



Development of FAST.Farm: A New Multiphysics Engineering Tool for Wind Farm Design and Analysis

Preprint

Jason Jonkman, Jennifer Annoni,
Greg Hayman, Bonnie Jonkman,
and Avi Purkayastha
National Renewable Energy Laboratory

*To be presented at AIAA SciTech 2017
Grapevine, Texas
January 9–13, 2017*

**NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Conference Paper
NREL/CP-5000-67528
January 2017

Contract No. DE-AC36-08GO28308

NOTICE

The submitted manuscript has been offered by an employee of the Alliance for Sustainable Energy, LLC (Alliance), a contractor of the US Government under Contract No. DE-AC36-08GO28308. Accordingly, the US Government and Alliance retain a nonexclusive royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for US Government purposes.

This report was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or any agency thereof.

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Available electronically at SciTech Connect <http://www.osti.gov/scitech>

Available for a processing fee to U.S. Department of Energy and its contractors, in paper, from:

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
OSTI <http://www.osti.gov>
Phone: 865.576.8401
Fax: 865.576.5728
Email: reports@osti.gov

Available for sale to the public, in paper, from:

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312
NTIS <http://www.ntis.gov>
Phone: 800.553.6847 or 703.605.6000
Fax: 703.605.6900
Email: orders@ntis.gov

Cover Photos by Dennis Schroeder: (left to right) NREL 26173, NREL 18302, NREL 19758, NREL 29642, NREL 19795.

NREL prints on paper that contains recycled content.

Development of FAST.Farm: A New Multiphysics Engineering Tool for Wind Farm Design and Analysis*

Jason Jonkman,[†] Jennifer Annoni,[‡] Greg Hayman,[§] Bonnie Jonkman,^{**} and Avi Purkayastha^{**}
National Renewable Energy Laboratory (NREL), Golden, Colorado, 80401

This paper presents the development of FAST.Farm, a new multiphysics tool applicable to engineering problems in research and industry involving wind farm performance and cost optimization that is needed to address the current underperformance, failures, and expenses plaguing the wind industry. Achieving wind cost-of-energy targets—which requires improvements in wind farm performance and reliability, together with reduced uncertainty and expenditures—has been eluded by the complicated nature of the wind farm design problem, especially the sophisticated interaction between atmospheric phenomena and wake dynamics and array effects. FAST.Farm aims to balance the need for accurate modeling of the relevant physics for predicting power performance and loads while maintaining low computational cost to support a highly iterative and probabilistic design process and system-wide optimization. FAST.Farm makes use of FAST to model the aero-hydro-servo-elastics of distinct turbines in the wind farm, and it is based on some of the principles of the Dynamic Wake Meandering (DWM) model, but avoids many of the limitations of existing DWM implementations.

Nomenclature

$a(r)$	= axial induction factor, distributed radially
C_{HWkDfl}^O , C_{HWkDfl}^{OY} , C_{HWkDfl}^x , and C_{HWkDfl}^{xY}	= calibrated parameters in the horizontal wake-deflection correction
$C_{NearWake}$	= calibrated parameter in the near-wake correction
$C_{WakeDiam}$	= calibrated parameter in the wake-diameter calculation
C_{vAmb}^{DMax} , C_{vAmb}^{DMin} , C_{vAmb}^{Exp} , and C_{vAmb}^{FMin}	= calibrated parameters in the eddy-viscosity filter function for ambient turbulence
C_{vShr}^{DMax} , C_{vShr}^{DMin} , C_{vShr}^{Exp} , and C_{vShr}^{FMin}	= calibrated parameters in the eddy-viscosity filter function for the shear layer
$AzimAvg C_i(r)$ and $FilteredAzimAvg C_i(r)$	= azimuthally averaged thrust-force coefficient (normal to a rotor disk), distributed radially, and its low-pass time-filtered value
D^{Rotor} and $Filtered D_{n_p}^{Rotor}$	= rotor diameter and its low-pass time-filtered value at wake plane n_p
$D_{n_p}^{Wake}$	= wake diameter at wake plane n_p
f_c	= cut-off (corner) frequency of the low-pass time filter
$\bar{f}_{n_b}(r)$	= aerodynamic applied loads distributed radially per unit length for blade n_b
$F_{vAmb}(x)$	= eddy-viscosity filter function associated with ambient turbulence

*The submitted manuscript has been offered by an employee of the Alliance for Sustainable Energy, LLC (Alliance), a contractor of the U.S. Government under Contract No. DE-AC36-08GO28308. Accordingly, the U.S. Government and Alliance retain a nonexclusive royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

[†]Senior Engineer, National Wind Technology Center (NWTC), 15013 Denver West Parkway, Golden, CO, 80401. AIAA Professional Member.

[‡]Postdoctoral Researcher, NWTC, 15013 Denver West Parkway, Golden, CO, 80401.

[§]Senior Engineer, NWTC, 15013 Denver West Parkway, Golden, CO, 80401.

^{**}Senior Scientist, NWTC, 15013 Denver West Parkway, Golden, CO, 80401.

- $F_{vShr}(x)$ = eddy-viscosity filter function associated with the shear layer
 I = three-by-three identify matrix
 k_{vAmb} = calibrated parameter for the influence of ambient turbulence in the eddy viscosity
 k_{vShr} = calibrated parameter for the influence of the shear layer in the eddy viscosity
 n = discrete-time-step counter
 N_b and n_b = number of rotor blades and blade counter
 N^{Wake} and n^{Wake} = number of wakes overlapping a given wind data point in the wind domain and wake counter
 $N_{n_p}^{Wind}$ and n^{Wind} = number of wind data points that reside within wake volume n_p and point counter
 N_p and n_p = number of wake planes and wake-plane counter
 N_r and n_r = number of radial nodes and radii counter
 N_t and n_t = number of wind turbines and turbine counter
 \vec{p}^{Hub} = global position of a rotor center
 $\vec{p}_{n_p}^{Plane}$ = global position of the center of wake plane n_p
 r and r^{Plane} = radius in the axisymmetric coordinate system
 \hat{r}^{Plane} = radial unit vector in the axisymmetric coordinate system
 TI_{Amb} and $FilteredTI_{Amb,n_p}$ = ambient turbulence intensity of the wind at a rotor and its low-pass time-filtered value
for wake plane n_p
 u^d = discrete-time inputs
 \vec{V}_{Amb}^{High} = ambient wind across a high-resolution wind domain around a turbine
 \vec{V}_{Amb}^{Low} = ambient wind across a low-resolution wind domain throughout the wind farm
 \vec{V}_{Dist}^{High} = disturbed wind (ambient plus wakes) across a high-resolution wind domain around a turbine
 \vec{V}_{Dist}^{Low} = disturbed wind (ambient plus wakes) across a low-resolution wind domain throughout the wind farm
 $\vec{V}_{n_p}^{Plane}$ = advection, deflection, and meandering velocity of wake plane n_p
 V_r = radial velocity in the axisymmetric coordinate system
 $V_{r,n_p}^{Wake}(r)$ = radial wake-velocity deficit at wake plane n_p , distributed radially
 $DiskAvgV_x^{Rel}$ and $FilteredDiskAvgV_x^{Rel}$ = rotor-disk-averaged relative wind speed (ambient plus wakes plus turbine motion), normal to the disk, and its low-pass time-filtered value
 V_x = axial velocity in the axisymmetric coordinate system
 $V_{x,n_p}^{Wake}(r)$ = axial wake-velocity deficit at wake plane n_p , distributed radially
 $DiskAvgV_x^{Wind}$ and $FilteredDiskAvgV_{x,n_p}^{Wind}$ = rotor-disk-averaged ambient wind speed, normal to the disk, and its low-pass time-filtered value at wake plane n_p
 x and $x_{n_p}^{Plane}$ = downwind distance from a rotor to wake plane n_p in the axisymmetric coordinate system
 $X, Y,$ and Z = inertial-frame coordinates, with Z directed vertically upward, opposite gravity, X directed horizontally nominally downwind (along the zero-degree wind direction), and Y directed horizontally transversely
 $\hat{X}, \hat{Y},$ and \hat{Z} = unit vectors of the inertial-frame coordinate system, parallel to the $X, Y,$ and Z coordinates
 x^d = discrete-time states
 $X^d(\)$ = discrete-time state functions
 \hat{x}^{Disk} = orientation of a rotor centerline
 $\hat{x}_{n_p}^{Plane}$ = orientation of wake plane n_p
 y^d = discrete-time outputs

$Y^d ()$	= discrete-time output functions
α	= low-pass time-filter parameter
Δt	= discrete time step (increment)
γ^{YawErr} and $\gamma_{n_p}^{Filtered YawErr}$	= nacelle-yaw error of a rotor and its low-pass time-filtered value at wake plane n_p
ν_T	= eddy viscosity
ρ	= air density

I. Introduction

WIND farms composed of many multimegawatt (MW) wind turbines are vastly underperforming, with power losses as high as 30% below expectations, higher-than-anticipated mechanical loads, premature component failures, and millions of dollars in lost revenue. Achieving wind energy cost parity with fossil fuels without tax subsidies will require improved wind farm performance, component reliability, and forecasting, together with reduced uncertainty, capital expenditures, and operation/maintenance costs. Attaining these goals is complicated by the multiphysics, multiscale, and multidisciplinary nature of the wind farm design problem. Physically, the performance and loading of a wind farm is driven by mesoscale to microscale interactions, atmospheric phenomena in the boundary layer including turbulence and stability, complex terrain effects, rotor wake dynamics and their interactions between wind turbines in an array, aero-elastic coupling, and individual wind turbine and farm-wide controllers. Furthermore, nonlinearities in the dynamics necessitate time-domain analyses.

To address these challenges, engineering tools are needed for wind farm design and analysis. Although a range of tools exist with varying fidelity, none of them meet the full wind farm analysis needs. On the low-fidelity side, the FLOW Redirection and Induction in Steady state (FLORIS) model¹⁻⁴—used for wind farm performance estimation and for wind farm controls design—is based on the well-known Jensen model, which considers uniform wakes and steady-state wind farm flow based on averaged rotor thrust, but extended for improved accuracy in controls applications to account for multiple wake zones and wake deflection in yawed flow. Steady-state models, however, do not account for dynamics and cannot predict wind turbine dynamic loads. On the midfidelity side, the Dynamic Wake Meandering (DWM) model⁵⁻¹³ combines three distinct submodels to predict wind farm performance and loads with dynamics: 1) a steady-state wake-deficit model based on the axisymmetric thin-shear-layer approximation of the Navier-Stokes equation with an eddy-viscosity model for turbulence closure,¹⁴ 2) a wake-transport model that transverses the wake deficit over time as a result of large-scale atmospheric turbulence, as a passive tracer, and 3) a wake-turbulence model to account for the added turbulence introduced by the wake. The DWM model of Hao¹⁰⁻¹³ has been coupled to the National Renewable Energy Laboratory’s (NREL’s) FAST wind turbine engineering tool¹⁵ to predict the loads in a serial manner, meaning that a wind farm is treated by first modeling the most-upstream wind turbine, then modeling its wake influence on the next downwind wind turbine, and so on, in a sequential fashion throughout the wind farm. Although this implementation makes it possible to model an entire wind farm on a standard personal computer (PC), the serial solution limits its accuracy and minimizes its usefulness for wind farm controls design and optimization because two-way interactions between turbines are not considered. On the high-fidelity modeling (HFM) side, NREL’s Simulator fOr Wind Farm Applications (SOWFA),¹⁶⁻¹⁸ which couples large-eddy simulation (LES) with FAST to predict wind farm performance and loads—see Figure 1 for an example result—relies on high-performance computing (HPC) with high resource demand. SOWFA directly resolves the larger energy-containing scales of turbulence and wake dynamics. These computations require up to 1-billion mesh cells and days to weeks of computing time per 10-minute simulation, relegating the application of SOWFA to very few simulations, which is impractical to engineering design.

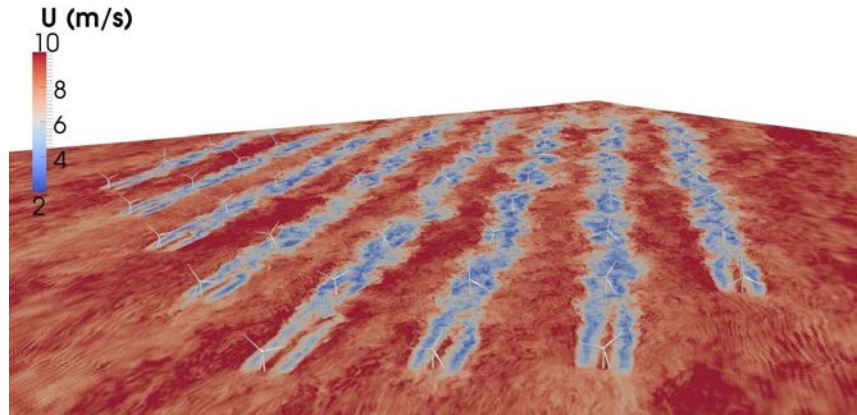


Figure 1. Example SOWFA simulation showing the interaction of wakes with atmospheric turbulence in a wind farm.

