Implementation of a Double-Multiple Streamtube Solver in AeroDyn for Cross-Flow Turbine Analysis

GENERAL CONSIDERATIONS

The double-multiple streamtube (DMST) model will be implemented within the AeroDyn framework and called by the AeroDyn driver. When activated, it will replace the functionality of the blade element momentum and free vortex wake theories currently used to model rotor wake and induction. This implementation of the DMST model is only valid for simple cross-flow turbine geometries (i.e., straight, untwisted blades with no cant, constant radius, and constant preset pitch angle) and does not account for support structure effects, confined flow, active blade pitch, or unsteady aero/hydrodynamics. These simplifications restrict some of the modeling options that can be used with the DMST theory, which are discussed in the Initialize section.

DMST THEORY

The DMST theory presented here is based on the methods of Strickland (1975), Paraschivoiu (1988), McIntosh (2009), and Ayati et al. (2019).

Overview

DMST models divide the swept area of a cross-flow turbine rotor into multiple streamtubes parallel to the inflow direction, as shown in Fig. 1. Each streamtube is further divided along the midline of the rotor, separating the upstream and downstream blade sweeps. Within each streamtube, turbine dynamics are represented by a pair of actuator disks. Separating the upstream and downstream blade sweeps allows DMST models to account for the passage of downstream blades through the wake of upstream blades, improving their performance compared to single streamtube models. The rotor can also be divided in the vertical direction, allowing for vertical variations in the free-stream velocity and rotor geometry.

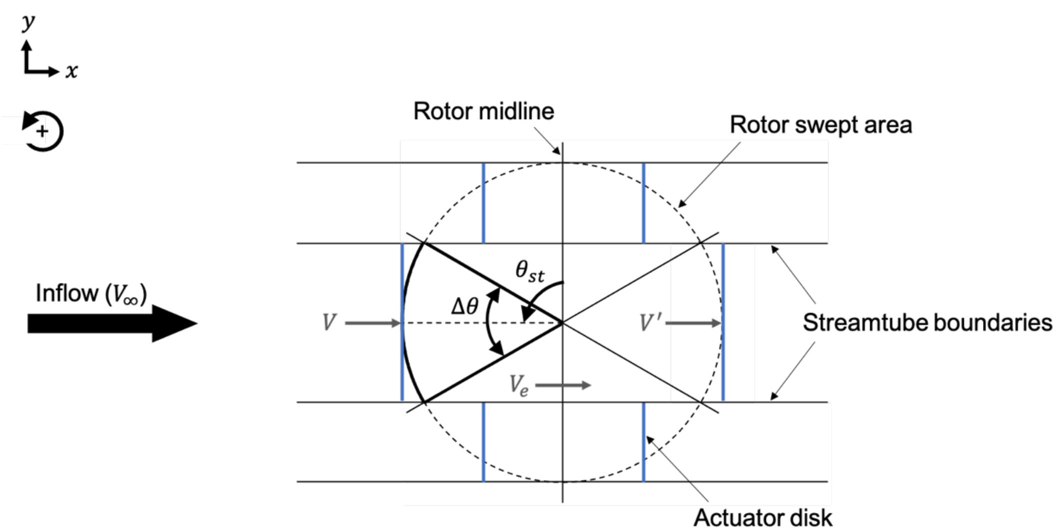

Fig. 1 Swept area of a cross-flow turbine rotor divided into multiple streamtubes. Actuator disks, shown in blue, represent turbine dynamics within each streamtube.

The influence of the rotor on inflow velocities is accounted for using an upstream and downstream induction factor. The induction factor varies with azimuthal position but is assumed constant within a streamtube. Applying both blade element theory and linear momentum theory within each streamtube yields a solution method for the induction factor. Blade element theory can then be used to calculate the thrust, torque, and power coefficients.

Coordinate Systems

The coordinate systems used in this implementation of DMST theory are illustrated in Figs. 2 and 3. Inflow is assumed to be unidirectional, with the free-stream and free-stream minus induced velocities (V_∞, V) positive as shown. All schematics show a top view, with the positive vertical direction pointing out of the page. The tangential velocity (ωR) acts along the line tangent to the blade sweep, and the relative velocity (V_{rel}) is the vector sum of the inflow, induced, and tangential velocities. The angle of attack (α) is defined from the relative velocity vector to the chord line. The preset pitch angle (α_p) is defined from the tangent line to the chord line. All angles are defined as positive in the counterclockwise direction, so the preset pitch angle is negative as shown, with the leading edge rotated outward. If the preset pitch angle is zero, the tangent and chord lines align. The radius (r) is defined from the axis of rotation to the quarter chord ($c/4$) of the blade, which is the reference position for all velocities and forces. The azimuthal blade position (θ) is defined as zero when the blade leading edge, without pitch or twist, is pointing directly upstream. The inflow angle (ϕ) is the angle between the relative velocity vector and the tangent line. The normal and tangential forces (F_n, F_t) act perpendicular and parallel to the tangent line, respectively. The lift and drag forces (F_L, F_D) act perpendicular and parallel to the relative velocity vector, respectively. All forces are positive as shown.

