

## **EXPERIENCES WITH OBJECT-ORIENTED AND EQUATION BASED MODELING OF A FLOATING SUPPORT STRUCTURE FOR WIND TURBINES IN MODELICA**

Matthias Brommundt  
Michael Muskulus

Mareike Strach  
Michael Strobel  
Fabian Vorpahl

Department of Civil and Transport Engineering  
Norwegian University of Science and Technology  
Høgskoleringen 7 A  
NO-7491 Trondheim, NORWAY

Fraunhofer Institute for Wind Energy  
and Energy System Technology (IWES)  
Am Seedeich 45  
D-27572 Bremerhaven, GERMANY

### **ABSTRACT**

A floating substructure for wind turbines is modeled using the object-oriented modeling language Modelica in a coupled simulation environment. The equation-based modeling facilitates the implementation for engineers due to declarative model descriptions and acausal formulations. Predefined components from the *Modelica Standard Library* are used to represent several parts of a wind turbine. Especially the *MultiBody* library combined with the graphical editing feature is a powerful method to model the rigid body motions of a floating structure as shown herein. This paper illustrates how the resulting nonlinear differential-algebraic equation system can be implemented and solved in a convenient way. Different solvers can be easily tested to detect the solver with the best performance, without changing the code of the model. The developed model of the floating substructure has been verified with results of the Offshore Code Comparison Collaboration (OC3)-project and the results show good agreement.

### **1 INTRODUCTION**

Modelica is an object-oriented, equation based language to model multi-physical domains in a convenient way. The Modelica Standard has been developed by the non-profit Modelica Association since 1996 (Modelica Association 2012). The freely-available *Modelica Standard Library* (Modelica 2012) is applicable to model the dynamic behavior of technical systems from different engineering domains, such as mechanical, electrical, thermal and control (Modelica Association 2012). For example, models for generators, drive trains or gear-boxes are available and can be edited graphically to develop wind turbine models in a convenient way. The object-oriented approach of Modelica facilitates the handling of large systems due to inheritance, reuse of classes and modification of generic classes. Furthermore, acausal modeling through equations in the Modelica language simplifies the implementation of complex models, such as is the case for an offshore wind turbine. Rigid body motions and coordinate transformations can be described by components of the *MultiBody* library easily (Otter, Elmqvist, and Cellier 1996). Mechanical components and systems, such as mass-spring-damper systems, are generated without having detailed knowledge in multi-body dynamics or in object-oriented modeling.

Shallow water locations in the North Sea are utilized for the installation of bottom-fixed offshore wind turbines on monopile and gravity-based substructures. For intermediate water depth (20-50 m) the tripod and the jacket structure are among the most promising concepts. In deeper water locations (more than 50 m) the material cost for bottom-fixed structures becomes a limiting factor and the natural frequencies of the substructure approach the range of wave excitation frequencies (Hartnett 2000). In many coastal regions deep water offshore solutions for wind turbine substructures are required, as for the coast of the United States, Norway or Japan. The wind blows stronger and more steady far offshore, hence the power

production increases considerably. Furthermore, noise and visual impacts are less of an issue with some distance from the coastline. At deeper water locations floating substructures for wind turbines become a competitive alternative to their bottom-fixed counterparts. However, to become profitable, a further cost reduction of the floating platform and the mooring system is essential. Aside from standard structural design issues, the additional dynamics of floating substructures due to wind and wave induced motions have to be considered. For floating substructures one challenge is to ensure stability of the platform. The large mass of the rotor-nacelle assembly and the rotor thrust force at the high tower top elevation have to be balanced. There are three main approaches to achieve adequate restoring for floating wind turbines (Jonkman 2007):

1. Stabilization due to gravity: The center of mass lies below the center of buoyancy. A typical structure is the spar buoy type, which is used in the Hywind concept from Statoil (2012). The Hywind prototype supports a 2.3 MW wind turbine in a water depth of 210 m.
2. Mooring stabilized units: An excess buoyancy and stiff vertical tethers provide restoring of the floating structure. These tension-leg platforms require expensive taut mooring systems, which can interact with the structural responses of the wind turbine.
3. Water-plane area or column stabilized platforms. A well-known type is the semi-submersible technology from the offshore oil and gas industry. Prototypes for semi-submersible stabilized wind turbines are the WindFloat concept (Roddier et al. 2010) and the floating platform planned in the ongoing EU-project HiPRWind (Quesnel, Bard, and Hanssen 2011).

Offshore wind turbines are exposed to aerodynamic and hydrodynamic loads which can excite the platform in its rigid body degrees of freedom (DOF) and the structural members of the wind turbine close to the eigenfrequencies. The analysis of offshore wind turbines relies on the use of aero-hydro-servo-elastic codes, which combine aerodynamic (aero), hydrodynamic (hydro), control system (servo) and structural-dynamic (elastic) models in a coupled time domain simulation (Jonkman 2007). The state of the art of such simulations is summarized in Vorpahl et al. (2012). Design standards of offshore wind turbines require integrated load calculations with such tools for the certification of wind turbines (IEC 61400-3 2009).

At the Fraunhofer Institute for Wind Energy and Energy System Technology IWES the *OneWind* Modelica library is being developed (Strobel et al. 2011). This library contains models of offshore wind turbine components as well as models for aerodynamic and hydrodynamic load calculations. The in-house tool OneWind from IWES demonstrates how coupled analyses of an offshore wind turbine are performed based on Modelica component models. In Strach et al. (2012) the extension of the *OneWind* library with a spar buoy type substructure was introduced. The floating substructure is described by Jonkman (2010) and the models are verified in the Offshore-Code-Comparison-Collaboration (OC3) project within IEA Wind Task 23 (Jonkman and Musial 2010). The spar buoy type floating substructure from the *OneWind* library is referred to as floater in the following.

