The OneWind® Modelica Library for Floating Offshore Wind Turbine Simulations with Flexible Structures

Mareike Leimeister Philipp Thomas

Fraunhofer Institute for Wind Energy and Energy System Technology IWES Northwest, Germany,
{mareike.leimeister,philipp.thomas}@iwes.fraunhofer.de

Abstract

Floating offshore wind turbines are getting more and more into the focus of interest, as industries aim for larger turbines and deeper water areas. Fully coupled analyses of those highly complex systems are challenging. In this paper, the hierarchical programming structure in Modelica is used to model a fully flexible floating wind turbine system. The single components, as well as special difficulties that have to be dealt with during modeling, are addressed. On basis of a reference floating offshore wind turbine, the implemented fully flexible model is compared with its rigid equivalent, as well as results from code-to-code comparisons of free-decay simulations. The findings are satisfactory and confirm the theoretical assumptions. In addition, further applications of the created model are shown.

Keywords: offshore wind energy, floating platform, fully coupled aero-hydro-servo-elastic simulation, Euler-Bernoulli beam, OneWind Modelica Library, MultiBody

1 Introduction and Outline

Many promising offshore sites for wind energy utilization are in deep water. For water depths larger than 50 *m*, commonly used bottom-fixed foundations, as for example monopiles, jackets, or tripods, are no longer suitable. However, floating platforms, such as spar-buoys, semi-submersibles, or TLPs (tension leg platforms), could be a potential solution for deep water operations. Easier and faster installation due to onshore assembly, as well as reduced noise during erection are some advantages that floating support structures have over bottom-fixed designs. On the other hand, floating wind turbines are very complex systems. Motion-coupling, wave excitation, and additional components like mooring lines are inter alia new challenges for accurately modeling and simulating those systems, and allowing fully coupled load analyses.

Extensive research on floaters is conducted and several prototypes are designed[1,2,3]. Even in the IEA Wind Tasks[4], floating wind turbine systems are included. In order to contribute to code-to-code comparison analyses, a fully flexible model for floating wind turbine systems is developed in the OneWind® Modelica Library.

In this paper, first, the different components of a floating offshore wind turbine system and their implementation in Modelica, based on the Modelica MultiBody Library, are explained in Chapter 2. Chapter 3 outlines the limitations of the implemented floating wind turbine model. The OCx offshore wind turbine designs, elaborated in the IEA Wind Tasks, are used in Chapter 4 as basis for comparison of reference load case simulation results, as well as for demonstrating the high flexibility for adaptions and ease of model modifications. Finally, Chapter 5 summarizes the developed approach and gives recommendations for further work on fully flexible floating offshore wind turbine systems in Modelica.

2 Components and Implementation in Modelica

Object-oriented programming in Modelica enables a hierarchical structure of the complex wind turbine system. The implemented floating wind turbine model contains six main components (rotor, nacelle, operating control, support structure, wind, and waves), which are possibly using further subcomponents, as presented in the following Modelica code and in Figure 1.

```
model OffshoreWindTurbine
extends OneWind.WindTurbine.OffshoreWT
(
  //=== rotor ===
  ,redeclare model Rotor = OneWind.Rotor
  (
    redeclare record RotorData
    ,redeclare model Hub
    ,redeclare model Blade
  )
  //=== nacelle ===
  ,redeclare model Nacelle =
      OneWind.Nacelle
```

[1] https://www.statoil.com/en/news/hywindscotland.html (Accessed: 02 March 2017)

[2] http://principlepowerinc.com/en (Accessed: 02 March 2017)

[3] http://ideol-offshore.com/en (Accessed: 02 March 2017)

[4] http://www.ieawind.org/taskWebSites.html (Accessed: 23 September 2016)

```
  (
    redeclare record NcData
    ,redeclare model Drivetrain
    ,redeclare model Generator
    ,redeclare model YawController
  )
  //=== operating control ===
  ,redeclare model OperatingControl =
     OneWind.OperatingControl
  (
    redeclare record OperatingControlData
    ,redeclare model MainControl
    (
      redeclare model PitchControl
      ,redeclare model
         GeneratorTorqueControl
    )
    ,redeclare model GeneratorSpeedFilter
  )
  //=== support structure ===
  ,redeclare model SupportStructure =
     OneWind.FlexibleFloater
  //=== wind ===
  ,redeclare model Wind = OneWind.Wind
  (
    redeclare record WindData
  )
  //=== waves ===
  ,redeclare model Waves = OneWind.Wave
  (
    redeclare record WaveData
  )
);
end OffshoreWindTurbine;
```

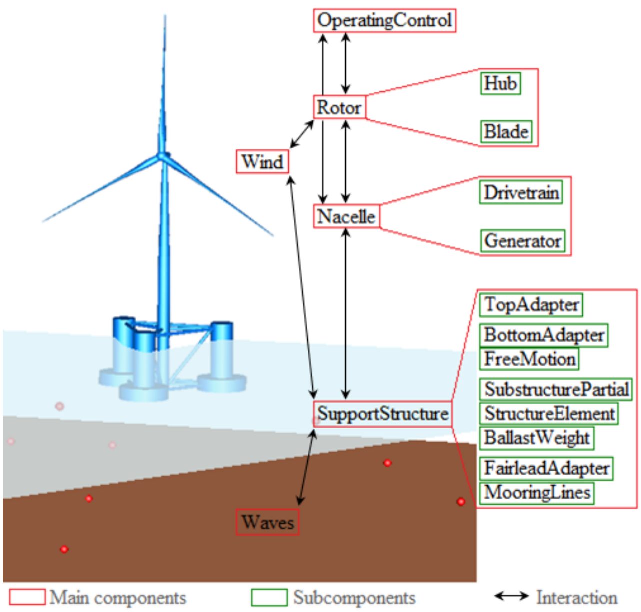


Figure 1. Components and interactions of a floating wind turbine, using the example of a semi-submersible platform.

