FAST modularization framework for wind turbine simulation: full-system linearization

FAST modularization framework for wind turbine simulation: full-system linearization

J M Jonkman and B J Jonkman

National Renewable Energy Laboratory, Golden, CO 80401, USA

E-mail: jason.jonkman@nrel.gov

Abstract. The wind engineering community relies on multiphysics engineering software to run nonlinear time-domain simulations e.g. for design-standards-based loads analysis. Although most physics involved in wind energy are nonlinear, linearization of the underlying nonlinear system equations is often advantageous to understand the system response and exploit well-established methods and tools for analyzing linear systems. This paper presents the development and verification of the new linearization functionality of the open-source engineering tool FAST v8 for land-based wind turbines, as well as the concepts and mathematical background needed to understand and apply it correctly.

1. Introduction

To support design and analysis—so that wind turbines are innovative, optimized, reliable, and cost-effective—the wind industry and research communities rely on engineering software (i.e. design tools) capable of predicting the coupled dynamic loads and responses of the wind system. FAST, developed by the National Renewable Energy Laboratory (NREL) through U.S. Department of Energy support, is an open-source multiphysics tool practical to the engineering design of wind turbines [1]. FAST models the important physical phenomena and system couplings, including the environmental excitation (wind, waves, and current) and full-system dynamic response (rotor, drivetrain, nacelle, support structure, and controller) under both normal (for fatigue) and extreme (for ultimate) loading conditions. The FAST model enables the analysis of a range of wind turbine configurations, including two- or three-blade horizontal-axis rotors, pitch or stall regulation, rigid or teetering hub, upwind or downwind rotor, and lattice or tubular towers. The wind turbine can be modeled on land or offshore on bottom-fixed or floating substructures.

Over the past several years, FAST has undergone a major restructuring through the introduction of a modularization framework (FAST version 8) that enables coupling of various modules, each representing different components or physics domains of the wind system [2]. FAST v8 includes a mesh-mapping utility allowing each module to be independently discretized in space and time and a mathematically rigorous solution procedure supporting loose coupling of modules with implicit-coupling relations [3]. For land-based wind turbine simulations, FAST has modules for wind inflow (InflowWind); aerodynamics (AeroDyn); control and electrical-drive dynamics (ServoDyn); and blade, drivetrain, nacelle, and tower structural dynamics (ElastoDyn). A new structural-dynamics module (BeamDyn) enables the modeling of advanced aeroelastically tailored blades, including large deflection, anisotropic composite material couplings, and a reference axis that permits blades that are not straight. Additional FAST modules (not further discussed in this paper) support offshore multimember substructures, hydrodynamics, mooring statics and dynamics, and sea-ice dynamics.

The primary use of FAST is to run nonlinear time-domain simulations e.g. for design-standards-based loads analysis. Although most physics involved in wind energy are nonlinear, linearization of the underlying nonlinear system equations is often advantageous to understand the system response and exploit well-established methods and tools for analyzing linear systems. For example, linear state-space models can be transformed to transfer functions, impulse-response functions, or frequency-response functions. The ability to generate linearized models is important for eigenanalysis (to derive structural natural frequencies, damping ratios, and mode shapes), controls design (based on linear state-space models), stability analysis, gradients for optimization problems, and support for the development of reduced-order models. Even though the FAST modularization framework was originally designed with the intent of enabling full-system linearization across all coupled (aero-hydro-servo-elastic) modules, the linearization functionality had not been implemented prior to now.[1]

The focus of the linearization effort to date has been on (1) structuring the FAST v8 source code to enable linearization; (2) developing the general approach to linearizing the mesh-mapping within the module-to-module input-output coupling relationships, including rotations; (3) linearizing core (but not all) features of the InflowWind, AeroDyn, ServoDyn, and ElastoDyn modules and their coupling; and (4) verifying this implementation through application to sample cases. Linearization functionality for other important features, BeamDyn, and offshore wind turbines will be added in the future.

This paper presents the development and verification of the new linearization functionality of FAST v8 for land-based wind turbines, as well as the concepts and mathematical background needed to understand and apply it correctly. Without going into details on specific modules or the mesh-mapping, the overall linearization approach that the FAST modularization framework was designed to support is explained in [2] and is consistent with the present implementation. Without replicating most of the information, this paper uses the same approach and nomenclature of [2], adding details about the linearization to date of specific modules and their coupling, including mesh-mapping.

2. Approach and methods

The linearization of FAST v8 involves (1) finding an operating point (OP), (2) linearizing the underlying nonlinear equations of each module about the OP, (3) linearizing the module-to-module input-output coupling relationships in the FAST glue code about the OP, and (4) combining all linearized matrices into the full-system linear state-space model and exporting those matrices and the OP to a file. Each step is highlighted in the subsections below.

2.1. Operating-point determination

OP (or fixed-point) determination is an important first step in the linearization process because a linear representation of a nonlinear system is only valid for small deviations (perturbations) from an OP. In the current release of FAST v8, an OP can be defined by given initial conditions (time zero) or a given time (or times) in the nonlinear time-marching process. Introducing methods to find a static-equilibrium (constant displacement), steady-state (constant velocity), or periodic steady-state (variation in response with rotor-azimuth angle) condition will be considered in the future, including an optional trim calculation, whereby a control input (nacelle yaw, generator torque, or blade pitch) is varied to achieve a desired rotor speed set point based on a simple proportional feedback control law on the rotor-speed error. It is usually important for the OP to be a static-equilibrium condition (for parked/idling turbines) or steady-state condition (for operating turbines); otherwise, the OP may have an undesirable effect on the linear system matrices.

An OP is defined by given values for the continuous-time states, $x\big|_{op}$; discrete-time states, $x^d\big|_{op}$; inputs, $u\big|_{op}$; and time, $t\big|_{op}$ for each module; equations (1a), (1c), and (1d) from [2] can then be used to

[1] The linearization functionality of older versions of FAST (v4 through v7) was limited to structural linearization only, requiring disabling or assuming a quasi-static treatment of other physics.

calculate the OP values of the first time derivative of the continuous-time states, $\dot{x}\big|_{op}$; constraint (algebraic) states, $z\big|_{op}$; and outputs, $y\big|_{op}$ for each module. Each of these variables can be perturbed (represented by Δ) about their respective OP values as given by equation (11) from [2] e.g. for module inputs $u = u\big|_{op} + \Delta u$. One complication in the linearization process is dealing with rotations (orientations) in three dimensions (3D), which do not reside in a linear space. As explained in [3], module inputs and outputs involving rotations in 3D in the FAST modularization framework are expressed as 3 by 3 direction-cosine matrices (DCMs), Λ, and the mapping transfer involves the use of a logarithmic map to convert DCMs to three rotational parameters and a matrix exponential to convert back to DCMs. In terms of DCMs, a rotational perturbation about the OP orientation can be written as

$$\Lambda = \Lambda\big|_{op}\, \Delta\Lambda\,, \tag{1}$$

where $\Delta\Lambda$ is the DCM perturbation. Regardless of the OP DCM that may be based on large rotations, $\Lambda\big|_{op}$, the rotational-parameter perturbations in the linearization process of the FAST modularization framework are taken to be the three small-angle rotations about the global *X*, *Y*, and *Z* axes of the inertia frame, $\Delta\vec{\theta} = \begin{Bmatrix} \Delta\theta_X \\ \Delta\theta_Y \\ \Delta\theta_Z \end{Bmatrix}$ (order is not important because the angles are small). As shown in [4], the DCM resulting from this rotational parameterization is

$$\Delta\Lambda = \begin{bmatrix} \frac{\Delta\theta_X^2\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}+\Delta\theta_Y^2+\Delta\theta_Z^2}{\left(\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2\right)\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}} & \frac{\Delta\theta_Z\left(\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2\right)+\Delta\theta_X\Delta\theta_Y\left(\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}-1\right)}{\left(\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2\right)\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}} & \frac{-\Delta\theta_Y\left(\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2\right)+\Delta\theta_X\Delta\theta_Z\left(\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}-1\right)}{\left(\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2\right)\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}} \\ \frac{-\Delta\theta_Z\left(\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2\right)+\Delta\theta_X\Delta\theta_Y\left(\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}-1\right)}{\left(\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2\right)\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}} & \frac{\Delta\theta_X^2+\Delta\theta_Y^2\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}+\Delta\theta_Z^2}{\left(\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2\right)\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}} & \frac{\Delta\theta_X\left(\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2\right)+\Delta\theta_Y\Delta\theta_Z\left(\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}-1\right)}{\left(\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2\right)\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}} \\ \frac{\Delta\theta_Y\left(\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2\right)+\Delta\theta_X\Delta\theta_Z\left(\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}-1\right)}{\left(\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2\right)\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}} & \frac{-\Delta\theta_X\left(\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2\right)+\Delta\theta_Y\Delta\theta_Z\left(\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}-1\right)}{\left(\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2\right)\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}} & \frac{\Delta\theta_Z^2+\Delta\theta_Y^2+\Delta\theta_Z^2\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}}{\left(\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2\right)\sqrt{1+\Delta\theta_X^2+\Delta\theta_Y^2+\Delta\theta_Z^2}} \end{bmatrix}, \tag{2}$$

