REAL-TIME MULTIBODY ANALYSIS OF WIND-TUNNEL ROTORCRAFT MODELS FOR VIRTUAL EXPERIMENT PURPOSES

Pierangelo Masarati
Assistant Professor
Dipartimento di Ingegneria Aerospaziale
Politecnico di Milano, Italy

Michele Attolico
Ph.D. Fellow
Dipartimento di Ingegneria Aerospaziale
Politecnico di Milano, Italy

Mark W. Nixon
Chief, Loads and Dynamics Division
Vehicle Technology Directorate
Army Research Labs, Hampton, VA

Paolo Mantegazza
Professor
Dipartimento di Ingegneria Aerospaziale
Politecnico di Milano, Italy

ABSTRACT

This work is focused on the implementation of a general multibody analysis framework for the real-time simulation of complex systems. It is applied to the modeling of the Semi-Articulated Soft-InPlane (SASIP) rotor recently installed and tested on the Wing and Rotor Aeroelastic Testing System (WRATS) at NASA Langley. The aim is to obtain a detailed, yet efficient virtual experiment, based on state of the art multibody analysis capabilities and low cost, off the shelf hardware, to predict the behavior of an aeroelastic rotor model and its interaction with a real wind-tunnel experimental setup.

INTRODUCTION

Real-time analysis is a hot topic in multibody research; since the computational efficiency is fundamental to allow hard real-time scheduling of the simulation task, it is usually addressed by exploiting dedicated minimal set models and formulations (1, 2), typically problem specific, e.g. for tree-like structures, which can be oversimplifying, usually allow limited modeling flexibility, in some cases cannot cope with topology changes in the model, and may require costly and error-prone dedicated mathematical and software development. Recent trends in multibody analysis, including real-time issues, focus on model reduction by means of object-oriented symbolic manipulation of problem equations (3, 4). Although very interesting results have been obtained for robotics, and the library of available models is growing at a fast pace together with the multibody capabilities (5). it is still unclear whether such general approaches will be able to deal with structural deformability, unsteady aerodynamic loads and other very specialistic aspects of rotorcraft modeling. Moreover, the recent availability of high-performance hardware at accessible costs is making "brute force" approaches to the real-time simulation feasible, by allowing non-dedicated implementations to be run at adequate sample rates. This work represents an attempt to use a general purpose multibody analysis software and high-performance low-end software to analyize, without any undue limitation, detailed models of a real-life experiment, the Semi-Articulated Soft-InPlane (SASIP) rotor system, recently tested at model scale on the Wing and Rotor Aeroelastic Testing System (WRATS), a 1/5-size aeroelastic wind-tunnel model based on the V-22 (6, 7, 8). The objective is to assess whether a highly efficient, yet general purpose multibody analysis for-

Presented at the AHS 4th Decennial Specialists' Conference on Aeromechanics, Fisherman's Wharf, San Francisco, CA, January 21-23 2004.

mulation, implemented in the open source software MBDyn (9), can be used for realistic hard real-time simulations, for hardware/software-in-the-loop investigations of rotorcraft experiments. The user-space hard real-time capabilities are provided by the Linux Operating System (OS) extension RTAI (10), as discussed in (11). The availability of this software tool could alleviate the burden and the expense of setting up rotorcraft experiments, by allowing realistic experimental setup preparation and testing. Moreover, its unique feature is to allow realistic control systems to be tested without incurring in risks for the experimental setup, as may happen when investigating model stabilization procedures.

The current status of the project entails:

I) the real-time implementation of the multibody analysis;
II) the model specializations that are necessary to meet the very stringent real-time requirements;
III) the validation of the resulting multibody model;
IV) a verification of the possibilities in terms of real-time analysis timings and possible evolution directions.

Future applications of the real-time simulation to the WRATS SASIP experimental setup are foreseen.

The adaptation of the multibody software MBDyn to the real-time application interface RTAI for the Linux OS is discussed first. Several aspects of the multibody modeling for the efficient and accurate analysis of rotorcraft, with special attention to the problems encountered in the analysis of the WRATS SASIP system, are discussed. Conclusions on the feasibility and appropriateness of the proposed simulation setup are finally drawn.

REAL-TIME APPLICATION INTERFACE

The core of the real-time simulation framework is the Real-Time Application Interface (RTAI) (10), an open source package that gives the Linux OS highly versatile real-time capabilities. It has been originally developed at the Dipartimento di Ingegneria Aerospaziale of the University Politecnico di Milano and it is now supported by an international open development effort.