2.1 Rotor

The rotor model extends the basic model for a hub with one blade to a three-bladed rotor. The blades are implemented either as rigid bodies or as flexible structures, which could be based on modal reduction techniques or finite-elements (Thomas et al., 2014). The structure model is connected to the aerodynamic model, which uses unsteady blade element momentum theory for load calculation, and takes aero-structure-coupling into account.

2.2 Nacelle

The model of the nacelle contains two subcomponents: the drivetrain and the generator. Furthermore, the yaw controller is included. The nacelle is basically represented as rigid link with mass and inertia, while drivetrain and generator provide also stiffness and damping (Strobel et al., 2011).

2.3 Operating Control

The operating control covers algorithms and parameters for pitch and generator torque control, using either built-in PID-algorithms (Jonkman et al., 2009) or an external control DLL. The latter one is obtained from a simulation tool, Bladed (GL Garrad Hassan, 2010) or Hawc2 (Larsen and Hansen, 2014), and accessed via a generic DLL interface. A bus system forms the link between rotor, nacelle, and operating control (Otter, 2009). There is no direct link to the support structure, as the control parameters are initially adjusted based on the floating system design. Furthermore, different operating phases, such as startup, shutdown, or idling, can be realized.

2.4 Support Structure

The support structure model defines everything related to the floating device, including the tower from the RNA (Rotor Nacelle Assembly) down to the substructure, the floater itself, station-keeping system, and all loads acting on the entire support structure. Furthermore, it contains the *FreeMotion*, relevant for modeling the motions of the floating body. An overview of the structure of the model *SupportStructure* is given in the following:

```
model SupportStructure
//--- substructurePartial ---
extends OneWind.SubstructurePartial;
//--- structureElement ---
TopologyData = OneWind.FloaterTopologyData;
StructureElement = OneWind.BernoulliBeam3D;
//--- adapters ---
OneWind.AdapterFemFrameToFrame_free
    topAdapter;
OneWind.AdapterFemFrameToFrame_fixed
    bottomAdapter;
OneWind.AdapterFemFrameToFrame_free
    fairleadAdapter[3];
//--- additional weights ---
OneWind.AdditionalWeightLoadElement
    ballastWeight[noElementsBallast];
```


DOI 10.3384/ecp17132633

```
OneWind.AdditionalWeightLoadElement
    capsWeight[noElementsCaps];
//--- station-keeping system ---
OneWind.DynamicMooringLines mooringLines
    [3];
MultiBody.Parts.FixedTranslation
    fTrFairleads[3];
MultiBody.Parts.FixedTranslation fTrAnchors
    [3];
//--- freeMotion ---
FreeMotion = MultiBody.Joints.FreeMotion;
end SupportStructure;
```

2.4.1 SubstructurePartial

The basis of the support structure model is formed by the partial model *SubstructurePartial*. This covers all main loads, as well as the visualization of the environment, represented by a squared *FixedShape* for the seabed and a moving surface for animating the wave motion, and contains the model *World*. The structure of the partial model *SubstructurePartial* is presented in the following:

```
partial model SubstructurePartial
outer MultiBody.World world;
//--- visualization ---
MultiBody.Visualizers.FixedShape ground;
MultiBody.Visualizers.Advanced.Surface
    surface;
//--- wave loads ---
OneWind.MorisonLoadElement waveLoads[
    noElementsUnderWater];
OneWind.MorisonLoadHeavePlate
    morisonLoadHeavePlate[noHeavePlates] if
     heavePlates;
//--- wind loads ---
OneWind.TowerLoadElement windLoads[
    noElementsOverWater];
//--- buoyancy loads ---
OneWind.BuoyancyLoadElement buoyancyLoads[
    noStructureElements];
end SubstructurePartial;
```

The determination of the loads due to waves, wind, and buoyancy is covered in the following in more detail. The gravity force is not elaborated explicitly, as its computation is directly included in the setup of the subcomponent *StructureElement* (Subsection 2.4.2). A schematic overview of these main loads is presented in Figure 2.

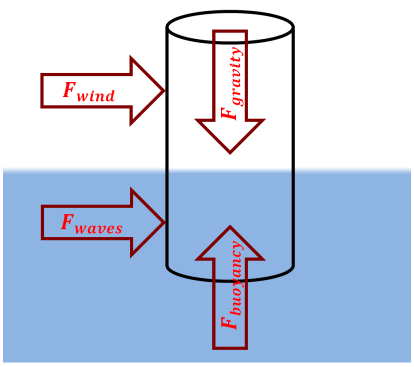


Figure 2. Schematic representation of the main loads acting on the support structure.

