PERFORMANCE IMPROVEMENTS IN REAL-TIME GENERAL-PURPOSE MULTIBODY VIRTUAL EXPERIMENT ON ROTORCRAFT SYSTEMS

Marco Morandini,[†] Pierangelo Masarati, and Paolo Mantegazza

Dipartimento di Ingegneria Aerospaziale
Politecnico di Milano, Italy

Abstract

This work presents the application of a general purpose multibody analysis software to the real-time simulation of realistic rotorcraft systems. The objective is to show how the same class of general-purpose simulation tools that are currently used for rotorcraft analysis and design could be used to simulate also the same type of problems in real-time, thus broadening their application to fields like realistic flight simulation, virtual wind-tunnel testing, pilot-in-the-loop rotorcraft design, and realistic rotorcraft-pilot coupling investigation. The issues and the limitations arising from the real-time requirements are illustrated using already existing multibody models of full-scale helicopters and tiltrotor wind tunnel models, so to highlight the possibility to share model components already available during the analysis phase of a rotorcraft design. The entire system runs on PC-class, off-the-shelf hardware and is entirely based on free software.

Introduction

Real-time simulation of the dynamics of complex systems represents a valid means to save time and resources when conducting experimental activity on expensive equipments in dangerous operating conditions, like aeroelastic stability clearance of rotorcraft models in wind-tunnel tests, or assessment of critical performances in equipments candidate for space deployment. Real-time simulation is usually performed by means of dedicated software, based on reduced set formulations to obtain maximal performances [Refs. 1, 2, 3]. However, these models can be inadequate for very sophisticated analyses and design, so such an approach leads to an inevitable duplication of software and model development, debugging, validation and tuning. On the contrary, a general-purpose multibody approach allows to model complex dynamical systems with increasing levels of sophistication in a single modeling environment. General-purpose multibody simulations provide accurate and realistic prediction of deformable aerospace mechanisms, and is becoming an industrial standard for the aerospace industry. The capability to exploit general-purpose modeling from the fully detailed, highly sophisticated analysis of the dynamics of deformable rotorcraft systems to the real-time simulation of simplified models, extreme of requirements spectrum, within just one software, or at least a single family of codes, using incrementally sophisticated versions of a single model, may represent a big advantage in terms of overall modeling and analysis efficiency, with significant savings in terms of time, training, hardware, software and human resources.

Multibody Simulation

The multibody simulation is performed by means of MBDyn, a free general purpose simulation software developed at the Dipartimento di Ingegneria Aerospaziale of the university "Politecnico di Milano" [Ref. 4, 5]. It is mostly aimed at for the solution of Initial Value Problems (IVP) in the form of Differential Algebraic Equations (DAE) by direct numerical integration, using a broad class of A/L-stable multistep algorithms [6]. The generic mechanical problem is described in form of differential equations of motion of a set of free bodies, possibly connected by

Presented at the 31st European Rotorcraft Forum, Florence, Italy, September 13–15 2005.

†Corresponding author
Tel.: +39 02 2399 8362
Fax: +39 02 2399 8334
E-mail: *marco.morandini@polimi.it*

configuration-dependent internal forces, like springs, beam or component mode synthesis elements

$$\boldsymbol{M}(\boldsymbol{x})\,\dot{\boldsymbol{x}} = \boldsymbol{q} \tag{1}$$
$$\dot{\boldsymbol{q}} = \boldsymbol{F}(\boldsymbol{x}, \dot{\boldsymbol{x}}) \tag{2}$$

Equation (1) defines the momentum and the momenta moment (angular momentum) $\boldsymbol{q}$ of a generic body as functions of the time derivative of the configuration $\boldsymbol{x}$ and of the configuration-dependent inertia $\boldsymbol{M}(\boldsymbol{x})$, while Equation (2) describes the equilibrium of the body subjected to the configuration dependent forces and moments $\boldsymbol{F}(\boldsymbol{x}, \dot{\boldsymbol{x}})$. The bodies can also be connected by kinematic constraints in form of algebraic equations

$$\boldsymbol{M}(\boldsymbol{x})\,\dot{\boldsymbol{x}} = \boldsymbol{q} \tag{3}$$
$$\dot{\boldsymbol{q}} + \boldsymbol{\Phi}^T_{/\boldsymbol{x}}\boldsymbol{\lambda} = \boldsymbol{F}(\boldsymbol{x}, \dot{\boldsymbol{x}}) \tag{4}$$
$$\boldsymbol{\Phi}(\boldsymbol{x}) = 0 \tag{5}$$

resulting in the addition of algebraic variables $\boldsymbol{\lambda}$ in form of Lagrange multipliers

The efficient handling of finite rotations is fundamental to obtain significant computational performances without losing accuracy. An updated Lagrangian approach is applied to the Gibbs-Rodriguez parameters that are used in MBDyn to represent the incremental orientation with respect to the *predicted* configuration. As a consequence, the orientation unknwons, i.e. the corrections to the predicted Gibbs-Rodriguez parameters, are $o(|\omega|\,\Delta t^n)$, while the corresponding parameters between two time steps would be of $O(\Delta t)$ instead, where n is the order of accuracy of the integration method. This greatly simplifies the computation of the most expensive nonlinear terms of the Jacobian matrix related to the orientation and reduces the computational effort required for the analytical computation of the matrix.

The software allows to simulate multidisciplinary problems, including hydraulic systems, controls and aerodynamic forces of increasing sophistication ranging from strip theory (with simple inflow models for rotorcraft applications) to state-space representations, to free-wake modeling.