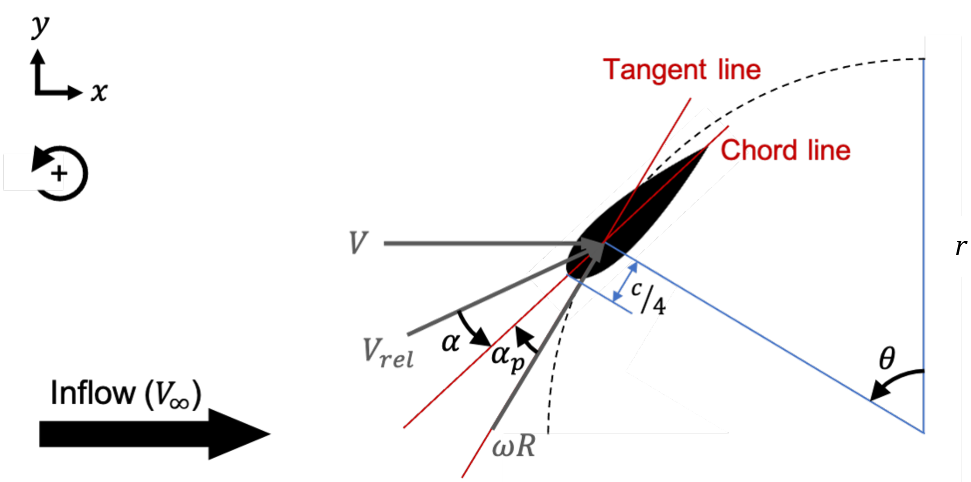

Fig. 2 The coordinate system used for the angles and velocities in this implementation of DMST theory.

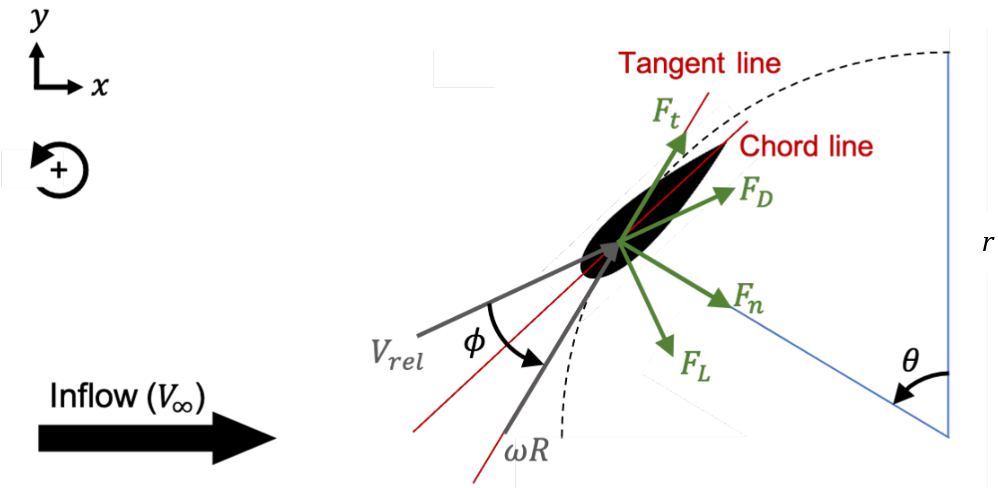

Fig. 3 The coordinate system used for the forces in this implementation of DMST theory.

Assumptions

The DMST model implemented in AeroDyn assumes identical, straight, untwisted blades with no cant and a constant radius and constant preset pitch angle. The base of the turbine is assumed fixed, and only constant rotor rotation is valid. The theory neglects spanwise forces and the pitching moment about the spanwise axis. It does not account for support structure effects, confined flow, active blade pitch, or unsteady aero/hydrodynamics. The model is based on blade element and linear momentum theories, so it also assumes steady, uniform, unidirectional inflow and a non-rotating wake.

INITIALIZE

The following changes will be made to the initialization routines to define additional inputs, validate inputs, and calculate parameters. There are restrictions on the modeling options that can be used with the DMST theory, and these are detailed in the Modeling Options section.

Inputs

- Add a “DMST” option to the WakeMod switch in the AeroDyn primary input file (0=none, 1=BEMT, 2=DBEMT, 3=OLAF, 4=DMST)
- Add an input parameter “Nst” (N_{st}, number of streamtubes) to the AeroDyn primary input file as part of a new “Double Multiple Streamtube Theory Options” section, defined as the number of streamtubes in each vertical section. Note that vertical sections are centered on blade nodes, so all calculations are performed at nodes.
    - $N_{st} > 0$
    - For now, N_{st} will be a single value, meaning the same number of streamtubes will be defined at each node position
    - In the future, this variable could be moved to the blade input file, allowing a different number of streamtubes to be defined at each node position; in this case, N_{st} would be an array of length NumBlNds

- Add an input parameter DMSTRes to the AeroDyn primary input file as part of a new "Double Multiple Streamtube Theory Options" section, defined as the resolution of the input array that serves as the initial guess for the induction factor (see DMST_CalcOutput section for a more detailed explanation of this parameter)
    - Enable a default option, with the default value set to 0.01

Modeling Options

Due to assumptions related to the turbine geometry and theory, certain simulation options are not valid with the DMST model. Detailed descriptions of the AeroDyn driver, primary, and blade input files options that can be used with the DMST model are given here.