Wave Loads The hydrodynamic load calculation uses Morison's equation, as given in Equation 1, and is performed for each structure element that is initially below the water surface, based on its diameter D, length ∂z, hydrodynamic drag coefficient C_D, and added mass coefficient C_a, as well as velocities (structure velocity $\dot{q}$, water particle velocity v_water, relative velocity $v_\mathrm{water} - \dot{q}$), accelerations (structure acceleration $\ddot{q}$, water particle acceleration $\dot{v}_\mathrm{water}$), and water density ρ_water.

$$\begin{aligned} F_\mathrm{waves} =& \frac{1}{2}\rho_\mathrm{water} C_\mathrm{D} D \left(v_\mathrm{water} - \dot{q}\right)\left|v_\mathrm{water} - \dot{q}\right| \partial z \\ &+ \rho_\mathrm{water}\left(1 + C_\mathrm{a}\right)\frac{\pi D^2}{4}\dot{v}_\mathrm{water}\partial z \\ &- \rho_\mathrm{water} C_\mathrm{a}\frac{\pi D^2}{4}\ddot{q}\partial z \end{aligned} \quad (1)$$

As offshore wind turbines often have to deal with large dimensioned support components, a separate parameter is introduced to select whether a fixed value for the added mass coefficient should be used, which is only valid for slender structures, or the added mass coefficient is calculated depending on the wave number, known as MacCamy-Fuchs approach for large diameters (Yu, 2015). Furthermore, if the floater is equipped with heave plates, acting as motion suppression device, as it is the case for semi-submersible platforms, an additional hydrodynamic heave force due to these heave plates is included.

Wind Loads In the aerodynamic load calculation, the drag forces at each emerged support structure element are computed by means of Equation 2, based on the density of air ρ_air, the aerodynamic drag coefficient C_d of the cylindrical element, its diameter D and length ∂z, as well as the local relative velocity, resulting from the local wind speed v_wind and the velocity of the structure element $\dot{q}$.

$$F_\mathrm{wind} = \frac{1}{2}\rho_\mathrm{air} C_\mathrm{d} D \left(v_\mathrm{air} - \dot{q}\right)\left|v_\mathrm{air} - \dot{q}\right| \partial z \quad (2)$$

Buoyancy Loads Because a floating wind turbine system is considered, buoyancy force and center of buoyancy will vary with the motion of the floater. Therefore, these two variables have to be computed for each structure element at each time step, depending on the actual position. The coordinate system, used in this calculation, as well as the degrees of freedom (DoFs) of a floating wind turbine are presented in Figure 3.

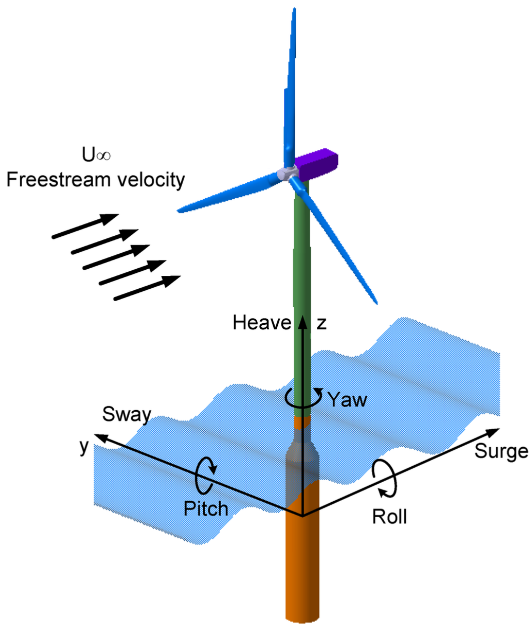


Figure 3. Coordinate system of a floating wind turbine, using the example of a spar-buoy platform, adapted by the author from (Tran et al., 2014).

In an extensive computation, first, the sequential rotation, defined by roll, pitch, and yaw angles, is transformed into a combined rotation, expressed in terms of a combined rotation angle α_{combined} and the corresponding axis of the combined rotation. Instead of having the complex floater geometry rotated, the following approach is used: it is assumed that the floater remains in its initial position and the water plane area is rotated with the combined rotation angle around the axis of combined rotation, however, in opposite direction, as schematically shown in Figure 4.

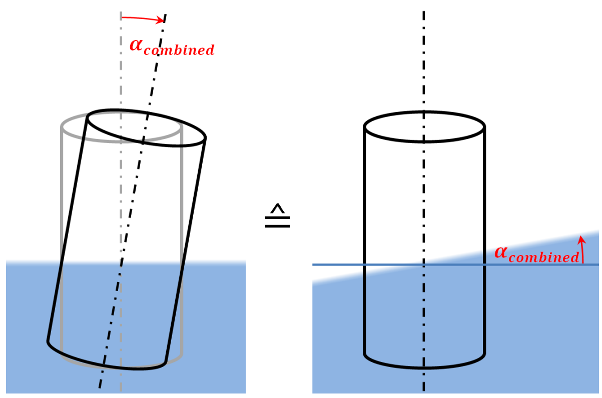


Figure 4. Schematic representation of the used buoyancy calculation approach.

Including the translational motion of the floater, given by surge, sway, and heave values of the platform, and used as the distance from the initial origin to the moved water plane, the equation for the rotated and translated water plane can be set up. With the node positions of the considered beam, defined in the subcomponent *StructureElement*, as further explained in Subsection 2.4.2, a straight equation can be defined for the considered structure element. The intersection of this straight with the moved water plane is analyzed according to the following case discrimination:

- An infinite number of cross points corresponds to the structure element lying exactly in the water plane, leading to a buoyant volume of half of the element volume.
- The solution of having no cross points corresponds to the structure element being parallel to the water plane. Depending on the node positions in relation to the translational motion, the element is either fully submerged or not submerged at all.
- Finally, when having one cross point of straight and plane, the buoyant volume can be computed as fraction of the element volume, if the cross point lies within the actual length of the structure element. If the straight would intersect the water plane at an extension of the structure element, the buoyant volume is either equal to the element volume or zero, depending on the relative position of the element nodes to the translated water plane.

