

Coupling of HAWC2 and Matlab: Towards an Integrated Simulation Platform

Braulio Barahona, Lars C. Henriksen, Anca D. Hansen,
Nicolaos A. Cutululis and Poul Sørensen
Risø DTU, Denmark

April 22nd, 2010

Abstract

In power systems with high percentage of electricity produced by wind power plants, wind turbines will be expected to perform under uncommon situations. For example: voltage faults, frequency deviations, and storms. Therefore, it is important to have a broad understanding of the different dynamic phenomena of wind turbines. Taking into consideration the interactions amongst aeroelastic, structural, electrical, and control systems of the wind turbine and the grid. An approach towards integrated design and analysis of wind turbines is to couple software dedicated to aeroelastic and structural simulations with software dedicated to control, and power system simulations. This work presents the coupling of the aeroelastic simulation tool HAWC2, and Matlab/Simulink, using shared libraries and inter-process communication. Simulations of a semi-variable speed wind turbine are presented to demonstrate the interface.

1 Introduction

Wind energy has become an essential part of the solutions to scarcity of fossil fuels, and high CO₂ emissions. The percentage of electricity produced by wind power plants has become significant in several countries. For example in Denmark, Portugal, and Spain more than 10% of the electricity usage is produced from wind energy [1]. In this context, wind power plants will be expected to perform under uncommon situations such as: voltage faults, frequency deviations, and storms. Therefore, it is important to have a broad view of the design and analysis of wind turbines that considers the interactions amongst aeroelastic, structural, electrical and control systems of the wind turbine and the grid.

A step towards integrated design of wind turbines is to interface different software that is dedicated to specific areas of wind turbine design and analysis. For example, the work in [2] presents the coupling of a wind turbine simulation code (FLEX5) and a finite element (FE) code (Poseidon) with the objective of observing the dynamics of floating wind turbines and its support structure, under dynamic wave and wind loading. In [3], the coupling of a power system simulation software (PowerFactory) and Matlab/Simulink is presented as a tool for simulating the dynamic response of wind turbines under realistic power system conditions. This paper demonstrates the cou-

pling of HAWC2 and Matlab/Simulink with simulations of a semi-variable speed wind turbine. This is a first stage towards an integrated design/analysis simulation platform that will be able to represent the dynamics of wind, rotor, structure, drive train, generator, control, and power system.

The following section presents a description of the HAWC2-Matlab coupling. Section 3 describes the wind turbine model. Simulations of a semi-variable speed wind turbine under a step in wind speed, and under a grid voltage fault are presented in Section 4.

2 HAWC2-Matlab coupling

This section describes the technical aspects of coupling HAWC2 and Matlab/Simulink. HAWC2 (Horizontal Axis Wind Turbine Code) is software developed at Risø DTU, dedicated to aeroelastic and structural simulations of wind turbines in the time domain. HAWC2 can model aeroelastic phenomena such as: tower shadow, dynamic inflow, dynamic stall, shear effects, and induction due to large deflections [4]. HAWC2's code is mainly written in FORTRAN, it can interface with other code via DLLs. Matlab is a well known high level computer language, and Simulink is a Matlab based software, oriented to control and dynamic system simulation. Matlab/Simulink is quite flexible, it has built-in features that support interfaces

to external code and applications [5], hardware and instruments [6].

The interface between HAWC2 and Matlab/Simulink was implemented keeping as a paradigm to keep both software as standalone applications. This is done through shared libraries (DLLs), and inter-process communication (TCP/IP), allowing both applications to run as separate processes that can exchange data dynamically, while Matlab manages the overall simulation. This method exploits Matlab's support for TCP/IP communications and for handling Java objects. The sketch in Fig.1 describes how the coupling works. A build-in routine [7], that sets up a Java object with a local TCP/IP socket (i.e. client), is run from Matlab while HAWC2 calls a DLL that listens to this socket every time step (i.e. server). In this way the HAWC2 simulation runs as an independent process, but every time step sends and receives data through the TCP/IP socket. Therefore, this interface of HAWC2 and Matlab can be run in different computers, allowing some degree of parallelization.

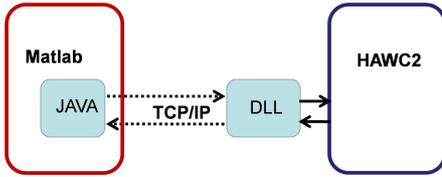


Figure 1: HAWC2 - Matlab TCP/IP interface.

