REAL-TIME SIMULATION OF COMPLEX SPACE SYSTEMS

P. Masarati[(1)], **A. Fumagalli**[(2)] **and M. Morandini**[(3)]

[(1)] *Politecnico di Milano, Dip. Ing. Aerosp., v. La Masa 34, 20156 Milano, Italy; pierangelo.masarati@polimi.it*
[(2)] *Altran Italia, Aerosp. & Defense Div., C.so Sempione 66, 20154 Milano, Italy; alessandro.fumagalli@altran.it*
[(3)] *Politecnico di Milano, Dip. Ing. Aerosp., v. La Masa 34, 20156 Milano, Italy; marco.morandini@polimi.it*

INTRODUCTION

This paper presents a framework for the modeling and simulation of the dynamics of complex deformable space systems. It is based on a suite of free software, mostly developed and distributed by the 'Dipartimento di Ingegneria Aerospaziale' of Politecnico di Milano (DIA-PM), that allows the real-time simulation of arbitrarily complex mechanical, mechatronic and aeroservoelastic systems. Real-time simulation is key to efficient, safe and cost-effective development of critical systems, whose testing is either virtually impossible, unsafe, or very expensive. The capability to test the functionality and the performance of critical components, like control systems, without using the real hardware allows Hardware-In-the-Loop (HIL) virtual experimenting. The modeling of the physics of the problem is delegated to the free general-purpose multibody dynamics software MBDyn ([1], `http://www.mbdyn.org/`). It integrates the equations of motion of constrained mechanical systems as Differential-Algebraic Equations (DAE), using implicit schemes. The free Real-Time Application Interface (RTAI, `http://www.rtai.org/`) takes care of scheduling the hard real-time execution of all involved tasks: the multibody simulation of the problem, process control, and monitoring. Control design is delegated to graphical modeling environments: Scilab/Scicos, Matlab/Simulink or MatrixX. Only the first is free software. Applications of the proposed framework to the dynamic analysis of typical space systems are presented: open- and closed-loop controlled rigid and flexible manipulators, parallel robots, and deployment of large deformable structures [2, 3, 4].

MULTIBODY FORMULATION

The proposed formulation is used to solve multisciplinary Initial Value Problems (IVP) with a monolithic approach. The mechanics of constrained systems are modeled using Lagrange's equations of the first kind for each Finite Element (FE)-like node that describes the motion of the bodies,

$$\mathbf{M}\dot{\mathbf{x}} = \mathbf{p} \tag{1a}$$

$$\dot{\mathbf{p}} + \boldsymbol{\phi}^T_{/\mathbf{x}}\boldsymbol{\lambda} = \mathbf{f}\left(\mathbf{x}, \dot{\mathbf{x}}, t\right) \tag{1b}$$

$$\boldsymbol{\phi}\left(\mathbf{x}, t\right) = \mathbf{0} \tag{1c}$$

with the nodal coordinates $\mathbf{x}$, including relative orientation parameters, momentum and momenta moments $\mathbf{p}$, inertia matrix $\mathbf{M}$, nodal forces $\mathbf{f}$, algebraic constraint equations $\boldsymbol{\phi}$, Lagrange's multipliers $\boldsymbol{\lambda}$. Arbitrary rotations are dealt with using the orientation matrix $\mathbf{R}$ according to an updated Lagrangian approach; the predicted orientation is taken as reference. The orientation correction is parametrized with Cayley-Gibbs-Rodrigues parameters. This prevents singularities in relative rotations while allowing to use a vectorial parametrization of rotations. Structural deformability is included in nodal forces $\mathbf{f}$. Geometrically exact, composite ready nonlinear FE beam elements allow to directly model slender structures [5]. Small amplitude flexible motion about the arbitrary rigid-body motion of complex structural components is modeled using the Component Mode Synthesis (CMS) approach.

REAL-TIME SCHEDULING AND CONTROL DESIGN

The execution of the multibody simulation can be scheduled in hard real-time exploiting the support provided by RTAI, thus allowing Hardware- and Man-In-the-Loop (H/MIL) simulations. Real-time visual interaction with the simulated model can be accomplished using an ad-hoc interface to Blender. This interface is developed under the separate project 'Blender and MBDyn' (`http://sourceforge.net/projects/blenderandmbdyn/`).