From the determined buoyant volume V_{B}, the buoyancy load of each structure element at each time step is obtained by multiplication with the water density ρ_{water} and gravitational acceleration g, as given in Equation 3.

$$F_{\text{buoyancy}} = V_{\text{B}} \rho_{\text{water}} g \tag{3}$$

The buoyancy force is then connected to the *frame_c* of the element (introduced in Subsection 2.4.2), which is located in the middle of the element axis, including deformation. As, however, the point of attack of the buoyancy force varies with the motion of the floater, the distance from the actual point of attack to the central point (*frame_c*) is computed according to the different element positions elaborated in the above case discrimination. The resulting moment due to the shifted center of buoyancy is finally added as torque load to the *frame_c*, so that correct loads due to buoyancy are represented.

2.4.2 StructureElement

In the subcomponent *StructureElement*, all members of the support structure (floater and tower up to the RNA) are defined, based on a record for the topology data. This record contains number and coordinates of the nodes, as well as number and definition of the members, specified by the two end nodes and the structural properties of the element. The tubular beam properties are defined by an isotropic material (with elastic modulus and shear modulus, density, and Rayleigh damping parameters),

DOI
10.3384/ecp17132633

as well as start and end diameters and wall thicknesses. This record can either be written manually, or generated by means of a MATLAB code. The latter method makes it easy to change subdivisions of the beam elements and is thus useful for a more comprehensive structural analysis. The *TopologyData* record is used for setting up the structure elements, using an extended 3D-Bernoulli beam element model, which is also applicable to branched geometries and has an external load connector *frame_c*.

Avoiding Closed Loops Offshore substructures might be branched structures, like semi-submersibles, TLPs, or also bottom-fixed support structures, such as tripods and jackets. This will lead to closed loops in multibody applications that use the floating frame of reference, what makes it impossible to calculate the unique orientation of each frame, especially where branches are connected. This problem is addressed here by excluding orientation from the node connectors that are used to build up the substructure by defining the position of each member and connecting the members. In case internal forces need to be resolved between local beam and world frame, a local beam orientation is constructed by means of *absoluteRotation()* and *axesRotations()*. This beam orientation only depends on the initial node positions and is independent of the body motion. Therefore it is combined with a reference orientation from the *bottomAdapter* to calculate an approximation of the local beam orientation, assuming small flexible body motion, which is sufficient for rigid body motion. Since the multiplicity of external floating frame connectors rely on correct orientation, each frame orientation is exactly calculated from a combination of reference orientation and local elastic rotation.

Adapters Since the structure is built by 3D-Bernoulli beams, which have FEM-nodes with node position, cut force, cut torque, elastic displacement, and elastic rotation as variables, adapters between the *FemFrame* connectors and the common Modelica *Frame* connectors, not having the elastic deformation variables but the frame orientation in addition, are needed. Two different adapters are used: *AdapterFemToFrame_fixed* and *AdapterFemToFrame_free*. The "fixed" adapter (*bottomAdapter*), where the boundary conditions are set, is needed for the structure node that will be connected to the *FreeMotion*, while the "free" adapter is for connecting any other components, such as mooring lines at the fairleads (*fairleadAdapter*), or RNA on the tower top (*topAdapter*).

2.4.3 Additional Weights

Besides the main loads due to waves, wind, and buoyancy, which are already included in the partial model *SubstructurePartial*, covered in Subsection 2.4.1, additional weight due to column caps, not considered as beam elements, and ballast have to be integrated into the model. The subcomponent *AdditionalWeightLoadElement*, similar to the *BuoyancyLoadElement*, however, just using the simple time- and position-independent weight calculation, has the weight as output, which is implemented in the vertical component of a force equation.

2.4.4 Station-Keeping System

The station-keeping system contains three different components: fairleads, mooring lines, and anchors, as schematically shown in Figure 5.

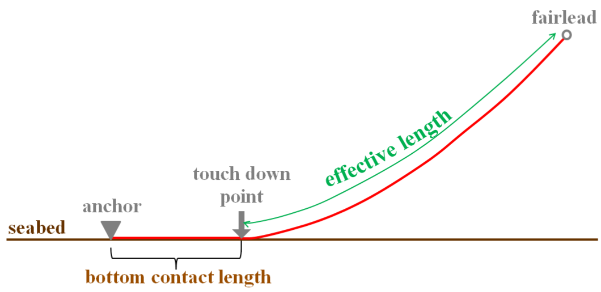


Figure 5. Schematic representation of a catenary mooring line.

