XVIII CONGRESSO NAZIONALE AIDAA
19–22 settembre 2005
VOLTERRA (PI)

GENERAL-PURPOSE MULTIBODY REAL-TIME SIMULATION AND CONTROL OF DEFORMABLE AEROSPACE MECHANISMS

M. MORANDINI[1], P. MASARATI[1], P. MANTEGAZZA[1]

[1] *Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano*

ABSTRACT

The paper illustrates the real-time simulation capabilities of general-purpose multibody analysis applied to the challenging problem of rotorcraft dynamics, where the need to use small sample rates to capture the dynamics related to the rotation of the rotor conflicts with the need for small worst-case computational time for each time step. The simulation system is entirely based on free software, most of which developed at the Dipartimento di Ingegneria Aerospaziale of the University “Politecnico di Milano”. Relevant performances resulting from its application to the analysis of different rotorcraft models are presented.

1 INTRODUCTION

General-purpose multibody simulation provides accurate and realistic prediction of deformable aerospace mechanisms, and is becoming an industrial standard for the aerospace industry. Real-time simulation of the dynamics of complex systems represents a valid means to save time and resources when conducting experimental activity on expensive equipment in dangerous operating conditions, like aeroelastic stability clearance of rotorcraft models in wind-tunnel experiments, or assessment of critical performance in equipment that is 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 analysis and design, so this approach leads to an inevitable duplication of software and model development, debug, 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. The capability to exploit this modeling paradigm in the two extremes of the requirement spectrum of the industry, i.e. from fully detailed, highly sophisticated model analysis to simplified real-time simulation, within just one, or at least a single family of codes, and 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, and hardware, software and human resources.

2 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). It is mainly intended for the solution of Initial Value Problems (IVP) in form of Differential Algebraic Equations (DAE) by direct numerical integration, using a broad class of A/L-stable multistep algorithms (5).

The generic mechanical problem is described in terms of the differential equations of motion of a set of free bodies, possibly connected by configuration-dependent internal forces (for instance, springs or beam elements)

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

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

The bodies can also be connected by kinematic constraints in form of algebraic equations, resulting in the addition of algebraic variables in form of Lagrange multipliers

$$\boldsymbol{M}(\boldsymbol{x})\,\dot{\boldsymbol{x}} = \boldsymbol{q} \tag{3}$$

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

$$\boldsymbol{\Phi}(\boldsymbol{x}) = 0 \tag{5}$$

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, aerodynamic forces of increasing sophistication ranging from strip theory (with simple inflow models for rotorcraft applications) to state-space representations, to free-wake modeling.

3 REAL-TIME SIMULATION

Real-time simulation capabilities have been obtained by enabling a general-purpose, open source multibody analysis software, MBDyn (`http://www.mbdyn.org/`, Ref. 4), to exploit the real-time utilities offered by the Real-Time Application Interface (RTAI) for the Linux OS (`http://www.rtai.org/`, Ref. 6). All of the above described software is free, which means that it is freely available in source form, thus giving the broadest accessibility to all of its internals. 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 as a bottom line data acquisition, conditioning and visualization, and model and experiment control. Most of these tasks can be directly performed by automatically generating 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. 7).

One fundamental requirement of the present work is that the real-time enabling of the general-purpose multibody software imply minimal impact on the original software and its behavior, unless the changes are

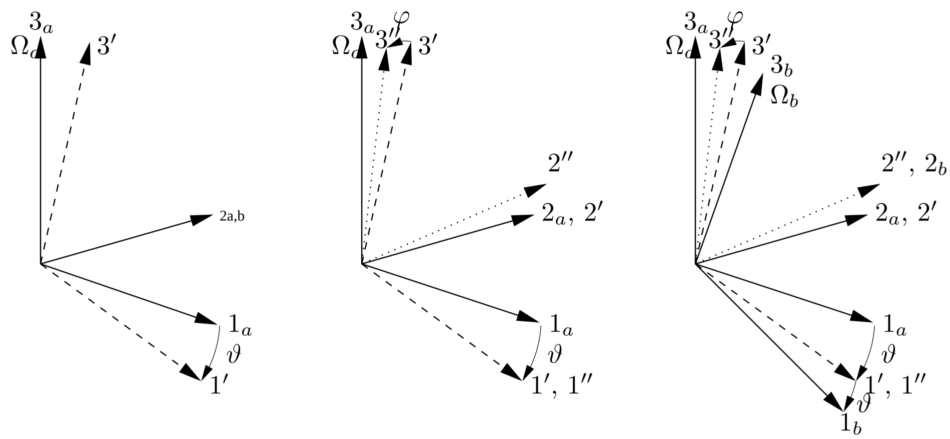

Figure 1: Gimbal relative orientation decomposition; the norm of Ω_b, the angular velocity of body b when angles ϑ and φ are constant, is equal to the norm of Ω_a, the angular velocity of body a.

beneficial also to batch simulations. The process of real-time enabling applies to any software, implying four aspects:

- avoid system calls which return control to the OS and thus cause loss of pre-emption;
- statically preserve enough 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. 8).

