DYNAMIC STABILITY OF SOFT-IN-PLANE TILTROTORS BY PARALLEL MULTIBODY ANALYSIS

Giuseppe Quaranta, Pierangelo Masarati [†], Massimiliano Lanz,
Gian Luca Ghiringhelli, and Paolo Mantegazza,
Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano
and
Mark W. Nixon,
Army Research Laboratory, NASA Langley Research Center

Abstract

The paper presents the parallel implementation of a multibody formulation for the analysis of the dynamic stability of a soft-in-plane tiltrotor configuration, with regard to the ground- and air-resonance phenomena. A multibody model of a tiltrotor semispan wind tunnel model has been used to investigate the phenomenon. An original multibody, multidisciplinary formulation has been used; its implementation in a coarse scale parallel environment, both on SMP and PC cluster, is proposed as a means to speed up massive parametric analyses. The dynamic stability of the rotorcraft is investigated to determine the parametric influence of various structural components, such as the damping properties of the blade root.

1 Introduction

The need for medium/heavy-weight tiltorotor aircraft is emphasizing the limitations in rotor design related to the conventional stiff-in-plane architecture currently used in tiltrotor design [1]. A soft-in-plane rotor design results in a lighter rotor, owing to the reduced in-plane bending stresses in the rotor blades and in the hub, and thus may lead to a higher payload or range.

The main problem related to such architecture is represented by the intrinsically soft support the rotor is attached to, namely the wing of the tiltrotor aircraft; since the main wing frequencies of a large scale tiltrotor are likely to be lower than the nominal rotation frequency of the rotor both on the ground and in flight, the aircraft is potentially subject to dynamic instability in every flight condition. Moreover, while it is known that the dynamic instability of a soft-in-plane rotorcraft is cured by a combination of mechanical damping both in the rotor blade hinges and in the support [2], in a tiltrotor the latter is supplied only by the wing structural damping, which is intrinsically limited and can hardly be influenced by the design.

An original multibody formulation, resulting in a code named MBDyn, has been used to model the tiltrotor [3]. This approach allowed to achieve a high level of detail, and significantly to avoid undue simplifications in modeling the kinematics of the system, which are intrinsically nonlinear.

This kind of analysis, when realistic problems are addressed, may result in comparatively large, nonlinear models. A thorough study of the stability of these systems requires long simulation times, both because the instability shows up at relatively low frequencies and because the rotor needs a realistic wind-up simulation, since a trimmed steady solution cannot be directly computed. To overcome this problem, the multibody formulation has been implemented in a parallel scheme based on a coarse scale parallelization with the Schur decomposition method. This solution allows to exploit the peculiar topology of the multibody rotorcraft system, as detailed later, which allows reduced interfaces between loosely connected subsystems. As a result, a dramatic speed-up of the computations can easily be obtained, thus allowing to carry out sensitivity analyses in a reasonable time.

2 Multibody Formulation

The proposed multibody formulation is based on the direct writing of the equilibrium equations

[†]Corresponding Author
via La Masa, 34 20156 Milano, Italy
Tel.: +39(02)23998365
Fax: +39(02)23998334
E-mail: *masarati@aero.polimi.it*

Presented at the 26th European Rotorcraft Forum, 26–29 September 2000, The Hague, The Netherlands

of the independent bodies, which are then connected by algebraic and/or flexible constraints. The former lead to a Differential-Algebraic Equations problem of index 3, whose solution requires special care from a computational point of view because of the algebraic unknowns resulting from the constraint equations, which are imposed in a Lagrangian Multipliers form.

2.1 Dynamics Equations

The dynamics of the bodies are written in a first-order differential scheme by considering the definitions of the momentum and of the momenta moment of each body:

$$\begin{aligned} \beta &= m\dot{x} + S \times^T \omega, \\ \gamma &= S \times \dot{x} + J\omega, \end{aligned}$$

where $\dot{x}$ and ω are the translational and angular velocities of a point related to an inertial body of mass m and of first- and second-order inertia moments S and J, both expressed in an absolute orientation frame. The derivatives of the momentum and of the momenta moment participate in the dynamics equations of the body as the inertial contribution; the equations are

