Delft University of Technology

Additional Thesis
AES4011-10

Estimation of Turbulence Intensity from Cup Anemometer-Based Mean Wind Speed Data via Fractal Interpolation

Pouriya Alinaghi
5078946

September 15, 2020

Abstract

In the well-known process of the turbine design, turbulence intensity (TI) plays a vital role in prediction of the power output and loads on the turbine's structure. TI is believed to be an important statistical parameter of the wind speed that can be extracted from the signals recorded by the dedicated sensors in the wind energy area. Despite the limitations of the mast-mounted sensors, they are probably more popular than LIDARs in wind energy applications. Although the sonic anemometers are reference tools in measuring turbulent features of wind, they are expensive instruments to be employed in a large-scale. In this regard, the cup anemometers appear to be the most commonly used instruments in the wind energy community. Accordingly, it would be tremendously advantageous if the 1-Hz cup anemometer data can be employed with the synthetic down-scaling idea to build the turbulence-like velocity signal fields. In this research, small-scale fluctuations are constructed via the Fractal Interpolation (FI) technique. In addition, this study aims to assess the compatibility of the FI technique in enhancing the cup anemometer data. The analysis has been carried out for the data collected in September 2018. Through this analysis, it is deduced that the cup anemometer data can be improved using the FI method. Subsequently, by applying the FI method, in most of the cases, the standard deviation values of the cup anemometer data are increased.

1 Introduction

Although the wind's irregularity seems fairly random, it has particular characteristics. These properties are described with some statistical parameters such as turbulence intensity, the probability density function of wind speed, the auto-correlation function, and the power spectral density function. In this regard, turbulence intensity (TI) is one of the most crucial parameters in the wind energy community that has noticeable applications in this field. TI is the ratio of the wind speed standard deviation to the mean wind speed [8]. In the well-known process of turbine design, both the power output estimation, as well as loads on the turbine's structure, are significantly affected by TI [18]. A rise in TI causes an increase in not only the power output but also the loads on the turbine's system [18]. Thus, TI plays a deterministic role in the turbine's system design.

To calculate TI, it is necessary to record a velocity signal of the wind. To this end, the wind speed can be measured using some popular instruments such as the cup anemometer, the sonic anemometer, and LIDAR. Cup and sonic anemometers are mast-mounted instruments, while LIDAR can be placed in both offshore and onshore platforms [6]. Besides, cup anemometers can be affected by environmental conditions like blowing dust or ice. The presence of dust and ice can lead to the friction increment and thus the reduction of the speed readings. Accordingly, the maintenance, service visits, and calibration of these instruments are substantially required [8]. Moreover, sonic anemometers can crash during the prolonged span of measurements under some meteorological situations like sandy storms, heavy rain, and snow, as well as icing [20]. Nowadays, with increasing the wind turbines' hub and rotor heights, meteorological towers must be constructed taller to measure the wind speed at desired levels. Not only does this lead to a more expensive project, but also in some countries it is prohibited to build towers taller than a particular height [6]. Nonetheless, tower-mounted sensors such as the cup and the sonic anemometers are still widely used in the wind energy field [6]. Furthermore, the wind energy community is comprehensively utilizing the anemometers, despite the fact that LIDAR is believed to be a high-tech instrument. This fact is due to the two reasons: control of the wind turbines' functionality, and estimating of the wind energy output in the field [10]. Besides, LIDAR is an extremely more expensive instrument compared to anemometers.

This research concentrates on the cup and sonic anemometers. These two anemometers are different in terms of the measurement techniques. With sonic anemometers, it is possible to have a straightforward measurement of the wind velocity in three directions x, y, and z. This measurement method allows the users to utilize a very high sampling frequency until 100 Hz or even beyond. These properties lead to a straight-forward measurement of turbulent momentum as well as turbulent buoyancy flux with the help of the eddy covariance method [17]. In [9], Peña stated that the present comprehension of atmospheric turbulence, to a serious extent, is dependent on three-dimensional sonic anemometers mounted on towers. Therefore, the sonic anemometer appears to be a reference tool in measurements of turbulence-like velocities in the wind energy and atmospheric turbulence field as well. On the other hand, among the various instruments dedicated to estimating wind speed, the cup anemometer is currently the most utilized gadget in the wind energy area [11], due to being more economical in comparison with other tools such as the sonic anemometer. Additionally, it represents a linear response in the range of normal wind speed [10]. Kristensen [7] referred to the cup anemometer as a strong and trustworthy tool that is simple to install, calibrate as well as it is a highly accurate sensor [7].

Since the cup anemometer is an inexpensive and high-accuracy tool in the wind energy community, this study suggests that it would be extremely convenient and worthwhile if one uses cup anemometer data with a synthetic down-scaling approach. The Fractal Interpolation (FI) technique is applied into the cup anemometer data, to

build synthetic turbulence. In this regard, this research wants to investigate whether the sonic anemometer data can be replaced by post-processed cup anemometer data using the FI technique. The data is collected from NREL website [19] at Colorado, USA. In section 2, the data will be described more in detail. In the sequel, section 3 is about the FI technique that is proposed by Barnsley [1] and improved by Ding in his paper [3] that is about Spatially-Randomized Fractal Interpolation (SRFI). Then, section 4 includes the results of the application of the aforementioned techniques in every ten-minute velocity signal of NREL website. Finally, conclusions about this idea would be presented in section 5.