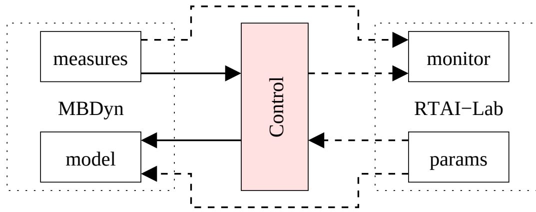

Figure 1: Control scheme layout: hard (—) & soft (- -) real-time messages.

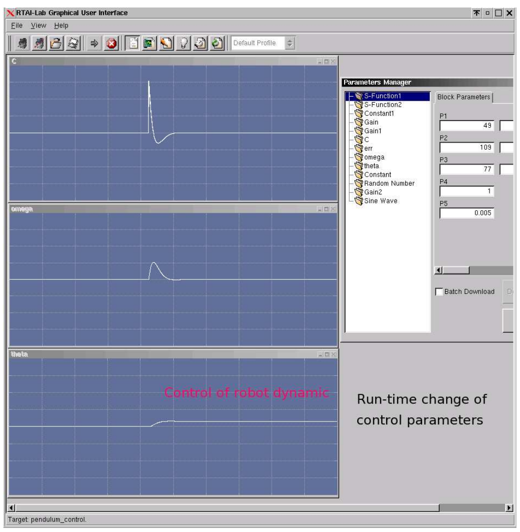

Figure 2: RTAI-Lab GUI.

A very efficient linear solver, specialized for small sparse problems [6], allows the real-time simulation and control of complex models at relatively high frequencies (up to 2 kHz) [7]. Monitoring is delegated to RTAI-Lab [8], a soft real-time graphical interface that visualizes the output of controller and simulated process. Figure 2 presents the Graphical User Interface (GUI) provided by RTAI-Lab. It can be customized to fit the needs of specific problems. The interface can receive soft real-time messages from the controller and the simulator tasks. It also allows to send soft real-time messages to both processes, e.g. to manually tune controller parameters during hard real-time execution. Implicit numerical integration requires the iterative solution of a nonlinear algebraic problem. Consequently, the computational cost of a time step is not bounded. Special care had to be paid to comply with real-time requirement; extreme efficiency of the linear- and nonlinear solvers, and fault-tolerant convergence strategies have been pursued [7, 3].

The design of the control system is performed using generic graphical tools like Scicos, Simulink or MatrixX. These tools allow to automatically generate the source code (in C) of the controller, using RTAI's primitives for IPC. The resulting controller task can be directly used to control the actual hardware, or the simulated process provided by MBDyn. Figure 1 shows a typical control scheme layout as presented by graphical control design tools. The figure highlights the role played by the RTAI-Lab monitoring interface and the simulator process.

APPLICATIONS

Dynamics of Slender Robotic Arm

A very flexible robotic arm testbed, discussed in [9], has been modeled using MBDyn's CMS model, as in the original work, and MBDyn's nonlinear beam model. It consists in a very slender uniform beam, with the end

Mode	**[9]**	**CMS**	**Nonlinear beam**	
			10 nodes	20 nodes
#1	5.28	5.38	5.27	5.28
#2	20.9	20.9	20.7	20.8
#3	46.7	46.9	46.0	46.6
#4	83.4	83.6	80.8	82.4

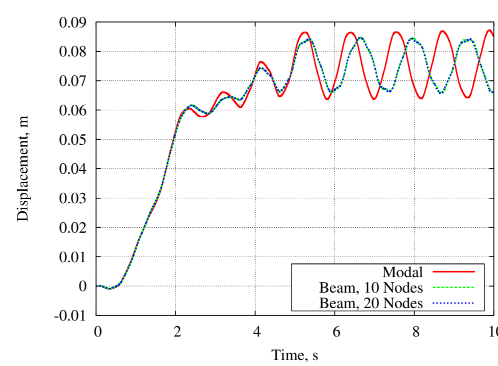

Figure 3: Frequencies (in Hz) and tip motion of space robotic arm testbed (data from [9])

