Blade Element Momentum Theory

Exercise to Lecture #2 of Master Course "Introduction to Wind Turbine Aerodynamics"

David Schlipf

13.05.2024

Building your own BEM code for the NREL 5MW Wind Turbine

The goal of this exercise is to implement your own BEM code to simulate the NREL 5MW Reference Wind Turbine [1] and compared it to the simulations with FAST [2]. The BEM description and code from [3] or notes from the lecture "Certification and Load Assumption" should be helpful.

Please follow the following steps:

1. Please download the zip-archive Exercise02.zip from STUD.IP and unzip it. Then execute the Matlab script Exercise Exercise02.m to get Figure 1.
2. Organize yourself in groups of maximum 5 persons and plan the code development.
3. Hand in the code and present the results in the final lecture.

The current code includes besides the FAST files:

- Exercise06.m: Main script which simulates SLOW [4] with different aerodynamics via Euler forward and compares the results to FAST. For the test simulation, an Actuator Disk (AD) approach has been used for the aerodynamics, implemented in Aerodynamics_AD.m.
- NREL5MWParameterSLOW.m: Loads all parameter for the SLOW model.
- ReadAirfoils.m: Reads in AeroDyn data from FAST for BEM code.
- SLOW.m: Calculates the right side of the ODEs from SLOW.
- PowerAndThrustCoefficientRegion2NREL5MW.mat: c_P and c_T over λ for $\theta = 0\,\mathrm{deg}$ for the Actuator Disk aerodynamics of SLOW.

Please implement the BEM-code in Aerodynamics_BEM.m. You can use a function/file to apply BEM to each blade section. Then change the `Parameter.AeroMode` to `BEM` to use the BEM code instead of the Actuator Disk (AD) code for the aerodynamics. The rest of the code shouldn't be modified (exception RK4, see below). The simulation results should be similar to Figure 1.

The "Matlab onramp" tutorial at www.mathworks.com/learn/tutorials/matlab-onramp.html is helpful to get started. Please provide clean code, use comments etc. A Matlab style guidelines by R. Johnson can be found at www.datatool.com/downloads/MatlabStyle2%20book.pdf.

A basic implementation should include:

- Iteration of the axial and angular induction factor for the blade sections defined in the AeroDyn input file (`RNodes`). Drag can be ignored as in the example in [3].
- Calculation of the aerodynamic thrust and torque for each blade section and for the rotor.
- Aerodynamic damping by considering the tower top speed.

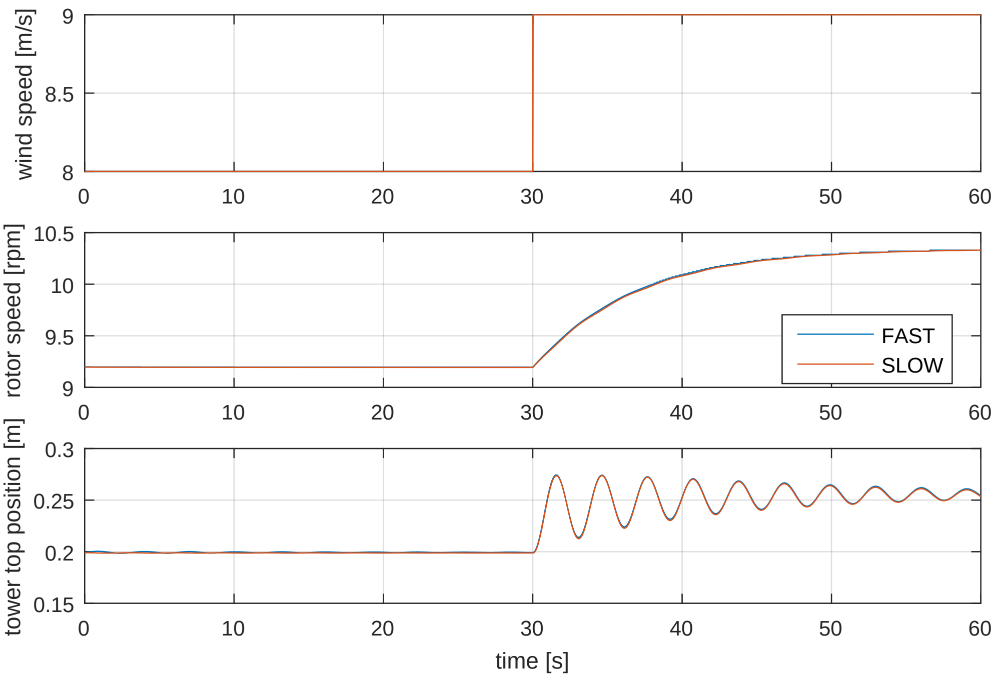

Figure 1: Results comparing FAST and SLOW simulations including both only rotor and tower motion. FAST is using BEM and SLOW an actuator disk approach for the aerodynamics.

A more advanced implementation should additionally include:

- Iteration of the axial and angular induction factor considering drag and tip losses.
- Tilt of the rotor should be considered.

Additional points can be obtained by:

1. Code optimization to be close to or faster than the FAST simulation, e.g. use adequate tolerances and useful initial conditions for the induction factors. Use `profile` to identify slow code.
2. Adjust code such that blade pitch angle changes could be considered (pitch should still be zero).
3. Using RK4 (4th order Runge-Kutta method) for simulation.

Alternatively to Matlab, Python can be used.

References

[1] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. *Definition of a 5-MW Reference Wind Turbine for Offshore System Development*. Tech. rep. TP-500-38060. NREL, 2009. DOI: 10.2172/947422.

[2] J. Jonkman and M. L. Buhl. *FAST User's Guide*. Tech. rep. EL-500-38230. NREL, 2005. URL: https://www.nrel.gov/docs/fy06osti/38230.pdf.

[3] G. Ingram. *Wind Turbine Blade Analysis using the Blade Element Momentum Method*. 2011. URL: www.dur.ac.uk/g.l.ingram.

[4] D. Schlipf. “Lidar-Assisted Control Concepts for Wind Turbines”. PhD thesis. University of Stuttgart, 2015. DOI: 10.18419/opus-8796.