$$\begin{aligned} \dot{\beta} &= F\left(x, \dot{x}, R, \omega, a, \dot{a}, \ldots\right), \\ \dot{\gamma} + \dot{x} \times \beta &= C\left(x, \dot{x}, R, \omega, a, \dot{a}, \ldots\right), \end{aligned}$$

where the forces and couples, F and C, may depend on the state and on other parameters: x and R represent the position and the orientation of the frame the momenta are referred to, and the unknowns a are internal states that can be associated to multidisciplinary fields to be simultaneously considered. The formulation has been extended to include electric and hydraulic subsystems, which are modeled in the spirit of the multibody approach, that is by assembling elementary components to build an entire system. These components are used to implement the control system of the rotorcraft, ranging from the swashplate actuation system to a sophisticated active control system based on the Generalized Predictive Control concept [4].

2.2 Algebraic Constraints

Constraints are added in form of algebraic equations depending on the configuration of the system. Holonomic constraints involve the position and the orientation of the bodies, and result in DAEs of index 3, while non-holonomic constraints involve the time derivatives of the configuration as well, leading to index 2 DAEs:

$$\begin{aligned} \Phi_h\left(x, R, \ldots\right) &= 0, \\ \Phi_{nh}\left(x, \dot{x}, R, \omega, \ldots\right) &= 0. \end{aligned}$$

The constraint equations are added in a Lagrangian Multipliers form, which implies that the corresponding multipliers apply to the dynamics equations as reaction forces and couples.

2.3 Flexibility

The forces and couples that act on the bodies in the dynamics equations may depend on the configuration; this is the case of the elastic forces. Elasticity has been accounted for in a lumped scheme. Lumped elastic elements such as rods and springs are used to model basic flexible elements such as the pitch links. More sophisticated flexible elements have been implemented in form of beams; an original finite volume scheme has been adopted [5]. It consists in directly writing the equilibrium of the finite portions of beam obtained by cutting an element at appropriate points between the nodes. The internal forces and couples at such "evaluation points" are expressed as functions of the generalized strain and curvature of the beam by means of an arbitrary constitutive law. The strains in turn are written as functions of the nodal configurations, which are interpolated by parabolic shape functions. The equilibrium equations result in

$$\begin{aligned} &\mathcal{U}\left(p_b - x_0\right)^T \vartheta_b - \mathcal{U}\left(p_a - x_0\right)^T \vartheta_a \\ &\quad = \int_a^b \mathcal{U}\left(p\left(\xi\right) - x_0\right)^T \tau \, dp, \end{aligned} \tag{1}$$

where p is the reference line that defines the beam, x_0 is the reference pole the moments are referred to, the ϑ are the internal forces and couples at the extremities of the beam portion and τ are the distributed forces and couples, while the matrix $\mathcal{U}\left(p\right)$ is defined as

$$\mathcal{U}\left(p\right) = \begin{bmatrix} I & p \times^T \\ 0 & I \end{bmatrix}.$$

A three-node beam element has been implemented, so Equation 1 is applied to the domains $\left[p\left(-1\right), p\left(-1/\sqrt{3}\right)\right]$, $\left[p\left(-1/\sqrt{3}\right), p\left(1/\sqrt{3}\right)\right]$, $\left[p\left(1/\sqrt{3}\right), p\left(1\right)\right]$ with $p = p\left(\xi\right)$, being $\xi \in \left[-1, 1\right]$ a nondimensional abscissa along the beam element.

3 Parallel Solver

To obtain an efficient distribution of the computational load, it is mandatory to spread on the different CPUs both the assembly phase of the Jacobian matrix and of the residual array, and possibly the linear solution phase. To exploit the topological properties that the multibody model computational domains under investigation have, a technique belonging to the class of non overlapping domain decomposition has been chosen [6, 7]. Among these, the substructuring method, that has been traditionally used also in structural analysis, has been considered. It basically requires the generation of s disjointed subdomains, so that the original system is reduced to a smaller interface that becomes the core problem. Connecting a small number of off-the-shelf personal computers by fast communication networks is increasingly gaining acceptance as a low cost and effective alternative to large supercomputers [8]. Therefore, special attention has been dedicated to produce a code that can efficiently run on this kind of platform.

