Real-Time Aeroservoelastic Analysis of Wind-Turbines by Free Multibody Software

Luca Cavagna, Alessandro Fumagalli, Pierangelo Masarati, Marco Morandini, and Paolo Mantegazza

Abstract This work illustrates the feasibility of the implementation of innovative, efficient and low-cost solutions for fast-prototyping and customization of controlled mechanical and aeroservoelastic systems. A controlled constant-speed wind-turbine is considered as an example of the proposed methodologies, where the physical aeromechanics problem is controlled by a controller process scheduled for execution in real-time on a PC-class computer. The physical problem is simulated by a general-purpose multibody process that is scheduled in real-time as well. The processes communicate using real-time inter-process communication primitives. All the involved tools are based on free software.

1 Introduction

Wind-turbines represent an important means to extract energy from the environment in a 'green' manner. The concept of extracting energy from the wind dates back thousands of years, including not only power generation (e.g. mills, water pumps) but also direct locomotion (e.g. sailing). Modern wind-energy technology relies on efficient aerodynamic design and durable mechanical systems.

Nonetheless, efficient and reliable energy harvesting from winds poses several significant challenges, including:

Luca Cavagna
Politecnico di Milano, Dipartimento di Ingegneria Aerospaziale, Italy, e-mail: luca.cavagna@polimi.it; now at FOI (Swedish Defence Research Agency), Sweden

Alessandro Fumagalli
Politecnico di Milano, Dipartimento di Ingegneria Aerospaziale, Italy, e-mail: alessandro.fumagalli@polimi.it; now at Altran, Italy

Pierangelo Masarati · Marco Morandini · Paolo Mantegazza
Politecnico di Milano, Dipartimento di Ingegneria Aerospaziale, Italy, e-mail: pierangelo.masarati@polimi.it, marco.morandini@polimi.it, paolo.mantegazza@polimi.it

- optimal harvesting with rather irregular and erratic wind conditions;
- tolerance to wear and fatigue with minimal maintenance;
- controllability and survivability during exceptional weather and operating conditions;
- efficient and fault-tolerant integration with power grids.

It is anticipated that key to many of the issues mentioned above is control. A recent, extensive review of the state of the art in wind-turbines aerodynamics and aeroelasticity is presented in [1]. Smart rotor control research for wind-turbines is presented in [2].

The goal of this work is to illustrate the rapid feasibility of the implementation of innovative, efficient and low-cost solutions for fast-prototyping and customization of controlled mechanical and aeroservoelastic systems. The specific problem of designing control systems for wind turbines is not addressed. The work rather focuses on providing analysis tools that can be used for this purpose.

2 Approach

The free Real-Time Operating System (RTOS) Real-Time Application Interface (RTAI, [3]) and the free general-purpose multibody software MBDyn [4], both originating from research at the Dipartimento di Ingegneria Aerospaziale of Politecnico di Milano, Italy, are at the core of the present work. The use of free software that runs on low-cost hardware gives any organization, significantly the academia and Small-Medium Enterprises (SMEs), access to powerful and versatile analysis and simulation capabilities.

A controlled constant-speed wind-turbine [5] is considered as one of the possible applications of the developed methodologies. Fig. 1 shows a sketch of the controlled model under analysis [6]. It consists in a multibody model of the Controls Advanced Research Turbine (CART), a research wind-turbine in use at the National Renewable Energy Laboratory (NREL, [7]) for experimental purposes.

The box containing a picture of the aerogenerator on the left represents the multibody model of the wind-turbine, which accounts for its kinematics, structural dynamics and aerodynamics. The multibody model outputs the angular speed of the shaft, which is fed to the control system. The control modeling environment takes care of:

- generating the gusty airstream input for the multibody analysis;
- determining the generator's torque to be applied to the shaft of the wind-turbine;
- computing the blade pitch input to be used by the multibody model in order to control the behavior of the wind-turbine.

The latter item represents the control task. Given the relative simplicity of the controller considered in this work, the whole problem could have been modeled monolithically within the general-purpose multibody simulation environment. Nonetheless, the control-related part has been intentionally modeled in a separate general-

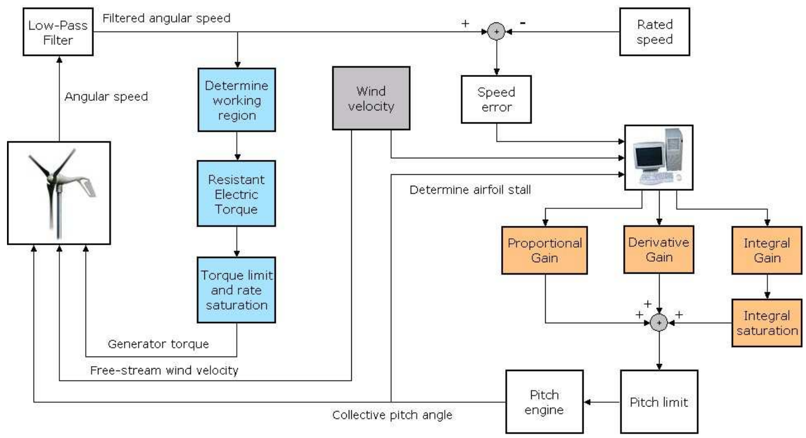


Fig. 1 Sketch of the controlled model.

purpose, graphically driven mathematical modeling environment, for the sake of generality. This kind of graphical environment represents the natural modeling environment for control systems, and is offered by many popular software packages, e.g. The Mathworks' Matlab/Simulink [8], INRIA's Scilab/Scicos [9, 10], Labview [11] and National Instrument's MATRIXx [12].