RTAI allows to execute Linux tasks in hard real-time, provided that all the interactions with the operating system occur through its interface. As a consequence, existing software can be easily run in real-time by clearly separating the real-time core from any non real-time part, delegating any interaction to RTAI proper inter-task communication, either in local or distributed mode.

In this sense, coupling MBDyn with RTAI simply requires to (a) carefully avoid conventional system calls, wrapping those explicitly required within RTAI calls; (b) statically preserve enough stack memory for the normal execution, to prevent memory paging or swapping; (c) insert a minimum amount of control statements to initialize the real-time task, force the execution into hard real-time and wait for resume at time step completion. Despite MBDyn being a large program, essentially exploiting most of the advanced Object-Oriented features of C++, the real-time support was successfully added with minimal effort, proving the effectiveness and ease of use provided by RTAI in turning existing software into hard real-time tasks running in user-space.

Using the real-time simulator is as easy as running the real experiment. In fact, as soon as a validated multibody model is available, which fits in the sampling rate time constraints that will be discussed later, one needs to design the sensing (and possibly the controlling) signals, provide for their hooking into the simulation, and prepare the corresponding human-interface devices. The RTAI environment offers a wide range of choices to connect real-time signals to the user, both on the local or on remote machines with no extra effort for the user. Local and remote Inter-Process Communication (UNIX System V IPCs) style mailboxes can be set up transparently to and from remote hosts as well; of course, it is preferable to do soft real-time monitoring remotely on a non dedicated network, while dedicated, fast and deterministic netwroks are required for hard real-time interaction.

The proposed setup uses mailboxes for inter-process communication at the hard real-time level, to send the output measures and, if required, to collect control inputs. Extra measures can also be sent to a lower priority, soft or non real-time proxy for debugging or monitoring purposes, to limit the potential interference with the hard real-time task.

The design of measure and control interaction devices is facilitated by the availability of automatic code generation tools in commercial and open-source software suites like MATLAB™ Real-Time Workshop, SCILAB/SCICOS math environment, using RTAI specific test suprevisor, RTAILab, for online monitoring and parameter tuning (Ref. 12).

SASIP WRATS REAL-TIME MODEL

As a starting point, the multibody model of the WRATS system has been considered. It was presented in (9) and investigated in subsequent works for different tasks, ranging from active load alleviation and flutter suppression to detailed subsystem analysis, including the hydraulic control system. It is currently being updated to model the four-bladed, soft-inplane setup called SASIP, that was experimentally investigated in (8).

Two conflicting requirements must be satisfied by the analytical model of the tiltrotor system to be suitable for reliable real-time simulations: (1) compactness, to allow time-step integration at practical sample rates, and (2) accuracy, to grant reasonably realistic predictions.

The use of general-purpose multibody analysis software in a real-time environment for rotorcraft model-scale testing poses a number of challenges:

1. general purpose multibody software tends to generate large models;
2. solution time is related to model size and model complexity;
3. simplified models, although smaller, usually are less accurate than detailed ones, unless appropriate measures are taken;
4. many solution techniques (e.g. iterative) are very efficient on an average basis only;
5. real-time simulation requires that the worst case be tightly timed;
6. simulation speed can be improved by faster hardware and distributed computation;
7. model scale rotorcraft experiments usually require high rotation speeds;
8. accurate time integration may require finer sample rates than adequate data acquisition;

Each of the above reported points is discussed in the following; some of them are trivial, but become of utmost importance when facing the real-time challenge. Whenever appropriate, actions were taken accordingly to improve the quality of the real-time simulations at hand.

Model Specialization

The multibody model that was prepared in earlier works, and that is being currently upgraded, was characterized by a number of degrees of freedom (DOFs) in the order of the thousands, mostly required by the detailed modeling of the blade and yoke deformability. However, to be able to conduct a real-time simulation with a sample rate of 1 KHz on relatively inexpensive, yet powerful off-the-shelf personal computers (a dual 2.4 GHz Athlon in the present case), a model with slightly less than 200 DOFs is required. To satisfy the requirements, different models of incresing complexity have been investigated.