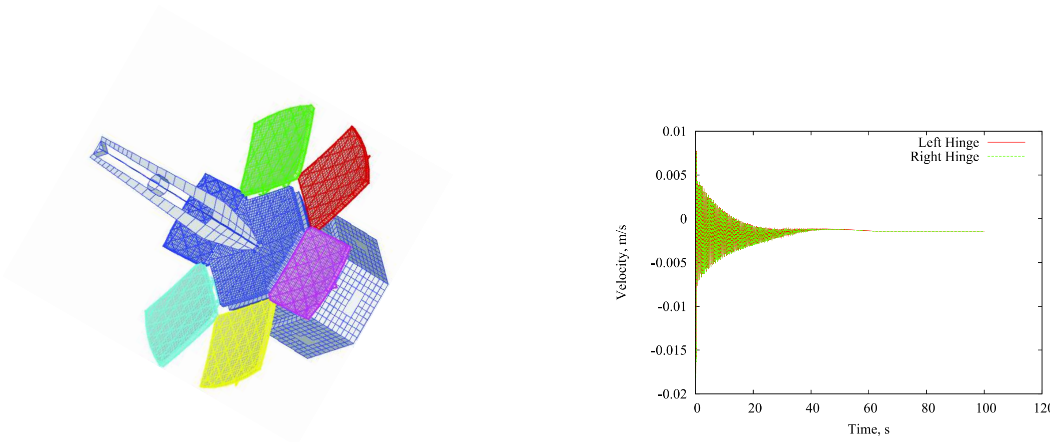

Figure 4: Orbital telescope: sketch (left); deployment transient vibrations at a petal hinge (right).

pinned to the ground actuated by a motor, while the free one carries a large mass. Figure 3 illustrates a fairly good correlation in terms of natural frequencies.

It also illustrates the motion of the free tip in response to a sequence of step torque inputs. Results of the CMS model are consistent with the corresponding one presented in [9]. The FE model appears to respond with a first fundamental period longer than the one of the CMS model. This is related to a significant axial loading of the arm resulting from the vibration induced by the motion. The axial load alters the bending dynamics, highlighting the importance to consider geometrical nonlinearities when vibrations of slender beams need to be modeled.

Unfolding of Satellite Antennas/Solar Arrays

The generality of the CMS element allows to model the kinematics and dynamics of arbitrarily complex systems. Figure 4 illustrates the unfolding of the petals of a orbital telescope. The CMS model is built from a FE model of the whole satellite. The analysis captures the vibrations resulting from the unfolding maneuver, the corresponding accelerations and the loads at the hinges and on the deformable mirror. The very same model, interfaced with Simulink, is used to assess possible interactions between the active mirror control system and the satellite attitude control.

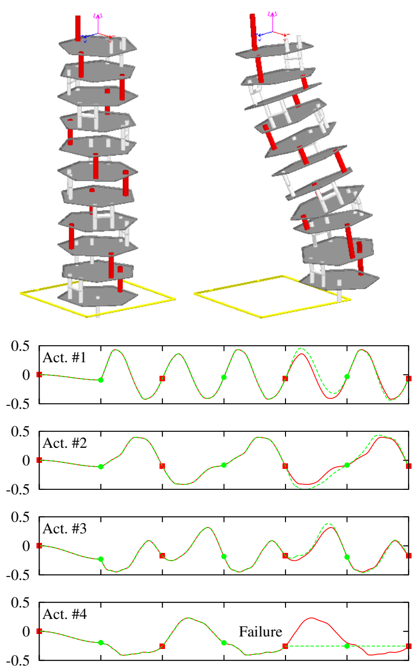

Figure 5: Bio-mimetic 11 DoF robot (VILAS Italia): real-time motion planning.

