

Object-oriented modelling of general flexible multibody systems

Gianni Ferretti^{a*}, Alberto Leva^a and Bruno Scaglioni^b

^aPolitecnico Di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria DEIB Via Ponzio 34/5, 20133 Milano, Italy; ^bMUSP Lab, Via Tirotti 9, Le Mose, 29122 Piacenza, Italy

(Received 7 March 2013; final version received 18 May 2013)

1. Introduction

Accurate and efficient computer simulations of flexible multibody systems play an important role in the design and performance evaluation of many complex technological systems. However, multibody dynamics is frequently just one of the physical domains involved. For example, the design of mechatronic systems requires an integrated approach to mechanical, electronics and control design [1].

In order to master the complexity of a multidomain physical modelling, a modular approach is thus required, where the complexity is confined to the development of the equations of the single component, and system-level model organization is managed by means of the definition of standard component interfaces.

The overall purpose of the flourishing research on the matter is a *real* modular approach to modelling, thus based on the following characteristics:

- realization of models of large, complex and heterogeneous physical systems by connection of basic components;
- description of components' behaviours by stating physical principles rather than by writing calculation algorithms (*acausal* modelling);

An important result of the mentioned research is the so-called object-oriented modelling and simulation (OOMS) framework. Object-oriented modelling techniques and tools have

*Corresponding author. Email: gianni.ferretti@polimi.it

started emerging at the end of the 70's (e.g. ASCEND [2], OMOLA [3], gPROMS [4], MOSES [5]). As a main result of this research, some specific paradigms have been stated [6] and implemented in the Modelica language [7], and some modelling and simulation environments have been designed, both commercial (Dymola [8], MathModelica [9], SimulationX [10]) and open-source (OpenModelica [11]).

Coming to the specific case of multibody systems, and sticking to the Modelica language, although the presented ideas are more general, the adoption of a fully modular approach requires the description of the dynamics of every single body in terms of *local* variables, while the interaction between different bodies has to be described using the *connectors* of the standard Modelica multibody library [12]. A connector is associated with a reference frame, so that a connection is equivalent to a rigid junction of the two connector frames. The connector assumes the cut force and torque as *flow* variables [6], 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*,¹ describing the relative orientation between the world frame and the connector frame.

A local description of a body's dynamics of the body naturally calls for a floating frame of reference (FFR) approach [13], which is currently the most widely used method in computer simulation of flexible multibody systems. In the FFR formulation, each body is attached to a moving frame of reference undergoing large (rigid) motion, while the (small) elastic displacements are obtained in local coordinates with respect to the reference frame. In turn, according to the classical Rayleigh–Ritz method [14], the elastic displacements are in most cases defined as a linear combination of shape functions, whose coefficients are the so-called elastic or *modal* coordinates. In the case of simple geometries, such as beams [6], the shape functions can be determined analytically, but in more general cases the use of finite elements (FE) computer codes as preprocessors is necessary. In this last case the huge number of *nodal* coordinates must be reduced to a much smaller number of modal coordinates, through the classical Craig–Bampton method [15] or other recently proposed methods [16–19].

One of the characteristics of the FFR formulation is that the inertia terms are highly nonlinear, so that updating them at every time step can become the most central processing unit (CPU) intensive task in simulation. Recently, in [20], two different methods for performing this task have been compared from the point of view of the computational efficiency. The first method, referred to as the projection method, is based on the corotational approach by Géradin and Cardona [21], and consists of projecting the constant FE mass matrix into the generalized coordinates, by means of a variable projection matrix. In the second one [13] the inertia terms are calculated from some *inertia shape integrals*, in turn obtained in a preprocessing stage from the FE model. The projection method is very simple to implement, but has the main drawback that the CPU time depends anyway on the size of the original FE model, regardless of the number of mode shapes chosen. On the contrary, with the preprocessing method, the CPU time depends only on the size of the reduced model, so that the mesh can be refined as much as needed without introducing any penalty to the simulation time. The results of the comparison of [20] confirmed that computational efficiency is largely increased with the preprocessing method, which is therefore adopted in our approach.

In the case of large deformation problems the absolute nodal coordinate formulation (ANCF) [13] is also often adopted. The ANCF, adopting absolute displacements and global slopes as nodal coordinates, results in a constant mass matrix, while the most demanding task becomes the computation of the elastic forces, which are highly nonlinear in terms of the nodal coordinates.

On the other hand, in order to model large deformations, it is possible to subdivide a body into several flexible components or *substructures*, adopting for each substructure a FFR formulation [22]. Using this approach (or *substructuring* method), it is usually necessary to formulate the substructure compatibility constraints, to be imposed at the substructures interfaces, explicitly. The problems arising from the necessity of this explicit formulation can be avoided adopting a fully modular object-oriented approach, since the compatibility constraints are automatically generated at the connection of the substructures' modules, if they are defined through connectors. In this work, first the last (and crucial) statement is demonstrated by realizing it in the Modelica language, delegating to the compiler (in this work Dymola) the task of assembling the whole model, recovering computational efficiency through the aid of symbolic manipulation algorithms. For the sake of completeness it must be mentioned that, as recently pointed out by Heckmann [23], the substructuring method is a technique to find a specific solution for a specific problem but it is not systemically supported as the ANCF formulation, and requires skilled modellers (in particular, using more substructures does not necessarily improve the accuracy of the solution).

Another important question is the choice of the boundary conditions for mode shapes in flexible multibody systems, thoroughly analysed in [23]. However, in the case of FE preprocessing, this choice is inherited from the FE analysis and embedded in the body data, while in the case of the analytical beam model the choice between tangent, pinned and free boundary conditions is still left to the user by setting a model parameter [6,24].

In this paper an object-oriented modelling of general flexible multibody systems is described, allowing the inclusion of the flexible multibody dynamics in object-oriented multidomain physical modelling.

