp1(time_fast,heave,beams_time);
        for j=1:50
            %%%--- rotation matrix
            R1=[cosd(pitch_int(j)) 0 sind(pitch_int(j));0 1 0;-sind(pitch_int(j)) 0 ...
                cosd(pitch_int(j))];
            R2=[1 0 0;0 cosd(roll_int(j)) sind(roll_int(j));0 -sind(roll_int(j)) ...
```

```
                cosd(roll_int(j))];
            R0=[cosd(yaw_int(j)) -sind(yaw_int(j)) 0;sind(yaw_int(j)) cosd(yaw_int(j)) 0;0 ...
                0 1];
            R=R0*R1*R2;
            n_z2(j,:)=n_z(j,:)*R;
            %%%--- position of the beams with rotation
            pos_beams2(j,:)=[z_lo+n_z2(j,3)*f_d y_lo+n_z2(j,2)*f_d ...
                (x_l(i)+vel_l*t_s(j))'+n_z2(j,1)*f_d];

            %%%--- mapping velocities at the lidar positions with rotation
            u_l2(j,1)=surge_int(j)+Fu(pos_beams2(j,:));
            v_l2(j,1)=Fv(pos_beams2(j,:)); %%%--- here you could include sway motion
            w_l2(j,1)=heave_int(j)+Fw(pos_beams2(j,:));
        end

        %%%--- ZephIR calculations with rotation
        vr_ZX2(i,:)=(u_l2(:,1).*n_z2(:,1)+v_l2(:,1).*n_z2(:,2)+w_l2(:,1).*n_z2(:,3))';
        fit=lsqcurvefit(@(x,X) abs(x(1)*cosd(X-x(2))+x(3)),...
            [0 0 0],theta',abs(vr_ZX2(i,:)),[],[],options);
        U_ZX2(i,1)=fit(1,1)/sind(phi);

    end

    %%%--- statistics of velocities
    var_ZX(1,k)=std(U_ZX)^2;
    var_ZX2(1,k)=std(U_ZX2)^2;
end

%%%--- mean variance ratio
mean(var_ZX2(1,:)./var_ZX(1,:))
```

DTU Wind Energy is a department of the Technical University of Denmark with a unique integration of research, education, innovation and public/private sector consulting in the field of wind energy. Our activities develop new opportunities and technology for the global and Danish exploitation of wind energy. Research focuses on key technical-scientific fields, which are central for the development, innovation and use of wind energy and provides the basis for advanced education at the university.

We have more than 250 staff members of which approximately 45 are PhD students. Research is conducted within three divisions: Wind Energy Materials & Components, Wind Turbine Design and Wind Energy Systems.

Technical University of Denmark
Department of Wind Energy
Frederiksborgvej 399
Building 118
4000 Roskilde
Denmark
Telephone 46 77 50 85

info@vindenergi.dtu.dk
www.vindenergi.dtu.dk