Real-Time Simulation

Real-time simulation capabilities have been obtained by augmenting the already mentioned general-purpose, open source multibody analysis software MBDyn with the real-time utilities offered by the Real-Time Application Interface (RTAI) for the Linux OS [`http://www.rtai.org/`, Ref. 7]. All of the above described software is free (it is released under the GPL license), which means that it is freely available in source form, thus giving the broadest accessibility to all of its internals without limitations on its usage. On the one hand, it is worth stressing the importance of this aspect for highly advanced applications; on the other hand, this can help reducing the costs of the analysis infrastructure at a company-wide level, an issue that is critical especially at the small-medium enterprise (SME) level.

The use of a multitasking, network enabled OS as underlying platform allows to perform the analysis in a fully integrated computational environment. This is fundamental because the simulation must interoperate with the rest of the experimental setup, including data acquisition, conditioning and visualization, and model and experiment control. Most of these tasks can be performed directly by automatical generation of operation and monitoring (O&M) and control code from Matlab's Simulink or Scilab's Scicos, which is obtained by using RTAI's companion RTAILab [Ref. 8]. An example of the RTAILab graphical user interface is illustrated in Figure 1, where outputs from the simulation of a robot manipulator [Ref. 9] are monitored, while control gains can be adjusted on the fly.

One fundamental requirement of the present work is that the real-time extension of the general-purpose multibody software implies minimal impact on the original software and its behavior; in fact, the changes are beneficial also to batch simulations. The process of real-time enabling applies to any software, but needs caring of four aspects:

- avoid system calls which return control to the OS and thus cause loss of pre-emption;
- preserve enough momory locked stack space before entering real-time mode, to avoid memory paging and swapping during the simulation;
- insert a minimum amount of specific control statements to initialize the real time task, force the execution into hard real-time and synchronize with the process scheduling;
- provide appropriate I/O mechanisms based on primitive real-time communication

The modifications have been successfully applied to MBDyn with minimal impact thanks to its modular design [Ref. 10].

Modeling Issues

Despite the overall efficiency of the simulation software, real-time simulations pose very strict constraints on the execution time for each time step. As a result, only very compact models can be simulated at the sampling rates required by sophisticated control systems. Special care has been put in eliminating all

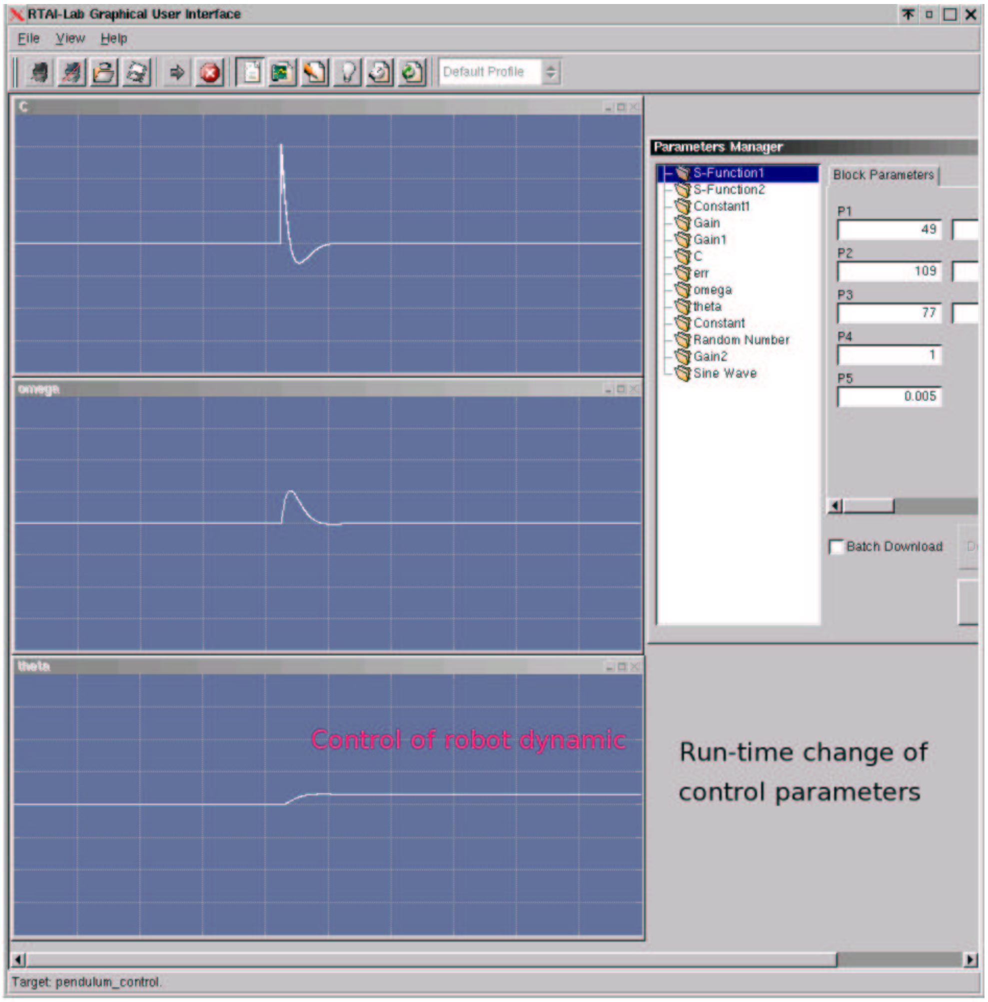

Figure 1: RTAILab control panel.

the unnecessary unknowns, by developing appropriate modeling strategies. Significant improvements have been obtained by simplifying the model whenever the impact of the approximation was minimal with respect to the target analysis, e.g. by condensing the inertia of subcomponents or joint details.