NREL has initiated the development of a new multiphysics engineering tool—named FAST.Farm¹⁹—that aims to balance the need for accurate modeling of the relevant physics for predicting power performance and loads while maintaining low computational cost to support a highly iterative and probabilistic design process and system-wide optimization. FAST.Farm is DWM-like, in that it makes use of FAST to model the aero-hydro-servo-elastics of distinct turbines in the wind farm and relies on some of the DWM modeling principles. However, FAST.Farm avoids many of the limitations of existing DWM implementations, includes the controls capability of FLORIS, and functions more like SOWFA. Insight from well-validated SOWFA simulations is being used to support the development and parameter calibration of FAST.Farm. Potential future applications for FAST.Farm include reducing wind farm underperformance and loads uncertainty, developing wind farm controls to enhance the operation of existing wind farms, optimizing the siting and topology of new wind farms, and innovating the design of wind turbines for the wind farm environment.

The implementation of FAST.Farm in source code is nearing completion. This paper presents the development of FAST.Farm, as well as the concepts and mathematical background needed to understand and apply it correctly. Model calibration, verification, validation, and applications will be presented in future work.

II. Dynamic Wake Meandering Principles and Limitations Being Addressed

The main idea behind the DWM model is to capture key wake features pertinent to accurate prediction of wind farm power performance and wind turbine loads, including the wake-deficit evolution (important for performance) and the wake meandering and wake-added turbulence (important for loads). Although fundamental laws of physics are applied, appropriate simplifications have been made to minimize the computational expense, and HFM solutions are used to inform and calibrate the submodels. In the DWM model, the wake-flow processes are treated via the “splitting of scales,” in which small turbulent eddies (less than two diameters) affect wake-deficit evolution and large turbulent eddies (greater than two diameters) affect wake meandering.

The presence of thrust from the wind turbine rotor causes the wind speed to decrease and the pressure to increase just upwind of the rotor. In the near-wake region just downwind of the rotor—illustrated in Figure 2—coherent vortices break down, the pressure recovers to free stream, the wind speed decreases further, and the wake expands radially. In the far-wake region further downwind, the wake deficit is approximately Gaussian and recovers to free stream due to the turbulent transfer of momentum into the wake from the ambient wind across the wake shear layer. This flow-speed reduction and gradual recovery to free stream is known as the wake-deficit evolution. In most DWM implementations, the wake-deficit evolution is modeled via the thin shear-layer approximation of the Reynolds-averaged Navier-Stokes equations under quasi-steady-state conditions in axisymmetric coordinates—illustrated in Figure 3. Turbulence closure is captured by using an eddy-viscosity formulation, dependent on small turbulent eddies. This wake-deficit evolution solution is only valid in the far wake (which is most important for wind farm analysis because wind turbines are not typically spaced closely). But because the wake-deficit evolution solution begins at the rotor, a near-wake correction is applied at the inlet boundary condition to improve the accuracy of the far-wake solution.

Wake meandering is the large-scale movement of the wake deficit transported by large turbulent eddies. This wake-meandering process is treated pragmatically in DWM⁵ by modeling the meandering as a passive tracer, which transfers the wake deficit transversely (horizontally and vertically) to a moving frame of reference (MFoR)—as illustrated in Figure 4—based on the ambient wind (including large turbulent eddies) spatially averaged across planes of the wake.

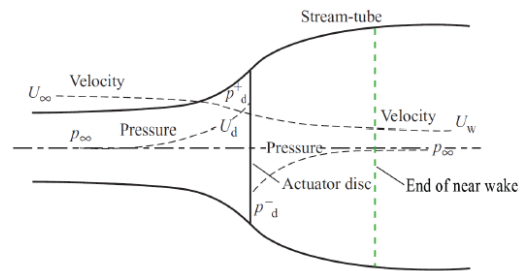


Figure 2. Near-wake region.¹¹

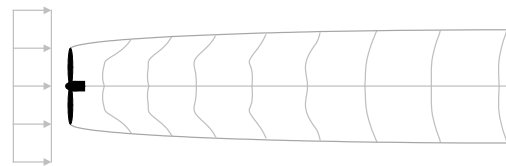


Figure 3. Axisymmetric wake-deficit evolution.

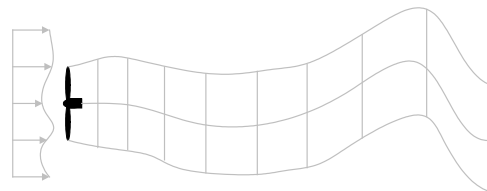


Figure 4. Wake meandering.

Wake-added turbulence is the additional small-scale turbulence generated from the turbulent mixing in the wake. It is often modeled in DWM by scaling up the background (undisturbed) turbulence.

Several variations of DWM have been implemented, for example, by the Technical University of Denmark (DTU)⁵⁻⁹ and the University of Massachusetts.¹⁰⁻¹³ Although the exact limitations of existing DWM implementations depend on the implementation, specific limitations that are addressed in developing FAST.Farm are summarized in Table 1 and are discussed where appropriate in the next section.

Table 1. Limitations of DWM being addressed by FAST.Farm.

Limitation	Solution/Innovation
<ul style="list-style-type: none"> The ambient wind is solved per individual rotor and generated synthetically based on the Taylor’s frozen-turbulence assumption, not coherent across the wind farm or based on mesoscale conditions or local terrain. 	<ul style="list-style-type: none"> Compute the ambient wind farm-wide from the Atmospheric Boundary Layer Solver (ABLSolver) precursor to SOWFA (or equivalent).
<ul style="list-style-type: none"> There is no treatment of a wind farm super controller. 	<ul style="list-style-type: none"> Include a wind farm super controller.
<ul style="list-style-type: none"> The wake advects at the mean ambient wind speed, not accelerating from near wake to far wake or affected by local flow conditions. 	<ul style="list-style-type: none"> Advect the wake based on the local spatially averaged ambient wind speed and wake deficit.
<ul style="list-style-type: none"> The wake deficit is not distorted by inflow skew (i.e., when looking downwind, the wake looks circular, not elliptical). The wake centerline is not deflected by inflow skew. 	<ul style="list-style-type: none"> Solve the wake deficit in planes parallel to rotor disk. Deflect the wake centerline based on the inflow skew.
<ul style="list-style-type: none"> The wake deficit and centerline are based only on mean conditions, not updated for transients in inflow, turbine control, or wind turbine motion (the latter is especially important for floating offshore wind turbines). 	<ul style="list-style-type: none"> Update the wake deficit and centerline based on low-pass-filtered inflow, wind turbine control, and wind turbine motion.
<ul style="list-style-type: none"> The wind turbine and wake dynamics are solved individually or serially, not considering two-way wake-merging interactions. The wake impingement is based only on the strongest wake deficit—not considering cumulative effects from multiple upwind wind turbines—and/or the wake impingement approach is treated differently below and above rated wind speed (i.e., a discrete change). There is no method available to calculate the disturbed wind in zones of wake overlap. 	<ul style="list-style-type: none"> Solve the individual wind turbine and wake dynamics in parallel on multiple cores (or an HPC), interconnected through a message-passing interface (MPI). Allow the wake merging to influence wake dynamics. Calculate the wake deficits of downwind wind turbines dependent on the impingement of wakes from upwind wind turbines. Superimpose the wake deficits in the axial direction based on the root-sum-square (RSS) method.
<ul style="list-style-type: none"> The wakes meander laterally, but not axially. 	<ul style="list-style-type: none"> Meander the wakes both laterally and axially.

III. FAST.Farm Development

FAST.Farm is a nonlinear time-domain multiphysics engineering tool composed of multiple submodels, each representing different physics domains of the wind farm. FAST.Farm is implemented as open-source software that follows the programming requirements of the FAST modularization framework,²⁰ whereby the submodels are implemented as modules interconnected through a driver code. The submodel hierarchy of FAST.Farm is illustrated in Figure 5. Wake advection, deflection, and meandering; near-wake correction; and wake-deficit increment are submodels of the wake-dynamics (WD) model, implemented in a single module (called WD). Ambient wind and wake merging are submodels of the ambient wind and array effects (AWAE) model, implemented in a single module (called AWAE). The super controller (SC) and FAST (F) are separate modules, meaning that FAST.Farm has four modules (SC, F, WD, and AWAE) and one driver. There are multiple instances of the F and WD modules—one instance for each wind turbine/rotor. To handle the memory requirements and parallelization, the entire wind farm is solved in parallel on multiple cores (or an HPC); but the solution will require only a modest HPC resource (a bit more than two cores per turbine in the wind farm)—not out of reach for industry use. Each submodel/module is described in the subsections below.

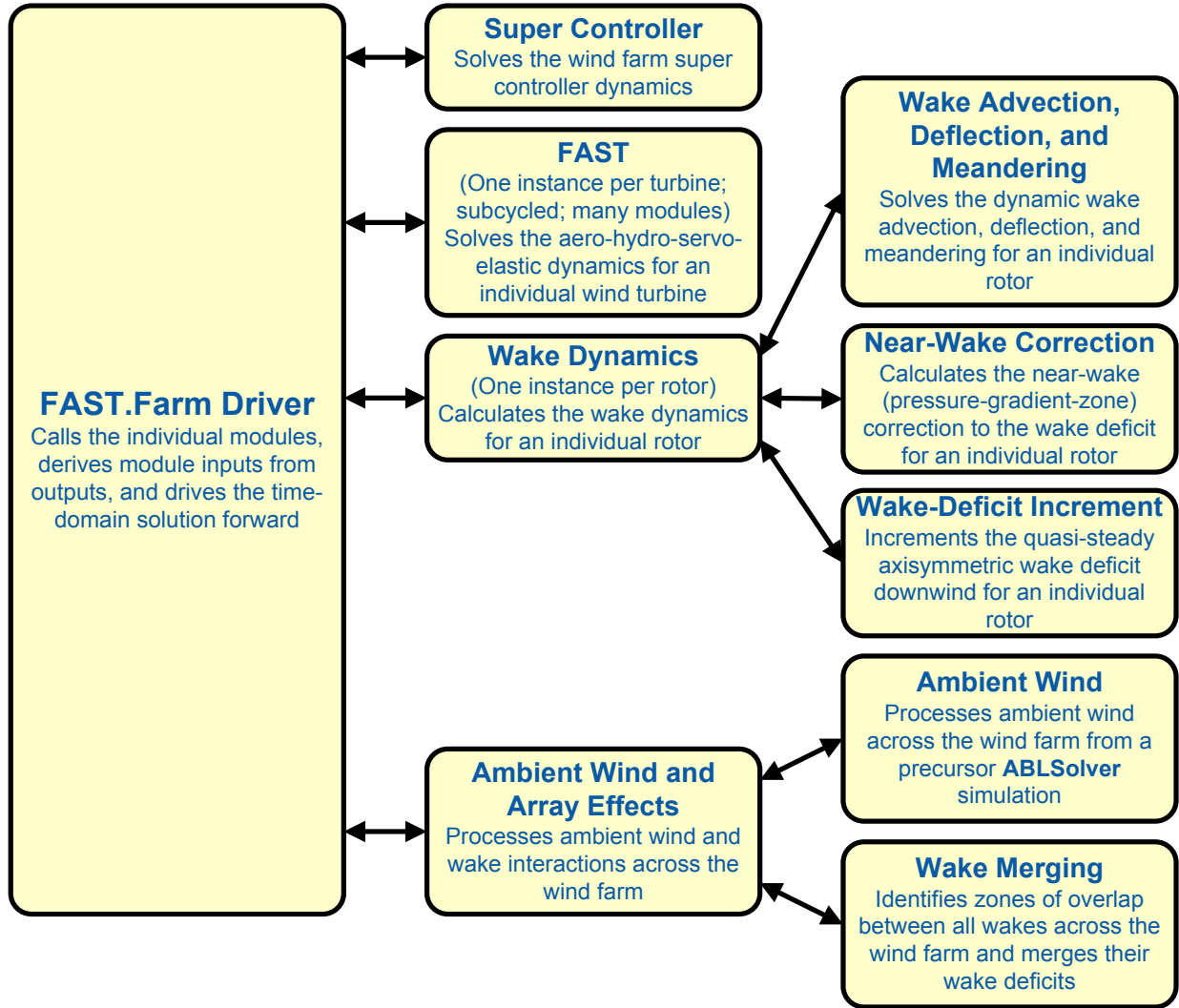


Figure 5. FAST.Farm submodel hierarchy.

A. FAST.Farm Driver

The FAST.Farm driver, also known as the “glue code,” is the code that couples individual modules together and drives the overall time-domain solution forward. Additionally, the FAST.Farm driver reads an input file of simulation parameters, checks the validity of these parameters, initializes the modules, writes results to a file, and releases memory at the end of the simulation.