The multibody analysis, instead, represents an effective means to provide the virtual simulation of the real process that needs to be controlled. For this reason, it only contains the bare physical process, in order to allow to test the real controller that would be used in a real-world application. The two processes typically communicate by means of real-time capable network primitives, or by inter-process communication when running on the same computer, on separate CPUs in case of SMP architectures.

There exists a number of software that can be proficiently used to analyze the aeroservoelastic behavior of wind-turbines. Some are dedicated to this task, while others are general-purpose. A recent survey of some of them is presented in [13]. That reference compares software based on accuracy with respect to benchmark problems. MBDyn has been coupled to NREL's AeroDyn library in order to exploit the availability of a well-proven wind-turbine aerodynamic code [14]. However, there are other factors that may come into play, significantly those related to software accessibility, to the capability of modeling problems with an arbitrary level of detail, and to fulfill control design requirements.

The problem of accessibility is addressed by using 'free software'. Problems can be analyzed with an arbitrary level of detail when general-purpose software is used. Control design requirements, and significantly the capability to perform hardware-in-the-loop virtual testing, are met by enabling tight real-time scheduling and execution of the simulation and control software. The proposed virtual testing environment meets all the requirements illustrated above.

The use of general-purpose multibody software typically results in solving larger problems, especially when the formulation is based on the redundant coordinate approach. This may represent a challenge for real-time simulation; for this reason, real-time simulations are ofter approached reducing the problem to a minimal set of coordinates. The redundant coordinate approach, however, usually results in very sparse problems. When sparsity is efficiently handled by specialized linear algebra solvers, as the one proposed in [15], very good performances can be obtained in terms of computational time. A detailed comparison of the effects of different linear solvers on the overall efficiency of real-time multibody simulation is presented in [16]. Complex mechanical systems, including robots and rotorcraft wind-tunnel models, can be simulated in real-time with the desired accuracy using general-purpose multibody software, with an acceptable trade-off between model detail and real-time implementation [17, 18, 19]. This is the case of MBDyn [20, 21], the free multibody software used and adapted for the purpose of this research.

3 Wind-Turbine Description

The CART wind-turbine, shown in Fig. 2, is located at the National Wind Technology Center (NWTC) of NREL in Colorado. It is used as state of the art test-bench for controls research in wind-engineering [22, 23, 24, 25]. The main focus is currently on testing control strategies to improve the performances and the handling of wind-turbines subjected to exceptional operational conditions. It is an upwind machine with a nacelle tilt of 3.8 deg and two teetering blades with zero precone.

The rotor diameter and the hub height are respectively 43.3 m and 36.6 m. Power energy is rated at 600 kW and generator speed through a gearbox with a ratio of 43.165 is rated at 1800 rpm. The rotor is thus rated at an angular speed $\Omega_{\text{rated}} = 41.7$ rpm.

This model has been selected because it is described in detail in the above mentioned publicly accessible documents. Moreover, since it is characterized by a two-blade, teeter rotor, it requires less computational effort than more modern, three-blade turbines. As illustrated in the following, this choice was conservative, as the proposed analysis leaves margin for further increase in model complexity without violating the real-time requirements.

4 Baseline Controller

The baseline controller is composed by independent electric torque and collective pitch algorithms. Both controllers use the rotor angular speed measurement as sole input.

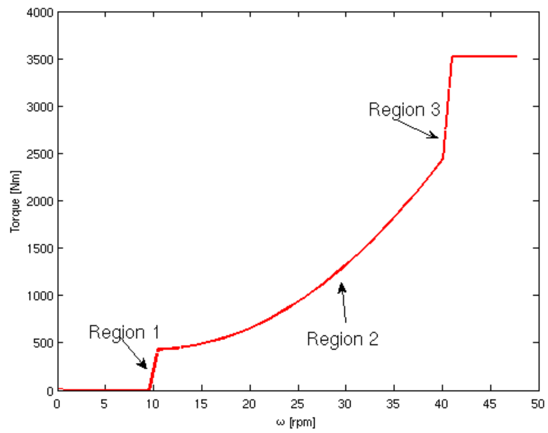
Fig. 2 The Controls Advanced Research Turbine (CART).

The task of the control-system is to maximize power capture below, and regulate a constant speed above, the rated operating point. Currently, no effort is undertaken to regulate the high speed generator shaft brake nor the nacelle yaw (which, in the real wind turbine, is limited to only ± 0.5 deg and simply used for tracking relatively small wind changes).

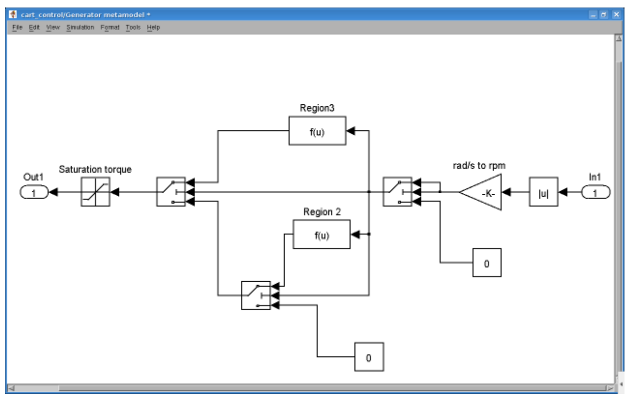
Generator commands are calculated by means of a piece-wise function. Below the cut-in speed of 10 rpm (Region 1), no electric torque is generated, to let the wind accelerate the rotor at maximum angular acceleration. The quadratic region (Region 2) is designed to keep the tip-speed ratio at the optimal value for maximum power. Above 99% of the rated rotor speed Ω_{rated}, a constant torque of 3524 Nm is required. Between 98% and 99% of Ω_{rated}, the transition is linear, equivalent to a slip of 5% (Region 3) [26, 27, 28]. Fig. 3(a) shows the piece-wise working function for the electric generator (top), and its block-diagram model (bottom).