2 Description of the data

The data is collected from the website [19] of National Renewable Energy Laboratory (NREL), National Wind Technology Center (NWTC), Colorado, USA. At the NREL site, there are three meteorological towers M2, M4, and M5 with the height of 80, 135, and 135 meters, respectively. In this study, the M5 tower located at the northeast end of the site is chosen for the further investigation. This tower measures the inflow into a wind turbine and contains various instruments at different heights from 3 to 134 meters. These instruments determine meteorological conditions including temperature, wind speed, and wind direction.

This research is interested in two known sensors that measure wind speed, the sonic anemometer, and the cup anemometer. The sonic anemometer at M5 tower measures wind speed in three directions x, y, and z, while the cup anemometer measures a single horizontal-direction wind speed. Also, the sonic anemometer is a 20-Hz sensor, whereas the cup anemometer is a 1-Hz sensor at the NERL site. Unfortunately, the sonic and the cup anemometers are not positioned at the same level in M5 tower. Therefore, two different heights with the least difference should be chosen. Hence, the data from the cup anemometer at 122 m altitude and sonic anemometer at 119 m altitudes is considered to be the input data for the analysis in this study.

As mentioned before, this research aims to build the synthetic turbulence signal with 1-Hz cup anemometer data to construct small-scale fluctuations and thus larger turbulence intensity. The sonic anemometer data is used to validate the synthetic velocity signals. Owing to the fact that these two sensors are not at the same level, the mean wind speed that is recorded by each sensor may differ from each other. Therefore, it is not possible to compare the TI values of two sensors. To this end, the mean wind speed is removed from the formula of TI and the standard deviation of the signals are considered to obtain an apple to apple comparison.

Moreover, the analysis is performed for the data that are collected in the course of September 2018. The Fractal Interpolation method is applied for every 10-minute velocity signal that is provided on the website [19]. In more detail, in the NERL website, for each day, there are 144 10-minute Matlab files containing different values for multiple instruments. Since the cup anemometer measures the horizontal wind speed, the horizontal wind speed components of the sonic anemometer u_x, u_y should be utilized. In this regard, the NERL data-set has the equivalent horizontal wind speed option for sonic anemometer that could be calculated using $\sqrt{{u_x}^2 + {u_y}^2}$.

3 Materials and Methods

To construct synthetic turbulence, two methods are utilized in this research. The first method is a fractal interpolation (FI) that was introduced by Barnsley [1]. The second one is the so-called spatially-randomized fractal interpolation (SRFI) that has been proposed by Ke-Qi Ding [3]. In the following subsections, these methods will be explained.

3.1 Fractal Interpolation

This technique is a kind of iterative affine mapping process to generate an artificial small-scale signal for the known large-scale anchor points. This process has been determined by Barnsley in his book [1] via an iterative function system (IFS). Let us consider only three anchor points $\{(x_0, \tilde{u}_0), (x_1, \tilde{u}_1), (x_2, \tilde{u}_2)\}$. In this case, the IFS is:

$$w_n \begin{pmatrix} x \\ u \end{pmatrix} = \begin{bmatrix} a_n & 0 \\ c_n & d_n \end{bmatrix} \begin{pmatrix} x \\ u \end{pmatrix} + \begin{pmatrix} e_n \\ f_n \end{pmatrix}, n = 1, 2 \tag{1}$$

where the term w_n is the FI operator and d_n is the vertical stretching factor. In addition, a_n, c_n, f_n and e_n are parameters that can be calculated in terms of d_n for $n = 1, 2$:

$$a_n = \frac{x_n - x_{n-1}}{x_2 - x_0}$$
$$c_n = \frac{\tilde{u}_n - \tilde{u}_{n-1}}{x_2 - x_0} - d_n \frac{\tilde{u}_2 - \tilde{u}_0}{x_2 - x_0}$$
$$e_n = \frac{x_2 x_{n-1} - x_0 x_n}{x_2 - x_0}$$
$$f_n = \frac{x_2 \tilde{u}_{n-1} - x_0 \tilde{u}_n}{x_2 - x_0} - d_n \frac{x_n \tilde{u}_0 - x_0 \tilde{u}_n}{x_2 - x_0}$$

Furthermore, to guarantee the continuity, the transformation should have these two conditions for $n = 1, 2$:

$$w_n \begin{pmatrix} x_0 \\ \tilde{u}_0 \end{pmatrix} = \begin{pmatrix} x_{n-1} \\ \tilde{u}_{n-1} \end{pmatrix}, w_n \begin{pmatrix} x_2 \\ \tilde{u}_2 \end{pmatrix} = \begin{pmatrix} x_n \\ \tilde{u}_n \end{pmatrix}.$$

After each iteration, there will be a new point exactly between each two anchor points. In the sequel, the newly generated points are going to be the anchor points for the next iteration. After some iterations a new synthetic signal is produced that has small-scale fluctuations compared to the initial setpoints, as long as the stretching factor d_n fulfills $(0 \leq |d_n| < 1)$ [1]. With this condition, FI is a contractive mapping function in the interval $[x_0, x_2]$.

The vertical stretching factor d_n plays a key role in terms of fractal dimensions. In this regard, Scotti and Meneveau in [12, 13], decided to choose $|d_1| = |d_2| = 2^{-1/3}$. Their choice is in agreement with the fractal sizes that turbulent velocity signal fields poss. Besides, the energy spectrum of their synthetic fields obeys the $-5/3$ scaling law.

In [2], Basu improved the method by taking advantage of choosing two different values for the stretching factors. He aimed to ensure the multi-fractal characteristics of turbulent fields. To this end, $d_1 = -0.887, d_2 = 0.676$ were determined as vertical scaling factors in his study.