To simplify the coupling algorithm in the FAST.Farm driver and ensure computational efficiency, all module states (x^d), inputs (u^d), outputs (y^d), and functions (X^d for state updates and Y^d for outputs) in FAST.Farm are expressed in discrete time, $t = n\Delta t$, where t is time, n is the discrete-time-step counter, and Δt is the user-specified discrete time step (increment). Thus, the most general form of a module in FAST.Farm is simpler than that permitted by the FAST modularization framework,²⁰ represented mathematically as^{††}

$$x^d[n+1] = X^d(x^d[n], u^d[n], n), \quad (1a)$$

$$y^d[n] = Y^d(x^d[n], u^d[n], n). \quad (1b)$$

^{††} x^d and X^d are identical to what is described in Ref. 18. u^d , y^d , and Y^d are identical to u , y , and Y from Ref. 18, but are only evaluated in discrete time, $t = n\Delta t$, and so, are marked here with superscript d .

Moreover, the SC, F, and WD modules do not have direct feedthrough of input to output, meaning that their output functions simplify to $y^d[n] = Y^d(x^d[n], n)$, and the AWAE module does not have states, reducing the module to a feed-forward-only system and a module form that simplifies to $y^d[n] = Y^d(u^d[n], n)$. (The ability of module F to be written in the above form is explained in Section III.C.) For functions in this paper, square brackets $[]$ denote discrete functions and round parentheses $()$ denote continuous functions; the brackets/parentheses are dropped when implied. The states, inputs, and outputs of each of the FAST.Farm modules (SC, F, WD, and AWAE) are listed in Table 2 and explained further in the sections below.

After initialization, within each time step, the states of each module (SC, F, and WD) are updated (from time t to time $t + \Delta t$, or equivalently, n to $n + 1$), time is incremented, and the module outputs are calculated and transferred as inputs to other modules. Because of the simplifications in form, the state updates of each module can be solved in parallel, the output-to-input transfer does not require a large nonlinear solve, and overall correction steps of the solution are not needed, which is a major simplification of the coupling algorithm used within FAST.^{21, 22} Furthermore, the output calculations of modules SC, F, and WD can also be parallelized, followed then by the output calculation of module AWAE. Parallelization has been implemented in FAST.Farm through a message-passing interface (MPI).

Because of the small timescales involved and sophisticated physics, the FAST submodel (module F) is the computationally slowest of the FAST.Farm modules. And because the output calculation of module AWAE is the only major calculation that cannot be solved in parallel to FAST, the FAST.Farm solution executes only slightly slower than standalone FAST simulations—computationally inexpensive enough to run the many simulations necessary for wind turbine/farm design and analysis.

B. Super Controller (Module SC)

Wind-farm-wide super controllers have the potential to achieve global benefit of improving overall power performance and reducing turbine loads, based on modifying wake deficits through variations in blade pitch or generator torque and/or redirecting (steering) wakes through variations in nacelle yaw or tilt, as illustrated in Figure 6.

The SC module of FAST.Farm—essentially identical to the super controller available in SOWFA¹⁸—allows the user of FAST.Farm to implement their own wind-farm-wide control logic in discrete time and without direct feedthrough of input to output—perhaps developed through the application of FLORIS.⁴ The inputs to module SC are the global (e.g., wind) measurements (output from module AWAE) and commands or measurements from individual turbine controllers (output from module F), and the outputs of module SC are the controller commands globally and to individual turbine controllers (inputs to module F).

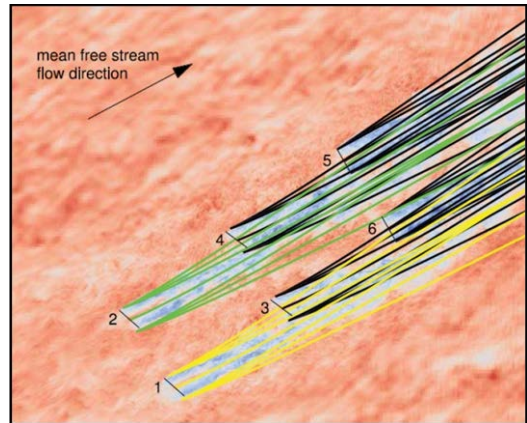


Figure 6. Nacelle-yaw control used to redirect wakes away from downwind wind turbines.²

C. FAST (Module F)

FAST.Farm makes use of FAST version 8¹⁵ to model the dynamics (loads and motions) of distinct turbines in the wind farm. FAST captures the environmental excitations (wind inflow, and for offshore systems, waves, current, and ice) and coupled system response of the full system (the rotor, drivetrain, nacelle, tower, controller, and for offshore systems, the substructure and station-keeping system). FAST itself is an interconnection of various modules, each corresponding to different physical domains of the coupled aero-hydro-servo-elastic solution. The details of the FAST solution are outside the scope of this paper, but can be found in Ref. 15 and associated references.

The F module of FAST.Farm is a wrapper that enables the coupling of FAST to FAST.Farm—similar to the FAST wrapper available in SOWFA, but with different inputs and outputs (described below). This wrapper also controls subcycling of the FAST state updates; for accuracy and numerical stability reasons, the FAST time step is typically much smaller than that required of FAST.Farm (the timescales solved within FAST are much smaller than those solved within FAST.Farm). There is one instance of module F for each wind turbine.

Table 2. Module states, inputs, and outputs in FAST.Farm.

Module	States (Discrete-Time)	Inputs	Outputs
Super Controller (SC)	<ul style="list-style-type: none"> User-defined 	<ul style="list-style-type: none"> Global measurements. Commands/measurements from individual turbine controllers. 	<ul style="list-style-type: none"> Global controller commands. Commands to individual turbine controllers.
FAST (F)	<ul style="list-style-type: none"> None in the FAST wrapper, but there are many states internal to FAST. 	<ul style="list-style-type: none"> Global controller commands. Commands to the individual turbine controller. \vec{V}_{Dist}^{High} 	<ul style="list-style-type: none"> Commands/measurements from the individual turbine controller. \hat{x}^{Disk} \vec{p}^{Hub} D^{Rotor} γ^{YawErr} $DiskAvg V_x^{Rel}$ $AzimAvg C_t(r)$
Wake Dynamics (WD)	<ul style="list-style-type: none"> $FilteredDiskAvg V_x^{Rel}$ $FilteredAzimAvg C_t(r)$ For $0 \leq n_p \leq N_p - 1$: <ul style="list-style-type: none"> $Filtered D_{n_p}^{Rotor}$ $Filtered \gamma_{n_p}^{YawErr}$ $FilteredDiskAvg V_{x_{n_p}}^{Wind}$ $Filtered TI_{Amb_{n_p}}$ $x_{n_p}^{Plane}$ $\hat{x}_{n_p}^{Plane}$ $\vec{p}_{n_p}^{Plane}$ $V_{x_{n_p}}^{Wake}(r)$ $V_{r_{n_p}}^{Wake}(r)$ 	<ul style="list-style-type: none"> \hat{x}^{Disk} \vec{p}^{Hub} D^{Rotor} γ^{YawErr} $DiskAvg V_x^{Rel}$ $AzimAvg C_t(r)$ $\vec{V}_{n_p}^{Plane}$ for $0 \leq n_p \leq N_p - 1$ $DiskAvg V_x^{Wind}$ TI_{Amb} 	<ul style="list-style-type: none"> For $0 \leq n_p \leq N_p - 1$: <ul style="list-style-type: none"> $\hat{x}_{n_p}^{Plane}$ $\vec{p}_{n_p}^{Plane}$ $V_{x_{n_p}}^{Wake}(r)$ $V_{r_{n_p}}^{Wake}(r)$ $D_{n_p}^{Wake}$
Ambient Wind and Array Effects (AWAE)	<ul style="list-style-type: none"> None. 	<ul style="list-style-type: none"> For each turbine and $0 \leq n_p \leq N_p - 1$: <ul style="list-style-type: none"> $\hat{x}_{n_p}^{Plane}$ $\vec{p}_{n_p}^{Plane}$ $V_{x_{n_p}}^{Wake}(r)$ $V_{r_{n_p}}^{Wake}(r)$ $D_{n_p}^{Wake}$ 	<ul style="list-style-type: none"> For each turbine: <ul style="list-style-type: none"> \vec{V}_{Dist}^{High} $\vec{V}_{n_p}^{Plane}$ for $0 \leq n_p \leq N_p - 1$ $DiskAvg V_x^{Wind}$ TI_{Amb}

FAST itself has various modules with different inputs, outputs, states, and parameters—including continuous-time, algebraic, and other (e.g., logical) states. However, for the purposes of coupling FAST to FAST.Farm, module F functions in discrete time and without direct feedthrough of input to output. This is achieved by calling module F at the rate dictated by the FAST.Farm time step, Δt , and by introducing a one-time-step (Δt) delay of the output relative to the input; this one-time-step delay is not expected to be problematic because of the slow timescales solved within FAST.Farm.

At initialization, the number of wind turbines, N_t (with n_t the turbine counter such that $1 \leq n_t \leq N_t$), their FAST primary input file, and their origin (i.e., the intersection of the undeflected tower centerline and the ground, or mean

sea level for offshore systems) in the global X - Y - Z inertial-frame coordinate system are specified by the user of FAST.Farm. The global inertial-frame coordinate system is defined with Z directed vertically upward (opposite gravity), X directed horizontally nominally downwind (along the zero-degree wind direction), and Y directed horizontally transversely.

The global and turbine-dependent commands from the super controller (outputs from module SC) are used as input to module F to enable the individual turbine controller to be guided by wind-farm-level effects; likewise, the turbine-dependent commands or measurements are output from module F for access by the super controller (inputs to module SC).

Module F also uses the disturbed wind (ambient plus wakes) across a high-resolution wind domain (in both time and space) around the turbine (output from module AWAE—see Section III.E for more information), \vec{V}_{Dist}^{High} , as input, to ensure that the individual turbine loads and response calculated by FAST are accurately driven by flow through the wind farm, including wake and array effects. Spatially, the high-resolution wind domain must be large enough to encompass yawing of the rotor, blade deflection, and motion of the support structure (the latter is especially important for floating offshore wind turbines). FAST uses a four-dimensional (three space dimensions plus one time dimension) interpolation to determine the wind local to its analysis nodes.

Module F computes several outputs needed for calculating wake dynamics (inputs to module WD). These include the orientation of the rotor centerline, \hat{x}^{Disk} ; the global position of the rotor center, \vec{p}^{Hub} ; the rotor diameter, D^{Rotor} ; the nacelle-yaw error of the rotor, γ^{YawErr} ; the rotor-disk-averaged relative wind speed (ambient plus wakes plus turbine motion), normal to the disk, $^{DiskAvg}V_x^{Rel}$; and the azimuthally averaged thrust-force coefficient (normal to the rotor disk), distributed radially, $^{AzimAvg}C_t(r)$, where r is the radius. In this paper, an over arrow ($\vec{\cdot}$) denotes a three-component vector and a hat ($\hat{\cdot}$) denotes a three-component unit vector. For clarity in this paper, (r) is used to denote radial dependence as a continuous function, even though the radial dependence is stored/computed on a discrete radial finite-difference grid within FAST.Farm. All of these variables except γ^{YawErr} and $^{AzimAvg}C_t(r)$ were computed within FAST before the development of FAST.Farm. γ^{YawErr} is defined as the angle about global Z from the rotor centerline to the rotor-disk-averaged relative wind velocity (ambient plus wakes plus turbine motion), both projected onto the horizontal global X - Y plane—see Figure 7 for an illustration. $^{AzimAvg}C_t(r)$ is computed by

$$^{AzimAvg}C_t(r) = \frac{\sum_{n_b=1}^{N_b} \{\hat{x}^{Disk}\}^T \vec{f}_{n_b}(r)}{\frac{1}{2} \rho 2\pi r \left(^{DiskAvg}V_x^{Rel} \right)^2}, \quad (2)$$

where N_b is the number of rotor blades (with n_b the blade counter such that $1 \leq n_b \leq N_b$), $\{\cdot\}^T$ denotes a vector transpose, ρ is the air density, and $\vec{f}_{n_b}(r)$ are the aerodynamic applied loads distributed per unit length along a line extending radially outward in the plane of the rotor disk for blade n_b , which are derived using the Line2-to-Line2 mesh-mapping algorithm of FAST^{21,22} to transfer the aerodynamic applied loads distributed per unit length along the deflected/curved blade as calculated within FAST. The numerator of Eq. (2) is the aerodynamic applied loads distributed per unit length projected normal to the rotor disk, i.e., the radially dependent thrust force, and the denominator is the normalizing factor for the radially dependent thrust coefficient, including the circumference at the given radius, $2\pi r$, and the dynamic pressure of the rotor-disk-averaged relative wind speed, $\frac{1}{2} \rho \left(^{DiskAvg}V_x^{Rel} \right)^2$.

D. Wake Dynamics (Module WD)

The WD module of FAST.Farm calculates wake dynamics for an individual rotor—including wake advection, deflection, and meandering; a near-wake correction, which treats the near-wake (pressure-gradient zone) correction to the wake deficit; and a wake-deficit increment, which increments the quasi-steady-state axisymmetric wake deficit nominally downwind. Each submodel is described in the subsections below. There is one instance of the module WD for each rotor.

