

# Object-oriented modelling of flexible beams

Francesco Schiavo · Luca Viganò · Gianni Ferretti

Received: 28 July 2005 / Accepted: 7 March 2006  
© Springer Science + Business Media B.V. 2006

**Abstract** In this paper the problem of modelling flexible thin beams in multibody systems is tackled. The proposed model, implemented with the object-oriented modelling language *Modelica*, is completely modular, allowing the realization of complex systems by simple aggregation of basic components. The finite element method is employed as the basic scheme to spatially discretize the model equations. Exploiting the modular features of the language, the beam substructuring discretisation scheme (mixed finite element-finite volume) is derived as well. Selected simulation results are presented in order to validate the model with respect to both theoretical predictions and literature reference results.

**Keywords** Object-Oriented and modular modelling · Flexible beams · Modelica · Finite element method

## 1. Introduction

Many engineering applications require the development of simulation models for flexible multibody systems (e.g., robot manipulators, helicopter rotors, aircraft wings, space structures, machining tools, etc.) both dynamically accurate and computationally affordable.

The task of developing models for generic-shaped, fully deformable bodies is usually demanded to specialized simulation tools, due to the complexity of the task. Such models are usually adequate for structural analysis and design tasks, while being far too complex for affordable dynamics simulation and analysis or for control systems prototyping.

On the other hand, particular classes of deformable bodies, such as flexible beams, can be represented with less complex models which are still able to represent all the dynamically relevant deformation effects.

Flexible beams are continuous non linear dynamical systems characterized by an infinite number of degrees of freedom. Obviously, dealing directly with infinite dimensional models

---

F. Schiavo, L. Viganò, G. Ferretti (✉)  
Politecnico Di Milano, Dipartimento Di Elettronica Ed Informazione (DEI), Via Ponzio 34/5, 20133  
Milano, Italy  
e-mail: schiavo,vigano,ferretti@elet.polimi.it

is impractical both for dynamic analysis and simulation purposes. Hence, it is necessary to introduce methods to describe flexibility with a discrete number of parameters.

Three different approaches have been traditionally used to derive approximated finite dimensional models: lumped parameters, assumed modes and finite element method [16],[26].

The lumped parameter approach is the simplest one. In this method each flexible beam is divided into a finite number of rigid beams, introducing pseudojoints, and the flexibility is represented by springs that restrict the motion of each pseudojoint. This method is however rarely used because of the difficulty in determining the spring constants of the pseudojoints and then of achieving a suitable accuracy up to the desired approximation frequency.

The assumed modes model formulation has been widely used in the literature [25]. It describes beam flexibility using truncated modal series, based on spatial mode eigenfunctions and time varying vibrational modes. One of the best features offered by such a method is the fine control on the accuracy up to the desired approximation frequency. Although conceptually simple, this description requires to find out the best selection for spatial modal shapes and the boundary conditions, which is not at all a trivial task. In addition to that, the selection of the appropriate eigenfunctions and the resulting vibrational modes could depend on the boundary conditions for the specific case at hand, thus ruling out the possibility of a modular approach for the model development.

In the finite element method approach [34], the flexible beam is divided into several elements, with a local description of the deformation field by the use of element-wise basis functions. Although such approach could be computationally more demanding than the modal one (it is usually necessary to use a larger number of elements than of modal eigenfunctions to obtain the same accuracy), it allows a formulation which is independent of the actual boundary condition [29]. The finite element method is then a viable choice for the representation of flexible beams within a modular environment [7].

As far as the theory of elasticity to be used is concerned, it must be pointed out that beam deflection, with respect to the rigid configuration, is generally assumed to be small, which allows to adopt the linear theory. In this case the Euler-Bernoulli theory [31] can be used to describe beam flexibility, neglecting the effects of shear deformation and assuming uniform cross-sectional properties along the beam. In this paper, we consider linear elasticity theory for the modelling of flexible *thin* beams. On the other hand, Timoshenko theory [31] should be used for models where such effects need to be taken into account (e.g., for short beams).

A main issue addressed in this paper regards the model development with a real modular approach, whose characteristics can be summarized as follows:

- realization of models of large, complex and heterogeneous physical systems by basic components *connection*;
- models description by stating physical principles rather than by writing calculation algorithms (*acausal* modelling);
- charging the simulation environment with the task of defining the computational causality of the assembled model.

Among the various modular physical modelling languages and tools developed from the end of the 70's (e.g., ASCEND [23], OMOLA [12], gPROMS [17], MOSES [10]), *Modelica* [28], [13] and *Dymola* [5] have emerged respectively as a standard for a modular, acausal modelling language and for a complete and efficient modelling and simulation environment. *Dymola* is also considered in this work as the development framework. In particular, a main contribution of the paper is the development of the model of a flexible thin beam, based on the finite element method, as a component of the *Modelica MultiBody* library [18]. The model is valid as long as the deformation field is small compared to the beam length, as it is the

case, for example, when studying the dynamics of vibrations in machining tools. However, just by exploiting the modular approach and the symbolic manipulation capabilities of the environment, several beams can be easily connected in mixed finite element-finite volume models, in order to account for large deformations.

The paper is organized as follows: Section 2 contains an introduction to the modular modelling techniques for multibody systems; in Section 3 the problem of the representation of a generic deformable body in a multibody system is introduced and the development of the equations of motions is shown; in Section 4 the *Modelica* implementation is analyzed; Section 5 contains selected simulation results; finally, in Section 6 the main results are summarized and future developments are introduced.

## 2. Object-oriented approach to multibody systems modelling

The development of the equations of motion for a multibody system is a task whose complexity increases very rapidly with the number of bodies involved.

Such problem can be effectively tackled by adopting an *object-oriented modelling approach*, which does not mean simply adopting an object-oriented programming language but involves an innovative systematic approach to the modelling task. Object-oriented modelling is then an approach to physical systems modelling based on specific paradigms:

- **A-causal modelling.** The equations of each model are written in a declarative form, i.e., by stating physical principles rather than by writing calculation algorithms, independently of the actual boundary conditions, and without deciding a-priori which are the input and which are the outputs. This results in a-causal models, described by DAE (Differential-Algebraic Equations) systems, which represent in the most natural and physically consistent way each system component. The causality of the model is determined automatically by the simulation environment at the aggregate level, when a system model is assembled out of elementary ones. In this way, models are much easier to write, document, and reuse, while the burden of determining the actual sequence of computations required for the simulation is entirely left to automatic symbolic manipulation tools.
- **Code transparency.** Equations written in a model tightly match the way they are written on the paper, so that it is very easy to understand what's inside a given model, as well as modify or enhance it.
- **Encapsulation.** The interaction among components and (sub)models is implemented only through rigorously defined interfaces, called connectors, whose design is of paramount importance. A connector is defined by a set of *effort* variables and by a set of *flow* variables and the simulation environment implements a connection by equating the effort variables and by balancing the flow variables. For example, electrical connectors carry a voltage (*effort*) and a current (*flow*), thermal connectors carry a temperature (*effort*) and a heat flow (*flow*), drive train connectors an angle (*effort*) and a torque (*flow*) and so on. As long as two different components have compatible connectors, they can be bound together, regardless of their inner details. This feature is essential for the development of libraries of re-usable models; moreover, it allows to easily replace a part of a system model with a more detailed or a more simplified one, without affecting the rest of the model.
- **Inheritance.** Model libraries can be given a hierarchical structure, in which more complex models are obtained from basic models by adding specific variables, equations or even

models. For example, it is possible to write the equations describing the flow of a generic gas in a tube; the model of the flow of a particular, e.g. 10-component ideal gas mixture can be obtained by inheriting from the general one and by adding the specific model of the ideal gas.

- **Multi-physics modelling.** The modelling scope is general, as it provides modelling primitives such as generic algebraic, differential and difference equations, and it is not tied to any particular engineering domain, such as mechanics, electrical engineering, or thermodynamics. It is then quite straightforward to model systems having a multidisciplinary nature, such as mechatronic systems, resulting from the interaction of mechanical, electrical and control sub-systems, or nuclear plants, resulting from the interaction of thermo-hydraulic, nuclear and control sub-systems.
- **Reusability.** A-causal modelling and encapsulation are a strong incentive towards the development of libraries of general-purpose reusable models for different engineering domains.

Thus, adopting such approach, the complexity of the modelling process is confined to the development of the equations for the single components and to the definition of standardized component interfaces. Constructing a model by aggregation or “connection” of simple components is both easy and intuitive, and generally results in a system layout similar to the physical one. The usefulness of such approach for multibody systems has also been showed in [9],[27] and [24].

The *Modelica* language [28] offers a suitable framework to support such approach. *Modelica* is an object-oriented language for modelling large, complex and heterogeneous physical systems [13–30] which is endowed with all the object-oriented modelling paradigms previously listed in this section. *Modelica* is suited for multi-domain modelling, including, for example, mechatronic systems in robotics, automotive and aerospace applications (with mechanical, electrical, hydraulic and control subsystems), process oriented applications and, of course, multibody systems.

The concepts behind the designed features of the *Modelica* language come from the theory of *port-based* physical system modelling, an overview of which is given in [32]. Within such theory, complex physical systems are viewed as an aggregation of simple and basic components, interacting with each other through “ports” which are associated with a generalized “power” exchange. Multibody systems modelling via linear graph [27–24], bond-graph [24–3] and acausal object-oriented techniques [28–13] can be cast within the general framework of such modelling paradigm.

