

The OneWind[®] Modelica Library for Wind Turbine Simulation with Flexible Structure — Modal Reduction Method in Modelica

Philipp Thomas Xin Gu Roland Samlaus Claudio Hillmann Urs Wihlfahrt
Fraunhofer Institute for Wind Energy and Energy System Technology IWES
Am Seedeich 45, 27572 Bremerhaven, Germany

Abstract

The OneWind[®] Modelica Library¹ [15] for coupled wind turbine loads calculation developed at Fraunhofer IWES uses a structural element based on a modal reduction method to model the motion and deformation of flexible wind turbine rotor blades and tower. The degrees of freedom (DOF) are rigid body motions and modal DOF. The `ModalElement` model allows the simulation of coupling effects like bend-twist coupling in wind turbine rotor blades and the structural behavior is dependent on the selected eigenmodes. This paper gives an overview about the Modelica implementation of the theory of modal elements, the advantages over other methods (finite-elements), how the `ModalElement` model is included into the OneWind[®] Modelica Library, and how it is used for load calculation.

Keywords: modal, blade, OneWind, OneModelica, load calculation, wind turbine loads, coupled wind turbine simulation

1 Introduction and Motivation

In wind turbine simulations, flexible multibody dynamics is widely used to describe the large motion and flexible deformation of blades. A flexible multibody system is based on the floating Frame of reference formulation. In a standard flexible multibody method, a finite element discretization is often combined with the multibody formulation to describe the beam flexibility. In the finite element method, the flexible beam is discretized into several beam elements. Each element is described with local shape functions weighted with nodal displacements. The numerical accuracy of the finite element method is dependent on the number of elements. A large number of elements demand quite a

lot computational costs, which are usually not suitable for wind turbine simulation.

A modal reduction approach can dramatically reduce the number of DOF of the beam by transforming nodal coordinates to modal coordinates. This approach is based on the assumption that the displacement of the beam can be expressed as the superposition of a series of eigenmodes weighted with modal coordinates.

The objective is to achieve the same accuracy as the finite element method with less computational time. This is a requirement for a wind turbine load calculation with a large number of load cases which are necessary for fatigue analysis. Therefore, the modal reduction method is used to create a structural element in the Modelica programming language which can represent the motion and deformation of flexible components of a wind turbine. With the OneWind[®] Modelica Library, a package of necessary Modelica models for aero-servo-elastic wind turbine simulations is finally available.

The paper is structured as follows: Section 2 describes the theoretical approach of flexible multibody and concludes with the equation of motion in modal coordinates. Section 3 shows the implementation of the flexible multibody with modal reduction in Modelica. In Section 4, an overview of the OneWind[®] Modelica Library with `ModalElement` model for rotor blades and tower is given. Moreover, the internal layout of wind turbine components is shown with a focus on interaction between models. Section 5 shows results of wind turbine deflection with the `ModalElement` model for flexible tower. Lastly, Section 6 concludes the presented work and gives an outlook for further applications of the `ModalElement` model.

¹Version 1.0 for onshore wind turbine load calculation was released and is now available under dual license model. For further information contact: info@onewind.de

2 Flexible multibody with modal reduction

The ModalElement model is developed based on multibody dynamics and a modal reduction approach. The multibody system is based on the floating reference Frame. The floating reference Frame indicates that the deformation of the body can be formulated with respect to the corresponding body coordinate system, which moves together with the body. The absolute motion of an arbitrary point P on the body is defined by

$$r_P = R + A(\bar{u}_0 + \bar{u}_f), \quad (1)$$

where R represents the location of the origin of the body reference, A is a transformation matrix, \bar{u}_0 is the local position vector in undeformed state, and \bar{u}_f is the deformation vector.

The representation of a deformable body in the world Frame requires $6 + N$ DOF, which includes 6 rigid translational and rotational DOF and N flexible DOF

$$q = [q_r \quad q_f]^T = [R \quad \theta \quad q_f]^T, \quad (2)$$

where θ are the orientation angles of the reference Frame and q_f is the N -dimensional vector of flexible DOF. In the Modelica model, q_f is formulated based on the finite element method and thus, q_f represents the nodal displacements in the reference Frame.

Using the principle of virtual work [14], the equations of motion for a flexible body in a multibody system can be described as

$$\begin{bmatrix} m_{RR} & m_{R\theta} & m_{Rf} \\ m_{\theta R} & m_{\theta\theta} & m_{\theta f} \\ m_{fR} & m_{f\theta} & m_{ff} \end{bmatrix} \begin{bmatrix} \ddot{R} \\ \ddot{\theta} \\ \ddot{q}_f \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & K_{ff} \end{bmatrix} \begin{bmatrix} R \\ \theta \\ q_f \end{bmatrix} + \begin{bmatrix} C_R^T \\ C_\theta^T \\ C_f^T \end{bmatrix} \lambda = \begin{bmatrix} Q_{eR} \\ Q_{e\theta} \\ Q_{ef} \end{bmatrix} + \begin{bmatrix} Q_{vR} \\ Q_{v\theta} \\ Q_{vf} \end{bmatrix}. \quad (3)$$

A modal approach is implemented in the Modelica model to reduce the size of the system. In the modal approach, the specified displacement fields are formulated with the eigenvectors of a series of eigenmodes. The generalized displacement expression can be written as

$$q_f = \sum_{k=1}^n \phi_k Y_k, \quad (4)$$

where ϕ_k is the eigenvector in the k^{th} mode and Y_k is the generalized modal coordinate in the k^{th} mode. The modal coordinate is time dependent, which is solved together with the multibody system. The eigenvector

is calculated in the preprocessing with OneModelica (see Section 4.6).