At first, the motion equations of a general flexible body are fully developed (up to the Modelica code, reported in Appendix C), also with the aim of identifying the necessary parameters: undeformed positions and orientation of connectors, modal shapes, structural damping and stiffness matrices, and mass distribution, in the form of inertia invariants.

In the case of bodies of general shape, however, the analytical computation of most of the said parameters (an example in the case of a beam is briefly reported in Appendix B) is unfeasible and the use of finite element method (FEM) preprocessing tools is required. In fact, most commercial FEM packages already store in files, either in binary or in ASCII format, the flexible body parameters, as the result of a modal reduction step. In this paper, however, in order to compute the gyroscopic and centripetal terms relevant to the flexible degrees of freedom in a compact form, two additional inertia invariants are defined, which can be anyway computed from the stored modal reduction results (inertia invariants).

It must be also pointed out that the efficient choice of the generalized coordinates, implemented in the Modelica standard (rigid) multibody library, has been maintained. Thus, when a body is a component of a tree structure, the motion of the FFR is actually calculated by propagation of the kinematic quantities from the root of the tree while, in the case of floating bodies, the body itself is a root, introducing its own generalized coordinates for position and orientation.

Some simulation results, relevant to well-known benchmark cases, are finally reported, comparing analytical models of flexible beams to models obtained through a FEM preprocessing with Abaqus.

The paper is organized as follows. In Section 2, the state of the art relevant to the import of FEM models in multibody packages is reviewed. In Section 3, the motion equations of a general flexible body are outlined, defining the set of body data needed to characterize the flexible body for multibody simulation. In Section 4, the structure of the Modelica model of

a flexible body is sketched. In Section 5, the simulation results obtained with reference to benchmark case studies, both with analytical and FEM beam models, are discussed. Finally, some conclusion and perspective for further studies are briefly outlined in Section 6.

2. State of the art

Two commercial packages exist that preprocess the output of FE codes to get the Modelica model of a flexible body, one has been developed by the German Aerospace Center (DLR) [24] and distributed by Bausch-Gall GmbH [25], the other is distributed by Claytex Services Ltd [26].

The DLR FlexibleBodies library simply provides two Modelica classes (models): one for the beam model (Beam) and one for general flexible bodies exported from FE codes (ModalBody). The results of the FE analysis performed by several general purpose codes (Abaqus [27], Ansys [28], MSC.Nastran [29], NX Nastran [30], PERMAS [31]) are first processed by another commercial code: FEMBS, distributed by SIMPACK AG [32], performing modal reduction in a two-step process. Guyan or Craig–Bampton reduction methods are applied in the first step to keep the flexible body input file to FEMBS small, while in the second reduction step the modes of interest (i.e. in the frequency range of interest) for multibody simulation are selected. The reduced modal representation is then stored in a standard input data (SID) file [33,34], an object-oriented data structure developed to define a standard format to exchange data between FE and MBS codes. When a ModalBody class is instantiated the user has just to specify the name of the SID file containing the modal description of the body.

The structure of the SID file is quite complex, and contains data relevant to the mass distribution of the body, the mode shapes, the location of attachment points, the modal mass, stiffness and damping matrices. The complexity mainly stems from the fact that the displacement field is approximated by a second order Taylor expansion, with space-dependent mode shapes and time-dependent modal amplitudes, whose focus is not to represent large deformations (e.g. for crash analysis) but the inclusion of stress stiffening and softening effects. Nonlinear strain-displacement relationships can be also characterized, which may become important in some cases, as shown in [35]. Finally, the SID file also stores the original mesh of the FE model, used by Dymola to perform the animation of the simulated motion of the flexible body.

The Claytex library generates directly the Modelica model of a flexible body from the output of the modal reduction process performed by three FE codes: namely Nastran, Genesis and Abaqus.

With reference to the fundamental step of FE model reduction, in addition to classical techniques, such as the modal truncation [36] or the component mode synthesis [15], several other techniques have been recently investigated, taking into account specific criteria, such as computability for large scale systems, stability preservation, quality of the reduced order model, model error evaluation, model tuning in a given frequency range, and more. For example, reduction methods based on Krylov subspaces, singular value decomposition (SVD), Gramian matrices, proper orthogonal decomposition (POD) have been considered and implemented in the software package Morembs [19], which gives the user a tool for selecting several model order reduction techniques. The package imports FE data from Ansys, Abaqus and Permas and generates a SID file for Simpack containing the reduced elastic body; the reduced system matrices can be also directly used in the multibody simulation package Neweul-M² [37].

3. Motion equations of a general flexible body

To start the treatise, it is convenient to recall the generalized Newton–Euler equations for a generic unconstrained flexible body, formulated with respect to a local reference frame rigidly attached to the body itself [6,13,18,20,34,38] and, based on that, to introduce the inertia invariants, assuming a continuous mass distribution.

The position (in local coordinates) of a point on a flexible body, see Figure 1, is given by

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

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 bar above a vector indicates that the vector is expressed with reference to the local FFR.

If small elastic deflections are considered, the infinite dimensional deformation field on the body can be approximated by a functional basis space with finite dimension, say M , so that the vector $\bar{\mathbf{u}}_f$ can be expressed by the finite dimensional product

$$\bar{\mathbf{u}}_f = \mathbf{S}\mathbf{q}, \quad (1)$$

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} is the M -dimensional vector of deformation degrees of freedom.

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

$$\mathbf{p} = \mathbf{r} + \mathbf{A}\bar{\mathbf{u}} = \mathbf{r} + \mathbf{A}\bar{\mathbf{u}}_0 + \mathbf{A}\mathbf{S}\mathbf{q},$$

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.