As a matter of fact, the bond-graph approach and the *Modelica* approach share many common aspects, especially those regarding the definition of the objects interfaces, with a set of effort variables and of flow variables. On the other hand, the most significant difference between the bond-graph approach and the *Modelica* one is in terms of notation for the models constitutive equations: the former forces the user to adopt a quite complicated graphical notation made of *bonds* and *junctions*, while the latter lets the user write the model equation in the most natural and acausal way. Furthermore, several disadvantages of the bond-graph approach, applied to the modelling of complex 3D multibody systems, have been pointed out in [24]. These disadvantages include the use of absolute generalized velocities, the use of frame-dependent vectors, the generation of constraints at velocity level and the difficulty of implementing a causality assignment procedure for multibody graphs.

The relation between the *Modelica* approach and the linear graph approach is also very strict, with many common aspects: the flow and cross variables are used in both and the *vertex* and *circuit* postulate for the linear graph, which are the generalization of the Kirchoff’s laws

for electrical circuits, generate the same constraints between the components port variables as the connections in a *Modelica* model. Nevertheless, when it comes to 3D multibody systems, the edges and vertex notation of linear graph theory can get quite complicated, while the system representation in *Modelica* is always very clear and representative of the physical layout.

## 2.1. The *Modelica* multibody library

The *Modelica* Multibody library provides pre-defined models of 3D mechanical components to represent complex multibody systems. Though a great number of components is provided (e.g., revolute joints, prismatic joints, bodies of different shapes, springs, dampers, etc. . .), the library cannot yet handle bodies with distributed flexibility. On the other hand, it is possible to easily develop new components to be used with the pre-defined ones.

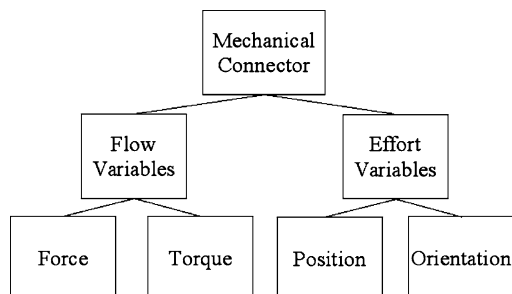
The standard interface (i.e., the *connector*) is associated with a component reference frame, so that a connection between two components is equivalent to a rigid overlapping of the two connector frames (it is the abstraction of an ideal “welding” realized at the frame location). The connector assumes the cut force and torque as flow variables, while the effort variables are given by the position of the origin of the connector frame with respect to the world frame and by an “orientation object”, describing the relative orientation between the world frame and the connector frame (Figure 1).

The orientation object is designed to efficiently represent the body relative orientation in any dynamic condition. The internal representation of the orientation (i.e., the parametrization of the relative rotation matrix) is completely masked to the user. In addition to that, the object is endowed with methods to compute rotation matrices and angular velocities, to rotate vectors, etc., so that knowledge about the actual description form of orientation is not necessary.

The linear velocity and acceleration and the angular acceleration can be simply computed by applying the derivative operator  $\text{der}()$  to the corresponding variables.

Multibody models can then be obtained by simple *drag&drop* of components from the library and with suitable connections; a *world* component must also be present in every model, to define the reference frame and the type of gravity field (no gravity, uniform gravity or point gravity). An example is shown in Figure 2, which depicts the model for a common kinematic loop, often used as basic constructing unit in mechanisms, consisting of six revolute joints, one prismatic joint and four bars. This example demonstrates that usually no particular knowledge of the user is needed to handle kinematic loops. It is just necessary to connect the joints and bodies together according to the real system.

**Fig. 1** Mechanical connector scheme



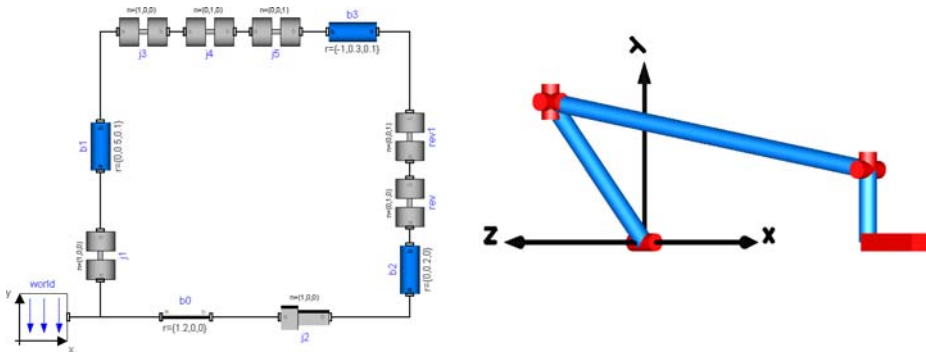


Fig. 2 Four bar mechanism: components connection scheme (left) and animation screenshot (right)

2.2. The simulation environment

In order to simulate the system, it is necessary to translate the *Modelica* code into some different form that can be directly executed.

The mathematical simulation problem consists of the equations of all the components of the model and the equations generated by connections. The complete model is then usually represented by a system of differential-algebraic equations (DAE). The direct integration of such systems is very hard or even impossible, e.g., if the DAE system has index greater than one [22]. Any *Modelica*-based simulation environment must then be equipped with automatic symbolic manipulation capabilities [4] in order to reduce the equations system to a form that can be easily solved.

The generation of a numerically efficient procedural form from the acausal description of the model in the *Modelica* language is carried out in several steps. First, the object-oriented code is “flattened” in a set of constants, variables, functions and equations and the connection equations are generated. Then, after the conversion of the system of equations in BLT (*Bounded Lower Triangular*) form, an algebraic simplification follows, removing the trivial equations and resulting in a minimal set of equations. If needed index reduction is also performed [19–14]. When the system of equations has been reduced to a DAE of index one or to a ODE form, it can be simulated with efficient, state-of-the-art numerical integration algorithm.

Different tools, both commercial and open source, are available to carry such steps [5],[11],[12]. Among these, Dymola is one of the most advanced and has been chosen to perform all the simulation reported in the paper. However, it should be pointed out that all the models developed are completely portable and compatible with any *Modelica*-based simulation environment.

Dymola offers a special support for multibody models, allowing the connection of components in a nearly arbitrary fashion. If kinematical loop structures occur, they are automatically handled in an efficient way by a special symbolic manipulation techniques. Several numerical integration algorithms are available to perform the simulations, both variable step (DASSL [22], Radau [8], etc.) and fixed step (Euler and Runge-Kutta of different orders).

In addition to that, Dymola features a graphical editor for *Modelica* models, so that a model can be defined by drawing a composition diagram by positioning icons that represent the models of the components, drawing connections and giving parameter values in dialogue boxes. Furthermore, whenever a multibody system model is constructed, all defined components

are automatically visualized with a 3D animation, using appropriate user-definable sizes and colors, allowing an easy visual check of the constructed model, without extra work of the modeler.

### 3. Motion equations in the local coordinate frame

In the previous section, it has been shown that the adoption of a fully modular approach for multibody systems modelling requires the description of the dynamics of every single body in terms of local variables, while the interaction between different bodies can be described using connectors. This approach leads to a formulation of the equations of motion in terms of a local coordinate system rather than of a global coordinate system, the reason being the fact that the local frame fits naturally to the connector concept. The equation of motion for a generic flexible body will thus be formulated with respect to a local reference frame rigidly attached to the body itself.

Consider a generic deformable body in a multibody system (Figure 3). The position, in local coordinates, of a point on the body has the following expression:

$$\bar{\mathbf{u}} = \bar{\mathbf{u}}_0 + \bar{\mathbf{u}}_f, \tag{1}$$

where  $\bar{\mathbf{u}}_0$  is the “undeformed” (i.e., rigid) position vector and  $\bar{\mathbf{u}}_f$  is the deformation contribution to position (i.e., the deformation field).

The formal and mathematically sound description of the generic deformation of a body requires the deformation field to belong to an infinite dimensional functional space, requiring, in turn, an infinite number of deformation degrees of freedom.

In this paper, the deformation field is described by an approximation of the functional basis space it belongs to, supposing such space has a finite dimension, say  $M$ , so that the vector  $\mathbf{u}_f$  can be expressed by the following finite dimensional product:

$$\bar{\mathbf{u}}_f = \mathbf{S} \mathbf{q}_f, \tag{2}$$

where  $\mathbf{S}$  is the  $[3 \times M]$  shape functions matrix (i.e., a matrix of functions defined over the body domain and used as a basis to describe the deformation field of the body itself) and  $\mathbf{q}_f$  is the  $M$ -dimensional vector of deformation degrees of freedom.