Equation (4) can be rewritten in the matrix form as

$$q_f = \phi Y, \quad (5)$$

where Y are the modal coordinates and ϕ is the eigenvector matrix given by

$$\phi = [\phi_1 \quad \phi_2 \quad \dots \quad \phi_n]. \quad (6)$$

This procedure is the modal displacement superposition method. The number n represents the number of DOF in modal analysis. According to the characteristics of modal analysis, the first several modes dominate the contribution to the displacement field. A proper selection of these modes leads to a good approximation of displacements with a small number of DOF. For example, if m eigenmodes are selected, q_f can be described approximately by

$$q_f \approx \phi_m Y_m. \quad (7)$$

Using the eigenvector matrix as the modal transformation matrix, Equation (3) can be written in terms of modal coordinates as

$$\begin{bmatrix} m_{RR} & m_{R\theta} & m_{Rf}\phi_m \\ m_{\theta R} & m_{\theta\theta} & m_{\theta f}\phi_m \\ \phi_m^T m_{fR} & \phi_m^T m_{f\theta} & \phi_m^T m_{ff}\phi_m \end{bmatrix} \begin{bmatrix} \ddot{R} \\ \ddot{\theta} \\ \ddot{Y}_m \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \phi_m^T K_{ff} \phi_m \end{bmatrix} \begin{bmatrix} R \\ \theta \\ Y_m \end{bmatrix} + \begin{bmatrix} C_R^T \\ C_\theta^T \\ C_f^T \phi_m \end{bmatrix} \lambda = \begin{bmatrix} Q_{eR} \\ Q_{e\theta} \\ \phi_m^T Q_{ef} \end{bmatrix} + \begin{bmatrix} Q_{vR} \\ Q_{v\theta} \\ \phi_m^T Q_{vf} \end{bmatrix}, \quad (8)$$

where all matrices related to flexible DOF are modal reduced if the number of modal coordinates Y_m is less than the number of nodal coordinates q_f .

3 Implementation in Modelica

A flexible multibody q_f model with modal reduction has been implemented by using the Modelica language. This model is named ModalElement (see Figure 1) in the OneWind[®] Modelica Library. Frame_a and Frame_b are the interfaces which are used to connect to other structural components. Frame_c is the interface which is used to connect external loads.

The modal shape matrix, the modal reduced stiffness matrix, and the modal reduced mass matrix are

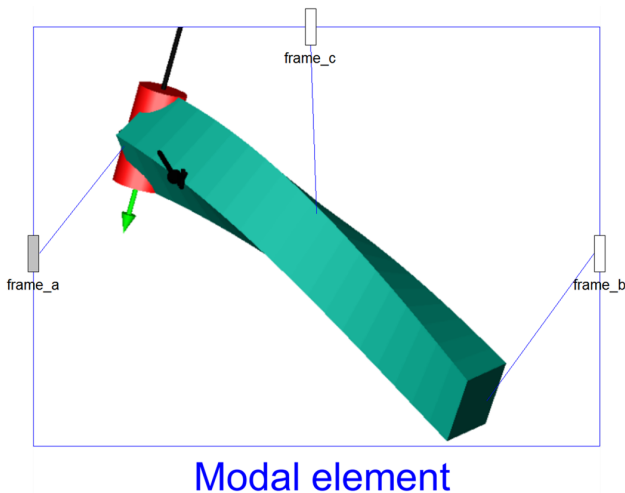


Figure 1: ModalElement icon

calculated in Java in a preprocessing step due to problems with calculating large matrices in Modelica (the details are described in Section 4.6).

The input modal reduced representations of the model are as follows:

```
parameter Real phi[divisions*6,nModes]
=modalElementData.phiMatrix;
parameter Real modalReducedStiffnessMatrix[nModes,nModes]=modalElementData.modalReducedStiffnessMatrix;
parameter Real modalReducedMassMatrix[nModes,nModes]=modalElementData.modalReducedMassMatrix;
```

Listing 1: Modal reduced representations

Dynamic equations (see Equation (8)) of ModalElement with damping are implemented in Modelica language as follows:

```
M*ddQ + D*dQ[7:6+nModes] + K*Q[7:6+nModes]
+ CqT[1:6+nModes,:]*lambda = Qe + Qv;
```

Listing 2: Dynamics equations

3.1 Boundary conditions

To solve a finite element problem, suitable boundary conditions are defined in the finite element system. In a pure finite element system, boundary conditions are usually enforced externally. However, in our flexible multibody formulation, boundary conditions are enforced internally in the model. This is because the floating Frame only exchanges the location of the origin of the body reference, the orientation of the body

reference, and the forces with other components. The deformation displacements of the flexible multibody model are not accessible externally.