A noteworthy example is the ideal gimbal joint that is used in some tiltrotor configurations to allow the tilting of the angular velocity vector of the rotor disk along with the disk itself, in order to avoid in-plane tilting moments that would introduce high loads on the root of the blades and in the wing.

A first-order approximation of a gimbal may be obtained by means of the so-called "universal" joint (the Cardano joint) which, in the redundant coordinate set approach used in MBDyn, adds four algebraic equations and four Lagrange multipliers to the problem. This approximation may have limited impact on the analysis of the aeroelastic stability of a tiltrotor model, when the tilt angle of the rotor disk is small enough; however, the 2/rev oscillations introduced by a single Cardano joint may become an issue when vibratory loads are addressed, or when finite tilting must be considered; in those cases, an ideal gimbal joint is mandatory.

An ideal gimbal consists in a sequence of two Cardano joints that undergo half of the relative orientation each, as shown in Figure 2. As a consequence, the 2/rev second-order perturbations on the axial velocity of the shafts introduced by each joint are mutually cancelled. Its modeling using general purpose base joint elements in MBDyn requires two extra nodes (6 equations each), connected by one spherical (3 equations) and two revolute (5 equations each) joints, for a total of 25 equations. The use of a specially designed joint reduces this figure to 5 algebraic equations, resulting in an appreciable improvement in computational time for models of the order of 100÷200 equations.

The relative orientation $\boldsymbol{R}_{rel}$ between bodies a and b is

$$\boldsymbol{R}_{rel} = \boldsymbol{R}_a^T \boldsymbol{R}_b \tag{6}$$

According to the definition of the kinematics of this joint, by calling $\boldsymbol{e}_i$ the unit vector in direction i, the relative orientation between the two bodies must take the form

$$\begin{aligned} \boldsymbol{R}_{rel} &= \exp\left(\vartheta \boldsymbol{e}_2 \times\right) \exp\left(\varphi \boldsymbol{e}_1 \times\right) \exp\left(\vartheta \boldsymbol{e}_2 \times\right) \\ &= \boldsymbol{R}_{\vartheta,\varphi} \end{aligned} \tag{7}$$

where ϑ and $\varphi/2$ are the angles about local axes 1 and 2 of each of the two Cardan joints, while the torque is trasmitted about axis 3. The gimbal equations result in

$$\operatorname{ax}\left(\exp^{-1}\left(\boldsymbol{R}_{rel}\right)\right) - \operatorname{ax}\left(\exp^{-1}\left(\boldsymbol{R}_{\vartheta,\varphi}\right)\right) = \boldsymbol{0} \tag{8}$$

$$\boldsymbol{e}_2^T \left(\boldsymbol{I} + \exp\left(\varphi \boldsymbol{e}_1 \times\right) \exp\left(\vartheta \boldsymbol{e}_2 \times\right)\right) \boldsymbol{\lambda} = 0 \tag{9}$$

$$\boldsymbol{e}_1^T \exp\left(\vartheta \boldsymbol{e}_2 \times\right) \boldsymbol{\lambda} = 0 \tag{10}$$

where Equation (8) constrains the relative orientation of the two bodies to be equal to its representation as a function of ϑ and $\varphi/2$, while Equations (9–10) define the values of the Cardan joint angles. The $\boldsymbol{\lambda}$ are the Lagrange multipliers that represent the reaction couples; their projection in the global frame occurs by way of the orientation of node a:

$$\boldsymbol{C}_a = \boldsymbol{R}_a \boldsymbol{\lambda} \tag{11}$$

$$\boldsymbol{C}_b = -\boldsymbol{R}_a \boldsymbol{\lambda} \tag{12}$$

More details about the formulation of this joint can be found in the technical manual of MBDyn, available from its website. It has been used and specifically developed for the analysis of an advanced tiltrotor model described in the applications section of the present paper.

Computational Issues

Today's state-of-the-art general-purpose multibody simulation is heavily oriented toward redundant coordinate set formulations, which make automatic equation generation very easy and efficient, while the handling of the resulting large size problems is delegated to efficient sparse solvers. The minimal coordinate set is losing appeal, since the reduced size of the problem is obtained at the cost of a high computational effort to perform the reduction in a numerical way; symbolic manipulation does not appear to be a valid alternative yet, although yielding good results in robots and manipulators simulation [Refs. 11, 12, 13, 14].
However, available sparse solvers, although very efficient in terms of memory footprint, are often tailored for very large problems, e.g. FEM and CFD analysis, losing appeal for the small and very small size problems resulting from affordable real-time simulation of space robotics and relevant aerospace mechanisms in general. In fact, is has been noticed that state-of-the art sparse solvers cannot compete with state-of-the-art dense solvers below 100 unknowns (e.g. the publicly available Umfpack 4.4, the default Matlab sparse solver, or SuperLU, as opposed to Lapack); however, there is room for some speedup, at the expense of memory consumption. A very specialized sparse solver has been implemented for this purpose in MBDyn.

