DWM User's guide: Version 2.2

Yujia Hao

Revised April 4, 2015 for FAST 8

1. Introduction

The Dynamic wake meandering model (DWM) is a simulator implemented in FAST, which calculates the wake deficit and meandered wake centers behind a turbine. The advantage of incorporating the DWM model into FAST is to provide the ability to obtain both the turbine power and blade loads of a downstream turbine, while maintaining an acceptable low calculation cost compared to CFD. DWM uses the rotor induction factor calculated by AeroDyn as well as some other parameters such as ambient turbulence intensity as the model input. The underlying theory behind the DWM includes the Navier Stokes equation (N-S equation) and Taylor's frozen turbulence hypothesis, which are applied to obtain the wake velocity and the meandered wake center position.

2. DWM Routines implemented in FAST

2.1. Overview

The DWM model is currently constituted by two major sub-models – a model of the quasi-steady wake deficit which includes the effect of the small scale wake added turbulence, and a stochastic model of the downstream wake meandering process which includes the effect of the large scale wake added turbulence, as seen in Fig. 1. The DWM model is implemented in the NWTC Computer-Aided Engineering (CAE) tools framework, which includes FAST, TurbSim and AeroDyn. TurbSim is used to generate random stochastic wind fields, and AeroDyn is used to calculate the aerodynamic loads on wind turbine blade elements based on the velocities and positions provided by the dynamics analysis routines from FAST. For a single turbine, the inputs of the DWM model come from two sources – the AeroDyn loads results and the DWM input text files. For a downstream turbine, the DWM inputs also include the wake deficit and the meandered wake centers of upstream turbines. The main outputs of the DWM model are the wake deficit velocity and the meandered wake center positions in the space domain behind the investigated turbine. The basic simulation flowchart for a single turbine is summarized in Fig. 2. More details can be found in the paper "*Implementing the Dynamic Wake Meandering Model in the NWTC Design Codes*" by Hao et al.[1]

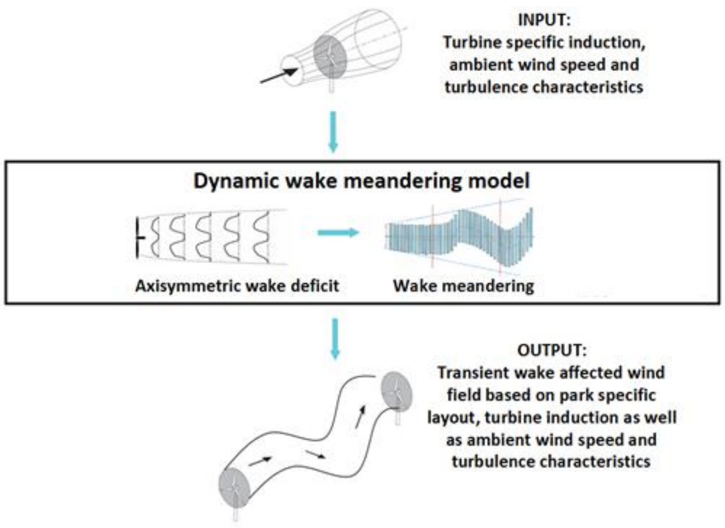

INPUT:
Turbine specific induction, ambient wind speed and turbulence characteristics
Dynamic wake meandering model
Axisymmetric wake deficit
Wake meandering
OUTPUT:
Transient wake affected wind field based on park specific layout, turbine induction as well as ambient wind speed and turbulence characteristics

Figure 1. Overview of the DWM sub-models.[2]