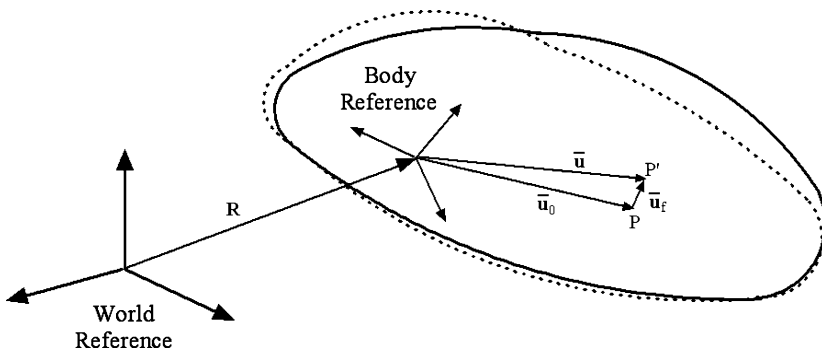
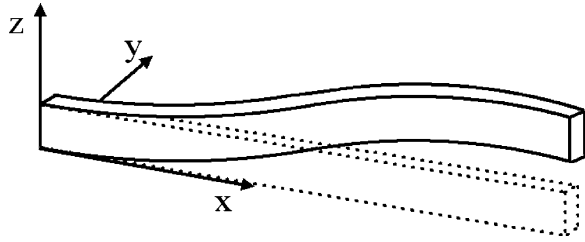


Fig. 3 Flexible body reference systems

Fig. 4 Planar beam deformation



The position of a point on a deformable body can then be expressed in world reference as follows:

$$\mathbf{r} = \mathbf{R} + \mathbf{A}\mathbf{u} = \mathbf{R} + \mathbf{A}\mathbf{u}_0 + \mathbf{A}\mathbf{S}\mathbf{q}_f, \tag{3}$$

where  $\mathbf{R}$  is the vector identifying the origin of the body local reference system and  $\mathbf{A}$  is the rotation matrix for the body reference system.  $\mathbf{R}$  and  $\mathbf{A}$  can be easily obtained from the connector structure.

The representation of a generic deformable body in world reference requires then  $6 + M$  d.o.f. (i.e., 6 corresponding to rigid displacements and rotations and  $M$  to deformation fields):

$$\mathbf{q} = [\mathbf{q}_r \ \mathbf{q}_f]^T = [\mathbf{R} \ \boldsymbol{\theta} \ \mathbf{q}_f]^T, \tag{4}$$

where  $\boldsymbol{\theta}$  represents the undeformed body orientation angles and  $\mathbf{q}_r$  is a vector containing the 6 rigid degrees of freedom.

The equations of motion for a generic flexible body in a multibody system can then be developed applying the principle of virtual work [16]. It should be pointed out that the same results could be obtained using the classical Lagrangian approach (as in, e.g., [26]), though such approach is quite difficult to use in practice, due to the complexity of the required analytical differentiation of the kinetic energy expression.

The equations of motion, expressed in body axes, result in:

$$\begin{bmatrix} \mathbf{m}_{RR} & \tilde{\mathbf{S}}_t^T & \tilde{\mathbf{S}} \\ \tilde{\mathbf{I}}_{\theta\theta} & \tilde{\mathbf{I}}_{\theta f} & \\ & \mathbf{m}_{ff} & \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{R}} \\ \ddot{\boldsymbol{\alpha}} \\ \ddot{\mathbf{q}}_f \end{bmatrix} = \begin{bmatrix} \mathbf{O}_3 \\ \mathbf{O}_3 \\ -\mathbf{K}_{ff} \mathbf{q}_f \end{bmatrix} + \begin{bmatrix} \mathbf{Q}_v^R \\ \mathbf{Q}_v^\theta \\ \mathbf{Q}_v^f \end{bmatrix} + \begin{bmatrix} \bar{\mathbf{Q}}_e^R \\ \bar{\mathbf{Q}}_e^\theta \\ \bar{\mathbf{Q}}_e^f \end{bmatrix}, \tag{5}$$

where  $\ddot{\mathbf{R}}$ ,  $\ddot{\boldsymbol{\alpha}}$ ,  $\ddot{\mathbf{q}}_f$  are the body linear, angular and deformation accelerations in the local reference frame, respectively.

The details of the calculations and the definition needed to develop Equations (5) are reported in Appendix A.

Equations (5) are valid for a general deformable body, though many of the quantities involved (e.g., the matrix  $\mathbf{K}_{ff}$ ) depend on specific body characteristics such as the shape or the material properties.

From now on, the case of a *thin beam* will be considered. In detail, it will be assumed that the body is a 1D elastic *continuum* with constant cross-sectional properties. Furthermore, it will be assumed that the beam constitutive material is homogeneous, isotropic and perfectly elastic (i.e., the elastic internal forces are conservative). Finally, it will be assumed that the deformation field is restricted to lie within the  $xy$  plane of the beam local reference system (Figure 4).



It should be pointed out that such assumptions do not restrict the model validity or generality, since the model remains still representative for a large number of dynamic simulation applications (e.g., almost all the flexible robots commonly studied have flexible links which can be represented by such model [29]).

### 3.1. The element point of view

The finite element method is based upon a discretisation of the beam into  $N$  elements. A single element can itself be viewed as a thin beam characterized by a planar deformation field. It is then possible to define the local dimensionless *abscissa* as  $\xi = x/\ell$ , where  $x$  is the longitudinal local coordinate and  $\ell$  is the element length.

In [34] it is shown that the partial differential equations associated with the deformation problem at hand, under the hypothesis of elastic constitutive law for the material, require, for a consistent finite element formulation, the use of linear and Hermite cubic polynomials for the approximation of the axial and transversal deformation field, respectively. Thus, for a single element, the generic equations of motion (5) can be expanded as follows:

$$\begin{aligned} \bar{\mathbf{u}}_{f,el} &= [\bar{u}_{f1,el} \ \bar{u}_{f2,el} \ \bar{u}_{f3,el}]^T = \mathbf{S}_{el} \mathbf{q}_{f,el}, \\ \mathbf{S}_{el} &= \begin{bmatrix} 1 - \xi & 0 & 0 & \xi & 0 & 0 \\ 0 & 1 - 3\xi^2 + 2\xi^3 & \ell(\xi - 2\xi^2 + \xi^3) & 0 & 3\xi^2 - 2\xi^3 & \ell(\xi^3 - \xi^2) \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{S}_{el1} \\ \mathbf{S}_{el2} \\ \mathbf{S}_{el3} \end{bmatrix}, \\ \mathbf{q}_{f,el} &= [q_{f1,el} \ q_{f2,el} \ q_{f3,el} \ q_{f4,el} \ q_{f5,el} \ q_{f6,el}]^T, \end{aligned} \tag{6}$$

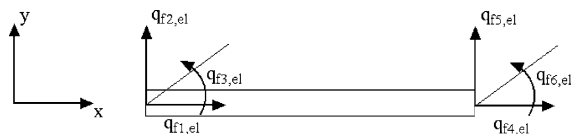
where the subscript “ $el$ ” is used to refer the quantities to a single element.

Figure 5 depicts the element coordinate systems associated with the deformation degrees of freedom:  $q_{f1,el}$  and  $q_{f4,el}$  are associated with axial compression,  $q_{f2,el}$  and  $q_{f5,el}$  with transversal displacement and  $q_{f3,el}$  and  $q_{f6,el}$  with beam ends rotation.

Since the third row of the shape matrix  $\mathbf{S}_{el}$  is composed only by zeros, it could be noted that, despite the fact that the motion equations have been developed for a general 3D case, the deformation field is assumed to lie within the local  $xy$  plane.

The planar deformation hypothesis and the assumption of a homogeneous, isotropic and elastic material for the beam, allow to exploit the Euler-Bernoulli theory and to calculate the elastic potential energy  $U_{el}$ , neglecting the contribution of shear stresses and

**Fig. 5** Element coordinate systems



considering only the work of the resulting axial force  $N_{el}$  and bending moment  $M_{el}$ , as follows [34]:

$$\begin{aligned}
 U_{el} &= \frac{1}{2} \int_{\ell} \left( \frac{N_{el} N'_{el}}{EA} + \frac{M_{el} M'_{el}}{EJ} \right) dx = \frac{1}{2} \int_{\ell} (EJ \bar{u}''_{f2,el} + EA \bar{u}'^2_{f1,el}) dx \\
 &= \frac{1}{2} \mathbf{q}_{f,el}^T \mathbf{K}_{ff,el} \mathbf{q}_{f,el}, \tag{7}
 \end{aligned}$$

where  $E$  is the material Young’s modulus,  $A$  is the (constant) cross-sectional area and  $J$  is the (constant) cross-sectional second moment of area. The analytical expression for the case at hand for the matrix  $\mathbf{K}_{ff,el}$ , usually known as the structural stiffness matrix, is reported in Appendix B.

### 3.2. Finite element method equations assembly

The equations of motion for the entire beam can be obtained by assembling the equations of motion for beam elements as the one defined in the previous subsection. The body reference system will be the local reference system located at the root of the first element, so that the rigid degrees of freedom, common to all the elements, will be referred to such coordinate system.

Let then  $m$  and  $L$  be the mass and length of the entire beam, and  $N$  the number of elements to be used, so that  $\ell = L/N$ . Indicating with  $\hat{x}$  the reference system unit vector along the beam axis, the expression of the generic position  $\bar{\mathbf{u}}_j$  of a point of element  $j$  can be expressed as:

$$\bar{\mathbf{u}}_j = \bar{\mathbf{u}}_{0j} + \mathbf{S}_{el} \mathbf{B}_j \mathbf{q}_f = [\xi_j \ell + (j - 1)\ell] \hat{x} + \mathbf{S}_{el} \mathbf{B}_j \mathbf{q}_f, \tag{8}$$

where  $\bar{\mathbf{u}}_{0j}$  is the position of the root of the  $j$ th element,  $\mathbf{S}_{el}$  is the shape functions matrix defined by (6),  $\mathbf{B}_j$  is the so-called *connectivity matrix* and  $\mathbf{q}_f$  is a vector containing the deformation degrees of freedom for the whole beam.