In this study, FI is applied in every 10-minute signal in a way that, every three consecutive points like $\{(x_0, \tilde{u}_0), (x_1, \tilde{u}_1), (x_2, \tilde{u}_2)\}$ is divided into two intervals $[x_0, x_1]$ and $[x_1, x_2]$. For each interval, the scaling factor is determined through a random procedure from this set $\{\pm 0.887, \pm 0.676\}$. It should be noted that, absolute value of the chosen scaling factors must be different. For instance, this combination $d_1 = -0.887, d_2 = 0.887$ is not acceptable.

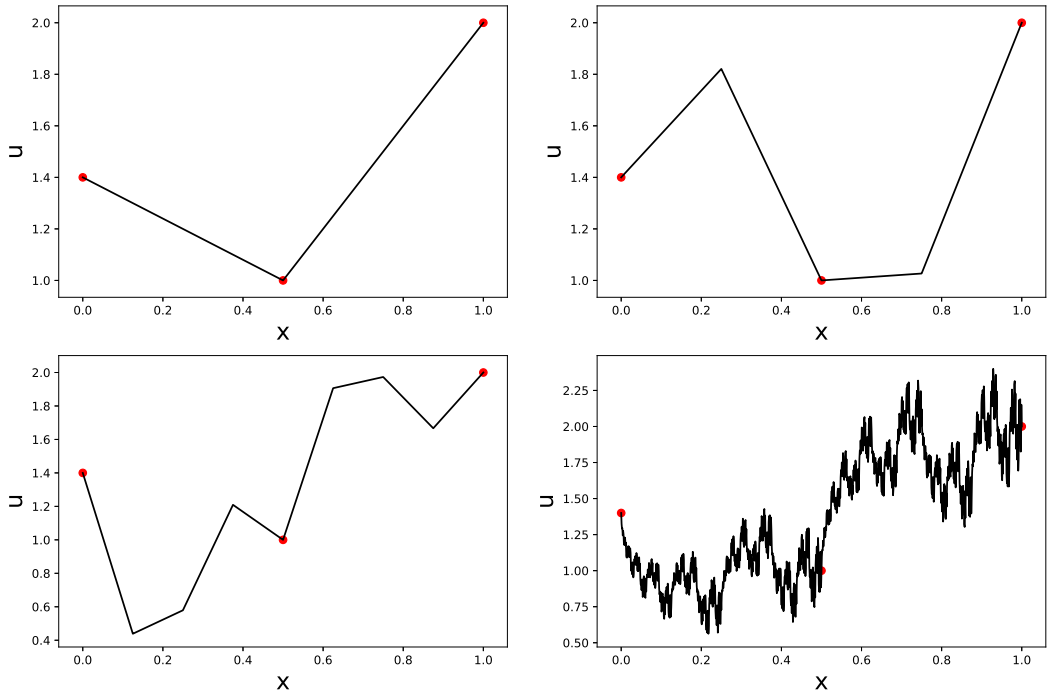

Figure 1: The figure represents the FI scheme, with given anchor points $\{(0, 1.4), (0.5, 1), (1, 2)\}$. The plots at the top-left, top-right, bottom-left and bottom-right, show the constructed turbulence-like signals after 1,2,3 and 10 iterations, respectively.

In [3], Ding mentioned, although FI can build small-scale fluctuations, there is a local similarity property in this process that can be rarely found in the real turbulent fields. In turbulence-like velocity fields, the position of similar fluctuations has a homogeneous random distribution. Besides, FI is highly dependent on the condition of each three points concerning each other. When the three points are nearly on a line, FI produces too small fluctuations that lead to the so-called co-linearity problem.

3.2 Spatially-Randomized Fractal Interpolation

In 2010, Ding offered another approach to build a synthetic turbulence signal [3]. Because of the local similarity problem in FI, Ding suggested the use of Spatially-Randomized Fractal Interpolation (SRFI). Besides, one of the largest differences between FI and SRFI is the absolute value of the vertical scaling factor. In the following, the SRFI procedure will be explained.

Initially, consider there are $2N$ anchor points $\{x_i\}, i = 0, 1, ..., (2N-1)$ in the interval $[x_0, x_{(2N-1)}]$ which is the original signal domain. At each step, through the particular IFS, the SRFI method is implemented in this research. Equation 2 represents the IFS of SRFI method:

$$\begin{aligned} &W_{SRFI}[u](x) = \tilde{u}_i + \frac{\tilde{u}_{i+1} - \tilde{u}_i}{x_{i+1} - x_i}(x - x_i) + d_i V_{j(i)}[u](y), \\ &x \in [x_i, x_{i+1}], i = 0, 1, 2, ..., (2N-1), \\ &y = x_{2j(i)} + 2(x - x_i) \end{aligned} \tag{2}$$

Where, W_{SRFI} is the SRFI operator, the first and second term is the linear interpolation for the original signal, d_i is the vertical stretching factor, and the term $V_{j(i)}[u](y)$ is calculated as

$$\begin{aligned} &V_{j(i)}[f](x) = f(x) - \left(\tilde{u}_{2j} + \frac{\tilde{u}_{2j+2} - \tilde{u}_{2j}}{x_{2j+2} - x_{2j}}(x - x_{2j})\right), \\ &x \in [x_{2j}, x_{2j+2}], j = 0, 1, 2, ..., (N-1) \end{aligned} \tag{3}$$

Equation 3 generates the large-scale fluctuations that is defined for any continuous function f with identical values at the anchor points $\{x_i\}, i = 0, 1, ..., (2N-1)$. Each large-scale fluctuation in the interval $[x_{2j}, x_{2j+2}], j = 0, 1, 2, ..., (N-1)$, corresponds to only two sub-intervals $[x_{i_1}, x_{i_1+1}], [x_{i_2}, x_{i_2+1}]$ if $i_1, i_2 \in \{0, 1, 2, ..., (2N-1) | i_1 \neq i_2\}$. In other words, there are only and only two small-fluctuations that are made out from a particular large-scale fluctuation. This mapping is done through a homogeneous random process.

