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

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

⋆Politecnico di Milano, Dipartimento di Ingegneria Aerospaziale
via La Masa 34, 20156 Milano, Italy
e-mail: cavagna,fumagalli,masarati,morandini,mantegazza@aero.polimi.it,
web page: http://www.aero.polimi.it/

March 30, 2009

Keywords: Wind-Turbine, Aeroservoelasticity, Multibody, Real-Time Simulation, Free Software

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, applied to a controlled constant-speed wind-turbine as an example of the proposed methodologies. A physical problem is controlled by a controller process that is 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

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 free Real-Time Operating System (RTOS) Real-Time Application Interface (RTAI, [1]) and the free general-purpose multibody software MBDyn [2], 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 academy and Small-Medium Enterprises (SMEs), access to powerful and versatile analysis and simulation capabilities.

A controlled constant-speed wind-turbine [3] is considered as one of the possible applications of the developed methodologies. Fig. 1 shows a sketch of the controlled model under analysis [4]. 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, [5]) 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:

Submitted to MULTIBODY DYNAMICS 2009, ECCOMAS Thematic Conference
K. Arczewski, J. Fraczek, M. Wojtyra (eds.) Warsaw, Poland, 29 June–2 July 2009

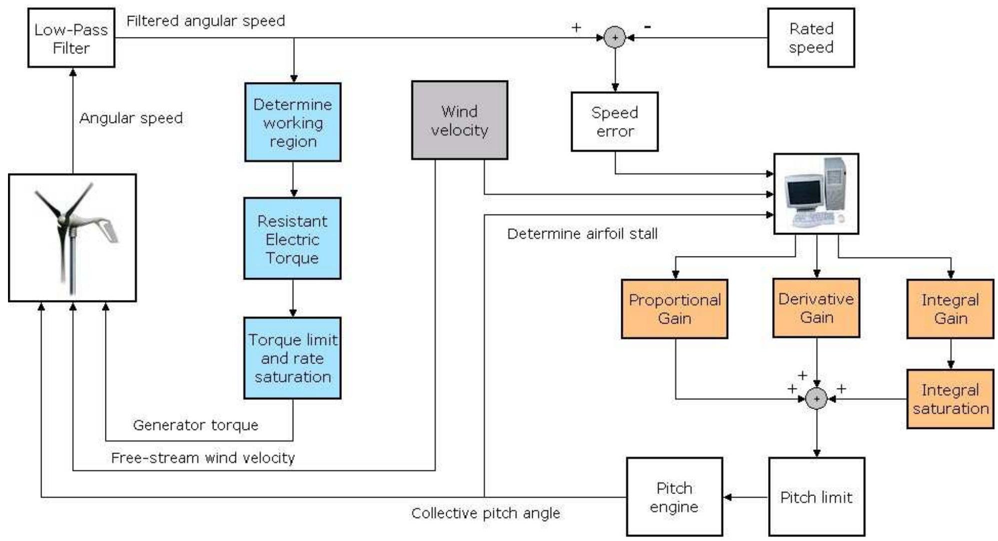


Figure 1: Sketch of the controlled model.

a) generating the gusty airstream input for the multibody analysis;

b) determining the generator's torque to be applied to the shaft of the wind-turbine;

c) computing the blade pitch input to be fed to 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, the whole problem could have been modeled within the general-purpose multibody simulation environment. On the contrary, the control-related part has been intentionally modeled in a separate general-purpose, graphically driven mathematical modeling environment. This represents the natural modeling environment for control systems. The multibody analysis, instead, represents a simulation of the real process to be controlled by the controller. 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 architecture.

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 comparison of some of them is presented in [6]. This document essentially compares software based on their capability to provide adequate results for benchmark problems. Of course this aspect is essential. For this purpose, MBDyn has been coupled to NREL's AeroDyn library in order to exploit the availability of a well-proven wind-turbine aerodynamic code [7]. 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.

Figure 2: The Controls Advanced Research Turbine (CART).

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 the real-time simulation. However, the redundant coordinate approach usually results in very sparse problems. When sparsity is efficiently handled by specialized linear algebra solvers, as the one proposed in [8], very good performances can be obtained in terms of computational time. This statement is supported by results recently obtained in the dynamic simulation of multibody systems: for models resulting from an acceptable trade-off between detail and synthesis, and sample rates complying with the desired accuracy requirements, general-purpose multibody software can be successfully used to perform real-time simulations of complex mechanical systems, including robots and rotorcraft wind-tunnel models [9, 10, 11]. This is the case of MBDyn [12, 13], the free multibody software that has been used and modified for the purpose of this research.