Furthermore, the DLL-TCP/IP HAWC2 interface can interact with other applications than Matlab. For example with computational fluid dynamics (CFD) codes. This would allow running aeroelastic simulations coupled to CFD models instead of the classical engineering models. Such simulations could be run in super computers or clusters.

3 Wind turbine model

The system implemented for this work is a 2 MW, semi-variable speed, pitch regulated wind turbine with doubly-fed asynchronous generator (DFIG). It is connected directly to a grid that is assumed to be infinitely stiff.

The objective is to couple the constrained equations of motion solved by HAWC2 (Eq.1-2), and the state state-space model of the generator and control Eq.3-4. The variables use for coupling the system are: shaft rotational speed ω_g , blade angle θ_p , and generator torque T_g .

$$\mathbf{M}\dot{\mathbf{x}} = \mathbf{F} + \mathbf{F}_c \quad (1)$$

$$\mathbf{Q}(\mathbf{x}) = 0 \quad (2)$$

$$\dot{\boldsymbol{\lambda}} = \mathbf{A}\boldsymbol{\lambda} + \mathbf{B}\mathbf{u} \quad (3)$$

$$\mathbf{Y} = \mathbf{C}\boldsymbol{\lambda} \quad (4)$$

In Fig.2 the system is described by blocks that represent different functional subsystems of the wind turbine. The blocks left/down of the dotted line are simulated with HAWC2 and those up/right are simulated in Matlab/Simulink. The variables exchange between the two software are the rotational speed ω_g , which is an state in Eq.1, is sent from HAWC2 to Matlab. ω_g is then used in matrix \mathbf{A} , \mathbf{B} and \mathbf{C} of Eq.3-4. The generator torque T_g that is an output, and blade angle ω_p that is a control input, are sent from Matlab to HAWC2. In HAWC2, T_g is an external force \mathbf{F} and θ_p is a position constraint that goes into Eq.2.

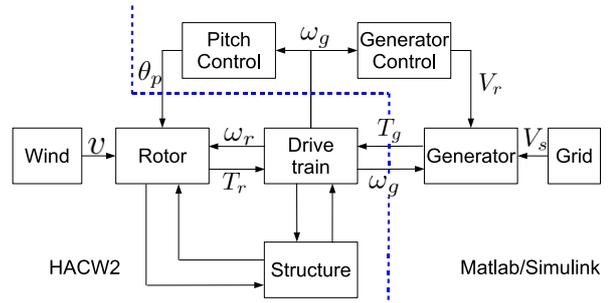


Figure 2: Diagram of wind turbine model for integrated simulations.

3.1 Wind turbine

The aerodynamics, and the mechanical and structural subsystems of the wind turbine are simulated in HAWC2. In Fig.2, they are represented by blocks named after the function of the different subsystems: wind, rotor, structure, and drive train. In terms of the input files used to setup the HAWC2 model, the blocks corresponding to the structural input are

- structure (flexible tower),
- drive train (flexible low speed shaft with generator inertia), and
- rotor (3 identical flexible blades with the hub inertia attached to the root),

each of this bodies has a number of elements and nodes, each node has 6 degrees of freedom.

3.2 Asynchronous generator

There is an extensive amount of literature regarding models of asynchronous and its control. The dynamic model implemented for this work is formulated in state-space in a dq -frame [8; 9; 10]. The model

assumes symmetry (uniformly distributed windings, and uniform air gap), neglects saturation, and neglects skin effect. The procedure for developing such model resumes to expressing the voltage equations in a manner that makes it easier to solve them for the selected state variables. Namely, the voltage equations are written in dq -frame, which eliminates the time varying inductances, making the solution of the dynamic equations less computationally expensive. Choosing fluxes λ as state variables, the voltage equations can be written as Eq.5.

$$\begin{aligned} \mathbf{v}_{dqs} &= \mathbf{r}_s \mathbf{i}_{dqs} + \dot{\lambda}_{dqs} \\ \mathbf{v}'_{dqr} &= \mathbf{r}'_r \mathbf{i}'_{dqr} + \dot{\lambda}'_{dqr} \end{aligned} \quad (5)$$

The dynamics of the stator are much faster than those of the rotor, making the system stiff. At this stage it is considered unnecessary to include such wide range of dynamics, the stator flux transients are therefore neglected. Rotor fluxes λ'_{dqr} are kept as state variables, and stator fluxes λ_{dqs} are solved with algebraic equations.