$$\approx \begin{bmatrix} 1 & \Delta\theta_Z & -\Delta\theta_Y \\ -\Delta\theta_Z & 1 & \Delta\theta_X \\ \Delta\theta_Y & -\Delta\theta_X & 1 \end{bmatrix}$$

where $\approx$ denotes linearization.[2] A function to return the nonlinear form of the DCM perturbation is written as $\Delta\Lambda = f_{\Delta\Lambda}\left(\Delta\vec{\theta}\right)$ and its functional inverse is written as $\Delta\vec{\theta} = f_{\Delta\Lambda}^{-1}\left(\Delta\Lambda\right)$. The linearized form of the DCM perturbation can be written equivalently in terms of the cross-product function, which is a function that returns a 3 by 3 skew-symmetric matrix for cross-product calculation via matrix multiplication (valid for any vector $\Delta\vec{\theta}$)—that is

$$\Delta\Lambda \approx I - f_\times\left(\Delta\vec{\theta}\right) \text{ with } f_\times\left(\Delta\vec{\theta}\right) = \begin{bmatrix} 0 & -\Delta\theta_Z & \Delta\theta_Y \\ \Delta\theta_Z & 0 & -\Delta\theta_X \\ -\Delta\theta_Y & \Delta\theta_X & 0 \end{bmatrix}. \tag{3}$$

[2] Although this information is not needed in the linearization process, it is interesting to note for small rotational perturbations, that the angle of rotation used within the logarithmic map [3] is $\theta \approx \sqrt{\Delta\theta_X^2 + \Delta\theta_Y^2 + \Delta\theta_Z^2}$ and the logarithmic map itself is $\vec{\lambda} \approx -\Delta\vec{\theta}$, where $\approx$ denotes application of small-angle approximations.

The linearized module inputs and outputs of FAST v8 are written in terms of $\Delta\vec{\theta}$, rather than $\Delta\Lambda$, and the cross-product function, $f_\times$, as well as nonlinear functions $f_{\Delta\Lambda}$ and $f_{\Delta\Lambda}^{-1}$, appear in many of the linearized equations given in subsequent sections. (In subsequent equations, $=$ is used to denote linearization in place of $\approx$.)

2.2. Module linearization

As explained in [2], the FAST modularization framework supports a very general (need-not-be-linear) state-space formulation, with any combination of continuous-time-state, discrete-time-state, constraint- (algebraic-) state, other- (e.g. logical) state, and output equations. However, for a module to support linearization, the formulation is limited to a hybrid semiexplicit differential-algebraic equation (DAE) of index 1,[3] which has the following limitations: (1) the continuous-time state derivatives and discrete-time state updates must be written as an explicit function of the states, inputs, and parameters; (2) the constraints must be of index 1; and (3) other states are used only for time-integration or when acting as parameters in the linearization process.

To support linearization, a module must also be able to export Jacobian matrices for the state and output equations with respect to the states and inputs. The FAST module states, inputs, and outputs kept in the linearization process for the features linearized to date are summarized in table 1. The FAST module features linearized to date include only continuous-time and constraint states (no features with discrete-time states have yet been linearized).

The linearized form of a general module is given by equations (12) and (13) from [2]; the simplified forms for each module linearized to date are given next, along with a description of how each module is linearized.

The InflowWind (IfW) module has no states; linearization is permitted only when steady or uniform wind is enabled. The linearized form is given by $\Delta y^{(IfW)} = D^{(IfW)}\Delta u^{(IfW)}$, with the input-transmission matrix, $D^{(IfW)} = \left.\dfrac{\partial Y}{\partial u}\right|_{op}^{(IfW)}$, where the Jacobian of the output equations relative to the inputs about the OP is formed analytically.

The internal nacelle-yaw actuator and generator models of the ServoDyn (SrvD) module have no states; linearization is not permitted when the high-speed-shaft brake model, tuned-mass-damper (TMD) models, external user-specified controllers, and control logic for start-up and shut-down maneuvers are enabled. The linearized form is given by $\Delta y^{(SrvD)} = D^{(SrvD)}\Delta u^{(SrvD)}$, with the input-transmission matrix, $D^{(SrvD)} = \left.\dfrac{\partial Y}{\partial u}\right|_{op}^{(SrvD)}$, where the Jacobian of the output equations relative to the inputs about the OP is formed analytically. The input-transmission matrix contains the stiffness and damping of the internal nacelle-yaw actuator and generator models.

The ElastoDyn (ED) module has continuous-time states, there are no restrictions to linearization, and the linearized form of the equations of motion and output equations is given by

$$\Delta\dot{x}^{(ED)} = A^{(ED)}\Delta x^{(ED)} + B^{(ED)}\Delta u^{(ED)} \text{ and} \tag{4a}$$

$$\Delta y^{(ED)} = C^{(ED)}\Delta x^{(ED)} + D^{(ED)}\Delta u^{(ED)}. \tag{4b}$$

[3] Indeed, linearization is not possible or desirable for all models, but only those limited to this restricted form. Most models are simplifications of this most general form, e.g., for a module without discrete-time and constraint states, the continuous-time state equations form ordinary differential equations.

Table 1. Module states, inputs, and outputs kept in the FAST linearization process to date.

Module	States	Inputs	Outputs
InflowWind (IfW)	• None	• Positions where the undisturbed (inflow) wind will be output • Disturbances of horizontal wind speed, power-law shear exponent, and wind-propagation direction	• Undisturbed (inflow) wind velocity at input positions • User-selected wind-inflow outputs
ServoDyn (SrvD)	• None	• Nacelle-yaw angle and rate • Generator speed	• Blade-pitch-angle command (independent) • Nacelle-yaw moment • Generator torque and electrical power • User-selected control and electrical-drive outputs
ElastoDyn (ED)	• Structural degrees-of-freedom (DOFs) and their first time derivatives (continuous states)	• Applied point forces and moments distributed along the blades and tower • Applied point forces and moments lumped on the hub, nacelle, and platform • Blade-pitch-angle command (both independent and rotor-collective) • Nacelle-yaw moment • Generator torque	• Translational displacements, orientations, translational and rotational velocities, and translational and rotation accelerations of points along the blades and tower • Translational displacements, orientations, translational and rotational velocities, and translational and rotation accelerations of the blade-root, nacelle, and platform reference points • Translational displacement, orientation, and rotational velocity of the hub reference point • Nacelle-yaw angle and rate • Generator speed • User-selected structural outputs (motions and/or loads)
AeroDyn (AD)	• Inflow angle at each blade analysis node/airfoil (constraint states)	• Translational displacements, orientations, and translational velocities of analysis nodes along the blades and tower • Orientation of the blade-root reference point • Translational displacement, orientation, and rotational velocity of the hub reference point • Undisturbed (inflow) wind velocities at analysis nodes along the blades and tower	• Aerodynamic applied line (per-unit length) forces and moments distributed along the blades and tower • User-selected aerodynamic outputs

The continuous-state matrix, input matrix, continuous-state output matrix, and the input-transmission matrix are the Jacobians of the state and output equations relative the states and inputs about the OP, computed numerically via a central-difference perturbation technique:

$$A^{(ED)} = \left.\frac{\partial X}{\partial x}\right|_{op}^{(ED)} = \left.\frac{X\left(x|_{op} + \Delta x, u|_{op}, t|_{op}\right) - X\left(x|_{op} - \Delta x, u|_{op}, t|_{op}\right)}{2\Delta x}\right|^{(ED)}, \tag{5a}$$

$$B^{(ED)} = \left.\frac{\partial X}{\partial u}\right|_{op}^{(ED)} = \left.\frac{X\left(x|_{op}, u|_{op} + \Delta u, t|_{op}\right) - X\left(x|_{op}, u|_{op} - \Delta u, t|_{op}\right)}{2\Delta u}\right|^{(ED)}, \tag{5b}$$

$$C^{(ED)} = \left.\frac{\partial Y}{\partial x}\right|_{op}^{(ED)} = \left.\frac{Y\left(x|_{op} + \Delta x, u|_{op}, t|_{op}\right) - Y\left(x|_{op} - \Delta x, u|_{op}, t|_{op}\right)}{2\Delta x}\right|^{(ED)}, \text{ and} \qquad (5c)$$

$$D^{(ED)} = \left.\frac{\partial Y}{\partial u}\right|_{op}^{(ED)} = \left.\frac{Y\left(x|_{op}, u|_{op} + \Delta u, t|_{op}\right) - Y\left(x|_{op}, u|_{op} - \Delta u, t|_{op}\right)}{2\Delta u}\right|^{(ED)}, \qquad (5d)$$