The full-span collective blade pitch angle commands are computed by means of a PID controller on the error of the rated angular speed, with saturation on the integral term to limit wind-up. Special care is taken to avoid working in post-stall regions during the initial acceleration phase. This allows to use the same controller for both rotor start-up and speed control at the rated speed.

Fig. 4 shows the controlled system in the typical Scicos/Simulink/MATRIXx environment. The box labeled ‘from MBDyn’ represents the output of the multibody model that is input in the control system: the basic control system considered in this work requires only a measure of the angular velocity of the rotor. The box labeled ‘to MBDyn’ represents the inputs to the multibody model: the free-stream wind velocity, the torque absorbed by the electric generator and the desired blade pitch.



(a) Working function.



(b) Block-diagram.

Fig. 3 Electric generator piecewise working function and block-diagram model.

The controlled CART model can be run either in batch or real-time mode. In the latter case, the real-time scheduling can be delegated to the operating system using a standard POSIX interface, or tightly enforced using RTAI. When executing in batch mode, a generic interface between MBDyn and the external control software, based on standard UNIX inter-process communication primitives (local and TCP/IP sockets), is used. When executing in hard real-time mode, the controller code is automatically generated by any of Scicos, Simulink or MATRIXx from the very same model, and run by the RTAI operating system. In this case, MBDyn is scheduled in real-time by RTAI as well, to emulate the real wind-turbine.

In real time mode, the RTAILab graphical user interface [29] can be used to monitor the controller, either locally or remotely, and to tune gains and other system parameters on the fly, as shown later in an example.

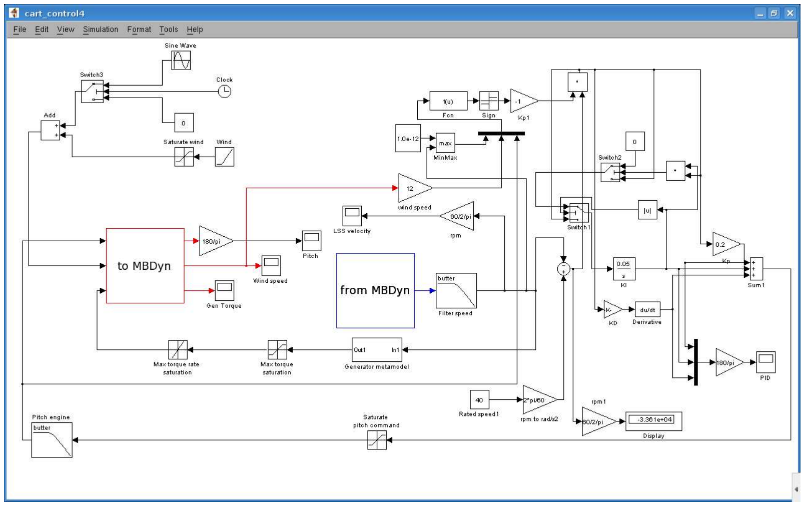


Fig. 4 Controlled system in the typical Scicos/Simulink/MATRIXx environment.

5 Multibody Model

The multibody approach is definitely suited for the analysis of complex multidisciplinary systems where exact mechanism kinematics, nonlinear structural dynamics, and arbitrary control-related components need to be simultaneously analyzed [30, 31]. Thanks to its versatility and availability, the multibody formalism proposed in this work is being used in many fields related to aeroservoelasticity.

The analysis is based on an original formulation, implemented in the free general-purpose software MBDyn [4]. It performs the direct time integration of Initial Value Problems (IVP) written as a system of first-order Differential-Algebraic Equations (DAE), using implicit (nearly) L-stable integration algorithms [32].

5.1 Unconstrained Dynamics

The equations of motion of each unconstrained body are written in first order form using the Newton-Euler approach. The definitions of momentum, β_i, and momenta moment about the node, $\gamma_{\mathbf{x}_i}$, for the i-th node are

$$m_i \dot{\mathbf{x}}_i + \omega_i \times \mathbf{s}_{i\mathbf{x}_i} = \beta \tag{1a}$$

$$\mathbf{s}_{i\mathbf{x}_i} \times \dot{\mathbf{x}}_i + \mathbf{J}_{i\mathbf{x}_i} \omega_i = \gamma_{i\mathbf{x}_i} \tag{1b}$$

where m_i is the mass of the body connected to the i-th node, $\mathbf{x}_i$ is the location of the node, ω_i is the angular velocity of the node, $\mathbf{s}_{i\mathbf{x}_i}$ is the static moment of the body referred to the node's location, and $\mathbf{J}_{i\mathbf{x}_i}$ is the inertia tensor of the body referred to the node's location. The equilibrium of each node yields

$$\dot{\beta}_i = \sum \mathbf{f}_i \tag{2a}$$
$$\dot{\gamma}_{i\mathbf{x}_i} + \dot{\mathbf{x}}_i \times \beta_i = \sum \mathbf{m}_{i\mathbf{x}_i} \tag{2b}$$

where all external forces, $\mathbf{f}_i$, and moments, $\mathbf{m}_{i\mathbf{x}_i}$, acting on the node are considered. The external forces and moments can arbitrarily depend on the motion of all nodes the i-th one is connected to. The equations of motion of all the unconstrained nodes can be summarized as

$$\mathbf{M}\dot{\mathbf{q}} = \mathbf{p} \tag{3a}$$
$$\dot{\mathbf{p}} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{p}, t) \tag{3b}$$

where $\mathbf{q} \in \mathbb{R}^n$ summarizes the kinematic variables of the nodes (n corresponds to 6 times the number of bodies, n_{b}), while $\mathbf{p} \in \mathbb{R}^n$ summarizes the momentum and momenta moments. The function $\mathbf{f} : \mathbb{R}^{3n+1} \mapsto \mathbb{R}^n$ represents the generic configuration-dependent forces acting on the nodes. It includes the contributions related to structural deformability.