4 MODELING ISSUES

Despite the overall efficiency of the simulation software, real-time simulation poses 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 the unnecessary unknowns, by developing appropriate modeling strategies. Sensible gains 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.

4.1 Gimbal Joint

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

An ideal gimbal consists in a sequence of two Cardan joints that undergo half of the relative orientation each, as shown in Figure 1. As a consequence, the second-order perturbations on the axial velocity of the shafts introduced by each joint are cancelled. Its modeling using simple joint elements in MBDyn would require two extra nodes (6 equations each) and 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 indicating with $\boldsymbol{e}_i$ 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 the software 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.

4.2 Friction

Friction can severely impact the ability of a controller to precisely position a robot arm; for this reason, the multibody model used to simulate the control of robotic manipulators accounts for friction and for its most notable effect, stiction. The friction model should be able to reproduce the dependency of the friction coefficient from the relative sliding velocity, i.e. the Stribeck effect. From a computation point of view, real-time simulations require a friction model that does not need discrete state transitions, because state transitions often, if not always, require additional Jacobian assembly and matrix refactorization to be accounted for, thus increasing the likelihood of overruns of the simulator.

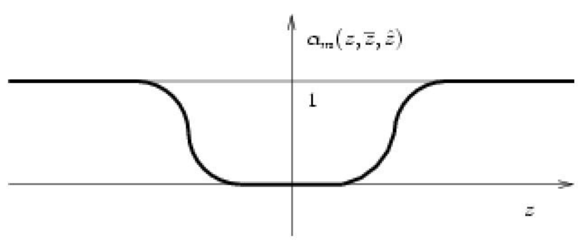

Figure 2: $\alpha(z, \dot{x})$ for $\operatorname{sgn}(\dot{x}) = \operatorname{sgn}(z)$

A whole class of friction models with internal states has been developed in the past years. All these models deal with stiction without needing discrete state transitions. Among them, the modified LuGre friction model proposed in (9) has been selected. In this particular friction model the friction coefficient is a function of an internal state z, of its derivatives and of the relative velocity $\dot{x}$. A differential equation describes the evolution of the internal state z, leading to the law

$$
\begin{aligned}
f &= \sigma_0 z + \sigma_1 \dot{z} + \sigma_2 \dot{x}, \qquad\qquad \sigma_0, \sigma_1, \sigma_2 > 0 \\
\dot{z} &= \dot{x} \left(1 - \alpha(z, \dot{x}) \frac{\sigma_0}{|f_s(\dot{x})|} \frac{\dot{x}}{|\dot{x}|} z \right),
\end{aligned}
\tag{13}
$$

where the function $\alpha(z, \dot{x})$ must satisfy the condition

$$
\alpha(z, \dot{x}) = 0 \qquad \forall z \in \{|z| \leq \overline{z}\}, \quad \forall \dot{x} \in \Re. \tag{14}
$$

A possible expression for α is

$$
\alpha(z, \dot{x}) = \begin{cases} \left. \begin{array}{cc} 0, & |z| \leq \overline{z} \\ \alpha_m(z, \overline{z}, \hat{z}), & |\overline{z}| < |z| < |\hat{z}| \\ 1, & |z| \geq |\hat{z}| \end{array} \right\} & \mathrm{sgn}\,(\dot{x}) = \mathrm{sgn}\,(z) \\ \quad 0, & \mathrm{sgn}\,(\dot{x}) \neq \mathrm{sgn}\,(z) \end{cases} \tag{15}
$$

where $\hat{z} = |f_s(\dot{x})|/\sigma_0$ is the elastic relative displacement at stiction, and the function $\alpha_m(z, \overline{z}, \hat{z})$ is shown in Figure 2. The resulting friction model can reproduce stiction, elastic relative displacements before sliding, the Stribeck effect and the memory effect that has been observed in sliding contacts with velocity variations. This model can be linearized analytically, thus allowing for a fast and robust implicit time integration of the equations of motion even in presence of friction.

5 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. 10, 11, 12, 13).