where $X^{(ED)}$ are the continuous-state functions and $Y^{(ED)}$ are the output functions of ElastoDyn. For outputs that are rotations in 3D, i.e., $Y = \Lambda$ and $\Delta y = \Delta\vec{\theta}$, it is implied that the right-hand side (RHS) of equation (5c) is rewritten as $\left.\frac{f_{\Delta\Lambda}^{-1}\left(\left[\Lambda\left(x|_{op} - \Delta x, u|_{op}, t|_{op}\right)\right]^T \Lambda\left(x|_{op} + \Delta x, u|_{op}, t|_{op}\right)\right)}{2\Delta x}\right|^{(ED)}$ and the RHS of equation (5d) is rewritten as $\left.\frac{f_{\Delta\Lambda}^{-1}\left(\left[\Lambda\left(x|_{op}, u|_{op} - \Delta u, t|_{op}\right)\right]^T \Lambda\left(x|_{op}, u|_{op} + \Delta u, t|_{op}\right)\right)}{2\Delta u}\right|^{(ED)}$, respectively, where ^{T} denotes a matrix transpose. The default perturbation sizes are hard-coded within ElastoDyn (but can be customized by recompiling) and are 2° for rotational states and inputs; fractions of the blade length and tower length for the blade-, tower-, and translational platform-displacement states; fractions of a nominal thrust proportional to blade-length squared for the force inputs; and fractions of a nominal torque proportional to blade-length cubed for the moment inputs. The continuous-state matrix, $A^{(ED)}$ from equation (5a), contains the mass, stiffness, and damping of the structural system.

The AeroDyn (AD) module has constraint (algebraic) states based on quasi-steady blade-element/momentum (BEM) theory or a frozen-wake assumption, whereby the axial and tangential induced velocities, $-V_x a$ and $V_y a'$, are fixed during the linearization process. Linearization is permitted only when unsteady airfoil aerodynamics, including dynamic stall, is disabled. The linearized form is given by $\Delta y^{(AD)} = D^{(AD)} \Delta u^{(AD)}$, with the input-transmission matrix containing the aerodynamic stiffness and damping given by

$$D^{(AD)} = \left[\frac{\partial Y}{\partial u} - \frac{\partial Y}{\partial z}\left[\frac{\partial Z}{\partial Z}\right]^{-1}\frac{\partial z}{\partial u}\right]_{op}^{(AD)}, \qquad (6)$$

where the Jacobians of the state and output equations relative the states and inputs about the OP are computed numerically via a central-difference perturbation technique:

$$\left.\frac{\partial Z}{\partial z}\right|_{op}^{(AD)} = \left.\frac{Z\left(z|_{op} + \Delta z, u|_{op}, t|_{op}\right) - Z\left(z|_{op} - \Delta z, u|_{op}, t|_{op}\right)}{2\Delta z}\right|^{(AD)} \text{ with } \left|\left.\frac{\partial Z}{\partial z}\right|_{op}^{(AD)}\right| \neq 0, \qquad (7a)$$

$$\left.\frac{\partial Y}{\partial z}\right|_{op}^{(AD)} = \left.\frac{Y\left(z|_{op}+\Delta z, u|_{op}, t|_{op}\right) - Y\left(z|_{op}-\Delta z, u|_{op}, t|_{op}\right)}{2\Delta z}\right|^{(AD)}, \tag{7b}$$

$$\left.\frac{\partial Z}{\partial u}\right|_{op}^{(AD)} = \left.\frac{Z\left(z|_{op}, u|_{op}+\Delta u, t|_{op}\right) - Z\left(z|_{op}, u|_{op}-\Delta u, t|_{op}\right)}{2\Delta u}\right|^{(AD)}, \text{ and} \tag{7c}$$

$$\left.\frac{\partial Y}{\partial u}\right|_{op}^{(AD)} = \left.\frac{Y\left(z|_{op}, u|_{op}+\Delta u, t|_{op}\right) - Y\left(z|_{op}, u|_{op}-\Delta u, t|_{op}\right)}{2\Delta u}\right|^{(AD)}, \tag{7d}$$

where $Z^{(AD)}$ are the constraint-state residual functions and $Y^{(AD)}$ are the output functions of AeroDyn. For inputs that are rotations in 3D, i.e., $u = \Lambda$ and $\Delta u = \Delta\vec{\theta}$, it is implied that $u|_{op}^{(AD)} + \Delta u^{(AD)}$, $u|_{op}^{(AD)} - \Delta u^{(AD)}$, and $2\Delta u^{(AD)}$ in equations (7c) and (7d) are rewritten as $\Delta|_{op}^{(AD)} f_{\Delta\Lambda}\left(\Delta\vec{\theta}^{(AD)}\right)$, $\Delta|_{op}^{(AD)} f_{\Delta\Lambda}\left(-\Delta\vec{\theta}^{(AD)}\right)$, and $2\Delta\vec{\theta}^{(AD)}$, respectively. As shown in equation (6), the constraint-state (algebraic) equations are eliminated from the linearized system because, once linearized, the constraint-state equations can be easily solved for the perturbations of constraint states and essentially eliminated as separate variables. Because each annulus/analysis node of the BEM solution is solved independently, the Jacobian $\left.\frac{\partial Z}{\partial z}\right|_{op}^{(AD)}$ is a diagonal matrix. It is easily shown that its determinant from equation (7a) is nonzero, which means that the matrix inverse, $\left[\left.\frac{\partial Z}{\partial z}\right|_{op}^{(AD)}\right]^{-1}$ from equation (6), exists, is bounded in the neighborhood around the OP, and is easy to calculate (because it is also a diagonal matrix). The default perturbation sizes are hard-coded within AeroDyn (but can be customized by recompiling) and are 2˚ for rotational states and inputs and fractions of the blade length and tower length for the blade- and tower-displacement inputs. However, because of undefined regions and jump-discontinuities in the BEM solution space, AeroDyn does not permit perturbations in the linearization process that would cause the BEM solution region to change. When this happens (because the OP is too close to a boundary of the given solution region), the perturbation that would have caused the BEM solution region to change is neglected, and the numerical differentiation is limited to a one-sided perturbation.

2.3. Module-to-module input-output coupling relationships linearization

Most module inputs and outputs in FAST reside on spatial boundaries, which in the modularization framework are defined in terms of a mesh that consists of (1) nodes and point or line elements (nodal connectivity); (2) nodal reference locations (position and orientation); and (3) one or more nodal fields, expressing the value of the input or output on the spatial boundary e.g. motion (displacements, orientations, velocities, accelerations), loads (forces, moments), and/or scalar quantities. The module-to-module input-output coupling relationships in the FAST glue code are algebraic, and include spatial mesh-to-mesh mapping involving two steps as detailed in [3]: (1) a mapping search where nearest-neighbor nodes/elements are found between source and destination meshes and (2) a mapping transfer where nodal field quantities are transferred to the destination mesh from the mapped nodes of the

source mesh. Although the mapping search is unaffected by linearization (because it depends only on system parameters), the linearization of the mapping transfer and other algebraic input-output coupling relationships in the FAST modularization framework is formed analytically about the OP. The appendix provides details on the linearization of the mesh-mapping transfer following the approach and nomenclature of [3].

The linearized input-output transformation functions, U, are given be equation (15) from [2], repeated here for convenience:

$$0 = \left.\frac{\partial U}{\partial \tilde{u}}\right|_{op} \Delta u + \left.\frac{\partial U}{\partial y}\right|_{op} \Delta y \quad \text{with} \quad \left|\left.\frac{\partial U}{\partial \tilde{u}}\right|_{op}\right| \neq 0 \,. \tag{8}$$

