Analysis of Rotating Systems Using General-Purpose Multibody Dynamics

Alessandro Tasora and Pierangelo Masarati

Abstract This work illustrates the application of general-purpose multibody formulations to the analysis of rotating systems dynamics. Various benchmark problems encompassing multiple deformable components are presented and analyzed. The suitability of the approach is assessed and conclusions are drawn on the basis of correlating the numerical simulations with analogous examples from the open literature.

Keywords Multibody dynamics · Deformable systems · Beam model

1 Introduction

Multibody dynamics is a viable technology for the analysis of complex multidisciplinary mechanisms. Rotating systems represent a special class of mechanisms, characterized by the non-negligible angular motion they are subjected to. Traditionally, systems of this kind have been analyzed by dedicated formulations and software tools that intrinsically take into account the reference rotation motion of the system. Such approaches may be extremely efficient and effective, but may suffer from lack of generality. For example, in the field of helicopter aeromechanics, so-called "comprehensive tools" have been developed (e.g. CAMRAD JA, [1, 2]). Such formulations are specifically tailored for most of the needs in the analysis of helicopter rotor aeroelasticity, posing several restrictions on the topology of the

A. Tasora (✉)
Dipartimento di Ingegneria Industriale, Università degli Studi di Parma,
viale delle Scienze 181/a, 43100 Parma, Italy
e-mail: alessandro.tasora@unipr.it

P. Masarati
Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano,
via La Masa 34, 20156 Milan, Italy
e-mail: pierangelo.masarati@polimi.it

problem and on the type of analysis. Later, the need to consider rather general configurations and non-standard analysis pushed for the incorporation of features that are typical of multibody dynamics (e.g. CAMRAD II, [3]).

Nowadays, rather general multibody formulations are used to model and analyze the dynamics of a variety of systems. MBDyn[1] [4] and Chrono::Engine[2] [5, 6] are noteworthy examples of original formulations implemented in freely distributed software originating from the academia. This work illustrates the application of general-purpose multibody formulations to the analysis of flexible rotating systems dynamics, and presents benchmarks that demonstrate the suitability of this approach.

The need to simulate flexible parts with large rotations within a multibody dynamics framework calls for advanced formulations, mostly rooted in finite element approximations of structural components; see for example [7–9].

The three dimensional beam is one of the most used models for this class of problems. The case of large rotations poses difficulties that researchers tried to overcome in different ways. A recent approach is based on the absolute nodal coordinate formulation, see for instance [10], whereas other approaches are based on the so-called corotational formulation, as in [11–13].

In this paper we will present benchmarks based on public data available at [14] and recently discussed in [15] within a joint effort for flexible multibody formulations and software benchmarking, using software rooted on two different approaches. MBDyn adopts the geometrically-exact beam formulation (GEBF) described in [16] and [17], whereas Chrono::Engine uses the element-independent corotational formulation presented in [18], with some modifications. Both use an incremental approach for the time integration of large rotations. Whenever available, equivalent results obtained with Dymore are also presented. Dymore is based on the GEBF, and uses the Wiener-Milenković parameters to represent finite rotations. For the sake of completeness, we succinctly present the corotational formulation implemented in Chrono::Engine, whereas for the GEBF implemented in MBDyn we refer to the above mentioned literature.

2 Corotational Formulation

Corotational formulations foster the reuse of finite element algorithms and theories whose behavior in the linear field are already well known and tested.

Figure 1 shows the concept of the corotational formulation in Chrono::Engine. A floating coordinate system F follows the deformed element, thus the overall gross motion into the deformed state $\mathcal{C}_D$ can be seen as the superposition of a large rigid body motion from the reference configuration $\mathcal{C}_0$ to the so called *floating* or *shadow*

---

[1]http://www.mbdyn.org/.

[2]http://www.projectchrono.org/.

Fig. 1 A schematic representation of the corotational concept
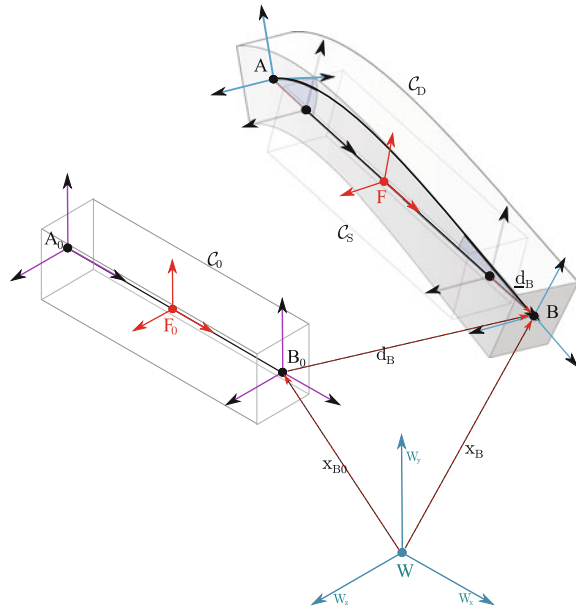
configuration $\mathcal{C}_S$, plus a local small-strain deformation from $\mathcal{C}_S$ to $\mathcal{C}_D$. In this work, underlined symbols will represent variables expressed in the basis of the floating reference F.