Path Planning, Inverse Kinematics and Model-Based Control of Robots with MBDyn

Robotic simulation and model-based robot manipulator control requires to solve related problems: a) inverse kinematics, to compute joint motion based on the desired end-effector motion; b) inverse dynamics, to compute motor torques; c) HIL simulation, to assess the feedback control required to compensate model errors and disturbances. Inverse kinematics (a) is straightforward when the number of joint coordinates is equal to the number of constraints on the end-effector motion, while it is not when the robot is under- or over-actuated. In the latter case, a procedure based on weighted least-squares has been developed to determine the motor motion [2]. The procedure uses the multibody solver to statically compute motor motion. For typical problems it operates in real-time. Its output is fed to the inverse dynamics solver. The inverse dynamics solver (b) computes the motor torques required to track the motion designed in (a) [3]. HIL simulation (c) directly integrates the motion of the controlled system subjected to feedforward torques (b) and to feedback torques required to compensate motion errors. In typical HIL analysis of robots, the three tasks are performed by a cascade of separate processes running MBDyn. The desired motion resulting from inverse kinematics is fed to the inverse dynamics and to the HIL simulator. The latter also receives the torques computed by the inverse dynamics. Figure 5 shows an 11 degrees of freedom bio-inspired robot, whose motors control the angle betweed adjacent plates. Each plate rotates with respect to the previous one about a hinge that is skewed with respect to the previous module. The multibody model consists of more than 200 equations. HIL simulations of the system have been performed in real-time at sample rates of practical interest. The over-actuation is exploited to design fail-safe trajectories on the fly, taking into account hardware failures. Figure 5 illustrates how the motion planning changes when a failure is detected in motor #4. An interesting generalization to the control of non-collocated systems, where actuation torques do not directly do work for the imposed motion, has been proposed [10].

It is important to highlight that all the algorithms/computations presented in this paper have been performed using MBDyn, which combine the generality of a multibody software with these specific, problem-related, features. The availability of the source code allows to easily customize the software at low cost, to account for even more specific applications required by the user.

CONCLUSIONS

A framework for the real-time simulation of arbitrarily complex space systems, based on general-purpose free multibody system dynamics software, has been presented. The software natively runs under the free RTOS RTAI and, exploiting its real-time features, can be used as a powerful tool not only for physical simulation but also for the development, and eventually the testing, of control systems. Its application to realistic examples has been illustrated.

The simulation of the space telescope was partially sponsored by the INTERREG IIIC EC MATEO project. The simulations were performed by Giuseppe Quaranta. The satellite configuration and its FE model were kindly provided by Carlo Gavazzi Space S.p.A. and A.D.S. International S.r.l., whose support is gratefully acknowledged.

REFERENCES

[1] Paolo Mantegazza, Pierangelo Masarati, Marco Morandini, and Giuseppe Quaranta. Computational and design aspects in multibody software development. In J. C. Garcia Orden, J. M. Goicolea, and J. Cuadrado, editors, *Multibody Dynamics*, volume 4, pages 137–158. Springer, 2007. doi:10.1007/978-1-4020-5684-0_7.

[2] A. Fumagalli, G. Gaias, and P. Masarati. A simple approach to kinematic inversion of redundant mechanisms. In *ASME IDETC/CIE 2007*, Las Vegas, Nevada, 4–7 September 2007. (DETC2007-35285).

[3] A. Fumagalli and P. Masarati. Real-time computed torque control using general-purpose multibody software. *Multibody System Dynamics*, 22(1):47–68, 2009. doi:10.1007/s11044-009-9153-7.

[4] A. Fumagalli, P. Masarati, M. Morandini, and P. Mantegazza. Control constraint realization for multibody systems. In *ASME IDETC/CIE 2009*, San Diego, USA, August 30–September 2 2009. (DETC2009-86949).

[5] Gian Luca Ghiringhelli, Pierangelo Masarati, and Paolo Mantegazza. A multi-body implementation of finite volume beams. *AIAA Journal*, 38(1):131–138, January 2000. doi:10.2514/2.933.

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

[7] 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.

[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] John R. MacLean, An Huynh, and Leslie J. Quiocho. Investigation of boundary conditions for flexible multibody spacecraft dynamics. In *IDETC/CIE 2007*, Las Vegas, USA, September 4–7 2007.

[10] A. Fumagalli, P. Masarati, M. Morandini, and P. Mantegazza. Control constraint realization for multibody systems. *J. of Computational and Nonlinear Dynamics.* To be published.