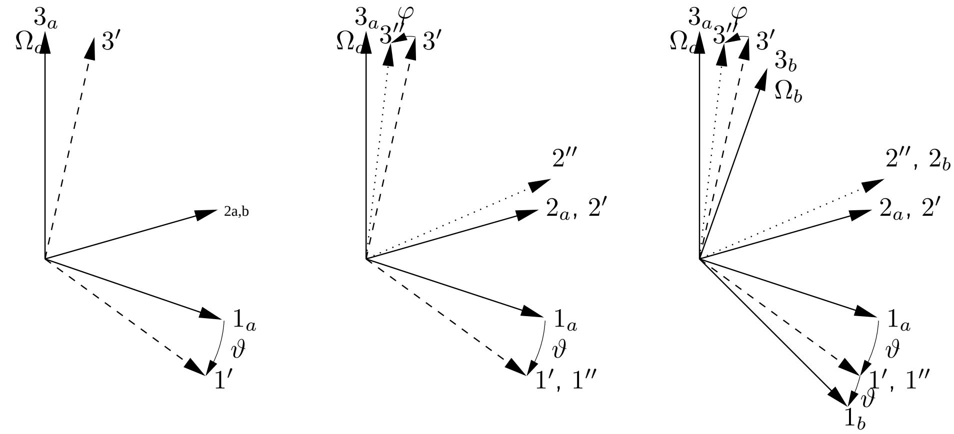

Figure 2: Gimbal relative orientation decomposition; the norm of Ω_b, the angular velocity of body b when angles ϑ and φ are constant, is equal to the norm of Ω_a, the angular velocity of body a.

Its application to space robot and rotorcraft analysis showed overall execution time reductions of roughly 40% compared to using state-of-the-art sparse solvers [Ref. 15, 16].

As opposed to common expectation, the parallelization of the problem assembly and of the matrix factorization and linear algebra solution did not yield any appreciable overall execution time reduction for this specific class of problems. However, its investigation has not been dropped, because it resulted beneficial for larger problems [> 500 equations, Ref. 16], and, as such, will likely become significant for that class of problems as soon as hardware and software development will make them affordable in real-time.

Applications

The real-time simulator described in the first part of this paper has so far been applied in two distinct fields:

- the simulation of wind-tunnel rotorcraft models [Ref. 17];
- the simulation of controlled deformable space robot arms [Refs. 10, 9].

This paper only addresses selected results from rotorcraft simulations; robotics applications are mentioned to highlight the implications of the different fields in defining the requirements for a general purpose software tool.

The presented performance figures have been obtained on an Athlon 64 3000+ (2 GHz) single CPU, 64 bit architecture, with a 512 KB L2 cache. Some of the test cases have been also simulated on an Athlon XP 2400+ 2 GHz single CPU, with a 256 KB L2 cache; other tests have been carried on an Athlon MP 2200+ 1.8 GHz dual CPU, which was used to investigate the potential for speedup improvements associated to parallel execution of MBDyn in a multithreaded SMP environment as described in [Ref. 16]. The performances on the single CPU 32b hardware were relatively good, although not directly comparable to those obtained with the Athlon 64. No significant improvements have been obtained in the SMP case.

Simulation of Tiltrotor Wind-Tunnel Models

The availability of multibody data of two different wind-tunnel models of tiltrotors allowed to assess the feasibility of their simulation in real-time. Of course, the original models, with deformable blades and wing, highly detailed hub and control system kinematics, had to be reduced to relatively coarse models, essentially with rigid blades and in some cases with slightly simplified hub kinematics.

WRATS model. First, the 1:5 scale model of the V–22, known as the Wing Rotor Aeroelastic Test System (WRATS) has been addressed in its recent four-blade, soft-inplane configuration (Figure 3). Initial, encouraging results have been obtained without aerodynamic forces, as reported in [Ref. 17], but the yet high computational time required by the simulation did not allow the investigation of realistic setups including aerodynamic forces.

The above tests have been repeated with the new sparse linear solver and other software improvements on more powerful hardware. Full real-time simulation capabilities have not been reached yet for that model on the available hardware and with the desired accuracy. To give a rough figure, in [Ref. 17] the best simulations of the fully articulated, rigid blade model with control system details, component mode synthesis wing support and aerodynamics, resulting in 174

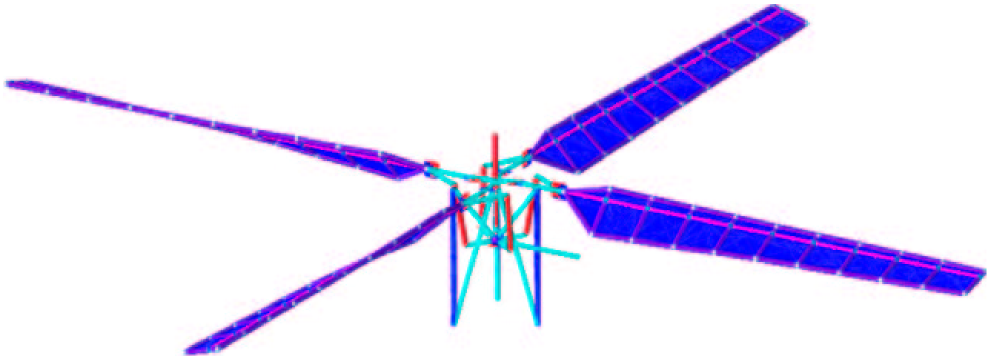

Figure 3: WRATS SASIP wind tunnel model

Table I: Advanced tiltrotor wind-tunnel model at 1 kHz (87 steps/rev)

Wing model	Gimbal el.	N. Eqs.	Real/sim.
yes	no	159	0.840
no	no	151	0.705
yes	yes	139	0.648
no	yes	131	0.533

equations, could be run in real-time only at a sampling rate of 100 Hz with a rotation speed of 200 rpm, corresponding to 22% of the full hover rpm and to a fairly inaccurate 30 steps/rev. The same model, after the mentioned software improvements and on the Athlon 64 hardware can now be run in real-time at the nominal hover rotation speed, 875 rpm, at a sampling rate slightly below 600 Hz, corresponding to 40 steps/rev. The availability of similar hardware with clock frequencies 1.5 times faster promises to allow 60 steps/rev and above, which should yield an appreciable improvement in terms of accuracy.