The wake-dynamics calculations involve many user-specified parameters that are dependent, for example, on turbine operation or atmospheric conditions that can be calibrated through HFM, e.g., by running SOWFA (or

equivalent) as a preprocessor. Default values are being derived for each calibrated parameter based on SOWFA simulations, but these can be overwritten by the user of FAST.Farm.

The wake-deficit evolution is solved in discrete time on an axisymmetric finite-difference grid consisting of a fixed number of wake planes, N_p (with n_p the wake-plane counter such that $0 \leq n_p \leq N_p - 1$), each with a fixed radial grid of nodes (because the wake deficit is assumed to be axisymmetric, the radial finite-difference grid can be considered a plane). A wake plane can be thought of as a cross section of the wake wherein the wake deficit is calculated.

Module WD uses as input \hat{x}^{Disk} , \bar{p}^{Hub} , D^{Rotor} , γ^{YawErr} , $^{DiskAvg}V_x^{Rel}$, and $^{AzimAvg}C_t(r)$ as computed by FAST for an individual turbine (output by module F), as well as the advection, deflection, and meandering velocity of the wake planes for the rotor, $\vec{V}_{n_p}^{Plane}$ for $0 \leq n_p \leq N_p - 1$, which are calculated by spatial averaging of the disturbed wind (ambient plus wakes); the rotor-disk-averaged ambient wind speed, normal to the disk, $^{DiskAvg}V_x^{Wind}$; and the ambient turbulence intensity of the wind at the rotor, TI_{Amb} (output from module AWAE—see Section III.E for more information).

Module WD computes several outputs needed for the calculation of disturbed wind (ambient plus wakes) (input to module AWAE). These include the orientations of the wake planes (defined using the unit vectors normal to each plane), i.e., the orientation of the wake-plane centerline, $\hat{x}_{n_p}^{Plane}$; the global positions of the centers of the wake planes, $\vec{p}_{n_p}^{Plane}$; the axial and radial wake-velocity deficits at the wake planes, distributed radially, $V_{x_{n_p}}^{Wake}(r)$ and $V_{r_{n_p}}^{Wake}(r)$; and the wake diameters at the wake planes, $D_{n_p}^{Wake}$ (each for $0 \leq n_p \leq N_p - 1$).

1. Wake Advection, Deflection, and Meandering

By simple extensions to the passive tracer solution for transverse (horizontal and vertical) wake meandering, the wake-dynamics solution in FAST.Farm is extended to account for wake deflection, as illustrated in Figure 7, and wake advection, as illustrated in Figure 8, among other physical improvements. That is, 1) the velocity of each wake plane, $\vec{V}_{n_p}^{Plane}$ for $0 \leq n_p \leq N_p - 1$, is calculated by

spatially averaging the disturbed wind (ambient plus wakes)—see Section III.E—instead of the ambient wind; 2) the wake planes are oriented with the rotor centerline instead of the wind direction; and 3) the local conditions at the rotor, as input to module WD, are low-pass time filtered to account for transients in inflow, turbine control, and/or turbine motion instead of considering time-averaged conditions. With these extensions, the passive tracer solution enables 1) the wake centerline to deflect based on inflow skew (because in skewed inflow, the wake deficit normal to the disk introduces a velocity component that is not parallel to the ambient flow), 2) the wake to accelerate from near wake to far wake (because the wake deficits are stronger in the near wake and weaken downwind), 3) the wake-deficit evolution to change based on conditions at the rotor (because low-pass time filtered, instead of time-averaged, conditions are used), 4) the wake to meander axially in addition to transversely (because local axial winds are considered), and 5) the wake to look elliptical in skewed flow when looking downwind (circular when looking down the rotor centerline). Low-pass time filtering is important for item 3) because the wake reacts slowly to changes in local conditions at the rotor

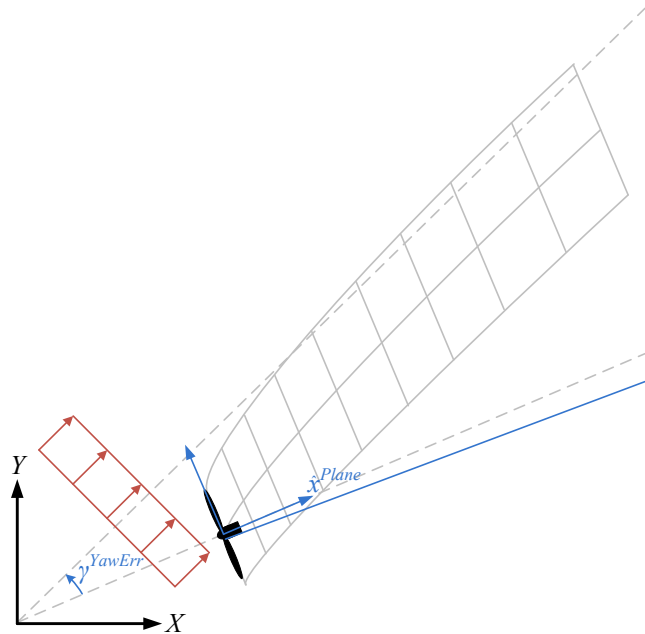


Figure 7. Wake deflection resulting from inflow skew, including a horizontal wake-deflection correction. The lower dashed line represents the rotor centerline, the upper dashed line represents the wind direction, and the solid blue line represents the horizontal wake-deflection correction (offset from the rotor centerline).

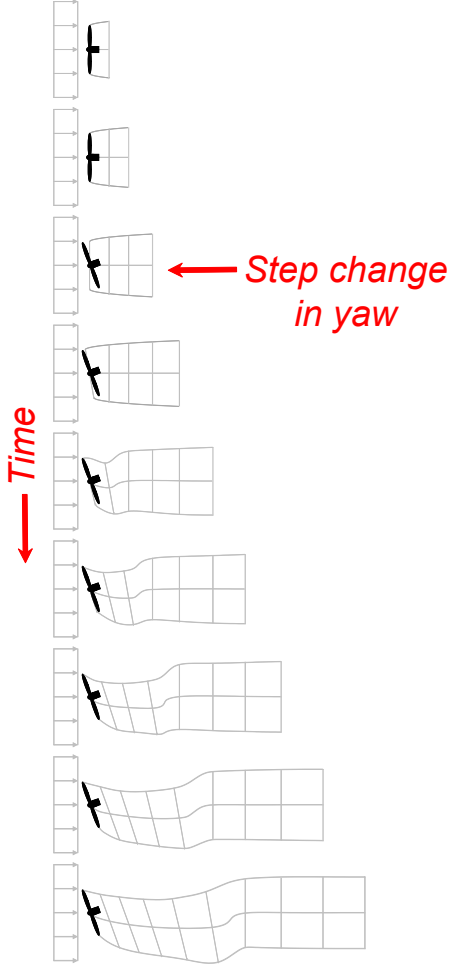


Figure 8. Wake advection.

and because the wake evolution is treated in a quasi-steady-state fashion. Furthermore, a horizontally asymmetric correction to the wake deflection from item 1)—i.e., a correction to the wake deflection resulting from $\vec{v}_{n_p}^{Plane}$, which physically results from the combination of wake rotation and shear not modeled directly in module WD—is accounted for (see Figure 7 for an illustration). This horizontal wake-deflection correction is a simple linear correction (with slope and offset), similar to the correction implemented in the wake model of FLORIS,⁴ and it is important for accurate modeling of nacelle-yaw-based wake-redirection (wake-steering) wind farm control.

Mathematically, the low-pass time filter is implemented using a recursive, single-pole filter with exponential smoothing.²³ The discrete-time recursion (difference) equation for this filter is²⁴

$$x^d [n+1] = x^d [n] \alpha + u^d [n] (1 - \alpha), \quad (3a)$$

where x^d is a discrete-time state storing the low-pass time-filtered value of input u^d and $\alpha = e^{-2\pi M f_c}$ is the low-pass time-filter parameter, with a value between 0 (minimum filtering) and 1 (maximum filtering) (exclusive), where f_c is the user-specified cut-off (corner) frequency (the time constant of the low-pass time filter is $\frac{1}{f_c}$).

To be consistent with the quasi-steady-state treatment of the wake-deficit evolution (see Section III.D.3), the conditions at the rotor are maintained as fixed states of a wake plane as the plane propagates downstream

$$x_{n_p}^d [n+1] = x_{n_p-I}^d [n] \quad \text{for } (1 \leq n_p \leq N_p - 1), \quad (3b)$$

where subscript n_p is used to denote the state associated with wake-plane n_p . In this paper, subscripts are dropped when implied, e.g., Eq. (3a) is not shown with subscript n_p , but applies at the rotor disk, where $n_p = 0$.

Equations (3a) and (3b) apply directly to the WD module inputs of D^{Rotor} , γ^{YawErr} , $\overset{DiskAvg}{V}_x^{Wind}$, and $\overset{TI}{A}_{mb}$; the associated states are $\overset{Filtered}{D}_{n_p}^{Rotor}$, $\overset{Filtered}{\gamma}_{n_p}^{YawErr}$, $\overset{Filtered}{\overset{DiskAvg}{V}}_{x_{n_p}}^{Wind}$, and $\overset{Filtered}{\overset{TI}{A}}_{mb_{n_p}}$, respectively (each for $0 \leq n_p \leq N_p - 1$). (Variations in the rotor diameter, D^{Rotor} , e.g., as a result of blade deflection, are likely small; but this input to module WD is treated the same as other inputs for consistency.) WD module inputs $\overset{DiskAvg}{V}_x^{Rel}$ and $\overset{AzimAvg}{C}_t(r)$ are needed for the boundary condition at the rotor, but are not otherwise needed in the wake-deficit evolution calculation and so do not need to be propagated downstream with the wake planes. So although Eq. (3a) applies to these inputs, Eq. (3b) does not; the associated states are $\overset{Filtered}{\overset{DiskAvg}{V}}_x^{Rel}$ and $\overset{Filtered}{\overset{AzimAvg}{C}}_t(r)$. Equation (3) applies in a modified form to WD module inputs \hat{x}^{Disk} and \vec{p}^{Hub} to derive the state associated with the downwind distance from the rotor to each wake plane in the axisymmetric coordinate system, $x_{n_p}^{Plane}$, and the states and outputs associated with the orientations of the wake planes, normal to the planes, $\hat{x}_{n_p}^{Plane}$, and the global center positions of the wake planes, $\vec{p}_{n_p}^{Plane}$, (each for $0 \leq n_p \leq N_p - 1$) as follows:

$$\hat{x}_{n_p}^{Plane} [n+1] = \begin{cases} \frac{\hat{x}_{n_p}^{Plane} [n] \alpha + \hat{x}^{Disk} (1-\alpha)}{\|\hat{x}_{n_p}^{Plane} [n] \alpha + \hat{x}^{Disk} (1-\alpha)\|_2} & \text{for } (n_p = 0) \\ \hat{x}_{n_p-1}^{Plane} [n] & \text{for } (1 \leq n_p \leq N_p - 1) \end{cases}, \quad (4)$$

$$x_{n_p}^{Plane} [n+1] = \begin{cases} 0 & \text{for } (n_p = 0) \\ x_{n_p-1}^{Plane} [n] + \left\{ \hat{x}_{n_p-1}^{Plane} [n] \right\}^T \vec{V}_{n_p-1}^{Plane} \Delta t & \text{for } (1 \leq n_p \leq N_p - 1) \end{cases}, \text{ and} \quad (5)$$

$$\vec{p}_{n_p}^{Plane} [n+1] = \begin{cases} \vec{p}_{n_p}^{Plane} [n] \alpha + \vec{p}^{Hub} (1-\alpha) \\ + \left(C_{HWkDfl}^O + C_{HWkDfl}^{OY} \text{Filtered} \gamma_{YawErr} [n+1] \right) \left\{ \frac{\left(\left\{ \hat{x}_{n_p}^{Plane} [n+1] \right\}^T \hat{X} \right) \hat{Y} - \left(\left\{ \hat{x}_{n_p}^{Plane} [n+1] \right\}^T \hat{Y} \right) \hat{X}}{\left\| \left\{ \hat{x}_{n_p}^{Plane} [n+1] \right\}^T \hat{X} \right\} \hat{X} + \left\{ \left\{ \hat{x}_{n_p}^{Plane} [n+1] \right\}^T \hat{Y} \right\} \hat{Y}} \right\} & \text{for } (n_p = 0) \\ \vec{p}_{n_p-1}^{Plane} [n] + \vec{V}_{n_p-1}^{Plane} \Delta t \\ + \left(\left(C_{HWkDfl}^{OX} + C_{HWkDfl}^{OXY} \text{Filtered} \gamma_{YawErr} [n+1] \right) \left(\left\{ \hat{x}_{n_p-1}^{Plane} [n] \right\}^T \vec{V}_{n_p-1}^{Plane} \Delta t \right) + \left(C_{HWkDfl}^O + C_{HWkDfl}^{OY} \text{Filtered} \gamma_{YawErr} [n+1] \right) \right) \left\{ \frac{\left(\left\{ \hat{x}_{n_p}^{Plane} [n+1] \right\}^T \hat{X} \right) \hat{Y} - \left(\left\{ \hat{x}_{n_p}^{Plane} [n+1] \right\}^T \hat{Y} \right) \hat{X}}{\left\| \left\{ \hat{x}_{n_p}^{Plane} [n+1] \right\}^T \hat{X} \right\} \hat{X} + \left\{ \left\{ \hat{x}_{n_p}^{Plane} [n+1] \right\}^T \hat{Y} \right\} \hat{Y}} \right\} & \text{for } (1 \leq n_p \leq N_p - 1) \end{cases}. \quad (6)$$