Fairlead and anchor positions are defined by fixed translations, while the mooring lines are implemented by means of a separate model. This (1) models the mooring lines, divided into several elements, as mass-spring-damping systems, (2) considers velocity-dependent (Morison) and inner damping, (3) computes weight and buoyancy of the lifted parts of the catenary lines, and (4) includes bottom contact reaction forces. Thus, the shape of the mooring lines, as well as the effective lengths are internally determined at each time step, based on the common catenary equation, given mooring line parameters, and the actual fairlead positions. (Feja, 2013)

2.5 Wind

Several wind models, either deterministic, based on gust models, or stochastic, using binary or ASCII data, are available. Different gust profiles, such as 1-cosine gust, extreme coherent gust, extreme direction change, or extreme operating gust, can be selected; wind shear can be included by means of an exponential or logarithmic profile; and the tower effect can be considered either for upwind or downwind turbines. Two different guidelines can be chosen: IEC-61400-1 edition 3 (International Standard, 2005) or GL guideline for certification of wind turbines (GL Rules and Guidelines, 2010). Corresponding to this, the wind turbine class (I, II, III), depending on the reference wind speed average over 10 minutes, the turbulence characteristic (A, B, C), for high, medium, or low turbulence, as well as the turbulence model (e.g. normal or extreme) have to be specified. For the simulation of the wind, ramped, steady, turbulent, as well as upwind or downwind steady or turbulent wind types for turbine wake simulation using two or more turbines, can be chosen. Finally, the basic parameters, such as hub wind speed, density and dynamic viscosity of air, wind direction, and flow inclination, are defined in the *WindData* record. (Thomas et al., 2014)

2.6 Waves

Two basic wave models are implemented in Modelica: one model for regular waves and one for irregular random waves. Water parameters, like water depth and density, as well as the option to use Wheeler stretching or linear extrapolation method, are common for both wave models. The regular waves are further specified by wave period, wave height, and phase angle. The irregular waves, on the other hand, are defined by a Pierson Moskowitz or JONSWAP wave spectrum, significant wave height, spectral period, random phase angles, and number of frequencies, because irregular waves are obtained as superimposition of several regular waves of different frequencies.

3 Limitations

Holistic modeling of a flexible floating offshore wind turbine system is, because of non-linear physics and fully coupled aero-hydro-servo-elastic simulation, very complex and extensive. Therefore, some simplifications have to be made in the first step of implementation, which are depicted in the following:

- Additional weights due to caps and ballast are computed for each element and connected to their *frame_c*, which is located at the midpoint along the central axis of each element. However, this does not correspond to the correct center of gravity in case of the caps and the uppermost element containing ballast, if this element is only partially filled with ballast. This inaccuracy can be removed by adding a torque load to the *frame_c* resulting from the different center of gravity, similar to the method applied in the buoyancy load calculation, described in Subsection 2.4.1.

- In case of a branched structure, like the semi-submersible floater, there is an overlap of elements. For example, the pontoons are connected to the nodes at the central axis of the columns, however, the pontoon structure itself should just start from the column surface instead of the column center. This leads to some additional incorrect weight, which has to be removed, for example by using massless elements for connecting branched elements to the surface of another element. However, the type and characteristics of those massless elements have to be chosen such that they would not affect the real structural performance.

- Wave and wind loads are actually only calculated based on the elements above and below the still water level in the initial undisplaced position, not taking into account that elements could emerge or submerge during simulation due to the motion of the floater. In addition, the wave load calculation only accounts for relative velocities but not for relative accelerations, as otherwise the initialization of the time-domain simulation in Dymola does not finish. Any loads on the submerged structure due to currents are not included. Furthermore, for correct simulation of floating offshore wind turbines in different sea states, the actual wave height has to be included in the buoyancy calculation.

Those simplifications are rather minor and do not affect the system performance in the free-decay simulations, except for the neglect of the relative acceleration in the wave load calculation, as shown in Section 4.1. However, for an offshore floating wind turbine system, which should represent accurate system performances and valid results for any simulation and load case, the above mentioned points have to be included in the model.

4 Results and Applications

The practical use of the offshore wind turbine model in Modelica is presented by analyzing simulation results based on the implemented code (Section 4.1) and pointing out the feasibility of model adaption (Section 4.2).

4.1 Simulation Results

In order to examine the developed code for a floating offshore wind turbine system, the NREL offshore 5-MW reference wind turbine (Jonkman et al., 2009) on top of a floating spar-buoy, defined in OC3[5] Phase IV (Jonkman, 2010), is implemented in Modelica, based on the created floating wind turbine model presented in Chapter 2, as shown in Figure 6. In order to point out

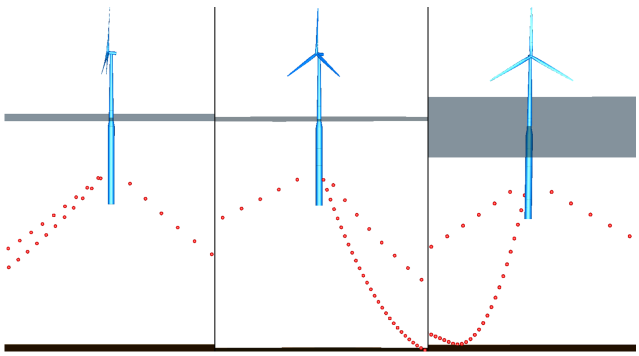


Figure 6. Visualization of the OC3-spar floating wind turbine system (top grey area: water surface, dotted red lines: mooring lines, bottom brown area: seabed).

[5]Offshore Code Comparison Collaboration

DOI 10.3384/ecp17132633

the complexity of the implemented model, some main statistics are presented in Table 1. The simulation settings and performance, as well as the hardware properties are listed in Table 2.

Table 1. Dymola statistics of translated OC3-spar wind turbine model.

Statistical Parameter	*Value*
Continuous time states (scalars)	866
Time-varying variables (scalars)	34,341
Sizes after manipulation of linear systems	{436, 3, 2}
DAE scalar equations	121,139

Table 2. System properties, simulation settings, and performance.