The rationale of the corotational approach is a procedure to compute a global tangent stiffness $\boldsymbol{K}_e$ and a global force $\boldsymbol{f}_e$ for each element e, given its local $\underline{\boldsymbol{K}}$, its local $\underline{\boldsymbol{f}}$ and the rigid body motion of the frame F in $\mathcal{C}_0$ to F in $\mathcal{C}_S$.

Whenever the element moves, the position and rotation of F is updated. In literature there are many options to this end; to avoid dependence on connectivity [19], in our implementation we decided to put the origin of F in the midpoint of the AB segment, as $\boldsymbol{x}_F = \frac{1}{2}(\boldsymbol{x}_B - \boldsymbol{x}_A)$, and we align its $\boldsymbol{X}$ axis with $\boldsymbol{x}_B - \boldsymbol{x}_A$. The remaining $\boldsymbol{Y}$ and $\boldsymbol{Z}$ axes of F are obtained with a Gram-Schmidt orthogonalization, enforcing $\boldsymbol{Y}$ to bisect the $\boldsymbol{Y}$ axes of A and B when projected on the plane orthogonal to $\boldsymbol{X}$. This is important in case of torsion.

The rotation matrix of F is $\boldsymbol{R}_F \in SO3$; it is parametrized with the unit quaternion $\boldsymbol{\rho}_F \in \mathbb{H}_1$. Similarly, quaternions are used to store the rotations of the two nodes, with $\boldsymbol{\rho}_A$ and $\boldsymbol{\rho}_B$. Hence the state of the system is $\boldsymbol{s} = [\boldsymbol{q}, \boldsymbol{v}]$ with $\boldsymbol{q} = [\boldsymbol{x}_1, \boldsymbol{\rho}_1, \boldsymbol{x}_2,$ $\boldsymbol{\rho}_2, \ldots, \boldsymbol{x}_n, \boldsymbol{\rho}_n] \in \mathbb{R}^{(3+4)n}$ and $\boldsymbol{v} = [\boldsymbol{v}_1, \boldsymbol{\omega}_1, \boldsymbol{v}_2, \boldsymbol{\omega}_2, \ldots, \boldsymbol{v}_n, \boldsymbol{\omega}_n] \in \mathbb{R}^{(3+3)n}$. Note that, because of some algorithmic optimizations, we consider $\boldsymbol{\omega}_i$ to be expressed in the *local* basis of the ith node unlike $\boldsymbol{x}_i, \boldsymbol{\rho}_i, \boldsymbol{v}_i$ that are considered in the global basis.

For each element, given the global positions and rotations of the two end nodes, stored in the $\boldsymbol{s}$ state at indexes i_A and i_B, it is possible to compute the actual displacement part of $\underline{\boldsymbol{d}}$ as $\underline{\boldsymbol{d}}_A = \underline{\boldsymbol{x}}_A - \underline{\boldsymbol{x}}_{A_0} = \boldsymbol{R}_{F_0}^t(\boldsymbol{x}_{A_0} - \boldsymbol{x}_{F_0}) - \boldsymbol{R}_F^t(\boldsymbol{x}_A - \boldsymbol{x}_F)$ and $\underline{\boldsymbol{d}}_B = \underline{\boldsymbol{x}}_B - \underline{\boldsymbol{x}}_{B_0} = \boldsymbol{R}_{F_0}^t(\boldsymbol{x}_{B_0} - \boldsymbol{x}_{F_0}) - \boldsymbol{R}_F^t(\boldsymbol{x}_B - \boldsymbol{x}_F)$.

The rotation part, however, introduces a complication, owing to the fact that finite rotations do not compose as vectors and must be dealt with special algebraic tools. First, one must compute the local rotation of nodes respect to the F floating reference with $\underline{\boldsymbol{R}}_A = \boldsymbol{R}_F^t \boldsymbol{R}_A \boldsymbol{R}_{A_A}^t$ and $\underline{\boldsymbol{R}}_B = \boldsymbol{R}_F^t \boldsymbol{R}_B \boldsymbol{R}_{B_A}^t$. Equivalently, one can use quaternion products to write: $\underline{\boldsymbol{\rho}}_A = \boldsymbol{\rho}_F^t \boldsymbol{\rho}_A \boldsymbol{\rho}_{A_A}^t$ and $\underline{\boldsymbol{\rho}}_B = \boldsymbol{\rho}_F^t \boldsymbol{\rho}_B \boldsymbol{\rho}_{B_A}^t$. Here the optional term $\boldsymbol{\rho}_{A_A}^t$ (or $\boldsymbol{R}_{A_A}^t$) represents the initial rotation of node A respect to F_0.