5.2 Constrained Dynamics

The constrained system dynamics are modeled by explicitly adding kinematic constraints between the nodes in form of algebraic equations, using Lagrange's multipliers formalism. The addition of m_{h} holonomic and m_{nh} non-holonomic constraints, respectively expressed by $\phi(\mathbf{q}, t) = \mathbf{0} : \mathbb{R}^{n+1} \mapsto \mathbb{R}^{m_{\mathrm{h}}}$ and $\psi(\dot{\mathbf{q}}, \mathbf{q}, t) = \mathbf{0} : \mathbb{R}^{2n+1} \mapsto \mathbb{R}^{m_{\mathrm{nh}}}$, result in

$$\mathbf{M}\dot{\mathbf{q}} = \mathbf{p} \tag{4a}$$
$$\dot{\mathbf{p}} + \phi^T_{/\mathbf{q}}\lambda + \psi^T_{/\dot{\mathbf{q}}}\mu = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{p}, t) \tag{4b}$$
$$\phi(\mathbf{q}, t) = 0 \tag{4c}$$
$$\psi(\dot{\mathbf{q}}, \mathbf{q}, t) = 0 \tag{4d}$$

where $\lambda \in \mathbb{R}^{m_{\mathrm{h}}}$ and $\mu \in \mathbb{R}^{m_{\mathrm{nh}}}$ respectively are the multipliers related to the holonomic and non-holonomic constraints.

5.3 Structural Flexibility

The modeling of structural flexibility is fundamental for accurate aeroelastic analysis. However, accuracy may require a considerable number of degrees of freedom.

As a consequence, accuracy may need to be traded for efficiency, especially when real-time simulation is considered.

Conventional horizontal axis wind turbines are characterized by a slender tower and very slender blades. The slenderness of current large size turbines will probably increase further as the size grows from $2 \div 5$MW on.

Historically, structural flexibility has been considered in multibody dynamics using lumped components first. Eventually, the need to bring the level of detail of finite elements led to combining the arbitrary reference rigid body motion peculiar of multibody dynamics with small perturbed deformation given by linear finite elements into the so-called floating frame approach [33]. In this case, a Ritz-like linear combination of deformation shapes is used to express a deformation with respect to a reference frame that undergoes arbitrary motion. This approach, in the case of wind turbines, suffers from the fact that an accurate basis consisting in normal vibration modes may require a significant number of degrees of freedom, since the normal modes are considerably influenced by the centrifugal stiffening [34]. This may not be an issue for systems rotating at constant angular velocity, but in general wind turbines can operate at an arbitrary velocity, and transient analysis capability is essential.

Accurate modeling of structural components can be achieved using finite elements directly in the multibody model [35]. The behavior of slender structural components can be efficiently described by the beam model. In many cases the beam model is fairly accurate and at the same time synthetic; thus, it leads to efficient models, allowing to meet real-time simulation requirements. In this work, an original, geometrically exact, composite-oriented beam formulation based on a finite-volume approach is used [36]. The beam model in the multibody analysis takes care of the one-dimensional flexibility of slender structural components. In order to give accurate results, it requires a correct and accurate characterization of the cross-section inertial and structural properties. In the present work, this pre-processing step is based on the section characterization procedure first proposed by Giavotto *et al.* [37], that allows to characterize the 6×6 stiffness matrix of a generalized Timoshenko beam. A detailed review of different beam section characterization procedures is presented in [38]. Other formulations, including the one proposed in [39], are compared to standard wind turbine blade characterization approaches in [40].

5.4 Numerical Integration

The implicit DAE problem of Eqs. (4) can be written in the generic form

$$\mathbf{g}(\dot{\mathbf{y}}, \mathbf{y}, t) = \mathbf{0}, \tag{5}$$

where $\mathbf{y} = \{\mathbf{q}; \mathbf{p}; \lambda; \mu\}$ summarizes all the variables of Eqs. (4). Its solution at the generic time step t_k, using a generic implicit multistep integration scheme, requires to solve Eq. (5) for $\dot{\mathbf{y}}_k$, with

$$\mathbf{y}_k = \sum_{r=1,n} a_r \mathbf{y}_{k-r} + h \sum_{s=0,n} b_s \dot{\mathbf{y}}_{k-s}. \tag{6}$$

The coefficients a_r and b_s characterize the numerical integration method; $b_0 \neq 0$ for implicit schemes. Eq. (5) is solved using a Newton-Raphson scheme, namely

$$\mathbf{g}_{/\dot{\mathbf{y}}} \delta \dot{\mathbf{y}}_k + \mathbf{g}_{/\mathbf{y}} \delta \mathbf{y}_k = -\mathbf{g}. \tag{7}$$

According to Eq. (6), $\delta \mathbf{y}_k = h b_0 \delta \dot{\mathbf{y}}_k$; as a consequence, Eq. (7) yields

$$\left(\mathbf{g}_{/\dot{\mathbf{y}}} + h b_0 \mathbf{g}_{/\mathbf{y}} \right) \delta \dot{\mathbf{y}}_k = -\mathbf{g}. \tag{8}$$

The linear problem of Eq. (8) needs to be solved iteratively to convergence. The DAE nature of the problem implies that either of matrices $\mathbf{g}_{/\dot{\mathbf{y}}}$, $\mathbf{g}_{/\mathbf{y}}$ can be structurally singular, or both. However, when the problem is well posed, the matrix pencil $(\mathbf{g}_{/\dot{\mathbf{y}}} + \lambda \mathbf{g}_{/\mathbf{y}})$ is not structurally singular; thus, Eq. (8) can be solved [41].