Parameter	*Value*
Clock frequency	3.10 *GHz*
Operating system	64-bit
Simulation interval	600 *s*
Output interval length	0.05 *s*
Solver	Esdirk45a
Tolerance	1.0E-4
CPU time for integration	38,041.8 *s*
CPU time for initialization	83.3 *s*

With this model, free-decay simulations, as specified in OC3 Phase IV (Jonkman et al., 2010), are carried out in Dymola[6]. OC3 mainly focuses on "(1) discussing modeling strategies, (2) developing a suite of benchmark models and simulations, (3) running the simulations and processing the simulation results, and (4) comparing and discussing the results" (Jonkman et al., 2010, pp. 1-2). The OC3 participants, together with their simulation tools, are listed in Figure 7, which represents the legend to Figure 8(a).

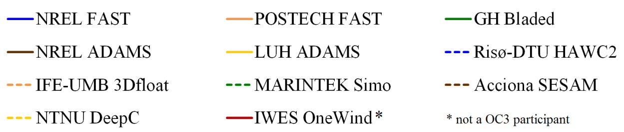
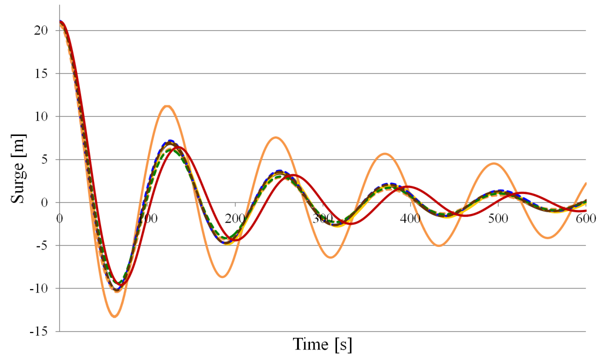


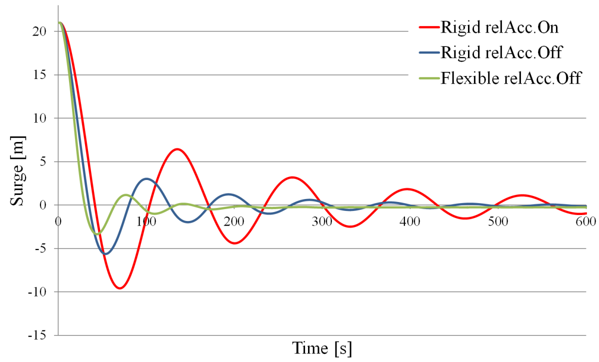
Figure 7. Participants and used simulation tools within the OC3 code-to-code comparison.

The free-decay tests are performed with the fully flexible support structure, while the turbine is modeled as rigid structure. Furthermore, aerodynamic damping is deactivated, so that the hydrodynamic damping can be elaborated in detail. The obtained motion response time series are then compared with the results from the code-to-code comparisons (Jonkman et al., 2010). As, however, the relative acceleration is not included in the wave load calculation, the same simulations are performed with the fully rigid equivalent of the floating wind turbine model, once considering, once neglecting the relative acceleration.

Figure 8 presents the time series of the free-decay simulations exemplarily for the surge DoF. The rigid wind turbine model, taking the relative acceleration into account, yields similar results as obtained by the code-to-code comparison (Jonkman et al., 2010), shown in Figure 8(a). The effect of neglecting the relative acceleration is shown on the rigid model and compared to the results from the fully flexible model, not yet capable of taking this parameter into account. From Figure 8(b) it can be seen that the shorter eigenperiod and stronger damping, obtained by the time series of the fully flexible model, are mainly due to the disregarded relative acceleration in the wave load calculation.



(a) Rigid model and code-to-code comparison results, legend given in Figure 7



(b) Consideration and neglect of relative acceleration

Figure 8. Free-decay time series in surge.

The system response by the end of the decay process turns out to depend on the chosen solver. This, however, is expected to be caused by the damping parameters set in the *TopologyData* record of the *StructureElement*. At this stage, the Rayleigh damping parameters are computed manually, based on the system eigenfrequencies, and

[6]http://www.3ds.com/products-services/catia/products/dymola (Accessed: 22 August 2016)

used for all beam elements. However, for the sake of accuracy and in order to obtain more realistic estimates for the structural damping parameters, it is recommended to compute those Rayleigh damping parameters for each beam element individually and internally within Modelica.