The boundary condition for ModalElement is the “clamped-free” boundary condition. In this model all six flexible DOF are fixed at the root of the beam. Both displacements and rotations are equal to zero as

$$q_{f1} = q_{f2} = q_{f3} = q_{f4} = q_{f5} = q_{f6} = 0. \quad (9)$$

3.2 Animation

Animation is an important feature in wind turbine simulation. The immediate visual feedback about the dynamic behavior of the wind turbine helps the user with troubleshoot during a simulation. The 3D animation of ModalElement is realized by using the Surface function in Dymola with the following code:

```
model ModalElementSurface
extends Modelica.Mechanics.MultiBody.Visualizers.Advanced.Surface(redeclare function surfaceCharacteristic=OneWind.BeamElement.FlexibleMultibody.BeamVisualizer.BeamVisualizer (
    radius=radius,
    extra=1,
    r=r_i,
    qf=qf,
    angle=angle,
    der_angle=der_angle,
    scalingFactor=scalingFactor));
parameter Integer nl "number of visualization points along length";
parameter Real scalingFactor "scaling factor to visualize deformation";
input Modelica.SIunits.Position r_i[nl,3] "initial Position of visualization points";
input Modelica.SIunits.Distance qff[nl,3] "displacements of visualization points";
input Modelica.SIunits.Angle angle[nl,3] "angle at visualization points";
input Modelica.SIunits.AngularVelocity der_angle[nl,3] "angular velocity at visualization points";
end ModalElementSurface;
```

Listing 3: 3D ModalElement animation

4 OneWind[®] Modelica Library for coupled flexible wind turbine simulation

The OneWind[®] Modelica Library is based on the MultiBody Library from the Modelica Standard Library and contains all models for aero-servo-elastic load simulation. The models are grouped according to wind turbine components as seen in Figure 2 for an

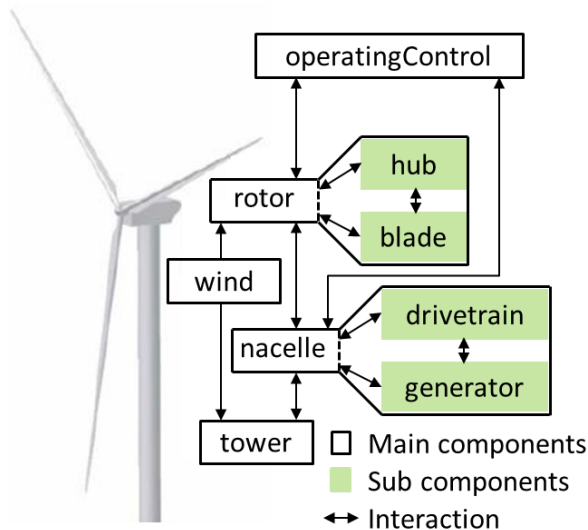


Figure 2: Onshore wind turbine with corresponding component instances and interaction

onshore wind turbine. This Figure shows the model instances and corresponding interaction for the main wind turbine components: the rotor (with the sub-components of blade and hub), the tower, the nacelle (with the sub-components of drivetrain and generator), operating control, and the wind. The corresponding Modelica model for the main wind turbine components is written as:

```

model WindTurbine
  extends OneWind.OnshoreWindTurbine
  (
    //=== rotor ===
    redeclare OneWind.RotorModal rotor
    //=== tower ===
    ,redeclare OneWind.TowerModal tower
    //=== nacelle ===
    ,redeclare OneWind.NacelleRigid nacelle
    //=== operating control ===
    ,redeclare OneWind.Control operatingControl
    //=== wind ===
    ,redeclare OneWind.WindTurbulent wind
  );
end WindTurbine;

```

Listing 4: Modelica representation of Figure 2

To simulate at different levels of detail, several models for almost every wind turbine component are available. To exchange component models, the Modelica concept of replacing classes in the modifier with redeclare (with corresponding replaceable), as seen in Listing 3 and Listing 4, is used. The model OnshoreWindTurbine contains the declaration of instances of necessary wind turbine component models with all connections. These models are

exchanged in WindTurbine while extending from the OnshoreWindTurbine model. To ensure subtype compatibility, models that represent one turbine component extend the same partial model, which contain a replaceable, basic parameter record definition, and the external interface (Figure 3).

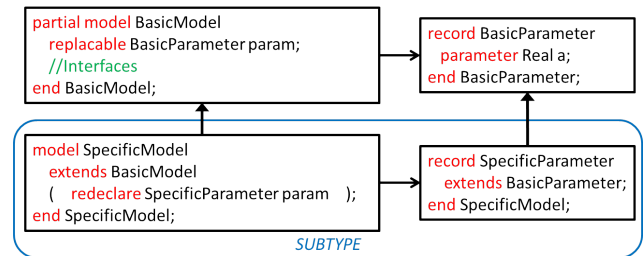


Figure 3: Inheritance concept for subtype compatibility

4.1 Rotor model

The rotor model is the most important component of the wind turbine since it transforms wind velocity through aerodynamic lift to torque, which drives the electrical generator.