The only remaining parameter that should be driven is the vertical stretching factor d_i. In each step, d_i has a random sign and its absolute value $|d_i|$ is produced by a Log-Poisson distribution [4, 15] in equation 4,

$$\begin{aligned} &P\left[|d_i| = \left(\frac{1}{2}\right)^{\gamma} \beta^n\right] = e^{-\lambda}\frac{\lambda^n}{n!}, i = 0, 1, 2, ... \\ &\lambda = \frac{1 - 3\gamma}{1 - \beta^3} ln(2) \end{aligned} \tag{4}$$

In this research, the values of β, γ are chosen to be $(2/3)^{1/3}, 1/9$. The reader can gain more information about the parameters β, γ in the paper [3].

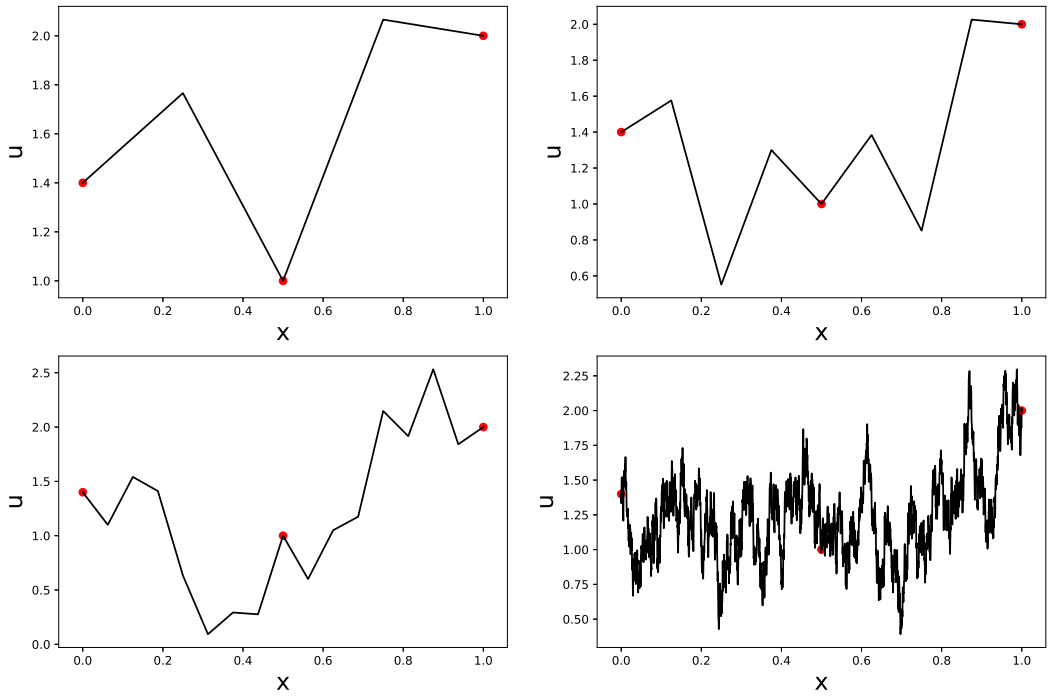

Figure 2: The figure represents the SRFI scheme, with given anchor points $\{(0, 1.4), (0.5, 1), (1, 2)\}$. The plots at the top-left, top-right, bottom-left and bottom-right, show the constructed turbulence-like signals after 1,2,3 and 10 iterations, respectively.

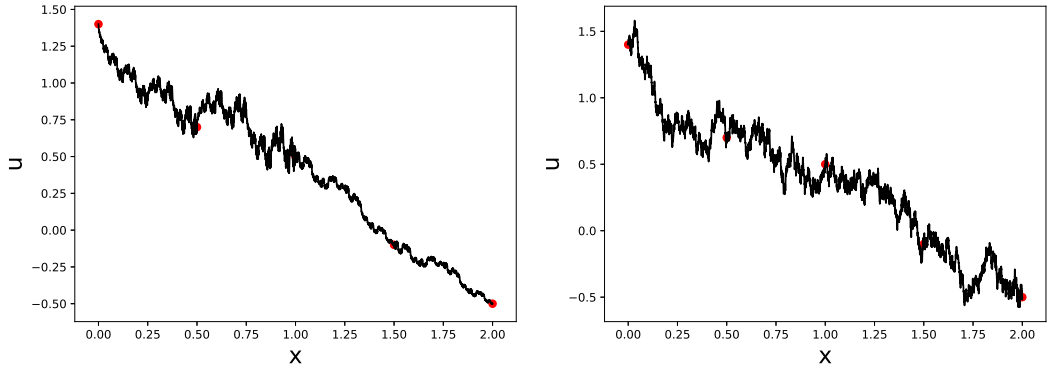

Figure 3: The left and right plots depict the FI and SRFI scheme after 10 iterations, with given anchor points $\{(0, 1.4), (0.5, 0.7), (1, 0.5), (1.5, -0.1), (2, -0.5)\}$. It can be seen that when three points are nearly on a line, FI cannot produce turbulence-like fluctuations. Rather, SRFI can be better regarding the co-linearity problem.