Advanced tiltrotor model. The multibody aeroelastic model of an advanced tiltrotor wind-tunnel model, initially developed to support an experimental whirl-flutter investigation campaign, has been turned into a real-time simulation model as well, in view of its possible use for the training of the wind-tunnel crew that will pilot the model during aeroelastic stability tests. This suggested the development of the previously described ideal gimbal joint element as well as other minor adjustments. A single load path model of the otherwise rather complicated stiff-inplane hingeless hub had to be prepared, to further reduce the number of equations required by the problem and allow its simulation at realistic time steps. Table I summurises some some significant timing results. The "Real/sim." column reports the ratio between the minimum wall clock time required to perform the simulation and the simulated time; that is, the "Real/sim." index is equal to 0.5 if the simulation of 10 s takes at least 5 s. The lower the index, the more idle time is left to the system when the simulation is performed in real-time. All the results of Table I are obtained with the single load path model in forward flight at reduced rotation speed (half the model-scale nominal hover rpm, resulting in roughly 75% of the model-scale forward flight speed). The wing is modeled by means of the component mode synthesis approach; the first four modes, resulting in 8 extra equations, are used. Note how the use of the presented gimbal joint element, by saving 20 equations ($\approx$15% of the model size), greatly reduces the computational time, thus allowing to increase the sample rate, either to allow more accurate simuations (more steps/rev) or higher rotation speeds towards the forward flight nominal model scale speed. For example, the model with gimbal, but without wing, runs in 0.940 real/simulated time at 2 KHz at the nominal forward flight rpm with 113 steps/rev, with a simultaneous increase in rotor speed and simulation accuracy.

Figures 5 and 6 show the behavior of the advanced tiltrotor model when the wind-tunnel is started, and the unpowered model, in wind-mill configuration, spins up during the airstream speed transient. The rotational speed of the model is controlled by means of an integral controller, that feeds the integral of the speed error into the collective control. Figures 7 and 8 show the same parameters when a sinusoidal perturbation of the wind tunnel velocity is introduced. The amplitude of the perturbation is 10 m/s, which is about 20% of the wind tunnel speed, and the frequency is 0.2 Hz. With a relatively low control gain, the error in the rotor speed is very limited, roughly 1%. These simulations illustrate how realistic operating conditions can be easily simulated by the system, providing a detailed and complete environment for both the training of the model operators and the verification of the real test equipment that will be actually used in the wind tunnel.

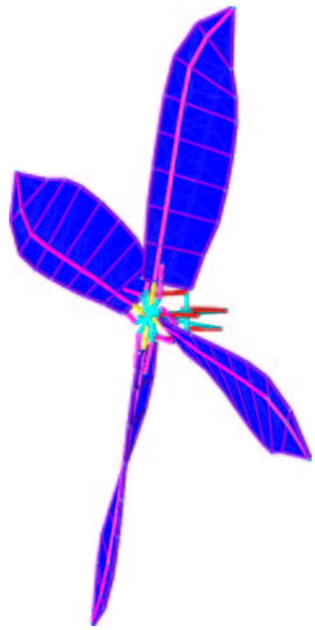

Figure 4: Advanced tiltrotor multibody model.

Simulation of Helicopters

The real-time simulation of wind-tunnel models suffers from the fact that, while having roughly the same modeling complexity of a real helicopter, they need to spin much faster because of scaling issues. Typically, rotation speeds depend on the scaling factor (s.f.) in the ratio of $(1{:}\sqrt{\text{s.f.}})$ for Froude-scale models or (1:s.f.) for Mach-scale models, where typical rotor model scale factors range from 1:2.5 to 1:7 ÷ 1:8 and higher.

An essential requirement is the capability to run about 80÷100 time steps per revolution, for the accurate integration of the blade dynamics. The previously presented wind-tunnel models nearly match it; the 1:5 WRATS model because of the Froude scale, and the 1:2.5 advanced tiltrotor model because of the unusually large scale. However, smaller scale, Mach-scale models would hardly meet such a strict requirement. Nonetheless, a real helicopter, which rotates at much lower speeds (250÷400 rpm compared to 800÷1200 and higher) is more likely to fit into the constraints of current affordable hardware.

For this reason, the real-time simulation of the main rotor of the AS330 Puma is presented. It is worth noticing that this model has been developed in view of its use for the fluid structure interaction investigation described in [Ref. 18]; the very same model is here used with the very same code for a completely different application, illustrating the versatility of the proposed multibody approach.

Table II: AS330 Puma flight 123 parameters [Ref. 19].

Advance ratio, μ	0.321	
Shaft angle of attack, α_s	-6.0	deg
Collective pitch, θ_c	13.2	deg
Lateral cyclic pitch, θ_{1c}	2.1	deg
Longitudinal cyclic pitch, θ_{1s}	-7.15	deg

The AS330 Puma model has not been optimized for real-time simulation yet; in fact, some of the blade root joints could be synthesized in much more compact constraints, with minimal impact, if any, on the quality of the model, while saving between 50 and 90 equations on a total of 283. Nonetheless, it runs in real-time at 400 Hz with 90 steps per revolution, with a 0.830 real/simulated time ratio that gives some slight margin for further improvements. The simulated flight condition is the high speed test flight indicated as “flight 123” [Ref. 19], and reported in Table II. Figure 10 shows the very same multibody rotor model in a fluid-structure interaction simulation of the same flight condition [Ref. 18].