3.1 Schur Complement Technique

The computational domain Ω is first split into the s subdomains Ω_i via an element-based partition. This means that no element is shared by two subdomains, i.e. all the information related to a given element is mapped on the same processor. As a result, there is no need for information exchange while the assembly phases are performed. The unknowns are reordered, labeling the interface nodes last. The linear system associated with the problem has the following structure:

$$\begin{pmatrix} B & E \\ F & C \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{2}$$

where the x array is composed of a succession of the local subdomain Ω_i internal unknowns x_i, and y is the vector of the interface unknowns. Assuming that matrix B is non-singular, the unknown x can be expressed as

$$x = B^{-1}(f - Ey). \tag{3}$$

By substituting Equation 3 into the second block-row of equation (2), the following reduced system is obtained:

$$Sy = g - FB^{-1}f. \tag{4}$$

where the matrix S is called the *Schur complement matrix*; it takes the form:

$$S = C - FB^{-1}E. \tag{5}$$

A solution method based on this approach involves five steps:

1. The local matrices B_i are factored.
2. The local parts of the right-hand side of the reduced system (4) are assembled and transmitted to a "master" processor that will deal with the interface problem.
3. The local parts of the Schur complement matrix are assembled and transmitted as well.
4. The reduced system is solved.
5. The other unknowns are obtained by back-substitution using Eq. (3).

Only the 4^{th} step cannot be performed concurrently in a parallel environment, so it must be considered the bottleneck phase. Furthermore, when a modified Newton-Raphson method is used to solve the nonlinearity, since the Jacobian matrix is not updated, there is no need for steps (1) and (3), and many of the operations required during the assembly phase (2) do not need to be performed either. Usually, the direct substructuring method described here is not considered feasible for large structural problems, because the dimension of the interface problem grows very rapidly; moreover, the Schur matrix presents a lower grade of sparsity than the original system, so an iterative inner solver may be built, which does not require the explicit assembly of matrix S. On the contrary, this strategy can be very effective when some special conditions are met. This is the case of complex systems with a peculiar topology of the computational domain that allows the generation of a partition with very small interfaces [9]. Many common mechanical problems show a topology that fits such a requirement; a clear example is represented by those described later in the results section. It is not easy to correctly define a dimension for the computational domain that is analyzed with a multibody multidisciplinar simulator. The multidisciplinarity requirement obviously does not allow any structure in the domain, because structurable, i.e. related to the physical space, and non-structurable, i.e. abstract, unknowns coexist in the solution space. Anyway the domain is usually quasi-monodimensional, with some multiple paths or close-circuits. This is basically true because the underlying structure is usually made of rigid or flexible bodies connected by algebraic constraints, that represent non-reducible topological items of the computational domain. Owing to these characteristics, the computational grid can be subdivided into parts with an optimal ratio between internal and interface nodes. Clearly, the search for a minimal-interface partition is crucial, so this task cannot be performed manually, demanding an automatic partition tool as described later.

3.2 Induced Velocity Element

The rotor element performs a specific task: it computes the velocity that the rotor disk induces on each aerodynamic body based on some global model, e.g. momentum theory or dynamic inflow [10], which requires the knowledge of the overall forces and moments generated by the rotor. For this purpose, a two-way communication between the rotor element and all the related aerodynamic elements is required. As a consequence, another communication channel must be implemented, which differs from the one related to the exchange of information on the interface problem. It transmits the force contribution from the aerodynamic elements to the rotor element, and the induced velocity from the rotor to the aerodynamic elements during the assembly steps. These communications may result extremely inefficient, so non-blocking communications are used in the attempt to minimize their influence on the overall simulation time [8]. The assembly of the rotor elements is scheduled far from that of the aerodynamic ones, while the information transmission is performed as soon as possible, and its completion is checked only when strictly required.