Equation (4) is identical to Eq. (3) except that after applying Eq. (3a) to low-pass time-filter input \hat{x}^{Disk} , the state is renormalized to ensure that the three-component vector remains unit length; Eq. (4) ensures that the wake-plane orientation is maintained as the planes propagate nominally downwind. Equation (5) expresses that each wake plane propagates downwind in the axisymmetric coordinate system by a distance equal to that traveled by the wake-plane velocity projected along the plane's orientation over the time step; the initial wake plane ($n_p = 0$) is always at the rotor disk. Equation (6) expresses the global center positions of the wake plane following the passive tracer concept, similar to Eq. (5), but considering the full three-component movement of the wake plane (including deflection and meandering). The last term on the right-hand side of Eq. (6) for each wake plane is the horizontal wake-deflection correction, where C_{HWkDfl}^O , C_{HWkDfl}^{OY} , C_{HWkDfl}^{OX} , and C_{HWkDfl}^{OXY} are user-specified calibration parameters defining the horizontal offset at the rotor, the horizontal offset at the rotor scaled with nacelle-yaw error, the horizontal offset scaled with downstream distance, and the horizontal offset scaled with downstream distance and nacelle-yaw error, respectively, and \hat{X} , \hat{Y} , and \hat{Z} are unit vectors parallel to the inertial-frame coordinates X , Y , and Z ,

respectively. In this horizontal wake-deflection correction, $\left\{ \frac{\left(\left\{ \hat{x}_{n_p}^{Plane} [n+1] \right\}^T \hat{X} \right) \hat{Y} - \left(\left\{ \hat{x}_{n_p}^{Plane} [n+1] \right\}^T \hat{Y} \right) \hat{X}}{\left\| \left\{ \hat{x}_{n_p}^{Plane} [n+1] \right\}^T \hat{X} \right\} \hat{X} + \left\{ \left\{ \hat{x}_{n_p}^{Plane} [n+1] \right\}^T \hat{Y} \right\} \hat{Y}} \right\}$ is a

three-component unit vector in the horizontal global X - Y plane orthogonal to $\hat{x}_{n_p}^{Plane}$, $\left(C_{HWkDfl}^O + C_{HWkDfl}^{OY} \text{Filtered} \gamma_{YawErr} [n+1] \right)$ is the offset at the rotor, $\left(C_{HWkDfl}^{OX} + C_{HWkDfl}^{OXY} \text{Filtered} \gamma_{YawErr} [n+1] \right)$ is the slope, and $\left(\left\{ \hat{x}_{n_p-1}^{Plane} [n] \right\}^T \vec{V}_{n_p-1}^{Plane} \Delta t \right)$ is the nominally downwind increment of the wake plane (from Eq. (5)).

The consistent output equation corresponding to the low-pass time filter of Eq. (3a) is $y^d [n] = x^d [n] \alpha + u^d [n] (1-\alpha)$, i.e., $Y^d = X^d$, or equivalently, $y^d [n] = x^d [n+1]$.²⁴ To avoid having direct feedthrough of input to output within module WD, however, the output is delayed by one time step (Δt), yielding $y^d [n] = x^d [n]$. This one-time-step delay is applied to all outputs of module WD and is not expected to be problematic because of the slow timescales solved within FAST.Farm.

2. Near-Wake Correction

The near-wake correction submodel of module WD computes the axial and radial wake-velocity deficits at the rotor disk, as an inlet boundary condition for the wake-deficit evolution described in Section III.D.3. To improve the accuracy of the far-wake solution, the near-wake correction accounts for the drop-in wind speed and radial expansion of the wake in the pressure-gradient zone behind the rotor that is not otherwise accounted for in the solution for the wake-deficit evolution. For clarity, the equations in this section are expressed using continuous variables; but within FAST.Farm, the equations are solved discretely on an axisymmetric finite-difference grid.

First, the axial induction at the rotor disk, distributed radially, $a(r)$, is derived from the low-pass time-filtered azimuthally averaged thrust-force coefficient (normal to the rotor disk), $^{FilteredAzimAvg}C_t(r)$, evaluated at $n+1$ using Eq. (7). The formulation at low thrust-force coefficients ($< \frac{24}{25}$) follows from the momentum region of blade-element/momentum (BEM) theory; the formulation at high thrust-force coefficients ($\geq \frac{24}{25}$) follows from Glauert's correction with Buhl's modification to the empirical region.²⁵ (The propeller brake region is not considered for very high thrust-force coefficients (> 2 .) To avoid unrealistically high induction at the ends of a blade, Eq. (7) does not directly consider hub- or tip-loss corrections, but these may be accounted for in the calculation of the applied aerodynamic loads within FAST (depending on the aerodynamic options enabled within FAST), which have an effect on $^{FilteredAzimAvg}C_t(r)$.

$$a(r) = \begin{cases} \frac{l}{2} \left(1 - \sqrt{1 - ^{FilteredAzimAvg}C_t(r)} \right) & \text{for } \left(^{FilteredAzimAvg}C_t(r) < \frac{24}{25} \right) \\ \frac{2 + 3\sqrt{14 ^{FilteredAzimAvg}C_t(r) - 12}}{14} & \text{for } \left(\frac{24}{25} \leq ^{FilteredAzimAvg}C_t(r) \leq 2 \right) \end{cases} \quad (7)$$

The states and outputs associated with the axial and radial wake-velocity deficits, distributed radially, $V_{x_{n_p}}^{Wake}(r)$ and $V_{r_{n_p}}^{Wake}(r)$, are derived at the rotor disk ($n_p = 0$) from $a(r)$ and the low-pass time-filtered rotor-disk-averaged relative wind speed (ambient plus wakes plus turbine motion), normal to the disk, $^{FilteredDiskAvg}V_x^{Rel}$, evaluated at $n+1$ using Eqs. (8) and (9). In Eq. (8), r^{Plane} is the radial expansion of the wake associated with r , r' is a dummy variable of r , and $C_{NearWake}$ is a user-specified calibration parameter greater than unity, which determines how far the wind speed drops and wake expands radially in the pressure-gradient zone before recovering in the far wake; a value of $C_{NearWake} = 2$ is expected from first principles, but $C_{NearWake}$ can be calibrated by the user of FAST.Farm to better match the far wake to known solutions. The right-hand side of Eq. (8) represents the axial-induced velocity at the end of the pressure-gradient zone; the negative sign appears because the axial wake deficit is in the opposite direction of the free stream axial wind—see Section III.D.3 for more information. The radial expansion of the wake in the left-hand side of Eq. (8) results from the application of the conservation of mass within an incremental annulus in the pressure-gradient zone, i.e., the incremental mass flow is $d\dot{m} = 2\pi r dr \rho ^{FilteredDiskAvg}V_x^{Rel} (1 - a(r)) = 2\pi r^{Plane} dr^{Plane} \rho ^{FilteredDiskAvg}V_x^{Rel} (1 - C_{NearWake} a(r))$, so $r^{Plane} dr^{Plane} = \frac{l - a(r)}{1 - C_{NearWake} a(r)} r dr$, which can then be integrated along the radius. The radial wake deficit is initialized to zero, as given in Eq. (9). Because the near-wake correction is applied directly at the rotor disk, the solution to the wake-deficit evolution for downwind distances within the first few diameters of the rotor (i.e., in the near wake) is not expected to be accurate; as a result, modifications to FAST.Farm would be needed to accurately model closely spaced wind farms.

$$V_{x_{n_p}}^{Wake}(r^{Plane}) \Big|_{n_p=0, r^{Plane}} = \sqrt{2 \int_0^r \frac{l - a(r')}{1 - C_{NearWake} a(r')} r' dr'} = - ^{FilteredDiskAvg}V_x^{Rel} C_{NearWake} a(r) \quad (8)$$

$$V_{r_{n_p}}^{Wake}(r) \Big|_{n_p=0} = 0 \quad (9)$$

3. Wake-Deficit Increment

As with most DWM implementations, the WD module of FAST.Farm models the wake-deficit evolution via the thin shear-layer approximation of the Reynolds-averaged Navier-Stokes equations under quasi-steady-state conditions in axisymmetric coordinates, with turbulence closure captured by using an eddy-viscosity formulation. The thin shear-layer approximation drops the pressure term and assumes that the velocity gradients are much bigger

in the radial direction than in the axial direction. With these simplifications, analytical expressions for the conservation of momentum (Eq. (10)) and conservation of mass (continuity, Eq. (11)) are as follows:

$$V_x \frac{\partial V_x}{\partial x} + V_r \frac{\partial V_x}{\partial r} = \frac{1}{r} \frac{\partial}{\partial r} \left(r v_T \frac{\partial V_x}{\partial r} \right), \text{ or equivalently, } r V_x \frac{\partial V_x}{\partial x} + r V_r \frac{\partial V_x}{\partial r} = v_T \frac{\partial V_x}{\partial r} + r v_T \frac{\partial^2 V_x}{\partial r^2} + r \frac{\partial v_T}{\partial r} \frac{\partial V_x}{\partial r}, \quad (10)$$

$$\frac{\partial V_x}{\partial x} + \frac{1}{r} \frac{\partial}{\partial r} (r V_r) = 0, \text{ or equivalently, } V_r + r \frac{\partial V_r}{\partial r} + r \frac{\partial V_x}{\partial x} = 0, \quad (11)$$

where V_x and V_r are the axial and radial velocities in the axisymmetric coordinate system, respectively, and v_T is the eddy viscosity (all dependent on x and r , i.e., the downwind distance and radius in the axisymmetric coordinate system). The equations on the left are written in a form common in literature; the equivalent equations on the right are written in the form implemented within FAST.Farm. For clarity, the equations in this section are first expressed using continuous variables, but within FAST.Farm, the equations are solved discretely on an axisymmetric finite-difference grid consisting of a fixed number of wake planes, as summarized at the end of this section. For the continuous variables, subscript n_p , corresponding to wake plane n_p , is replaced with (x) , except for variables that remain constant as the wake propagates downstream, following Eq. (3b), where the subscript is dropped, i.e., $Filtered D_{n_p}^{Rotor}$, $FilteredDiskAvg V_{x_{n_p}}^{Wind}$, and $Filtered \Pi_{Amb_{n_p}}$ are written as $Filtered D^{Rotor}$, $FilteredDiskAvg V_x^{Wind}$, and $Filtered \Pi_{Amb}$, respectively.

V_x and V_r are related to the low-pass time-filtered rotor-disk-averaged ambient wind speed, normal to the disk, $FilteredDiskAvg V_x^{Wind}$, and the states and outputs associated with the axial and radial wake-velocity deficits, distributed radially, $V_x^{Wake}(x, r)$ and $V_r^{Wake}(x, r)$, by Eqs. (12) and (13). $V_x^{Wake}(x, r)$ and $V_r^{Wake}(x, r)$ can be thought of as the change in wind velocity in the wake relative to free stream; so, $V_x^{Wake}(x, r)$ usually has a negative value.

$$V_x(x, r) = FilteredDiskAvg V_x^{Wind} + V_x^{Wake}(x, r) \quad (12)$$

$$V_r(x, r) = V_r^{Wake}(x, r) \quad (13)$$

Several variations of the eddy-viscosity formulation have been used in prior implementations of DWM. The eddy-viscosity formulation currently implemented within FAST.Farm is given by Eq. (14), where $F_{vAmb}(x)$ and $F_{vShr}(x)$ are filter functions associated with ambient turbulence and the shear layer, respectively, and k_{vAmb} and k_{vShr} are user-specified calibration parameters weighting the influence of ambient turbulence and the shear layer on the eddy viscosity. The filter functions currently implemented within FAST.Farm are given by Eqs. (15) and (16), where C_{vAmb}^{DMax} , C_{vAmb}^{DMin} , C_{vAmb}^{Exp} , and C_{vAmb}^{FMin} and C_{vShr}^{DMax} , C_{vShr}^{DMin} , C_{vShr}^{Exp} , and C_{vShr}^{FMin} are user-specified calibration parameters for the functions associated with ambient turbulence and the shear layer, respectively. Although not matching any specific eddy-viscosity formulation found in prior implementations of DWM, the chosen implementation within FAST.Farm is simple to apply and inherently tailorable, allowing the user to properly calibrate the wake evolution to known solutions.