The capability to run models of this type in real-time opens the prospect of designing helicopter flight simulators based on accurate, first principles flight mechanics with relatively detailed rotor dynamics, which could be used to investigate complex pilot-in-the-loop flight dynamics phenomena like Rotorcraft-Pilot Couplings (RPC).

One issue, when designing a real-time simulator for conventional helicopters in free-flight, may be the tail rotor, which falls into the rpm range of wind tunnel models and above. Currently, an equivalent dynamic model must be used, otherwise the size of the problem would increase while the time step would need to be decreased, making the real-time simulation absolutely unfeasible. To overcome this limitation, the possibility of concurrently running a separate simulator for each rotor is being explored. However, it is worth noticing that in any case aerodynamic interaction issues would likely make the detailed real-time simulation of tail rotor dynamics approximate.

Concluding Remarks

A real-time, general purpose multibody simulation environment based on free(dom) software has been presented. Although not specifically designed for this task, the system shows that general-purpose analysis can be run in real-time, with the due simplifications, on off-the-shelf hardware. The capability to perform real-time simulation with a tool that is also capable of running detailed analysis of rotorcraft allows commonality of tools, models and expertise. The simulation

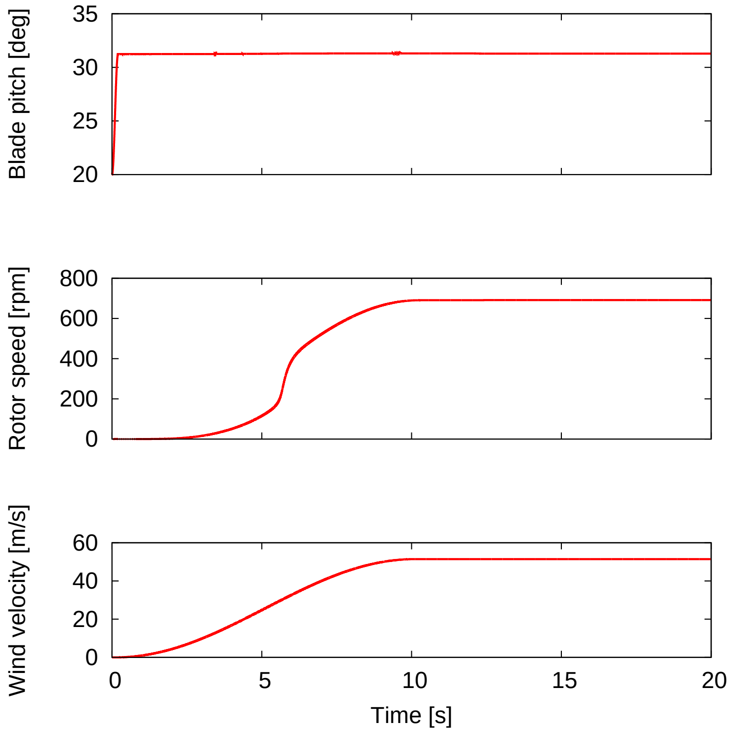

Figure 5: Advanced tiltrotor model real-time simulation of automatic velocity control by means of the swash-plate controls during windup: wind-tunnel speed, rotor speed and blade pitch.

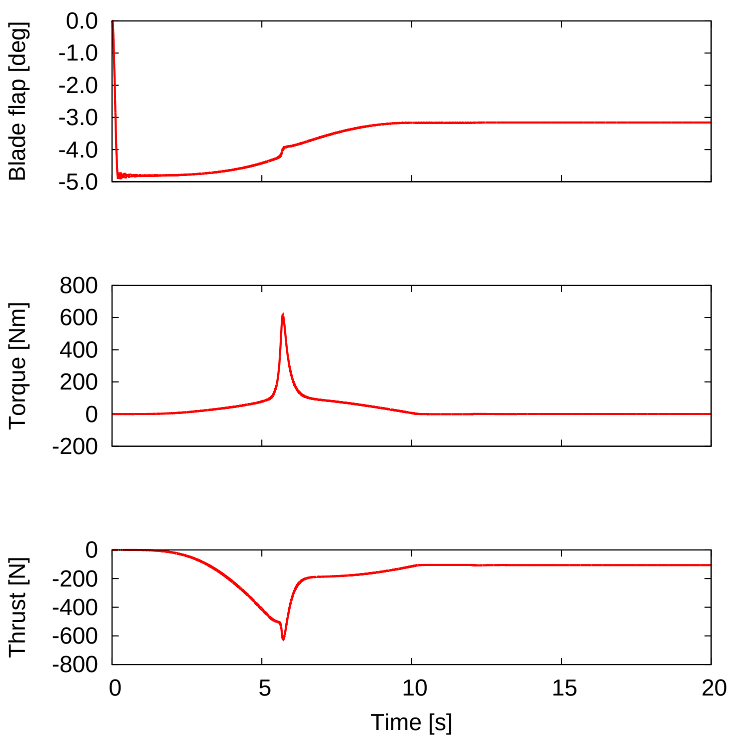

Figure 6: Advanced tiltrotor model real-time simulation of automatic velocity control by means of the swash-plate controls during windup: thrust, torque and blade flap.