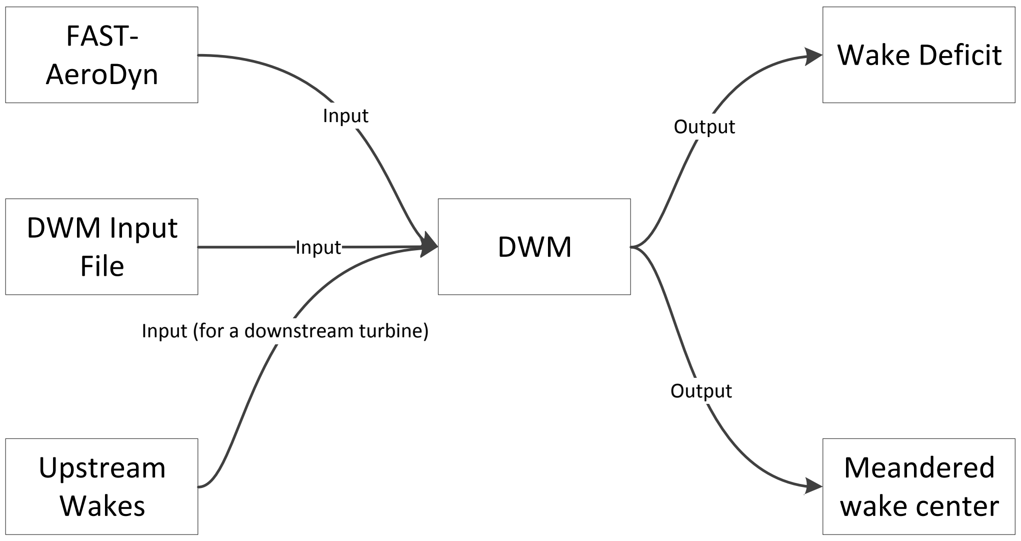


Figure 2. The flowchart of DWM implemented in FAST.

2.2. Compiling of the DWM implemented in FAST

The standard version of FAST by default includes the DWM routines, thus the approach of compiling DWM implemented in FAST is the same as compiling FAST with some additional DWM source files added. It is not necessary to re-compile FAST unless you want to make changes to the code or want to perform the DWM simulation using an operating system different from Windows. The method that compiles FAST can be found in the FAST web page at:
http://nwtc.nrel.gov/FAST8

3. DWM Driver Program

3.1. Overview

For a DWM simulation applied for more than a single turbine, individual FAST program runs sequentially and separately for each turbine from the very upstream one to the very downstream – and the outputs are written out as files for each turbine. Thus a driver program has been built to manage the series of FAST simulations. The driver program pre-screens the wind farm and read in the model parameters, and manages the running of FAST program for each turbine.

To perform a series of DWM simulations, instead of *the* FAST V8 executable file, the *DWM_Driver.exe* should be run (the DWM driver program will call a series of FAST to run). The FAST V8 executable file as well as the *DWM_Driver.exe* should both be put under the same directory location where the FAST primary input files are located.

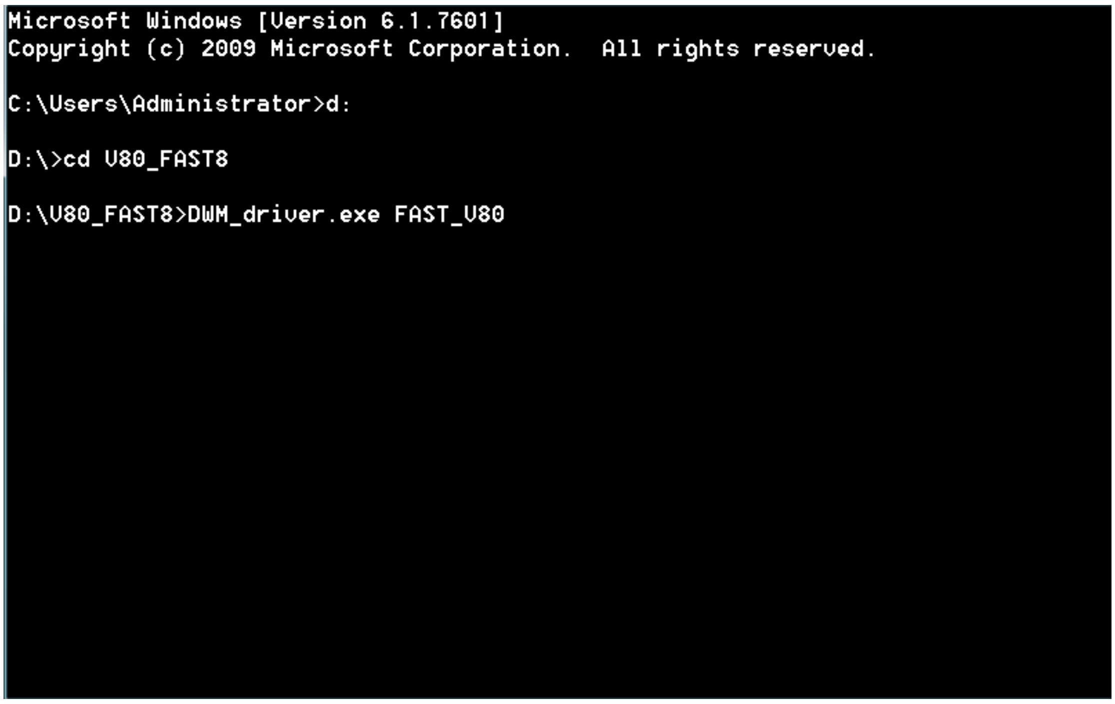
The input file *wind_farm.txt* (the name and sub-directory of this file should not be changed) which includes the DWM routines parameters is read by the DWM driver program. This input file should be put in the sub-directory */DWM-driver*. The input file will be discussed in detail in Section 4. When the DWM simulation finishes, the DWM outputs files will be written out under the sub-directory */DWM-results*.