3.3 Control

The system has generator control, and blade angle control. The generator control regulates active and reactive power, and torque. It consists of two loops of cascade anti-windup proportional integral control (PI). Fig.3 shows the loop of the active power control, the reference power P_{ref} comes from a power vs. rotational speed table corresponding to operational points of the wind turbine in steady-state. P is the sum of stator and rotor measured power. The output of the first PI block is a current reference for the second PI block, where I_{qr} is the measured rotor current in the q -axis. The output of this loop is the rotor voltage in the q -axis V_{qr} .

Similarly, a cascade loop for the control of reactive power, takes the stator measured reactive power Q_s , as a reference for the first PI block. The output of this first block is then again a current reference. The rotor d -axis current I_{dr} , is the measured variable for the second block, the controlled output is the rotor voltage in d -axis.

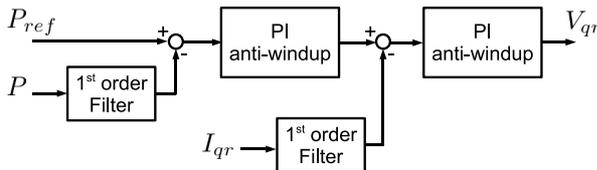


Figure 3: Block diagram of active power control loop.

A conceptual diagram of the blade angle control is shown in Fig.4. The blade angle control consist of

a PI anti-windup with gain schedule, and a first order system representing the blade angle servo. The reference speed ω_{ref} is taken from a table of wind speed vs. rotational speed. ω_{mea} is the measured rotational speed of the shaft.

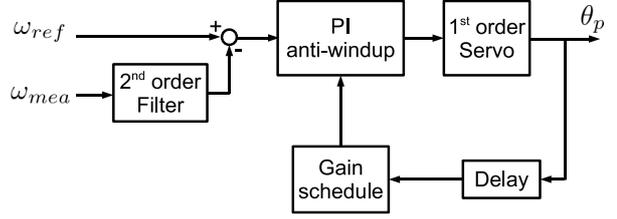


Figure 4: Block diagram of blade angle control.

4 Integrated Simulation

This section presents simulations of the system previously described, with the objective of qualitatively demonstrate the HAWC2-Matlab interface. Two simple cases are presented, a step change in the wind speed, and a symmetrical fault in the generator stator terminals.

Regarding the execution of the simulation, once the HAWC2 and the Matlab/Simulink systems are setup, the simulation can be started and managed from Matlab. The control and generator models in Simulink are solved using a implicit variable step solver, while HAWC2 solves its system with a fixed step implicit method.

4.1 Wind speed step change

The simulation of a step change in the wind speed is done from the rated wind speed, the step change is 1 m/s. Fig.5 shows the expected response of the blade angle control, increasing the blade angle to reduce the aerodynamic torque. The corresponding response of the shaft speed, and the shaft torsion are shown in Fig.6.

4.2 Grid fault

The following figures show the structural response of the wind turbine when a fault occurs in the stator terminals of the generator, without a fault-ride through control. The fault simulated is an ideal symmetrical 3-phase fault, where the stator voltage goes to zero for 100 ms. After 100 ms, the stator voltage is immediately restored. Fig.7 shows the generator torque and the blade angle.

Blade root bending moments during the fault are shown in Fig.9. Moments in the base of the tower are

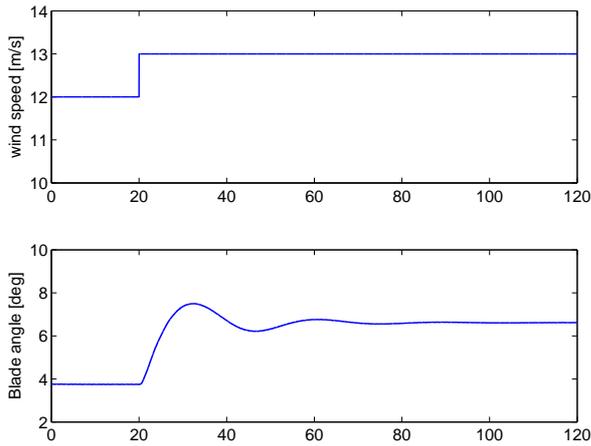


Figure 5: Wind speed step change, and blade angle response.