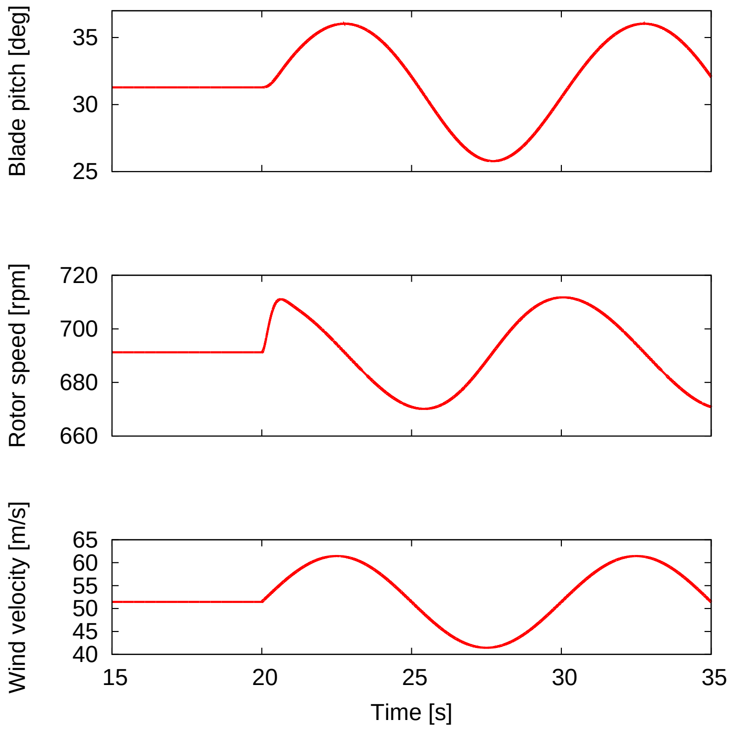

Figure 7: Advanced tiltrotor model real-time simulation of automatic velocity control by means of the swash-plate controls subjected to wind-tunnel velocity oscillations: wind-tunnel speed, rotor speed and blade pitch.

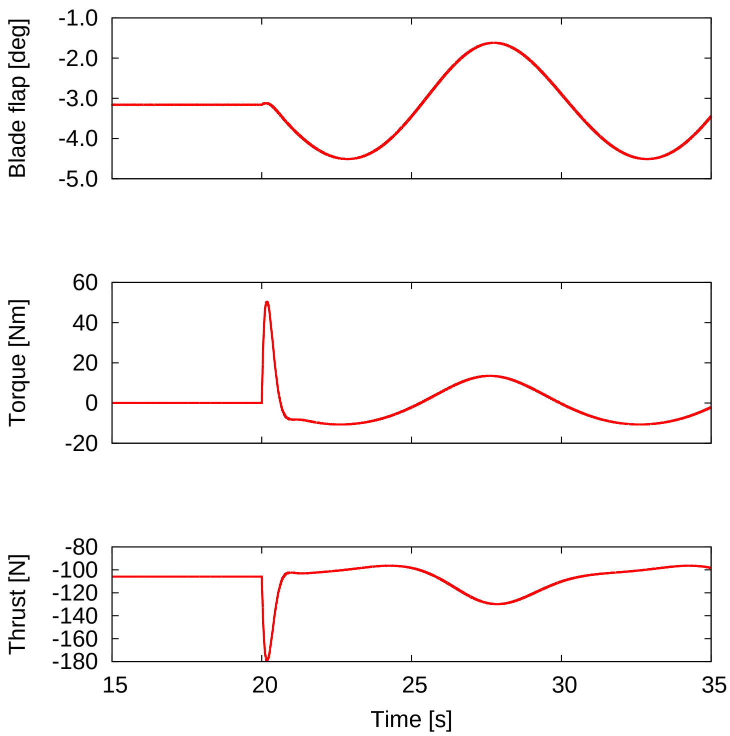

Figure 8: Advanced tiltrotor model real-time simulation of automatic velocity control by means of the swash-plate controls subjected to wind-tunnel velocity oscillations: thrust, torque and blade flap.

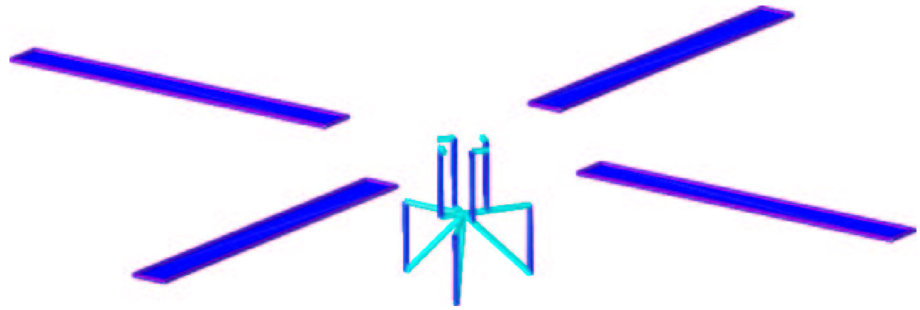

Figure 9: AS330 Puma multibody model.