As is evident from table 1, the InflowWind, ServoDyn, ElastoDyn, and AeroDyn modules were developed so that for the most part—other than mapping between independent spatial discretizations—the input of one module equals the output of another. It follows that with $U = \begin{Bmatrix} U^{(IfW)} \\ U^{(SrvD)} \\ U^{(ED)} \\ U^{(AD)} \end{Bmatrix}$, $\Delta u = \begin{Bmatrix} \Delta u^{(IfW)} \\ \Delta u^{(SrvD)} \\ \Delta u^{(ED)} \\ \Delta u^{(AD)} \end{Bmatrix}$, and $\Delta y = \begin{Bmatrix} \Delta y^{(IfW)} \\ \Delta y^{(SrvD)} \\ \Delta y^{(ED)} \\ \Delta y^{(AD)} \end{Bmatrix}$, the Jacobian matrices evaluated at the OP from equation (8) for these four modules are given by

$$\left.\frac{\partial U}{\partial \tilde{u}}\right|_{op} = \begin{bmatrix} I & 0 & 0 & \dfrac{\partial U^{(IfW)}}{\partial \tilde{u}^{(AD)}} \\ 0 & I & 0 & 0 \\ 0 & 0 & I & \dfrac{\partial U^{(ED)}}{\partial \tilde{u}^{(AD)}} \\ 0 & 0 & 0 & \dfrac{\partial U^{(AD)}}{\partial \tilde{u}^{(AD)}} \end{bmatrix}_{op} \quad \text{and} \quad \left.\frac{\partial U}{\partial y}\right|_{op} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \dfrac{\partial U^{(SrvD)}}{\partial y^{(ED)}} & 0 \\ 0 & \dfrac{\partial U^{(ED)}}{\partial y^{(SrvD)}} & \dfrac{\partial U^{(ED)}}{\partial y^{(ED)}} & \dfrac{\partial U^{(ED)}}{\partial y^{(AD)}} \\ \dfrac{\partial U^{(AD)}}{\partial y^{(IfW)}} & 0 & \dfrac{\partial U^{(AD)}}{\partial y^{(ED)}} & 0 \end{bmatrix}_{op} , \tag{9}$$

where I and 0 are appropriately sized identity and zero matrices and the sub-Jacobian matrices are composed of Is, 0s, and the linearized matrices from the mapping transfers given in the appendix. Due to their large size, the sub-Jacobian matrices are not shown here, but are described qualitatively instead:

- The first equation of (8) for InflowWind inputs, $0 = U^{(IfW)}$, expresses that the position perturbations where the undisturbed (inflow) wind will be output as input to InflowWind are derived from the translational-displacement perturbations of analysis nodes along the blades and tower as input to AeroDyn.
- The second equation of (8) for ServoDyn inputs, $0 = U^{(SrvD)}$, expresses that the nacelle-yaw-angle and -rate and generator-speed perturbations as input to ServoDyn are derived from the equivalent outputs from ElastoDyn.
- The third equation of (8) for ElastoDyn inputs, $0 = U^{(ED)}$, expresses that the applied point force and moment perturbations distributed along the blades and tower as input to ElastoDyn are derived from the aerodynamic applied line (per-unit length) force and moment perturbations distributed along the blades and tower as output from AeroDyn. This linearized load-mapping transfer also depends on the translational-displacement perturbations of analysis nodes along the blades and tower as input to AeroDyn and output from ElastoDyn.

Additionally, the blade-pitch-angle-commands, nacelle-yaw-moment, and generator-torque perturbations as input to ElastoDyn are derived from the equivalent outputs from ServoDyn.

- The fourth equation of (8) for AeroDyn inputs, $0 = U^{(AD)}$, expresses that the translational-displacement, orientation, translational-velocity, and rotational-velocity perturbations of analysis nodes along the blades and tower, blade-root reference point, and hub reference point as input to AeroDyn are derived from the equivalent outputs from ElastoDyn. Additionally, the undisturbed (inflow) wind-velocity perturbations at analysis nodes along the blades and tower as input to AeroDyn are derived from the equivalent outputs from InflowWind.

The Jacobian $\left.\frac{\partial U}{\partial \tilde{u}}\right|_{op}$ has ones along its entire diagonal and it is easily shown that its determinant from equation (8) is nonzero.

2.4. Final matrix assembly

Once all individual modules and input-output relationships are linearized about the OP, the linearized model of the complete coupled system can be assembled. Linearization of the full-system model produces a linear state-space model representation of the complete nonlinear system about the OP, including the influence of system state and input perturbations on the system response and outputs. The general linearized form of the complete coupled system is given by equations (18) and (19) from [2]. With $\Delta x = \Delta x^{(ED)}$, the simplified form for the FAST v8 features linearized to date (without discrete-time states and with ElastoDyn as the only module with continuous-time states) is

$$\Delta \dot{x} = A\Delta x + B\Delta u^{+} \text{ and} \tag{10a}$$

$$\Delta y = C\Delta x + D\Delta u^{+}, \tag{10b}$$

where Δu^{+} are the additional input perturbations (explained further in [2]) and where

$$A = A^{(ED)} - \begin{bmatrix} 0 & 0 & B^{(ED)} & 0 \end{bmatrix} \left[G\big|_{op} \right]^{-1} \left.\frac{\partial U}{\partial y}\right|_{op} \begin{bmatrix} 0 \\ 0 \\ C^{(ED)} \\ 0 \end{bmatrix}, \tag{11a}$$

$$B = \begin{bmatrix} 0 & 0 & B^{(ED)} & 0 \end{bmatrix} \left[G\big|_{op} \right]^{-1} \left.\frac{\partial U}{\partial \tilde{u}}\right|_{op}, \tag{11b}$$

$$C = \begin{bmatrix} 0 \\ 0 \\ C^{(ED)} \\ 0 \end{bmatrix} - \begin{bmatrix} D^{(IfW)} & 0 & 0 & 0 \\ 0 & D^{(SrvD)} & 0 & 0 \\ 0 & 0 & D^{(ED)} & 0 \\ 0 & 0 & 0 & D^{(AD)} \end{bmatrix} \left[G\big|_{op} \right]^{-1} \left.\frac{\partial U}{\partial y}\right|_{op} \begin{bmatrix} 0 \\ 0 \\ C^{(ED)} \\ 0 \end{bmatrix}, \text{ and} \tag{11c}$$

$$D = \begin{bmatrix} D^{(IfW)} & 0 & 0 & 0 \\ 0 & D^{(SrvD)} & 0 & 0 \\ 0 & 0 & D^{(ED)} & 0 \\ 0 & 0 & 0 & D^{(AD)} \end{bmatrix} \left[G\big|_{op} \right]^{-1} \left.\frac{\partial U}{\partial \tilde{u}}\right|_{op}, \tag{11d}$$

where $G\big|_{op}$ —explained more in [2] for the general case—for the FAST v8 linearization to date is

$$G\big|_{op} = \left[\begin{array}{cccc} I & 0 & 0 & \frac{\partial U^{(IfW)}}{\partial \tilde{u}^{(AD)}} \\ 0 & I & \frac{\partial U^{(SrvD)}}{\partial y^{(ED)}} D^{(ED)} & 0 \\ 0 & \frac{\partial U^{(ED)}}{\partial y^{(SrvD)}} D^{(SrvD)} & I + \frac{\partial U^{(ED)}}{\partial y^{(ED)}} D^{(ED)} & \frac{\partial U^{(ED)}}{\partial \tilde{u}^{(AD)}} + \frac{\partial U^{(ED)}}{\partial y^{(AD)}} D^{(AD)} \\ \frac{\partial U^{(AD)}}{\partial y^{(IfW)}} D^{(IfW)} & 0 & \frac{\partial U^{(AD)}}{\partial y^{(ED)}} D^{(ED)} & \frac{\partial U^{(AD)}}{\partial \tilde{u}^{(AD)}} \end{array}\right]\Bigg|_{op} \quad \text{with } \left|G\big|_{op}\right| \neq 0 . \quad (12)$$

The matrix $G\big|_{op}$ has ones along its entire diagonal and it is easily shown that its determinant from equation (12) is nonzero, which means that the matrix inverse, $\left[G\big|_{op}\right]^{-1}$ from equation (11), exists and is bounded in the neighborhood around the OP.

The input-transmission matrices impact all matrices of the linearized coupled system, highlighting the important role played by direct feedthrough of input to output in the coupled system response. For example, while the continuous-state matrix of ElastoDyn, $A^{(ED)}$, contains stiffness and damping only directly associated with the structural model, the full-system continuous state matrix, A, contains stiffness and damping associated with coupled aeroelastics.

When the linearized full-system matrices A, B, C, and D are exported to a file by FAST, the additional input perturbations, Δu^{+}, can be chosen by the user to be (1) the inputs of all modules, (2) none of the module inputs (removing B and D from the file), or (3) a standard subset of these inputs, which include the standard wind turbine control inputs of nacelle-yaw moment, generator torque, and blade-pitch-angle commands (both independent and rotor-collective) and the standard wind-inflow disturbances of horizontal wind speed, power-law shear exponent, and wind-propagation direction. Likewise, the output perturbations, Δy, can be chosen by the user to be (1) the outputs of all modules, (2) none of the module outputs (removing C and D from the file), or (3) only the subset of output variables selected by the user through the FAST module input file(s). Regardless of what the user selects to be exported to a file, all of the module inputs and outputs are used to form the linearized full-system matrices in equation (11), but only a subset of these matrices are exported based on the user selection.

3. Results

The results of two sample cases are presented to highlight the functionality and verification of the new linearization capability of FAST v8 for land-based wind turbines, in this case the NREL 5-MW baseline wind turbine [5]. The first case involves the calculation of the Campbell diagram for the turbine spinning in a vacuum whereby the full-system natural frequencies are calculated as a function of rotor speed in the absence of aerodynamic loading. The second case involves the calculation of the aerodynamic derivatives of power and thrust with respect to rotor-collective blade-pitch angle and wind speed. In both cases, the results from FAST v8 have been verified by comparing to those derived from the older linearization functionality of FAST v6. Consistency between the results from the new and old implementations was achieved, so, only those of FAST v8 are presented.