Wing and Pylon. Basically, the deformability of the wing and of the pylon can be described by means of *modal* models, which represent a practical choice when the low frequency dynamics of deformable components, not subjected to large displacements and rotations, is addressed. This reduces the model complexity from hundreds of DOFs to slightly more than twice the number of selected modes. Since the wing of the WRATS is clamped to the wall of the wind tunnel, there is no need to account for the rigid body displacement and rotation of the modal superelement; the multibody code has been modified to allow this simplification, thus saving 18 DOFs: 12 from the rigid body motion, and 6 from the ground constraint. Moreover, the original WRATS multibody model used a separate body for the pylon, to allow the simulation of conversion maneuvers. Since the wind tunnel model does not allow the conversion DOF, the pylon has been integrated into the wing modal superelement, at the cost of computing a modal superelement for each desired nacelle inclination. This saved at worst 24 DOFs: 6 from an extra superelement interface node, 6 from the constraint of the interface node and 12 from the pylon node when, at worst, it is modeled by a single rigid body.

Hub and Swashplate. Very little could be done at this level, since the swashplate mechanism was already idealized. However, there is some potential for improvements, because to set up the essential kinematics of the control system, and to ease the application of arbitrary control signals, 7 different joints and 6 scalar degrees of freedom, plus 4 extra general purpose elements for connectivity have been used. In case extreme specialization is mandatory, all of this could be integrated into one dedicated element, reducing the computational cost, but violating one of the requirements of this work at the same time: i.e. that no coding not specifically dedicated to the real-time adaptation be required.

Blade Root. The blade root kinematics is characterized by the flap hinge, the lag hinge, and the pitch bearing (in this sequence). The flap hinge and the pitch bearings are modeled as ideal joints; the lag hinge actuates an elastomeric damper. In a typical multibody model, three nodes and three revolute joints are required, accounting for 51 DOFs: 12 for each node and 5 for each revolute joint; in

Table I: Model Figures

Component	Nodes[a]	Joints[b]	Inertia[c]	Beams[d]	DOFs	Notes
Wing/pylon	1	1	0	0	18–22	3–5 normal modes
Hub/swashplate	3	8	1	0	48[e]	
Rigid blade (×4)	2	4	1	0	27	per blade
Deformable blade (×4)	2	0	2	1	24	per extra beam element
Rigid model	12	25	5	0	174–178	
1-beam blade model	20	25	13	4	270–274	

[a]Static nodes use 6 DOFs; dynamic nodes use 12 DOFs. [b]Joints use different number of DOFs. [c]Inertia elements do not use DOFs; modal superelements are counted as joints, although they also contribute to inertia. [d]3-node beam elements. [e]The hub model also includes 6 DOFs for internal states that are used to control the swashplate attitude.

fact, since there is some floating inertia related to each part, dynamic nodes should be used. However, since the three hinges are quite close to eachother, the lead-lag hinge can be considered coincident with the pitch bearing; as a consequence, only two nodes, a revolute and a universal joint can be considered, causing the DOF count to drop to 33: 12 for each node, 5 for the revolute and 4 for the universal joint. Moreover, if the approximation of splitting the mass of the flapping body between the hub and the blade, it can be modeled by means of a static node, thus requiring only 27 DOFs.

Blade. The SASIP model is characterized by articulated rotor blades; as a consequence, a very basic model can be obtained by modeling the blades as rigid, with large savings in computational effort; very coarsely discretized blades were also investigated, to preserve some detail about the deformability of the system.

Remarks. Table I shows some figures about the model size. The resulting DOFs count is about 100 less than the original WRATS SASIP model described in Ref. 13. This in principle does not mean that the original model was poorly coded, because most of the savings have been obtained by simplifications deemed acceptable in view of the overwelming time constraints of the real-time simulation. The saving is about 37% in terms of dof count, corresponding to a 31% saving in execution time. Note that the time saving drops to 28% when full output is requested, due to intensive exploitation of system I/O resources; the execution time nearly doubles. As a consequence, during real-time simulations the amount of simulated measures should be carefully crafted.

A first approximation dof count shows that, without resorting to any undue simplifications, a rigid blade model cannot yet be run at 1 Khz, as opposed to preliminary estimates; a deformable blade model, with one beam element per blade, might require, on current hardware, a higher sampling rate. In fact, it is still debated whether this is acceptable in terms of accuracy, or even feasible in terms of convergence, since 1 Khz results in $70 \div 80$ steps/rev at the typical rotating speeds of the WRATS model ($750 \div 875$ rpm in forward flight and hover, respectively); this will be discussed in more detail in later sections.

Specialized Model Validation

Table II illustrates the first modes of one blade, with root hinges and bearings, rigidly attached to the swashplate, calculated with models of different complexity, and compared to experimental results.