3.2. Running DWM driver program

To run the driver program *DWM_Driver.exe*, open a command prompt window in the directory where the FAST primary input files are located. The command-line syntax to run the driver program is:

```
DWM_Driver.exe  <FAST primary input file, without the .fst extension>
```

An example of the DWM driver program command line input is shown in Fig.3. The file type suffix of the FAST input file is assumed to be *.fst*.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Administrator>d:

D:\>cd V80_FAST8

D:\V80_FAST8>DWM_driver.exe FAST_V80
```

Figure 3. Example DWM-Driver program command line input.

More information related to the compiling and running of the DWM driver program and the FAST program are provided in the document “DWM-FAST compiling introduction”, which is zipped in the DWM Driver Archives.

3.3. DWM driver program archive

Currently, the DWM routines are treated as routines of AeroDyn such that the DWM routines source files are included in FAST V8. The DWM driver program archive contains the driver program executable file, the drive program source code, and some sample files listed in Table 1.

DWM-Driver Source File	Description
DWM_driver.exe	DWM driver program executable file
DWM_driver_wind_farm.f90	DWM-Driver program main file
DWM_driver_wind_farm_mod.f90	Contains the DWM-Driver data
DWM_driver_wind_farm_sub.f90	Contains the subroutines that used by DWM-Driver
Wind_farm.txt	Driver program input file

Table 1. DWM-Driver program archive file.

4. DWM Driver Program Input File

DWM reads a text input file to set the parameters required for the program to execute. Do not change the name of the text input file unless corresponding modifications are done in the driver program source file. DWM assumes that parameters are located on specific lines, so do not add or remove lines from the sample input files included in the archive except the turbine coordinate section. The heading for each parameter following each input in the file contains its name, a description, and the units for that parameter. The names of the parameters are provided for reference, and DWM does not read those names from the input file. An example DWM input file is included in Appendix A. The DWM input parameters are:

HubHt: The turbine hub height [m]

This is the turbine hub height. Although it is not checked for consistency, the value of this parameter should match the equivalent value in the FAST input file.

RotorR: The radius of rotor blade [m]

This parameter specifies the distance from the rotor apex to the blade tip. Although it is not checked for consistency, the value of this parameter should match the equivalent value in the FAST input file.

NumWt: The total number of wind turbines [-]

This parameter specifies the total number of the wind turbine that to be simulated. The value of this parameter cannot be smaller than 1.

Uambient: The ambient wind velocity [m/s]

This is the mean wind speed at the turbine hub height. Although it is not checked for consistency, the value of this parameter should match the equivalent value in the TurbSim input file when creating the wind file.

TI: The ambient turbulence intensity [%]

This is the average ambient turbulence intensity in percentage at the hub height or the TI of the TurbSim generated wind file. The value of this parameter cannot be negative.

ppR: The number of points per radius [-]