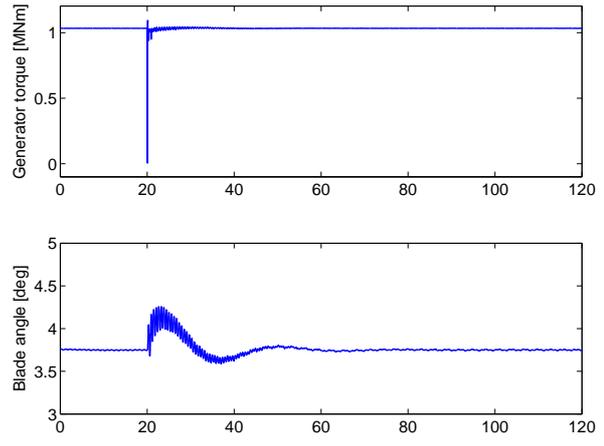


Figure 7: Wind speed and blade angle during grid fault.

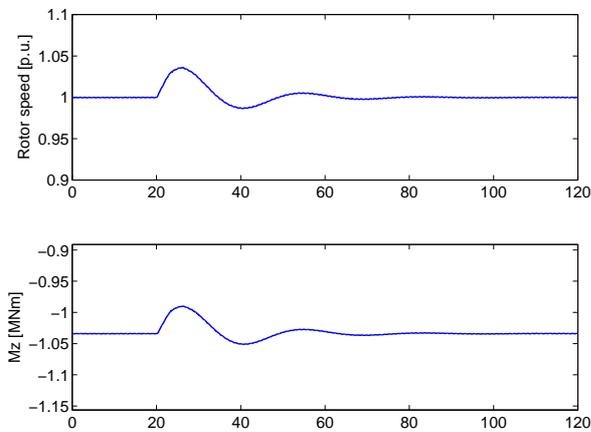


Figure 6: Response of shaft speed, and shaft torsion to step change in wind speed.

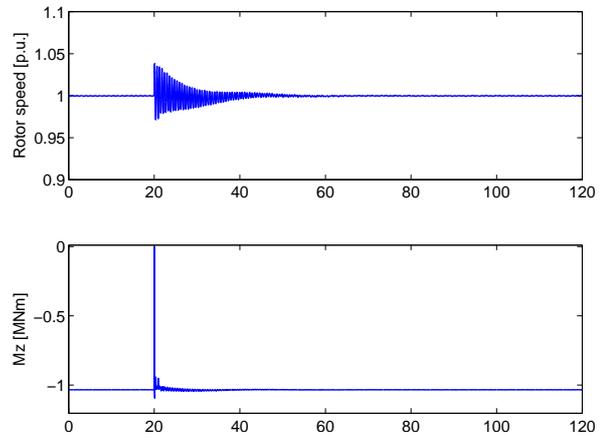


Figure 8: Rotor speed, and torsion in the shaft during grid fault.

presented in Fig.10, observe that the tower side to side moment M_y , is significantly affected and it continues to oscillate for a long time.

Finally, Fig.11 shows the tower top moments. As in the case of the tower base moments, the side to side moment is the most affected. In this case, however the oscillations are quickly damped.

5 Conclusions

A HAWC2-Matlab interface based on shared libraries, and TCP/IP sockets was presented. This is not the most computationally efficient way of coupling HAWC2 with Matlab. However, it is functional, allows for HAWC2 to be kept as a standalone application, and it involved relatively few modifications of the HAWC2 core code. It is also flexible in the sense that it works as any other DLL in HAWC2.

Future considerations for interfacing with HAWC2

may include shared memory interfaces, and a HAWC2 version that can be executed as a shared library by other applications.

Simulations of a semi-variable speed wind turbine were presented with an integrated simulation approach that couples HAWC2 and Matlab/Simulink. Expected behavior was observed in the structural responses of shaft, blades, and tower. Further work, in this area can involve realistic load cases analysis for special operating conditions required by the power system, such as low voltage ride-through and frequency control.