Table III shows a comparison of the fundamental rotating blade modes obtained by the present multibody analysis with different discretization levels, with deformable pitch link. The 5 beam element model has been validated by comparison with experimental results and with results from independent computations (13). The single beam element model has been obtained by simply condensing the degrees of freedom of the previous one; a minimal tuning of the blade properties should improve the agreement in the rotating case at least up to the first torsional mode with a single beam element per blade.

However, the need to perform the integration in real-time imposes a strict constraint on the time step, of the order of few tenths of the rotor fundamental period (60–80 steps/rev), which inhibits the accurate integration of the rotor dynamics in terms of free vibrations. As a consequence, there is no reason to require accurate higher frequency modeling, because the corresponding dynamics would show a substantial phase error and, moreover, they will need to be algorithmically dissipated to stabilize the computation. Note that the frequencies reported in Table III where estimated using an identification procedure (Ref. 15) applied to time series resulting from time integration at very short

Table II: Hub Fixed Blade Modes

Mode No.	Mode Type	Exp.	NASTRAN 25 El.	UMARC/G 25 El.	MBDyn Rigid	MBDyn 1 El.[a]	MBDyn 5 El.[a]
1	F1	0.00	0.00	0.00	0.00	0.00	0.00
2	L1	6.46	6.54	6.43	6.45	6.61	6.32
3	F2	21.70	19.48	20.06	n.a.	18.24	19.37
4	F3	61.15	63.11	64.20	n.a.	42.41	62.43
5	F4	n.a.	91.26	96.21	n.a.	n.a.	88.11
6	T1	107.94	107.44	103.50	n.a.	86.36	106.58

[a]Parabolic three-node finite volume beam elements (14), roughly equivalent, in terms of number of dofs, to twice the elements in UMARC/G.

Table III: Rotating Blade Modes (875 RPM)

Mode No.	Mode Type	MBDyn Rigid[a]	MBDyn 1 El.[b]	MBDyn 5 El.[b]
1	F1	15.13	15.12	15.13
2	L1	10.37	10.32	10.13
3	F2	n.a.	35.23	40.37
4	F3	n.a.	48.18	86.91
5	F4	n.a.	108.37	95.42
6	T1	100.50	87.69	100.30

[a]Pitch link stiffness decreased to about 45% of the nominal value. [b]See note (a) of Table II.

time steps, compared to those that would be used in real-time simulations, to minimize the phase shift occurring in numerical integration.

The forced response does not suffer from the phase error of the numerical integration, in terms of error on the natural frequencies of the system. Of course, the integration of the response to periodic excitation with excessively coarse time step discretization will lead to phase errors in the response as well; for instance, when rotorcraft simulation in forward flight is addressed, the sample rate will be governed by the need to accurately model the response to forcing terms of the order of the n_b/rev harmonic.

As a consequence, the rigid model should be considered a reasonable approximation of the structural model as soon as the fundamental frequencies of the blade are captured *at the desired sampling rate*, e.g. far from the numerical phase shift region.

With respect to aeromechanical behavior, the specialized model suffers from the elimination of the blade elastic couplings, that partially affect some of the aerodynamic performances by means of blade static aeroelasticity. Figures 1 and 2 show how the aeroelastic effect related to the blade elastic axis offset towards the blade nose slightly changes the thrust/collective and the cone/collective curve slope. It is up to the analyst to decide whether or

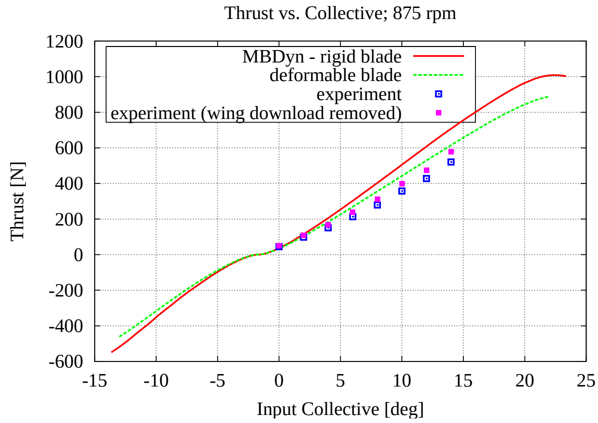

Figure 1: Thrust vs. input collective.