3.1. Campbell diagram

To compute the full-system natural frequencies of the NREL 5-MW baseline turbine as a function of rotor speed in the absence of aerodynamic loading, the ElastoDyn and ServoDyn modules are enabled

and the InflowWind and AeroDyn modules are disabled. All pertinent structural DOFs are enabled in the ElastoDyn module—including blade-bending, drivetrain-torsion, generator-rotation, nacelle-yaw, and tower-bending DOFs—and gravitational loading and structural (but no aerodynamic) damping is included. The ServoDyn module provides nacelle-yaw actuator stiffness and damping. A separate linearization analysis is run at each rotor speed from 0 to 14 rpm, in steps of 2 rpm. The rotor-collective blade-pitch angles were fixed at 0°. For each rotor speed, a periodic steady-state condition is found by marching the nonlinear solution in time long enough for start-up transients to die out (the start-up transients exist because gravity and rotor-rotation have an influence on the structural displacements, which are assumed undisplaced at time zero, and die out as a result of structural damping). After a periodic steady-state condition is reached, one more rotor (360°) revolution is simulated, and an OP is set and the linearized full-system state matrix (A) is computed for each azimuth angle of the rotor in 36 steps of 10°. The multiblade coordinate (MBC) postprocessor [6] is used to read-in the periodic state matrices, apply the MBC transformation to the rotating states to transform them to states in the fixed frame, azimuth-average the MBC-transformed state matrices, compute the eigensolution of the resulting azimuth-averaged matrix, and extract the natural (undamped) frequencies from this eigensolution. The results are presented in figure 1.

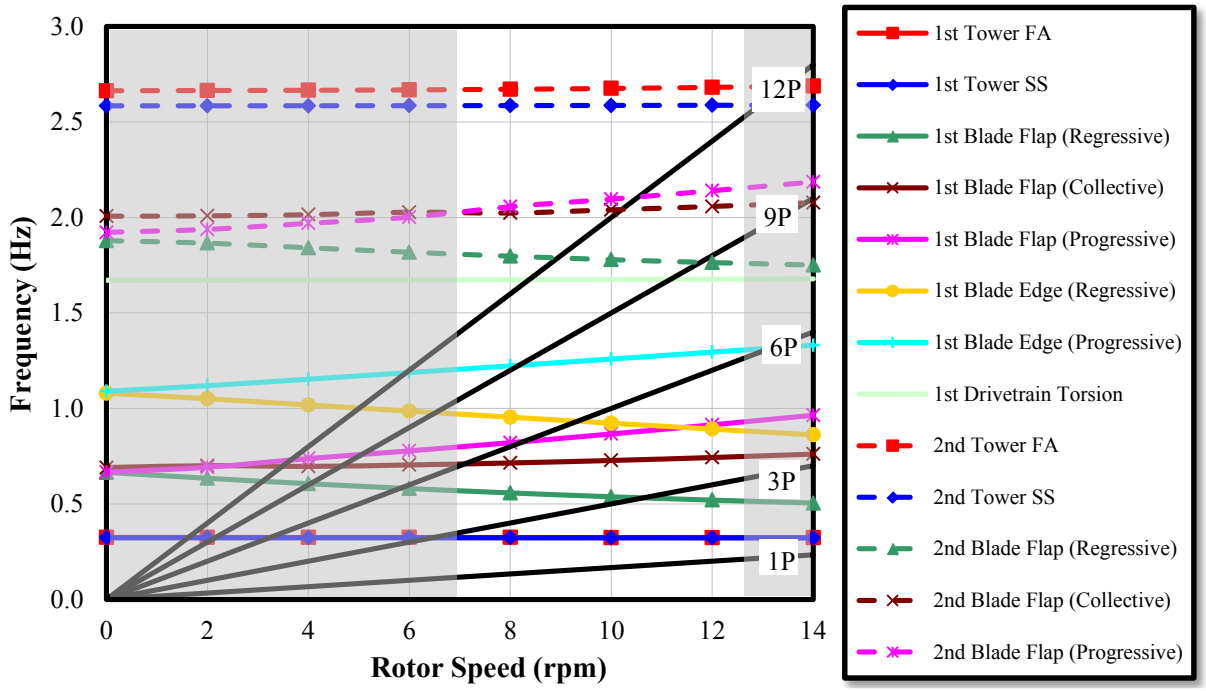

Figure 1. Campbell diagram of the NREL 5-MW baseline wind turbine spinning in a vacuum.

In figure 1, the grayed-out areas denote rotor speeds outside the operational range of the NREL 5-MW baseline wind turbine and the solid black lines denote the frequencies associated with once-per-revolution (1P), three-per-revolution (3P), etc. for 6P, 9P, and 12P, frequencies in which aerodynamic excitation and rotation-induced forcing is expected. The natural frequencies are identified by their primary mode, but as these are full-system modes, most modes include coupling of all DOFs. As expected, the tower fore-aft- (FA-) bending and side-to-side- (SS-) bending and free-free drivetrain torsion modes have natural frequencies that are quite independent of rotor speed. However, the blade flapwise- (flap-) and edgewise- (edge-) bending are strongly dependent on rotor speed for both the first- (1st-) and second (2nd-) modes. The term "collective" in figure 1 is used to indicate all blades of the rotor bending in phase, whereas "progressive" and "regressive" are used to indicate modes where the blades bend asymmetrically and the center of mass of the rotor whirls in the same and opposite direction as the rotor rotation, respectively. The zero-frequency rigid-body mode of the generator is

not shown in figure 1, as well as the nacelle-yaw mode and the additional first blade edgewise-bending mode, which have frequencies above 3 Hz. The soft-stiff design of the tower is clearly visible, i.e., the first tower-bending natural frequencies fall within 1P and 3P in the operational range of rotor speeds. Also, in the operational range of rotor speeds, potential resonances exist that may benefit from active control for the first blade flapwise-bending at 3P and 6P, for the first blade edgewise-bending at 6P and 9P, and for the second blade flapwise-bending and drivetrain-torsion at 9P and 12P. These are insights into the physical response of the turbine that would be very difficult to attain without linearization capability.

3.2. Aerodynamic derivatives

To compute the aerodynamic derivatives of the NREL 5-MW baseline turbine, the ElastoDyn, InflowWind, and AeroDyn modules are enabled, and the ServoDyn module is disabled. While the ElastoDyn module is enabled (to specify the turbine geometry, rotor speed, and rotor-collective blade-pitch angle), all structural DOFs are disabled to make the turbine rigid and (for simplification) to eliminate the aeroelastic influence on the aerodynamic derivatives. Steady uniform wind without shear is set in the InflowWind module. In the AeroDyn module, BEM theory with the Pitt and Peters skewed-wake correction model is enabled, but tower influence and unsteady airfoil aerodynamics are disabled, and separate analyses are run with both frozen wake and with the induction recalculated during the linearization process using BEM theory (equilibrium wake). A separate linearization analysis is run at each wind speed in Region 3 (above-rated) operation from rated (11.4 m/s) to cut-out (25 m/s) wind speeds, in steps of 1 m/s (11.4, 12, 13,…, 25 m/s). The rotor speed was fixed at the rated value of 12.1 rpm and the rotor-collective blade-pitch angle at each wind speed was set to the appropriate value needed to achieve rated aerodynamic power (slightly above 5 MW)—from 0˚ at 11.4 m/s to 23.47˚ at 25 m/s—as documented in [5]. For each wind speed/blade-pitch angle, one rotor (360 ˚) revolution is simulated, and an OP is set and the linearized full-system input-transmission matrix (D) is computed for each azimuth angle of the rotor in 36 steps of 10˚. As the turbine is modeled rigidly and the only states in the FAST v8 model are of the constraint type, there are no start-up transients and the periodic steady-state solution is calculated immediately (periodicity is driven by the small amount of shaft tilt (5˚) of the NREL 5-MW baseline turbine and corresponding skewed-wake). Of all the derivatives available in the input-transmission matrix, only the aerodynamic derivatives of power (P) and thrust (T) with respect to rotor-collective blade-pitch angle (θ) and wind speed (V) are postprocessed. The MBC postprocessor [6] is used to read-in and azimuth-average the periodic input-transmission matrices and the results are presented in figure 2 (no MBC transform is applied because power, thrust, rotor-collective blade-pitch angle, and wind speed are not expressed in the rotating frame).