The matrices  $\mathbf{B}_j$  have the following form:

$$\mathbf{B}_j = [ \mathbf{O}_{6,3(j-1)} | \mathbf{I}_6 | \mathbf{O}_{6,3(N-j)} ], \quad \forall j = 1, \dots, N. \tag{9}$$

The connectivity matrices are used to relate the vector  $\mathbf{q}_f$ , which contains the deformation degrees of freedom for the whole beam, to the corresponding  $j$ th element, according to the expression:

$$\mathbf{q}_{f,el_j} = \mathbf{B}_j \mathbf{q}_f. \tag{10}$$

The dynamics of the complete flexible beam can then be described by Equation (5), using the following expressions:

$$\begin{aligned}
 \bar{\mathbf{S}} &= \sum_{j=1}^N \frac{m}{L} \int_{V_j} \mathbf{S}_{el} \mathbf{B}_j dV_j, \\
 \bar{\mathbf{S}}_t &= \sum_{j=1}^N \frac{m}{L} \int_{V_j} \bar{\mathbf{u}}_j dV_j,
 \end{aligned}$$

$$\begin{aligned}
 \bar{\mathbf{I}}_{\theta\theta} &= \sum_{j=1}^N \frac{m}{L} \int_{V_j} \begin{pmatrix} \bar{u}_{2f_j}^2 & -\bar{u}_{2f_j}\bar{u}_{1j} & 0 \\ & \bar{u}_1^2 & 0 \\ & & \bar{u}_1^2 + \bar{u}_{2f_j}^2 \end{pmatrix} dV_j, \\
 \bar{\mathbf{I}}_{\theta f} &= \sum_{j=1}^N \frac{m}{L} \int_{V_j} \begin{pmatrix} \mathbf{O}_{(3N,1)} \\ \mathbf{O}_{(3N,1)} \\ \bar{u}_{1j}\mathbf{S}_{el2} - \bar{u}_{2j}\mathbf{S}_{el1} \end{pmatrix} dV_j, \\
 \mathbf{m}_{ff} &= \sum_{j=1}^N \frac{m}{L} \mathbf{B}_j^T \left( \int_{V_j} \mathbf{S}_{el}^T \mathbf{S}_{el} dV_j \right) \mathbf{B}_j \\
 \mathbf{K}_{ff} &= \sum_{j=1}^N \mathbf{B}_j^T \mathbf{K}_{ff,el} \mathbf{B}_j \\
 \mathbf{Q}_v^f &= - \sum_{j=1}^N \frac{m}{L} \int_{V_j} [\mathbf{B}_j^T \mathbf{S}_{el}^T (\bar{\omega}^2 \bar{\mathbf{u}}_j + 2\bar{\omega} \mathbf{S}_{el} \mathbf{B}_j \dot{\mathbf{q}}_f)] dV_j. \tag{11}
 \end{aligned}$$

The computation of the above terms can be easily carried out by observing that the integral of a generic quantity  $\mathcal{F}$ , varying along the beam, onto the volume of a single element can be computed as follows:

$$\int_{V_j} \rho \mathcal{F} dV_j = \frac{m}{L} \int_0^1 \ell \mathcal{F}(\xi) d\xi = \frac{m}{N} \int_0^1 \mathcal{F}(\xi) d\xi. \tag{12}$$

### 3.3. Deformation boundary conditions

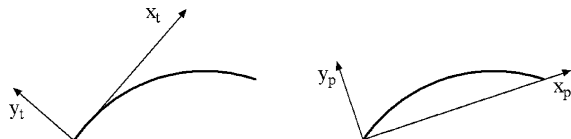
The equations of motion for the whole beam must be completed by enforcing suitable boundary conditions for the finite element approximation of the deformation partial differential equations. That means assuming prescribed values for some of the deformation displacements, rotations and velocities (linear or angular) at the body boundaries which are, for the case at hand, the beam root and tip.

The most commonly used boundary conditions for flexible beams are of two kinds, commonly associated with two different reference system: the *tangent* frame and the *pinned* frame condition (Figure 6).

In both cases six conditions are given: the *tangent* one enforce null deformation at the beam root (i.e.,  $q_{f1}, q_{f2}, q_{f3}, \dot{q}_{f1}, \dot{q}_{f2}, \dot{q}_{f3}$  equal to zero for the first element), while the *pinned* one enforce null axial and transversal displacement at the beam root (i.e.,  $q_{f1}, q_{f2}, \dot{q}_{f1}, \dot{q}_{f2}$  equal to zero for the first element) and transversal displacement at the beam tip (i.e.,  $q_{f5}$  and  $\dot{q}_{f5}$  equal to zero for the last element).

The choice between the two set of conditions largely depends on the problem at hand.

**Fig. 6** Tangent (left) and pinned (right) reference systems



Boundary conditions can be enforced into Equations (5) with suitable modifications of the connectivity matrices  $\mathbf{B}_1$  and  $\mathbf{B}_N$ , by zeroing some entries. For example, for the *tangent* reference conditions,  $\mathbf{B}_N$  remains unvaried and  $\mathbf{B}_1$  becomes

$$\mathbf{B}_1 = \left[ \begin{array}{c|c} \mathbf{O}_{3,3} & \mathbf{O}_{3,3} \\ \mathbf{O}_{3,3} & \mathbf{I}_3 \end{array} \middle| \mathbf{O}_{6,3(N-1)} \right]. \tag{13}$$

### 3.4. Beam substructuring

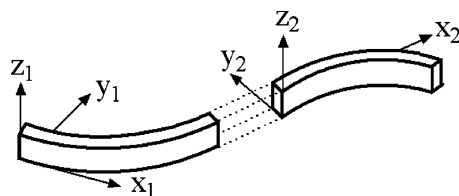
In the finite element formulation for the equation of motion for a flexible beam, the reference directions of the internal actions are the same for all the elements. Such representation is acceptable as long as the deformation field is small compared to the beam length.

On the other hand, when large deformations are involved, the internal actions reference directions should change according to the deformation field. That means that it is necessary to define a local reference system for each element (Figure 7). This corresponds to the application of the finite volume method to assemble the equations of motion defined over each element (i.e., over each volume). This representation is valid for large beam deformation, accounting also for geometric non-linearity, as long as the deformation field is small compared to the volumes length. Furthermore, it is possible to assemble the equation of motion for a mixed (finite element-finite volume) formulation by dividing every volume into several elements.

Such discretization technique is known in the scientific literature as the “beam substructuring” method, so that a volume containing one or more elements is generally referred to as a “substructure” [33]. Using this approach, it is usually necessary to explicitly formulate the “substructure compatibility constraints”, which are the motion constraints arising from the compatibility conditions to be imposed at substructures interfaces; such constraints are then manually added, using the Lagrange multipliers technique, to the substructure dynamic equations which are generally in the form of (5).

The beam substructuring method accounts then for geometric non-linearity by requiring to explicitly formulate a complex mixed system of differential algebraic equations of motion for the unconstrained flexible beam [33]. This is not true any more when adopting a fully modular approach as the one proposed in this work: the beam substructuring method can be easily employed without explicitly formulating the compatibility constraint for each substructure. As a matter of fact, when composing the beam model by *connecting* in a *Modelica* sense several substructures, the “substructure compatibility constraints” are automatically generated by the connection equations and the equations of motion for the full beam are automatically calculated with the aid of symbolic manipulation algorithms, as it will be shown in Section 4. This means that by exploiting the features of a fully modular approach for the finite element beam model, the beam substructuring model can be obtained completely effortlessly, thus significantly shortening the model development time and ruling out the possibility of

**Fig. 7** Volume coordinate systems



inserting coding bugs while setting up by hand the complex mixed system of differential algebraic equations of motion explicitly incorporating the substructures constraints.

#### 4. Modelica implementation

The finite element formulation has been implemented using the *Modelica* language, creating thus a new component, called *FlexBeamFEM* (Figure 8). The component interfaces are two standard mechanical flanges (conventionally named *a* and *b*) from the *MultiBody* library [18]. The component is fully compatible with the library, so that it is possible to connect directly the flexible beam component with pre-defined models such as mechanical constraints (revolute joints, prismatic joints, etc.), parts (3D rigid bodies) and forces elements (springs, dampers, forces, torques).

In detail, the flexible beam component uses two mechanical connectors (named flange *a* and *b*) as physical representation of the two ends of the beam while the motion is ruled by equations (5), with addition of a damping term ( $-\mathbf{D}_{ff}\dot{\mathbf{q}}_f$ ) for the structural dynamics part. The damping term is added to model the dissipative properties of the material and is calculated as  $\mathbf{D}_{ff} = \alpha\mathbf{m}_{ff} + \beta\mathbf{K}_{ff}$ , where  $\alpha$  and  $\beta$ , the so-called *Rayleigh* damping coefficients, depend on the specific beam characteristics.