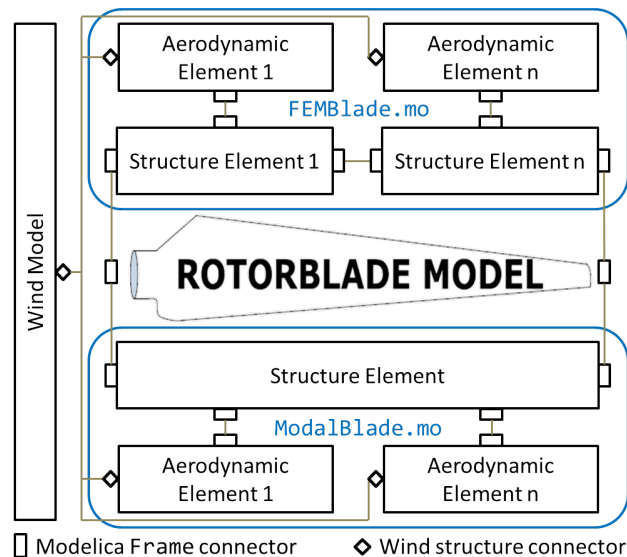


Figure 4: Different layout between the modal and finite element model and interaction between the structure model and the aerodynamic model in the rotor blade model

The rotor blade model consists of two main components which are connected via the Frame connector: the aerodynamic load element and the structure ele-

ment. In Figure 4, the connector scheme and the interaction between the structure model and the aerodynamic model is shown. The rotor blade models differ in regards to their structural part. The `ModalBlade` model consists of only one structure element, the `ModalElement` model, whereas the `FEMBlade` modal consists of n structure elements (in case of the NREL 5-MW reference wind turbine [7], n is 17 per blade and 11 for the tower). The structure elements are internally connected to the blade root, the blade tip, and to the aerodynamic load element through the use of the `Frame` connector. All rotor blade models extend a `PartialBlade` model, where the aerodynamic load element model is defined as well as the external interface root `Frame`, tip `Frame`, and the specific connector `StructureWind`, which is used to connect the wind velocity and position to the aerodynamic load element.

4.1.1 Wind models

Deterministic and stochastic wind models are available. Deterministic wind uses several gust models. The stochastic wind data is read from binary or ASCII file in TurbSim [8] format.

All wind models consider the effect of the tower on wind velocity: in case of upwind, the potential flow method is applied and in case of downwind, an empirical model is used. Furthermore, exponential or logarithmic wind shear and vertical and horizontal inclination angle are accounted. The wind velocity is given as an input to the `StructureWind` connector.

All available wind models fulfill requirements defined in IEC 61400-3 [5].

4.1.2 Aerodynamic models

Inside the aerodynamic load element, the `StructureWind` connector delivers the wind velocity which is used to calculate aerodynamic forces and moments. The aerodynamic load calculation uses either the blade element momentum theory (BEM, described in [11], [2], and extended in [1], [3]) or the general dynamic wake (GDW, described in [4], [16]) method. The BEM is implemented as an iterative algorithm while the GDW is a set of differential equations. For the BEM, corrections for the dynamic wake and the dynamic stall can be used. Dynamic stall correction is also available for GDW. The aerodynamic forces and moments are given as an input to aero load `Frame` connector.

4.1.3 Blade models

Blade models differ in regards to their representation of flexible body motion. The `ModalBlade` model uses modal degrees of freedom, the `FEMBlade` model based on the Bernoulli beam element formulation uses 6 degrees of freedom at each node, and the `RigidBlade` model has no flexibility at all (only rigid body motion). The flexible blades consider centrifugal stiffening and pre-twist. All formulations are linear which are the standard in wind turbine load calculations. Non-linear formulations for large deflections but small deformations are under development.

Aerodynamic loads from the aero load `Frame` connector act as external forces and moments and lead to the rotation of the blades. The resulting forces and moments are transferred to the blade root `Frame` connector. Aero-Structure-Coupling enables the simulation of the coupling between the rotor blade deflection and the aerodynamic load calculation.

4.2 Tower models

As for blade models, rigid and flexible models are available for the tower. The structure is basically the same: on the outside, the tower consists of a bottom `Frame`, a tip `Frame`, and load `Frame` connectors along its vertical length. Internally, the tower consists of the structure elements for load and deflection calculation and load elements for the calculation of the aerodynamic drag force from the wind. The connections are realized via `Frame` connectors (except for wind which uses a specific connector similar to the rotor model).

4.3 Control system

The operating control system is implemented as discrete algorithms based on [7] and consists of a PI-pitch algorithm to control the power production above rated rotor rotation speed and a generator-torque controller which requests the counter torque from the electrical generator due to rotor torque. A single pole, low pass speed filter with cut-off frequency of 0.25 Hz transfers the continuous generator rotation speed signal to a discrete signal for control logic. For communication of control signals between sensors inside the wind turbine components and operating control, the `SignalBus` and the `SignalSubBus` connectors are used to build a hierarchic network of expandable connectors (as described in [9]).