$$v_T(x, r) = F_{vAmb}(x) k_{vAmb} Filtered \Pi_{Amb} FilteredDiskAvg V_x^{Wind} \frac{Filtered D^{Rotor}}{2} + F_{vShr}(x) k_{vShr} MAX \left(\left[\frac{D^{Wake}(x)}{2} \right]^2 \left| \frac{\partial V_x}{\partial r}(x, r) \right|, \frac{D^{Wake}(x)}{2} MIN_r(V_x(x, r)) \right) \quad (14)$$

$$F_{vAmb}(x) = \begin{cases} C_{vAmb}^{FMin} & \text{for } (x \leq C_{vAmb}^{DMin} Filtered D^{Rotor}) \\ C_{vAmb}^{FMin} + (1 - C_{vAmb}^{FMin}) \left[\frac{x}{Filtered D^{Rotor}} - C_{vAmb}^{DMin} \right]^{C_{vAmb}^{Exp}} & \text{for } (C_{vAmb}^{DMin} Filtered D^{Rotor} < x < C_{vAmb}^{DMax} Filtered D^{Rotor}) \\ 1 & \text{for } (x \geq C_{vAmb}^{DMax} Filtered D^{Rotor}) \end{cases} \quad (15)$$

$$F_{vShr}(x) = \begin{cases} C_{vShr}^{FMin} & \text{for } (x \leq C_{vShr}^{DMin} \text{ Filtered } D^{Rotor}) \\ C_{vShr}^{FMin} + (1 - C_{vShr}^{FMin}) \left[\frac{x}{\text{Filtered } D^{Rotor}} - C_{vShr}^{DMin} \right]^{C_{vShr}^{Exp}} & \text{for } (C_{vShr}^{DMin} \text{ Filtered } D^{Rotor} < x < C_{vShr}^{DMax} \text{ Filtered } D^{Rotor}) \\ 1 & \text{for } (x \geq C_{vShr}^{DMax} \text{ Filtered } D^{Rotor}) \end{cases} \quad (16)$$

Equation (14) expresses the influence of the ambient turbulence (first term on the right-hand side) and shear layer (second term) on the turbulent stresses in the wake. The dependence of the eddy viscosity on x and r is explicitly given in Eq. (14) to make it clear which terms depend on the downwind distance and/or radius. The first term on the right-hand side of Eq. (14) is similar to that given by Madsen et al.,⁶ with a characteristic length taken to be the rotor radius, $\frac{\text{Filtered } D^{Rotor}}{2}$. The second term is similar to that given by Keck,⁸ but without consideration of

atmospheric shear. In this second term, the characteristic length is taken to be the wake half-width, $\frac{D^{Wake}(x)}{2}$; $MIN(V_x(x, r))$ is used to denote the minimum value of V_x along the radius for a given downstream distance; and the $MAX()$ operator is used to denote the maximum of the two shear-layer methods, where the second method is needed to avoid underpredicting the turbulent stresses from the first method at radii where the radial gradient of the axial velocity, $\left| \frac{\partial V_x}{\partial r} \right|$, approaches zero.

The filter functions of Eqs. (15) and (16), representing the delay in the turbulent stress generated by ambient turbulence and the development of turbulent stresses generated by the shear layer, respectively, are made general in FAST.Farm. Each filter function is split into three regions of downstream distance, including 1) a fixed minimum value (between zero and unity, inclusive) near the rotor, 2) a fixed value of unity far downstream from the rotor, and 3) a transition region for intermediate distances, where the value can transition linearly or via any rational exponent of the normalized downstream distance within the transition region.

The definition of wake diameter is somewhat ambiguous and not defined consistently in DWM literature. FAST.Farm allows the user to choose one of several methods to calculate the wake diameter, $D^{Wake}(x)$ —which is an output of module WD—including taking the wake diameter to be 1) the rotor diameter, $\text{Filtered } D^{Rotor}$; 2) the diameter at which the axial velocity of the wake is the $C_{WakeDiam}$ fraction of the ambient wind speed; 3) the diameter that captures the $C_{WakeDiam}$ fraction of the mass flux of the axial wake deficit across the wake plane; or 4) the diameter that captures the $C_{WakeDiam}$ fraction of the momentum flux of the axial wake deficit across the wake plane, where $C_{WakeDiam}$ is a user-specified calibration parameter between zero and unity (exclusive). Through the use of a $MAX()$ operator, models 2) through 4) have a lower bound set equal to the rotor diameter when the wake-diameter calculation otherwise returns smaller values so as to avoid numerical problems (too few wind data points) in the spatial averaging used to compute the wake-meandering velocity—see Section III.E for more information.

The momentum and continuity equations above are solved numerically in the wake-deficit-increment submodel of FAST.Farm module WD using a second-order accurate finite-difference method at $n + \frac{1}{2}$, following the implicit

Crank-Nicolson method.²⁶ That is, central differences are used for all derivatives; e.g., $\frac{\partial V_x}{\partial x} = \frac{V_{x_{n_p}}^{Wake}(r)[n+I] - V_{x_{n_p-1}}^{Wake}(r)[n]}{\Delta x}$ for the momentum equation, where $\Delta x = x_{n_p}^{Plane}[n+I] - x_{n_p-1}^{Plane}[n]$, or equivalently from Eq. (5), $\Delta x = \{ \hat{x}_{n_p-1}^{Plane}[n] \}^T \vec{V}_{n_p-1}^{Plane} \Delta t$ (for $(I \leq n_p \leq N_p - I)$).

For the momentum equation, for each wake plane downstream of the rotor $(I \leq n_p \leq N_p - I)$, the terms V_x , V_r , v_T , and $\frac{\partial V_T}{\partial r}$ are calculated at n (or equivalently $x = x_{n_p-1}^{Plane}[n]$)—e.g., $V_x = \text{FilteredDiskAvg } V_{x_{n_p-1}}^{Wind}[n] + V_{x_{n_p-1}}^{Wake}(r)[n]$ and

$V_r = V_{r_{n_p-1}}^{Wake}(r)[n]$ —to avoid nonlinearities in the solution for $n+1$, which will prevent the solution from achieving second-order convergence, but has been shown to remain numerically stable. Although the definition of each central difference is outside the scope of this paper, the end result is that for each wake plane downstream of the rotor ($1 \leq n_p \leq N_p - 1$), $V_{x_{n_p}}^{Wake}(r)[n+1]$ can be solved via a linear tridiagonal matrix system of equations in terms of known solutions of $V_{x_{n_p-1}}^{Wake}(r)[n]$, $V_{r_{n_p-1}}^{Wake}(r)[n]$, and other previously calculated states, e.g., $FilteredDiskAvg V_{x_{n_p-1}}^{Wind}[n]$. The linear tridiagonal matrix system of equations is solved efficiently in FAST.Farm via the Thomas algorithm.²⁷

For the continuity equation, a different finite-difference scheme is needed because when the same finite-difference scheme used for the momentum equation is used for the continuity equation, the resulting tridiagonal matrix is not diagonally dominant, resulting in a numerically unstable solution. Instead, the finite-difference scheme used for the continuity equation is based on a second-order accurate scheme at $n + \frac{1}{2}$ and $n_r - \frac{1}{2}$, e.g.,

$V_r = \frac{1}{4} \left(V_{r_{n_p, n_r}}^{Wake}[n+1] + V_{r_{n_p, n_r-1}}^{Wake}[n+1] + V_{r_{n_p-1, n_r}}^{Wake}[n] + V_{r_{n_p-1, n_r-1}}^{Wake}[n] \right)$, where n_r is the radii counter for N_r radial nodes ($0 \leq n_r \leq N_r - 1$) (subscript n_r has been used here in place of (r)). Although the definition of each central difference is outside the scope of this paper, the end result is that for each wake plane downstream of the rotor ($1 \leq n_p \leq N_p - 1$), $V_{r_{n_p, n_r}}^{Wake}[n+1]$ can be solved explicitly sequentially from known solutions of $V_{x_{n_p}}^{Wake}(r)[n+1]$ (from the solution of the momentum equation), $V_{x_{n_p-1}}^{Wake}(r)[n]$, $V_{r_{n_p-1}}^{Wake}(r)[n]$, and $V_{r_{n_p, n_r-1}}^{Wake}[n+1]$ for $1 \leq n_p \leq N_p - 1$; note that the radial wake-velocity deficit at the centerline of the axisymmetric coordinate system ($n_r = 0$) is always zero,

$$V_{r_{n_p}}^{Wake}(r) \Big|_{r=0} = 0.$$

E. Ambient Wind and Array Effects (Module AWAE)

The AWAE module of FAST.Farm processes ambient wind and wake interactions across the wind farm, including the ambient wind submodel, which processes ambient wind across the wind farm from a precursor ABLSolver (or equivalent) simulation, and the wake-merging submodel, which identifies zones of overlap between all wakes across the wind farm and merges their wake deficits. Both submodels are described in the subsections below.

The calculations in module AWAE make use of wake volumes, which are volumes formed by a cylinder starting at a wake plane and extending axially downstream (along the wake plane's centerline) and ending where the cylinder is sliced by the next wake plane. The diameter of the cylinder is dependent on the calculation (see the subsections below). Figure 9 illustrates some of the concepts that will be detailed in the subsections below. The calculations in module AWAE also require looping through all wind data points, turbines, and wake planes; these loops have been sped up in FAST.Farm by implementation of parallelization.

The AWAE module does not have states, reducing the module to a feed-forward-only system whereby the module outputs are computed directly from the module inputs (with direct feedthrough of input to output). Module AWAE uses as input $\hat{x}_{n_p}^{Plane}$, $\bar{p}_{n_p}^{Plane}$, $V_{x_{n_p}}^{Wake}(r)$, $V_{r_{n_p}}^{Wake}(r)$, and $D_{n_p}^{Wake}$ (each for $0 \leq n_p \leq N_p - 1$) as computed by the wake-dynamics model for each individual wind turbine (output by module WD). Module AWAE computes output \vec{V}_{Dist}^{High} needed for the calculation of FAST for each individual wind turbine (input to module F), as well as outputs $\vec{V}_{n_p}^{Plane}$ for $0 \leq n_p \leq N_p - 1$, $^{DiskAvg} V_x^{Wind}$, and TI_{Amb} needed for the calculation of wake dynamics for each individual wind turbine (input to module WD).

1. Ambient Wind

Instead of relying on simplified synthetically generated turbulence local to an individual rotor and based on Taylor's frozen-turbulence assumption as the driver for wake meandering, FAST.Farm uses atmospheric phenomena generated by a precursor LES simulation of the entire wind farm (without wind turbines present), as is currently implemented in the ABLSolver preprocessor of SOWFA.¹⁸ This precursor atmospheric simulation captures more physics than synthetic turbulence—as illustrated in Figure 10—including stability and complex terrain effects, but it is much less computationally expensive than a SOWFA simulation with multiple wind turbines present. In place of ABLSolver, a precursor atmospheric simulation could also be run with a tool equivalent to ABLSolver.

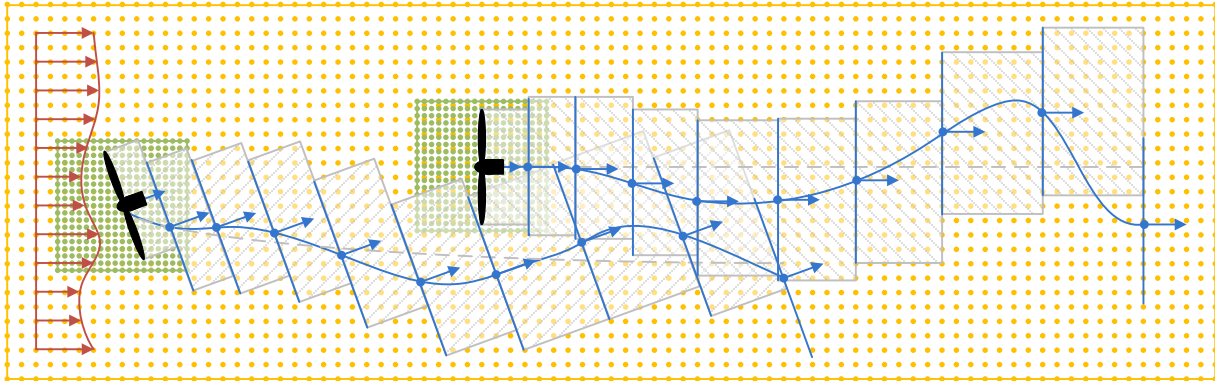


Figure 9. Wake planes, wake volumes, and zones of wake overlap for a two-turbine wind farm, with the upwind turbine yawed. The yellow points represent the low-resolution wind domain and the green points represent the high-resolution wind domains around each turbine. The blue points and arrows represent the centers and orientations of the wake planes, respectively, with the wake planes identified by the blue lines normal to their orientations. The gray dashed lines represent the mean trajectory of the wake and the blue curves represent an instantaneous (meandered) trajectory. The wake volumes associated with the upwind turbine are represented by the upward hatch patterns, the wake volumes associated with the downwind turbine are represented by the downward hatch patterns, and the zones of wake overlap are represented by the crosshatch patterns. (For clarity of the illustration, the wake volumes are shown with smaller diameters than they would practically have.)