The present paper describes the implementation of the floater for the *OneWind* library in more detail, and highlights the advantages of using the *MultiBody* library (Otter, Elmqvist, and Cellier 1996) and the Modelica language in the implementation process. The coupling between the structural model of the floater and the hydrodynamic load calculation module, which was introduced by Quesnel et al. (2010), is described in detail. The floater model is implemented with the in-house tool OneWind, which serves as an editor, and is solved with the commercial simulation environment Dymola (Dymola 2007). In Section 2 it is illustrated how the dynamics of a floating support structure can be simplified as a nonlinear mass-spring-damper system with six DOFs. Section 3 describes the realization of the floater model in the Modelica language in more detail. In Section 4 the floater is verified with results from the OC3-project. Furthermore, the performance of different solvers is tested on the developed model. Section 5 discusses the experiences of the implementation of the floater model. Concluding remarks are given in Section 6.

## 2 DYNAMIC BEHAVIOR OF A SPAR BUOY TYPE FLOATING WIND TURBINE

This chapter introduces the dynamic behavior of a floating wind turbine under excitation from environmental loads. The floater is coupled with the Modelica model of the NREL offshore 5-MW baseline wind turbine (Jonkman et al. 2009), which is part of the *OneWind* library from IWES (Strobel et al. 2011). Subsequently, the equation of motion is introduced.

### 2.1 Structural Properties of the Floater

The upper part of the platform emerges 10 m above sea water level (SWL). Below SWL the OC3-Hywind spar buoy features a linearly tapered conical region to reduce the hydrodynamic loading close to the free surface. Figure 1 visualizes the geometric assembly of the floater.

Table 1 summarizes the main structural properties of the OC3-Hywind spar buoy for the equilibrium position (Jonkman 2010). The center of buoyancy  $c_b$  results from the displaced volume of the floater, which changes permanently due to displacement of the floater or changes in the water surface elevation  $\eta$ .

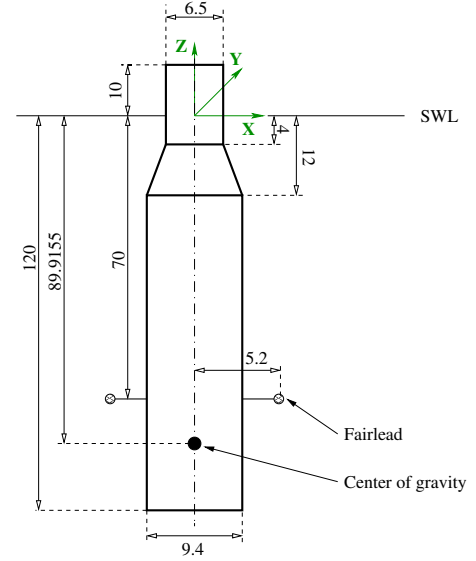


Figure 1: Dimensioning of OC3-Hywind spar buoy (in meter).

Table 1: Properties of OC3-Hywind spar buoy (Jonkman 2010).

|  |                         |
|--|-------------------------|
| Mass of support structure                                | 7,466,330 kg            |
| Location of center of gravity (below water level) $c_m$  | -89.92 m                |
| Location of center of buoyancy (below water level) $c_b$ | -62.19 m                |
| Displaced volume (rest position)                         | 8,029.21 m <sup>3</sup> |

### 2.2 Equation of Motion

The rigid body DOFs of a floating body are labeled surge, sway and heave ( $q_1$ ,  $q_2$  and  $q_3$ ) for translational displacement and roll, pitch and yaw ( $q_4$ ,  $q_5$  and  $q_6$ ) for rotational displacement. Figure 2 visualizes the convention of the DOFs for a floating wind turbine relative to the wind and wave directions. It is assumed that wind and waves are aligned. The equation of motion of the rigid floating wind turbine (Equation 1) is simplified as a nonlinear mass-spring-damper system. The nonlinearity results from the mooring system and viscous flow effects. The floater is moored with chains, which exhibit a nonlinear load-displacement characteristic. Since a floating body has six rigid DOFs the dimension of the matrices are 6 by 6. The matrix  $M$  in Equation 1 accounts for the structural mass of the floater (including the weight of the wind turbine),  $M_h$  is the hydrodynamic added mass matrix,  $B_{linear}$  stands for the linear damping matrix and  $B_{viscous}$  for the nonlinear viscous damping,  $C_{hydro}$  is the hydrostatic restoring matrix and  $C_{mooring}$  the nonlinear mooring system matrix. The right side of Equation 1 describes the excitation forces and moments due to wind, wave and current loads through the (6 by 1) vector  $F(t)$ .

$$(M + M_h) \cdot \ddot{q}(t) + B_{linear} \cdot \dot{q}(t) + B_{viscous} \cdot \dot{q}(t) + C_{hydro} \cdot q(t) + C_{mooring} \cdot q(t) = F(t). \quad (1)$$

The relative acceleration between floater and waves results in an additional force on the submerged part of the structure. This force can be described as a product of hydrodynamic added mass and the relative acceleration between water particles and floater. A description of the linear and nonlinear damping terms, the hydrostatic stiffness and equations for the nonlinear catenary mooring system are given in Jonkman

(2007). More detailed descriptions of the hydrodynamic load calculation and the behavior of floating bodies can be found in Faltinsen (1990).