4.4 Offshore wind turbine components

Several physical models in the Modelica language for the dynamics of offshore wind turbines and the modeling of water and waves are under development. Fraunhofer IWES is a participant of IEA Wind and the Cooperative Agreement² with OC3 [17] and OC4 [10] projects to compare offshore wind turbine simulation codes, where the developed models for offshore wind turbine simulation are verified (see [18]).

4.5 Interfaces between component models

The OneWind[®] Modelica Library basically uses three types of interfaces:

- Modelica Frame connector,
- Modelica FFlange connector,
- Component specific connector.

The Frame connector is utilized whenever loads are exchanged between different models and so does the aerodynamic load element calculate forces and moments from wind velocity using an aerodynamic theory. The transfer between aerodynamic load element and rotor blade is done via a Frame connector.

The FFlange connector is used for a limited degree of freedom representation of the drivetrain dynamics and transfers the torque and angle from the rotor to the electrical generator.

Additionally, user-defined connectors are used to transfer specific information between models that are not covered by the Frame or FFlange connector. Figure 4 exemplarily shows a user defined connector that transfers the wind velocity and position along the structure components to the corresponding aerodynamic load element.

4.6 OneWind[®] Modelica Library with OneModelica

The Modelica IDE OneModelica [12], which is based on the Eclipse RCP framework, is being developed at Fraunhofer IWES. Figure 5 illustrates the workflow of the Modelica model development with OneModelica.

Modelica models can be developed by reusing models from libraries like the Modelica Standard Library and the OneWind[®] Modelica Library. Both libraries are integrated into OneModelica by default. Additionally, OneModelica also provides functionality for the generation of turbulent wind (according to Kaimal [6]). Input files for Modelica models are generated in

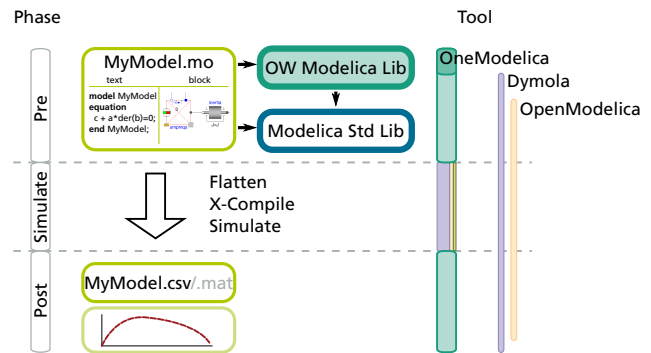


Figure 5: Modelica workflow with OneModelica

binary or ASCII format. The transformation of structure components into model reduced representations has been implemented in Java. This was necessary due to problems with large matrices when calculating the eigenfrequencies directly in Modelica.

In OneModelica, input models for the transformation are automatically recognized and a marker is added to the model files. Input models need to be of type record and must extend the record ModalInput which is included in the OneWind[®] Modelica Library. By right-clicking on the model file, the user can open the transformation dialog and provide a path and file name for the generated modal representation.

The record ModalInput defines distributed structural properties. At the user defined record level, parameters have to be redefined and the needed information for the FEM structure model have to be provided. In this way, the usage of the same data for modal and finite element model is ensured. The parameters are then extracted from the Modelica file and used by the transformation code developed with Java. The generated Modelica record with modal data contains the modal reduced mass and the stiffness matrix as well as the eigenforms and can be used with modal components. Changes in the structure model can easily be transferred to the modal component by the regeneration of the modal record.

The same workflow is used for the generation of turbulent wind data. A template record must be extended and parameterized. The generated wind file can then be used by a Modelica wind field generator that is included in the OneWind[®] Modelica Library. The user can re-use the generator and provide a path to the generated wind file.

Besides the described functionality, OneModelica supports several Modelica simulation tools for doing the flattening, compiling, and simulation and several simulations can be run in parallel. After a simula-