However, available sparse solvers, although very efficient in terms of memory footprint, are 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, vs. Lapack); however, there is room for improvement, at the expense of memory consumption. A very specialized sparse solver has been implemented for this purpose. Its application to space robot analysis showed overall execution time reductions of roughly 40% compared to using state-of-the-art sparse solvers (Ref. 14, 15).

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. 15), and, as such, will likely become significant for real-time application to that class of problems as soon as hardware and software development will make them affordable.

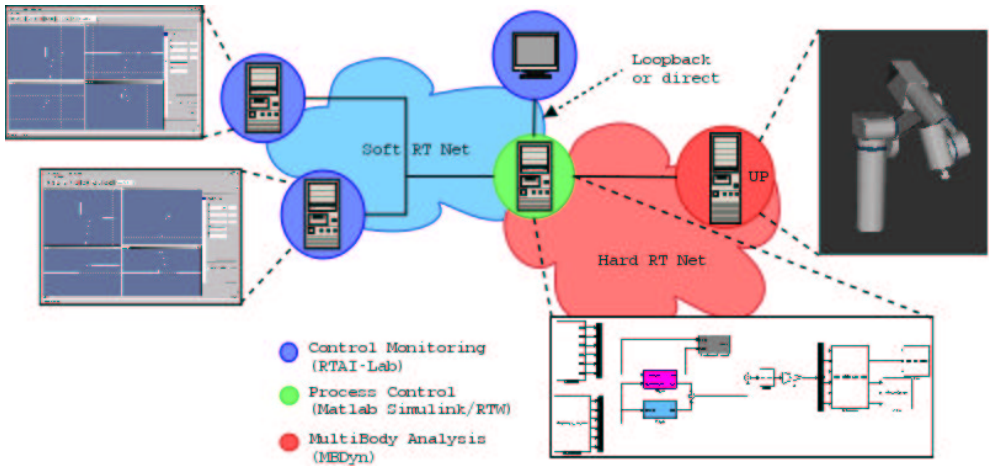

Figure 3: COMAU Smart virtual experiment setup

6 APPLICATIONS

So far, the real-time simulator has been applied in two distinct fields:

- the simulation of wind-tunnel rotorcraft models (Ref. 16);
- the simulation of controlled deformable space robot arms (Refs. 8, 17).

This paper addresses robot applications and selected results from rotorcraft simulation, to highlight the implications of the different fields in setting the requirements for a general purpose software tool.

The presented performance figures have been obtained on an Opteron 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. The performances were relatively good, although not directly comparable to those obtained with the Opteron.

6.1 Simulation of 6 DoF Robot

Figure 3 illustrates the setup for the distributed simulation and control of the COMAU Smart robot. The model represents a six-axis robot; all axes are controlled by step motors. The desired angular position of each joint is provided by the RTAILab management process, and the controller computes the torque that must be applied by the motors. Figure 4 illustrates the typical control interface and input-output signals for a robotic application. A dynamic friction model is used to verify the robustness of the (simple) control strategy with respect to different friction values. The rigid model is described by means of about 120 equations, and runs with a 2 kHz sample rate on the Athlon 2400+ PC. The sample rates obtained with the Opteron 3000+ for robot models with and without friction are reported in Table 1. The last column, marked “Time/s”, indicates the required run time per unit of simulated time; to allow the simulation to be run in real-time, it must be well below one (say, 0.8) to leave some margin for exceptional situations, e.g. when an extra iteration is needed, and also to allow extra processes (control, logging, and so on) to be run simultaneously with a lower priority.

When the deformability of the most slender arms is considered, a larger number of equations is required, and the convergence of the nonlinear iteration loop is challenged; lower sample rates are required to minimize the number of overruns.

6.2 Simulation of Tiltrotor Wind-Tunnel Models

The availability of multibody models 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