In addition, for the both aforementioned methods, higher-order structure functions are computed. The q^{th}-order structure function, $S_q(r)$ can be calculated like [2]:

$$S_q(r) = \langle |u(x + r) - u(x)|^q \rangle \approx r^{\zeta_q} \tag{5}$$

where r is the interval length that can change in a known range (inertial range in turbulence) and the angular bracket is dedicated to compute the mean in space. In case the velocity field is mono-affine, the term ζ_q should be a linear function of q. On the other hand, if ζ_q is a non-linear function of q, the field is called multi-affine. In this regard, Kolmogorov's celebrated 1941 hypothesis (K41) based on global scale invariance estimates that the term ζ_q should be $q/3$ in the inertial range [2, 5, 16]. Any deflection from $\zeta_q = q/3$ represents the inertial range intermittency and disprove the K41 hypothesis. Despite the strong experimental proofs for the existence of inertial range intermittency, it is still an unanswered issue [2].

Basu proved that with stretching factors of $|d_i| = 2^{-1/3}$, the FI method constructs a mono-affine field. In this case, the term ζ_q is compatible with the K41 prediction ($\zeta_q = q/3$). However, this is in contrast with the characteristics of the real turbulence-like velocity fields [2]. Accordingly, to consider the multi-fractal properties

of real velocity signal fields, Basu decided to choose the scaling factors $|d_1| = 0.887, |d_2| = 0.676$ for the FI method.

Moreover, the scaling exponents of the SRFI method, obey the She-Leveque 1994 (SL94) model [14]

$$\zeta_p = \frac{p}{9} + 2\left(1 - (\frac{2}{3})^{\frac{p}{3}}\right). \tag{6}$$

In Figure 4, the velocity structure functions of FI and SRFI are illustrated. It can be seen that SRFI produces more smooth structure functions, while FI's structure functions have some spurious oscillations.

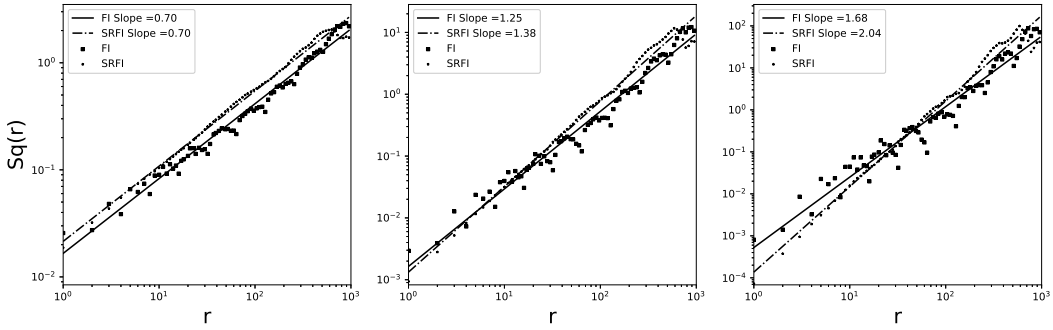

Figure 4: Plots from left to right show the 2nd, 4th, 6th orders of the velocity structure functions for the FI and the SRFI methods.

For readers who are interested in programming FI, SRFI, and velocity structure functions, the Python code of these functions is in the Appendix.

4 Results and Discussion

To apply FI and SRFI methods for real data, the analysis has been performed for September 2018. As the NREL website provides one file for every 10 minutes, there are 4320(144×30) files in the analysis for the entire month. Figure 5 represents an arbitrary 10-minute velocity signal. As illustrated in Figure 5, the sonic anemometer shows small-scale fluctuations (20 Hz), while the cup anemometer provides 1-Hz velocity signals. FI and SRFI are performed for each velocity signal of cup anemometer to increase the frequency rate from 1 Hz to more than 20 Hz. Such an analysis is done through all the 10-minute velocity signals for September 2018.

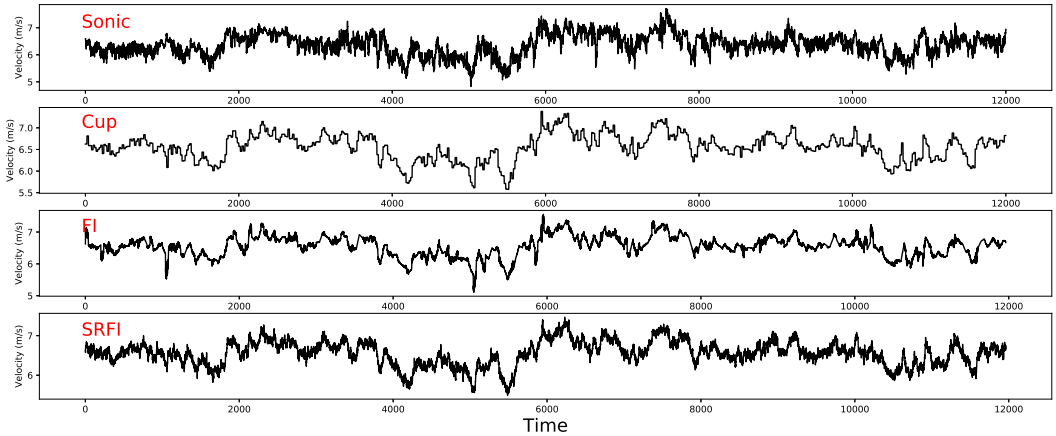

Figure 5: The figure demonstrates four plots which are the velocity signals of the sonic anemometer, cup anemometer, FI, and SRFI, subsequently.