- AeroDyn driver input file
    - Input Configuration
        - Echo: Either option (True/False) is valid
        - MHK: Option 0 (not an MHK turbine)
        - AnalysisType: Option 1 (with a single turbine) or option 3 is valid
        - TMax: Set such that the model runs for at least one full rotor rotation
            - $\mathrm{TMax} \geq \frac{2\pi}{RotSpeed}$
        - DT: Set such that each blade falls within each streamtube during a rotation
            - $\mathrm{DT} \leq \frac{\Delta\theta}{RotSpeed}$
        - AeroFile: List name of primary AeroDyn input file
    - Environmental Conditions
        - FldDens: List working fluid density
        - KinVisc: List working fluid kinematic viscosity
        - SpdSound: List speed of sound in working fluid
        - Patm: Unused, MHK effects are not currently considered
        - Pvap: Unused, MHK effects are not currently considered
        - WtrDpth: Unused, MHK effects are not currently considered
    - Inflow Data
        - CompInflow: Option 0 (steady wind) is valid
        - InflowFile: Unused
        - HWindSpeed: List steady horizontal wind speed (if AnalysisType=1, unused if AnalysisType=3)
        - RefHt: List reference height for horizontal wind speed
        - PLExp: List power law exponent for inflow profile, can be set to zero for a flat profile (if AnalysisType=1, unused if AnalysisType=3)
    - Turbine Data
        - NumTurbines: Only 1 turbine is valid
    - Turbine(1) Geometry: DMST theory has only been tested for a simple turbine geometry and orientation, and it is therefore recommended that the following definitions are used
        - BasicHAWTFormat(1): False for a cross-flow turbine
        - BaseOriginInit(1): 0,0,0 is simplest, although any coordinates can be used

    - BaseOrientationInit(1): 0,0,0 is recommended so that turbine base and global frame (and therefore inflow) have the same orientation
    - HasTower(1): False, support structure effects are not currently considered
    - HAWTprojection(1): False for a cross-flow turbine
    - TwrOrigin_t(1): Unused
    - NacOrigin_t(1): 0,0,X is recommended so that nacelle/tower top position is directly above base
    - HubOrigin_n(1): 0,0,X is recommended so that hub is aligned with nacelle
    - HubOrientation_n(1): 0,-90,0 is recommended so that hub x axis is pointing upward and aligned with axis of rotation
  - Turbine(1) Blades
    - NumBlades(1): Must be 1 or more to avoid a null solution
    - BldOrigin_h: Blades should be identical, evenly spaced in the azimuthal direction, and aligned vertically; the origin should be at the top of each blade; the origin of blade 1 should be positioned such that, without pitch or twist, the leading edge is pointing in the negative global X direction when undisplaced; an example for a three-bladed turbine with a 5 m radius is given below
      - BldOrigin_h(1_1): 5,5,0
      - BldOrigin_h(1_2): 5,-2.5,-4.33
      - BldOrigin_h(1_3): 5,-2.5,4.33
    - BldOrientation_h: Blade coordinate system should have the x axis pointing radially outward, the y axis pointing to the trailing edge, and the z axis pointing downward along the span, parallel to the axis of rotation/hub x axis; the orientation of blade 1 should be such that, without pitch or twist, the leading edge is pointing in the negative global X direction when undisplaced; an example for a three-bladed turbine is given below
      - BldOrientation_h(1_1): 0,-90,90
      - BldOrientation_h(1_2): 0,-90,210
      - BldOrientation_h(1_3): 0,-90,330
    - BldHubRad_bl: 0.0 is recommended so that entire blade span is modeled
  - Turbine(1) Motion (used only when AnalysisType=1)
    - BaseMotionType(1): Option 0 (fixed) is valid
    - DegreeOfFreedom(1): Unused
    - Amplitude(1): Unused
    - Frequency(1): Unused
    - BaseMotionFileName: Unused
    - NacMotionType(1): Option 0 (fixed yaw) is valid
    - NacYaw(1): 0 degrees for a cross-flow turbine
    - NacMotionFileName(1): Unused
    - RotMotionType(1): Option 0 (constant rotation) is valid
    - RotSpeed(1): List constant rotor rotational speed about axis of rotation
    - RotMotionFileName(1): Unused