The representation of a generic flexible body in the world reference frame requires then $6 + M$ d.o.f.: 3 corresponding to the rigid displacements \mathbf{r} , 3 to the undeformed body orientation angles $\boldsymbol{\theta}$ and M to the deformation d.o.f. \mathbf{q} .

In order to fulfill the derivation of motion equations, the relation between the derivative of the orientation angles $\dot{\boldsymbol{\theta}}$ and the relevant angular velocity $\boldsymbol{\omega}$ must be specified. Leaving

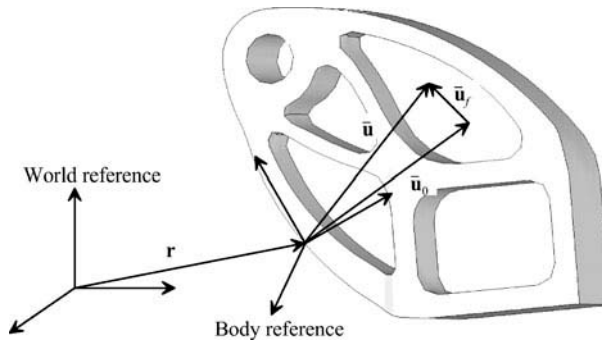


Figure 1. Floating reference system.

aside considerations about the singularity of the chosen set of orientation angles (*gimbal lock*), which can be avoided by resorting to a redundant set of parameters, such as quaternions, the angular velocity of a floating frame can be computed from the rotation matrix \mathbf{A} as a function of the chosen angles $\boldsymbol{\theta}$:

$$\tilde{\boldsymbol{\omega}} = \frac{d\mathbf{A}(\boldsymbol{\theta})}{dt} \mathbf{A}^T(\boldsymbol{\theta}),$$

where $\tilde{\mathbf{v}}$ is the skew-symmetric matrix associated to vector \mathbf{v} .

The motion equations for a generic flexible body, expressed in the local reference frame, can be developed applying the principle of virtual work [6,38,39], yielding

$$\begin{bmatrix} m\mathbf{I}_3 & m\tilde{\mathbf{d}}_C^T & \tilde{\mathbf{C}}_t^T \\ m\tilde{\mathbf{d}}_C & \tilde{\mathbf{J}} & \tilde{\mathbf{C}}_r^T \\ \tilde{\mathbf{C}}_t & \tilde{\mathbf{C}}_r & \mathbf{M}_e \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{r}} - \tilde{\mathbf{g}} \\ \dot{\tilde{\boldsymbol{\omega}}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{0}_3 \\ -\mathbf{K}_e \mathbf{q} - \mathbf{D}_e \dot{\mathbf{q}} \end{bmatrix} + \begin{bmatrix} \mathbf{h}_\omega^r \\ \mathbf{h}_\omega^\theta \\ \mathbf{h}_\omega^f \end{bmatrix} + \begin{bmatrix} \mathbf{h}_e^r \\ \mathbf{h}_e^\theta \\ \mathbf{h}_e^f \end{bmatrix} \quad (2)$$

where $\ddot{\mathbf{r}}$, $\dot{\tilde{\boldsymbol{\omega}}}$, $\ddot{\mathbf{q}}$ are the linear, angular and deformation accelerations, respectively, $\tilde{\mathbf{g}}$ is the gravity acceleration, \mathbf{I}_3 is the 3×3 identity matrix, \mathbf{M}_e , \mathbf{D}_e , \mathbf{K}_e are the structural mass, damping and stiffness matrix, respectively, \mathbf{h}_ω^r , \mathbf{h}_ω^θ , \mathbf{h}_ω^f are the vectors of gyroscopic and centripetal terms and \mathbf{h}_e^r , \mathbf{h}_e^θ , \mathbf{h}_e^f are the vectors of external forces, applied at the body connectors (recall the discussion in Section 1).

The terms in the generalized mass matrix depend on some inertia invariants, which in turn depend on the shape functions and on the body mass distribution, whose computation is reported in Appendix A:

$$m = I^1, \quad (3)$$

$$m\tilde{\mathbf{d}}_C = \tilde{\mathbf{I}}^2 + \sum_{i=1}^M \tilde{\mathbf{I}}_i^3 q_i, \quad (4)$$

$$\tilde{\mathbf{C}}_t^T = [\mathbf{I}_1^3 \ \mathbf{I}_2^3 \ \dots \ \mathbf{I}_M^3], \quad (5)$$

$$\tilde{\mathbf{C}}_r^T = \mathbf{I}^4 + \sum_{i=1}^M \mathbf{I}_i^5 q_i, \quad (6)$$

$$\mathbf{M}_e = \mathbf{I}^6, \quad (7)$$

$$\tilde{\mathbf{J}} = \mathbf{I}^7 - \sum_{i=1}^M (\mathbf{I}_i^{8T} + \mathbf{I}_i^8) q_i - \sum_{i=1}^M \sum_{j=1}^M \mathbf{I}_{ij}^9 q_i q_j, \quad (8)$$

The vectors of gyroscopic and centripetal terms relevant to the rigid degrees of freedom are given by

$$\mathbf{h}_\omega^r = -\tilde{\boldsymbol{\omega}} \times \tilde{\boldsymbol{\omega}} \times m\tilde{\mathbf{d}}_C - 2\tilde{\boldsymbol{\omega}} \times \tilde{\mathbf{C}}_t^T \dot{\mathbf{q}}, \quad (9)$$

$$\mathbf{h}_\omega^\theta = -\tilde{\boldsymbol{\omega}} \times \tilde{\mathbf{J}}\tilde{\boldsymbol{\omega}} - \tilde{\mathbf{J}}\dot{\tilde{\boldsymbol{\omega}}} - \tilde{\boldsymbol{\omega}} \times \tilde{\mathbf{C}}_r^T \dot{\mathbf{q}}, \quad (10)$$