MBDyn is mainly used by its developers to model the aeroservoelasticity of rotary wing aircraft (e.g. [42]). Analysis of wind-turbine systems is carried out by some of the independent users [43, 14].

5.5 CART Wind Turbine Multibody Model

The multibody model considered in this work consists in:

- a deformable tower, made of 5 three-node finite-volume beam elements, clamped to the ground at the lower extremity;
- a rigid nacelle, connected to the tower by a yaw hinge, with built-in pitch; the yaw degree of freedom in the analysis is restrained by a very stiff spring, since no yaw control is considered;
- a rigid low-speed shaft, connected to the nacelle by ideal bearings; the rotational inertia includes that of the high-speed shaft, accounting for the low- to high-speed shaft gear ratio;
- an ideal generator, consisting in an internal torque applied between the nacelle and the low-speed shaft, whose value is computed by the control task;
- a rigid body that models the teetering hub, connected to the low-speed shaft by an ideal teeter joint;
- a pair of deformable blades, modeled with 5 three-node finite-volume beam elements each, including blade element aerodynamics coupled to an induced flow model; each blade is hinged to the teetering body by means of a revolute hinge that allows to impose the pitch angle. Although the pitch of each blade can be controlled independently, in this study the same value is applied for simplicity.

Table 1 summarizes the number of nodes, elements and degrees of freedom of each component of the model. It results in a total of 465 equations: 432 related to the dynamics of the nodes, and the remaining 33 related to holonomic constraints.

The typical sparsity pattern of the matrix pencil of Eq. (8) is shown in Figure 5. The number of non-zero coefficients is 6351, which implies a fill-in less than 3%. The matrix is intrinsically non-symmetric.

Table 1 Summary of CART model.

Component	Nodes	Joints	Bodies	Beams	Aero	Forces	DoFs
Tower	11	1	10	5			138
Nacelle	1	2	1				17
Shaft	1	1	1				17
Generator						1	
Teeter	1	1	1				17
Blades, $2\times$	11	1	11	5	5		138
Total	36	7	35	15	10	1	465

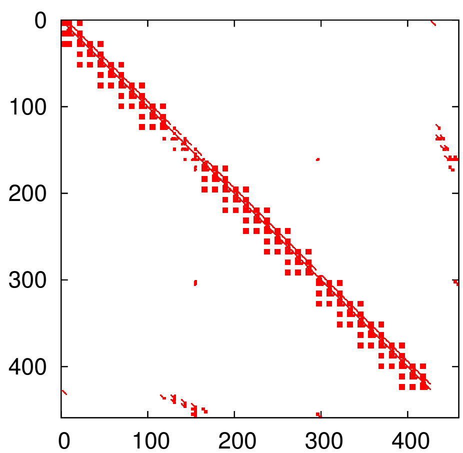


Fig. 5 Non-zero coefficients of the CART model matrix; fill-in is about 3%.

The blade pitch is controlled by simultaneously rotating the blades at the root node by an amount that is determined by the controller task. An angular velocity sensor measures the low-speed shaft velocity and feeds it into the control task.

The wind-turbine pictures in Fig. 6 are generated by an enhanced version of the free visualization software EasyAnim [44] (the modified version is available in [4]).

The structural model of the tower and of the blades is likely too refined, considering the very low rotational velocity and the bandwidth of interest. Figure 7(a) illustrates the convergence on the frequency of the first 10 modes of the tower plus nacelle model as the discretization is refined from 1 to 10 three-node beam elements. Figure 7(b) refers to the entire model at null angular velocity, with the blades

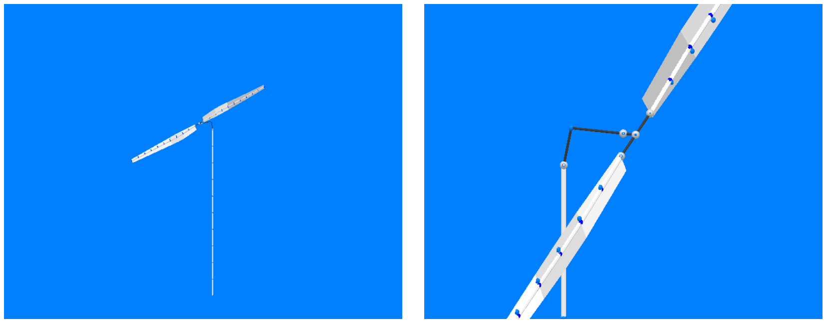
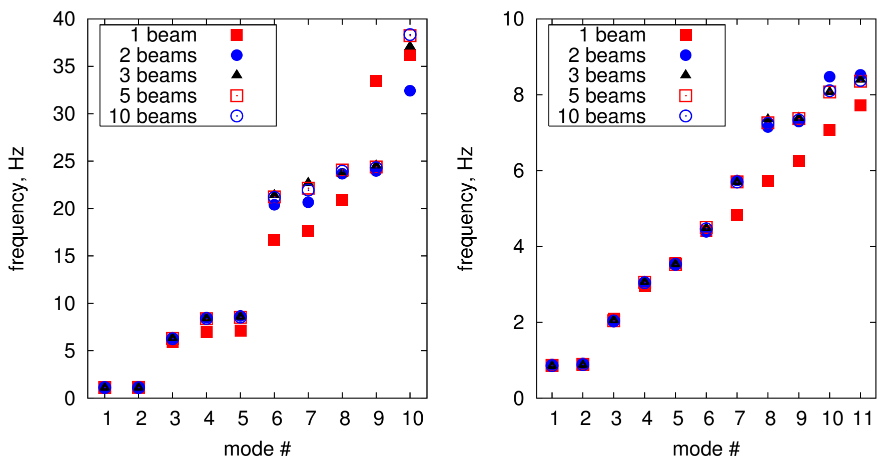


Fig. 6 Graphical representation of the CART model.