As it is mentioned before, sonic and cup anemometers are positioned at two different heights on M5 tower. In this regard, the comparison of the TI values is irrelevant since the mean values of the velocity signals are different. Thus, the standard deviation is chosen as a relevant parameter for the assessment of FI and SRFI. Figures 6 and 7 show the probability density functions of standard deviation (STD) ratios for FI and SRFI. Firstly, it should be noted that in the data-set from the NREL website, cup anemometer data are very well

in agreement with the sonic anemometer data, but in most of the cases, cup anemometer data underestimate the STD values since the sampling rate of cup anemometer is much less than the sampling rate of the sonic anemometer. Nevertheless, the middle plot in Figures 6 and 7 demonstrates that FI and SRFI increase the STD values up to 8% and 4%. Figure 6 depicts that more than 70% of cup anemometer STD values are increased up to 8%. Similarly, Figure 7 illustrates that SRFI increases the cup anemometer STD values in more than 40% of cases.

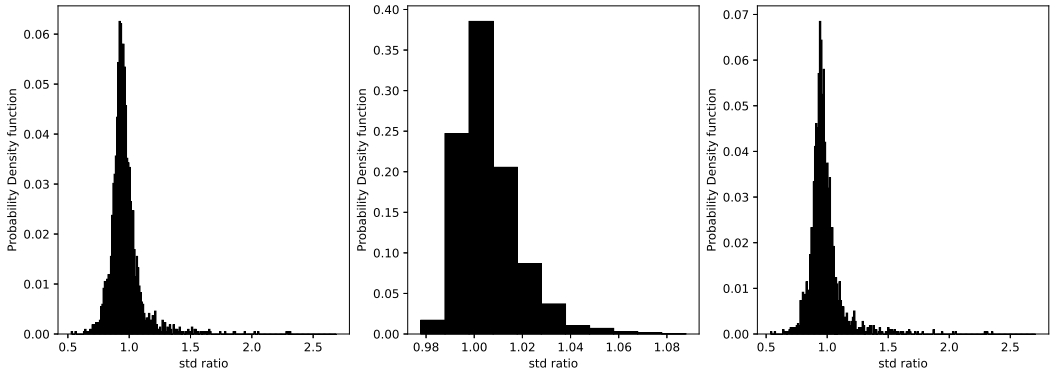

Figure 6: The figure includes three probability density function (PDF) plots. The plots from left to right represent the PDF of $\frac{Standard\ deviation\ of\ the\ cup\ anemometer\ data}{Standard\ deviation\ of\ the\ sonic\ anemometer\ data}$, $\frac{Standard\ deviation\ of\ FI\ data}{Standard\ deviation\ of\ the\ cup\ anemometer\ data}$, $\frac{Standard\ deviation\ of\ FI\ data}{Standard\ deviation\ of\ the\ sonic\ anemometer\ data}$.

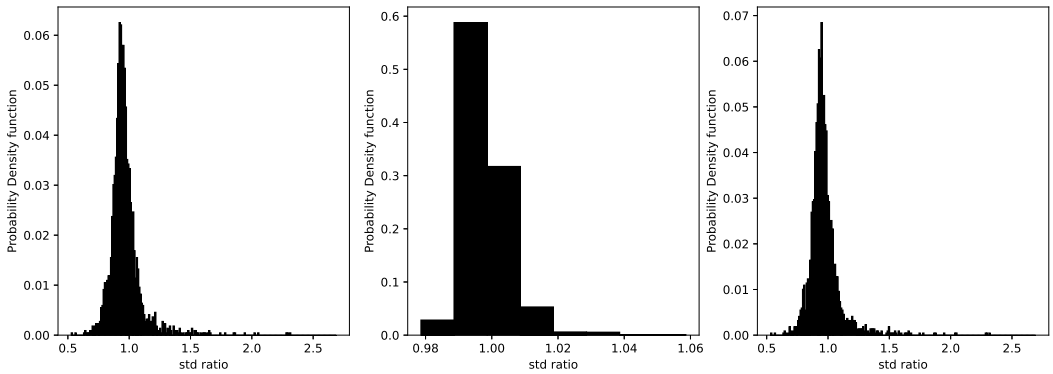

Figure 7: The figure provides three probability density function (PDF) plots. The plots from left to right represent the PDF of $\frac{Standard\ deviation\ of\ the\ cup\ anemometer\ data}{Standard\ deviation\ of\ the\ sonic\ anemometer\ data}$, $\frac{Standard\ deviation\ of\ SRFI\ data}{Standard\ deviation\ of\ the\ cup\ anemometer\ data}$, $\frac{Standard\ deviation\ of\ SRFI\ data}{Standard\ deviation\ of\ the\ sonic\ anemometer\ data}$.

Moreover, it ought to be noticed that 1 Hz cup anemometer data are not generally provided by the wind energy companies. In the wind energy community, companies generally provide 10-minute averaged data. For future work, the author suggests that it would be more applicable if one can apply the FI technique for the 10-minute averaged data. But one should be careful that every three points in the 10-minute averaged data can not be used directly in the FI technique since those triplets are the mean of each 10-minute interval. Also, in this research, it is assumed that three points represent down-samples of the original series, while in [13] Scotti assumed that three points represent both filtered and down-samples of the original series. Based on [13], it is possible to retrieve STD values of the FI signals with three averaged points as inputs. In this way, the integral moments of FI signals should be calculated. For more information regarding this issue, the reader is encouraged to read papers [2, 12, 13].

Besides, since both the FI and SRFI methods have random processes in their IFS, it would be better to carry out more realizations for each 10-minute velocity signal. In the sequel, through statistical relationships, one can use the 90^{th} percentile of produced STD values. Accordingly, with performing a large number of realizations one can gain more STD values.