The derivative of aerodynamic power with respect to rotor-collective blade-pitch angle with frozen wake, presented in the upper-left graph of figure 2, is consistent with what is reported in [5]. This derivative is important for computing the gains of the rotor-collective blade-pitch speed controller and varies considerably (but linearly) over Region 3, which necessitates gain scheduling based on the blade-pitch angle. The derivative with equilibrium wake (not reported in [5]) shows a similar trend to that of frozen wake, but (inaccurately) shows a loss of control authority—whereby the derivative is near zero—at rated conditions. As such, the use of frozen wake in the linearization process is recommended.[4]

The other aerodynamic derivatives presented in figure 2 again provide insights into the physical response of the turbine that would be very difficult to attain without linearization capability. In particular, the aerodynamic derivative of thrust with respect to wind speed presented in the lower-right graph is useful for understanding the influence of aerodynamics on the damping of tower fore-aft motion. The aerodynamic derivative of thrust with respect to rotor-collective blade-pitch angle

[4]Linearization with frozen wake was not a standard feature of older versions of FAST (v4 through v7), but a customized version was adapted with this capability [5].

presented in the lower-left graph is important for computing the gains of an active rotor-collective blade-pitch tower-damping controller. The frozen-wake results presented in figure 2 for these two derivatives are consistent with the results in [7].

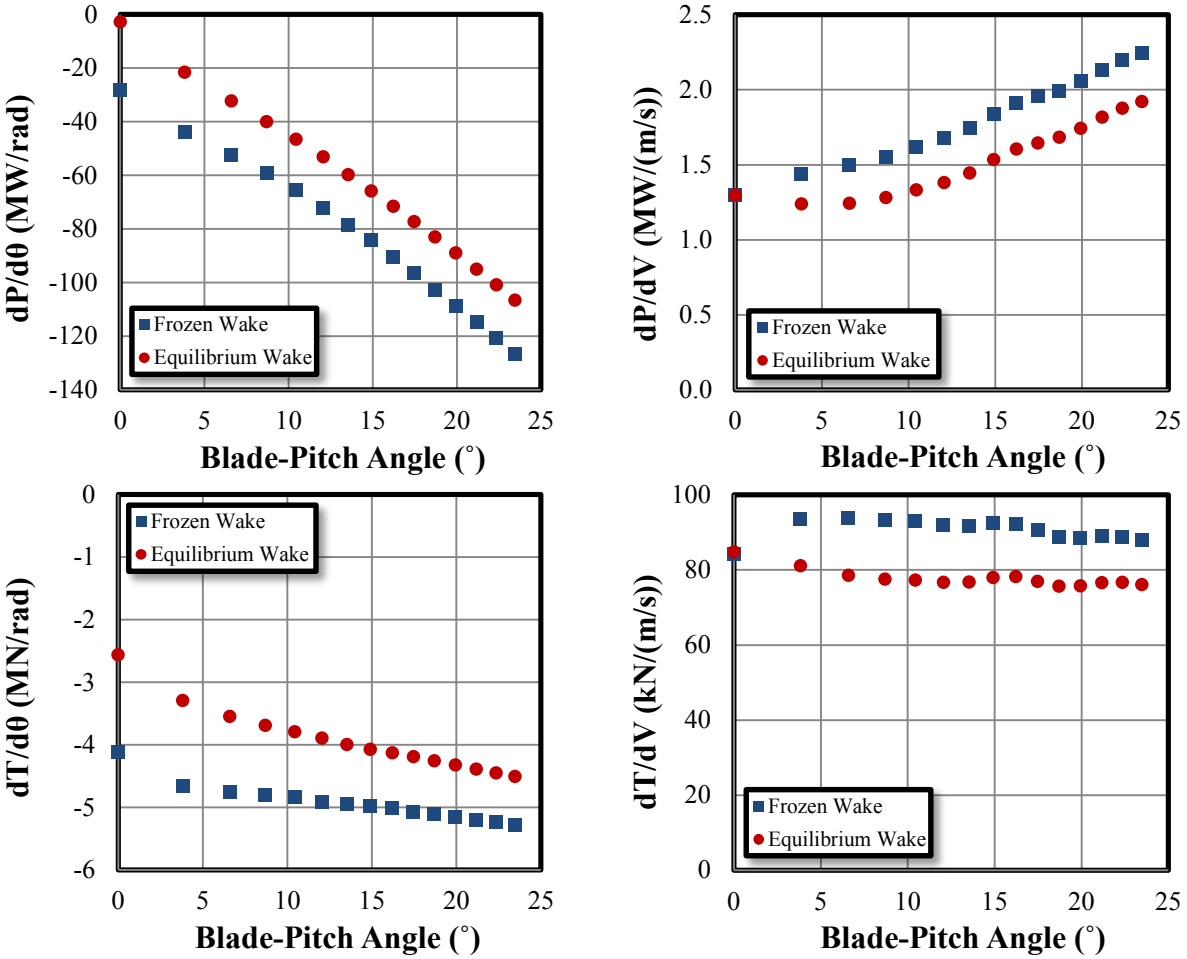

Figure 2. Aerodynamic derivatives of the NREL 5-MW baseline wind turbine in Region 3.

4. Conclusions

Linearization of the underlying nonlinear wind-system equations is often advantageous to understand the system response and exploit well-established methods and tools for analyzing linear systems. The development and verification of the new linearization functionality of the open-source engineering tool FAST v8 for land-based wind turbines has been presented, as well as the concepts and mathematical background needed to understand and apply it correctly. Details have been presented on (1) finding an OP, (2) linearizing the underlying nonlinear equations of each module about the OP, (3) linearizing the module-to-module input-output coupling relationships about the OP, and (4) combining all linearized matrices into the full-system linear state-space model. The presented results highlight the functionality and verification of the new linearization capability. It is envisioned that the new linearization functionality of FAST v8 and other recent enhancements will transform FAST into a powerful, robust, and flexible tool used throughout the wind engineering community for improving wind turbine physical understanding and for developing advanced wind technology.

Beyond the developments implemented here, potential future FAST v8 linearization enhancements include (1) introducing routines to find a static equilibrium, steady-state, or periodic steady-state condition for improved OP determination; (2) enabling the linearization of the TMD models and external user-specified controllers within ServoDyn for improved controls development and analysis;

(3) implementing a form of unsteady airfoil aerodynamics, including dynamic stall, within AeroDyn amendable to linearization e.g. from [8] and including that model within the linearization of AeroDyn for advanced aerodynamic and stability analysis; (4) linearizing BeamDyn for analysis of advanced aeroelastically tailored blades; (5) linearizing the offshore functionality of FAST for analysis of offshore wind systems; and (6) linearizing new features as the FAST models are further developed in the future.

Acknowledgements
The contributions of Mike Sprague of NREL to the development of the FAST modularization framework and to reviewing the linearization plans are greatly acknowledged.

This work was supported by the U.S. Department of Energy under Contract No. DE-AC36-08GO28308 with NREL. Funding for the work was provided by the DOE Office of Energy Efficiency and Renewable Energy, Wind and Water Power Technologies Office.

Appendix
The theoretical formulation of the mesh-to-mesh mapping algorithms for Point_to_Point, Line2_to_Line2, Point_to_Line2, and Line2_to_Point meshes is explained in the appendix of [3]. Without replicating most of the information, this appendix describes the linearized form of the mesh-mapping transfer equations using the same approach and nomenclature of [3].

Point_to_Point mapping transfer
The linearized mapping transfer of translational displacement perturbation is given by

$$\Delta\vec{u}^{D} = \left[I \quad f_{\times}\left(\vec{p}^{S} - \vec{p}^{D}\right)\right]\Big|_{op} \begin{Bmatrix} \Delta\vec{u}^{S} \\ \Delta\vec{\theta}^{S} \end{Bmatrix}, \tag{A.1}$$

where I is the 3 by 3 identity matrix and $\vec{p}^{D}\big|_{op} = \vec{p}^{DR} + \vec{u}^{D}\big|_{op}$ and $\vec{p}^{S}\big|_{op} = \vec{p}^{SR} + \vec{u}^{S}\big|_{op}$ are the absolute displaced positions of a node of the destination mesh and the mapped node of the source mesh at the OP, respectively. The second term on the RHS of equation (A.1) represents the translation displacement of a node of the destination mesh due to rigid-body rotation of the mapped node of the source mesh from its OP orientation and is zero if the rotation of the mapped node of the source mesh is not perturbed or if the absolute displaced positions of the node of the destination mesh and mapped node of the source mesh at the OP are coincident, whereby $\vec{p}^{D}\big|_{op} = \vec{p}^{S}\big|_{op}$.

The linearized mapping transfer of displaced rotation perturbation is given by

$$\Delta\vec{\theta}^{D} = \left[I\right]\Delta\vec{\theta}^{S}, \tag{A.2}$$

where it is clear that the node of the destination mesh will rotate the same amount as the mapped node of the source mesh (as a rigid body).

