Load_Analysis Documentation

LoadAnalysis consists of two functions: **RootMoments** and **FatigueAnalysis**

RootMoments is a function that extracts data from a FLORIS simulation and feeds the data into CCBlade. The function then calculates the Normal force and blade root moments on the turbine blades at every 2° azimuth angle using the imbedded BEM (Blade Element Momentum) theory equations and methods in CCBlade .

FatigueAnalysis is a filler for RootMoments that calculates the fatigue loads on the root of the turbine blades. This function calculates the damage equivalents for a specified number of cycles. This function also assumes that for the specified number of cycles, the turbine will experience slightly different wind speeds due to the turbulence intensity. Constant atmospheric conditions are assumed for the entire length of specified cycles due to the static nature of FLORIS.

RootMoments:

Inputs:

- Y_cut_value: The cut value location on the floris layout grid
- Blade_radius: the length of the blades being analyzed
- Yaw_angles: the yaw angles of each turbine in the simulations respectively
- Turb_idx: the index of the turbine in the floris simulation that is being analyzed
- R: radial locations along the blade defining where calculations should be done (values should be increasing)
- Chord: corresponding chord length at each radial location r
- Theta: corresponding blade twist angle at each radial location r (positive twist decreases angle of attack)
- Mu: dynamic viscosity
- Pitch: blade pitch angle
- BN: number of blades analyzed on turbine
- af: NACA airfoil databases (for cl and cd values) at each radial location r
- Json_file: ?

Lines 69-70:
The FLORIS interface is initialized
…
Not sure if the FLORIS interface should be defined in RootMoments or if the FLORIS interface should be defined outside the function and treated as an input into the RootMoments function?

Lines 72-96:
Defining inputs
…
Not sure if self. is the right way to do this?

Line 98:
Initializing the cut plane from FLORIS using the y_cut_value input

Lines 102-130:
The wind speed values are extracted from FLORIS. Values are extracted as an array, the array is reshaped into the resolution of cut plane, all wind speed values are set to 0 except the values inside rotor plane.

**Extracting the wind speed values from FLORIS has only been done for one situation:*

- *Turbine being analyzed is positioned at (0,0) in FLORIS*
- *Wind direction is 0.0°*
- *Second turbine, or upstream turbine, is placed directly upstream of turbine being analyzed*
- *center_x is set to 50 (will be different for different situations)*

Line 135-141:
Calculating turbulence intensity on rotor plane from the FLORIS function. Not yet figured out how to do this yet.

Line 143:
Turbulence intensity is defined to a specific number for the time being

Line 147:
The CCBlade object “rotor” is initialized

Lines 150-210:
Uinf (wind speed) values are defined at each radial location r along the blade at each 2 deg azimuth around the rotor plane. Because the wind speed array extracted from floris provides a 5m x 5m grid resolution grid plane, the wind speed values are interpolated at each radial location r along the blade at each azimuth using a simple 2-d interpolation equation.

$$P \approx \frac{(x_2 - x)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} Q_{11} + \frac{(x - x_1)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} Q_{21} + \frac{(x_2 - x)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} Q_{12} + \frac{(x - x_1)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} Q_{22}$$

****Center_x is needs to be improved (hard coded at the moment)****

Lines 221-231:
A new wind speed vector (Uinf_new) is created. The specified turbulence intensity is multiplied by a value between -.5 and .5 and this value is multiplied by the original Uinf vector. This ensures that at each grid point the turbulence intensity will be slightly different. This value is added to the original Uinf vector to create a new vector of similar but slightly random wind speeds.

****Uinf_new = Uinf + (Uinf * TI_rand)* where *TI_rand = random values between -.5 and .5 * TI*

Lines 232-241:
The new array of wind speeds is used to calculate the normal forces for each azimuth for one revolution at each radial blade section r. The blade root moment is then calculated by multiplying the normal force at each radial blade section r by the length of that section (integration). The sum of these values are multiplied by half of the length of the blade to calculate the blade root moment.

Outputs:
Np: normal force at each blade section at each azimuth
Np_int: the total normal force on the blade at that azimuth (integrated over blade)
Root_Moment: the total root moment on total amount of blades at each azimuth

FatigueAnalysis:

Inputs:

- ❖ Root_Moment: the array of moments at each azimuth calculated in RootMoments
- ❖ sigma_0: Initial S-N curve stress value
- ❖ m: S-N curve slope value
- ❖ cycles: the number of cycles the turbine will experience for the specific situation
- ❖ BN: the number of blades on the turbine

Line 274:
The total root moment on the turbine is divided by the number of blades. This will specify the stress for only one blade

Lines 282-284:
The amplitude of each oscillation (of root moments when rotating about the rotor) is calculated for the specific case.

Line 286:
Based on the sigma_0 and m inputs, the number of cycles until fail, or S_N_cycles, is calculated for the specific condition by comparing each sigma_a (oscillation amplitude) to the S-N curve of the material

Line 288-290:
For this specific situation, the number of user specified cycles is divided by the corresponding S_N_curve cycles to compute the total damage equivalent on the blade.

Output:
D_tot: the total damage equivalent for the specified number of cycles for the specified situation

What was altered in CCBlade:

Lines 755 and 769/770:
The _windComponents function is no longer used for calculating Vx and Vy. Vx and Vy are calculated and imported from FLORIS instead using :
Vx = np.ones(n) * Uinf
Vy = np.ones(n) * Omega*pi/30 * r + vy *** *vy is 0 in this situation*

Uinf and vy are wind speed vectors that are extracted from FLORIS*

Future Additions:

Wind speeds from FLORIS:

- ❖ Be able to extract a wind speed vector from a rotor plane when turbines are placed in any location, any wind direction is specified and any number of turbines are present
- ❖ Find a way to convert the layout location defined in the .json file in FLORIS to an index location in the wind speed array extracted from FLORIS.

Turbulence Intensity:

- ❖ Be able to calculate the turbulence intensity at each grid point on the rotor plane
- ❖ This will allow a random, more realistic turbulence intensity case when calculating damage equivalent loads
- ❖ Will allow the user to view more accurately the fatigue effects of different waking conditions
- ❖ At the moment, one turbulence intensity value is used for every type of case. Using this value, a random TI value is used at each grid point. This does not take into account different waking conditions at the moment.

Lifetime Calculation Function:

- ❖ In addition to calculating damage equivalent loads based on a user specified amount of cycles, be able to calculate the lifetime of the blade based on the specified conditions

Getting useful data for vy input to .dituributedAeroLoads:

More Accurate/Precise Load Calculations:

- ❖ Simple calculations are used at the moment
- ❖ Adding something like CRUNCH to the code to do more in depth load calculations

Cost Function

- ❖ Add a cost function to FLORIS that will optimize a wind farm by maximizing power output and minimizing loads