Then, the finite rotation pseudovectors $\underline{\boldsymbol{\theta}}_A$ and $\underline{\boldsymbol{\theta}}_B$ are obtained in the following way. It is known that, for an element $\boldsymbol{R}$ in Lie group SO(3) and an element $\boldsymbol{\Theta}$ in the corresponding Lie algebra so(3), one has $\boldsymbol{\Theta} = \text{skew}(\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is also an element of the Lie group Spin(3), double cover of SO(3). Vice versa, one can extract the $\boldsymbol{\theta}$ vector from the $\boldsymbol{\Theta}$ spinor by computing $\boldsymbol{\theta} = \text{axial}(\boldsymbol{\Theta})$. Also, it holds $\boldsymbol{R} = \exp(\boldsymbol{\Theta})$ and $\boldsymbol{\Theta} = \text{Log}(\boldsymbol{R})$, where $\exp(\cdot)$ builds the rotation matrix using an exponential; for details on this exponential and the implementation of $\text{skew}(\cdot)$, $\text{axial}(\cdot)$, $\text{Log}(\cdot)$, see for example [20].

In the work of other Authors, the theory above is used to compute $\underline{\boldsymbol{\theta}}_A = \text{axial}(\text{Log}(\underline{\boldsymbol{R}}_A))$, but in our case the adoption of quaternions lead to an alternative, more straightforward expression. In fact it is known that for a unit quaternion $\boldsymbol{\rho} \in \mathbb{H}_1$ it holds $\boldsymbol{\rho} = [\cos(\theta/2), \boldsymbol{u}\sin(\theta/2)]$, with rotation angle $\theta = |\boldsymbol{\theta}|$ about rotation unit vector $\boldsymbol{u} = \boldsymbol{\theta}/\theta$. Therefore it is possible to compute $\underline{\boldsymbol{\theta}}_A$ and $\underline{\boldsymbol{\theta}}_B$ as: $\theta_A = 2\arccos(\Re(\underline{\boldsymbol{\rho}}_A))$, $\boldsymbol{u}_A = \frac{1}{\sin(\theta_A/2)}\Im(\underline{\boldsymbol{\rho}}_A)$, and $\underline{\boldsymbol{\theta}}_A = \theta_A \boldsymbol{u}_A$ (the same for the B node).

Once $\underline{\boldsymbol{d}} = [\underline{\boldsymbol{d}}_A, \underline{\boldsymbol{\theta}}_A, \underline{\boldsymbol{d}}_B, \underline{\boldsymbol{\theta}}_B]$ has been computed, well-known theories are available to compute the stiffness matrix $\underline{\boldsymbol{K}} = \underline{\boldsymbol{K}}(\underline{\boldsymbol{d}})$. in this work we compute $\underline{\boldsymbol{K}}$ using the Eulero-Bernoulli theory, and in general we set $\underline{\boldsymbol{f}}_{in} = \underline{\boldsymbol{K}}\underline{\boldsymbol{d}}$.

The local data $\underline{\boldsymbol{K}}$ and $\underline{\boldsymbol{f}}_{in}$ must be mapped to the global reference: to this end we use the corotational approach expressed in [13], where the adoption of projectors that filter rigid body motion is used to improve the consistency and the convergence of the method. Such formulation requires the introduction of various matrices, in the following we succinctly report them, along with modifications that we use in our method.

- the $\boldsymbol{\Lambda}(\boldsymbol{\theta}) = \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{w}}$ matrix, whose analytic expression is $\boldsymbol{\Lambda}(\boldsymbol{\theta}) = \boldsymbol{I}_{3\times3} - \frac{1}{2}\text{skew}(\boldsymbol{\theta}) + \zeta\text{skew}(\boldsymbol{\theta})^2$ with $\zeta = \left(1 - \frac{1}{2}\theta\cot\text{an}(\frac{1}{2}\theta)\right)/\theta^2$,
- the $\boldsymbol{H}$ transformation matrix:

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{H_n}(\underline{\boldsymbol{\theta}}_A) & \boldsymbol{0}_{6\times6} \\ \boldsymbol{I}_{6\times6} & \boldsymbol{H_n}(\underline{\boldsymbol{\theta}}_B) \end{pmatrix}; \quad \boldsymbol{H_n}(\boldsymbol{\theta}) = \begin{pmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{\Lambda}(\underline{\boldsymbol{\theta}}) \end{pmatrix} \quad (1)$$

that tends to a unit matrix $\boldsymbol{I}_{12\times12}$ for $\theta \downarrow 0$,
- the $\underline{\boldsymbol{P}}$ projector matrix: $\underline{\boldsymbol{P}} = \boldsymbol{I}_{12\times12} - \underline{\boldsymbol{S}}^D\underline{\boldsymbol{G}}$ where $\underline{\boldsymbol{S}}^D$ is the so called *spin lever* matrix, built with $\underline{\boldsymbol{x}}_A$ and $\underline{\boldsymbol{x}}_B$, the positions of the end nodes respect to the center F of the beam, expressed in F basis:

$$\underline{\boldsymbol{S}}^D = \begin{pmatrix} -\text{skew}(\underline{\boldsymbol{x}}_A) \\ \boldsymbol{I}_{3\times3} \\ -\text{skew}(\underline{\boldsymbol{x}}_B) \\ \boldsymbol{I}_{3\times3} \end{pmatrix} \tag{2}$$