### 3 IMPLEMENTATION OF THE SUPPORT STRUCTURE IN MODELICA

The floating substructure is implemented under consideration of the characteristics of the Modelica language. In the following, classes are highlighted with cursive and start with a capital letter. The corresponding instance of a class has the same name, but begins with a lower-case letter.

#### 3.1 Model Composition

The class diagram of the floating wind turbine is presented in Figure 3. The floater is modeled using the principles of object-oriented programming. That is, the container class *OC3-Hywind* describes the floating wind turbine through inheritance and containment relationships. In this way, the various components of the wind turbine represented by classes such as *Rotor* and *Nacelle*, and the external load generation contained in the classes in *Wind* and *WaveGenerator*, are accessible in the model.

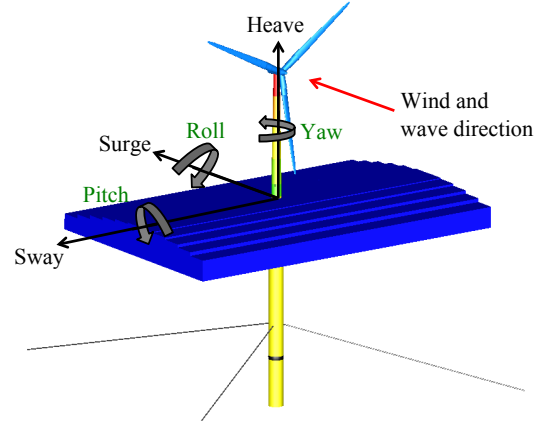


Figure 2: Coordinate system, rigid body degrees of freedom and mean wind and wave load direction of the OC3-Hywind spar buoy.

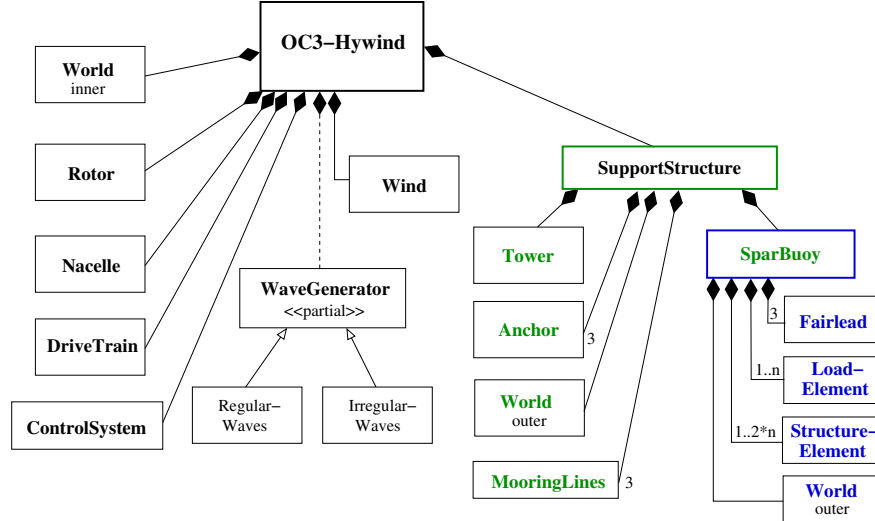


Figure 3: Class diagram of the OC3-Hywind system with wind turbine components, environmental loading classes and components of the floating support structure.

The global coordinate system is defined by the class *World* that is contained by class *OC3-Hywind*. In the following, frame is used as a synonym for coordinate system (Otter, Elmqvist, and Mattsson 2003). The class *SupportStructure* also has a containment relationship to another *World* class, which describes the local coordinate system of the floating support structure. The two instances of the class *World* (global and local frame) are related through an *inner-outer* relationship, which demonstrates a look-up through the instance hierarchy (Fritzson 2004). Thus, the defined common properties of the *inner-World* of the *OC3-Hywind*

class can be assessed by all components of the *OC3-Hywind* class. In this way, it is guaranteed that the same gravitational field is accessible for all referenced coordinate systems, which are declared as *outer* reference, through a look-up in the instance hierarchy. The kinematic calculations and all animations in the *MultiBody* library revert to the coordinate definitions of the *inner-World*. The rigid body motions of the floater are defined by the relative motion between these two coordinate systems. The floating frame of reference approach (Shabana 1998) is utilized to represent the rigid body motions, where the floating frame is attached to an arbitrary point in the body.

The class *SupportStructure* contains an instance of the *Tower* class, three instances of the classes *Anchor* and *MooringLines* and one instance of the class *SparBuoy*. The *SparBuoy* has containment relationships to the classes *StructureElement* and *LoadElement*, which describe the assembly of the floater and the hydrodynamic load calculation. Furthermore, it contains three components of the class *Fairlead*, which describe the location of the mooring fairleads relative to the local frame of the floater. *StructureElements* are described by *MultiBody.BodyShape* elements.

### 3.2 Connectors

Connectors are special classes in Modelica which describe the communication between classes in an equation-based manner. A connector contains the variables of a component that are part of a communication interface. Most predefined components in the Modelica language are connected acausally through equations, i.e., the data flow direction is not specified. Thus, the reusability of connectors increases. Connection interfaces can also be established causally, i.e., with predefined signal flow direction using the prefixes *input* or *output*. In this context, an *input* variable effects the model without being influenced by the rest of the model variables. Hence, it is not necessary for the compiler to determine the causality for such connectors during compilation time (Fritzson 2004). Connectors contain two different types of variables, potential and flow variables. Potential variables are set to zero when two frames coincide, whereas flow variables balance at each connection point according to Kirchhoff's current law.