3.3 Partitioning Strategies

A direct substructuring method like the one described in the previous section will be really effective only when coupled to an automated tool which identifies the best subdivision of the computational domain. The partition should meet two goals: 1) minimize the ratio between communications and computations, and 2) balance the computational load among the CPUs. The former requirement mainly results in searching for the subdivision with the minimal interface size. The latter is rather related to the assembly steps and, considering the wide assortment of different elements implemented in MBDyn, it involves an estimate of the number of operations required by each type of element. Graph partitioning problems are NP-complete [11], therefore efficient heuristic methods must be used to obtain satisfactory solutions in a reasonable time. A good tradeoff between quality and cost in terms of time is offered by multilevel partitioning methods, that do not require a geometry associated with the graph of the domain connections. For the purpose of this paper, the methods implemented in the METIS library proved very effective [12]. The problem is stated in terms of searching the partition that produces s disjoint subsets such that the sum of the vertex weights in each subset is the same, and the sum of the (possibly weighted) edges, whose incident vertices belong to different subsets, is minimized. Since all partitioning routines in METIS allow to specify multiple sets of weights, a high level of flexibility is achieved. To obtain an element-based partition, a connection graph is built, in which both nodes and elements are considered vertices; in this way, the two main objectives stated earlier can easily be taken into account. First of all only the elements contribute to the computational load balance, so each is assigned a computational weight, proportional to the elaboration time. Nodes are responsible for the communication time but, as they can be associated to a different number of degrees of freedom (there are scalar nodes, static structural nodes with 6 DOFs, dynamic structural nodes with 12 DOFs and so on), each is assigned a different communication weight. This emphasizes the unstructured nature of the computational domain, where only the presence of an element determines the connectivity of the nodes. For those elements that have internal states, the related unknowns are usually assigned to the local subdomain, unless an algebraic constraint is being considered, as explained in the following paragraph.

3.4 Algebraic Constraints

Kinematic constraints require special treatment during the partition generation phase in order to obtain a consistent subdivision. If the computational domain is cut between two nodes connected by a kinematic constraint, the internal unknowns related to the joint element must be positioned on the interface, otherwise a local singularity problem arises. If the unknowns that represent the reaction forces are assigned to a local subdomain, the local subproblem will be overconstrained because the node connected by the constraint has moved to the interface problem. A statically overdetermined problem results, so even if the global problem is consistently formulated, a local singularity will drive the parallel algorithm singular. To solve the problem, the unknowns related to the reaction forces must be moved to the interface. The local problem will now be underconstrained, in the sense that it will behave as a free subsystem, loaded by the reaction forces that come from the algebraic constraints. The back-substitution phase will restore the effect of the kinematic constraint on the local solution. Clearly, if the constraint element is connected to a static node, i.e. a node with no inertia, the latter must be replaced by a dynamic one, characterized by a very small inertia that prevents the system from becoming singular. This may be considered the only additional cost required by the procedure.

4 Numerical Results

The work is focused on the analysis of a significant test case, the semispan tiltrotor wind-tunnel model currently investigated at the NASA Langley Research Center under the denomination of Wing-Rotor Aeroelastic Test System (WRATS). It has been initially built by Bell Helicopter to support the development of the V–22 Osprey, and subsequently destined by the US Navy to the investigation of the tiltrotor technology [13]. A multibody model of such a system has been recently implemented to assess the capabilities and the performances of the multibody code MBDyn [3]. The analytical model describes the kinematics of the system to a comparatively high level of refinement. Significantly, it considers the gimbal joint, that links the hub to the mast in order to allow the flapping motion of the overall hub, and thus implementing a constant velocity joint, and the swashplate, with all the related components that are required to transmit the pitch controls to the rotor blades. The most significant flexible parts of the model are the wing and the rotor blades. All of them have been modeled by means of beam elements, while inertia has been concentrated in the nodes in a lumped scheme. A visualization of the analytical model by means of ADAMS/View is shown in Figure 1, while Figure 2 presents a detail of the hub mechanisms. The baseline rotor blades were originally modeled with four three-node beam elements; such discretization allowed to match the first significant modes of the blade within a range of 100 Hz, which corresponds to the first blade twist mode [3]. The improved performances of the parallel implementation allowed to refine the discretization while at the same time reducing the total computational time. The parallelization performances are presented first, followed by a discussion of the results.