where

$$\dot{\mathbf{J}} = - \sum_{i=1}^M (\mathbf{I}_i^{8T} + \mathbf{I}_i^8) \dot{q}_i - \sum_{i=1}^M \sum_{j=1}^M \mathbf{I}_{ij}^9 (\dot{q}_i q_j + q_i \dot{q}_j), \quad (11)$$

while the computation of the vector relevant to the elastic degrees of freedom is much more involved (see Appendix A) and requires the definition of two more inertia invariants, namely

$$\mathbf{h}_\omega^f = \begin{bmatrix} \bar{\omega}_1 \left(\sum_{i=1}^3 \mathbf{d}_{ii} \right) - \bar{\omega}^T \mathbf{D}_1 \\ \bar{\omega}_2 \left(\sum_{i=1}^3 \mathbf{d}_{ii} \right) - \bar{\omega}^T \mathbf{D}_2 \\ \bar{\omega}_3 \left(\sum_{i=1}^3 \mathbf{d}_{ii} \right) - \bar{\omega}^T \mathbf{D}_3 \end{bmatrix}^T \bar{\omega} - 2 \left[\bar{\omega}_1 (\mathbf{I}_{32}^{11} - \mathbf{I}_{23}^{11}) + \bar{\omega}_2 (\mathbf{I}_{13}^{11} - \mathbf{I}_{31}^{11}) + \bar{\omega}_3 (\mathbf{I}_{21}^{11} - \mathbf{I}_{12}^{11}) \right] \dot{\mathbf{q}}, \quad (12)$$

where

$$\mathbf{D}_i = \begin{bmatrix} \mathbf{d}_{i1} \\ \mathbf{d}_{i2} \\ \mathbf{d}_{i3} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{i1}^{10} \\ \mathbf{I}_{i2}^{10} \\ \mathbf{I}_{i3}^{10} \end{bmatrix} + \begin{bmatrix} \mathbf{q}^T \mathbf{I}_{i1}^{11} \\ \mathbf{q}^T \mathbf{I}_{i2}^{11} \\ \mathbf{q}^T \mathbf{I}_{i3}^{11} \end{bmatrix} \quad i = 1, 2, 3. \quad (13)$$

These last two invariants, \mathbf{I}_{ij}^{10} and \mathbf{I}_{ij}^{11} however can be computed from invariants \mathbf{I}_i^8 and \mathbf{I}_{ij}^9 (see Appendix A).

Matrix \mathbf{D}_e , modelling the dissipative properties of the material, is defined as $\mathbf{D}_e = \alpha \mathbf{M}_e + \beta \mathbf{K}_e$, where α and β are the so-called Rayleigh damping coefficients.

Assuming N_c connectors attached to a flexible body the vectors of external forces are given by

$$\mathbf{h}_e^r = \sum_{i=1}^{N_c} \bar{\mathbf{f}}_i, \quad (14)$$

$$\mathbf{h}_e^\theta = \sum_{i=1}^{N_c} \left(\bar{\tau}_i + \bar{\mathbf{l}}_i \bar{\mathbf{f}}_i \right), \quad (15)$$

$$\mathbf{h}_e^f = \sum_{i=1}^{N_c} \left(\mathbf{S}_i^T \bar{\mathbf{f}}_i + \hat{\mathbf{S}}_i^T \bar{\tau}_i \right), \quad (16)$$

where $\bar{\mathbf{f}}_i$ and $\bar{\tau}_i$ are the forces and torques acting at the connector frames projected on the local reference frame, $\bar{\mathbf{l}}_i$ is the position vector from the origin of the local body reference frame to the origin of the connector frame, \mathbf{S}_i and $\hat{\mathbf{S}}_i$ are the slices of the modal matrix corresponding to the translational and rotational d.o.f. of connectors [40]. Note that

$$\bar{\mathbf{l}}_i = \bar{\mathbf{u}}_{0i} + \mathbf{S}_i \mathbf{q}, \quad (17)$$

with $\bar{\mathbf{u}}_{0i}$ being the undeformed position of the i th connector.

In order to project the forces and torques acting at the connector frames on the local reference frame, *small* rotations of the connectors due to body deformation must be taken into account. These small rotations can be still obtained by a modal superposition as

$$\boldsymbol{\theta}_i = \begin{bmatrix} \theta_{i1} \\ \theta_{i2} \\ \theta_{i3} \end{bmatrix} = \hat{\mathbf{S}}_i \mathbf{q}, \quad (18)$$

so that the rotation matrix between the deformed and the undeformed connector orientation can be defined as

$$\hat{\mathbf{A}}_i = \mathbf{I}_3 + \tilde{\boldsymbol{\theta}}_i = \begin{bmatrix} 1 & -\theta_{i3} & \theta_{i2} \\ \theta_{i3} & 1 & -\theta_{i1} \\ -\theta_{i2} & \theta_{i1} & 1 \end{bmatrix}. \quad (19)$$

Therefore, if $\bar{\mathbf{A}}_i$ is the (constant) rotation matrix between the connector frame and the local frame in the undeformed configuration, the projection of the force \mathbf{f}_i and the torque $\boldsymbol{\tau}_i$ acting at the connector frame i is given by

$$\bar{\mathbf{f}}_i = \bar{\mathbf{A}}_i \hat{\mathbf{A}}_i \mathbf{f}_i, \quad (20)$$

$$\bar{\boldsymbol{\tau}}_i = \bar{\mathbf{A}}_i \hat{\mathbf{A}}_i \boldsymbol{\tau}_i. \quad (21)$$

To cope with the connector structure it is also necessary to compute the kinematic quantities \mathbf{A}_i , $\boldsymbol{\omega}_i$ and \mathbf{p}_i relevant to each connector, thus the rotation matrix between the frame connector and the world reference frame, its angular velocity and the position of the origin of the connector frame with respect to the world frame. This leads to

$$\mathbf{A}_i = \mathbf{A} \bar{\mathbf{A}}_i \hat{\mathbf{A}}_i, \quad (22)$$

$$\boldsymbol{\omega}_i = \boldsymbol{\omega} + \mathbf{A} \bar{\mathbf{A}}_i \hat{\mathbf{S}}_i \dot{\mathbf{q}}, \quad (23)$$

$$\mathbf{p}_i = \mathbf{r} + \mathbf{A} \bar{\mathbf{l}}_i. \quad (24)$$

As a result, the set of data describing a flexible body can be summarized as in [Table 1](#), while an example of analytical computation of said data in the case of a beam is reported in [Appendix B](#).