The local reference frame is conventionally located at flange *a*, and all quantities used in the equations of motion, such as the linear and angular velocities and accelerations ( $\dot{\mathbf{R}}, \dot{\omega}, \ddot{\mathbf{R}}, \ddot{\alpha}$ ) and the relative rotation matrix ( $\mathbf{A}$ ) are obtained from the connectors variables. The external actions (i.e., the terms  $\mathbf{Q}_e^R, \mathbf{Q}_e^\theta$ , and  $\mathbf{Q}_e^f$ ) are computed from the connectors variables as well, accordingly to the following equations:

$$\mathbf{Q}_e^R = \mathbf{F}_a + \mathbf{F}_b^a, \tag{14}$$

$$\mathbf{Q}_e^\theta = \tau_a + \tau_b^a + ([L \ 0 \ 0]^T + \mathbf{S}_{el}(1)\mathbf{B}_N\mathbf{q}_f) \times \mathbf{F}_b^a, \tag{15}$$

$$\mathbf{Q}_e^f = (\mathbf{F}_b^{aT} \mathbf{S}_{el}(1)\mathbf{B}_N + \tau_a^T \mathbf{S}_{d0}\mathbf{B}_1 + \tau_b^{aT} \mathbf{S}_{d1}\mathbf{B}_N)^T. \tag{16}$$

where  $\mathbf{F}_a, \mathbf{F}_b^a, \tau_a, \tau_b^a$  are the forces and moments at the beam ends (expressed in the local frame), while  $\mathbf{S}_{d0}$  and  $\mathbf{S}_{d1}$  are matrices used to select the flanges moments acting on the deformation field.  $\mathbf{F}_a$  and  $\tau_a$  are directly extracted from the flange *a* connector variables, while  $\mathbf{F}_b^a$  and  $\tau_b^a$  are obtained from the flange *b* variables (i.e., from  $\mathbf{F}_b$  and  $\tau_b$  which are expressed in the flange *b* reference frame) by pre-multiplication with a suitable relative rotation matrix.

The model parameters include the beam length and cross sectional area, the material density and Young modulus, the cross sectional inertia, the damping factor and the number of elements.

Particular care has been put into the implementation of automatic 3D animations for the models (Figure 9). The 3D visualization has revealed itself to be an important feature, giving significant insight and sensible feedback about the dynamical behavior of the model.

The modular approach adopted in the development of the finite element model significantly simplifies the implementation of the beam substructuring model, obtained by

Fig. 8 Component icon



**Fig. 9** Animation of a cart with flexible inverted pendulum



connecting several finite element beams, each one composed by one or more elements. The *connection* can be either drawn graphically or directly written within the code using a single instruction, as in the following example:

```
connect (SubStructure1.flangeB, SubStructure2.flangeA) ;
```

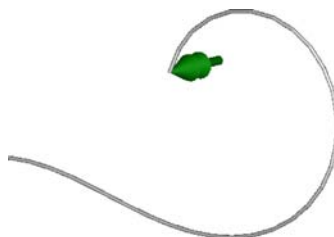
where the `connect` instruction means equating the connectors effort variables (i.e., position and orientation) and balancing the flow variables (i.e., forces and torques). The assembly of the equations of motion for the whole beam, resulting from the equations of motion of the single substructures and from the connection equations, is demanded to Dymola, exploiting its advanced symbolic manipulation techniques.

The dynamical properties of the assembled models are significantly complex and accurate, featuring a displacement description which is fully non-linear and allowing the simulation of large displacement due to deformation (Figure 10) at the cost, though, of a significant increase of the computational complexity with respect to the “pure” finite element model.

## 5. Simulation results

The different flexible beam models have been thoroughly tested and validated by several simulation analysis performed within the Dymola simulation environment. The following subsections contain a few significative simulation examples which have been chosen just to validate the proposed modular model with respect to reference literature results. However,

**Fig. 10** Large deformation of a thin beam



**Table 1** Theoretical and model natural frequencies

Mode	Freq.* [Hz]	Freq.† [Hz]	Error [%]
1	2.0854733	2.0854750	8.418e-005
2	13.0694381	13.0698705	3.308e-003
3	36.5948052	36.6041219	2.545e-002
4	71.7112127	71.7795490	9.529e-002
5	118.543772	118.842591	2.521e-001

\* Theoretical prediction

† Simulation result

it should be pointed out that the real new contribution here proposed is mainly constituted by the modular nature of the model, which in turn allows an easy and almost effortless set-up (literally by drag-and-drop and connection of components) of *any* desired simulation scenario.

All the simulations have been performed on a common PC equipped with a Pentium IV 3 GHz and with 1 GB RAM.

### 5.1. Free vibration

In this simulation the free vibration of a flexible beam is analyzed. The test-case has been set up in order to investigate the models properties with respect to theoretical predictions.

The beam component is connected to the world reference system, so that no rigid motion is allowed; furthermore, no gravity field is considered. At the initial time instant the beam is standing still with a non-null tip displacement, then it evolves, vibrating, towards steady state.

The beam, made by aluminium, has square cross section  $A = 1 \text{ cm}^2$ , length  $L = 2 \text{ m}$ , density  $\rho = 2700 \text{ kg/m}^3$ , Young's modulus  $E = 72 \cdot 10^9 \text{ N/m}^2$  and has been discretised with  $N = 10$  elements. The initial tip displacement is 1 cm.

The problem of calculating the vibration frequencies of a flexible beam clamped at the root has been widely investigated and well-assessed numerical solution are available [16].

Table 1 contains a comparison between the results for the first five vibrational modes obtained by simulation and by solving the problem numerically. The results are in good accordance, as it is shown also in Figure 11, depicting the tip displacement frequency spectrum.

### 5.2. Flexible pendulum

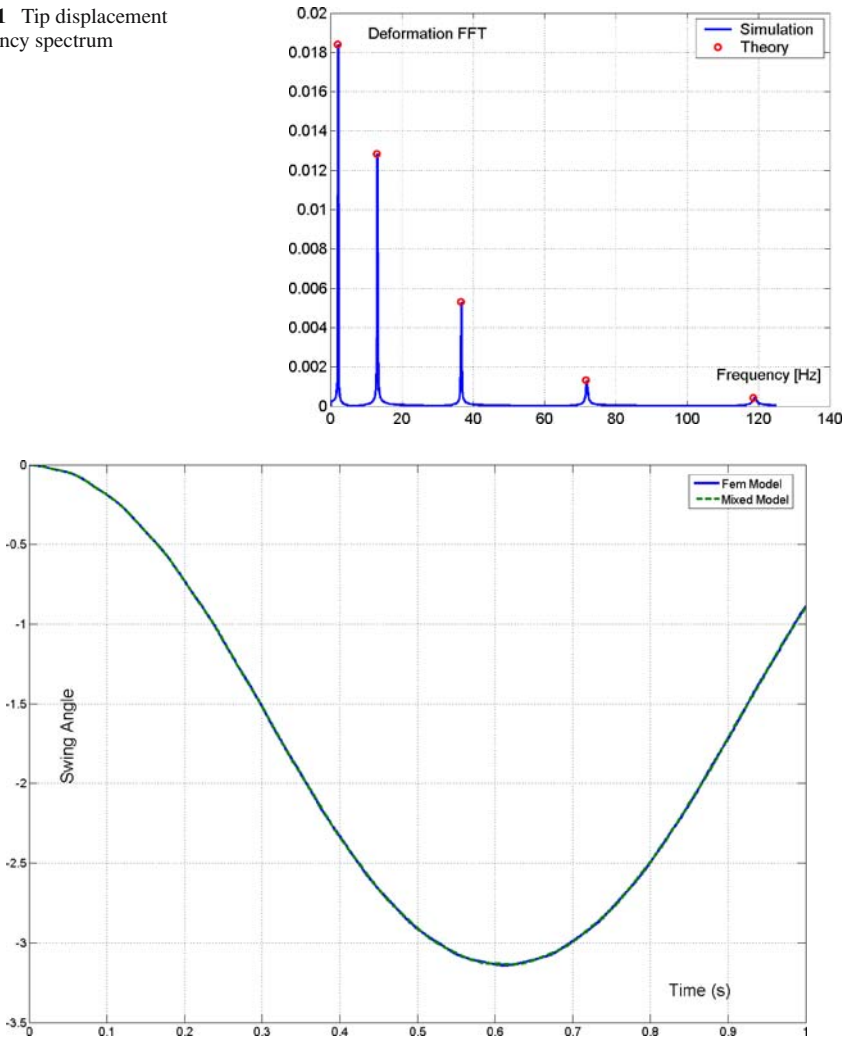
This simulation, reported also in [6], involves the analysis of the vibrations induced by motion in a flexible pendulum swinging under the action of gravity.

The pendulum, connected to the world reference system by a revolute joint, has a length  $L = 0.4 \text{ m}$ , cross sectional area  $A = 18 \text{ cm}^2$ , density  $\rho = 5540 \text{ kg/m}^3$ , second moment of area  $J = 1.215 \cdot 10^{-8} \text{ m}^4$  and modulus of elasticity  $E = 10^9 \text{ N/m}^2$ . Two different models have been simulated: the first one composed by 10 elements and the second one by 5 sub-structures (i.e. volumes) with 2 elements each.

In both cases, the integration algorithm is DASSL with a tolerance of  $10^{-6}$  requiring 1 s and 1.03 s of CPU time, respectively, to simulate 1 s.

In Figure 12 the swing angle is depicted for both cases.

**Fig. 11** Tip displacement frequency spectrum



**Fig. 12** Swing angle

The tip deformation, depicted in Figure 13, appears to be slightly different for the two models. The results reported in [6] are in accordance with the ones obtained with the mixed model, though.

### 5.3. Elastic Slider-Crank mechanism

The simulation of an elastic slider-crank mechanism, reported also in [6], has been performed to validate the models for use within closed-loop mechanical chains. The simulation set up involves a slider, a rod and a crankshaft connected by revolute joints (Figures 14 and 15).