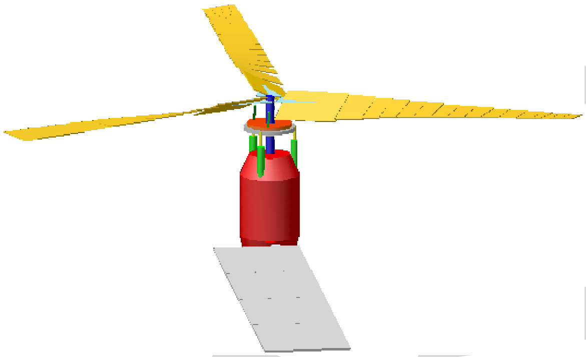

Figure 1: Tiltrotor Model

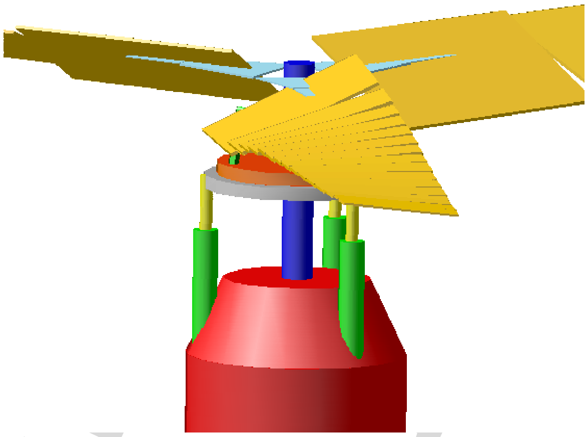

Figure 2: Tiltrotor Controls Detail

Table 1: 8 beam blade model (1097 DOFs)

CPUs	Interf. DOFs	T (s)	Speedup
1	—	801	1
2	58	314	2.55
3	111	240	3.34
4	91	154	5.20
5	158	232	3.45

4.1 Parallelization Performances

All data presented for parallel performance comparison refer to a 2 s simulation run on a 8 CPU HP 4000 platform. Tables 1–3 show excellent results for the speedup. With regard to the smaller problem, with eight flexible beams per blade, the speedup grows almost linearly with the number of CPUs. For larger problems, the behavior of the speedup becomes superlinear. This happens because the interface migrates from the hub area, where it involves a number of algebraic constraints, towards the blade tips, where the local interfaces consist in a single structural node each. Parallel programming has been traditionally applied to problems with a large number of unknowns (10^5 and over). The proposed results show how a linear scalability, with regard to the number of CPUs, is achievable even when the number of unknowns is reduced (less than

Table 2: 16 beam blade model (1673 DOFs)

CPUs	Interf. DOFs	T (s)	Speedup
1	—	1724	1
2	62	562	3.07
3	65	333	5.18
4	80	270	6.39
5	103	215	8.01
6	115	181	9.52

Table 3: 60 beam blade model (4841 DOFs)

CPUs	Interf. DOFs	T (s)	Speedup
1	—	8320	1
2	29	2412	3.45
3	48	1660	5.01
4	48	930	8.94
5	70	706	11.78
6	89	624	13.30
7	108	481	17.30
8	125	444	18.74

10^3). The algorithm scales correctly, except for few special partitions that are not suitable to exploit the model topology, until it reaches an interface size that is comparable to the local problem size.

4.2 Tiltrotor Dynamics Results

The WRATS model has been recently modified by Bell Helicopter by installing a soft-in-plane hub instead of the usual stiff-in-plane one. The soft-in-plane hub is intended as a proof-of-concept device to investigate the properties of such a configuration, mainly in terms of dynamic stability. The blades are attached to the hub by means of a lead-lag hinge, with a viscous damper and a set of springs that are used to emulate the behavior of a possible soft-in-plane flexbeam. An analogous setup has been applied to the analytical model. It is worth remarking that the analyses presented here are not part of a correlation with the tests recently performed at LaRC, nor are intended as a direct numerical prosecution of such work. In fact, since the rest of the model has been also modified, resulting in an appreciable change in the fundamental frequencies, there is no longer a direct correspondence between the test setup and the multibody model. As a consequence, the results here discussed should be viewed as an investigation on the feasibility and the efficiency of the multibody approach to the numerical investigation of this kind of problem.