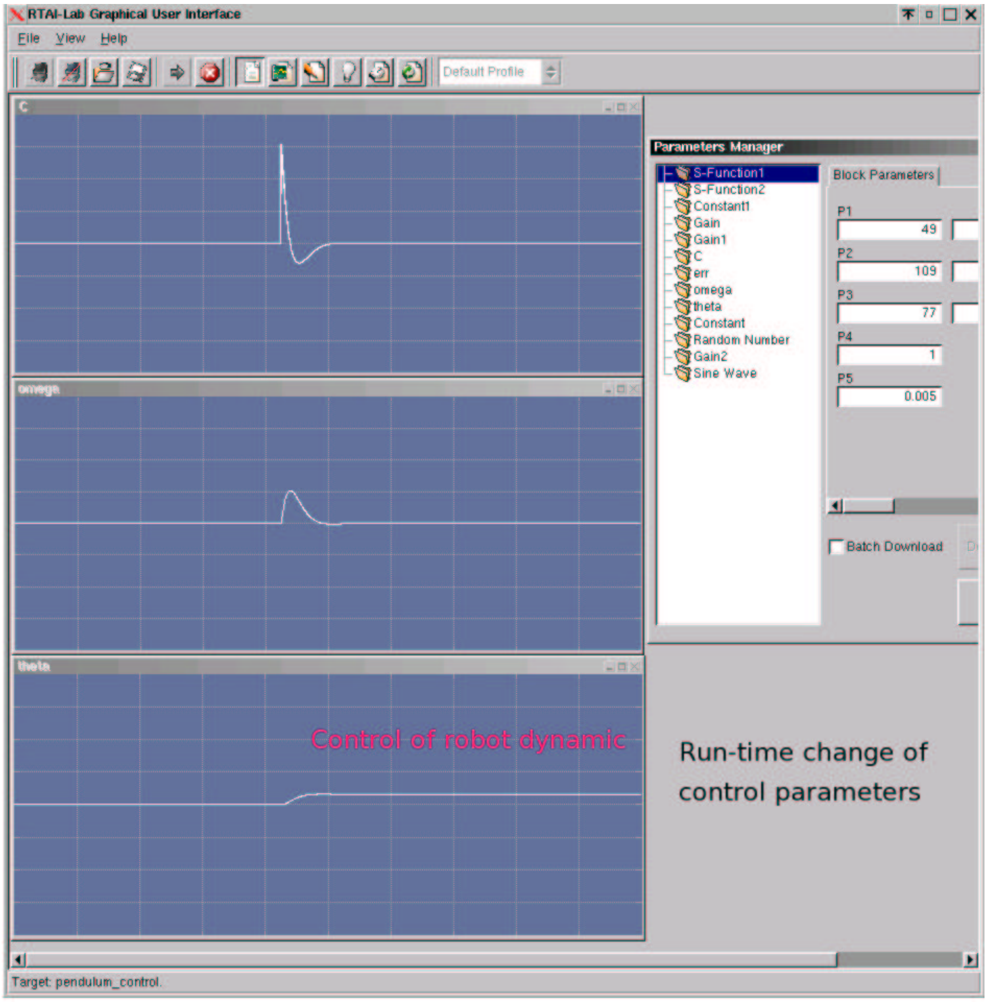

Figure 4: COMAU Smart RTAILab control panel

Table 1: COMAU Smart 6 DoF robot.

	N. Equations	Sample Rate [kHz]	Time/s
		2	0.580
W/o Friction	114	3	0.860
		4	1.145
		1	0.390
W/ Friction	120	2	0.720
		3	1.075

Table 2: Advanced tiltrotor wind-tunnel model at 1 kHz (87 steps/rev)

Wing model	Gimbal element	N. Equations	Time/s
yes	no	159	0.900
no	no	151	0.771
yes	yes	139	0.705
no	yes	131	0.581

and wing, and highly detailed hub kinematics, had to be reduced to very coarse models, essentially with rigid blades and in some cases with simplified hub kinematics.

First, the 1:5 scale model of the V–22 known as the Wing Rotor Aeroelastic Test System (WRATS) in its recent four-blade, soft-inplane configuration has been addressed. Initial results were reported in (Ref. 16), where encouraging results have been obtained in absence of aerodynamic forces, but the yet high computational time required by the simulation did not allow significant simulations.

The above tests have been repeated with the new sparse linear solver on more powerful hardware, resulting in far better performances.

The multibody aeroelastic model of an advanced tiltrotor wind-tunnel model 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 pilots the model during aeroelastic stability test campaigns. This suggested the development of the dedicated ideal gimbal joint element as well as other minor asjustments. Also, a single load path model of the otherwise complicated stiff-inplane hingeless hub has been prepared, to further reduce the number of equations required by the problem and allow its simulation at realistic time steps. Table 2 shows significant timing results obtained with the single load path model in high-speed simulations 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 synthesis approach; the first four modes, resulting in 8 extra equations, are used. Note that the use of the presented gimbal joint element, while saving 20 equations, 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 with 0.940 Time/s at 2 KHz at the nominal forward flight rpm with 113 steps/rev, increasing the rotor speed and the accuracy of the simulation simultaneously.

6.3 Simulation of Helicopters

The real-time simulation of wind-tunnel models suffers from the fact that, while roughly having the same modeling complexity of a real helicopter, they need to spin much faster because of scaling issues, in the range of ($1{:}\sqrt{\text{s.f.}}$) for Froude-scale models to (1:s.f.) for Mach-scale models, with scaling factors ranging from 1:2.5 to 1:7 $\div$ 1:8 and higher.