The crank has length  $L = 0.152$  m, cross sectional area  $A = 0.7854$  cm<sup>2</sup>, second moment of area  $J = 4.909 \cdot 10^{-10}$  m<sup>4</sup>, density  $\rho = 2770$  kg/m<sup>3</sup> and modulus of elasticity  $E = 10^9$  N/m<sup>2</sup>. The connecting rod has the same physical parameters of the crank, apart



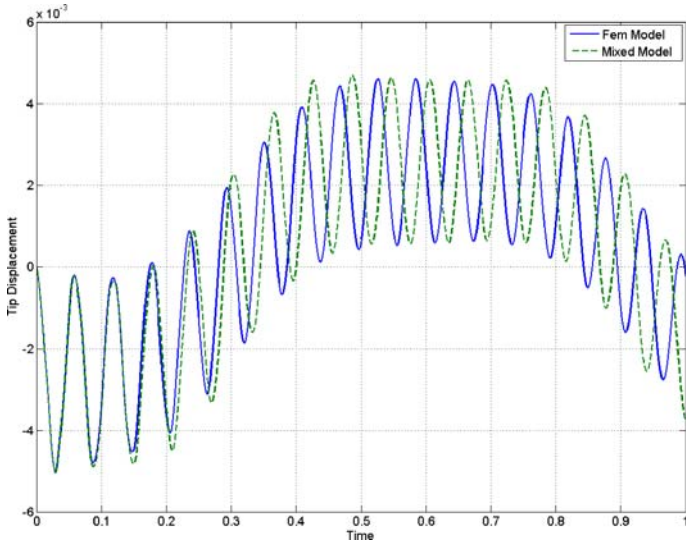


Fig. 13 Tip displacement

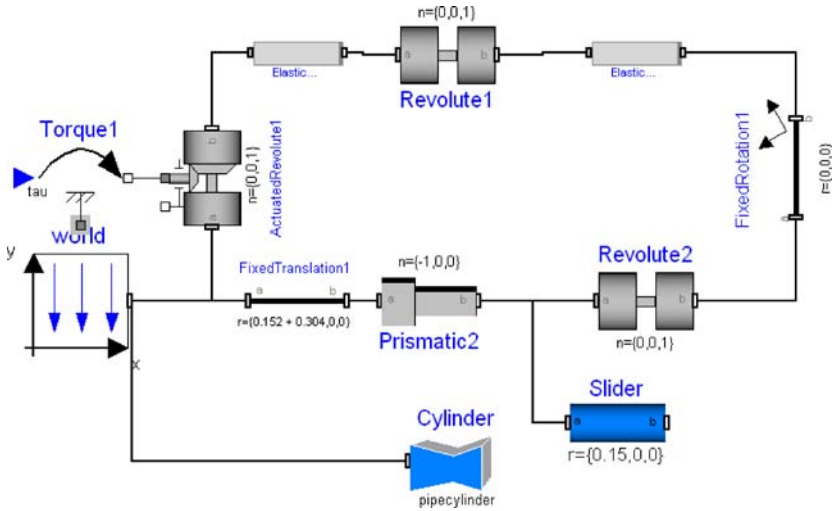


Fig. 14 Slider-crank mechanism (Dymola scheme)

from the length  $L = 0.304$  m and the Young’s modulus  $E = 5 \cdot 10^7$  N/m<sup>2</sup>. The crank and the connecting rod have been discretised with 3 and 8 elements, respectively. Finally, the slider block has been assumed to be a massless rigid body.

During the simulation, the crankshaft is driven by a torque with the following law:

$$\begin{cases} M(t) = [0.01 (1 - e^{-t/0.167})] \text{Nm}, & t \leq 0.7 \text{ sec} \\ 0, & t > 0.7 \text{ sec} \end{cases} \quad (17)$$

Fig. 15 Slider-crank mechanism

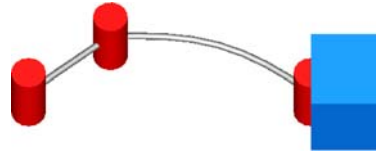
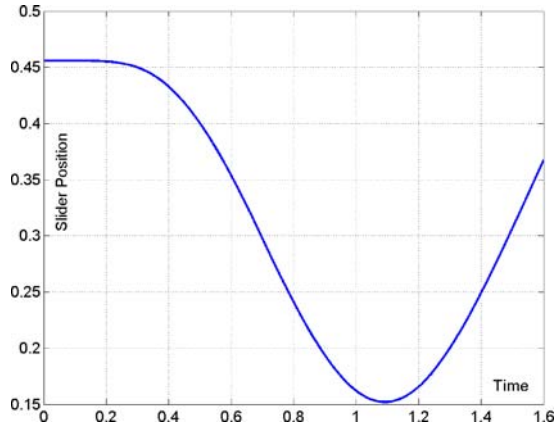


Fig. 16 Slider block position



The integration algorithm used is Radau with a tolerance of  $10^{-6}$  requiring 16.12 s of CPU time to simulate 1.6 s. However, replacing the elastic crank with a rigid one, the simulation time shortens to 3.01 s of CPU time, with results almost undistinguishable from the previous case.

Figures 16 and 17 show the slider position and the connecting rod tip transverse displacement, respectively, when performing the simulation with the elastic crank. The results are in accordance with those reported in [6].

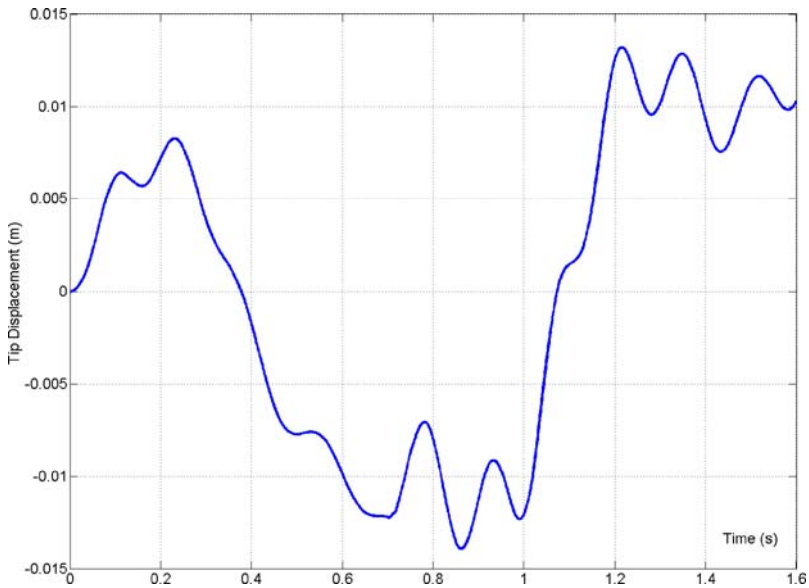
### 5.4. Centrifugal stiffening

The correct representation of the centrifugal stiffening effect can be very important, especially when simulating rotating beams at high angular velocities [1,2,15,20].

The finite-element model presented in the paper is not able to represent the centrifugal stiffening effect, due to the linear strain-displacement relationship adopted in the definition of the elastic potential energy in Equations (7). On the other hand, when considering the substructuring formulation the strain-displacement relationship for the whole beam (i.e. for the assembled volumes) is actually non-linear, allowing thus the representation of the centrifugal stiffening effects by using such formulation.

The reference simulation, the so-called “spin-up manoeuvre”, is taken from [2]: a flexible beam rotating with a prescribed angle about an axis passing through one of its ends. The rotation angle is given from the following law:

$$\theta(t) = \begin{cases} \frac{\Omega}{T} \left[ \frac{t^2}{2} + \left( \frac{T}{2\pi} \right)^2 \left( \cos \left( \frac{2\pi t}{T} \right) - 1 \right) \right], & t < T \\ \Omega (t - T/2), & t \geq T \end{cases}, \quad (18)$$



**Fig. 17** Transverse displacement of the tip of the connecting rod

representing a spin-up maneuver starting at time  $t = 0$  and ending at time  $t = T$ ; at the ending time a constant angular velocity  $\Omega$  is reached.

The considered beam has length  $L = 8$  m, cross sectional area  $A = 7.299 \cdot 10^{-5} \text{ m}^2$ , second moment of area  $J = 8.214 \cdot 10^{-9} \text{ m}^4$ , density  $\rho = 2767 \text{ kg/m}^3$  and modulus of elasticity  $E = 6.895 \cdot 10^{10} \text{ N/m}^2$ .

The discretisation is based on substructures with a single element; the number of substructures employed is different for each of the simulation experiments reported, depending on the value for the target velocity  $\Omega$ .

The tip deflection for three experiments corresponding to  $\Omega = 2$  rad/s with five volumes,  $\Omega = 4$  rad/s with five volumes and  $\Omega = 10$  rad/s with ten volumes are reported in Figure 18 ( $T = 15$  s in all cases): the results are in good accordance with the ones reported in [2].

The simulations have been performed using the DASSL integration algorithm, requiring 0.062 s, 0.094 s, 0.813 s of CPU time, respectively.

## 6. Conclusions and future work

In this paper, a modular model for flexible beams is introduced. The model is implemented with the acausal object-oriented language *Modelica* and is fully compatible with the *Modelica MultiBody* library for rigid bodies.

The discretisation scheme for the flexible beam is based on the application of the finite element method; exploiting the modular nature of the model a mixed formulation (finite volume-finite element) has been developed.

Selected simulation results have been presented in order to validate the model properties with respect to scientific literature reference cases.

Future work will include the model extension to handle full 3D deformation and distributed loads.