²http://www.ieawind.org/about_co-operative_agree.html

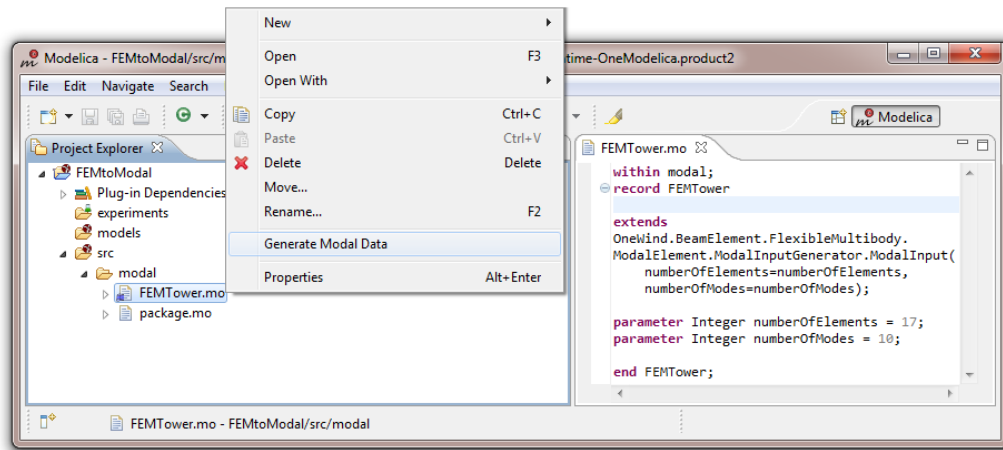


Figure 6: Transforming structure models to modal format

tion, the results are transferred back to OneModelica for post processing. Simulation settings and results are stored. The latest enhancement in OneModelica is an integrated framework for the automatic Modelica model testing [13].

5 Simulation Results

In the following section, simulation results with flexible wind turbine components are presented. All mentioned results were computed in Dymola³.

Figure 7 compares the deflection obtained with the ModalTower model and the finite element based TubularTowerModel model at the tower top Frame connector in the x- (downwind) and y-direction (side-to-side). The tower model is parametrized according to the NREL 5-MW reference wind turbine with the addition of a constant three dimensional force at the tower top. Both flexible models oscillate at the same frequency at each respective direction. The ModalTower model results strongly depend on the number of modes. The selected modes from the modal reduction must be able to represent the motion, which are enforced to the structure by external loading. Table 1 shows the error for a selection of eigenmodes (for a detailed view on convergence of error up to 30 Modes, see Figure 8). The reference is always the finite element model. Since the first eigenmode is only a 1-directional bending mode, it can only represent the bending in x-direction. Using more than 6 modes causes the error to drop below 1% in both directions. This means that 6 flexible (modal) DOF are enough

to represent the deflection of the ModalTower with almost the same accuracy as with TubularTowerModel. A reduced number of DOF will result in less computational time, especially with coupled wind turbine simulation.

To compare CPU-time, the load simulation of a aero-servo-elastic wind turbine is utilized. The model is parametrized according to the NREL 5-MW reference wind turbine. For the tower, the ModalTower model with six eigenmodes or the finite element based TubularTowerModel model is used. Integration algorithms are DASSL and Euler mixed⁴. For ModalElement based models, it is possible to choose Rkfix4⁵, which is not possible for the finite element model where no solution is found. Table 2 contains the results. As mentioned before, a reduced number in flexible DOF result in a reduction of computational time. The increase in simulation performance depends on the integration algorithm and lies between 300% and 550% for Euler mixed and DASSL. With Rkfix4, the simulation reaches real-time abilities.

6 Conclusion

This paper describes the approach and the implementation of a flexible structure element based on a modal reduction method in the Modelica programming language and its integration into the OneWind[®] Modelica Library for aero-servo-elastic load simulations of wind turbines.

The ModalElement model delivers comparable structural behavior as the finite element model. The