    - BldMotionType(1): Option 0 (fixed) is valid
    - BldPitch: Blade preset pitch should be the same for all blades and is defined as negative about the blade z axis, so that a positive pitch angle rotates the leading edge of the blade inward toward the rotor axis of rotation and a negative pitch angle rotates the leading edge of the blade outward away from the rotor axis of rotation (assuming a blade coordinate system defined as given above); for example, a pitch value of -5 would rotate the blade -5 degrees about the -z axis, resulting in the leading edge being rotated outward away from the rotor axis of rotation
    - BldMotionFileName: Unused
  - Time-Dependent Analysis
    - TimeAnalysisFileName: Unused
  - Combined-Case Analysis (used only when AnalysisType=3)
    - NumCases: List number of cases to run
    - All inputs have been described previously
  - Output Settings
    - OutFmt: List desired text output format
    - OutFileFmt: List desired file output format
    - WrVTK: Any option (0/1/2) is valid
    - VTKHubRad: List hub radius for visualization
    - VTKNacDim: List nacelle dimension for visualization
- AeroDyn primary input file
  - General Options
    - Echo: Either option (True/False) is valid
    - DTAero: Select “default” to use the value provided in the AeroDyn driver input file
    - WakeMod: Option 4 (DMST)
    - AFAeroMod: Option 1 (steady model)
    - TwrPotent: Option 0 (none), since support structure effects are not currently considered
    - TwrShadow: Option 0 (none), since support structure effects are not currently considered
    - TwrAero: False, since support structure effects are not currently considered
    - FrozenWake: Unused, linearization not supported
    - CavitCheck: False, MHK effects are not currently considered
    - CompAA: False, aeroacoustics calculation not supported
    - AA_InputFile: Unused
  - Environmental Conditions
    - Select “default” for all to use the values provided in the AeroDyn driver input file
  - Blade-Element/Momentum Theory Options
    - Unused
  - Dynamic Blade-Element/Momentum Theory Options
    - Unused

- OLAF Options
  - Unused
- DMST Options
  - Nst: List number of streamtubes, must be greater than 0
  - DMSTRes: List resolution of the input array that serves as the initial guess for the induction factors; a value of "default" sets DMSTRes to 0.01
- Beddoes-Leishman Unsteady Airfoil Aerodynamics Options
  - Unused
- Airfoil Information
  - AFTabMod: List interpolation method
  - InCol_Alfa: List angle of attack column
  - InCol_Cl: List lift coefficient column
  - InCol_Cd: List drag coefficient column
  - InCol_Cm: Unused, DMST theory neglects pitching moment
  - InCol_Cpmin: Unused, MHK effects are not currently considered (Cpmin is required only for a cavitation check)
  - NumAFfiles: List number of airfoil files
  - AFNames: List airfoil file names
- Rotor/Blade Properties
  - UseBlCm: False, DMST theory neglects pitching moment
  - ADBlFile: List name of file containing aerodynamic properties, should be the same file for all blades
- Tower Influence and Aerodynamics
  - NumTwrNds: 0, since support structure effects are not currently considered
- Outputs
  - SumPrint: Either option (True/False) is valid
  - NBlOuts: List the number of blade node outputs
  - BlOutNd: List the blade nodes whose values will be output
  - NTwOuts: 0, since support structure effects are not currently considered
  - TwOutNd: None, since support structure effects are not currently considered
  - OutList: List the desired output parameters; user-selectable outputs are detailed in the OutListParameters.xlsx spreadsheet
- Outputs for all blade stations
  - BldNd_BladesOut: List the number of blades to output all node information
  - BldNd_BlOutNd: "All", only option currently available (unused input)
  - OutListAD: List the desired output parameters; user-selectable outputs are detailed in the OutListParameters.xlsx spreadsheet

- AeroDyn blade input file
  - Blade Properties
    - NumBlNds: List the number of blade nodes used in the analysis, must be the same for all blades
    - BlSpn: List the distance in the positive z direction (of the blade coordinate system) from the blade origin to each node, with the first node corresponding to the origin (i.e., BlSpn is 0.0) and the last node

corresponding to the end of the blade; nodes do not need to be evenly spaced
- BlCrvAC: 0 for all nodes, since straight blades are assumed
- BlSwpAC: 0 for all nodes, since straight blades are assumed
- BlCrvAng: 0 for all nodes, since straight blades are assumed
- BlTwist: 0 for all nodes, since straight blades are assumed
- BlChord: List blade chord at each node
- BlAFID: List blade airfoil identifier at each node

Validation

The following checks will be added to the validation subroutine for the primary and blade input files (ValidateInputData, AeroDyn.f90).

- Add a WakeMod_DMST option to the line that checks for a valid WakeMod value
- Throw an error if WakeMod = 4 and AFAeroMod /= 1
- Throw an error if WakeMod = 4 and TwrPotent /= 0
- Throw an error if WakeMod = 4 and TwrShadow /= 0
- Throw an error if WakeMod = 4 and TwrAero = TRUE
- Throw an error if WakeMod = 4 and CavitCheck = TRUE
- Throw an error if WakeMod = 4 and CompAA = TRUE
- Throw an error if WakeMod = 4 and Nst <= 0
- Throw an error if WakeMod = 4 and DMSTRes <= 0 or DMSTRes > 0.5
- Throw an error if WakeMod = 4 and UseBlCm = TRUE
- Throw an error if WakeMod = 4 and NumTwrNds > 0
- Throw an error if WakeMod = 4 and NTwOuts > 0
- Throw an error if WakeMod = 4 and BlCrvAC /= 0 for any nodes
- Throw an error if WakeMod = 4 and BlSwpAC /= 0 for any nodes
- Throw an error if WakeMod = 4 and BlCrvAng /= 0 for any nodes
- Throw an error if WakeMod = 4 and BlTwist /= 0 for any nodes