Classes such as *OC3-Hywind* or *SupportStructure* contain aggregations of several connectors and components with complex interactions. In this case, it makes sense to structure the interface of such components hierarchically through *inside* and *outside* connection interfaces. Internal connectors belong to internal components of a structured component class, whereas *outside* connectors are the external interface of a structured component class to variables that are not a member of the class.

In the floater model three main *outside* communication interfaces are utilized to describe the relationship between different model components. Here, variables are declared as *output*, when the value is set in the connector from calculations inside a specific class, and as *input*, when the class variable is updated outside the class of the connected component.

Figure 4 visualizes the *WaterConnector* on both water and structure side. It establishes the interface between load elements of the structure and the hydrodynamic properties calculated by an instance of the class *WaveGenerator*. Here,  $v$  and  $a$  are velocity and acceleration of the water particles,  $\eta$  is the free-surface elevation and  $p_{dyn}$  is the dynamic pressure at position  $p$ , for which the hydrodynamic properties are calculated. The variables  $v$ ,  $a$  and  $p$  exhibit three components, each in one of the translational DOFs. The position  $p$  is constituted by the displacement of the floater at each time step, which is set at the structure side of the *WaterConnector*. On the water side, the positions are provided through the connector (*input*) and the water particle kinematics ( $v$ ,  $a$ ,  $\eta$  and  $p_{dyn}$ ) are calculated and set in the water side of the connector (*output*).

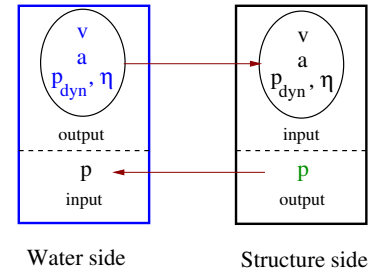


Figure 4: *WaterConnector*: the interface between hydrodynamic loading and structural components.

The *MooringConnector* accounts for the load-displacement characteristics of the mooring system. It contains the following information: displacement of the floater relative to the anchor positions ( $q_{rel}$ ), the vertical and horizontal forces at each fairlead ( $F_{ver}$  and  $F_{hor}$ ) and the fraction of the mooring line resting on the sea floor ( $L_B$ ). On the structure side  $q_{rel}$  is determined through the instantaneous displacement of the floater (see Figure 5). This information is passed through the connector to the instances of the class *MooringLines*, where  $F_{ver}$ ,  $F_{hor}$  and  $L_B$  are calculated.

The *Frame* connector from the Modelica *MultiBody* library is used as *StructureConnector*, which are the central connectors in the Modelica *MultiBody* library (see Figure 6). Here they describe the movement of local coordinate systems with respect to the inertial frame, the assembly of *structureElements*, relations between structural components of the floater and the coupling between load and structural elements. The variables of the *StructureConnector* are:

- Vector position  $r_0$  of the point of origin of the local coordinate system relative to the point of origin of the global coordinate system or the vector position  $r_a$  of a specific frame with respect to a body-fixed reference frame (*frame\_a*). The potential variables  $r$  are resolved in the inertial frame.
- The orientation  $R_0$  of the local relative to the global coordinate system or the orientation  $R_a$  of a specific frame with respect to a body-fixed reference frame (*frame\_a*). Orientations are potential variables and can be described by a direction cosine matrix or by quaternions.
- The flow variables forces  $F$  and moments  $M$  in coordinate direction and around the axes of the *StructureConnector*, respectively. Forces and moments are resolved in the local frame, where they can be associated with geometrical shapes of an appropriate element (Otter, Elmqvist, and Mattsson 2003).

### 3.3 Structural Model

The *structureElements* of the floater are represented by instances of the class *BodyShape* from the *MultiBody* library, which is referred as class *StructureElement* in the following. A *structureElement* has two structural connectors, one at the beginning (*frame\_a*) and one at the end (*frame\_b*) of each element. Individual *structureElements* are connected by frames with each other. Figure 7 visualizes the assembly of the rigid floater model.

The parameter of a *StructureElement* are combined mass  $m$ , location of center of gravity  $r_{cm}$ , inertia  $I$ , the distance between *frame\_a* and *frame\_b*  $r_{ab}$  and the diameter of the structural component. The *structureElements* do not contain information about mass and location of the center of gravity, only the geometrical dimensions  $r$  and diameter are defined. The mass and inertia data is specified in a special instance of the class *StructureElement*, the *centerOfGravity*. The *centerOfGravity* contains no geometrical information and is connected with the structural elements at the specific location of the center of gravity. Thereby, it is possible to discretize different parts of the floater with an arbitrary number of elements. The conical-shaped part can be approximated by a finer discretization of cylindrical elements and the hydrodynamical loads close to the free water surface can be calculated more accurately.

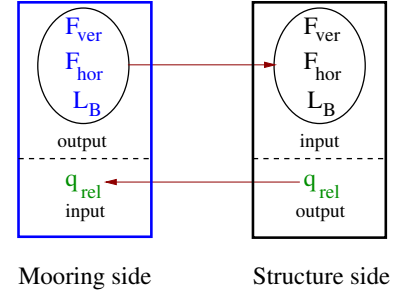


Figure 5: *MooringConnector*: the interface to account for the relative motion between floater fairleads and the fixed anchor positions ( $q_{rel}$ ).

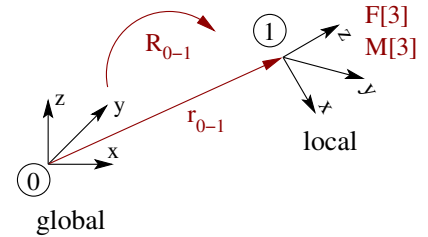


Figure 6: *StructureConnector* relations between global and local coordinate systems.

