# Development of free-spinning wind turbine representations for CFD

B. Johnson, G. Tabor

*Contact: Ben Johnson [bij201@exeter.ac.uk] [ben.johnson6283@gmail.com]*

## Purpose

Current wind turbine modelling in computational fluid dynamics (CFD) relies on averaged data of the wind speed over a period of days to months. However, the wind turbines will respond to fluctuations in a time-scale of minutes to hours dependent on the moment of inertia of the turbine. Therefore energy loss will occur due to these small time-scale fluctuations that current techniques are unable to predict. Consequently in order to evaluate this energy loss free-spinning CFD techniques must be developed.

## Free-spinning SRF

One approach to wind turbine modelling is the single reference frame (SRF). This is a moving geometry method, whereby the reference frame of the simulation rotates in order to represent the rotation of the turbine. The SRFPimpleFoam application in OpenFOAM was modified to add in the free-spinning calculations. This used an integral of the pressure over the geometry of the turbine to establish a torque, which is then used to establish the rotational speed. Although the implementation was successful in the basic test case the SRF model was found to be too computationally expensive to be viably used for full investigations.

## Actuator Package

Actuator Line and Disk models (ALMs and ADMs) are commonly used to represent wind turbines as they are computationally cheap. As yet, there is no flexible actuator package for OpenFOAM and there has never been a free-spinning actuator package. Therefore an actuator package for OpenFOAM was developed with the key features being:

- Polymorphic model structure allowing different models and control systems to be easily written in.
- Free-spinning ALM
- ADM-R and Goldstein-Optimum disk models
- Turbine control systems
- Any number of blades and turbines, with any combination of turbine models and controls.

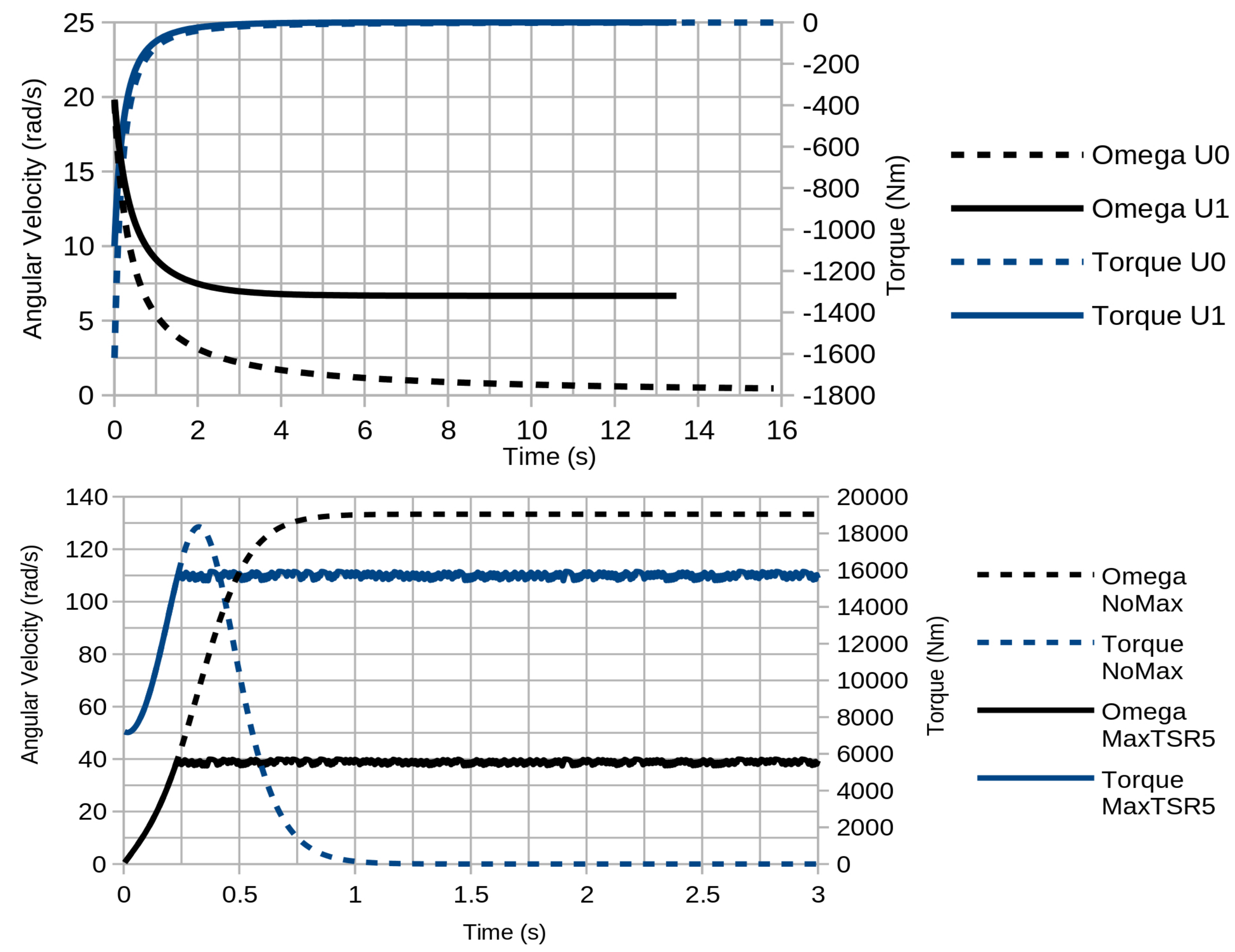
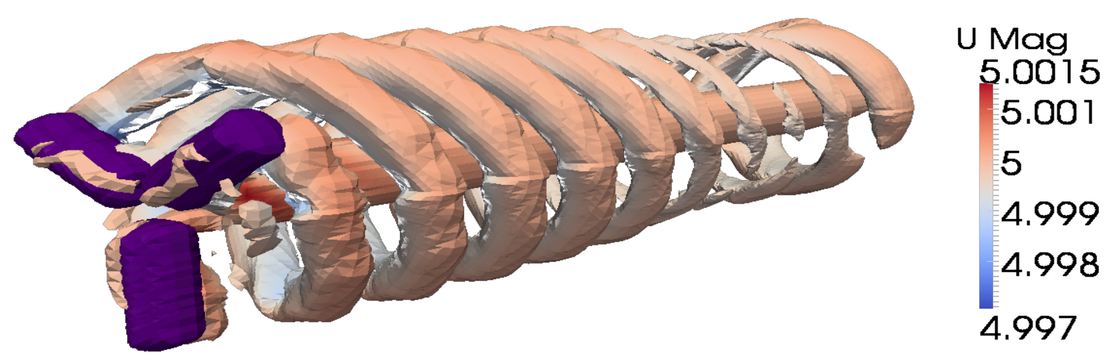