Parameters

The DMST model uses some variables that do not change with time (i.e., parameters). These parameters will be calculated during initialization and passed to relevant subroutines rather than recalculated at each time step. The parameters N_{bl}, N_{nds}, ν, and ρ are single values, whereas r and c are defined at each node position. Although r is currently a constant that does not vary with blade span, it is calculated at each node position to allow variable r values in future implementations. Similarly, $\Delta\theta$ depends on N_{st} and is currently a constant that does not vary with blade span. However, $\Delta\theta$ is calculated at each node position to allow N_{st} to vary in future implementations. The parameter θ_{st} is calculated for each streamtube at every node position. The subroutine to calculate DMST parameters (DMST_SetParameters) will be called by DMST_Init. Following the OpenFAST framework, DMST_Init will be called by Init_DMSTmodule, which is called by AD_Init. The DMST_Init subroutine will be added to DMST.f90 (new file for submodule source code).

N_{bl}: number of blades

- Assigned to p%rotors(NumRotors)%DMST%NumBlades
- Set from p%rotors(NumRotors)%NumBlades

N_{nds}: number of blade nodes

- Assigned to p%rotors(NumRotors)%DMST%NumBladeNodes
- Set from p%rotors(NumRotors)%NumBlNds

ν: kinematic viscosity of working fluid

- Assigned to p%rotors(NumRotors)%DMST%KinVisc
- Set from p%rotors(NumRotors)%KinVisc

ρ: density of working fluid

- Assigned to p%rotors(NumRotors)%DMST%AirDens
- Set from p%rotors(NumRotors)%AirDens

r: rotor radius

- Assigned to p%rotors(NumRotors)%DMST%radius(NumBlNds)
- Calculated as

$$\overrightarrow{X_{r,\iota}} = \overrightarrow{X_{b,\iota}} - \overrightarrow{X_h}$$

$$r_i = \sqrt{X_{r,i}^2 + Y_{r,i}^2}$$

$\overrightarrow{X_{r,\iota}}$ is the vector between the hub center and blade 1 node i initial positions in global coordinates

$\overrightarrow{X_{b,\iota}}$ is the initial position of blade 1 node i in global coordinates, given as u%rotors(NumRotors)%BladeMotion(1)%Position(:,NumBlNds)

$\overrightarrow{X_h}$ is the initial position of the hub node in global coordinates, given as u%rotors(NumRotors)%HubMotion%Position(:,1)

r_i is the rotor radius at blade node i

c: blade chord length

- Assigned to p%rotors(NumRotors)%DMST%chord(NumBlNds)
- Set from InputFileData%rotors(NumRotors)%BladeProps(1)%BlChord(NumBlNds)

$\Delta\theta$: total streamtube angle

- Assigned to p%rotors(NumRotors)%DMST%dTheta(NumBlNds)
- Calculated as

$$\Delta\theta_i = \frac{\pi}{N_{st}}$$

N_{st} is the number of streamtubes at each node position (constant for now)
$\Delta\theta_i$ is the total streamtube angle at blade node i

θ_{st}: azimuthal position of streamtube midpoint for upstream streamtubes

- Assigned to p%rotors(NumRotors)%DMST%theta_st(NumBlNds,1:Nst)
- Calculated as

$$\theta_{st,in} = \frac{\Delta\theta_i}{2} + \Delta\theta_i * (n-1)$$

n is $1{:}\,N_{st}$
$\theta_{st,in}$ is the azimuthal position of the upstream streamtube n at blade node i, defined in the range 0:pi

θ'_{st}: azimuthal position of streamtube midpoint for downstream streamtubes

- Assigned to p%rotors(NumRotors)%DMST%theta_st(NumBlNds,Nst+1:2*Nst)
- Calculated as

$$\theta'_{st,in} = \theta_{st,in} + \pi$$

$\theta'_{st,in}$ is the azimuthal position of the downstream streamtube n at blade node i, defined in the range pi:2pi

CALCULATE OUTPUT

The following section details the DMST solver implementation within AeroDyn. The solver will include two new subroutines, modeled after the BEMT and FVW frameworks: SetInputsForDMST and DMST_CalcOutput. These subroutines will be called by AD_CalcOutput. SetInputsForDMST will be added to AeroDyn.f90 and DMST_CalcOutput will be added to DMST.f90. The OLAF output subroutines (SetOutputsFromFVW and Calc_WriteOutput_FVW) will be called to calculate module and user-selectable outputs. Currently, all inputs are steady. However, to allow for future implementations of time-varying inputs, V_∞, ω, and α_p will be calculated at every time step within the SetInputsForDMST subroutine.