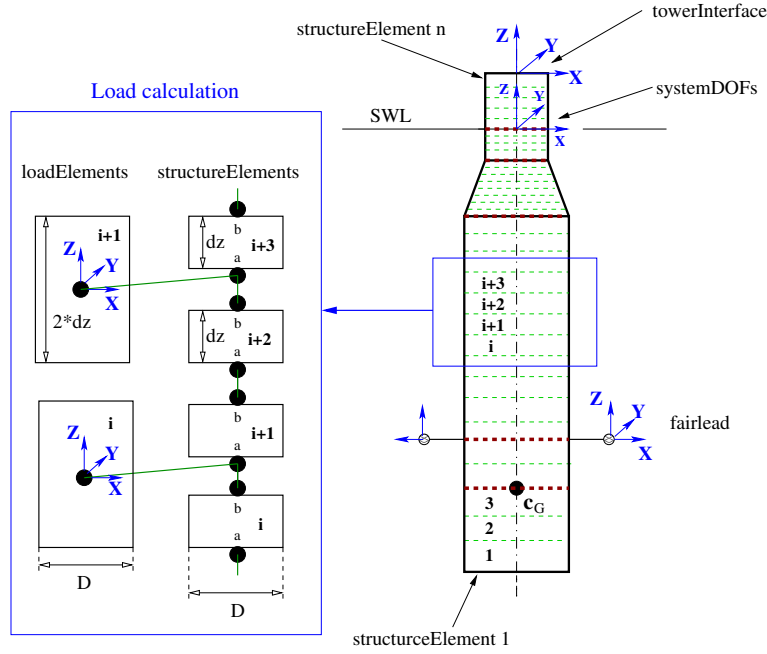


Figure 7: Discretization of the rigid floater model and connection of *loadElements* with *structureElements*. The relevant coordinate systems, such as mooring fairleads (*fairlead*), tower connection (*towerInterface*) and the coordinate system to define the system degrees of freedom (*systemDOFs*), are illustrated.

The interfaces of the floater, which are all described by instances of the *StructureConnector*, are as follows:

- The hydrodynamic load calculation is performed through the coupling of *structureElements* and *loadElements*. Here *frame\_a* of every second *structureElement* ( $2 \cdot n$ ) is coupled with each  $n$ -th *loadElement* (see Figure 7 left).
- The frames *fairlead* are used to describe the relative motion between floater and the fixed anchor positions. The mooring forces are applied here.
- The frame *towerInterface* accounts for the force and moment coupling between tower and floater. It transfers the aerodynamic loads and the mass of the wind turbine components tower, nacelle and rotor to the floater.
- The frame *systemDOFs* defines the local coordinate system of the floater (at SWL) in its equilibrium position. The main task of frame *systemDOFs* is to describe the motion of the floater relative to the global coordinate system. Additionally, the connector is used to apply external forces and moments on the platform, such as contributions from hydrostatic restoring and additional linear damping.

The hydrodynamic loads are determined through instances of the class *LoadElement* at the intersection point of two *structureElements*. Thus, it is not necessary to split the hydrodynamic loads on *frame\_a* and *frame\_b* of the corresponding structural element. The *loadElement* receives the information about the current position of the *structureElement*, where the hydrodynamic loads have to be calculated. Subsequently, the *loadElement* passes this information further to the *waterConnector* and receives the water kinematics at the current position of the floater in return. The water particle kinematics are transferred into the local coordinate system of the floater to account for the instantaneous position and orientation of the floater. Here, the water particle kinematics are used to calculate the hydrodynamic loads on the corresponding *loadElement* through a modified form of Morison's equation (Morison et al. 1950).



The velocity and accelerations of the *structureElements* are determined through derivations of the relative coordinate alterations between global and local coordinate system. The acceleration and velocity of the structure lead to additional force and moment components, which are considered in the equation of motion. The loads are then transferred to the *structureElement*, and included in the dynamic equation of motion together with turbine aerodynamics and mooring forces (Equation 1).

At every time-step the positions of the fairleads are defined through the instantaneous position and orientation of the floater, which is determined through the integration of the equation of motion. For a given horizontal and vertical distance between fairlead and the corresponding anchor, the horizontal and vertical forces are calculated by nonlinear force-displacement characteristics. The force-displacement characteristics are assumed to be quasi-static, which means that the mooring line is in a static equilibrium condition at each time step. Each mooring line is analyzed in the corresponding fixed anchor coordinate system, i.e., the fairlead coordinates are resolved in the anchor frame. Jonkman (2007) presents the equations for the mooring force calculations, which have the main contribution to the nonlinearity of the equation of motion.

## 4 MODEL VERIFICATION AND PERFORMANCE

This chapter presents results from a code-to-code verification of the floater model based on results of the OC3-project. Subsequently the performance of the floater model is investigated for various numerical solvers and floater discretizations. The performance of the model is compared for a linear and nonlinear mooring system.

### 4.1 Model Verification

In Strach et al. (2012) and Brommundt (2011) the floater model was verified using the results from the OC3-project (Jonkman and Muijs 2010). The coupled dynamic response of the NREL offshore 5-MW baseline wind turbine mounted on the OC3-Hywind spar buoy is modeled in a water depth of 320 m. The free decay load case of the OC3-project is investigated further herein for a linearized mooring system (Jonkman 2010). In this load case, the floater is disturbed in all rigid body DOFs separately. As an example, results of the free decay in surge are presented, where the floater is initially displaced about 21 m in surge and released afterwards. More results are summarized in Strach et al. (2012) and Brommundt (2011). Figure 8 presents the results of the free decay test. The results obtained with the model developed herein are labeled with a continuous green line. The response shows good agreement with most of the partner's results from the OC3-project, apart from ACCIONA.