(a) Tower & nacelle (b) All, blades vertical

Fig. 7 CART model convergence on the first normal modes (non-rotating).

in the vertical position. Similar trends are shown by all subcomponents. Five beam elements for the tower and five for each blade have been used because, as shown in Figure 7, they yield a dynamically well converged model, while allowing to meet the real-time execution constraint.

6 Real-Time Simulation

The proposed multibody analysis runs in real-time thanks to RTAI support, built-in in MBDyn when running on Linux [17, 45, 46].

Popular graphical tools for computer-assisted control system design and fast prototyping, with automatic control code generation, like Scicos, Simulink, and MA-

TRIXx, have been extended to support the generation of the controller source code in the C programming language, using RTAI's primitives for real-time scheduling and inter-process communication.

As a result, two real-time processes, one simulating the physical system and the other performing the appropriate control task, are executed on computers running the RTAI real-time extension for Linux.

The RTOS takes care of scheduling both processes with the desired periodicity. The processes typically communicate a set of measurements from MBDyn to the controller, and a set of control inputs from the controller to MBDyn. Interprocess communication uses RTAI mailboxes, a primitive that transparently uses shared memory when both tasks are executed on the same machine, or the UDP real-time support, provided by the NetRPC extension, when tasks are distributed on different machines.

In more sophisticated applications, a real-time instance of MBDyn itself can be embedded in the controller. In those cases, it is used to determine the control inputs required by the controlled process, which in turn can be a real or a simulated process [47].

Two scheduling approaches can be followed. In one case, the processes synchronize with each other by using RTAI semaphores. One process, usually the controller, is scheduled periodically. As soon as it sends the control input to the simulator, the simulator is woken up and starts simulating the time step. This approach guarantees that the subsequent time step receives the expected control input. In the other case, the two processes are independently scheduled periodically. Each process reads inputs and writes outputs according to its schedule. There is no strict guarantee that each process receives exactly the expected input. However, the error can be at most one sample period, and thus is treated as a disturbance.

The simulation must behave in a quasi-deterministic manner or, in other words, each sample interval needs to be completed within a given number of operations. This is not guaranteed when iteratively solving a nonlinear problem. In order to obtain a quasi-deterministic behavior, RT-MBDyn solves the nonlinear problem within up to a fixed number of iterations, using a modified Newton-Raphson scheme that consists in assembling and factorizing the matrix only at the first iteration of each time step. Errors due to lack of convergence to the desired accuracy can be reasonably assumed to be small after few iterations, thanks to the superlinear convergence properties of the modified Newton-Raphson scheme, provided the prediction at each time step is close enough to the actual solution. These errors can be treated as disturbances by the control scheme. Sporadic overruns can be accepted as disturbances, provided subsequent steps can "catch up" with the controller.

Fig. 8 shows a fairly broad layout of the real-time simulation setup, where the simulation and the controller are located on different computers connected by a hard real-time network via NetRPC, while multiple supervising stations monitor the output of the controller and of the simulation using soft real-time connections, with the possibility of optionally modifying the controller's parameters.

Figure 9 shows the result in terms of rotor angular speed and pitch command of a simulation in correspondence of a growing wind speed rated at an average level

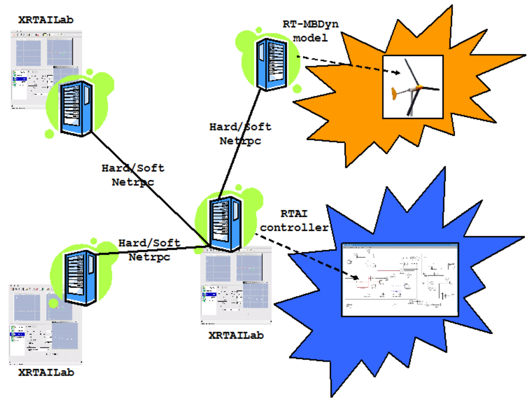
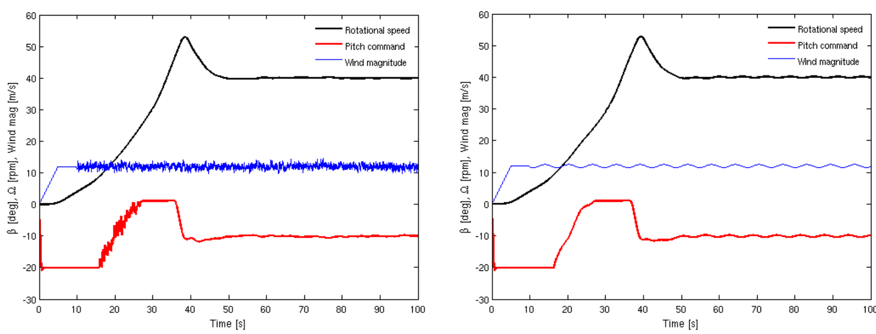


Fig. 8 Sketch of a generic distributed real-time simulation layout.

of 12 m/s. In detail, Fig. 9(a) refers to a random disturbance of 20% of the wind velocity magnitude (blue line), while Fig. 9(b) refers to a sinusoidal disturbance whose amplitude is 5% the wind velocity magnitude (blue line). In both cases, the resulting error on the final rated angular speed is less than 0.5%. The sample rate is 100 Hz. Fig. 10 shows the output of the controlled CART model within the RTAILab



(a) Random gust. (b) Sinusoidal gust.