5 Conclusions

It can be concluded that FI techniques are reliable mathematical tools for constructing turbulence-like velocities. Not only do these techniques provide small-scale fluctuations, but the multi-fractal properties of real turbulence-like velocity fields can be modeled up to a reliable extent. Besides, both methods are applicable in increasing STD values of cup anemometers data. In recent studies, the FI technique has been used to build synthetic turbulence in finer scales for models like Large-Eddy Simulation. In this study, the FI methods are applied to the cup anemometer data to make finer-scale fluctuations. Since the cup anemometers are not only reasonably economical compared to the sonic anemometers but also the most commonly used anemometers in the wind energy community. In the end, the author suggests that the application of the post-processed cup anemometers data via the FI techniques can be improved and widely used in the wind energy community by companies and universities. Economically, this idea is extremely beneficial for the organizations that are not interested in spending budgets on vast sonic anemometers.

Acknowledgments

In this section, I would like to express my highly appreciations towards Sukanta Basu because of his tremendous support during the summer and the Covid-19 situation. Also, I am grateful to him since he provided the opportunity for me to use his workstation as a remote system. Otherwise I was not able to analyze such a big data in this study.

Appendix

The Python code for FI, SRFI, and structure functions is available in this section.

```
import numpy as np

# FI

# This function produces small-scale fluctuations for every triplet.
# The NN term determines the number of iterations.
def FI(x,u,d,NN):
    b = x[2]-x[0]
    a = np.zeros(2)
    c = np.zeros(2)
    e = np.zeros(2)
    f = np.zeros(2)

    for n in [1,2]:
        a[n-1] = (x[n]-x[n-1])/b
        e[n-1] = (x[2]*x[n-1]-x[0]*x[n])/b
        c[n-1] = (u[n]-u[n-1]-d[n-1]*(u[2]-u[0]))/b
        f[n-1] = (x[2]*u[n-1]-x[0]*u[n]-d[n-1]*(x[2]*u[0]-x[0]*u[2]))/b

    M1 = np.zeros((2,2))
    M1[0] = [a[0],0]
    M1[1] = [c[0],d[0]]
    M2 = np.zeros((2,2))
    M2[0] = [a[1],0]
    M2[1] = [c[1],d[1]]
    EF1 = np.zeros((2,1))
    EF1[0] = e[0]
    EF1[1] = f[0]
    EF2 = np.zeros((2,1))
    EF2[0] = e[1]
    EF2[1] = f[1]

    OLD = np.zeros((2,3))
```

```
    OLD[0,:] = x
    OLD[1,:] = u
    for i in np.arange(1,NN):
        NEW_1 = M1 @ OLD + EF1*np.ones((1,len(OLD[0,:])))
        NEW_2 = M2 @ OLD + EF2*np.ones((1,len(OLD[0,:])))
        x = np.concatenate((NEW_1[0,:],NEW_2[0,:]), axis = None)
        u = np.concatenate((NEW_1[1,:],NEW_2[1,:]), axis = None)
        OLD = np.zeros((2,len(x)))
        OLD[0,:] = x
        OLD[1,:] = u

    x, xi = np.unique(x, return_index = True)
    u = u[xi]


    X = np.zeros((2,len(x)))
    X[0,:] = x
    X[1,:] = u

    return X

# FIT function is employed for performing FI function through a 10-min velocity signal.
def FIT(xx,uu,NN):
    XX = []
    UU = []


    L = len(xx)
    for i in np.arange(1,L-1,2):
        X = np.array([xx[i-1],xx[i],xx[i+1]])
        U = np.array([uu[i-1],uu[i],uu[i+1]])

        dx = [0.887,-0.887,-0.676,0.676]
        r1 = np.random.random(1)
        d1 = dx[0]*(r1>=0.5) + dx[1]*(r1<0.5)
        r2 = np.random.random(1)
        d2 = dx[2]*(r2>=0.5) + dx[3]*(r2<0.5)
        r3 = np.random.random(1)
        d = [d1,d2] * (r3>=0.5) + [d2,d1] * (r3<0.5)
        d = np.squeeze(d)

        f = FI(X,U,d,NN)
        XX.extend(f[0,:])
        UU.extend(f[1,:])

    X1 = np.zeros((2,len(XX)))
    X1[0,:] = XX #x2
    X1[1,:] = UU #y2

    return X1

# SRFI
# To build small-scale fluctuations using the SRFI method, three functions V, W and SRFI
# are utilized. To use the SRFI method, it is enough to use SRFI function
# in this code with N iterations.
def V(x,u):
    x = np.array(x)
    u = np.array(u)
```

```
    v = np.zeros(len(x))
    for i in np.arange(1,len(x),2):
        v[i] = u[i] - 0.5 * (u[i-1]+u[i+1])

    return v

def W(x,u):

    X = np.zeros(2*len(x)-1)
    for i in range(len(x)):
        X[2*i] = x[i]
    for i in range(len(x)-1):
        X[2*i+1] = 0.5*(x[i]+x[i+1])

    U = np.zeros(2*len(x)-1)
    for i in range(len(x)):
        U[2*i] = u[i]
    for i in range(len(x)-1):
        U[2*i+1] = 0.5*(u[i]+u[i+1])

    v = V(x,u)

    v_nonzero = np.zeros(int(len(v)/2))
    for i in range(len(v_nonzero)):
        v_nonzero[i] = v[2*i+1]
    fluc_raw = np.concatenate((v_nonzero,v_nonzero), axis = None)

    fluc_random = np.random.choice(fluc_raw, size = len(fluc_raw),replace = False)

    fr = fluc_raw
    nn = 1000 #int(m.factorial(len(fr))/m.factorial(len(fr)/2)**2)
    for i in range(nn):
        fluc_random = np.random.choice(fr, size = len(fr),replace = False)
        j = np.arange(len(fluc_random)-1)
        if (np.all(fluc_random[j]!=fluc_random[j+1])):
            break

    dx = [-0.887,0.887,-0.676,0.676]
    di = np.random.choice(dx,size = len(fluc_random))

    gamma = 1/9
    beta = (2/3) ** (1/3)
    lamda = (1-3*gamma)/(1-beta**3)*np.log(2)
    pp = [1,2]
    di = np.zeros(len(fluc_random))
    for i in range(len(fluc_random)):
        p1 = (np.random.poisson(lamda,1)[0])
        rr1 = np.random.choice(pp,1)[0]
        di[i] = ((1/2)**gamma * beta ** p1 * (-1)**rr1)

    fluc_random = fluc_random * di

    fluc = np.zeros(len(X))
    for i in range(len(fluc_random)):
        fluc[2*i+1] = fluc_random[i]

    U_new = U + fluc
```

```
    return X, U_new

def SRFI(x,u,N):

    for i in range(N):

        x,u = W(x,u)

    return x,u


# Sturcture Function
# Si: The i-th order of velocity structure function

def StructureFunction_NaN(x,noct):

    N = len(x)
    N = np.int(N)

    a0      = 1
    nvoie   = 10
    scales  = np.arange(np.log2(a0), noct+np.log2(a0) , 1/nvoie)
    R       = np.unique(np.around(2 ** scales))

    S1 = np.zeros(len(R))
    S2 = np.zeros(len(R))
    S3 = np.zeros(len(R))
    S4 = np.zeros(len(R))
    S5 = np.zeros(len(R))
    S6 = np.zeros(len(R))

    k = 0
    for i in np.arange(1,len(R)+1):
        r = R[i-1]  #int(R[i])
        r = np.int(r)
        j = np.arange(1,N - r + 1)   #1:(N-r);
        j = j.astype(int)
        y = np.abs(x[j+r-1]-x[j-1])

        S1[k] = np.nanmean(y**1)
        S2[k] = np.nanmean(y**2)
        S3[k] = np.nanmean(y**3)
        S4[k] = np.nanmean(y**4)
        S5[k] = np.nanmean(y**5)
        S6[k] = np.nanmean(y**6)

        k = k+1

    return [S1,S2,S3,S4,S5,S6,R]
```