2 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 [14, 15, 16, 17]. 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.

3 BASELINE CONTROLLER

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

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 nacelle yaw (in the real wind turbine it is limited to only 0.5 deg and is simply used for tracking relatively small wind changes) and the high speed generator shaft brake.

Generator commands are calculated by means of a piece-wise function. Below cut-in speed of 10 rpm (Region 1) no electric torque is generated to let the wind accelerate the rotor, i.e. provide 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) [18, 19, 20]. Fig. 3(a) shows the piece-wise working function for the electric generator and its block-diagram model.

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 region during the initial acceleration phase. This feature enables to use the same controller both for rotor acceleration starting from null angular speed and constant-speed maintenance at the rated speed.

Fig. 4 shows the controlled system in the typical Scicos/Simulink/Matrixx environment. The box labeled 'from MBDyn' represents the multibody model output to the control system: only the angular velocity of the rotor is required for the basic control system considered in the present work. The box labeled 'to MBDyn' represents the inputs to the multibody model, which are: free-stream wind velocity, electric resistant torque and controlled pitch command.

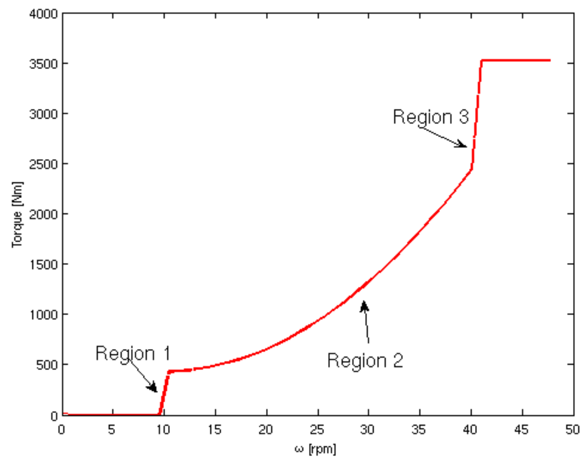
The controlled CART model can be run both in batch or in real-time mode. In the latter case, the real-time scheduling can be delegated to the operating system using a standard POSIX interface, or tightly enforced using RTAI. When executing in batch mode, a generic interface between MBDyn and the external 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 [21] can be used to monitor the controller, either locally or remotely, and to tune gains and other system parameters, as shown later in an example.

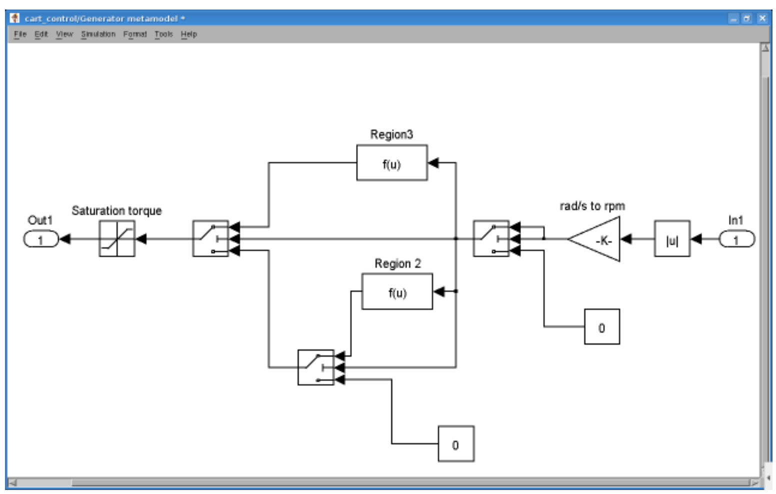
4 MULTIBODY MODEL

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 [22, 23]. 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 [2]. 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 [24]. The equations of motion of each unconstrained body are written in first order form using the Newton-Euler approach.



(a) Working function.



(b) Block-diagram.