The basic approach to the stability analysis of the system consists in winding the rotor up to a trimmed configuration and in assessing its behavior. Such a procedure is analogous to usual experiments, without the strict constraint of avoiding unstable conditions. This procedure does not require a trimmed rotating configuration, which cannot be easily determined. However, since the rotor is soft-in-plane, special care must be taken to ensure that the lead-lag motion resulting from too fast a wind-up does not disturb or alter the response of the system. The possibility to perform long runs in a reasonable time, achieved by means of the parallelization of the analysis, allowed the use of quasi-realistic wind-up times. Figure 3 shows the wing in-plane bending during a wind-up maneuver. The reference rotation speed is reached in

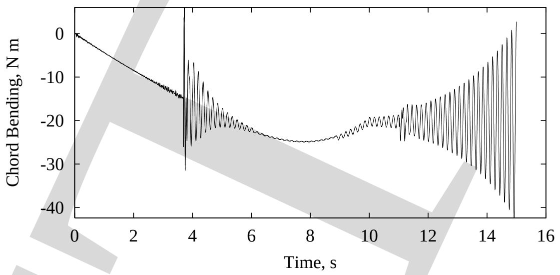

Figure 3: Tiltrotor Wind-Up: Wing Chord Bending for 20% Blade Root Damp

10 s with a half-period cosine wind-up law, namely $\Omega/\Omega_{ref} = 1/2\,(1+\cos(\pi t/10))$ for $0 < t < 10$ s. The system is unstable; in fact, a very conservative 0.2% damping of the wing modes has been considered, while the blade root has a 20% damping at rest. A lead-lag lock that prevents the blade from lagging is released after about 3 s (at 30% of the reference rotation speed); a peak in the bending moment can be clearly noticed in Figure 3 and in detail in Figure 4. The latter shows that there is no

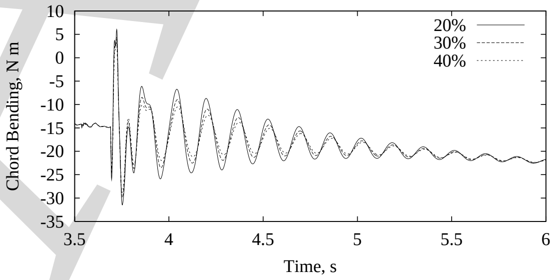

Figure 4: Tiltrotor Wind-Up: Wing Chord Bending After Lead-Lag Lock Release

dynamic instability at such an angular velocity; in fact the response of the system is quickly damped regardless of the damper setting. At 11 s from the beginning, namely one second after the nominal rotation speed of 888 rpm has been reached, a perturbation of the lead-lag motion is applied in form of two couples acting in opposition on two blades, to determine an in-plane displacement of the rotor center of gravity. As a result, the wing is excited by the excentricity of the rotor inertia, mainly in the in-plane and torsion modes. Figure 5 shows the wing in-plane bending in this condition for the previously mentioned blade damper settings. The first stable one is about 30%; a setting that results in a

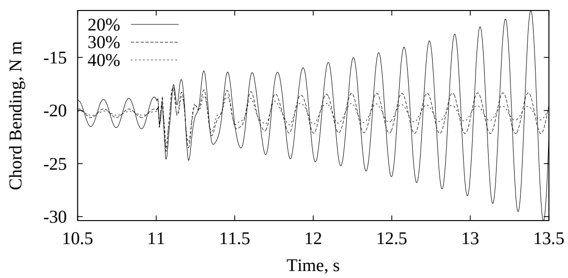

Figure 5: Tiltrotor Wind-Up: Wing Chord Bending After Lead-Lag Perturbation at 100% RPM

damping of about 40% at rest has been used in the subsequent simulations where stability is required. The conversion maneuver is peculiar to the tiltrotor aircraft. Such a maneuver requires the rotor to perform a rotation about an axis that is offset along the mast, and thus implies a movement of the rotor in its plane together with a tilting (see Figure 6). As a consequence, the lead-lag motion of the