Table 1. Flexible body data.

M	Number of deformation d.o.f.
$I^1, I^2, I_i^3, I^4, I_i^5, I^6, I^7, I_i^8, I_{ij}^9, I_{ij}^{10}, I_{ij}^{11}$	Inertia invariants
$\mathbf{D}_e, \mathbf{K}_e$	Structural damping and stiffness matrix
N_c	Number of connectors
$\mathbf{S}_i, \hat{\mathbf{S}}_i$	Slices of the modal matrix of connectors d.o.f.
$\bar{\mathbf{u}}_{0i}, \bar{\mathbf{A}}_i$	Undeformed position and orientation of connectors

4. Structure of the Modelica model

The Modelica model of a general flexible body is characterized by an array of N_c multibody connectors [12] (see model FEMBody in Figure 2), while the data in Table 1 have been suitably collected in the Modelica record BodyData. The record is defined as replaceable:

```
replaceable parameter FEMData.BodyData data;
```

so that it is possible, by exploiting the features of the Modelica language, to assign a different data record to each FEMBody instance, by simply replacing the record in the model declaration, as in

```
FEMBody FlexPendulum(redeclare FEMData.PendulumData data,
    alpha=1,
    beta=0.5,
    d=1);
```

where α , β , d are the parameters defining the damping matrix \mathbf{D}_e .

An important question arises when considering the state variables introduced by the flexible body model.

In the standard Modelica (rigid) multibody library [12], the generalized coordinates of joints are used as states if possible, otherwise, e.g. because bodies are freely floating in space, states are selected from the body coordinates. In this last case, the body frame is selected as a *root* of the connection tree and, in particular, the frame orientation and angular velocity are computed from state variables defined within the body model (by default, quaternions are used). If the body frame is not selected as root, its orientation and angular velocity are indirectly computed from the generalized positions and velocities of the joints connecting the body to the root of the tree. Once the orientation and the angular velocity of the body frame are determined, all other kinematical quantities are derived by differentiation and used in the Newton–Euler equations, formulated with respect to the body frame.

In the case of a floating flexible body it is natural to define the FFR as root, without necessarily having a connector attached, since every quantity is referred to it. Otherwise, for the sake of efficiency, it is necessary to find out a connector of the body belonging to the tree path, say connector r , through the utility function `rooted()` (`rooted(r.R)=true`),

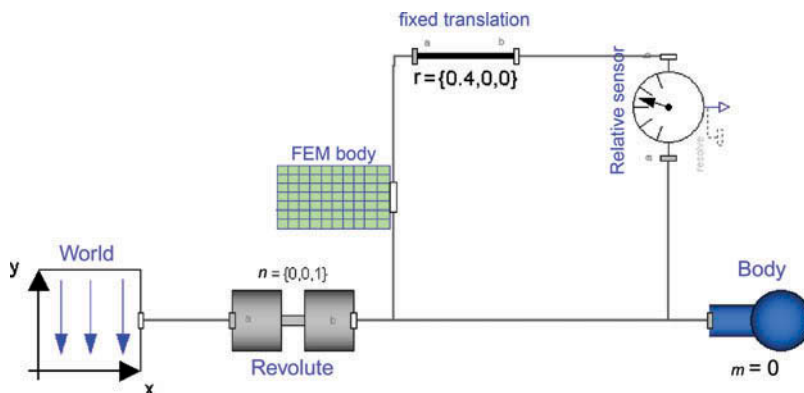


Figure 2. Modelica model of the flexible pendulum.

and to explicitly compute the orientation and the angular velocity of the FFR from the same quantities of the r connector frame:

$$\mathbf{A} = \left(\bar{\mathbf{A}}_r \hat{\mathbf{A}}_r \right)^{-1} \mathbf{A}_r, \quad (25)$$

$$\boldsymbol{\omega} = \boldsymbol{\omega}_r - \mathbf{A} \bar{\mathbf{A}}_r \hat{\mathbf{S}}_r \dot{\mathbf{q}} \quad (26)$$

with the orientation and the angular velocity of all other connector frames being still computed by Equations (22) and (23). It is worth mentioning that the said kinematic equations have been implemented in Modelica code by calling the methods of the Orientation class defining absolute, relative and inverse rotations between frames.

Again, having defined the orientation and the angular velocity of the FFR, the flexible body motion equations can be written in Modelica code as reported in Appendix C, where the numbers in comments refer to the above equations, in order to help the reader follow the equations to model conceptual flow.

5. Simulation results

Being focused on a general object-oriented formulation of flexible multibody systems, this paper adopts the classical modal approach, based on the model reduction performed by several packages in order to interface the results of the FEM modelling with ADAMS [40].

As a result of the model reduction step, most of the data reported in Table 1 are stored in a binary Modal Neutral File (.mnf), in turn converted into an ASCII file (.mtx). The inertia invariants \mathbf{I}_{ij}^{10} and \mathbf{I}_{ij}^{11} however, are not stored in the .mnf and .mtx files, so they are computed by two Modelica functions according to Equations (A14) and (A15), when the BodyData record is instantiated, i.e.

```
parameter Real inv10[3,3,M] = FEMPackage.calcInv10(M,inv8);
parameter Real inv11[3,3,M,M] = FEMPackage.calcInv11(M,inv9);
```

In this work, the FEM models have been created with Abaqus [27], which is endowed with a specific tool for the generation of the .mnf file. The procedure (described in the software documentation) consists of two steps, in the first a modal eigenvalues analysis must be performed.