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

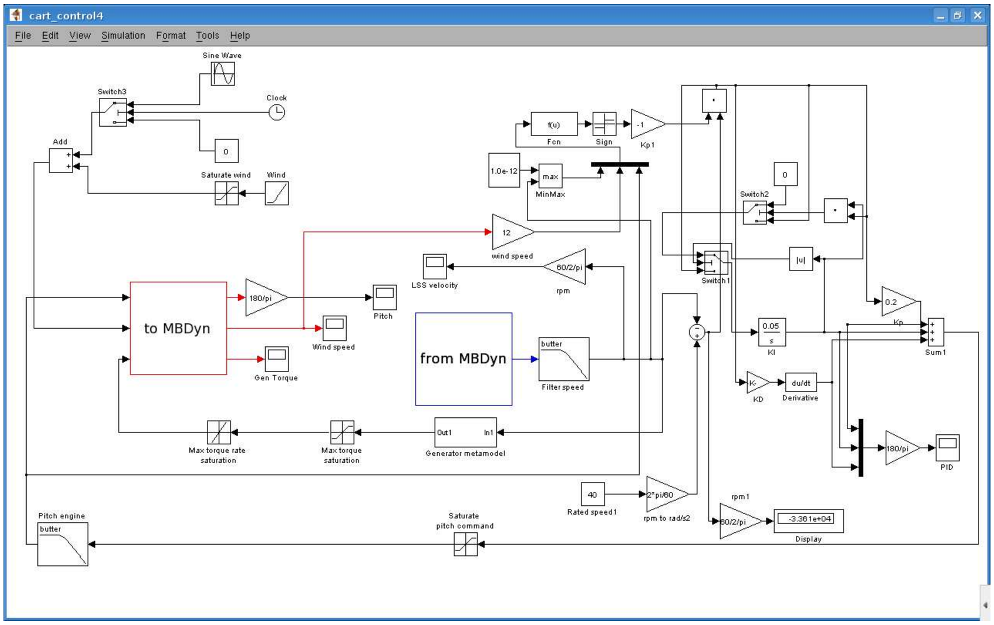


Figure 4: Controlled system in the typical Scicos/Simulink/Matrixx environment.

The definitions of momentum, $\boldsymbol{\beta}_i$, and momenta moment about the node, $\boldsymbol{\gamma}_{\mathbf{x}_i}$, for the i-th node are

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

$$\mathbf{s}_{i\mathbf{x}_i} \times \dot{\mathbf{x}}_i + \mathbf{J}_{i\mathbf{x}_i} \boldsymbol{\omega}_i = \boldsymbol{\gamma}_{i\mathbf{x}_i} \tag{1b}$$

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

$$\dot{\boldsymbol{\beta}}_i = \sum \mathbf{f}_i \tag{2a}$$

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

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

$$\mathbf{M}\dot{\mathbf{q}} = \mathbf{p} \tag{3a}$$

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

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, significantly those related to finite-volume beam elements [25], an original geometrically exact, composite ready beam analysis formulation.

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

$$\mathbf{M}\dot{\mathbf{q}}=\mathbf{p} \tag{4a}$$

$$\dot{\mathbf{p}}+\boldsymbol{\phi}_{/\mathbf{q}}^{T}\boldsymbol{\lambda}+\boldsymbol{\psi}_{/\dot{\mathbf{q}}}^{T}\boldsymbol{\mu}=\mathbf{f}(\mathbf{q},\dot{\mathbf{q}},\mathbf{p},t) \tag{4b}$$

$$\boldsymbol{\phi}(\mathbf{q},t)=0 \tag{4c}$$

$$\boldsymbol{\psi}(\dot{\mathbf{q}},\mathbf{q},t)=0 \tag{4d}$$

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

The implicit DAE problem of Eqs. (4) assumes the generic form

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

where $\mathbf{y}=\left[\mathbf{q}^{T},\mathbf{p}^{T},\boldsymbol{\lambda}^{T},\boldsymbol{\mu}\right]^{T}$ 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), $\delta\mathbf{y}_k=hb_0\delta\dot{\mathbf{y}}_k$; as a consequence, Eq. (5) yields

$$\left(\mathbf{g}_{/\dot{\mathbf{y}}}+hb_0\mathbf{g}_{/\mathbf{y}}\right)\delta\dot{\mathbf{y}}_k=-\mathbf{g}. \tag{8}$$

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 [26].

MBDyn is mainly used by its developers to model the aeroservoelasticity of rotary wing aircraft (e.g. [27]). Among the independent users, some use it also for the analysis of wind-turbine systems [28, 7].