DWM uses a finite difference scheme to solve the fluid dynamic system. This parameter defines the number of calculation node in the spanwise direction. In general, larger nodal values cost more calculation time but yield more precise results. The value of this parameter must be an integer and *50* is recommended.

Domain_R: The radial domain size [-]

DWM applies an axisymmetric coordinate system to solve the fluid dynamic system. This parameter scales the rotor radius to determine the radial distance from the center of the calculation domain to the very boundary edge. The value of this parameter cannot be smaller than 3 and should be adjusted based on the turbine layout if there are more than single turbines in this simulation. If the wind direction is not down the row, then the radial domain size should be larger than *min{3, spacing*sin(theta)}*, where spacing is the distance between the upwind and downwind turbines, and *theta* is the angle between the line connecting the two turbines and the inflow wind direction.

Domain_X: The longitudinal domain size [-]

This parameter scales the rotor radius to determine the longitudinal domain size counted from the investigated rotor plane. The value of this parameter must be positive and should be determined based on the turbine layout if there is more than a single turbine in this simulation. For example, this value should be larger than the maximum spacing between two neighboring turbines if a downstream turbine is only affected by the closest upwind turbine.

Meandering_simulation_time_length: The total number of simulation time steps in the meandering wake model for a fixed cross plane [-]

Meandering_Moving_time: The total number of simulation time steps in the meandering wake model for a moving cross plane [-]

These two parameters control the wake meandering sub-model. Taylor's frozen turbulence hypothesis is applied for the downstream advection of the wake, and the fundamental assumption of this approach is that the wake transport in the atmospheric boundary layer can be modeled, by considering the wake to act as a passive tracer driven by large scale turbulence.[3] To calculate the wake displacement in the vertical and lateral directions, the wake is modeled as constituted by a cascade of wake deficits, each "emitted" at consecutive equally spaced time increments, in agreement with the passive tracer analogy.

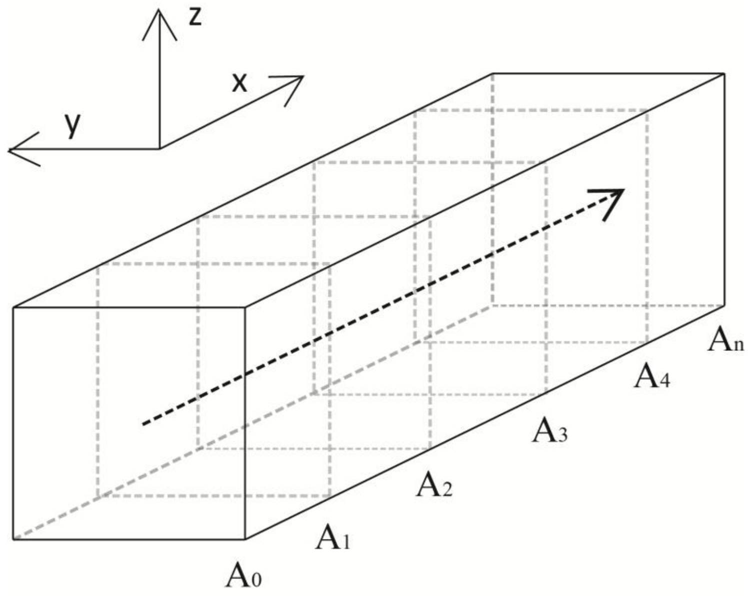


Figure 4. Illustration of the wake meandering model space domain.

At every time step, a cross-plane is released from the rotor plane and marches downstream while the wind properties of this cross-plane are constant. The entire space domain behind the turbine consists of multiple cross-planes, which are all released from the rotor plane with the spacing between two neighboring cross-planes equal to the ambient velocity multiplied by the time step interval. In each cross-plane there is a wake center position. For a single cross-plane, it marches a constant distance downstream at every time step, and a new wake center position is obtained using a filter function, with the averaged vertical and lateral velocity calculated based on the wake center position of the last time step.

As illustrated in Fig. 4, the wake meandering model space domain behind the turbine is a cuboid, with the x direction in the mean flow nominal direction, the y direction in the lateral direction, the z direction in the vertical direction, the investigated turbine located at the cross plane location A_0, and the very downstream boundary plane at A_n. The wake meandering model functions as follows.