4.2 Model Adaption

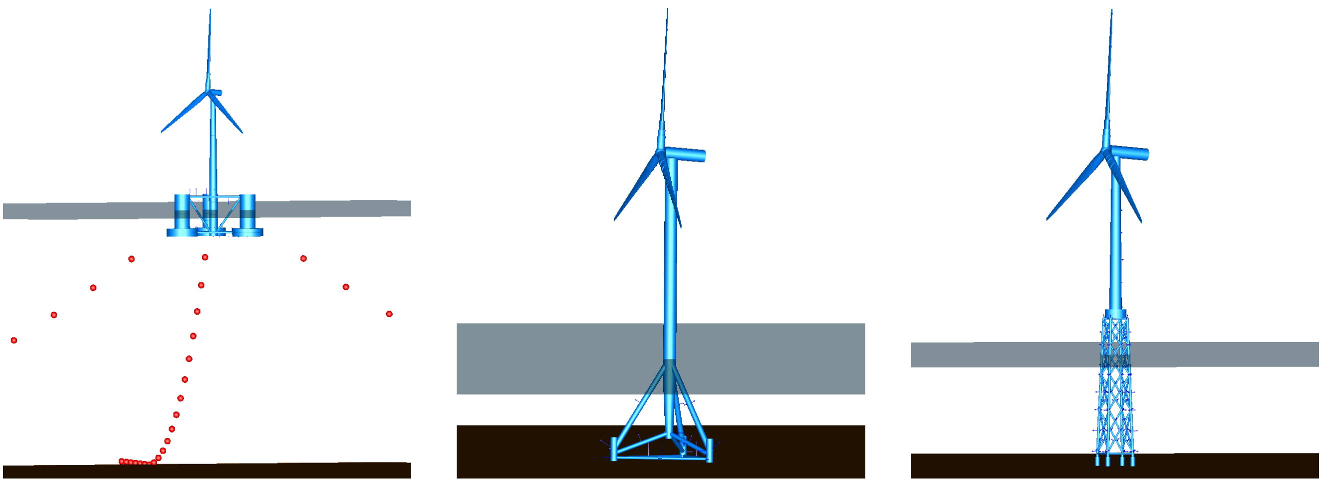
Due to the hierarchical programming in Modelica and the multibody approach, single components can easily be adapted or exchanged. This way, other floating wind turbine designs, bottom-fixed offshore or even onshore wind turbine systems can be modeled, using the basic structure of the implemented fully flexible model for a floating offshore wind turbine system. Thus, the presented model can be used as a simple tool for elaborating new research topics and different or innovative wind turbine system designs.

This flexibility of model adaption is demonstrated on the example of the OC4 semi-submersible platform (Robertson et al., 2014), the OC3 tripod (Nichols et al., 2009), and the OC4 jacket (Jonkman et al., 2012), shown in Figures 9(a), 9(b), and 9(c), respectively. Furthermore, Table 3 compares the complexity of those models, using the same statistics from Dymola as presented in Table 1 for the spar-buoy floating wind turbine model. This underlines the enlarged calculation effort due to the increased number of system parameters, which comes with more complex and highly branched structures. But nevertheless, it is feasible to model and simulate very sophisticated wind turbine system designs.

5 Conclusion and Outlook

This paper presents the modeling of a fully flexible floating offshore wind turbine system in Modelica. Based on the Modelica MultiBody Library and the hierarchical programming structure in Modelica, the complex system is implemented via six main components and several subcomponents. Floating systems bring new challenges, such as buoyancy, free motion, station-keeping system, as well as relative velocities and accelerations. Furthermore, closed loops have to be avoided, when handling branched structures in multibody applications, and certain adapters, as well as a special load frame, are needed to connect external components to the flexible Bernoulli beams.

Due to the complexity of fully flexible modeling of a floating offshore wind turbine system, some simplifications are made. Most of them have minor impact on the behavior of the system and the simulation results. Never-



(a) Semi-submersible (b) Tripod (c) Jacket

Figure 9. Other wind turbine systems, implementation based on the basic model.

Table 3. Dymola statistics of translated adapted wind turbine models.

Statistical Parameter		*Semi-submersible*	*Tripod*	*Jacket*
Continuous time states (scalars)		1,658	796	2,056
Time-varying variables (scalars)		64,139	35,079	75,425
Sizes after manipulation of linear systems	Vector length	78	94	366
	Maximum value	983	499	1,681
DAE scalar equations		237,246	148,435	374,173

 DOI
10.3384/ecp17132633

theless, the proposed methods for accurate modeling have to be implemented in the next stage. More challenging and more relevant for correct simulation results, however, is the inclusion of the relative acceleration in the wave load calculation. In order to account for this, further work on an alternative way to connect external loads to the structure elements is in progress. In this approach, each structure element is made up of two Bernoulli beams, while the mid-node is added separately and connected to a "free" adapter. To avoid too small beam elements, the *TopologyData* record is adapted and the number of nodes and members is reduced. A more realistic estimation of the Rayleigh damping parameters, which could be directly included as internal computation within Modelica, is as well of relevance for obtaining correct system responses. Further fine tuning and detailed examination of the environmental models should be carried out in order to get even closer to the reference results. Finally, with regard to the computational effort and simulation performance, additional work on speeding up the initialization process is recommended.

Thus, the presented model should be seen rather as a work in progress than as a fully established Modelica code, as further work on small but important details is still needed for proper representation of fully flexible floating wind turbine systems. However, the implemented model is a very good basis for simulation of fully flexible floating offshore wind turbines and already reproduces the dynamics of such a complex system quite well. Furthermore, using the Modelica MultiBody Library and the object-oriented programming in Modelica comes with the great advantage to quickly adapt the implemented basic model. This makes it a simple tool, which can be used in other research projects and for modeling of novel wind turbine designs.

Acknowledgements

This work is financially supported by the German Federal Ministry of Economics and Technology, funding code 0325841A.

References

P. Feja. *Dynamische Modellierung von Verankerungsleinen für schwimmende Offshore-Windenergieanlagen mit Modlica*. Bachelor's Thesis, RWTH Aachen University, Fraunhofer Institute for Wind Energy and Energy System Technology (IWES), Germany, 2013.

GL Garrad Hassan. *V4 Bladed Multibody dynamics*. Garrad Hassan & Partners Ltd., Bristol, UK, Bladed User Manual, Version 4.0, 2010.