SetInputsForDMST

This subroutine calculates the input variables V_∞, ω, and α_p at every time step. For compatibility with the output routines, this subroutine will also call the functions SetDisturbedInflow, DiskAvgValues, and GeomWithoutSweepPitchTwist. The variable thetaBladeNds, output by GeomWithoutSweepPitchTwist, will be assigned to m%PitchAndTwist.

V_∞: free-stream velocity

Currently, only steady inflow is supported, represented by a uniform profile with a power law shear in the vertical direction. V_∞ will be set at each node position based on the value of the free-stream velocity for blade 1.

- Assigned to m%rotors(NumRotors)%DMST_u%Vinf(coordinate,NumBlNds)
- Set from u%rotors(NumRotors)%InflowOnBlade(coordinate,NumBlNds,1)

ω: rotor angular velocity

Currently, only constant rotation is supported. A single value of ω applies to all node positions.

- Assigned to m%rotors(NumRotors)%DMST_u%omega
- Calculated as the angular velocity of the hub about the x-axis in local hub coordinates

$$\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} = [\mathrm{DCM}_h]\begin{bmatrix}\omega_X\\ \omega_Y\\ \omega_Z\end{bmatrix}$$

$$\omega = \omega_x$$

DCM_h is the instantaneous direction cosines matrix of the hub node, given as
u%rotors(NumRotors)%HubMotion%Orientation(:,:,1)

$\overrightarrow{\omega_X}$ is the angular velocity of the hub in global coordinates, given as
u%rotors(NumRotors)%HubMotion%RotationVel(:,1)

α_p: blade pitch angle

The pitch angle is assumed to be fixed, constant along the span, and the same for all blades. α_p is calculated at each node position at every time step so that future implementations can account for twisted blades and active pitch.

- Assigned to m%rotors(NumRotors)%DMST_u%pitch(NumBlNds)
- Calculated from the hub to blade rotation matrix, which includes constant and time-varying pitch and blade twist

$$\overrightarrow{\alpha_{b,i}} = \mathrm{EulerExtract}\left(\left[\mathrm{DCM}_{b,i}\right]\left[\mathrm{DCM}_h^T\right]\right)$$

$$\overrightarrow{\alpha_r} = \mathrm{EulerExtract}\left(\left[\mathrm{DCM}_{\mathrm{ref},r}\right]\left[\mathrm{DCM}_{\mathrm{ref},h}^T\right]\right)$$

$$\alpha_{p,i} = \overrightarrow{\alpha_{b,i}}(1) - \overrightarrow{\alpha_r}(1)$$

$\mathrm{DCM}_{b,i}$ is the instantaneous direction cosines matrix of blade 1 node i, given as
u%rotors(NumRotors)%BladeMotion(1)%Orientation(:,:,NumBlNds)

DCM_h^T is the transpose of the instantaneous direction cosines matrix of the hub node, given as u%rotors(NumRotors)%HubMotion%Orientation(:,:,1)

$\mathrm{DCM}_{\mathrm{ref},r}$ is the reference direction cosines matrix of the blade 1 root, given as
u%rotors(NumRotors)%BladeRootMotion(1)%RefOrientation(:,:,1)

$\mathrm{DCM}_{\mathrm{ref},h}^T$ is the transpose of the reference direction cosines matrix of the hub node, given as u%rotors(NumRotors)%HubMotion%RefOrientation(:,:,1)

DMST_CalcOutput

This subroutine guesses initial values for induction factors, calculates intermediate variables, calls airfoil coefficient lookup subroutines, solves for induction factors, and calculates induced velocity. Values of induced velocity are then mapped to blade nodes and output. This is done by tracking the azimuthal location of each blade and assigning values based on the streamtube that the blade falls in at every time step. Steps 3-15 and 17-19 are completed for each streamtube at every node position, in order from smallest to largest azimuthal angle.

1. Guess an initial range of values for the upstream induction factor (u)
   - An initial range of values is guessed and the value that gives the smallest error is chosen as the solution
   - This method avoids convergence issues associated with methods that start with a single value and iterate until the error is below a certain threshold
   - Refer to McIntosh (2009) for more details on this solution method
   - Steps 2-13 are completed for all values of u
   - Part of step 16 (corresponding to step 2), step 17, and part of step 18 (corresponding to steps 3-13) are completed for all values of u'

   $$u = 0.5 : \mathrm{DMSTRes} : 2$$

2. Calculate the upstream average thrust coefficient from linear momentum theory or Glauert's empirical correction ($\overline{C_{T,MO}}$)
   - Classic theory
     - If $u \geq 0.6$,
       - $\overline{C_{T,MO}} = 4u(1-u)$
     - If $u < 0.6$,
       - $\overline{C_{T,MO}} = 0.889 - \left(\frac{0.0203-(0.857-u)^2}{0.6427}\right)$
   - Ayati
     - If $u \geq 0.3$,
       - $\overline{C_{T,MO}} = \frac{4}{3}(1-u)\frac{2+u}{2-u}$
     - If $u < 0.3$,
       - $\overline{C_{T,MO}} = 0.889 - \left(\frac{0.0203-(0.857-u)^2}{0.6427}\right)$