and where $\underline{\boldsymbol{G}}$ is the so called *spin fitter* matrix, that takes into account the change of orientation of the F frame as the end nodes change position or rotation. For the two nodes beam, it is $\underline{\boldsymbol{G}} = [\partial\underline{\boldsymbol{\omega}}_F/\partial\underline{\boldsymbol{x}}_A, \partial\underline{\boldsymbol{\omega}}_F/\partial\underline{\boldsymbol{\omega}}_A, \ldots]$, and for our custom choice of orientation and position of F, described at the beginning of this section, we have

$$\underline{\boldsymbol{G}} = \begin{bmatrix} 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/L & 0 & 0 & 0 & 0 & 0 & -1/L & 0 & 0 & 0 \\ 0 & -1/L & 0 & 0 & 0 & 0 & 0 & 1/L & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

Note that this expression is different from the one reported in [18] because they put the F frame at the beginning of the beam whereas we put it in the middle, moreover it rotates a bit differently,

- the $\boldsymbol{R}_\diamond$ rotation-transformation matrix:

$$\boldsymbol{R}_\diamond = \begin{pmatrix} \boldsymbol{R}_F & & & \\ & \boldsymbol{R}_A^t \boldsymbol{R}_F & & \\ & & \boldsymbol{R}_F & \\ & & & \boldsymbol{R}_B^t \boldsymbol{R}_F \end{pmatrix} \tag{4}$$

Note that $\boldsymbol{R}_\diamond$ is different from the one often reported in literature, ex. in [18] or [13], because we update the rotation of nodes with rotation pseudovectors expressed in node *local* coordinates (hence the $\boldsymbol{R}_A^t$ and $\boldsymbol{R}_B^t$ transformation), coherently with what we said about angular velocities being expressed in local references and not in global reference, in our state $\boldsymbol{s}$.

Given the matrices above, one can compute the global version of internal forces:

$$\boldsymbol{f}_{in} = \boldsymbol{R}_\diamond \underline{\boldsymbol{P}}^t \boldsymbol{H}^t \underline{\boldsymbol{f}}_{in} \tag{5}$$

For the computation of the global tangent stiffness, one needs two additional matrices. First, we split the last part of Eq. (5) in four three-dimensional vectors: $\underline{\boldsymbol{P}}^t \boldsymbol{H}^t \underline{\boldsymbol{f}}_{in} = [\underline{\boldsymbol{n}}_A, \underline{\boldsymbol{m}}_A, \underline{\boldsymbol{n}}_B, \underline{\boldsymbol{m}}_B]$, then we build the $\boldsymbol{F}_{nm}$ and $\boldsymbol{F}_n$ matrices:

$$\underline{\boldsymbol{F}}_{nm} = \begin{pmatrix} \text{skew}(\underline{\boldsymbol{n}}_A) \\ \text{skew}(\underline{\boldsymbol{m}}_A) \\ \text{skew}(\underline{\boldsymbol{n}}_B) \\ \text{skew}(\underline{\boldsymbol{m}}_B) \end{pmatrix}, \quad \underline{\boldsymbol{F}}_n = \begin{pmatrix} \text{skew}(\underline{\boldsymbol{n}}_A) \\ \boldsymbol{0}_{3\times 3} \\ \text{skew}(\underline{\boldsymbol{n}}_B) \\ \boldsymbol{0}_{3\times 3} \end{pmatrix} \tag{6}$$

Finally one can compute the tangent stiffness matrix of the element in global coordinates, also accounting for geometric stiffening:

$$\boldsymbol{K} = \boldsymbol{R}_{\diamond}\left(\underline{\boldsymbol{P}}^t\boldsymbol{H}^t\underline{\boldsymbol{K}}\boldsymbol{H}\underline{\boldsymbol{P}} - \underline{\boldsymbol{F}}_{nm}\underline{\boldsymbol{G}} - \underline{\boldsymbol{G}}^t\underline{\boldsymbol{F}}_n^t\underline{\boldsymbol{P}} + \underline{\boldsymbol{P}}^t\boldsymbol{L}_H\underline{\boldsymbol{P}}\right)\boldsymbol{R}_{\diamond}^t \tag{7}$$

$$\boldsymbol{K} = \boldsymbol{R}_{\diamond}(\boldsymbol{K}_M - \boldsymbol{K}_{GR} - \boldsymbol{K}_{GP} + \boldsymbol{K}_{GH})\boldsymbol{R}_{\diamond}^t \tag{8}$$

We remark the following notes:

- the three terms $\boldsymbol{K}_{GR}$ (related to change in rotation of the F frame), $\boldsymbol{K}_{GP}$ (related to changes in projectors), $\boldsymbol{K}_{GH}$ (related to changes in $\boldsymbol{H}$) are responsible of the so called *geometric stiffness*,
- the $\boldsymbol{K}_{GH}$ term is not used in our formulation since we found no major benefits in computing it; see [13] for details on $\boldsymbol{L}_H$,
- the $\boldsymbol{K}_M$, which represents the so called *material stiffness*, is always symmetric (at least with Eulero-Bernoulli beams), but the terms for geometric stiffness introduce asymmetry,
- some Authors [21] show that, under mild assumptions, neglecting the asymmetric part does not hampers the convergence of Newton-Raphson iterations; hence the variant: $\boldsymbol{K}_{symm} = \boldsymbol{R}_{\diamond}\left(\underline{\boldsymbol{P}}^t\boldsymbol{H}^t\underline{\boldsymbol{K}}\boldsymbol{H}\underline{\boldsymbol{P}} - \underline{\boldsymbol{F}}_{sy}\underline{\boldsymbol{G}} - \underline{\boldsymbol{G}}^t\underline{\boldsymbol{F}}_{sy}^t\underline{\boldsymbol{P}}\right)\boldsymbol{R}_{\diamond}^t$ where $\underline{\boldsymbol{F}}_{sy} = \frac{1}{2}(\underline{\boldsymbol{F}}_{nm} + \underline{\boldsymbol{F}}_n)$.

3 Benchmark: Princeton Beam Experiment

This benchmark aims at the validation of the finite element implementations in a static problem with geometric nonlinearity. A thin cantilevered beam, constrained in O, is subject to large deformations and large rotations because of a tip load in E; for different angles θ one obtains out-of-plane displacements even if the load is vertical, and the beam is subject to a twisting action.

Experimental results, for a beam made with 7075 aluminium, are available in [22, 23] and are used for comparison.

We list the main properties, with reference to Fig. 2: beam length $L = 0.508$ m, section thickness $T = 3.2024$ mm, section height $H = 12.77$ mm, Young modulus $E = 71.7$ GPa, $v = 0.31$, $G = E\frac{(1+v)}{2} = 27.37$ GPa.

Three loading conditions are tested: $P_1 = 4.448$ N, $P_2 = 8.896$ N, and $P_3 = 13.345$ N, for increasing values of the θ angle in the $[0°, 90°]$ range.

Fig. 2 Setup of the benchmark for the Princeton beam experiment
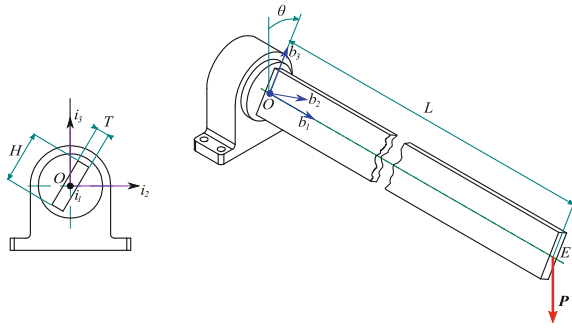
Results in Figs. 3, 4 and 5 show a good agreement between the present corotational beam formulation and the geometrically-exact beam formulations presented in [16] for Dymore and in [17] for MBDyn, as well as an agreement with the experimental results in [22, 23], which was recently discussed in [15].

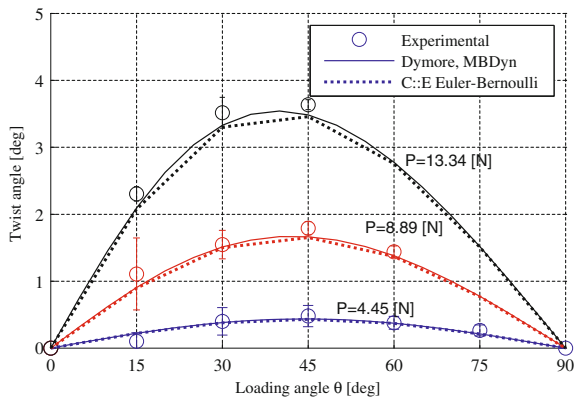**Fig. 3** Twist rotation of the beam for the Princeton experiment

Fig. 4 Flapwise displacement at the beam tip versus loading angle for three loading conditions
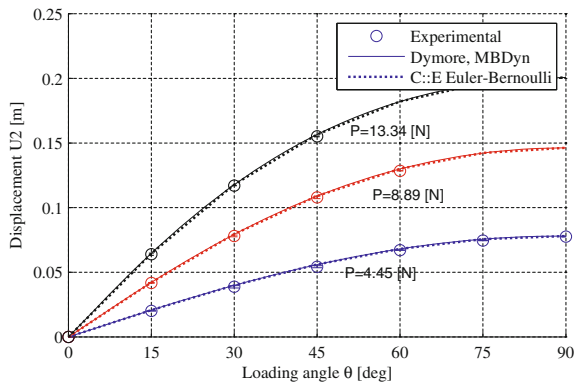
Fig. 5 Chordwise
displacement at the beam tip
versus loading angle for three
loading conditions
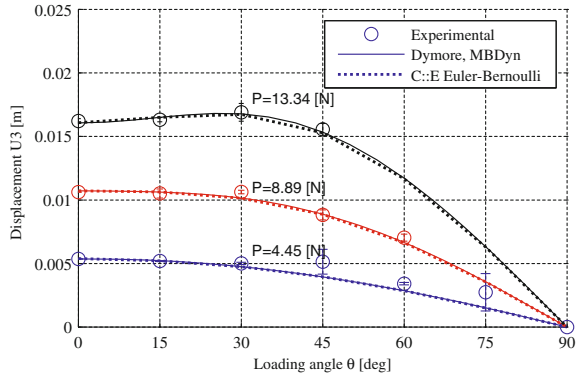
We remark that, because of the geometric nonlinearity, the solver has to perform few Newton-Raphson steps before obtaining a zero residual. For very large non-linearities, a continuation strategy might help the convergence of the Newton-Raphson solver.