The linearized mapping transfer of translational and rotation velocity perturbations is given by

$$\Delta\vec{v}^{D} = \left[f_{\times}\left(\vec{\omega}^{S}\right) \quad -f_{\times}\left(\vec{\omega}^{S}\right) \quad I \quad f_{\times}\left(\vec{p}^{S} - \vec{p}^{D}\right)\right]\Big|_{op} \begin{Bmatrix} \Delta\vec{u}^{D} \\ \Delta\vec{u}^{S} \\ \Delta\vec{v}^{S} \\ \Delta\vec{\omega}^{S} \end{Bmatrix} \text{ and} \tag{A.3}$$

$$\Delta\vec{\omega}^{D} = \left[I\right]\Delta\vec{\omega}^{S} . \tag{A.4}$$

The first, second, and fourth terms combined on the RHS of equation (A.3) represent the translation-velocity perturbation of a node of the destination mesh due to the displaced offset between the node of the destination mesh and mapped node of the source mesh and the rotational-velocity perturbation of the mapped node of the source mesh.[5] The node of the destination mesh will rotate the same as the mapped node of the source mesh (as a rigid body).

The linearized mapping transfer of translational- and rotation-acceleration perturbations is given by

$$\Delta\vec{a}^{D} = \left[\left[f_{\times}\left(\vec{\alpha}^{S}\right)+\left\{\vec{\omega}^{S}\right\}\left\{\vec{\omega}^{S}\right\}^{T}-\left\{\vec{\omega}^{S}\right\}^{T}\vec{\omega}^{S}I\right] \quad -\left[f_{\times}\left(\vec{\alpha}^{S}\right)+\left\{\vec{\omega}^{S}\right\}\left\{\vec{\omega}^{S}\right\}^{T}-\left\{\vec{\omega}^{S}\right\}^{T}\vec{\omega}^{S}I\right] \quad \left[f_{\times}\left(\vec{\omega}^{S}\times\left\{\vec{p}^{S}-\vec{p}^{D}\right\}\right)+\left\{\vec{p}^{S}-\vec{p}^{D}\right\}\left\{\vec{\omega}^{S}\right\}^{T}-\left\{\vec{p}^{S}-\vec{p}^{D}\right\}^{T}\vec{\omega}^{S}I\right] \quad I \quad f_{\times}\left(\vec{p}^{S}-\vec{p}^{D}\right)\right]\Big|_{op}\begin{Bmatrix}\Delta\vec{u}^{D}\\ \Delta\vec{u}^{S}\\ \Delta\vec{\omega}^{S}\\ \Delta\vec{a}^{S}\\ \Delta\vec{\alpha}^{S}\end{Bmatrix} \text{ and} \tag{A.5}$$

$$\Delta\vec{\alpha}^{D} = \left[I\right]\Delta\vec{\alpha}^{S} . \tag{A.6}$$

The first part of the first two terms and the fifth term combined on the RHS of equation (A.5) represent the tangential-acceleration perturbation of a node of the destination mesh due to the displaced offset between the node of the destination mesh and mapped node of the source mesh and the rotational-acceleration perturbation of the mapped node of the source mesh. The remaining parts of the first two terms and the third term combined on the RHS of equation (A.5) represent the centripetal-acceleration perturbation of a node of the destination mesh due to the displaced offset between the node of the destination mesh and mapped node of the source mesh and the rotational-velocity perturbation of the mapped node of the source mesh. The node of the destination mesh will rotate the same as the mapped node of the source mesh (as a rigid body).

The linearized mapping transfer of scalar quantity perturbation is given by

$$\Delta S^{D} = \left[I\right]\Delta S^{S} . \tag{A.7}$$

The linearized mapping transfer of concentrated (lumped) force and moment perturbations is given by

$$\Delta\vec{F}^{D} = \sum\left[I\right]\Delta\vec{F}^{S} \text{ and} \tag{A.8}$$

$$\Delta\vec{M}^{D} = \sum\left[f_{\times}\left(\vec{F}^{S}\right) \quad -f_{\times}\left(\vec{F}^{S}\right) \quad f_{\times}\left(\vec{p}^{S}-\vec{p}^{D}\right) \quad I\right]\Big|_{op}\begin{Bmatrix}\Delta\vec{u}^{D}\\ \Delta\vec{u}^{S}\\ \Delta\vec{F}^{S}\\ \Delta\vec{M}^{S}\end{Bmatrix} . \tag{A.9}$$

The first three terms combined on the RHS of equation (A.9) represent the additional moment perturbation of the mapped node of the destination mesh due to the displaced offset (moment arm) between the node of the source mesh and mapped node of the destination mesh and the concentrated force of the node of the source mesh. The summations in equations (A.8) and (A.9) denote the superposition of loads when a destination element has more than one mapped source element.

[5] Rearrangement of equation (A.1) reveals that $\Delta\vec{u}^{D}-\Delta\vec{u}^{S} = f_{\times}\left(\vec{p}^{S}-\vec{p}^{D}\right)\Big|_{op}\Delta\vec{\theta}^{S}$, so the terms involving translational displacements of the nodes of the destination and source meshes in equation (A.3) and other mapping-transfer equations could alternatively be written in terms of rotations of the source mesh.

Line_to_Line mapping transfer

The linearized mapping transfer of all motion and scalar quantity perturbations, via interpolation based on the projected location in the source Line2 element, is given by

$$(\cdot) = (\cdot)_1 \left(1 - \overline{l}^S \right) + (\cdot)_2 \, \overline{l}^S , \tag{A.10}$$

where the $(\cdot)$ on the left-hand side (LHS) of equation (A.10) is a placeholder for $\Delta\vec{u}^D$, $\Delta\vec{\theta}^D$, $\Delta\vec{v}^D$, $\Delta\vec{\omega}^D$, $\Delta\vec{a}^D$, $\Delta\vec{\alpha}^D$, or ΔS^D from the LHS of equations (A.1) through (A.7), respectively, and $(\cdot)$ on the RHS of equation (A.10) is a placeholder for the corresponding RHS of equations (A.1) through (A.7), with subscripts 1 and 2, respectively, denoting the first and second nodes of the mapped Line2 element of the source mesh. $\overline{l}^S$ used in equation (A.10) was solved via equation (36) from the Line2_to_Line2 mapping search for motion and scalar quantities (see Section C.1 of the appendix of [3]). The motion and scalar quantity perturbations are not interpolated if the projection lies on a node, whereby $\overline{l}^S = 0$ or $\overline{l}^S = 1$.

For load quantities, the fields of the new nodes of the augmented source mesh are first populated via interpolation of the fields from the original nodes of the source mesh. That is, equation (A.10) (where $(\cdot)$ is a placeholder) is used to calculate $\Delta\vec{u}^S$, $\Delta\vec{f}^S$, and $\Delta\vec{m}^S$ at the new nodes of the augmented source mesh, where $\overline{l}^S$ was solved via equation (37) from the Line2_to_Line2 mapping search for load quantities (see Section C.2 of the appendix of [3]).

The linearized lumping of distributed force and moment perturbations to concentrated point force and moment perturbations at the two nodes of each Line2 element of the augmented source mesh is given by

$$\Delta\vec{F}_1^S = \sum \frac{\left\| \vec{p}_2^{SR} - \vec{p}_1^{SR} \right\|_2}{6} \begin{bmatrix} 2I & I \end{bmatrix} \begin{Bmatrix} \Delta\vec{f}_1^S \\ \Delta\vec{f}_2^S \end{Bmatrix}, \tag{A.11}$$

$$\Delta\vec{F}_2^S = \sum \frac{\left\| \vec{p}_2^{SR} - \vec{p}_1^{SR} \right\|_2}{6} \begin{bmatrix} I & 2I \end{bmatrix} \begin{Bmatrix} \Delta\vec{f}_1^S \\ \Delta\vec{f}_2^S \end{Bmatrix}, \tag{A.12}$$

$$\Delta\vec{M}_1^S = \sum \frac{\left\| \vec{p}_2^{SR} - \vec{p}_1^{SR} \right\|_2}{6} \left[f_\times\left(\frac{\vec{f}_1^S + \vec{f}_2^S}{2} \right) \quad f_\times\left(\frac{\vec{p}_2^S - \vec{p}_1^S}{2} \right) \quad 2I \quad -f_\times\left(\frac{\vec{f}_1^S + \vec{f}_2^S}{2} \right) \quad f_\times\left(\frac{\vec{p}_2^S - \vec{p}_1^S}{2} \right) \quad I \right]_{lop} \begin{Bmatrix} \Delta\vec{u}_1^S \\ \Delta\vec{f}_1^S \\ \Delta\vec{m}_1^S \\ \Delta\vec{u}_2^S \\ \Delta\vec{f}_2^S \\ \Delta\vec{m}_2^S \end{Bmatrix}, \text{ and} \tag{A.13}$$

$$\Delta\vec{M}_2^S = \sum \frac{\left\| \vec{p}_2^{SR} - \vec{p}_1^{SR} \right\|_2}{6} \left[-f_\times\left(\frac{\vec{f}_1^S + \vec{f}_2^S}{2} \right) \quad -f_\times\left(\frac{\vec{p}_2^S - \vec{p}_1^S}{2} \right) \quad I \quad f_\times\left(\frac{\vec{f}_1^S + \vec{f}_2^S}{2} \right) \quad -f_\times\left(\frac{\vec{p}_2^S - \vec{p}_1^S}{2} \right) \quad 2I \right]_{lop} \begin{Bmatrix} \Delta\vec{u}_1^S \\ \Delta\vec{f}_1^S \\ \Delta\vec{m}_1^S \\ \Delta\vec{u}_2^S \\ \Delta\vec{f}_2^S \\ \Delta\vec{m}_2^S \end{Bmatrix}, \tag{A.14}$$

where the first, second, fourth, and fifth terms combined on the RHS of equations (A.13) and (A.14) represent the additional lumped moment perturbation of a node of the source mesh due to the

distributed force perturbation. The summations in equations (A.11) through (A.14) denote the superposition of loads when a given node of the source mesh is connected to multiple elements.