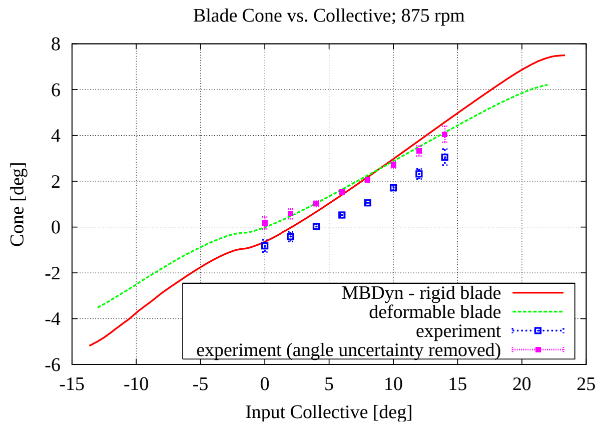

Figure 2: Blade cone vs. input collective.

not these errors are acceptable in view of the scope of the investigation.

Analysis Timings Validation

The need to run the model in real-time imposes two different types of constraints on the analysis: accuracy and efficiency. On the one hand, accuracy is related to the type of model and analysis: a rigid blade model is dominated by fundamentally rigid blade modes, which, in the case considered in this paper, are rigid flap and lag, i.e. about 1/rev. At the nominal hover rotating speed (875 RPM), the rotor fundamental frequency is about 14.6 Hz; for the accurate integration of free vibrations, at least 20 steps/rev are required; unfortunately, since a rotating system is being integrated, there are numerical constraints that require at least some 50–60 steps/rev for stable integration of finite rotations. Deformable blades allow to consider higher blade frequencies and thus a more representative model, but also require finer integration time steps to yield both a stable and accurate integration. For instance, the first torsional mode is at about 100 Hz (about 6.8/rev) thus at least a 2KHz sampling rate would be required to capture this mode within an acceptable accuracy. As a consequence, the sampling rate for a rigid blade model must be in the order of 1 KHz, and higher sampling rates are required if a more detailed dynamic representation is desired.

On the other hand, overall solution efficiency is essentially dictated by:

a) model complexity;
b) hardware performances;
c) computational efficiency;
d) worst-case integration time limitation.

Some of these factors can be partially controlled, some improvements can be expected in a relatively short time, as will be discussed in the following, but there should not be very much expectation for realistic short-term improvements.

Model Complexity. Model complexity has been discussed earlier; its impact on analysis efficiency can be dramatic, so the most essential model has been developed to provide a realistic insight into the potential for real-time simulations within general purpose multibody software.

Hardware Performances. With respect to hardware performances, one of the objects of this work is to obtain a real-time experiment simulator with off-the-shelf, low cost hardware and open-source software, so the fastest Intel/AMD-based available hardware should be considered the maximum expectable target architecture. In the present case, real-time experiments have been performed on a dual Athlon 1.7 GHz machine, so some improvements can be expected by moving to a higher clock CPU, with some caveats on cache exploitation. Dual-processor architecture is particularly suited to performing the desired task, because one CPU will be dedicated to the simulation, while the other can take care of running the OS, the controller, if any, and the Graphic User Interface (GUI) for soft real-time monitoring and basic user interaction. In this case, all the inter-process communication occurs inside the OS, with very limited overhead.

Computational Efficiency. Computational efficiency represents the area where most of the improvements can be obtained. First, the possibility to exploit parallel computing was considered. However, previous experience showed that the message-based parallel paradigm (MPI) currently implemented in MBDyn (Ref. 16) would not guarantee any performance improvement for three reasons:

- network latency and computational overhead in distributed analysis on remote machines;
- computational overhead in SMP (Symmetric Multi-Processor) analysis related to generic implementation for remote parallel analysis;
- excessive overhead of domain decomposition techniques for topologically compact models, resulting in small subdomains and large interfaces, which represent a scalar bottleneck for the parallel solution phase.

However, there is room for performance improvement by exploiting specialized parallel solution techniques for SMP architectures, e.g. performing in multithread environment all the inherently parallel operations related to vector and matrix assembly, taking the appropriate provisions to avoid CPU cache invalidation when concurrently write-accessing structured data. These improvements will need to be especially tailored for the real-time analysis needs, but will be beneficial for generic analysis as well. This is important because improvements resulting from this work must not alter the general applicability of the software, so the fundamental paradigm followed throughout the present investigation, to use general purpose multibody analysis software for real-time simulations can be reversed into providing specialized real-time improvements to general purpose analysis.