As far as the choice of the boundary conditions is concerned, it must be pointed out that it largely depends on the problem at hand. In this respect, a comprehensive survey is reported in [23] where, in the case of chain and tree-structured multibody systems, it is suggested to align the boundary conditions with the local d.o.f. of the connecting joints.

Then the Abaqus Adams interface tool must be executed, in order to extract the inertia invariants and the modal shape matrices from the Abaqus results file.

To generate the ASCII file containing the Modelica BodyData record, a parsing software tool has been built in the context of this work by using the Flex/Bison parser generation suite [41,42]. The only data needed to define the Modelica model, and not stored in the .mtx file, are the number and the spatial layout of the connector frames on the body. Accordingly, the parsing tool ask the user to specify an input list representing the FEM nodes in which Modelica connectors must be placed.

Several different numerical examples have been considered, so as to thoroughly test and validate the body model. In the following subsections the most significant test cases

are briefly reported, comparing the results obtained with an analytical beam model and a FEM beam model.

It must be pointed out that the object-oriented nature of the body model and the full compliance with the Modelica multibody library allows for an easy set-up of different simulation scenarios, as well as for the separation between the model dynamic equations and the specific body data.

All simulations have been performed on a workstation equipped with an Intel Xeon 4 Ghz and 2 Gb RAM processor, and the CPU simulation times have been reported. However, it must be pointed out that the CPU time varies largely, depending on the damping ratio, the numerical solver and the numerical tolerance.

5.1. Flexible Pendulum

The first benchmark example is taken from [43], where a flexible rod swings under the action of gravity.

The Modelica model of the system is sketched in Figure 2: the flexible body is connected to the world reference frame through a revolute joint and to a massless body on the opposite tip. In order to observe the tip deflection a RelativeSensor component has been used.

While in the case of the analytical beam model the FFR is always placed in one of the beam ends (referring to Figure 2 it is attached to the joint connector), in the case of a model derived by numerical modal reduction the FFR corresponds to the global coordinate system adopted in the FE solid modelling step.

The rod has a length $L = 0.4$ m, a cross sectional area $A = 0.0018$ m² and a density $\rho = 5540$ kg/m³. Differently from [43], in FEM-based models it is not possible to assign the second moment of area arbitrarily, as it is indirectly calculated in the FEM preprocessing stage. In order to maintain the same body data used in [43] the quantity $E \times J$ (second moment of area \times Young's modulus) was kept coherent with [43], assuming a round section for the rod ($J = \pi r^4/4$ m⁴). The FEM beam model has been therefore created with Abaqus, characterized by 20 nodes and 20 B31 one-dimensional elements. Then, after the frequency and substructuring analysis steps, a number of 15 eigenmodes was retained.

The differential algebraic systems solver integration algorithm was adopted, with a tolerance of 1×10^{-6} , and the simulation required 0.062 s of CPU time. The results were compared with those obtained in [44], where the same beam was modelled analytically, in this case the simulation required 0.032 s of CPU time. Figure 3 shows the translational

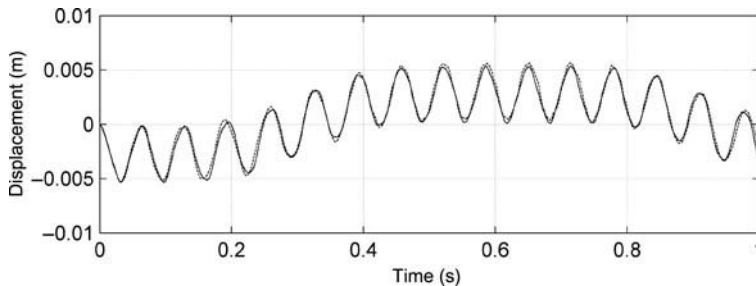


Figure 3. Flexible pendulum tip displacement. Solid line: FEM model, dashed line: analytical model.

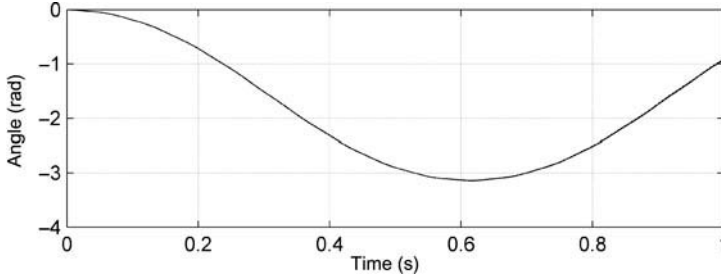


Figure 4. Flexible pendulum swing angle. Solid line: FEM model, dashed line: analytical model.

displacement of the pendulum tip in the motion direction, namely the distance between the tip of the flexible beam and the tip of a rigid beam, while Figure 4 shows the swing angle, namely the relative orientation between the connector frames at both ends of the flexible beam in the plane of motion. As it is apparent, the results obtained with the FEM model (solid line) are in good accordance with the analytic one (dashed line) and with the experiment reported in [43].

5.2. Elastic slider crank with FEM beam models

The elastic slider-crank mechanism is one of the most widely used benchmark for flexible multibody systems simulation, and involves a closed chain [45–47]. In this respect, here the same simulation experiment considered in [43] and [44] has been performed. The Modelica model, composed by a flexible crankshaft, a flexible rod and a rigid massless slider block, connected with frictionless revolute and prismatic joints, is shown in Figures 5 and 6.