Fig. 9 Rotor angular speed and pitch command for wind-up and gusty wind.

environment. The field labeled as ‘Gain’ in the top left portion of the control panel allows to change the controller’s parameters while the simulation is running in real-time. The control panel can be configured to allow access to any of the control parameters that are exported when the control system is designed.

The numerical simulations have been performed on a Dual Core AMD Opteron Processor 280 (1 GHz). In all cases the multibody model could be executed well within the required sample rate of 100 Hz. This leaves room for further model re-

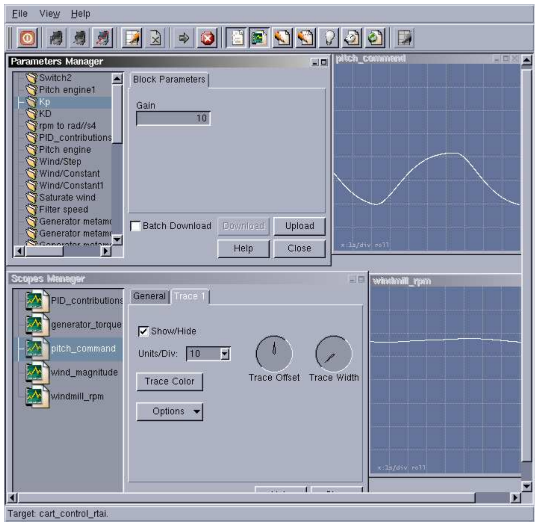


Fig. 10 Output of the controlled CART model within the RTAILab environment.

finement, e.g. an increase of the sampling rate, or the analysis of more complex turbines, e.g. three-bladed.

One of the distinguishing features of the proposed approach is that information related to distributed structural flexibility can be simulated and monitored in real-time. This paves the way to simulating in real-time the control of strains, stresses, and gust load reduction in general.

Conclusions

This work illustrates the implementation of what can essentially be considered a test bench to prove the feasibility of innovative, efficient and low-cost solutions for fast-prototyping and customization of controlled mechanical and aeroservoelastic systems. The proposed environment has been applied to the development and testing of a simple controller for wind-turbines. Further details can be added, both in the simulated physical process, to enhance system modeling with features that have not been considered so far in this work, and in the controller, to investigate more sophisticated control strategies.

Additional Material

The analysis was performed using RTAI 3.6.1, available for download at [3], and MBDyn 1.3.3, available for download at [4]. The wind-turbine models and the controller source code are available at [4], in the RT-MBDyn→wind turbine folder. Feedback using the mailing lists *rtai@rtai.org* and *mbdyn-users@mbdyn.org* is appreciated.

Acknowledgements This work has been partially funded by 'SI PARTE!', a R&D project of 'Regione Lombardia' that addressed the development of innovative solutions for Embedded Real-Time Applications.

References

1. Hansen MOL, Sørensen JN, Voutsinas S, Sørensen N, Madsen HAa (2006) State of the art in wind turbine aerodynamics and aeroelasticity. Progress in Aerospace Sciences 42(4):285–330
2. Barlas TK, van Kuik GAM (2009) Review of state of the art in smart rotor control research for wind turbines. Progress in Aerospace Sciences 46(1):1–27
3. Real-time application interface (RTAI). http://www.rtai.org/ (last accessed March 2009)
4. Multibody dynamics (MBDyn). http://www.aero.polimi.it/mbdyn/ (last accessed March 2009)
5. Garsch R, Twele J (2002) Wind Power Plants, Fundamentals, Design, Construction and Operation. James and James.
6. Jonkman J, Butterfield S, Musial W, Scott G (2009) Definition of a 5-MW Reference Wind Turbine for Offshore System Development. NREL/TP-500-38060
7. National renewable energy laboratory (NREL). http://www.nrel.gov/ (last accessed November 2009).
8. Matlab. http://www.mathworks.com/ (last accessed November 2009).
9. Scilab. http://www.scilab.org/ (last accessed November 2009).
10. Scicos. http://www.scicos.org/ (last accessed November 2009).
11. NI LabVIEW. http://www.ni.com/labview/ (last accessed November 2009).
12. NI MATRIXx. http://www.ni.com/matrixx/ (last accessed November 2009).
13. Buhl ML Jr, Manjock A (2006) A comparison of wind turbine aeroelastic codes used for certification. In 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada (USA), January 9–12
14. Meng F, Masarati P, van Tooren M (2009) Free/open source multibody and aerodynamic software for aeroservoelastic analysis of wind-turbines. In 47th AIAA Aerospace Sciences Meeting, Orlando, Florida (USA), January 5–8
15. Morandini M, Mantegazza P (2007) Using dense storage to solve small sparse linear systems. ACM Trans. Math. Softw. 33(1)
16. González M, González F, Dopico D, Luaces A (2008) On the effect of linear algebra implementations in real-time multibody system dynamics. Computational Mechanics, 41(4):607–615
17. Attolico M, Masarati P (2003) A multibody user-space hard real-time environment for the simulation of space robots. In Fifth Real-Time Linux Workshop, Valencia, Spain, November 9–11
18. Masarati P, Attolico M, Nixon MW, Mantegazza P (2004) Real-time multibody analysis of wind-tunnel rotorcraft models for virtual experiment purposes. In AHS 4th Decennial Specialists' Conference on Aeromechanics, Fisherman's Wharf, San Francisco, CA, January 21–23