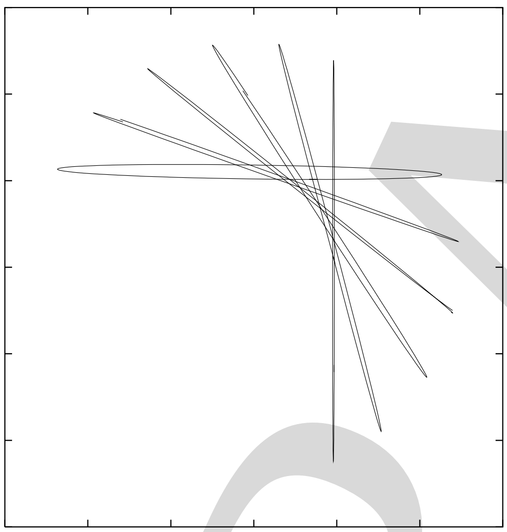

Figure 6: Tiltrotor Conversion: Blade Tip Path

blades can be excited during a transient phase in which the stiffness of the support is reduced because the conversion actuator is not locked, as in normal airplane configuration. The conversion maneuver has been simulated with the multibody model to investigate the possible excitations it may cause. After the usual 10 s wind-up, another 10 s conversion has been simulated. A constant angular speed maneuver at 9 deg/s has been considered, with some smoothing of the startup and stop to avoid excessive excitation of the system. Figures 8 and 7 show the bending moments in the wing and in the flexbeam. The oscillations in the flexbeam moments are related to the flapping caused by the movement of the rotor; they are not transmitted to the wing because they the overall excitation of the rotor is self-balanced. The retreating, lower frequency lead-lag motion can be clearly seen in Figure 7 in the in-plane bending of the flexbeam, both at the beginning and at the end of the maneuver. It dampens more slowly than the baseline oscillations related to the flapping motion. The out-of-plane bending, on the contrary, disappears as soon as the maneuver is completed because of the high aerodynamic damping related to the flapping motion. Figure 8 shows the wing out-of-plane bending rise during the wind-up because of the thrust of the rotor in helicopter configuration. Then, during the conversion, the thrust tilts forward, bending the wing in the in-plane direction while changing from helicopter to aircraft mode. At the beginning of the maneuver the in-plane bending is excited by the in-plane movement of the rotor; an analogous excitation can be found at the end of the conversion, this time in the wing out-of-plane bending.

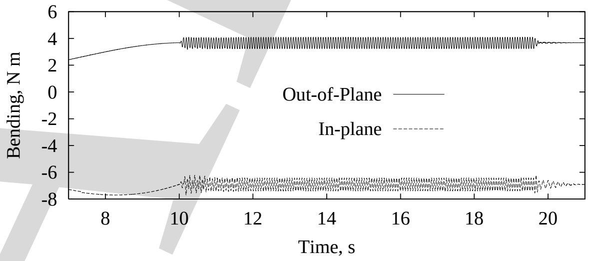

Figure 7: Tiltrotor Conversion: Flexbeam Bending

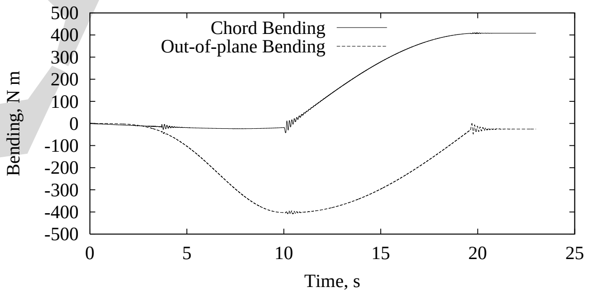

Figure 8: Tiltrotor Conversion: Wing Root Bending

Figure 9 shows an interesting dependency of the first wing bending mode damping on the collective pitch. At low thrust, with a 3.5 deg collective pitch, the 30% blade root damping configuration is stable but the damping is fairly limited. When the collective pitch is increased, an appreciable raise of the damping can be noticed. Notice also that the same excitation has been used in all the presented cases; the larger response at higher collective is related to the aerodynamic loads on the blades, which grow with the collective pitch.