GL Rules and Guidelines. *IV: Industrial Services, Part 1: Guideline for the Certification of Wind Turbines*. Germanischer Lloyd, Hamburg, Germany, 2010.

International Standard. *Wind turbines - Part1: Design requirements*. International Electrotechnical Commission, Geneva, Switzerland, IEC 61400-1, 3rd edition, 2005.

J. Jonkman. *Definition of the Floating System for Phase IV of OC3*. Technical Report NREL/TP-500-47535, National Renewable Energy Laboratory (NREL), Golden, Colorado, USA, 2010.

J. Jonkman, S. Butterfield, W. Musial, and G. Scott. *Definition of a 5-MW Reference Wind Turbine for Offshore System Development*. Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory (NREL), Golden, Colorado, USA, 2009.

J. Jonkman, T. Larsen, A. Hansen, T. Nygaard, K. Maus, M. Karimirad, Z. Gao, T. Moan, I. Fylling, J. Nichols, M. Kohlmeier, J. Pascual Vergara, D. Merino, W. Shi, and H. Park. 'Offshore Code Comparison Collaboration within IEA Wind Task 23: Phase IV Results Regarding Floating Wind Turbine Modeling'. In *2010 European Wind Energy Conference and Exhibition (EWEC), Warsaw, Poland*, April 2010. Conference Paper NREL/CP-500-47534. doi: 10.13140/2.1.3576.5768.

J. Jonkman, A. Robertson, W. Popko, F. Vorpahl, A. Zuga, M. Kohlmeier, T.J. Larsen, A. Yde, K. Saetertro, K.M. Okstad, J. Nichols, T.A. Nygaard, Z. Gao, D. Manolas, K. Kim, Q. Yu, W. Shi, H. Park, A. Vasquez-Rojas, J. Dubois, D. Kaufer, P. Thomassen, M.J. de Ruiter, J.M. Peeringa, H. Zhiwen, and H. von Waaden. 'Offshore Code Comparison Collaboration Continuation (OC4), Phase I - Results of Coupled Simulations of an Offshore Wind Turbine with Jacket Support Structure'. In *Proceedings of the 22nd International Society of Offshore and Polar Engineers Conference, 17-22 June 2012, Rhodes, Greece*, pages 337–346, June 2012. Conference Paper NREL/CP-5000-54124.

T.J. Larsen and A.M. Hansen. *How 2 HAWC2, the user's manual*. Risø-R-Report Risø-R-1597(ver. 4-5), Risø National Laboratory, 2014.

J. Nichols, T. Camp, J. Jonkman, S. Butterfield, T. Larsen, A. Hansen, J. Azcona, A. Martinez, X. Munduate, F. Vorpahl, S. Kleinhansl, M. Kohlmeier, T. Kossel, C. Böker, and D. Kaufer. 'Offshore Code Comparison Collaboration within IEA Wind Annex XXIII: Phase III Results Regarding Tripod Support Structure Modeling'. In *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition, 5-8 January 2009, Orlando, Florida, USA*, January 2009. Conference Paper NREL/CP-500-44810.

M. Otter. *Modeling, Simulation and Control with Modelica 3.0 and Dymola 7*. Technical Report, Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR) - Institut für Robotik und Mechatronik, Wessling, Germany, 2009.

A. Robertson, J.M. Jonkman, F. Vorpahl, W. Popko, J. Qvist, L. Frøyd, X. Chen, J. Azcona, E. Uzunoglu, C. Guedes Soares, C. Luan, H. Yutong, F. Pengcheng, A. Yde, T.J. Larsen, J. Nichols, R. Buils, L. Lei, T.A. Nygaard, D. Manolas, A. Heege, S. Ringdalen Vatne, T. Duarte, C. Godreau, H.F. Hansen, A.W. Nielsen, H. Riber, C. Le Cunff, F. Beyer, A. Yamaguchi, K.J. Jung, H. Shin, W. Shi, H. Park, and M. Alves. 'Offshore Code Comparison Collaboration

Continuation Within IEA Wind Task 30: Phase II Results Regarding a Floating Semisubmersible Wind System'. In *Proceedings of the ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering, OMAE 2014, San Francisco, California, USA*, volume 9B, OMAE2014-24040, page V09BT09A012. American Society of Mechanical Engineers, June 2014. doi:10.1115/OMAE2014-24040.

M. Strobel, F. Vorpahl, C. Hillmann, X. Gu, A. Zuga, and U. Wihlfahrt. 'The OnWind Modelica Library for Offshore Wind Turbines - Implementation and first results'. In *Proceedings of the 8th International Modelica Conference 2011, Dresden, Germany*, pages 603–609. Modelica Association, March 2011.

P. Thomas, X. Gu, R. Samlaus, C. Hillmann, and U. Wihlfahrt. 'The OneWind® Modelica Library for Wind Turbine Simulation with Flexible Structure - Modal Reduction Method in Modelica'. In *Proceedings of the 10th International Modelica Conference 2014, Lund, Sweden*, pages 940–948. Modelica Association, March 2014. doi:10.3384/ECP14096939.

T. Tran, D. Kim, and J. Song. 'Computational Fluid Dynamic Analysis of a Floating Offshore Wind Turbine Experiencing Platform Pitching Motion'. *Energies*, 7(8):5011–5026, 2014. doi:10.3390/en7085011.

W. Yu. *Dynamic Modeling of a Floating Wind Turbine*. Technical Report, Fraunhofer Institute for Wind Energy and Energy System Technology (IWES), Germany, 2015.