³Dymola 2014 FD01, <http://www.dymola.com>

⁴Mixed explicit/implicit Euler algorithm

⁵Runge-Kutta 4th order integration algorithm

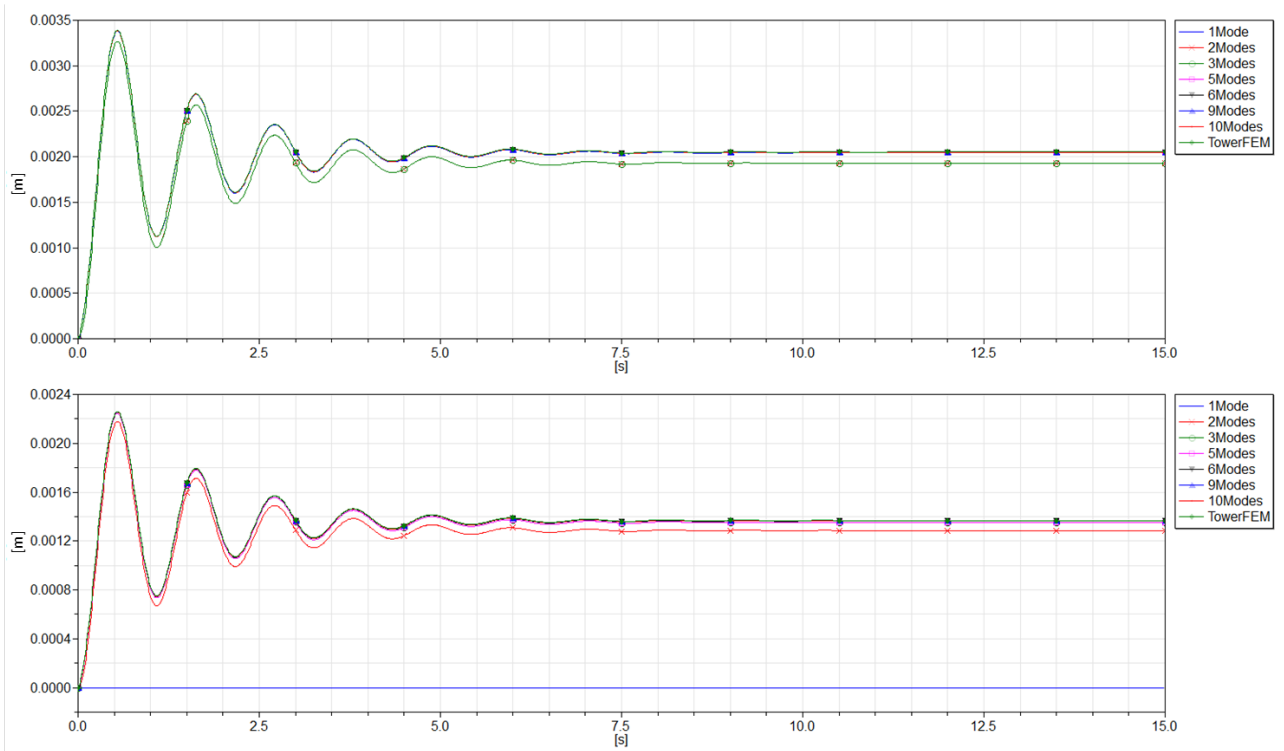


Figure 7: Tower top deflection: upper plot shows downwind displacement, lower plot shows side-to-side displacement

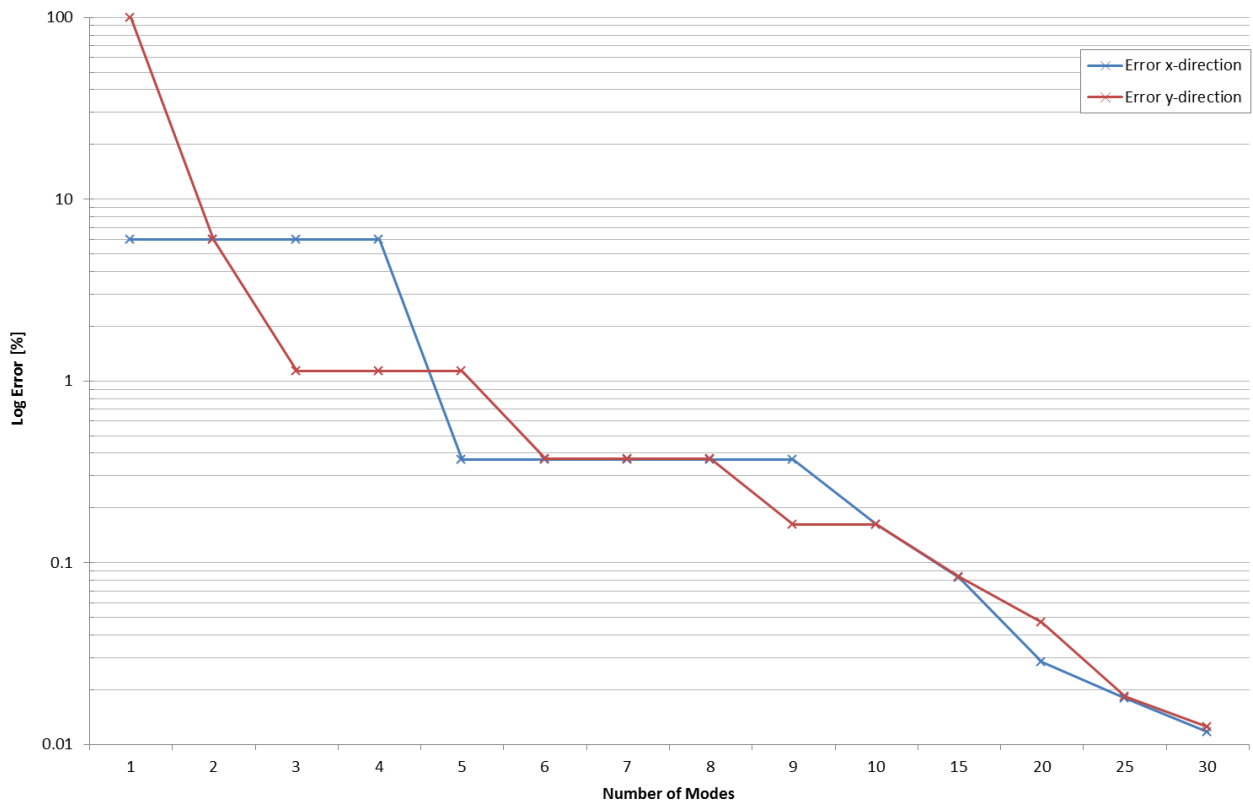


Figure 8: Simulation error for different numbers of Eigenmodes

Table 1: Simulation error for different number of eigenmodes; range: 1 mode (1-10) and 5 modes (10-30)

No. of Modes	x-Deflection [m]	Error [%]
1	1.9292E-03	5.989
⋮	⋮	⋮
5	2.0445E-03	0.371
⋮	⋮	⋮
10	2.0488E-03	0.161
15	2.0504E-03	0.083
20	2.0516E-03	0.028
25	2.0518E-03	0.018
30	2.0519E-03	0.012
FEM	2.0521E-03	-