Another possibility, to further improve the efficiency of the computation, is to use lower sampling rates together with a higher order integration scheme, that allows to obtain acceptable accuracy with longer time steps. Investigations in this direction are underway, and promising intermediate results have been obtained (Ref. 17). It is shown that time steps 10 to 20 times longer than required by second-order Backward Differentiation Formulas (BDF), a standard integration scheme for DAEs, can be used with the proposed third-order scheme, without incurring in significant phase lag. The implementation of a third-order L-stable integration method in MBDyn has been recently completed, and will be the subject of a future work.

All the above described clues are being scrutinized in view of a possible implementation.

Worst-Case Integration Time. Finally, there is a strong uncertainty about the worst-case integration time. The general purpose multibody analysis that is being considered uses a redundant coordinate approach that requires to integrate a system of Differential-Algebraic Equations (DAE); as a consequence, a L-stable numerical integration technique is required, which, by definition, needs to be implicit and thus requires an iterative solution approach. As a consequence, the solution cannot be obtained in a fixed number of operations; the maximum efficiency is obtained by forcing the solver to generate a single Jacobian matrix, while the number of iterations, and of required residual assembly, is left to the convergence test. The test tolerance, and the way the test itself is performed, must be tuned, based on user's experience, to find the best trade-off between computational accuracy and timing constraints. It has been found experimentally, when dealing with robot controllers, that few failures in meeting the real-time requirement do not necessarily compromise the evolution of the simulation even in case of controlled systems, so in many cases this constraint is not very stringent and admits occasional violations.

NUMERICAL APPLICATION

The complete WRATS SASIP model, represented by one rigid rotor mounted on a deformable semispan wing support, has been run in several configurations, to assess the capability to perform real-time simulations with hardware in the loop. Table IV shows the timing results of some configurations of interest for the simulation setup under investigation. All the presented configurations have been run for a few seconds at the operative condition described in column “RPM”, at the sampling ratio described in column “SR”. Column “steps/rev” reports the number of time steps per revolution, which, as discussed earlier, is representative of the accuracy and the detail one can expect from the simulation. Column “Overruns” indicates what fraction of the total time step number could not be completed on schedule, and column “Time” shows the average duration of the overruns, i.e. the total overrun time divided by the number of overruns, and normalized with respect to the time step:

$$T_{\text{overruns}\%} = \frac{T_{\text{overruns}}}{N_{\text{overruns}} \Delta t}$$

A basic conclusion is that simple models, with less that 100 degrees of freedom can nearly be analyzed in real-time at 1 KHz, since there is a factor of at least $1.6 \div 1.8$ between the fastest CPU in the class considered and the one that was used in this investigation. Complete models *in vacuo* can be analyzed at $200 \div 250$ Hz, while the presence of aerodynamic forces requires even lower sample rates, which are definitely impractical. This is partly a consequence of using explicit aerodynamic forces that do not contribute to the jacobian of the system. On one side, this makes the computation of the jacobian a bit simpler, but on the other side more iterations are required to achieve convergence. To minimize the worst-case integration time, the solver was required to compute exactly one jacobian per time step. The complete system in air has been analyzed in real-time at 200 RPM with a sample rate of 100 Hz, corresponding to 30 steps/rev. Figure 3 shows the flap and lag hinge angles of blade 1 during the simulation of a composite maneuver, consisting in:

- wind-up the rotor, from 0 to 200 RPM (0 to 5 s);
- simultaneously, move the collective from 22 deg (assembly position) to 0 deg (0 to 0.1 s);
- linearly increase the collective at a 1 deg/s ratio (1 to 13 s);
- lineraly decrease the collective at a -0.5 deg/s ratio (14 to 18 s);
- apply a “1 - cos” lateral cyclic control of 6 deg amplitude (20 to 21 s);
- return to zero lateral cyclic (22 to 23 s).

The analysis has been performed in real-time at a sampling rate of 100 Hz, and off-line at a sampling rate of 500 Hz as a cross-check. The resulting time series slightly differ quantitatively, but the results are satisfactorily close from a qualitative point of view; the differences reflect the difficulties in integrating rotating systems with an insufficient number of points per revolution.