A round section beam was considered for the rod, with a length $L = 0.304$ m, a cross sectional radius $r = 0.005$ m and a density $\rho = 2767$ kg/m³. The second moment of area was $J = 4.909 \times 10^{-10}$ m⁴, calculated as in Section 5.1, with the aim of keeping the quantity $E \times J$ consistent with [43] and [44]; the elasticity modulus E was set to 1×10^9 N/m². The FEM rod model was computed as a beam containing 30 B31 nodes, and 15 eigenmodes were retained.

The crankshaft was characterized by the same physical parameters of the rod but the length, which was $L = 0.152$ m, and the elasticity modulus: $E = 5 \times 10^7$ N/m²; the FEM model was made by 15 B31 elements, and 10 eigenmodes were retained.

In accordance with [43] and [44], a simulation time of 1.6 s was assumed, while driving the crankshaft with the following torque:

$$\tau = \begin{cases} M(t) = 0.01(1 - e^{-t/0.167}), & t \leq 0.7 \text{ s} \\ 0, & t > 0.7 \text{ s}. \end{cases} \quad (27)$$

The simulation was performed with a Radau algorithm and a tolerance of 10^{-6} and the simulation CPU time was 28.42 s, while the simulation of the analytic beam model required a CPU time of 21.14 s. Figures 7 and 8 compare again the results obtained with both analytical (dashed line) and FEM (solid line) beam model, which are also in good accordance with [43].

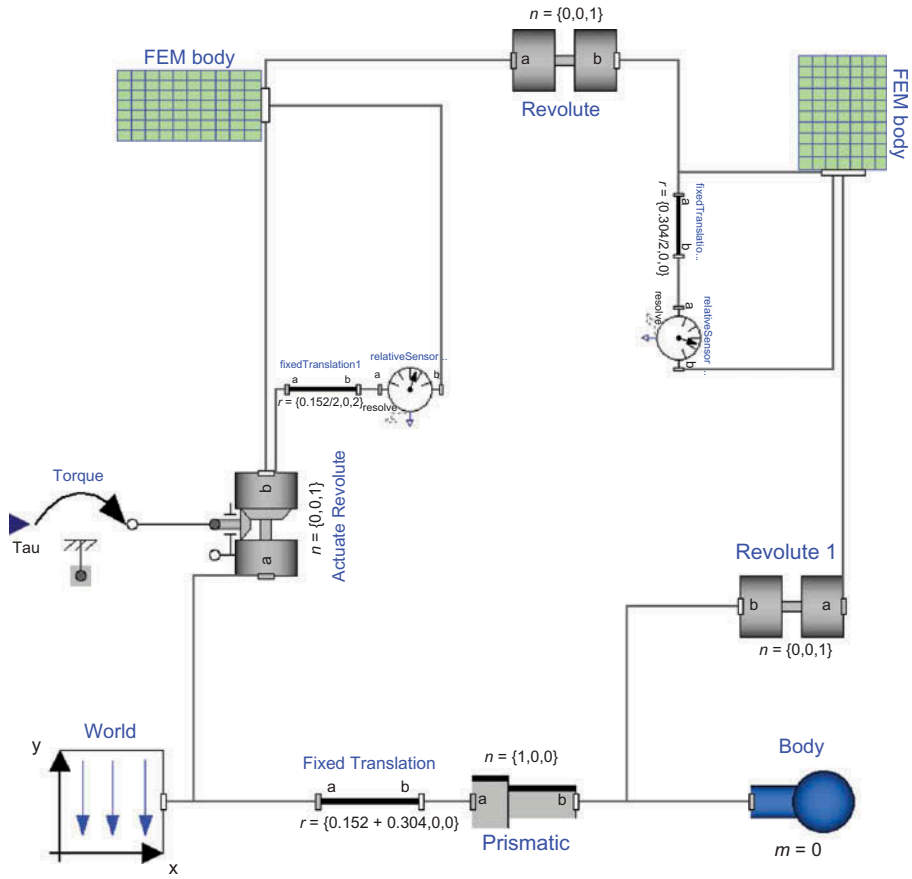


Figure 5. The slider-crank simulation scheme.

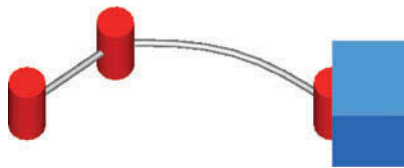


Figure 6. The elastic slider-crank model.

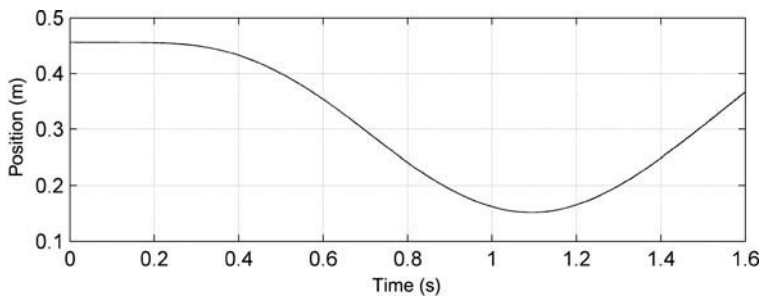


Figure 7. Position of the slider block. Solid line: FEM model, dashed line: analytical model.

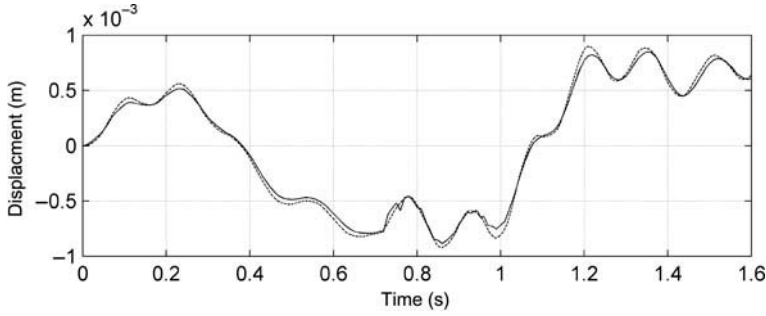


Figure 8. Displacement of the rod's middle point. Solid line: FEM model, dashed line: analytical model.