The multibody model considered in this work consists in:

- a deformable tower, made of 5 three-node finite-volume beam elements, whose lower end is clamped to the ground;
- a rigid nacelle, connected to the tower with built-in pitch and by a yaw hinge; however, the yaw degree of freedom is restrained by a very stiff spring, since no yaw control is implemented;
- 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, each made of 5 three-node finite-volume beam elements, 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 formally independent, the same pitch angle is applied to each blade for simplicity.

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×	11	1	11	5	5		138
Total	36	7	35	15	10	1	465

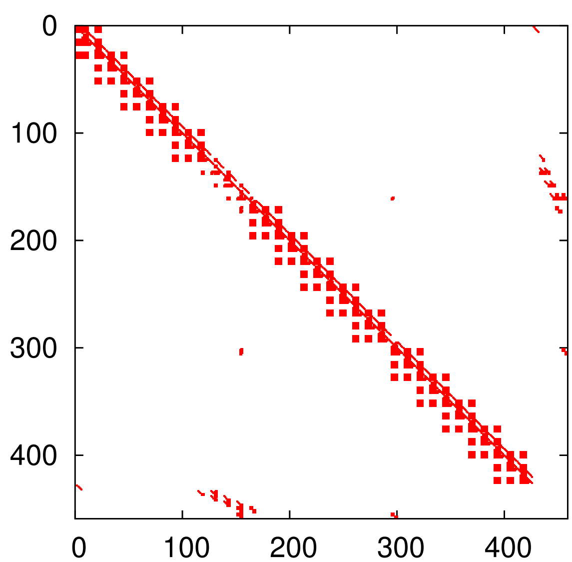


Figure 5: Non-zero coefficients of the CART model matrix; fill-in is about 3%.

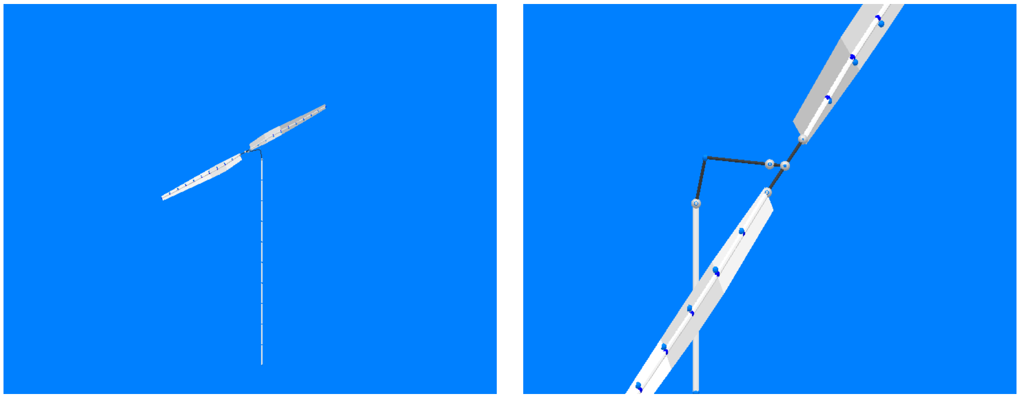


Figure 6: Graphical representation of the CART model.

Table 1 summarizes the number of nodes, elements and degrees of freedom of each component of the model. A total of 465 equations result: 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 of 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 [29] (the modified version is available in [2]).

The structural model of the tower and of the blades is even 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, non-rotating, with the blades in the vertical position. Similar trends are shown by all subcomponents.

5 REAL-TIME SIMULATION

The multibody analysis runs in real-time thanks to its support of RTAI, that is built-in in MBDyn when running on the Linux OS [9, 30, 31].