19. Attolico M, Masarati P, Mantegazza P (2005) Trajectory optimization and real-time simulation for robotics applications. In Multibody Dynamics 2005, ECCOMAS Thematic Conference, Madrid, Spain, June 21–24
20. Masarati P, Morandini M, Quaranta G, Mantegazza P (2003) Open-source multibody analysis software. In Multibody Dynamics 2003, International Conference on Advances in Computational Multibody Dynamics, Lisboa, Portugal, July 1–4
21. Masarati P, Morandini M, Quaranta G, Mantegazza P (2005) Computational aspects and recent improvements in the open-source multibody analysis software "MBDyn". In Multibody Dynamics 2005, ECCOMAS Thematic Conference, Madrid, Spain, June 21–24
22. Fingersh LJ, Johnson K (2002) Controls advanced research turbine (CART) commissioning and baseline data collection. NREL/TP-500-32879
23. Stol KA (2003) Geometry and structural properties for the controls advanced research turbine (CART) from model tuning. NREL/SR-500-32087
24. Wright AD, Fingersh LJ, Balas MJ (2006) Testing state-space controls for the controls advanced research turbine. In 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada (USA), January 9–12
25. Wright AD, Fingersh LJ, Stol KA (2007) Designing and testing controls to mitigate tower dynamic loads in the controls advanced research turbine. In 45th AIAA Aerospace Sciences Meeting and Exhibit, Wind Energy Symposium, Reno, Nevada (USA), January 8–11
26. Stol KA, Fingersh LJ (2003) Wind turbine field testing of state-space control designs. NREL/SR-500-35061
27. Fuglseth TP (2005) Modelling a 2.5 MW direct driven wind turbine with permanent magnet generator. Technical report, Department of Electrical Power Engineering, Norwegian University of Science and Technology http://www.elkraft.ntnu.no/smola2005/Topics/9.pdf
28. Hansen MH, Hansen A, Larsen TB, Øye S, Sørensen P, Fuglsang P (2005) Control design for a pitch regulated, variable speed wind turbine. RISØ R-1500(EN), Denmark
29. Bucher R, Dozio L (2003) CACSD under RTAI linux with RTAI-LAB. In Fifth Real-Time Linux Workshop, Valencia, Spain, November 9–11
30. Lee D, Hodges DH (2004) Multi-flexible-body analysis for application to wind turbine control design. NREL/SR-500-35228
31. Bottasso CL, Croce A, Savini B, Sirchi W, Trainelli L (2006) Aero-servo-elastic modeling and control of wind turbines using finite-element multibody procedures. Multibody System Dynamics, 16:291–308
32. Masarati P, Lanz M, Mantegazza P (2001) Multistep integration of ordinary, stiff and differential-algebraic problems for multibody dynamics applications. In XVI Congresso Nazionale AIDAA, pages 71.1–10, Palermo, September 24–28
33. Shabana AA (1997) Flexible multibody dynamics: Review of past and recent developments. Multibody System Dynamics, 1(2):189–222
34. Wallrapp O, Schwertassek R (1991) Representation of geometric stiffening in multibody system simulation. Int. J. Num. Meth. Engng. 32:1833–1850
35. Géradin M, Cardona A (2001) Flexible Multibody Dynamics: a Finite Element Approach. John Wiley & Sons, Chichester
36. Ghiringhelli GL, Masarati P, Mantegazza P (2000) A multi-body implementation of finite volume beams. AIAA Journal, 38(1):131–138
37. Giavotto V, Borri M, Mantegazza P, Ghiringhelli GL, Caramaschi V, Maffioli GC, Mussi F (1983) Anisotropic beam theory and applications. Computers & Structures, 16(1–4):403–413
38. Hodges DH (2006) Nonlinear Composite Beam Theory. AIAA - c2006 - XII, Reston, VA
39. Bauchau OA (1985) A beam theory for anisotropic materials. J. Appl. Mech., 107:416–422
40. Chen H, Yu W, Capellaro M (2009) A critical assessment of computer tools for calculating composite wind turbine blade properties. Wind Energy, published online December 14
41. Brenan KE, Campbell SLaV, Petzold LR (1989) Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations. North-Holland, New York
42. Quaranta G, Masarati P, Mantegazza P (2004) Assessing the local stability of periodic motions for large multibody nonlinear systems using POD. J. of Sound and Vibration, 271(3–5):1015–1038

43. Meng F, Pavel MD, van Tooren M (2008) Aeroelastic stability analysis of large scale horizontal axis wind turbines using reduced order system identification based on flexible nonlinear multi-body dynamics. In 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada (USA), January 7–10. AIAA 2008-1302.
44. EasyAnim. http://mecara.fpms.ac.be/EasyDyn/ (last accessed March 2009).
45. Morandini M, Masarati P, Mantegazza P (2005) A real-time hardware-in-the-loop simulator for robotics applications. In Multibody Dynamics 2005, ECCOMAS Thematic Conference, Madrid, Spain, June 21–24
46. Morandini M, Masarati P, Mantegazza P (2005) Performance improvements in real-time general-purpose multibody virtual experimenting of rotorcraft systems. In 31st European Rotorcraft Forum, Firenze, Italy, September 13–15
47. Fumagalli A, Masarati P (2009) Real-time computed torque control using general-purpose multibody software. Multibody System Dynamics, 22(1):47–68