Table IV: Simulation Timings

Model Description	DOFs	RPM	S.R.[a]	steps/rev	Overruns[b]	Time[c]
Single blade on hub in vacuo	88	875	1 KHz	68.5	100.0%	15.8%
		875	800 Hz	54.9	2.5%	1.1%
		875	625 Hz	42.9	0.0%	0.0%
in air		875	1 KHz	68.5	100.0%	55.6%
		875	625 Hz	42.9	93.0%	2.6%
		875	500 Hz	34.3	0.5%	1.9%
		400	500 Hz	75.0	0.0%	0.0%
Rotor + Wing in vacuo	174	400	200 Hz	30.0	100.0%	33.0%
		200	160 Hz	48.0	75.3%	11.9%
		200	125 Hz	37.5	0.0%	0.0%
in air		200	160 Hz	48.0	98.3%	21.6%
		200	125 Hz	37.5	2.1%	5.1%
		200	100 Hz	30.0	0.0%	0.0%

[a]Sampling Rate. [b]Fraction of time steps that could not be completed on schedule. [c]Average overrun time, computed on number of overruns and normalized with sampling rate.

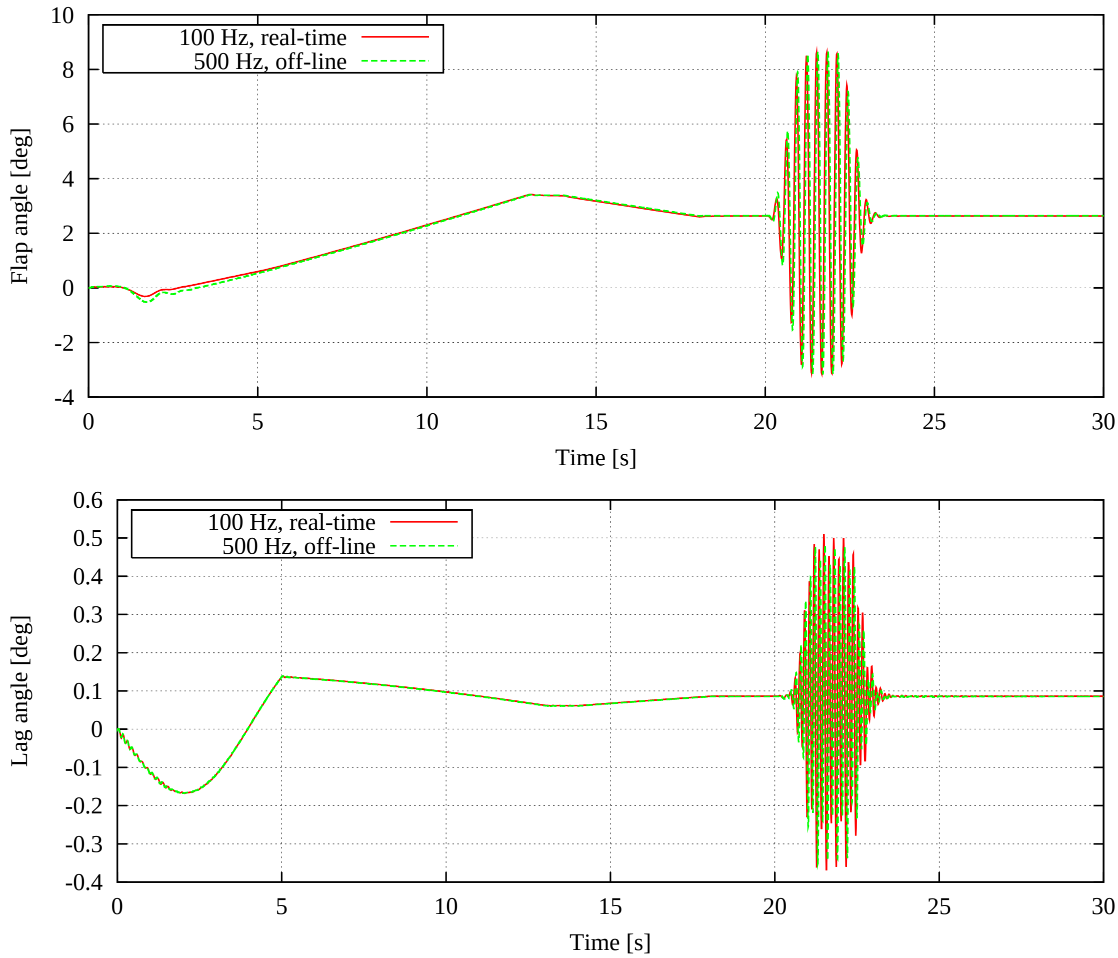

Figure 3: Blade 1 flap and lag angles during transient maneuver.