References

Michael F Barnsley. *Fractals everywhere*. Academic press, 2014.

Sukanta Basu, Efi Foufoula-Georgiou, and Fernando Porté-Agel. Synthetic turbulence, fractal interpolation, and large-eddy simulation. *Physical Review E*, 70(2):026310, 2004.

Ke-Qi Ding, Zhi-Xiong Zhang, Yi-Peng Shi, and Zhen-Su She. Synthetic turbulence constructed by spatially randomized fractal interpolation. *Physical Review E*, 82(3):036311, 2010.

Berengere Dubrulle. Intermittency in fully developed turbulence: Log-poisson statistics and generalized scale covariance. *Physical review letters*, 73(7):959, 1994.

Uriel Frisch and Andreĭ Kolmogorov. *Turbulence: the legacy of AN Kolmogorov*.

Jay Prakash Goit, Susumu Shimada, and Tetsuya Kogaki. Can lidars replace meteorological masts in wind energy? *Energies*, 12(19):3680, 2019.

L Kristensen. The perennial cup anemometer. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 2(1):59–75, 1999.

James F Manwell, Jon G McGowan, and Anthony L Rogers. *Wind energy explained: theory, design and application*. John Wiley & Sons, 2010.

Alfredo Peña, Ebba Dellwik, and Jakob Mann. A method to assess the accuracy of sonic anemometer measurements. *Atmospheric Measurement Techniques*, 12(1):237–252, 2019.

Santiago Pindado, Javier Cubas, and Félix Sorribes-Palmer. The cup anemometer, a fundamental meteorological instrument for the wind energy industry. research at the idr/upm institute. *Sensors*, 14(11):21418–21452, 2014.

Santiago Pindado, Enrique Vega, Alejandro Martínez, Encarnación Meseguer, Sebastián Franchini, and Imanol Pérez Sarasola. Analysis of calibration results from cup and propeller anemometers. influence on wind turbine annual energy production (aep) calculations. *Wind Energy*, 14(1):119–132, 2011.

A Scotti and C Meneveau. A fractal model for large eddy simulation of turbulent flow. *Physica D: Nonlinear Phenomena*, 127(3-4):198–232, 1999.

Alberto Scotti and Charles Meneveau. Fractal model for coarse-grained nonlinear partial differential equations. *Physical review letters*, 78(5):867, 1997.

Zhen-Su She and Emmanuel Leveque. Universal scaling laws in fully developed turbulence. *Physical review letters*, 72(3):336, 1994.

Zhen-Su She and Edward C Waymire. Quantized energy cascade and log-poisson statistics in fully developed turbulence. *Physical Review Letters*, 74(2):262, 1995.

Katepalli R Sreenivasan and RA Antonia. The phenomenology of small-scale turbulence. *Annual review of fluid mechanics*, 29(1):435–472, 1997.

Irene Suomi and Timo Vihma. Wind gust measurement techniques—from traditional anemometry to new possibilities. *Sensors*, 18(4):1300, 2018.

Matthias Türk and Stefan Emeis. The dependence of offshore turbulence intensity on wind speed. *Journal of Wind Engineering and Industrial Aerodynamics*, 98(8-9):466–471, 2010.

National Renewable Energy Laboratory website. https://wind.nrel.gov/MetData/.

S Yahaya and JP Frangi. Cup anemometer response to the wind turbulence-measurement of the horizontal wind variance. 2004.