First, at time t_0, a frozen cross plane C_0 is released from the turbine plane A_0 and starts to advect downstream. After time $\triangle t$, this frozen cross plane C_0 advects to the location of the cross plane A_1. Finally at time T this frozen cross plane C_0 reaches the location where the cross plane A_n is located. Throughout this whole process at each time step and at each cross plane location A, a meandered wake center vertical and lateral position coordinate in the cross plane C_0 is returned, and the wake center coordinate in the x direction is equal to the ambient velocity multiplied by the local total time which is counted from the original release. A cross plane C is released at every time step and the same approach discussed above is applied for each cross plane.

The parameter *Meandering_Moving_time* specifies the total time *T* for which a frozen cross plane travels, and the parameter *Meandering_simulation_time_length* specifies the total number of the cross planes that advect from the turbine cross plane A_0 to the very downstream boundary plane A_n.

The two parameters are closely correlated with the simulation time, thus the minimum value of the two parameters is shown in Eq. (1) and Eq. (2) below.

$$\mathrm{MSTL} \geq \mathrm{Tmax}/(\frac{\frac{2\mathrm{R}}{\mathrm{ppR}}}{0.32\mathrm{Uambient}} \cdot 10) + 1 \tag{1}$$

$$\mathrm{MMT} \geq \mathrm{spacing} \cdot \frac{\mathrm{ppR}}{10} + 1 \tag{2}$$

MSTL is the parameter *Meandering_simulation_time_length*, *MMT* is the parameter *Meandering_Moving_time, Tmax* is specified in the FAST input file which is the total run time (s), and *spacing* is the distance normalized by rotor diameter from the downstream investigated turbine to the very upstream turbine whose wake will affect this downstream turbine.

WFLowerBd: The lower bound of the wind file [m]

When generating the wind file in TurbSim, the wind file has a 2D y-z domain size. This parameter WFLowerBd corresponds to the lower bound height of the TurbSim generated inflow wind file. The value of WFLowerBd is equal to HubHt – 0.5*GridHeight, where HubHt and GridHeight are the parameters in the TurbSim input files. Although it is not checked for consistency, the value of this parameter should match the equivalent value in the TurbSim input file

Winddir: The ambient wind direction [degree °]

This parameter specifies the incoming ambient wind direction in units of degrees. The value of this parameter must be between 0 and 360. The degree measurements are top-wise as shown in Fig. 5 below. If looking down on the wind farm, 0° means the wind comes from the very top or north, 90° the very right or east, 180° the very bottom or south and 270° the very left or west.

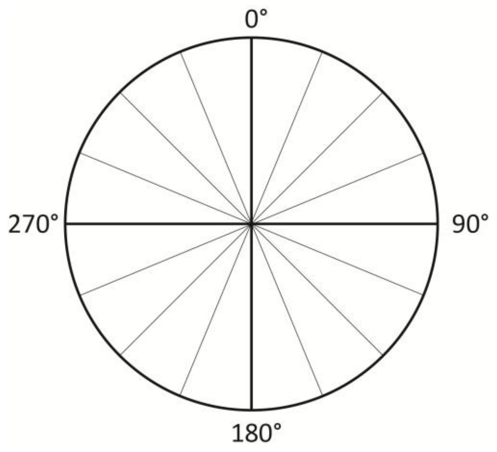


Figure 5. Illustration of wind direction degree measurements.

DWM.exe name: The file rootname of the FAST program

This is the root name of the FAST executable file.

XCoordinate, YCoordinate: The coordinate of turbine [-]