4 Benchmark: Lateral Buckling

This benchmark tests nonlinear effects in a dynamic context. A beam is bent in its plane of greatest flexural rigidity, up to the point that triggers lateral buckling. In a quasi-static non-linear analysis, results are visible in Fig. 7. In the context of dynamics, when buckling occurs, the beam snaps laterally and twists, inducing highly oscillatory motions. The corotational approach can capture the nonlinear nature of this phenomena.

As shown in Fig. 6, the *RC* beam is clamped at point *R*, its length is $L = 1$ m, and its rectangular section has size $H = 100$ mm and $B = 10$ mm (Fig. 7).

To induce the snapping, a tip load at *C* is imposed by mean of a rotating crank *GB* and a vertical rod *TB*, with a spherical joint in *C* and a revolute joint in *B*. An initial imperfection is simulated by displacing the vertical bar and the crank by an offset $d = 0.1$ mm in the off-plane direction i_2. The crank has length $L_c = 0.05$ m and a circular section with diameter $D_r = 24$ mm, while the vertical rod has a length $L_r = 0.25$ m and a circular section with diameter $D_r = 48$ mm. The rotation of the crank is initially enforced by a prescribed motion function $\phi_c(t) = \pi(1 - \cos(\pi t/T_c))/2$, with $T_c = 0.4$ s, then for $t > T_c$ it is $\phi_c(t) = \pi$.

All parts are made of aluminum, hence with Young modulus $E = 73$ GPa and Poisson ratio $v = 0.3$. Given the above mentioned sections, their inertia values I_{zz} and I_{yy} and their torsion constants J are computed using formulas available in classical textbooks.

In the Chrono::Engine test, the *RC* beam is modeled with 12 finite elements, whereas the crank and the rod are modeled with 3 elements each. Results in

Fig. 6 Setup of the benchmark for lateral buckling dynamics
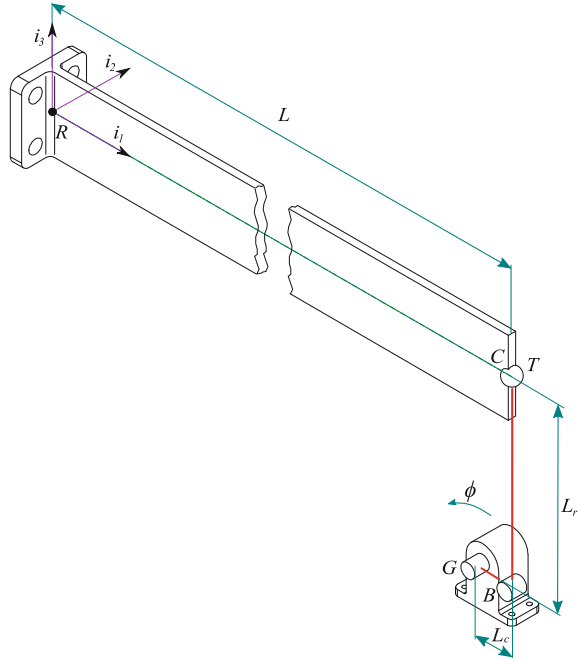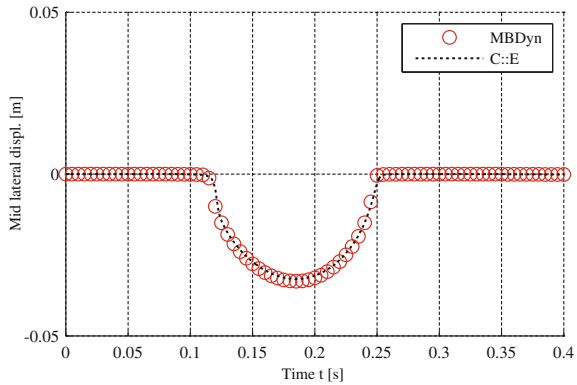

Fig. 7 Static displacement of the beam along i_2, at the mid point

Figs. 8 and 9 show that the lateral buckling is triggered exactly at the same moment for all the formulations, although the Chrono::Engine integral is more damped. The numerical damping is a consequence of the fact that the Chrono::Engine default integrator is a timestepper for DVI non-smooth problems [24, 25]. This, in the case of no frictional contacts, boils down to a linearly-implicit first-order scheme, hence it shows the same damping effect of an implicit Euler method. Other integrals are obtained with higher order methods and are affected by numerical damping to a much lower degree.