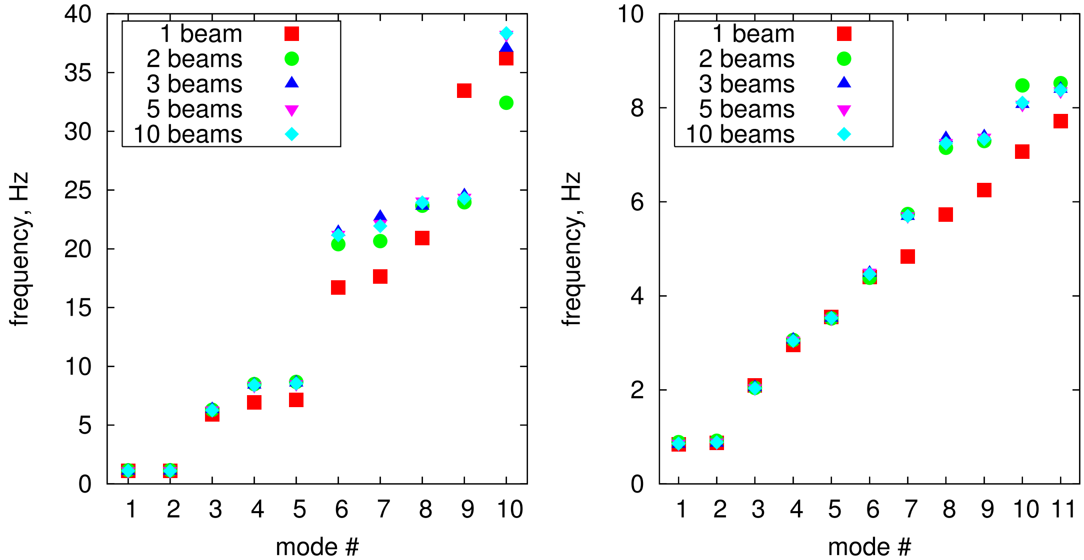
Popular graphical tools for control system design, like Scicos, Simulink, and Matrixx, have been extended to support the generation of the controller source code in the C programming language using RTAI's primitives for real-time scheduling and inter-process communication.

As a result, two real-time processes, one simulating the physical process and the other performing the appropriate control task, are executed on computers running the appropriate RTAI kernel modules 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. Inter-process communication uses RTAI mailboxes, a primitive that transparently uses shared memory or NetRPC when communicating either within a single computer, or different ones.

In more sophisticate 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 [32].

Two scheduling approaches can be followed. In one case, the processes synchronize one to 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



(a) Tower & nacelle

(b) All, blades vertical

Figure 7: CART model convergence on the first modes.

is waken 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 performed 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 factoring the matrix only at the first iteration of each time step. Errors due to the lack of convergence to the desired accuracy can be reasonably assumed to be small after few iterations, thanks to the nearly quadratic convergence properties of the modified Newton-Raphson scheme, as soon as the prediction at each time step is close enough to the actual solution. Errors due to the lack of convergence to the desired tolerance can be treated as disturbances by the control scheme. Only in case overruns are sporadic, they can be accepted and the lack of fresh measurements treated as a disturbance, 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 observers monitor the output of the controller and of the simulation, and optionally modify the parameters of the controller, using soft real-time connections.

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 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 environment.

The numerical simulations have been performed on a Dual Core AMD Opteron Pro-

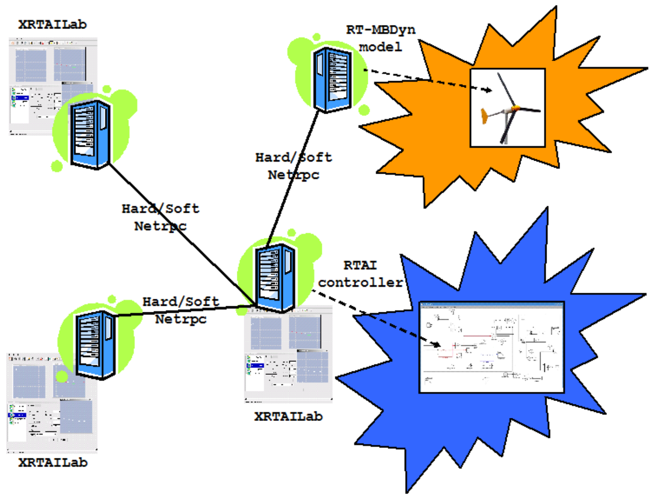
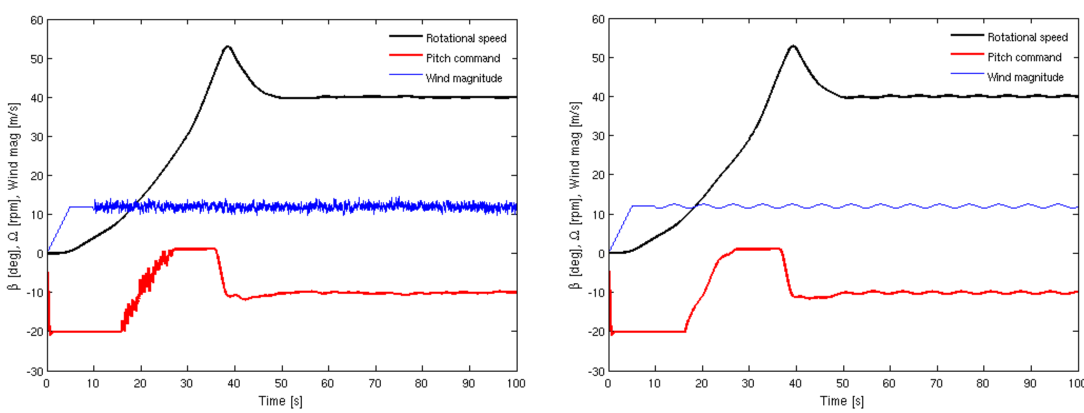