These parameters are the coordinates of the turbines at x and y directions in the Cartesian coordinate system which is scaled by the rotor diameter. The total number of the lines must be equal to the total number of the turbines. At each line, the x coordinate is typed in first and it is followed by the y coordinate of the same turbine. For example, if the radius of the circle in Figure 5 is 1D, then the coordinates of the intersections of the horizontal line and the circle from left to right are (-1, 0) and (1, 0) respectively; the coordinates of the intersections of the vertical line and the circle from top to bottom are (0, 1) and (0, -1) respectively. The driver program automatically sorts the turbines from the most upwind turbine to the most downwind turbine based on the turbines layout and the inflow wind direction.

5. Output Files

Besides the normal FAST output files that record the power and loads results, several text files are created by DWM as the output files. They have the root name of the turbine index and some keywords indicating what kind of results are stored in these text files.

FAST .out and .elm output files

Since the DWM runs sequentially for each turbine, if there is more than a single turbine, the FAST .out and .elm output files are renamed corresponding to the investigated turbine index. For example, the *FastOutput_Turbine_2.out* and *FastElm_Turbine_2.elm* stand for the FAST .out and FAST .elm output files for turbine 2 respectively.

Induction factor output files

The induction factor describes the amount the free stream wind speed is reduced by the wind turbine, and is one of the inputs for the DWM model. However, the axial induction factors given by AeroDyn must be interpreted carefully, as the tip loss correction in AeroDyn causes the calculated induction values near the tip to be too large and unphysical. To calculate the axial induction factors for DWM, which reflect the exchange of momentum between the fluid and the rotor, the induction factor *a* is generated directly from the thrust coefficient C_t, which is related to the thrust force *F*, the swept area *A*, the air density ρ and the local wind speed *U*. The equations reflecting the above relationships for the i^{th} annulus are shown in Eq. (3) and Eq. (4).

$$Ct_i = {F_i}\Big/\left(\frac{1}{2}\rho A_i U^2\right) \tag{3}$$

$$Ct_i = 4a_i(1 - a_i) \tag{4}$$

The number of induction factor values in the induction factor output files is equal to the number of blade nodes used for the AeroDyn analysis. The name of the induction factor output files are named corresponding to the investigated turbine index. For example, *Induction_Turbine_2.txt* stores the induction factors for the turbine 2.

Turbine inflow mean velocity output files [m/s]

The inflow mean velocity for a turbine is the time and spatial averaged axial velocity across the rotor disk throughout the whole simulation time domain. For example, *Mean_U_Turbine_2.txt* stores the mean velocity for turbine 2, and the total number of these mean velocity output files is equal to the total number of the turbines simulated.

Turbulence intensity output files [%]

The turbulence intensity (TI) reflects the intensity of the mixing between the wake and ambient flow. If a turbine is located in the wake of an upstream turbine, the turbulence intensity at the downwind turbine is no longer equal to the ambient turbulence intensity. The value in the output file corresponds to the total inflow turbulence intensity of this turbine, which combines the contribution from the wake added turbulence, and the apparent turbulence which is due to the large scale wake meandering form the upwind turbines. Each turbine has its own turbulence intensity output file and the TI is scaled at the rotor plane. For example, *TI_Turbine_2.txt* stores the turbulence intensity at turbine 2.