The linearized mapping transfer of lumped load quantity perturbations via splitting based on the projected location in the mapped destination Line2 element is given by

$$\left(\cdot\right)_1 = \sum\left(\cdot\right)\left(I - \overline{l}^{\,D}\right) \text{ and} \tag{A.15}$$

$$\left(\cdot\right)_2 = \sum\left(\cdot\right)\overline{l}^{\,D}, \tag{A.16}$$

where $\left(\cdot\right)$ on the LHS of equations (A.15) and (A.16) is a placeholder for $\Delta\vec{F}^{\,D}$ and $\Delta\vec{M}^{\,D}$ from the LHS of equations (A.8) and (A.9), respectively, and $\left(\cdot\right)$ on the RHS of equations (A.15) and (A.16) is a placeholder for the corresponding RHS of equations (A.8) and (A.9) (but without the summations), with subscripts 1 and 2, respectively, denoting the first and second nodes of the mapped Line2 element of the destination mesh. $\overline{l}^{\,D}$ used in equations (A.15) and (A.16) was solved via equation (39) from the Line2_to_Line2 mapping search for load quantities (see Section C.2 of the appendix of [3]). The load quantity perturbations are not split if the projection lies on a node, whereby $\overline{l}^{\,D} = 0$ or $\overline{l}^{\,D} = 1$. The summations in equations (A.15) and (A.16) denote the superposition of loads when a destination element has more than one mapped source element.

To linearly transform the lumped nodes of the destination mesh to distributed load perturbations,

$$\Delta\vec{F}_1^{\,D} = \sum\frac{\left\|\vec{p}_2^{\,DR} - \vec{p}_1^{\,DR}\right\|_2}{6}\begin{bmatrix}2I & I\end{bmatrix}\begin{Bmatrix}\Delta\vec{f}_1^{\,D}\\ \Delta\vec{f}_2^{\,D}\end{Bmatrix}, \tag{A.17}$$

$$\Delta\vec{F}_2^{\,D} = \sum\frac{\left\|\vec{p}_2^{\,DR} - \vec{p}_1^{\,DR}\right\|_2}{6}\begin{bmatrix}I & 2I\end{bmatrix}\begin{Bmatrix}\Delta\vec{f}_1^{\,D}\\ \Delta\vec{f}_2^{\,D}\end{Bmatrix}, \tag{A.18}$$

$$\Delta\vec{M}_1^{\,D} = \sum\frac{\left\|\vec{p}_2^{\,DR} - \vec{p}_1^{\,DR}\right\|_2}{6}\left[f_\times\left(\frac{\vec{f}_1^{\,D} + \vec{f}_2^{\,D}}{2}\right) \quad f_\times\left(\frac{\vec{p}_2^{\,D} - \vec{p}_1^{\,D}}{2}\right) \quad 2I \quad -f_\times\left(\frac{\vec{f}_1^{\,D} + \vec{f}_2^{\,D}}{2}\right) \quad f_\times\left(\frac{\vec{p}_2^{\,D} - \vec{p}_1^{\,D}}{2}\right) \quad I\right]_{op}\begin{Bmatrix}\Delta\vec{u}_1^{\,D}\\ \Delta\vec{f}_1^{\,D}\\ \Delta\vec{m}_1^{\,D}\\ \Delta\vec{u}_2^{\,D}\\ \Delta\vec{f}_2^{\,D}\\ \Delta\vec{m}_2^{\,D}\end{Bmatrix}, \text{ and} \tag{A.19}$$

$$\Delta\vec{M}_2^{\,D} = \sum\frac{\left\|\vec{p}_2^{\,DR} - \vec{p}_1^{\,DR}\right\|_2}{6}\left[-f_\times\left(\frac{\vec{f}_1^{\,D} + \vec{f}_2^{\,D}}{2}\right) \quad -f_\times\left(\frac{\vec{p}_2^{\,D} - \vec{p}_1^{\,D}}{2}\right) \quad I \quad f_\times\left(\frac{\vec{f}_1^{\,D} + \vec{f}_2^{\,D}}{2}\right) \quad -f_\times\left(\frac{\vec{p}_2^{\,D} - \vec{p}_1^{\,D}}{2}\right) \quad 2I\right]_{op}\begin{Bmatrix}\Delta\vec{u}_1^{\,D}\\ \Delta\vec{f}_1^{\,D}\\ \Delta\vec{m}_1^{\,D}\\ \Delta\vec{u}_2^{\,D}\\ \Delta\vec{f}_2^{\,D}\\ \Delta\vec{m}_2^{\,D}\end{Bmatrix} \tag{A.20}$$

are solved inversely.

Point_to_Line2 and Line2_to_Point mapping transfer

In the linearized Point_to_Line2 mapping transfer for motion and scalar quantity perturbations, for each destination-mesh Line2-element node, motion and scalar quantity perturbations are transferred from its mapped source Point-element node in a manner identical to the linearized Point_to_Point motion-mapping transfer.

In the linearized Line2_to_Point mapping transfer for motion and scalar quantity perturbations, for each destination-mesh Point-element node, motion and scalar quantity perturbations are interpolated (based on projection) and are transferred from its mapped source Line2 element in a manner identical to the linearized Line2_to_Line2 motion-mapping transfer.

In the linearized Point_to_Line2 mapping transfer for load quantity perturbations, for each source-mesh Point-element node, the point load perturbation is split based on its projected location in the mapped destination Line2 element, and is transferred as two point loads at the destination Line2-element nodes and transformed to distributed load perturbations in a manner identical to the linearized Line2_to_Line2 load-mapping transfer (but without augmentation and lumping of the source mesh).

In the linearized Line2_to_Point mapping transfer for load quantity perturbations, the fields of the new nodes of the augmented source mesh are first populated via interpolation of the fields from the original nodes of the source mesh. That is, equation (A.10) (where $(\cdot)$ is a placeholder) is used to calculate $\Delta\vec{u}^S$, $\Delta\vec{f}^S$, and $\Delta\vec{m}^S$ at the new nodes of the augmented source mesh, where $\overline{l}^S$ was solved via equation (36) from the Line2_to_Point mapping search for load quantities (see Section D.2 of the appendix of [3]). For each Line2 element of the augmented source mesh, distributed load perturbations are lumped as point load perturbations in a manner identical to lumping in the linearized Line2_to_Line2 load mapping. The lumped nodal load perturbations from each Line2-element node of the augmented source mesh are transferred to its mapped destination Point-element node in a manner identical to the linearized Point_to_Point load mapping.

References

[1] Jonkman J and Jonkman B 2016 NWTC information portal (FAST v8) https://nwtc.nrel.gov/FAST8

[2] Jonkman J M 2013 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* [online proceedings] http://arc.aiaa.org/doi/pdf/10.2514/6.2013-202

[3] Sprague M A, Jonkman J M and Jonkman B J 2015 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* [online proceedings] http://arc.aiaa.org/doi/pdf/10.2514/6.2015-1461

[4] Jonkman J M 2009 Dynamics of offshore floating wind turbines—model development and verification *Wind Energy* **12** 459–92 DOI: 10.1002/we.347

[5] Jonkman J, Butterfield S, Musial M and Scott G 2009 *Definition of a 5-MW reference wind turbine for offshore system development* NREL/TP-500-38060 (Golden: National Renewable Energy Laboratory) http://www.nrel.gov/docs/fy09osti/38060.pdf

[6] Bir G 2008 NWTC information portal (MBC) https://nwtc.nrel.gov/MBC

[7] Jonkman J M 2008 Influence of control on the pitch damping of a floating wind turbine *46th AIAA Aerospace Sciences Meeting and Exhibit, 7–10 January 2008, Reno, NV* [online proceedings] http://arc.aiaa.org/doi/pdf/10.2514/6.2008-1307

[8] Hansen M H, Guanaa M and Madsen H A 2004 *A Beddoes-Leishman type dynamic stall model in state-space and indicial formulations* Risø-R-1354 (EN) (Roskilde, Denmark: Risø National Laboratory)