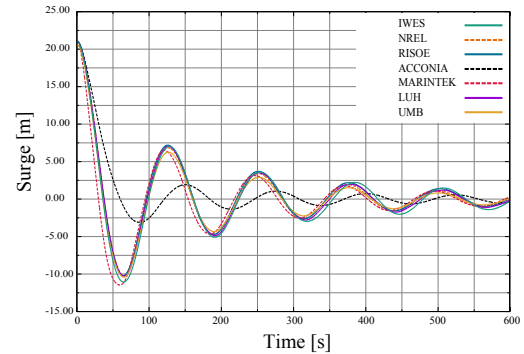


Figure 8: Free decay test for surge degree of freedom in still water and a linearized mooring system. The floater is initially displaced about 21 m in surge.

Table 2: CPU times of different Dymola solvers for a hydrodynamic load case simulation.

| Solver                   | Explicit Euler<br>(coarse) | DASSL<br>(coarse) | Esdirk23a<br>(coarse) | Explicit Euler<br>(fine) |
|--------------------------|----------------------------|-------------------|-----------------------|--------------------------|
| Linear mooring system    | 131 s                      | 999 s             | 575 s                 | 981 s                    |
| Nonlinear mooring system | 4164 s                     | 37031 s           | 29172 s               | 28948 s                  |



## 4.2 Performance of the Model

The model is simulated with OneWind, combined with Dymola as solver, using three different solvers, the explicit Euler method, the implicit Runge-Kutta solver Esdirk23a and the DASSL solver. The tolerances of all solvers are 0.01. A fixed integrator step of 0.125 s is used for the explicit Euler solver. The simulations were accomplished using a Windows 7, 64 Bit system with Intel Core i7 870 with 2.93 Ghz and 8 GB RAM. The solver comparison is performed for a load case, where the floater is excited by regular waves with 6 m wave height and 10 s wave period. Aerodynamic loads are not applied in this load case. Table 2 summarizes the CPU times for a simulation time of 300 s. For all realized calculations the developed model remains unchanged, only solver settings are adapted. Two different discretizations for the floater model are used, coarsely (16 discretization points along the floater) and finely (37 points).

For the coarse discretization and the linearized mooring system the explicit Euler solver shows the best performance, with a CPU time less than half of real time. The CPU time of Esdirk23a is almost two times of real time and that of the DASSL solver around three times. The Euler solver shows a better performance, but this is due to a loss in accuracy. The simulation time depends highly on the discretization of the floater, because the water particle kinematics have to be calculated at each discretization point. Finer discretization results in an increase of factor seven in simulation time of the Euler solver, for both the linear and nonlinear mooring system. With a nonlinear mooring system the simulation time increases for all solvers more than 30 times, compared to the CPU time of the linear mooring system. It is clearly visible that the nonlinearity of the mooring systems slows all applied solvers down.

The number of selected state variables is 12 for the linear and nonlinear model, all selected from the motion of the local frame of the floating body relative to the global frame. They consist of the relative position vector  $r_{rel}$ , the relative velocity  $v_{rel}$  and the relative angular acceleration  $\omega_{rel}$ , i.e., all variables consists of body coordinates. Three of the states have to be selected dynamically during simulation from the four quaternions, which describe the orientation of the floater in a redundant way. The orientation object  $R$  is derived dynamically during simulation from quaternions, which are used as potential states. The angular velocity is also determined from quaternions and its derivatives. When the orientation object and the angular velocity of the local floater frame are determined, all other kinematical variables can be calculated (Otter, Elmqvist, and Mattsson 2003).

## 5 DISCUSSION

A multi-body system, such as the modeled floater, is represented by a higher order differential-algebraic equation system (DAE system), because nonlinear algebraic equations appear in the equation system. These result from constraint equations, e.g., from joint connections in the rotor, generator or nacelle bearing of the wind turbine. DAE models are easy to derive (Section 3), but they may lead to index problems of up to order three with complex nonlinear constraints. The index of a DAE is the minimum number of times that a certain equation in the system needs to be differentiated in order to reduce the equation set to an ODE equation system, which can be numerically solved by normal ODE solvers (Fritzson 2004). Symbolic and numeric transformations are used to transform the equation set into a system with index one or zero. Here the Pantelides (1988) algorithm is applied. The index reduction is performed by an analytical symbolical differentiation or by applying the dummy derivative method.

As described previously body coordinates from the local floater coordinate system are used to describe the states of the floater. This is done, because algebraic loops can be avoided by a proper selection of generalized coordinates. Hence the problem is further simplified. Using generalized coordinates the Pantelides algorithm is able to reduce the equation system down to order one, which can be solved directly by a DAE solver. The index can also be reduced further to index zero by solving the kinematical relations (Otter, Elmqvist, and Mattsson 2003).

In the floater model 682 variables (both for the linear and nonlinear mooring system) were differentiated to transform the DAE systems to a lower index. The resulting DAE systems end up with 12 state variables.

The number and size of both linear and nonlinear systems, which have to be considered by the numerical solver, is decreased through model transformation. This simplification of the formerly very complicated DAE system is done by the Modelica compiler without any input from the user.

The comparison of the different solvers on two problems with different complexity (linear and nonlinear mooring system) indicates that the performance of the simulation strongly depends on the utilized solver. Although all model simplifications are done by the compiler, it is crucial to apply an appropriate solver to the problem. The specification of the developed model in the Modelica language and the implementation of the solver are separated. This makes it possible to change the properties of model and solver independent from each other. The performance for different solvers of the developed model can be verified without changing the code of the model, and hence the most efficient solver can easily be detected.