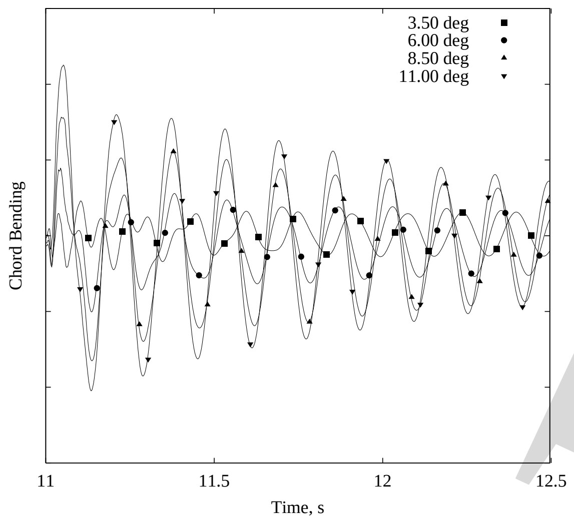

Figure 9: Tiltrotor Stability: Wing Chord Bending After Perturbation

The stability properties of the system are summarized in the so-called *Coleman* plot, shown in Figure 10. The continuous lines represent the theoretical frequencies of the undamped system, while the circles represent the values computed from the simulations. The stability of the system is governed by the lowest wing frequency, which meets the retreating lead-lag frequency of the rotor at about 90% of the reference rotation speed, as can be appreciated also in Figure 3, where the unstable response appears at about 8.5 s.

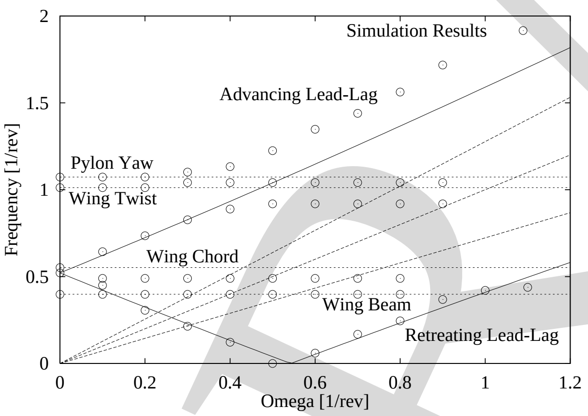

Figure 10: Tiltrotor Stability: Coleman Plot

Concluding Remarks

An efficient, coarse-scale parallelization of a multibody formulation has been presented. Its suitability for the analysis of typical multibody problems as those encountered in the rotorcraft industry has been shown by investigating the dynamic stability of a soft-in-plane tiltrotor. The parallel implementation allowed to consider sophisticated problems requiring long simulations without incurring in unnecessary limitations in the size of the problem. The multibody approach showed its unique ability in investigating unusual flight conditions as those involving transient analyses. Future work will consider a detailed correlation with experimental results, and the investigation of air-resonance problems as well.

References

[1] Raymond G. Kvaternik. A review of some tilt-rotor aeroelastic research at nasa-langley. *J. of Aircraft*, 13(5):357–363, May 1976.

[2] Robert P. Coleman and Arnold M. Feingold. Theory of self-excited mechanical oscillations of helicopter rotors with hinged blades. REPORT 1351, NACA, 1958.

[3] 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.

[4] Gian Luca Ghiringhelli, Pierangelo Masarati, Paolo Mantegazza, and Mark W. Nixon. Multibody analysis of an active control for a tiltrotor. In *CEAS Intl. Forum on Aeroelasticity and Structural Dynamics 1999*, pages 149–158, Williamsburg, VA, June 22–25 1999.

[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.

[6] P. Le Tallec. *Computational Mechanics Advances*, chapter Domain decomposition methods in computational mechanics, pages 121–220. North-Holland, 1994.

[7] B. Smith, P. Bjørstad, and W. Gropp. *Domain decomposition. Parallel multilevel methods for elliptic partial differential equations.* Cambridge University Press, New York, NY, 1996.

[8] T. Sterling, D. Becker, D. Savarese, J. Dorband, U. Ranawake, and C. Packer. BEOWULF: A parallel workstation for scientific computation. In *Proceedings of the 24*th *International Conference on Parallel Processing*, Oconomowoc, WI, 1995.

[9] P. E. Bjørstad and A. Hvidsten. Iterative methods for substructured elasticity problems in structural

analysis. In G. Meurant R. Glowinski, G. Golub and SIAM J. Périaux eds., editors, *Proceedings of the first international symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, 1988.

[10] Dale M. Pitt and David A. Peters. Theoretical prediction of dynamic-inflow derivatives. *Vertica*, 5:21–34, 1981.

[11] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified np-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.

[12] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20:359–392, 1998.

[13] Thomas B. Settle and David L. Kidd. Evolution and test history of the V–22 0.2-scale aeroelastic model. *Journal of the American Helicopter Society*, 37(1):31–45, January 1992.