Wake deficit velocity output files

When the wind flows past a wind turbine, the extraction of the axial momentum in the wind generates the wake deficit behind a turbine. The wake deficit velocity is one of the major outputs that DWM calculates. The wake is assumed to be axisymmetric, and DWM uses rotationally symmetric Navier-Stokes equations to resolve the wake velocity of each node in the (x,r) coordinate system. The space grid size is scaled by *ppR*, *DomainR* and *DomainX*. Due to the fact that wake’s change rate in the radial direction is larger, the grid size in the axial direction is as twice as large as the one in the radial direction.

The values of the wake velocity are between *0* and *1* since they are normalized by the ambient wind velocity. These wake velocity files are renamed corresponding to the investigated turbine index. For example, *WakeU_Turbine_2.txt* stores the wake velocity behind turbine 2. A MATLAB script file is included in the archive to post-process the wake deficit output, which generates the normalized wake velocity at a certain downstream plane as Fig. 6 presents below.

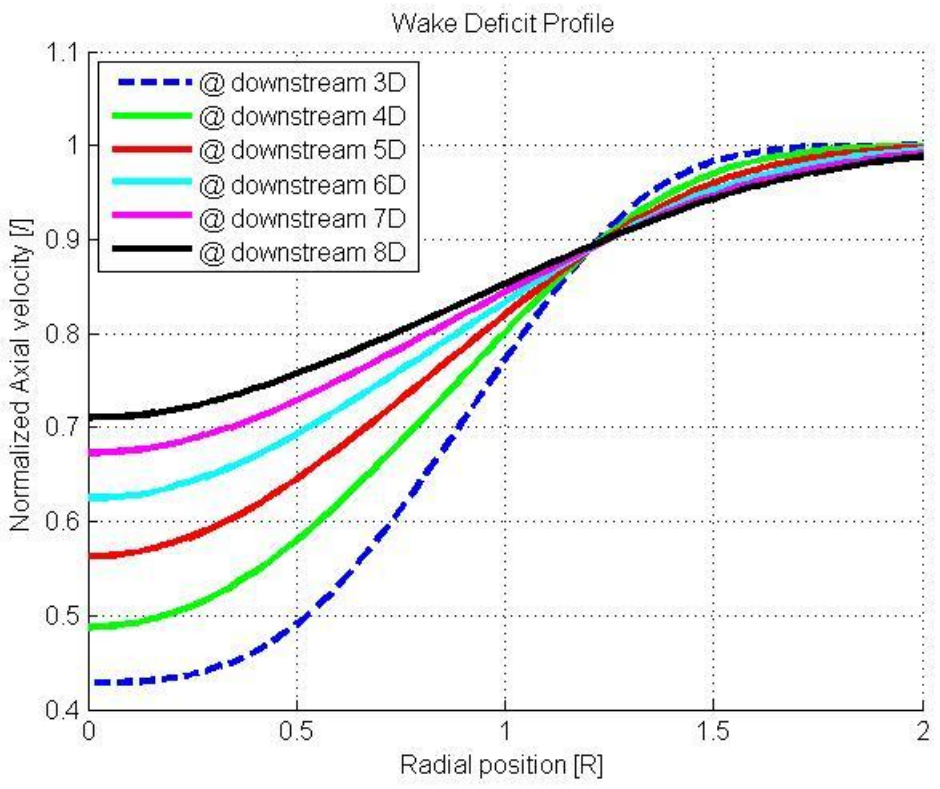


Figure 6. Example wake deficit profiles at various downstream locations plot.

Meandered wake center output files

Wake meandering is the term used to describe the large scale movement of the entire wake. Wake meandering is important because it might considerably increase extreme loads and fatigue loads, in particular yaw loads, on turbines downstream in wind farms, because the meandering causes the wake to be swept in and out of the rotor plane of downstream turbines. The wake center positions are expressed in the Cartesian coordinate system in *x-y-z* directions which is the same as the tower-base coordinate system used by FAST. At each time step, the current location (x, y, z) of the meandered wake center is given. The distance of the wake movement in x direction is determined based on the mean wind speed; the distances of the wake movement in y and z direction are determined by using a low-pass filter.

The meandered wake center output files are renamed corresponding to the investigated turbine index. For example, *WC_Turbine_2.txt* records the meandered wake center coordinates behind turbine 2 for all the simulation time steps. A MATLAB script file is included in the archive to post-process the meandered wake center coordinates output, which generates the meandered wake center coordinates at a certain time step. An example of the wake center coordinate in the vertical direction and lateral direction for different moving times are shown in Fig.7 and Fig.8 respectively.

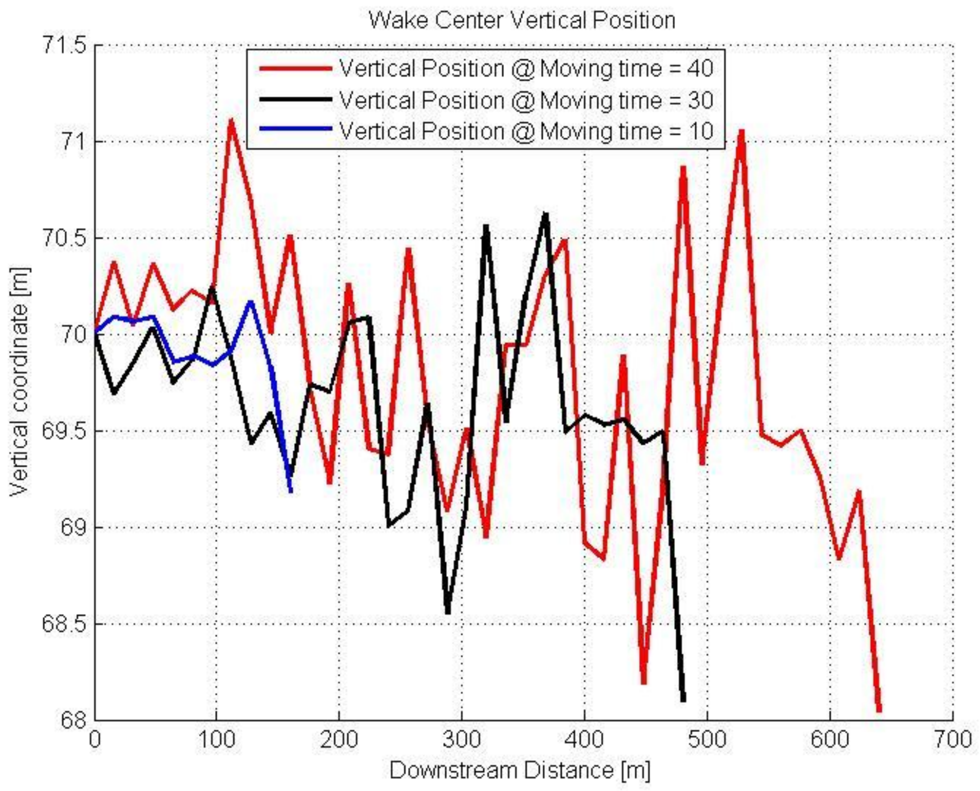


Figure 7. Example meandered wake center coordinates in vertical direction plot.

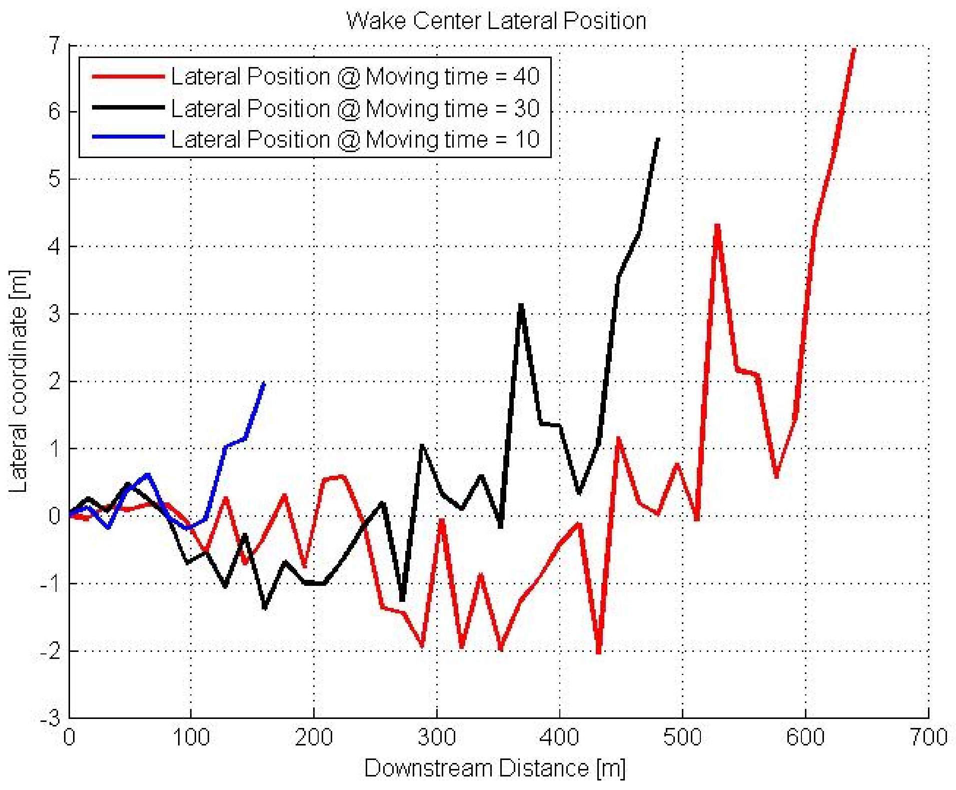


Figure 8. Example meandered wake center coordinates in lateral direction plot.

References

[1]Hao Y, Lackner M, Keck R, Lee S, Churchfield M, Moriarty P, “Implementing the Dynamic Wake Meandering Model in the NWTC Design Codes,” Dec 2013.

[2]Keck R, Veldkamp D, Madsen H, Larsen G, “Implementation of a Mixing Length Turbulence Formulation Into the Dynamic Wake Meandering Model,” *Journal of Solar Energy Engineering*, 134(2):021012-021012-13, 2012.

[3]Gunner C. Larsen, Helge Aa. Madsen, Ferhat Bingöl, Jakob Mann et al., “Dynamic wake meandering Modeling,” Risø-R-1607(EN), June 2007.

Appendix A: Sample DWM Input File