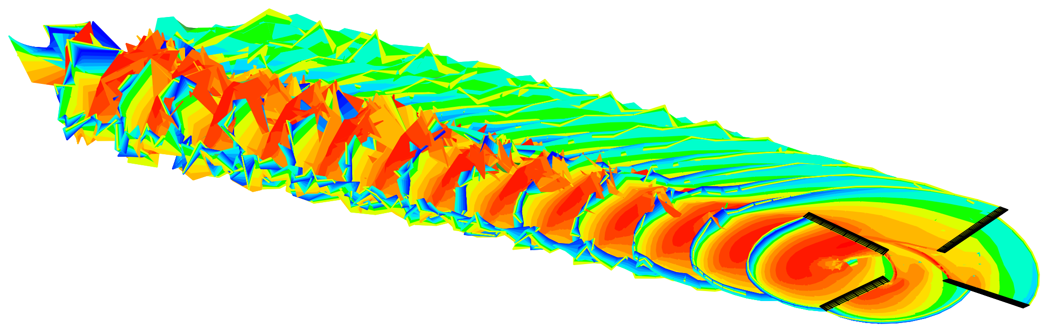

Figure 10: AS330 Puma flight 123 fluid-structure interaction simulation [Ref. 18].

can be cast in a broader simulation environment including control components automatically generated from Simulink or Scicos block diagrams, that are becoming the industrial standard for analysis and design of integrated system. Previously validated multibody models of tiltrotor wind-tunnel models and helicopters have been run in real-time in rather realistic operating conditions.

Acknowledges

The Authors acknowledge the contributions of Michele Attolico and Matteo Martegani to the implementation of the real-time extension of MBDyn and to the integration of RTAI in the virtual experiment setup.

References

[1] Dae-Sung Bae, Ruoh-Shih Hwang, and Edward J. Haug. A recursive formulation for real-time dynamic simulation of mechanical systems. *Journal of Mechanical Design*, 113: 158–166, June 1991.

[2] Javier Garcia de Jalon and Eduardo Bayo. *Kinematic and Dynamic Simulation of Multibody Systems: the Real Time Challenge*. Springer-Verlag, New York, 1994.

[3] J. García de Jalón, E. Álvarez, F.A. de Ribera, J.I. Rodríguez., and F.J. Funes. A fast and simple semi-recursive dynamic formulation for multi-rigid-body systems. In *Multibody Dynamics 2003, International Conference on Advances in Computational Multibody Dynamics*, Lisboa, Portugal, July 1–4 2003.

[4] Gian Luca Ghiringhelli, Pierangelo Masarati, Paolo Mantegazza, and Mark W. Nixon. Multi-body analysis of a tiltrotor configuration. *Nonlinear Dynamics*, 19(4):333–357, August 1999.

[5] Pierangelo Masarati, Marco Morandini, Giuseppe Quaranta, and Paolo Mantegazza. Open-source multibody analysis software. In *Multibody Dynamics 2003, International Conference on Advances in Computational Multibody Dynamics*, Lisboa, Portugal, July 1–4 2003.

[6] Pierangelo Masarati, Massimiliano Lanz, and Paolo 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.

[7] Lorenzo Dozio and Paolo Mantegazza. Real time distributed control systems using RTAI. In *Sixth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC)*, Hakodate, Hokkaido, Japan, May 14–16 2003.

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

[9] Marco Morandini, Pierangelo Masarati, and Paolo 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.

[10] Michele Attolico and Pierangelo 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.

[11] Michael Tiller. *Introduction to Physical Modeling with Modelica*. Kluwer Academic Publishers, 2001.

[12] Martin Otter, Hilding Elmqvist, and Sven Erik Mattsson. The new modelica multibody library. In Peter Fritzson, editor, *3$^{\text{rd}}$ International Modelica Conference*, Linköping, Sweden, November 3–4 2003.

[13] Gianni Ferretti, Marco Gritti, Gianantonio Magnani, and Paolo Rocco. A remote user interface to modelica robot models. In Peter Fritzson, editor, *Proceedings of the 3$^{\text{rd}}$ International Modelica Conference*, pages 231–240, Linköping, Sweden, November 3–4 2003.

[14] Gianni Ferretti, Gianantonio Magnani, Paolo Rocco, Luca Viganò, Marco Gritti, and Andrea Rusconi. Object-oriented modeling of a space robotic manipulator. In *Proceedings of the 8$^{\text{th}}$ ESA Workshop on Advanced Space Technologies for Robotics and Automation, ASTRA 2004*, ESTEC Noordwijk, The Netherlands, November 2–4 2004.

[15] Marco Morandini and Paolo Mantegazza. Using dense storage to solve small sparse linear systems. Submitted to *ACM Transactions on Mathematical Software* (ACM TOMS).

[16] Pierangelo Masarati, Marco Morandini, Giuseppe Quaranta, and Paolo 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.

[17] Pierangelo Masarati, Michele Attolico, Mark W. Nixon, and Paolo Mantegazza. Real-time multibody analysis of wind-tunnel rotorcraft models for virtual experiment purposes. In *AHS 4^{th} Decennial Specialists' Conference on Aeromechanics*, Fisherman's Wharf, San Francisco, CA, January 21–23 2004.

[18] Giuseppe Quaranta, Giampiero Bindolino, Pierangelo Masarati, and Paolo Mantegazza. Toward a computational framework for rotorcraft multi-physics analysis: Adding computational aerodynamics to multibody rotor models. In 30^{th} *European Rotorcraft Forum*, pages 18.1–14, Marseille, France, 14–16 September 2004.

[19] William G. Bousman, Colin Young, François Toulmay, Neil E. Gilbert, Roger C. Strawn, Judith V. Miller, Thomas H. Maier, Michel Costes, and Philippe Beaumier. A comparison of lifting-line and CFD methods with flight test data from a research Puma helicopter. TM 110421, NASA, October 1996.