No. of Modes	y-Deflection [m]	Error [%]
1	8.1974E-14	100.0
2	1.2862E-03	5.9901
3	1.3526E-03	1.1315
⋮	⋮	⋮
6	1.3630E-03	0.3713
⋮	⋮	⋮
9	1.3659E-03	0.1615
10	1.3659E-03	0.1615
15	1.3670E-03	0.0833
20	1.3675E-03	0.0468
25	1.3679E-03	0.0183
30	1.3679E-03	0.0124
FEM	2.0521E-03	-

Table 2: CPU-time for 10 s simulations of a wind turbine with a flexible tower

Algorithm	FEM	Modal	Speed-up
Euler mixed	187 s	61 s	307%
DASSL	2230 s	400 s	557.5%
Rkfix4	-	8.6 s	-

models oscillate with the same frequency. The error in deflection calculation drops below 1% beginning from the use of 6 eigenmodes. At this configuration, the simulation results are almost the same as with the finite element model and are available at a fraction of the finite element model computational time (see Table 2). Therefore, the ModalElement model is applicable for the load simulations of a flexible wind turbine.

The next step will be the application of the ModalElement model on substructure components for offshore wind turbine simulations and the integration to the OneWind[®] Modelica Library together with models for water and waves. Further development of the OneWind[®] Modelica Library aims to add:

- A structure element model for large deflections and large rotations,
- A state-machine based supervisory control,
- An interface for Bladed-style DLL for the controller,
- Real-time capability for a coupled flexible wind turbine simulation.

Acknowledgements

This work is financially supported by the Federal Ministry for the Environment, Nature Conservation, and Nuclear Safety based on a decision by the Parliament of the Federal Republic of Germany.

References

- [1] Betz A. Das Maximum der theoretisch möglichen Ausnutzung des Windes durch Windmotoren. In: Zeitschrift für das gesamte Turbinenwesen, 26:307-309, 1920.
- [2] Froude R.E. On the part played in propulsion by differences of fluid pressure, translations of the Institution of Naval Architects, 30:390-405, 1889.
- [3] Glauert H., Division L. Airplane propellers, aerodynamic theory, volume 4, Durand WF, Berlin, Germany, 1935.
- [4] He C. Development and Application of a Generalized Dynamic Wake Theory for Lifting Rotors, PhD thesis, Georgia Institute of Technology, 1989.
- [5] IEC. Wind turbines – Part 3: Design requirements for offshore wind turbines, IEC 61400-3, 1.0 edition, 2009.

- [6] Izumi Y., Kaimal J.C., Wyngaard J.C., Cote O.R. Spectral characteristics of surface-layer turbulence. In: Q.J.R. Meteorol. Soc., volume 98, pp. 563-598, 1972.
- [7] Jonkman J., Butterfield S., Musial W., Scott G. 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.
- [8] Jonkman J., Kilcher L. TurbSim user's guide: version 1.06.00, National Renewable Energy Laboratory (NREL), Golden, Colorado, USA, 2012.
- [9] Otter M. 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.
- [10] Popko W., Vorpahl F. Offshore Code Comparison Collaboration (OC4) Project – Task 30. In: IEA Wind 2012 Annual Report, Chapter 10, 45-48, PWT Communications, July 2013.
- [11] Rankine W.J. On the mechanical principles of the action of propellers, translations of the Institution of Naval Architects, 6:13-30, 1865.
- [12] Samlaus R, Hillmann C, Demuth B, Krebs M. Towards a model driven Modelica IDE. In: Proceedings of the 8th International Modelica Conference 2011, Dresden, Germany, Modelica Association, 20-22 March 2011.
- [13] Samlaus R., Strach M., Hillmann C., Fritzson P. MoUnit – A framework for automatic Modelica model testing, 10th International Modelica Conference 2014, Lund, Schweden, 2014.
- [14] Shabana, Ahmed A. Dynamics of multibody systems, Cambridge University Press, New York, USA, 2005.
- [15] Strobel M., Vorpahl R., Hillmann C., Gu X., Zuga A., Wihlfahrt U. The OnWind Modelica Library for offshore wind turbines – Implementation and first results. In: Proceedings of the 8th International Modelica Conference 2011, Dresden, Germany, Modelica Association, pp. 603 - 609, 20-22 March 2011.
- [16] Suzuki, A. Application of Dynamic Inflow Theory to Wind Turbine Rotors, PhD thesis, The University of Utah, 2000.
- [17] Vorpahl F., Strobel M., Jonkman J., Larsen T., Passon P. and Nichols J. Verification of aeroelastic offshore wind turbine design codes under IEA Wind Task XXIII. Wind Energy, doi: 10.1002/we.1588.
- [18] Wojciech P., Vorpahl F. et al. 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 Offshore and Polar Engineering Conference (ISOPE), Rhodes, Greece, International Society of Offshore and Polar Engineers, pp. 337-346, June 2012.