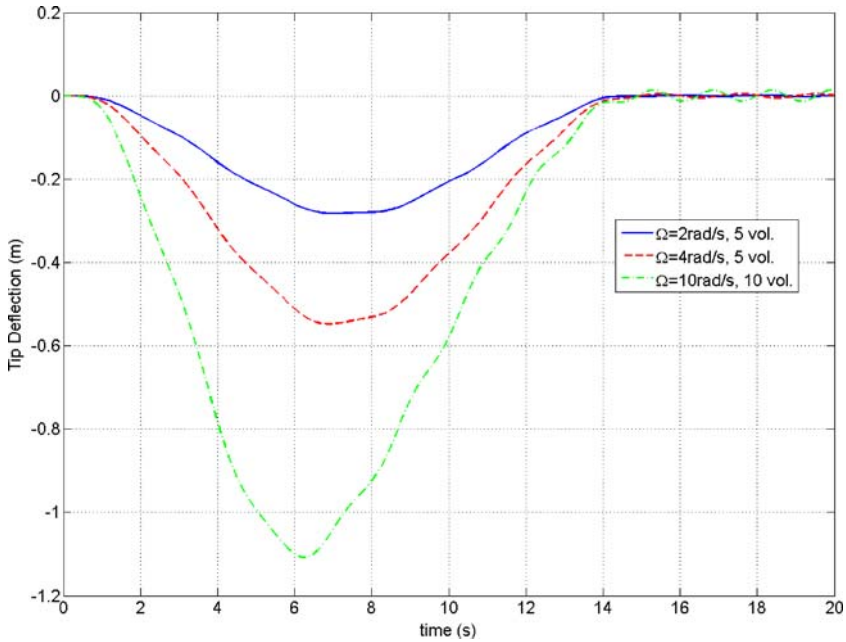


Fig. 18 Tip deflection in the spin-up maneuver of a rotating beam

A non-linear description of strain-displacement relationship, using exact curvature instead of second derivative approximation, will be considered, so as to better represents the centrifugal stiffening effect.

The model will also be further validated with experimental results and is currently being employed for the development of applications in the field of robot control and satellite attitude control.

**Appendix A: Development of the equations of motion**

The equations of motion for a generic flexible body in a multibody system can be developed applying the principle of virtual work [16], stating that the virtual work of the inertial forces  $\delta W_i$  must counterbalance the sum of the virtual work of the *continuum* elastic forces  $\delta W_s$  and of the external ones  $\delta W_e$ :

$$\delta W_i = \delta W_s + \delta W_e. \tag{19}$$

Note that, in case  $\delta W_i = 0$ , the problem reduces to the well-known problem of structural statics [34].

The terms of Equation (19) are defined as follows:

$$\delta W_i = \int_V \rho \delta \mathbf{r}^T \ddot{\mathbf{r}} dV, \tag{20}$$

$$\delta W_s = - \int_V \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV, \tag{21}$$

$$\delta \mathbf{W}_e = \int_V \delta \mathbf{r}^T \mathbf{F}_e dV + \int_{\Omega} \delta \mathbf{r}^T \mathbf{f}_e d\Omega, \tag{22}$$

where  $V$  is the body volume,  $\rho$  is the body density,  $\delta \mathbf{r}$  is an infinitesimal virtual displacement,  $\ddot{\mathbf{r}}$  is the body acceleration (in world reference),  $\delta \boldsymbol{\varepsilon}$  is a vector of virtual infinitesimal internal strains,  $\boldsymbol{\sigma}$  is the internal stresses vector,  $\mathbf{F}_e$  is the vector of external volume forces,  $\Omega$  is the body surface and  $\mathbf{f}_e$  is the vector of external surface forces.

The quantities  $\delta \mathbf{r}$  and  $\ddot{\mathbf{r}}$  can be computed using Equation (3):

$$\begin{aligned} \delta \mathbf{r} &= \delta \mathbf{R} + \delta \mathbf{A} \bar{\mathbf{u}} + \mathbf{A} \delta \bar{\mathbf{u}} = \delta \mathbf{R} + \boldsymbol{\theta}_d \times \mathbf{A} \bar{\mathbf{u}} + \mathbf{A} \delta \mathbf{q}_f, \\ \ddot{\mathbf{r}} &= \ddot{\mathbf{R}} + \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{u} + \boldsymbol{\alpha} \times \mathbf{u} + 2\boldsymbol{\omega} \times \mathbf{A} \dot{\bar{\mathbf{u}}} + \mathbf{A} \ddot{\bar{\mathbf{u}}}, \end{aligned} \tag{23}$$

where  $\boldsymbol{\alpha}$  and  $\boldsymbol{\omega}$  are the body angular acceleration and velocity (in world reference), respectively, and  $\boldsymbol{\theta}_d = \boldsymbol{\omega} dt$  represents a virtual-infinitesimal rotation.

The expressions in (23) can be substituted in (20), leading to

$$\delta \mathbf{W}_i = \int_V \rho \delta \mathbf{r}^T \ddot{\mathbf{r}} dV = \delta \mathbf{R}^T \mathbf{Q}_i^R + \boldsymbol{\theta}_d^T \mathbf{Q}_i^\theta + \delta \mathbf{q}_f^T \mathbf{Q}_i^f, \tag{24}$$

and the terms  $\mathbf{Q}_i^R, \mathbf{Q}_i^\theta$  and  $\mathbf{Q}_i^f$  can be calculated using the following definitions:

$$\mathbf{m}_{RR} = \int_V \rho dV, \tag{25}$$

$$\mathbf{m}_{R\theta} = \int_V \rho \mathbf{A} (\bar{\mathbf{u}} \times)^T \mathbf{A}^T dV, \tag{26}$$

$$\mathbf{m}_{Rf} = \int_V \rho \mathbf{A} \mathbf{S} dV, \tag{27}$$

$$\mathbf{m}_{\theta\theta} = - \int_V \rho \mathbf{A} \bar{\mathbf{u}} \times \bar{\mathbf{u}} \times \mathbf{A}^T dV, \tag{28}$$

$$\mathbf{m}_{\theta f} = \int_V \rho \mathbf{A} \bar{\mathbf{u}} \times \mathbf{S} dV, \tag{29}$$

$$\mathbf{m}_{ff} = \int_V \rho \mathbf{S}^T \mathbf{S} dV, \tag{30}$$

$$\bar{\mathbf{S}} = \int_V \rho \mathbf{S} dV = \mathbf{A}^T \mathbf{m}_{Rf}, \tag{31}$$

$$\bar{\mathbf{S}}_t = \int_V \rho \bar{\mathbf{u}} dV, \tag{32}$$

$$\bar{\bar{\mathbf{S}}}_t = \int_V \rho (\bar{\mathbf{u}} \times) dV = \mathbf{A} \mathbf{m}_{R\theta}^T \mathbf{A}^T, \tag{33}$$

$$\bar{\mathbf{I}}_{\theta\theta} = \int_V \rho (\bar{\mathbf{u}} \times)^T (\bar{\mathbf{u}} \times) dV = \mathbf{A}^T \mathbf{m}_{\theta\theta} \mathbf{A}, \tag{34}$$

$$\bar{\mathbf{I}}_{\theta f} = \int_V \rho (\bar{\mathbf{u}} \times) \mathbf{S} dV = \mathbf{A}^T \mathbf{m}_{\theta f}. \tag{35}$$

The vector  $\mathbf{Q}_i^R$  can then be obtained as follows:

$$\mathbf{Q}_i^R = \mathbf{m}_{RR}\ddot{\mathbf{R}} + \mathbf{m}_{R\theta}\boldsymbol{\alpha} + \mathbf{m}_{Rf}\ddot{\mathbf{q}}_f - \mathbf{A}\mathbf{Q}_v^R, \tag{36}$$

being  $\mathbf{Q}_v^R = -\bar{\boldsymbol{\omega}} \times \bar{\boldsymbol{\omega}} \times \bar{\mathbf{S}}_t - 2\bar{\boldsymbol{\omega}} \times \bar{\mathbf{S}}\dot{\mathbf{q}}_f$  the quadratic velocity vector (due to Coriolis and centrifugal forces) associated to translational degrees of freedom.

The second term of the generalized inertial forces can be expressed as

$$\mathbf{Q}_i^\theta = m_{R\theta}^T\ddot{\mathbf{R}} + \mathbf{m}_{\theta\theta}\boldsymbol{\alpha} + \mathbf{m}_{\theta f}\ddot{\mathbf{q}}_f - \mathbf{A}\mathbf{Q}_v^\theta, \tag{37}$$

where the quadratic velocity vector associated to the rotational degrees of freedom is  $\mathbf{Q}_v^\theta = -\bar{\boldsymbol{\omega}} \times \bar{\mathbf{I}}_{\theta\theta}\bar{\boldsymbol{\omega}} - \dot{\bar{\mathbf{I}}}_{\theta\theta}\bar{\boldsymbol{\omega}} - \bar{\boldsymbol{\omega}} \times \bar{\mathbf{I}}_{\theta f}\dot{\mathbf{q}}_f$ .

The term  $\mathbf{Q}_i^f$ , which is related to the deformation d.o.f.  $\mathbf{q}_f$ , can be expanded as follows:

$$\mathbf{Q}_i^f = \mathbf{m}_{Rf}^T\ddot{\mathbf{R}} + \mathbf{m}_{\theta f}^T\boldsymbol{\alpha} + \mathbf{m}_{ff}\ddot{\mathbf{q}}_f - \mathbf{Q}_v^f, \tag{38}$$

being  $\mathbf{Q}_v^f = -\int_V \rho \mathbf{S}^T (\bar{\boldsymbol{\omega}}^2 \bar{\mathbf{u}} + 2\bar{\boldsymbol{\omega}} \bar{\mathbf{S}}\dot{\mathbf{q}}_f) dV$ .