An essential requirement is the capability to run about 100 time steps per revolution, for the accurate integration of blade dynamics. The previously presented wind-tunnel models appear to 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$\div$400 rpm compared to 800$\div$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.

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,

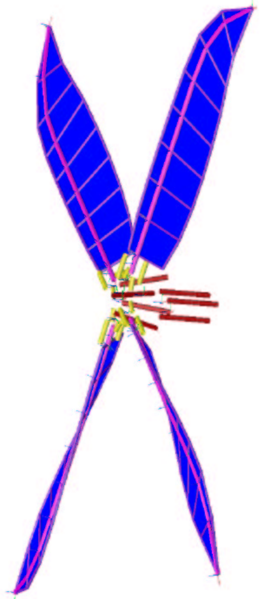

Figure 5: Advanced tiltrotor multibody model.

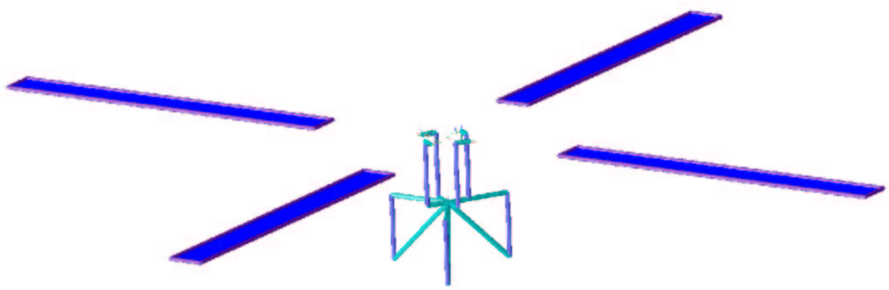

Figure 6: AS330 Puma multibody model.

Table 3: 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

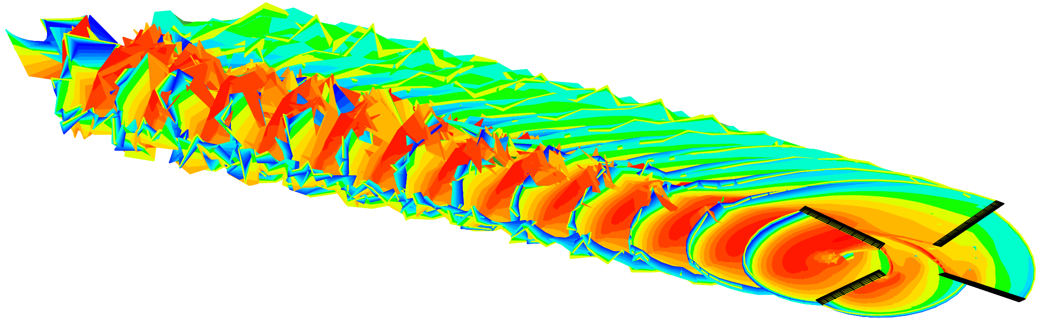

Figure 7: AS330 Puma flight 123 fluid-structure interaction simulation (Ref. 18).

it runs in real-time at 400 Hz with 90 steps per revolution, with a 0.830 time/s ratio. The simulated flight condition is the high speed test flight indicated as "flight 123" (Ref. 19), and reported in Table 3. Figure 7 shows the very same multibody rotor model in a fluid-structure interaction simulation of the same flight condition (Ref. 18).

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 unrealistic.

7 CONCLUDING REMARKS

This work shows that the real-time simulation of complex controlled systems can be performed, with the due simplifications, by means of general-purpose multibody analysis tools. The simulation 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. The proposed "brute force" approach is becoming possible thanks to both the rapid power increase in fairly inexpensive computers, and the refined optimization of the multibody models and solution procedure.

8 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 and RT-Net 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] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] Pierre Dupont, Vincent Hayward, Brian Armstrong, and Friedhelm Altpeter. Single state elastoplastic friction models. *IEEE Transactions on Automatic Control*, 47(5):787–792, May 2002.

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

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

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

[13] 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 8th ESA Workshop on Advanced Space Technologies for Robotics and Automation, ASTRA 2004*, ESTEC Noordwijk, The Netherlands, November 2–4 2004.

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

[15] 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.

[16] 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 4th Decennial Specialists' Conference on Aeromechanics*, Fisherman's Wharf, San Francisco, CA, January 21–23 2004.

[17] 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.

[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 30th *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.