3. Calculate the free-stream minus induced velocity (V)

   $$V = uV_\infty$$

4. Calculate the upstream local tip-speed ratio (λ)

   $$\lambda = \frac{\omega r}{V}$$

5. Calculate the upstream relative velocity (V_{rel})

$$V_{rel} = V\sqrt{1 + 2\lambda \cos\theta_{st} + \lambda^2}$$

6. Calculate the upstream blade Reynolds number (Re_b)

$$Re_b = \frac{V_{rel} c}{\nu}$$

7. Calculate the upstream angle of attack (α)

$$\alpha = \tan^{-1}\left(\frac{\sin\theta_{st}}{\lambda + \cos\theta_{st}}\right) + \alpha_p$$

8. Look up the upstream coefficients of lift (C_L) and drag (C_D) on the blade from tabulated data
   - Coefficients are functions of Re_b and α
   - Reynolds number of airfoil data (based on free-stream velocity and blade chord) should match Re_b
   - Call existing subroutine AFI_ComputeAirfoilCoefs

9. Calculate the upstream inflow angle (ϕ)

$$\phi = \alpha - \alpha_p$$

10. Calculate the upstream normal force coefficient on the blade (C_n)

$$C_n = C_D \sin\phi + C_L \cos\phi$$

11. Calculate the upstream tangential force coefficient on the blade (C_t)

$$C_t = C_D \cos\phi - C_L \sin\phi$$

12. Calculate the upstream average thrust coefficient from blade element theory ($\overline{C_{T,BE}}$); note that this equation is off by a factor of 2 from Ayati et al. equation 13

$$\overline{C_{T,BE}} = \frac{N_{bl} c V_{rel}^2}{\pi\, r \sin(\theta_{st})\, V_\infty^2}\left(C_t \cos(\theta_{st}) + C_n \sin(\theta_{st})\right)$$

13. Locate crossing points occurring between $\overline{C_{T,MO}}$ and $\overline{C_{T,BE}}$, where $\overline{C_{T,BE}} - \overline{C_{T,MO}}$ changes sign
    - Point i corresponds to the lower value of u
    - Point $i+1$ corresponds to the higher value of u
    - If multiple crossing points are identified, choose $u(i)$ that is closest to the u_{final} value of the previous streamtube

14. Calculate $\overline{C_T}$ using a double interpolation method
    - If $\overline{C_{T,BE}}(i) > \overline{C_{T,MO}}(i)$,
        - $\overline{C_T} = \overline{C_{T,BE}}(i) + \dfrac{\frac{\overline{C_{T,BE}}(i) - \overline{C_{T,MO}}(i)}{\overline{C_{T,MO}}(i+1) - \overline{C_{T,MO}}(i)}}{\frac{1}{\overline{C_{T,BE}}(i+1) - \overline{C_{T,BE}}(i)} - \frac{1}{\overline{C_{T,MO}}(i+1) - \overline{C_{T,MO}}(i)}}$
    - If $\overline{C_{T,MO}}(i) \geq \overline{C_{T,BE}}(i)$,
        - $\overline{C_T} = \overline{C_{T,MO}}(i) + \dfrac{\frac{\overline{C_{T,MO}}(i) - \overline{C_{T,BE}}(i)}{\overline{C_{T,BE}}(i+1) - \overline{C_{T,BE}}(i)}}{\frac{1}{\overline{C_{T,MO}}(i+1) - \overline{C_{T,MO}}(i)} - \frac{1}{\overline{C_{T,BE}}(i+1) - \overline{C_{T,BE}}(i)}}$

15. Calculate u_{final} from $\overline{C_T}$
    - Classic theory
        - If $u(i) \geq 0.6$,
            - $4u_{final}^2 - 4u_{final} + \overline{C_T} = 0$
        - If $u(i) < 0.6$,
            - $1.5559u_{final}^2 - 2.6669u_{final} + 2.0001 - \overline{C_T} = 0$
    - Ayati
        - If $u \geq 0.3$,
            - $u_{final}^2 + \left(1 - \frac{3}{4}\overline{C_T}\right)u_{final} + \frac{3}{2}\overline{C_T} - 2 = 0$
        - If $u < 0.3$,
            - $1.5559u_{final}^2 - 2.6669u_{final} + 2.0001 - \overline{C_T} = 0$

16. Repeat steps 1 and 2 for the downstream sweep
    - The following upstream variables are replaced by their downstream counterparts
        - $u \rightarrow u'$
        - $\overline{C_{T,MO}} \rightarrow \overline{C'_{T,MO}}$
    - The initial range of values for the downstream induction factor (u') is given as

    $u' = 0 : \mathrm{DMSTRes} : 2$

17. Calculate the downstream local free-stream velocity (V_e)

- Classic theory
  - $V_e = (2u - 1)V_\infty$
- Ayati
  - $V_e = \frac{u}{2-u} V_\infty$