**Figure 3:** Angular velocity and Torque as a function of time for the slow down test case (upper) and the spin up test case (lower)

The actuator code is structured as in Figure 2. The solid lines represent inheritance, the dotted lines show which classes have a member pointer to an instance of each other. At run-time the actuator class is selected as either a disk or a line, the models are then selected. For the actuatorLine class a child of the bladeModel class is selected as the model; for the actuatorDisk class a child of diskModel is selected. The turbine controls are also selected as a child of the turbineControl class. This allows any model or control system to be easily added in and allows for run-time selectable models.

The free-spinning ALM has been tested for spinning up to speed and slowing down, see Figure 3. The package has also been tested for various set-up configurations, see figure 4, and was found to be very efficient and can be used to achieve a long simulated time with a relatively low computational expense. The package will be released summer 2016, along with documentation.

**Figure 4:** Q-criterion coloured by the velocity magnitude for the ALM at a tip-speed ratio of 2



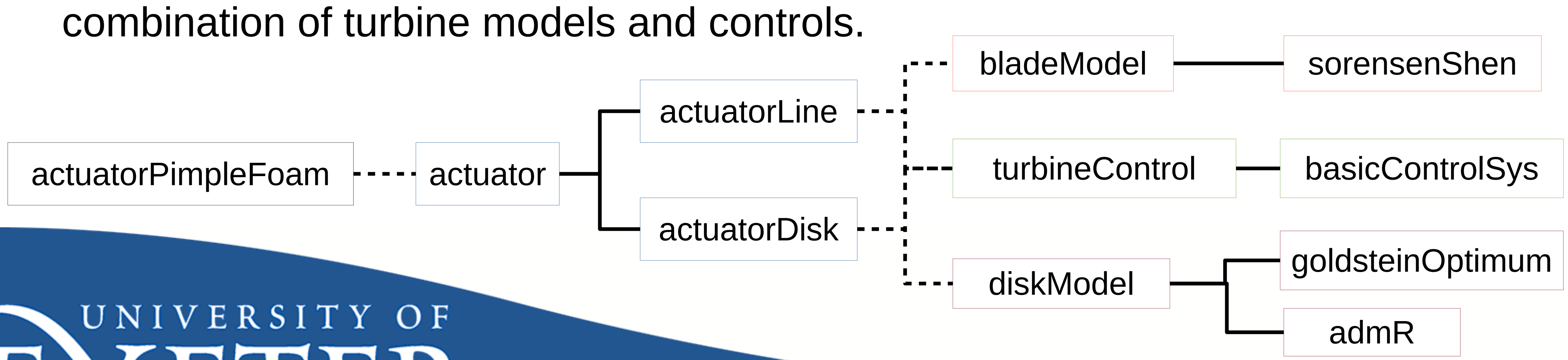




**Figure 2:** Diagram of the code outline, showing inheritance (solid lines) and dependency (dotted). Classes are coloured by inheritance group.

UNIVERSITY OF EXETER
College of Engineering, Mathematics and Physical Sciences