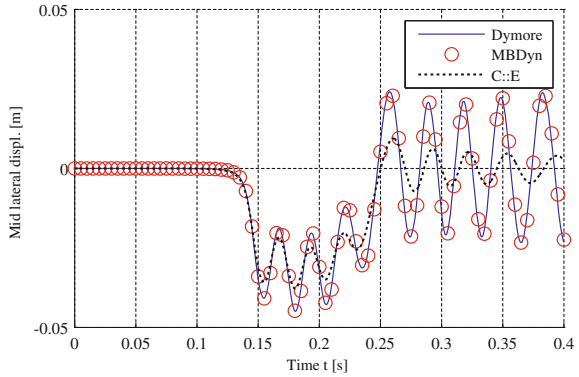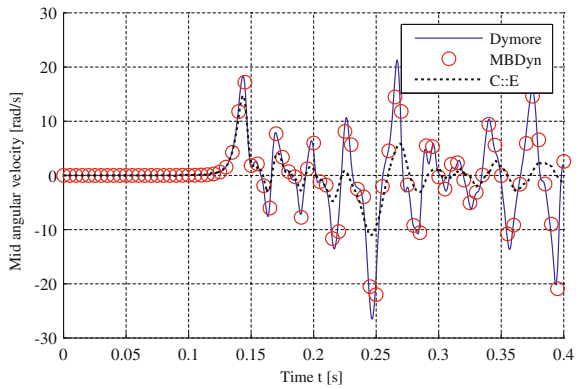**Fig. 8** Displacement of the beam along i_2, at the mid point

Fig. 9 Angular velocity of the beam, at the mid point

5 Benchmark: Unbalanced Rotating Shaft

This benchmark explores the reliability of the numerical method in the analysis of a flexible system rotating at finite angular velocity. A rotating unbalanced shaft of length $L = 6$ m is integrated in time. A rigid disk is connected to the shaft at mid-span, above the reference shaft axis by an offset $d = 0.05$ m. The shaft is made of steel (density $\rho = 7800$ kg/m^3, Young's modulus $E = 210$ GPa, Poisson's ratio $v = 0.3$). The cross section is annular ($r_i = 0.045$ m, $r_o = 0.05$ m). The mass of the disk is $m_d = 70.573$ kg, the radius is $r_d = 0.24$ m, and the thickness is $t_d = 0.05$ m. The system is subjected to gravity ($g = 9.81$ m/s^2) directed transversely. The end 'R' of the shaft is connected to the ground by a cylindrical joint (displacement along and rotation about the shaft's axis are permitted). The end 'T' is supported by a revolute joint; the relative angular velocity about the shaft axis is prescribed as a function of time,

Fig. 10 Setup of the
unbalanced rotating shaft
benchmark
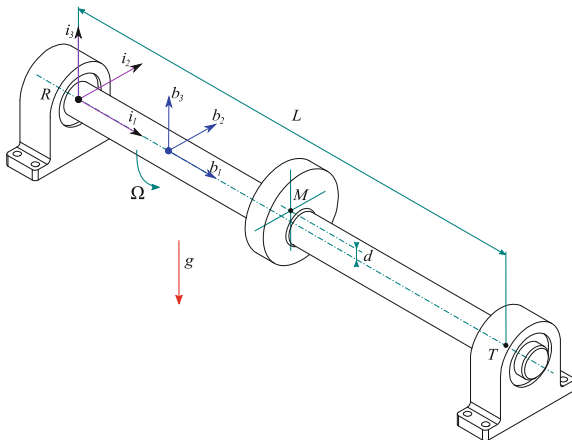
$$\Omega(t) = \begin{cases} A_1\omega(1 - \cos(\pi t/T_1))/2 & 0 \le t \le T_1 \\ A_1\omega & T_1 < t \le T_2 \\ A_1\omega + (A_2 - A_1)\omega(1 - \cos(\pi(t - T_2)/(T_3 - T_2)))/2 & T_2 < t \le T_3 \\ A_2\omega & T_3 < t \end{cases}$$

with $A_1 = 0.8$, $A_2 = 1.2$, $T_1 = 0.5$ s, $T_2 = 1$ s, $T_3 = 1.25$ s, and $\omega = 60$ rad/s, close to
the first natural frequency of the system. The shaft accelerates from zero and passes
from sub-critical to super-critical regime; when passing through the first natural
bending frequency of the system, lateral oscillations occur and significant forces
take place, as predicted by the linear theory of unbalanced rotors (Figs. 10 and 11).
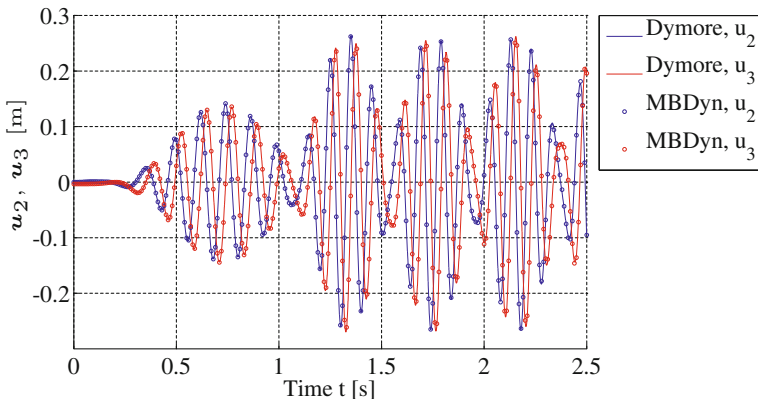