FAST.Farm requires ambient wind to be input in two different resolutions. Because wind will be spatially averaged across wake volumes within module AWAE, FAST.Farm needs as input a low-resolution wind domain (in both space and time) throughout the wind farm wherever wakes may potentially reside. The spatial resolution of the low-resolution domain—consisting of a (potentially unstructured) three-dimensional grid of wind data points—should be sufficient so that the spatial averaging is accurate, e.g., on the order of tens of meters for utility-scale wind turbines. The time step of the low-resolution domain dictates that of the FAST.Farm driver (Δt) and all FAST.Farm modules, and so, it should be consistent with the timescales of wake dynamics, e.g., on the order of seconds (FAST is subcycled within module F with a smaller time step). For accurate load calculation by FAST, FAST.Farm also needs as input high-resolution wind domains (in both space and time) around each wind turbine (encompassing any turbine displacement). The spatial and time resolution of each high-resolution domain should be sufficient for accurate aerodynamic load calculations, e.g., on the order of the blade chord length and fractions of a second. For simplicity of the algorithm and to minimize computational expense within FAST.Farm, the time step of the high-resolution domain must be an integer divisor of the low-resolution domain’s time step. The high-resolution domains will occupy the same space as portions of the low-resolution domain, i.e., the domains overlap.

Within each time step, the AWAE module reads in the three-component wind-velocity data across the high- and low-resolution domains— \vec{V}_{Amb}^{High} for each turbine and \vec{V}_{Amb}^{Low} , respectively—that were precomputed by ABLSolver (or equivalent) and stored in files for use in a given driver time step. The wind data files, including their spatial discretizations, are specified by users of FAST.Farm at initialization. These wind data from the combined low- and high-resolution domains within a given driver time step represent the largest memory requirement of FAST.Farm.

After the ambient wind is read in at a given time step, the ambient wind submodel of module AWAE computes as output the rotor-disk-averaged ambient wind speed, normal to the disk, $_{DiskAvg} V_x^{Wind}$, for each turbine using Eq. (17). In Eq. (17), $N_{n_p}^{Wind}$ is the number of wind data points in the low-resolution domain that reside within wake volume n_p (i.e., the wake volume associated with wake plane n_p) for the given wind turbine, n^{Wind} is the point counter such that $1 \leq n^{Wind} \leq N_{n_p}^{Wind}$ for wake volume n_p , and the equation is evaluated for the wake volume at the rotor disk ($n_p = 0$).

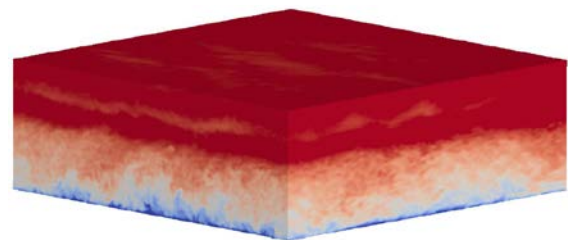


Figure 10. Example flow generated by ABLSolver.¹⁸

(If $N_{n_p}^{Wind}$ is not large enough to calculate an accurate spatial averaging, the user of FAST.Farm should change the low-resolution wind data, e.g., use a finer spatial resolution or a larger time step—to make the wake planes farther apart.) Wake volume n_p starts at wake plane n_p ; extends downwind to, and ends at, wake plane $n_p + I$; has a centerline determined by $\vec{p}_{n_p}^{Plane}$ and $\hat{x}_{n_p}^{Plane}$; and has a diameter of $2D_{n_p}^{Wake}$ (for $0 \leq n_p \leq N_p - 2$). Subscript n^{Wind} is appended to \vec{V}_{Amb}^{Low} in Eq. (17) to identify the wind data points in the low-resolution domain that reside in wake volume n_p , which are found by looping through all points, turbines, and wake planes and spatially determining if the given point resides in the corresponding wake volume. Intuitively, Eq. (17) states that the rotor-disk-averaged ambient wind speed, normal to the disk, for each turbine is calculated as the spatial average of the ambient wind velocity immediately downwind of the rotor disk projected along the low-pass time-filtered rotor centerline.

$$DiskAvg \vec{V}_x^{Wind} = \left(\left\{ \hat{x}_{n_p}^{Plane} \right\}^T \left\{ \frac{I}{N_{n_p}^{Wind}} \sum_{n^{Wind}=I}^{N_{n_p}^{Wind}} \vec{V}_{Amb, n^{Wind}}^{Low} \right\} \right) \Bigg|_{n_p=0} \quad (17)$$

The ambient wind submodel of module AWAE also calculates as output the ambient turbulence intensity around each rotor, TI_{Amb} , but the algorithm for calculating TI_{Amb} was not complete as of this writing and is not further discussed in this paper. The incorporation of wake-added turbulence is also left for future work.

2. Wake Merging

In previous implementations of DWM, the wind turbine and wake dynamics are solved individually or serially, not considering two-way wake-merging interactions, and there is no method available to calculate the disturbed wind in zones of wake overlap. Wake merging is illustrated by the SOWFA simulation of Figure 11.

In FAST.Farm, the wake-merging submodel of module AWAE identifies zones of wake overlap between all wakes across the wind farm and superimposes the wake deficits in the axial direction based on the root-sum-squared (RSS) method;²⁸ transverse components (radial wake deficits) are superimposed by vector sum. Although Katić et al.²⁸ applies the RSS method to wakes with axial deficits that are uniform across the wake diameter (and radial deficits are not considered), in FAST.Farm, the RSS method is applied locally at a given wind data point. The RSS method assumes that the local kinetic energy of the axial deficit in a merged wake equals the sum of the local energies of the axial deficits for each wake at the given wind data point. The RSS method only applies to an array of scalars, which works well for axial deficits because overlapping wakes likely have similar axial directions (and so only the magnitude of the vector is important in the superposition). A vector sum is applied to the transverse components (radial wake deficits) because any given radial direction is dependent on the azimuth angle in the axisymmetric coordinate system.

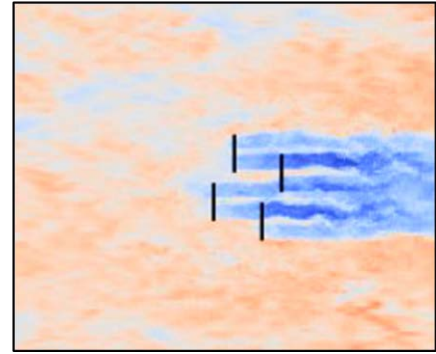


Figure 11. Wake merging for closely spaced rotors as solved by SOWFA.

The disturbed (ambient plus wakes) wind velocities across the high- and low-resolution domains— \vec{V}_{Dist}^{High} for each turbine and \vec{V}_{Dist}^{Low} , respectively—are computed using Eqs. (18) and (19), respectively. The first, second, and third terms on the right-hand side of Eqs. (18) and (19) represent the ambient wind velocity, the RSS superposition of the axial wake-velocity deficits, and the vector sum of the transverse wake-velocity deficits, respectively. Although many mathematical details are outside the scope of this paper, the nomenclature of Eqs. (18) and (19) is as follows: N^{Wake} is the number of wakes (or turbines, equivalently, because one wake is produced by each turbine) overlapping a given wind data point in the wind domain and n^{Wake} is the wake counter such that $1 \leq n^{Wake} \leq N^{Wake}$. These wakes, and their specific wake volumes, are found by looping through all points, turbines, and wake planes and spatially determining if the given point resides in a wake volume that has a diameter equal to the diameter (radial extent) of its corresponding wake plane. $V_{x, n^{Wake}}^{Wake}$, $V_{r, n^{Wake}}^{Wake}$, $\hat{x}_{n^{Wake}}^{Plane}$, and $\hat{r}_{n^{Wake}}^{Plane}$ are the axial wake-velocity deficit, radial wake-velocity deficit, axial orientation, and radial unit vector, respectively, associated with where the given wind data point lies within the specific wake volume and corresponding wake plane, where subscript n^{Wake} is used to identify the specific point in a wake plane in place of (r) and subscript n_p that were used in Section III.D. The calculations of $V_{x, n^{Wake}}^{Wake}$ and $V_{r, n^{Wake}}^{Wake}$

$V_{r_n^{Wake}}^{Wake}$ involve interpolation of the wake deficits in the radial direction, but not in the axial direction, to avoid overcomplicating the algorithm because adjacent wake planes are not necessarily parallel. The lack of interpolation in the axial direction is satisfactory because the thin-shear-layer approximation in the wake-deficit model assumes that the gradients in the radial direction are much larger than the gradients in the axial direction. The vector quantity $\left\{ V_{x_n^{Wake}}^{Wake} \hat{x}_{n^{Wake}}^{Plane} + V_{r_n^{Wake}}^{Wake} \hat{r}_{n^{Wake}}^{Plane} \right\}$ represents the total wake-velocity deficit associated with where the given wind data point lies within the specific wake volume and corresponding wake plane. Because each wake plane may have a unique orientation, what constitutes ‘‘axial’’ and ‘‘radial’’ in the superposition at a given wind data point is determined by weighted-averaging the orientations of each wake volume overlapping that point (weighted by the magnitude of each axial wake deficit). \bar{x}^{Plane} is the weighted-average axial orientation associated with a given point in the wind spatial domain; $\left\{ \bar{x}^{Plane} \right\}^T$ is used to project $\left\{ V_{x_n^{Wake}}^{Wake} \hat{x}_{n^{Wake}}^{Plane} + V_{r_n^{Wake}}^{Wake} \hat{r}_{n^{Wake}}^{Plane} \right\}$ along this axis; and $\left[I - \bar{x}^{Plane} \left\{ \bar{x}^{Plane} \right\}^T \right]$, where I is the three-by-three identity matrix, is used to calculate the transverse component of $\left\{ V_{x_n^{Wake}}^{Wake} \hat{x}_{n^{Wake}}^{Plane} + V_{r_n^{Wake}}^{Wake} \hat{r}_{n^{Wake}}^{Plane} \right\}$ normal to \bar{x}^{Plane} . For the disturbed (ambient plus wakes) wind velocities across the high-resolution domain for each turbine (needed to calculate the disturbed wind inflow to a turbine), \vec{V}_{Dist}^{High} , ‘‘for ... otherwise’’ statements are used in the summations to prevent a turbine from interacting with its own wake; here, $\left(n_{t_{n^{Wake}}} \neq n_t \right)$ signifies that wake n^{Wake} is not associated with the given turbine n_t .

$$\vec{V}_{Dist}^{High} = \vec{V}_{Amb}^{High} - \left\{ \sqrt{\sum_{n^{Wake}=1}^{N^{Wake}} \left[\begin{array}{ll} \left(\left\{ \bar{x}^{Plane} \right\}^T \left\{ V_{x_n^{Wake}}^{Wake} \hat{x}_{n^{Wake}}^{Plane} + V_{r_n^{Wake}}^{Wake} \hat{r}_{n^{Wake}}^{Plane} \right\} \right)^2 & \text{for } \left(n_{t_{n^{Wake}}} \neq n_t \right) \\ 0 & \text{otherwise} \end{array} \right]} \right\} \bar{x}^{Plane} \quad (18)$$

$$+ \sum_{n^{Wake}=1}^{N^{Wake}} \left\{ \left[I - \bar{x}^{Plane} \left\{ \bar{x}^{Plane} \right\}^T \right] \left\{ V_{x_n^{Wake}}^{Wake} \hat{x}_{n^{Wake}}^{Plane} + V_{r_n^{Wake}}^{Wake} \hat{r}_{n^{Wake}}^{Plane} \right\} \right\} \text{for } \left(n_{t_{n^{Wake}}} \neq n_t \right) \\ \vec{0} \text{ otherwise} \\ \vec{V}_{Dist}^{Low} = \vec{V}_{Amb}^{Low} - \left\{ \sqrt{\sum_{n^{Wake}=1}^{N^{Wake}} \left(\left\{ \bar{x}^{Plane} \right\}^T \left\{ V_{x_n^{Wake}}^{Wake} \hat{x}_{n^{Wake}}^{Plane} + V_{r_n^{Wake}}^{Wake} \hat{r}_{n^{Wake}}^{Plane} \right\} \right)^2} \right\} \bar{x}^{Plane} \\ + \sum_{n^{Wake}=1}^{N^{Wake}} \left[I - \bar{x}^{Plane} \left\{ \bar{x}^{Plane} \right\}^T \right] \left\{ V_{x_n^{Wake}}^{Wake} \hat{x}_{n^{Wake}}^{Plane} + V_{r_n^{Wake}}^{Wake} \hat{r}_{n^{Wake}}^{Plane} \right\} \quad (19)$$

Once the distributed (ambient plus wakes) wind velocities across the low-resolution domain have been found, the wake merging submodel of module AWAE computes as output the advection, deflection, and meandering velocity of each wake plane, $\vec{V}_{n_p}^{Plane}$ for $0 \leq n_p \leq N_p - 2$, for each turbine, using Eq. (20) (the advection, deflection, and meandering velocity of the most downwind wake plane ($n_p = N_p - 1$) for each turbine is not needed because that plane is not propagated further.) As in Eq. (17), in Eq. (20), $N_{n_p}^{Wind}$ is the number of wind data points in the low-resolution domain that reside within wake volume n_p for the given wind turbine and n^{Wind} is the point counter such that $1 \leq n^{Wind} \leq N_{n_p}^{Wind}$ for wake volume n_p , where wake volume n_p is determined by $\bar{p}_{n_p}^{Plane}$ and $\hat{x}_{n_p}^{Plane}$, with a diameter of $2D_{n_p}^{Wake}$ (for $0 \leq n_p \leq N_p - 2$). The wake-volume diameter of $2D_{n_p}^{Wake}$ follows from the characteristic dimension important to wake meandering proposed by Larsen et al.⁵ Subscript n^{Wind} is appended to \vec{V}_{Dist}^{Low} in Eq. (20) to identify the wind data points in the low-resolution domain that reside in wake volume n_p , which are found by looping through all points, turbines, and wake planes and spatially determining if the given point resides in the wake volume. Qualitatively, Eq. (20) states that the advection, deflection, and meandering velocity of each wake plane for

each turbine is calculated as the spatial average of the disturbed (ambient plus wakes) wind velocity of the corresponding wake volume.

$$\vec{V}_{n_p}^{Plane} = \frac{I}{N_{n_p}^{Wind}} \sum_{n^{Wind}=1}^{N_{n_p}^{Wind}} \vec{V}_{Dist_{n^{Wind}}}^{Low} \quad (20)$$

IV. Conclusion

FAST.Farm is a new multiphysics tool applicable to engineering problems in research and industry involving wind farm performance and cost optimization that is needed to address the current underperformance, failures, and expenses plaguing the wind industry. FAST.Farm aims to balance the need for accurate modeling of the relevant physics for predicting power performance and loads while maintaining low computational cost to support a highly iterative and probabilistic design process and system-wide optimization. FAST.Farm makes use of FAST to model the aero-hydro-servo-elastics of distinct turbines in the wind farm and is based on some of the principles of the DWM model. However, FAST.Farm avoids many of the limitations of existing DWM implementations, includes the controls capability of FLORIS, and functions more like SOWFA. Insight from well-validated SOWFA simulations is being used to support the development and parameter calibration of FAST.Farm.