Figure 8: Sketch of a generic distributed real-time simulation layout.



(a) Random gust.

(b) Sinusoidal gust.

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

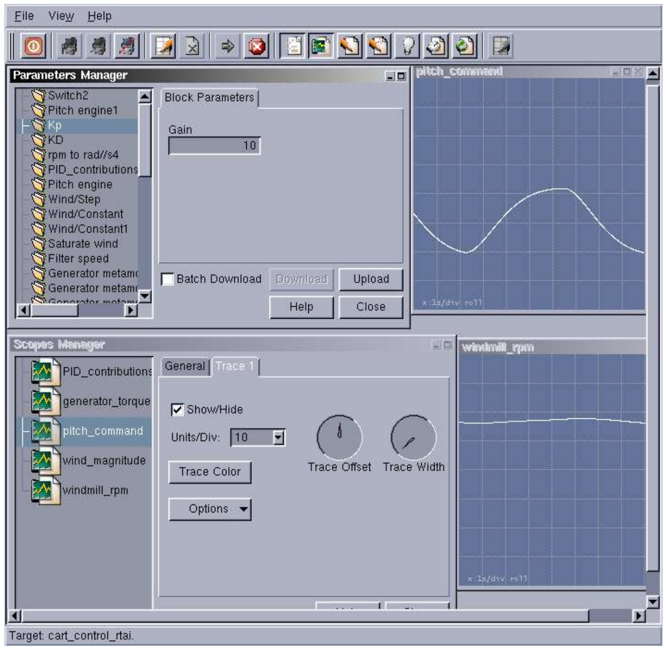


Figure 10: Output of the controlled CART model within the RTAILab environment.

cessor 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 refinement, for an increase of the sample rate, or for the analysis of more complex turbines, e.g. three-blade ones.

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 the control of strains, stresses, and gust load reduction in general.

CONCLUDING REMARKS

This work illustrates the implementation of what can be essentially 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. It has been applied to the development and testing of a controller for wind-turbines. Further details can be introduced both in the simulated physical process, to enhance system modeling with features that have not been considered so far as the primary target of this work.

ACKNOWLEDGMENTS

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.

ADDITIONAL MATERIAL

The analysis was performed using RTAI 3.6.1, available for download at [1], and MBDyn 1.3.3, available for download at [2]. The wind-turbine models and the controller source code are available at [2], in the RT-MBDyn→wind turbine folder. Feedback using the related mailing lists is appreciated.

References

[1] Real-time application interface (RTAI). `http://www.rtai.org/` (last accessed March 2009).

[2] Multibody dynamics (MBDyn). `http://www.aero.polimi.it/mbdyn/` (last accessed March 2009).

[3] R. Garsch and J. Twele. *Wind Power Plants, Fundamentals, Design, Construction and Operation*. James and James, 2002.

[4] J. Jonkman. NREL/NWTC report. Technical report, National Renewable Energy Laboratory, February 15 2007.

[5] National renewable energy laboratory (NREL). `http://www.nrel.gov/` (last accessed March 2009).

[6] M.L. Buhl Jr. and A. Manjock. A comparison of wind turbine aeroelastic codes used for certification. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada (USA), January 9–12 2006.

[7] F. Meng, P. Masarati, and M. van Tooren. 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 2009.

[8] M. Morandini and P. Mantegazza. Using dense storage to solve small sparse linear systems. *ACM Trans. Math. Softw.*, 33(1), March 2007.

[9] M. Attolico and P. Masarati. 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 2003.