The virtual work of the internal elastic forces, under the hypothesis of elastic constitutive law for the material, can be expressed as:

$$\delta \mathbf{W}_s = -\int_V \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV = -\delta \mathbf{q}_f^T \mathbf{K}_{ff} \mathbf{q}_f, \tag{39}$$

where  $\mathbf{K}_{ff}$  represents the structural stiffness matrix. The form of such matrix depends on the specific material constitutive law and on the body shape.

The virtual work of external forces reads as follows:

$$\delta \mathbf{W}_e = \delta \mathbf{R}^T \mathbf{Q}_e^R + \boldsymbol{\theta}_d^T \mathbf{Q}_e^\theta + \delta \mathbf{q}_f^T \mathbf{Q}_e^f, \tag{40}$$

where  $\mathbf{Q}_e^R$ ,  $\mathbf{Q}_e^\theta$  and  $\mathbf{Q}_e^f$  represent, respectively, the generalized components of the active forces associated to translational, rotational and deformation coordinates.

Equation (19) must be satisfied for every virtual displacement so that the following identities must hold:

$$\mathbf{Q}_i^R = \mathbf{Q}_e^R, \tag{41}$$

$$\mathbf{Q}_i^\theta = \mathbf{Q}_e^\theta, \tag{42}$$

$$\mathbf{Q}_i^f = -\mathbf{K}_{ff} \mathbf{q}_f + \mathbf{Q}_e^f. \tag{43}$$

Equations (41), (42) and (43) are the equations for 3D motion of a generic flexible body characterized by an elastic constitutive law for its material. In the scientific literature, such expressions are generally referred to as the *generalized Newton-Euler* equations (see e.g., [26]).

The equations of motion can be easily expressed in body axes, resulting in:

$$\begin{bmatrix} \mathbf{m}_{RR} & \tilde{\mathbf{S}}_t^T & \bar{\mathbf{S}} \\ & \bar{\mathbf{I}}_{\theta\theta} & \bar{\mathbf{I}}_{\theta f} \\ & & \mathbf{m}_{ff} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{R}} \\ \ddot{\boldsymbol{\alpha}} \\ \ddot{\mathbf{q}}_f \end{bmatrix} = \begin{bmatrix} \mathbf{O}_3 \\ \mathbf{O}_3 \\ -\mathbf{K}_{ff} \mathbf{q}_f \end{bmatrix} + \begin{bmatrix} \mathbf{Q}_v^R \\ \mathbf{Q}_v^\theta \\ \mathbf{Q}_v^f \end{bmatrix} + \begin{bmatrix} \bar{\mathbf{Q}}_e^R \\ \bar{\mathbf{Q}}_e^\theta \\ \bar{\mathbf{Q}}_e^f \end{bmatrix}. \tag{44}$$

## Appendix B: Structural stiffness matrix

$$\mathbf{K}_{ff,el} = \begin{bmatrix} \frac{EA}{\ell} & 0 & 0 & -\frac{EA}{\ell} & 0 & 0 \\ & \frac{12EJ}{3\ell} & \frac{6EJ}{3\ell} & 0 & -\frac{12EJ}{3\ell} & \frac{6EJ}{2\ell} \\ & & \frac{4EJ}{\ell} & 0 & -\frac{6EJ}{2\ell} & \frac{2EJ}{\ell} \\ & & & \frac{EA}{\ell} & 0 & 0 \\ & & & & \frac{12EJ}{3\ell} & -\frac{6EJ}{2\ell} \\ & & & & & \frac{4EJ}{\ell} \end{bmatrix}$$

## References

1. Absy, H.E., Shabana, A.A.: Geometric stiffness and stability of rigid body modes. *Journal of Sound and Vibration* **207**(4), 465–496 (1997)
2. Berzeri, M., Shabana, A.A.: Study of the centrifugal stiffening effect using the finite element absolute nodal coordinate formulation. *Multibody System Dynamics* **7**(4), 357–387 (2002)
3. Breedveld, P.C.: Physical systems theory in terms of bond graphs. PhD thesis, Twente University, Enschede, The Netherlands (1984)
4. Cellier, F.E., Elmqvist, H.: Automated formula manipulation supports object-oriented continuous-system modelling. *IEEE Control Systems Magazine* **13**(2), 28–38 (1993)
5. Dynasim AB. Dymola—dynamic modelling laboratory. Lund, Sweden. <http://www.dynasim.se/dymola.htm>.
6. Escalona, J.L., Hussien, H.A., Shabana, A.A.: Application of the absolute nodal co-ordinate formulation to multibody system dynamics. *Journal of Sound and Vibration* **214**(5), 833–851 (1998)
7. Ferretti, G., Schiavo, F., Viganù, L.: Object-oriented modelling and simulation of flexible multibody thin beams in modelica with the finite element method. 4th Modelica Conference, Hamburg-Harburg, Germany, March 7–8 (2005)
8. Hairer, E., Wanner, G.: Stiff differential equations solved by radau methods. *Journal of Computational and Applied Mathematics* **11**(1), 93–111 (1999)
9. Kübler, R., Schiehlen, W.: Modular simulation in multibody system dynamics. *Multibody System Dynamics* **4**(2–3), 107–127 (2000)
10. Maffezzoni, C., Girelli, R.: MOSES: Modular modeling of physical systems in an object-oriented database. *Mathematical Modeling of Systems* **4**(2), 121–147 (1998)
11. MathCore, A.B.: MathModelica. Linköping, Sweden. [urlhttp://www.mathcore.com/](http://www.mathcore.com/).
12. Mattsson, S.E., Andersson, M.: Omola – An Object-Oriented Modeling Language, Recent Advances in Computer Aided Control Systems. Elsevier Science pp. 291–310 (1993)
13. Mattsson, S.E., Elmqvist, H., Otter, M.: Physical system modeling with modelica. *Control Engineering Practice* **6**(4), 501–510 (1998)
14. Mattsson, S.E., Söderlind, G.: Index reduction in differential-algebraic equations using dummy derivatives. *SIAM Journal on Scientific Computing* **14**(3), 677–692 (1993)
15. Mayo, J.M., Garcia-Vallejo, D., Dominguez, J.: Study of the geometric stiffening effect: comparison of different formulations. *Multibody System Dynamics* **11**(4), 321–341 (2004)
16. Meirovitch, L.: Analytical methods in vibration. Macmillan Publishing, New York (1967)
17. Oh, M., Pantelides, C.C.: A modeling and simulation language for combined lumped and distributed parameter systems. *Computers and Chemical Engineering* **20**, 611–633 (1996)
18. Otter, M., Elmqvist, H., Mattsson, S.E.: The new modelica multibody library. In 3rd Modelica Conference, Linköping, Sweden, November 3–4 (2003)
19. Pantelides, C.C.: The consistent initialization of differential-algebraic systems. *SIAM Journal Scientific Statistical Computation* **9**(2), 213–231 (1988)

20. Pascal, M.: Some open problems in dynamic analysis of flexible multibody systems. *Multibody System Dynamics* **5**(4), 315–334 (2001)
21. PELAB. OpenModelica. Linköping, Sweden. <http://www.ida.liu.se/pelab/modelica/OpenModelica.html>
22. Petzold, L., Brenan, K.E., Campbell, S.L.: The numerical solution of initial value problems in differential-algebraic equations. Elsevier Science Publishing Co., New York (1989)
23. Piela, P.C., Epperly, T.G., Westerberg, K.M., Westerberg, A.W.: ASCEND: An Object-Oriented Computer Environment for modelling and analysis: the modelling language. *Computers and Chemical Engineering* **15**(1), 53–72 (1991)
24. Sass, L., Mcphee, J., Schmitke, C., Fiset1, P., Grenier, D.: A comparison of different methods for modelling electromechanical multibody systems. *Multibody System Dynamics* **12**(3), 209–250 (2004)
25. Shabana, A.A.: Flexible multibody dynamics: Review of past and recent developments. *Multibody System Dynamics* **1**(2), 189–222 (1997)
26. Shabana, A.A.: Dynamics of multibody systems. Cambridge University Press, Cambridge (1998)
27. Shi, P., Mcphee, J.: Dynamics of flexible multibody systems using virtual work and linear graph theory. *Multibody System Dynamics* **4**(4), 355–381 (2000)
28. The Modelica Association. Modelica – A unified object-oriented language for physical systems modeling. Language Specification Version 2.2, 2005. [www.modelica.org](http://www.modelica.org).
29. Theodore, R., Ghosal, A.: Comparison of the assumed modes and finite element models for flexible multilink manipulators. *International Journal of Robotics Research* **14**(2), 91–111 (1995)
30. Tiller, M.M.: Introduction to physical modeling with modelica. Kluwer Academic Publisher, Norwell (2001).
31. Timoshenko, S., Young, D., Weaver, W.: Vibration problems in engineering. John Wiley & Sons, New York (1974)
32. Wellstead, P.E.: Introduction to physical system modelling. Academic Press, London (1979)
33. Wu, S.C., Haug, E.J.: Geometric non-linear substructuring for dynamics of flexible mechanical systems. *International Journal for Numerical Methods in Engineering* **26**, 2211–2276 (1988)
34. Zienkiewicz, O.C., Taylor, R.L.: The finite element method: solids and fluid mechanics, dynamics and non-linearity. McGraw Hill, London (1991)