Fig. 11 Mid-point transverse displacement of unbalanced rotating shaft

6 Conclusions

Benchmarks show that the presented multibody software frameworks can accommodate flexible elements of beam type that can match the requirements of rotor dynamics, yet keeping the benefits of general-purpose multibody tools, such as unlimited number of constraints, actuators and rigid parts.

References

1. Johnson W (1981) Development of a comprehensive analysis for rotorcraft—I. Rotor model and wake analysis. Vertica 5:99–129
2. Johnson W (1981) Development of a comprehensive analysis for rotorcraft—II. Aircraft model, solution procedure and applications. Vertica 5:185–216
3. Johnson W (1994) Technology drivers in the development of CAMRAD II. In: American helicopter society aeromechanics specialists conference, San Francisco, California
4. Masarati P, Morandini M, Mantegazza P (in press) An efficient formulation for general-purpose multibody/multiphysics analysis. J Comput Nonlinear Dyn. doi:10.1115/1.4025628
5. Mazhar H, Heyn T, Pazouki A, Melanz D, Seidl A, Barthlomew A, Tasora A, Negrut D (2013) Chrono: a parallel multi-physics library for rigid-body, flexible-body, and fluid dynamics. Mech Sci 4:49–64
6. Tasora A (2014) www.chronoengine.info
7. Cardona A, Geradin M (1988) A beam finite element non-linear theory with finite rotations. Int J Numer Meth Eng 26:2403–2438
8. Argyris J (1982) An excursion into large rotations. Comput Methods Appl Mech Eng 32:85–155
9. Bauchau OA (1998) Computational schemes for flexible, non-linear multi-body systems. Multibody Syst Dyn 2:169–225
10. Sugiyama H, Escalona JL, Shabana AA (2003) Formulation of three-dimensional joint constraints using the absolute nodal coordinates. Nonlinear Dyn 31:167–195. doi:10.1023/A:1022082826627
11. Crisfield M (1990) A consistent co-rotational formulation for non-linear, three-dimensional, beam-elements. Comput Methods Appl Mech Eng 81:131–150
12. Rankin CC, Brogan FA (1986) An element independent corotational procedure for the treatment of large rotations. J Press Vessel Technol 108(2):165–174
13. Felippa, C, Haugen B (2005) A unified formulation of small-strain corotational finite elements: I. theory. Comput Methods Appl Mech Eng 194:2285–2335 (Computational Methods for Shells)
14. Bauchau O (2014) http://www.dymoresolutions.com/
15. Bauchau OA, Wu G, Betsch P, Cardona A, Gerstmayr J, Jonker B, Masarati P, Sonneville V (2014) Validation of flexible multibody dynamics beam formulations using benchmark problems. In: IMSD 2014, Korea
16. Bauchau OA, Kang NK (1993) A multibody formulation for helicopter structural dynamic analysis. J Am Helicopter Soc 38:3–14
17. Ghiringhelli GL, Masarati P, Mantegazza P (2000) A multi-body implementation of finite volume beams. AIAA J 38:131–138
18. Rankin C, Nour-Omid B (1988) The use of projectors to improve finite element performance. Comput Struct 30:257–267
19. Masarati P, Morandini M (2010) Intrinsic deformable joints. Multibody Syst Dyn 23:361–386. doi:10.1007/s11044-010-9194-y

20. Simo J, Vu-Quoc L (1986) A three-dimensional finite-strain rod model. Part II: Computational aspects. Comput Methods Appl Mech Eng 58:79–116
21. Nour-Omid B, Rankin C (1991) Finite rotation analysis and consistent linearization using projectors. Comput Methods Appl Mech Eng 93:353–384
22. Dowell EH, Traybar JJ (1975) An experimental study of the nonlinear stiffness of a rotor blade undergoing flap, lag, and twist deformations. Aerospace and Mechanical Science Report 1194, Princeton University
23. Dowell EH, Traybar JJ (1975) An experimental study of the nonlinear stiffness of a rotor blade undergoing flap, lag, and twist deformations. Aerospace and Mechanical Science Report 1257, Princeton University
24. Negrut D, Tasora A, Mazhar H, Heyn T, Hahn P (2012) Leveraging parallel computing in multibody dynamics. Multibody Syst Dyn 27:95–117
25. Tasora A, Anitescu M (2011) A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics. Comput Methods Appl Mech Eng 200:439–453