Models in Modelica are described acausally by DAEs, i.e., the modeling is based on equations instead of assign statements (Fritzson 2004). The causality is analysed automatically by the Modelica compiler when the corresponding equation system is solved. The advantage is that the code is reusable in the object-oriented formulation, since the class description is valid for different causality requirements. The code is easier to change and it is more concise to describe the actual problem by a physical model, and not in an intransparent sequence of algorithms. Object-orientation combined with equation based modeling facilitates the assembly of complex systems in a structured way, close to a physical description.

As described in Strach et al. (2012) the object-oriented description of the spar buoy model will be used to extend the offshore wind turbine support structure engineer design data (EDD), which is a part of the Fraunhofer in-house tool OneWind. Here, different kinds of wind turbine substructures are implemented. The advantage is that the user can set up multi-body systems in a parametric description, without being a specialist in multi-body modeling. Equation-based modeling and the usage of the *MultiBody* and *OneWind* library in conjunction with the multi-physical *Modelica Standard Library* is a powerful method to model the dynamics of such a complex system. As stated by Cellier and Kofman (2006) the *MultiBody* library in conjunction with the Dymola solver can compete with commercial multi-body tools such as ADAMS (MSC Software Corporation 2012), but has furthermore the feature to incorporate additional multi-physics domains.

A dynamical system in Modelica is expressed in a declarative way through equations, and hence it is superfluous to implement explicit time-stepping algorithms. The programmer declares mathematical relationships by formal equations, and does not need to keep track of algorithmic details, because model reduction and numerical solution are accomplished automatically. As an example, in other programming and modeling languages the user would be forced to rewrite the equations and interface with a solver to find the solution of the problem. This is time-consuming and departs the user from the actual physical modeling.

## 6 CONCLUSION

It is demonstrated how a multi-physics domain application of a floating wind turbine can be modeled in Modelica in an effective way without being an expert in multi-body dynamics or object-oriented programming. The advantages of using equation-based modeling become apparent, because time consuming implementation of algorithms can be avoided. Predefined models from the *Modelica Standard Library*, the *MultiBody* and *OneWind* library were utilized in a graphical editor, which further facilitates the build-up of physical models in a convenient way. However, it is crucial to apply the right solver, when it comes to performance issues. The developed model in the Modelica language is independent of the solver implementation, hence different solvers can easily be tested to detect the solver with the best performance for solving the problem. The model of the floater has been verified with results of the OC3-project and the results show a very good agreement.

## REFERENCES

- Brommundt, M. 2011. "Gleichungs- und objektorientierte Modellierung einer schwimmenden Tragstruktur für Windenergieanlagen". Diploma thesis, Institut für Thermodynamik, Technical University of Braunschweig, Braunschweig, Germany.
- Cellier, F. E., and E. Kofman. 2006. *Continuous System Simulation*. New York: Springer.
- Dymola 2007. *Dymola- Dynamic Modeling Laboratory- User Manual*. 5 ed. Lund, Sweden: Dynasim AB.
- Faltinsen, O. M. 1990. *Sea loads on ships and offshore structures*. Cambridge: Cambridge University Press.
- Fritzson, P. A. 2004. *Principles of object-orientated modeling and simulation with Modelica 2.1*. Piscataway, New Jersey: IEEE Press.
- Hartnett, M. 2000. "The application of a spectral response model to fixed offshore structures". *Computers & Structures* 78 (1-3): 355–364.
- IEC 61400-3 2009. *IEC 61400-3 Ed.1 Wind Turbines- Part 3: Design Requirements for Offshore Wind Turbines*. Geneva, Switzerland: International Electrotechnical Commission (IEC).
- Jonkman, J. 2010. "Definition of the Floating System for Phase IV of OC3". Technical Report NREL/TP-500-47535, National Renewable Energy Laboratory, Golden, Colorado.
- Jonkman, J., S. Butterfield, W. Musial, and G. Scott. 2009. "Definition of a 5-MW Reference Wind Turbine for Offshore System Development". Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, Golden, Colorado.
- Jonkman, J., and W. Musial. 2010. "Offshore Code Comparison Collaboration (OC3) for IEA Task 23 Offshore Wind Technology and Deployment". Technical Report NREL/TP-5000-48191, National Renewable Energy Laboratory, Golden, Colorado.
- Jonkman, J. M. 2007. *Dynamics Modeling and Loads Analysis of an Offshore Floating Wind Turbine*. Ph.D. thesis, National Renewable Energy Laboratory, Golden, Colorado. Available via <http://www.nrel.gov/docs/fy08osti/41958.pdf> [accessed January 10, 2012].
- Modelica 2012. *Modelica - A Unified Object-Oriented Language for Physical Systems Modeling, Language Specification*. Version 3.2 Revision 1 ed. Linköping, Sweden: Modelica Association.
- Modelica Association 2012. "Modelica". Accessed April 4, 2012. <http://www.modelica.org/>.
- Morison, J., M. O'Brien, J. Johnson, and S. Schaaf. 1950. "The force exerted by surface waves on piles". *Petroleum Transactions, AIME* 189:149–154.
- MSC Software Corporation 2012. "Adams- Multibody Dynamics Simulation". Accessed April 4, 2012. <http://http://www.mscsoftware.com/Products/CAE-Tools/Adams.aspx>.
- Otter, M., H. Elmqvist, and F. E. Cellier. 1996. "Modeling of Multibody Systems with the Object-Oriented Modeling Language Dymola". *Nonlinear Dynamics* 9:91–112.
- Otter, M., H. Elmqvist, and S. E. Mattsson. 2003. "The New Modelica MultiBody Library". In *Proceedings of the 3rd International Modelica Conference, November 3-4*, edited by P. Fritzson, 311–330. Linköping, Sweden: Modelica Association.
- Pantelides, C. C. 1988. "The consistent initialization of differential-algebraic systems". *SIAM Journal of Scientific and Statistical Computing* 9:213–231.
- Quesnel, L., J. Bard, and J. E. Hanssen. 2011. "Introducing HiPRWind - high power, high reliability offshore wind technology". In *Proceedings of the EWEA Conference 2011*, edited by European Wind Energy Association, 1–7. Brussels.
- Quesnel, L., F. Vorpahl, M. Strobel, and H.-G. Busmann. 2010. "Hydrodynamics meet wind turbines: specification and development of a simulation tool for floating wind turbines with Modelica". In *Proceedings of the Twentieth (2010) International Offshore and Polar Engineering Conference, June 20-25*, edited by J. S. Chung, W. C. Kan, E. Fontaine, and J. wen Chen, 704–710. Beijing, China.
- Roddier, D., C. Cermelli, A. Aubault, and A. Weinstein. 2010. "WindFloat: a floating foundation for offshore wind turbines". *Journal of Renewable and Sustainable Energy* 2:033104–34.
- Shabana, A. A. 1998, March. *Dynamics of Multibody Systems*. 2 ed. Cambridge, UK: Cambridge University Press.