[10] P. Masarati, M. Attolico, M. W. Nixon, and P. Mantegazza. 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 2004.

[11] M. Attolico, P. Masarati, and P. Mantegazza. Trajectory optimization and real-time simulation for robotics applications. In *Multibody Dynamics 2005, ECCOMAS Thematic Conference*, Madrid, Spain, June 21–24 2005.

[12] P. Masarati, M. Morandini, G. Quaranta, and P. Mantegazza. Open-source multibody analysis software. In *Multibody Dynamics 2003, International Conference on Advances in Computational Multibody Dynamics*, Lisboa, Portugal, July 1–4 2003.

[13] P. Masarati, M. Morandini, G. Quaranta, and P. Mantegazza. 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 2005.

[14] L. J. Fingersh and K. Johnson. Controls advanced research turbine (CART) commissioning and baseline data collection. NREL/TP 500–32879, National Renewable Energy Laboratory, October 2002.

[15] K. A. Stol. Geometry and structural properties for the controls advanced research turbine (CART) from model tuning. NREL/SR 500–32087, National Renewable Energy Laboratory, August 25–November 30 2003.

[16] A. D. Wright, L. J. Fingersh, and M. J. Balas. Testing state-space controls for the controls advanced research turbine. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada (USA), January 9–12 2006.

[17] A. D. Wright, L. J. Fingersh, and K. A. Stol. 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 2007.

[18] K. A. Stol and L. J. Fingersh. Wind turbine field testing of state-space control designs. NREL/SR 500–35061, National Renewable Energy Laboratory, August 25–November 30 2003.

[19] T. P. Fuglseth. 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, 2005. `http://www.elkraft.ntnu.no/smola2005/Topics/9.pdf`.

[20] M. H. Hansen, A. Hansen, T. B. Larsen, S. Øye, P. Sørensen, and P. Fuglsang. Control design for a pitch regulated, variable speed wind turbine. RISØ R–1500(EN), Riso National Laboratory, Denmark, January 2005.

[21] R. Bucher and L. Dozio. CACSD under RTAI linux with RTAI-LAB. In *Fifth Real-Time Linux Workshop*, Valencia, Spain, November 9–11 2003.

[22] D. Lee and D. H. Hodges. Multi-flexible-body analysis for application to wind turbine control design. NREL/SR 500–35228, National Renewable Energy Laboratory, September 2004.

[23] C. L. Bottasso, A. Croce, B. Savini, W. Sirchi, and L. Trainelli. Aero-servo-elastic modeling and control of wind turbines using finite-element multibody procedures. *Multibody System Dynamics*, 16:291–308, 2006. doi:10.1007/s11044-006-9027-1.

[24] P. Masarati, M. Lanz, and P. Mantegazza. Multistep integration of ordinary, stiff and differential-algebraic problems for multibody dynamics applications. In *XVI Congresso Nazionale AIDAA*, pages 71.1–10, Palermo, 24–28 Settembre 2001.

[25] G. L. Ghiringhelli, P. Masarati, and P. Mantegazza. A multi-body implementation of finite volume beams. *AIAA Journal*, 38(1):131–138, January 2000.

[26] K. E. Brenan, S. La V. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. North-Holland, New York, 1989.

[27] G. Quaranta, P. Masarati, and P. Mantegazza. Assessing the local stability of periodic motions for large multibody nonlinear systems using POD. *Journal of Sound and Vibration*, 271(3–5):1015–1038, 2004.

[28] F. Meng, M. D. Pavel, and M. van Tooren. 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), 7–10 January 2008. AIAA 2008-1302.

[29] EasyAnim. `http://mecara.fpms.ac.be/EasyDyn/` (last accessed March 2009).

[30] M. Morandini, P. Masarati, and P. Mantegazza. A real-time hardware-in-the-loop simulator for robotics applications. In *Multibody Dynamics 2005, ECCOMAS Thematic Conference*, Madrid, Spain, June 21–24 2005.

[31] M. Morandini, P. Masarati, and P. Mantegazza. Performance improvements in real-time general-purpose multibody virtual experimenting of rotorcraft systems. In *31st European Rotorcraft Forum*, Firenze, Italy, 13–15 September 2005.

[32] A. Fumagalli and P. Masarati. Real-time computed torque control using general-purpose multibody software. *accepted by Multibody System Dynamics*, 2009. doi:10.1007/s11044-009-9153-7.