18. Repeat steps 3-15 for the downstream sweep
    - The following upstream variables are replaced by their downstream counterparts
      - $V \rightarrow V'$
      - $V_\infty \rightarrow V_e$
      - $\lambda \rightarrow \lambda'$
      - $V_{rel} \rightarrow V'_{rel}$
      - $\theta_{st} \rightarrow \theta'_{st}$
      - $Re_b \rightarrow Re'_b$
      - $\alpha \rightarrow \alpha'$
      - $C_L \rightarrow C'_L$
      - $C_D \rightarrow C'_D$
      - $\phi \rightarrow \phi'$
      - $C_n \rightarrow C'_n$
      - $C_t \rightarrow C'_t$
      - $\overline{C_{T,BE}} \rightarrow \overline{C'_{T,BE}}$
      - $u_{final} \rightarrow u'_{final}$
      - $\overline{C_T} \rightarrow \overline{C'_T}$

19. Calculate the upstream and downstream final induced velocities in global coordinates (V_{ind}, V'_{ind})
    - Classic theory
      - $V_{ind} = \begin{bmatrix} u_{final} - 1 \\ 0 \\ 0 \end{bmatrix} V_\infty$
      - $V'_{ind} = \begin{bmatrix} 2u_{final}u'_{final} - u'_{final} - 2u_{final} + 1 \\ 0 \\ 0 \end{bmatrix} V_\infty$
    - Ayati
      - $V_{ind} = \begin{bmatrix} u_{final} - 1 \\ 0 \\ 0 \end{bmatrix} V_\infty$
      - $V'_{ind} = \begin{bmatrix} \frac{u_{final}u'_{final} - u_{final}}{2-u_{final}} \\ 0 \\ 0 \end{bmatrix} V_\infty$

20. Determine the azimuthal location of each blade (θ_b)

$$\begin{bmatrix} \theta_{b,x} \\ \theta_{b,y} \\ \theta_{b,z} \end{bmatrix} = \text{EulerExtract}\left([\text{DCM}_h]\left[\text{DCM}_{\text{ref},h}^T\right]\right)$$

$$\theta_{b1} = \theta_{b,x}$$

$$\theta_{bj} = \theta_{b1} + \frac{2\pi}{N_{bl}}(j-1)$$

θ_{bj} is the azimuthal location of blade j, relative to the blade 1 position when, without pitch or twist, its leading edge is pointing directly upstream

21. Calculate the range of azimuthal angles that corresponds to each streamtube at every node position
    - The range of angles that falls within each streamtube ($\theta_{st,r}$, $\theta'_{st,r}$) is calculated as
        - $\theta_{st} - \frac{\Delta\theta}{2} \leq \theta_{st,r} < \theta_{st} + \frac{\Delta\theta}{2}$
        - $\theta'_{st} - \frac{\Delta\theta}{2} \leq \theta'_{st,r} < \theta'_{st} + \frac{\Delta\theta}{2}$

22. Determine the streamtube that each blade falls within at every node position based on the blade azimuthal location and range of azimuthal angles included in each streamtube

23. Assign induced velocity values to each node on every blade based on the value in the corresponding streamtube

SetOutputsFromLL

This subroutine, currently named SetOutputsFromFVW, takes the induced velocity in global coordinates at each blade node and applies blade element theory to calculate the blade forces and moments, which are assigned to the AeroDyn blade load mesh. Intermediate variables are also calculated and output for use when writing outputs to file. The SetOutputsFromFVW subroutine will be renamed SetOutputsFromLL and generalized such that it can be called after any lifting line solver that outputs induced velocity at blade nodes.

Calc_WriteOutput_LL

This subroutine, currently named Calc_WriteOutput_FVW, takes outputs from SetOutputsFromLL and calculates user-selectable outputs. The Calc_WriteOutput_FVW subroutine will be renamed Calc_WriteOutput_LL and generalized such that it can be called after any lifting line solver.

UPDATE STATES

The update states routine should be modified such that the BEMT modules are not automatically called when WakeMod is not set to WakeMod_FVW (i.e., BEMT modules should not be called when the DMST model is used). Additionally, the induction factor in each streamtube can be stored as a constraint state. In the case of multiple solutions in a given streamtube, the solution in the previous streamtube can be referenced and the solution that is most similar to the previous selected.

END

The end subroutine should be unaffected by these changes.

REFERENCES

A. Ayati, K. Steiros, M. A. Miller, S. Duvvuri, and M. Hultmark. A double-multiple streamtube model for vertical axis wind turbines of arbitrary rotor loading. Wind Energy Science, 4:653–662, 2019. doi:https://doi.org/10.5194/wes-4-653-2019.

S. C. McIntosh, H. Babinsky, and T. Bertényi. Convergence failure and stall hysteresis in actuator-disk momentum models applied to vertical axis wind turbines. Journal of Solar Energy Engineering, 131:034502-1-034502-5, 2009. doi:https://doi.org/10.1115/1.3142826.

I. Paraschivoiu. Double-multiple streamtube model for studying vertical-axis wind turbines. Journal of Propulsion and Power, 4:370–377, 1988. doi:https://doi.org/10.2514/3.23076.

J. H. Strickland. The Darrieus turbine: A performance prediction model using multiple streamtubes. Technical Report, Sandia Laboratories, Albuquerque, New Mexico, 1975.