It is envisioned that FAST.Farm, once functional, will create a paradigm shift in wind farm design capability, with potential future applications in reducing wind farm underperformance and loads uncertainty, developing wind farm controls to enhance the operation of existing wind farms, optimizing the siting and topology optimization of new wind farms, and innovating the design of wind turbines for the wind farm environment. The existing implementation of FAST.Farm also forms a solid foundation for further development of wind farm dynamics modeling as wind farm physics knowledge grows from future computations and experiments.

Future Work

The source-code implementation of FAST.Farm will be completed soon—including the completion of the algorithm for calculating the AWAE module output of ambient turbulence intensity around each rotor, TI_{Amb} —and model calibration, verification, validation, and applications will be presented in future work. Although FAST.Farm avoids many of the limitations of existing DWM implementations, there are potential limitations of FAST.Farm that may need to be addressed in the future, including 1) incorporating wake-added turbulence; 2) improving the eddy-viscosity formulation with additional physics; 3) reflecting wakes off of the ground; 4) addressing deep-array effects for large wind farms; 5) accounting for flow speed-up around the edges of a wind farm, i.e., tackling the wind farm blockage effect; 6) allowing for a more general module form, e.g., supporting continuous states, direct feedthrough of input to output, and full-system linearization; and 7) pursuing additional wake-modeling approaches, including applying a free-vortex method for the near wake and deforming the base-wake deficit (introducing asymmetry) as a result of background turbulence (in addition to wake meandering).

Beyond addressing these potential limitations, possible future development pathways for FAST.Farm include 1) developing wind farm control strategies, based on modifying wake deficits and/or redirecting (steering) wakes, to achieve global benefit of improving performance and reducing loads and comparing to conventional control strategies to prove the new strategy and demonstrate FAST.Farm’s unique capabilities; 2) interfacing FAST.Farm to the Wind-Plant Integrated System Design & Engineering Model (WISDEMTM)²⁹ for systems-engineering applications (multidisciplinary design analysis and optimization [MDAO], uncertainty quantification [UQ], and so on); 3) developing a wrapper for standalone AeroDyn^{††30} (or an equivalent BEM tool) as an alternative to FAST to support advanced performance-only wind farm analysis that is much more computationally efficient than FAST.Farm analysis using FAST; 4) addressing unique offshore wind energy challenges, e.g., ensuring consistent waves across an offshore wind farm and supporting the air-water interface; and 5) adopting the capability to support undersea marine turbine arrays (which may require supporting direct feedthrough of input to output to handle the added-mass effects).

Acknowledgments

This work was performed at NREL in support of the U.S. Department of Energy under Contract number DE-AC36-08GO28308 and under a project funded through an NREL Laboratory Directed Research and Development

^{††}AeroDyn is the aerodynamics module of FAST.

(LDRD) grant titled, “Multi-Physics Engineering Tool for Wind-Plant Design and Analysis.” The contributions of Matt Churchfield, Paul Fleming, Pieter Gebraad, Sang Lee, Pat Moriarty, Eliot Quon, and Qi Wang of NREL and their support of this LDRD project are acknowledged. The contributions of Matt Churchfield and Sang Lee of NREL to generating Figure 1 and Figure 11 are appreciated.

The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

References

- ¹Gebraad, P. M. O. *Data-Driven Wind Plant Control*. Ph.D. Thesis. Delft Center for Systems and Control and Delft University Wind Research Institute, Technical University of Delft, Delft, The Netherlands, 2014.
- ²Gebraad, P. M. O.; Teeuwisse, F. W.; van Wingerden, J. W.; Fleming, P. A.; Ruben, S. D.; Marden, J. R.; and Pao, L. Y. “Wind Plant Power Optimization Through Yaw Control Using a Parametric Model for Wake Effects – a CFD Simulation Study.” *Wind Energy*. Vol. 19, No. 1, January 2016, pp. 95–114; DOI: 10.1002/we.1822.
- ³Annoni, J.; Gebraad, P. M. O.; Scholbrock, A. K.; Fleming, P. A.; and van Wingerden, J. W. “Analysis of Axial-Induction-Based Wind Plant Control Using an Engineering and a Higher-Order Wind Plant Model.” *Wind Energy*. Vol. 19, No. 6, June 2016, pp. 1135–1150; DOI: 10.1002/we.1891.
- ⁴Web page: <https://github.com/WISDEM/FLORISSE> (accessed May 28, 2016).
- ⁵Larsen, G. C.; Madsen, H. A.; Thomsen, K.; and Larsen, T. A. “Wake Meandering: A Pragmatic Approach.” *Wind Energy*. Vol. 11, No. 4, July/August 2008, pp. 337–395; DOI: 10.1002/we.267.
- ⁶Madsen, H. A.; Larsen, G. C.; Larsen, T. J.; Troldborg, N.; and Mikkelsen, R. “Calibration and Validation of the Dynamic Wake Meandering Model for Implementation in an Aeroelastic Code.” *Journal of Solar Energy Engineering*. Vol. 32, No. 4, November, 2010, pp. 041014-1–041014-14; DOI: 10.1115/1.4002555.
- ⁷Larsen, T. J.; Madsen, H. A.; Larsen, G. C.; and Hansen, K. S. “Validation of the Dynamic Wake Meander Model for Loads and Power Production in the Egmond aan Zee Wind Farm.” *Wind Energy*. Vol. 16, No. 4, May 2013, pp. 605–624; DOI: 10.1002/we.1563.
- ⁸Keck, R. E. *A Consistent Turbulence Formulation for the Dynamic Wake Meandering Model in the Atmospheric Boundary Layer*. Ph.D. Thesis. Department of Wind Energy, Danish Technical University, Lyngby, Denmark, July 2013.
- ⁹Madsen, H. A.; Larsen, T. J.; Larsen, G. C.; and Hansen, K. S. “Wake Flow Characteristics at High Wind Speed.” *Proceedings of the AIAA SciTech Forum and Exposition, 4–8 January 2016, San Diego, CA*. AIAA 2016-1522; <http://dx.doi.org/10.2514/6.2016-1522>.
- ¹⁰Hao, Y.; Lackner, M. A.; Keck, R. E.; Lee, S.; Churchfield, M. J.; and Moriarty, P. “Implementing the Dynamic Wake Meandering Model in the NWTC Design Codes.” *Proceedings of the AIAA SciTech Forum and Exposition, 13–17 January 2014, National Harbor, MD*. AIAA 2014-1089; <http://dx.doi.org/10.2514/6.2014-1089>. NREL/CP-5000-61115.
- ¹¹Churchfield, M. J.; Moriarty, P. J.; Hao, Y.; Lackner, M. A.; Barthelmie, R.; Lundquist, J.; Oxley, G. S. “A Comparison of the Dynamic Wake Meandering Model, Large-Eddy Simulation, and Field Data at the Egmond aan Zee Offshore Wind Plant.” *Proceedings of the AIAA SciTech Forum and Exposition, 5–9 January 2015, Kissimmee, FL*. AIAA 2015-0724; <http://dx.doi.org/10.2514/6.2015-0724>. NREL/CP-5000-63321.
- ¹²Hao, Y. *Wind Farm Wake Modeling and Analysis of Wake Impacts in a Wind Farm*. Ph.D. Thesis. Department of Mechanical Engineering, University of Massachusetts, Amherst, MA, March 2016.
- ¹³Web page: <https://nwtc.nrel.gov/DWM> (accessed May 28, 2016).
- ¹⁴Ainslie, J. F. “Calculating the Flowfield in the Wake of Wind Turbines.” *Journal of Wind Engineering and Industrial Aerodynamics*. Vol. 27, 1988, pp. 213–224.
- ¹⁵Web page: <https://nwtc.nrel.gov/FAST8> (accessed May 28, 2016).
- ¹⁶Churchfield, M. J.; Lee, S.; Moriarty, P. J.; Martinez, L. A.; Leonardi, S.; Vijayakumar, G.; and Brasseur, J. G. “A Large-Eddy Simulation of Wind-Plant Aerodynamics.” *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 9–12 January 2012, Nashville, TN*. AIAA 2012-0537. <http://dx.doi.org/10.2514/6.2012-537>. NREL/CP-5000-53554.
- ¹⁷Churchfield, M. J.; Lee, S.; Michalakes, J.; and Moriarty, P. J. “A Numerical Study of the Effects of Atmospheric and Wake Turbulence on Wind Turbine Dynamics.” *Journal of Turbulence*. Vol. 13, No. 14, May 2012, pp. 1–32; DOI: 10.1080/14685248.2012.668191.
- ¹⁸Web page: <https://nwtc.nrel.gov/SOWFA> (accessed May 28, 2016).
- ¹⁹Web page: <https://nwtc.nrel.gov/FAST.Farm> (accessed May 28, 2016).
- ²⁰Jonkman, J. M. “The New Modularization Framework for the FAST Wind Turbine CAE Tool.” *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 7–10 January 2013, Grapevine (Dallas/Ft. Worth Region), TX*. AIAA 2013-0202. <http://dx.doi.org/10.2514/6.2013-202>. NREL/CP-5000-57228.
- ²¹Sprague, M. A.; Jonkman, J. M.; and Jonkman, B. J. “FAST Modular Wind Turbine CAE Tool: Nonmatching Spatial and Temporal Meshes.” *AIAA Science and Technology Forum (SciTech 2014), 13–17 January 2014, National Harbor, MD*. AIAA-2014-0520. <http://arc.aiaa.org/doi/pdf/10.2514/6.2014-0520>. NREL/CP-5000-60742.

²²Sprague, M. A.; Jonkman, J. M.; and Jonkman, B. J. “FAST Modular Framework for Wind Turbine Simulation: New Algorithms and Numerical Examples.” *AIAA Science and Technology Forum and Exhibition (SciTech 2015)*, 5–9 January 2015, Kissimmee, FL. AIAA-2015-1461. <http://arc.aiaa.org/doi/pdf/10.2514/6.2015-1461>. NREL/CP-5000-60759.

²³Smith, S. W. *The Scientist and Engineer’s Guide to Digital Signal Processing*. California Technical Publishing, San Diego, California, 2006.

²⁴Jonkman, J.; Butterfield, S.; Musial, W.; and Scott, G. *Definition of a 5-MW Reference Wind Turbine for Offshore System Development*. NREL/TP-500-38060. February 2009.

²⁵Buhl Jr., M. L. *A New Empirical Relationship between Thrust Coefficient and Induction Factor for the Turbulent Windmill State*. NREL/TP-500-36834, 2005.

²⁶Crank, J. and Nicolson, P. “A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of the Heat-Conduction Type.” *Proceedings of the Cambridge Philosophical Society*, Vol. 43, No. 50, 1947, pp. 50–67.

²⁷Thomas, L. H. *Elliptic Problems in Linear Differential Equations over a Network*. Watson Scientific Computing Laboratory, Columbia University, New York, 1949.

²⁸Katić, I.; Højstrup, J.; and Jensen, N. O. “A Simple Model for Cluster Efficiency.” *Proceedings of the European Wind Energy Association Conference and Exhibition, 7–9 October 1986, Rome, Italy*. pp. 407–410.

²⁹Web page: <https://nwtc.nrel.gov/WISDEM> (accessed May 28, 2016).

³⁰Web page: <https://nwtc.nrel.gov/AeroDyn> (accessed May 28, 2016).