```
--------------------------------------------------------------------------------
------- DWM WIND FARM INPUT FILE ------------------------------------------------
DWM wind farm specification.
Implemented in the FAST8 with DWM module
---------------------- Wind Farm Parameters -------------------------------------
70    HubHt                             - The hub height (m)
40    RotorR                            - The Rotor radius (m)
4     NumWT                             - The total number of wind turbines (-)
8     Uambient                          - The ambient wind velocity (m/s)
6     TI                                - Ambient turbulence intensity (%)
50    ppR                               - The number of points per radius  (-)
5.0   Domain_R                          - Radial domain size (R) (should be larger than spacing*sin(theta) where theta is the turbine alignment angle)
36    Domain_X                          - Longitudinal domian size  (R) (should be larger than turbine spacing)
300   Meandering_simulation_time_length - The total simulation time steps in the meandering wake model for a fixed cross plane (-)
60    Meandering_Moving_time            - The total simulation time steps in the meandering wake model for a moving cross plane(-)
5     WFLowerBd                         - The height of the lower bound of the wind file (m)
180   Winddir                           - The ambient wind direction (degree)
"FAST_win32"  DWM.exe name           - The file rootname of the DWM-FAST program
XCoordinate     YCoordinate
0               10
0               20
0               30
0               40
```