CONCLUSIONS

The adaptation of a general purpose, open-source multibody/multidisciplinary analysis software to the real-time simulation of rotorcraft systems has been discussed. The investigation has been only partially successful, because the current software and hardware setup does not allow the real-time simulation of appropriate models of the WRATS SASIP system at sampling rates of interest. However, it showed that the current technology is getting very close to this objective, so the following conclusions can be drawn:

1. the adaptation of general purpose analysis software to real-time capabilities is easy and straightforward with the Real-Time Application Interface;

2. preliminary results on the analysis and control of space manipulators showed that the hard real-time simulation of sophisticated multibody models, with less than 100 degrees of freedom, by means of a general purpose multibody software, is possible at practical sampling rates of 1 kHz, typical of rotorcraft model testing;
3. a detailed multibody model of the WRATS tiltrotor in the SASIP configuration is currently available for the MBDyn code; it has been validated and calibrated according to the results of an already concluded experimental campaign, and cross-checked with results from other software of analogous capabilities;
4. the number of degrees of freedom of the model has been reduced, without resorting to any undue simplification, by using a coarser discretization, replacing deformable with rigid components where acceptable, and using a modal superelement instead of a FEM model of the wing;
5. The feasibility of the real-time simulation of the resulting model on currently available hardware has been assessed; it is not yet possible to run the model at the desired sampling rate, because there is still a factor about 2 between the resonably satisfactory sampling rates obtainable and the desired ones; however, taking into account the current WRATS SASIP nominal rotation speed, this means that real-time simulation of full-scale rotorcraft systems can be performed with today's technology.
6. Directions for further improvement of the overall efficiency of the proposed setup have been drawn, which promise to make the real-time simulation of model-scale rotorcraft feasible, and even expandable to more sophisticated models if an almost linear scaling of the computational effort can be achieved by a multithread implementation of the simulation software.

Further development, exploiting more powerful hardware of analogous class, and further refinement of the analysis procedure, will consist in the execution of practical experiments with the simulated system replacing the real model.

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] Martin Otter and Hilding Elmqvist. Languages, libraries, tools, workshop and eu-project realsim. *Simulation News Europe*, pages 3–8, December 2000.

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

[5] 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.

[6] David J. Piatak, Raymond G. Kvaternik, Mark W. Nixon, Chester W. Langston, Jeffrey D. Singleton, Richard L. Bennett, and Ross K. Brown. A wind-tunnel parametric investigation of tiltrotor whirl-flutter stability boundaries. In *American Helicopter Society* 57th *Annual Forum*, Washington, DC, May 9–11 2001.

[7] Mark W. Nixon, Chester W. Langston, Jeffrey D. Singleton, David J. Piatak, Raymond G. Kvaternik Lawrence M. Corso, and Ross Brown. Aeroelastic stability of a soft-inplane gimballed tiltrotor model in hover. In *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Seattle, WA, April 16–19 2001.

[8] Mark W. Nixon, Chester W. Langston, Jeffrey D. Singleton, David J. Piatak, Raymond G. Kvaternik, Lawrence M. Corso, and R. K. Brown. Aeroelastic stability of a four-bladed semi-articulated

soft-inplane tiltrotor model. In *American Helicopter Society* 59^{th} *Annual Forum*, Phoenix, AX, May 6–8 2003.

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

[10] 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.

[11] 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.

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

[13] Pierangelo Masarati, David J. Piatak, Jeffrey D. Singleton, and Paolo Mantegazza. An investigation of soft-inplane tiltrotor aeromechanics using two multibody analyses. In *American Helicopter Society* 4^{th} *Decennial Specialists' Conference on Aeromechanics*, Fisherman's Wharf, San Francisco, CA, January 21–23 2004.

[14] Gian Luca Ghiringhelli, Pierangelo Masarati, and Paolo Mantegazza. A multi-body implementation of finite volume beams. *AIAA Journal*, 38(1):131–138, January 2000.

[15] Giuseppe Quaranta, Pierangelo Masarati, and Paolo Mantegazza. Dynamic characterization and stability of a large size multibody tiltrotor model by pod analysis. In *ASME* 19^{th} *Biennial Conference on Mechanical Vibration and Noise (VIB)*, Chicago Il, September 2–6 2003.

[16] Giuseppe Quaranta, Pierangelo Masarati, and Paolo Mantegazza. Multibody analysis of controlled aeroelastic systems on parallel computers. *Multibody System Dynamics*, 8(1):71–102, 2002.

[17] 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.