- Statoil 2012. “Hywind”. Accessed April 4, 2012. <http://www.statoil.com/no/TechnologyInnovation/NewEnergy/RenewablePowerProduction/Offshore/Hywind/Pages/HywindPuttingWindPowerToTheTest.aspx>.
- Strach, M., F. Vorpahl, C. Hillmann, M. Strobel, and M. Brommundt. 2012. “Modeling Offshore Wind Turbine Substructures Using Engineer Design Data - a Newly Developed Parametric Approach”. In *Proceedings of the 22nd (2012) International Offshore and Polar Engineering Conference, June 17-22*, edited by J. S. Chung, I. Langen, S. Y. Hong, and S. J. Prinsenber, 196–201. Rhodes, Greece.
- Strobel, M., F. Vorpahl, C. Hillmann, X. Gu, A. Zuga, and U. Wihlfahrt. 2011. “The OnWind Modelica Library for Offshore Wind Turbines - Implementation and first results”. In *Proceedings of the 8th International Modelica Conference, March 20-22*, edited by C. Clau, 603–609. Dresden, Germany: Modelica Association.
- Vorpahl, F., H. Schwarze, T. Fischer, M. Seidel, and J. Jonkman. 2012. “Offshore wind turbine environment, loads, simulation and design”. *WIREs Energy and Environment - Wind Energy WE-11-0091*. (in press).

## AUTHOR BIOGRAPHIES

**MATTHIAS BROMMUNDT** studied mechanical engineering at the Technical University of Braunschweig, Germany. He wrote his diploma thesis at Fraunhofer IWES in the field of floating support structures for wind turbines. Since July 2011 he has been working as research assistant in offshore wind turbines and mooring system optimization at the Norwegian University of Science and Technology. His email address is [matthias.brommundt@ntnu.no](mailto:matthias.brommundt@ntnu.no).

**MICHAEL MUSKULUS** is an Associate Professor and head of the Offshore Wind Turbine Technology group at the Norwegian University of Science and Technology (Department of Civil and Transport Engineering). He represents the Department in the Norwegian Research Centre for Offshore Wind Technology (NOWITECH) and in the FP7 EU project HiPRwind. Originally trained as a physicist at Hamburg University, he has a PhD in applied mathematics on numerical and statistical methods for dynamical systems from Leiden University. His research interests are in the dynamics of offshore wind turbines, high-performance computing, structural optimization, and the statistical analysis of stochastic and dynamical systems. He also serves as Associate Editor for Frontiers in Fractal Physiology. His email address is [michael.muskulus@ntnu.no](mailto:michael.muskulus@ntnu.no).

**MAREIKE STRACH** is a researcher at Fraunhofer IWES. She is responsible for the development of offshore wind turbine substructure models — both fixed and floating — and the calculation of hydrodynamic loads for these models. She is also working on industry projects dealing with offshore wind turbine substructures and load analysis. She studied Naval Architecture and Offshore Engineering at TU Berlin and NTNU Trondheim. Her email address is [mareike.strach@iwes.fraunhofer.de](mailto:mareike.strach@iwes.fraunhofer.de).

**MICHAEL STROBEL** studied Civil and Software Engineering at the Technical University Berlin. He worked as a software and structural engineer in the fields of automotive and aircraft development. Today he is leading the OneWind project and heading the Department for Wind Turbine Simulation and Assessment at the Fraunhofer Institute for Wind Energy and Energy System Technology. His email address is [michael.strobel@iwes.fraunhofer.de](mailto:michael.strobel@iwes.fraunhofer.de).

**FABIAN VORPAHL** is heading the research group loads and concepts in Michael Strobel’s department at IWES. He is responsible for the scientific development of the department and has worked in applied research dealing with offshore wind turbine dynamics and loads for the last six years. He graduated from Karlsruhe University (today Karlsruhe Institute of Technology, KIT) as a mechanical engineer. His email address is [fabian.vorpahl@iwes.fraunhofer.de](mailto:fabian.vorpahl@iwes.fraunhofer.de).