Acknowledgments

The authors would like to acknowledge their colleague, Peter Bjørn Andersen from the Aeroelastic group in the Wind Energy Division of Risø DTU, who started the development of the HAWC2-Matlab inter-

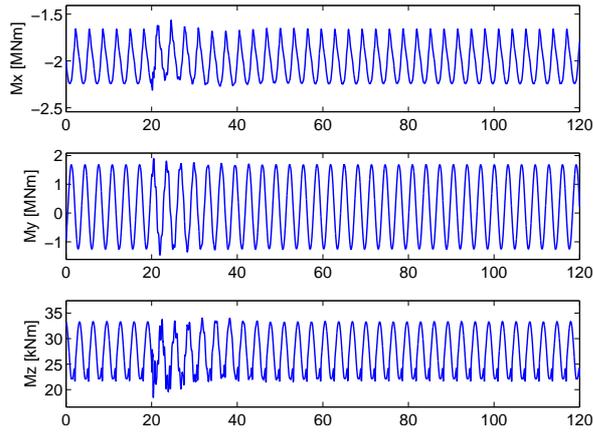


Figure 9: Blade root bending moments during grid fault.

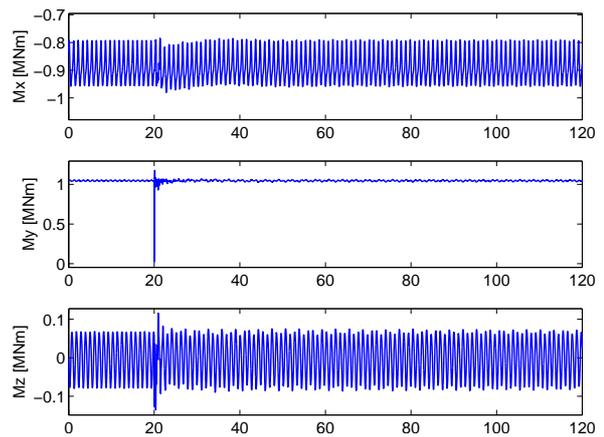


Figure 11: Tower top bending moments response to wind step.

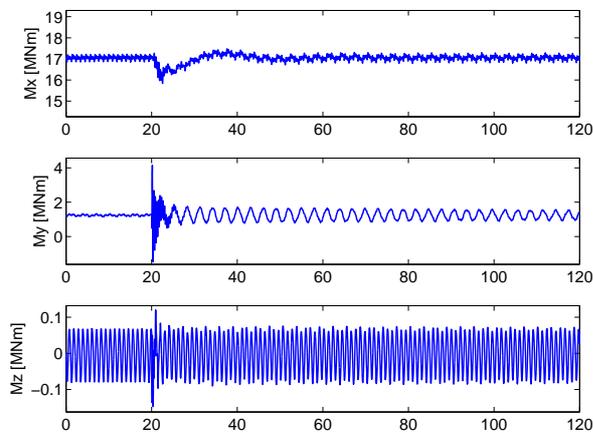


Figure 10: Tower base bending moments during fault.

face. His implementation was then continued by Lars C. Henriksen and turned into the interface used for this work.

References

- [1] IEA, "IEA WIND ENERGY annual report 2008," tech. rep., International Energy Agency, 2009.
- [2] D. Kaufer, N. Cosack, C. Böker, M. Seidel, and M. Kühn, "Integrated Analysis of the Dynamics of Offshore Wind Turbines with Arbitrary Support Structures," in *European Wind Energy Conference*, 2009.
- [3] A. D. Hansen, N. A. Cutululis, F. Iov, and B. Barahona, "Integrated electrical design of DFIG wind turbines for power system studies," in *Nordic Wind Power Conference*, 2009.
- [4] T. J. Larsen, "How 2 HAWC2, the user's manual," tech. rep., Risø DTU, January 2009.

[5] The MathWorks, Inc., *MATLAB External Interfaces*, online ed., October 2008.

[6] The MathWorks, Inc., *Instrument Control Toolbox*, online ed., July 2005.

[7] R. Thomson, *TCP/IP Socket Communications in MATLAB*. TheMathWorks, online ed., August 2008.

[8] C.-M. Ong, *Dynamic Simulation of Electric Machinery using Matlab/Simulink*. Prentice Hall PRT, 1998.

[9] P. C. Krause, O. Wasynczuk, and S. D. Sudhoff, *Analysis of Electric Machinery and Drive Systems*. IEEE Press Series on Power Engineering, WileyBlackwell, 2 ed., 2002.

[10] F. Iov, A. D. Hansen, P. Sørensen, and F. Blaabjerg, "Wind Turbine Blockset in Matlab/Simulink. General Overview and Description of the Model," tech. rep., Aalborg University, 2004.