5.3. Centrifugal stiffening of a FEM Beam model

When flexible beams are rotating at high angular velocities, the effect of inertial, centrifugal and Coriolis forces becomes significant, producing a geometric stiffening effect that is referred to in the literature as *centrifugal stiffening* [48–51].

The FFR formulation presented in the paper is not able to represent the centrifugal stiffening effect by itself, due, in particular, to the existing coupling between bending and axial displacements, which cannot be represented by the typical linear strain–displacement relationship adopted in the definition of the elastic potential energy in the FFR formulation.

Among the various methods proposed in the literature to account for the centrifugal stiffening [22,49,50,52,53] the easiest one is the application of the *substructuring* technique [22], dividing a flexible body into smaller rigidly connected substructures, each one described by a local FFR. In this way, although each substructure is characterized by small deformations, the whole assembled structure can represent very large deformations.

The classical spin-up manoeuvre problem, another frequently used benchmark in flexible multibody simulation in order to point out centrifugal stiffening [48,54,55], is here considered. A flexible beam rotates about an axis passing through one of its ends, according to the following law [48]:

$$\theta(t) = \begin{cases} \frac{\Omega}{T} \left[\frac{t^2}{2} + \left(\frac{T}{2\pi} \right)^2 (\cos(\frac{2\pi t}{T}) - 1) \right], & t < T \\ \Omega(t - T/2), & t \geq T, \end{cases} \quad (28)$$

thus, the spin-up manoeuvre starts at $t = 0$ and ends at $t = T$, reaching a constant angular velocity Ω .

It is worth noticing that the analytical model considers a planar deformation field, while the FEM model considers a 3D deformation field. As such, in order to compare the results obtained with both models, in a first simulation experiment no gravity was considered. The following data were assumed for the beam: length $L = 8$ m, cross sectional area $A = 7.299 \times 10^{-5}$ m² and density $\rho = 2767$ kg/m³. As in the other experiments the product $E \times J$ was kept coherent, so the following value was assumed for modulus of elasticity $E = 1.3359 \times 10^{12}$ N/m².

Three different simulation experiments were performed, corresponding to the target angular velocities of $\Omega = 2$ rad/s, $\Omega = 4$ rad/s and $\Omega = 10$ rad/s and the deflection of the tip, namely the difference between the tips of a rigid and a flexible beam in the plane of motion, was recorded [48]. In the case of $\Omega = 2$ rad/s and $\Omega = 4$ rad/s the FEM beam

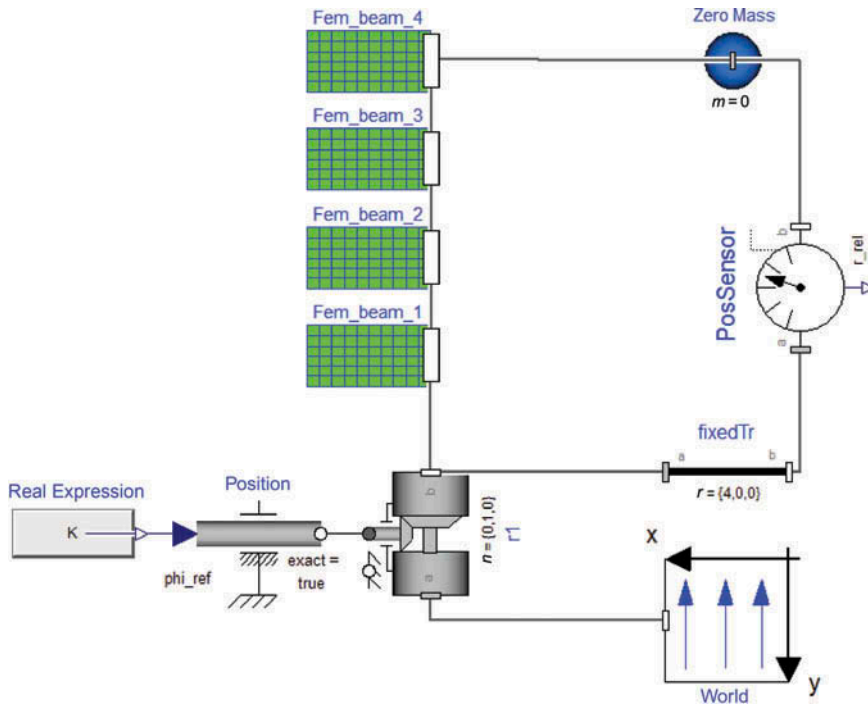


Figure 9. Beam substructuring.

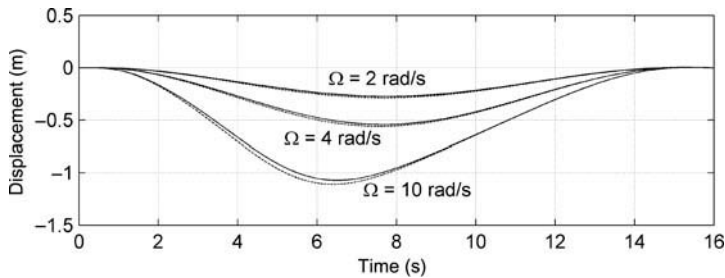


Figure 10. Tip deflection in spin-up maneuver at different angular velocities. Solid line: FEM model, dashed line: analytical model.

model was substructured in 4 elements (Figure 9), each one with a length $L = 2$ m and consisting of 20 B31 nodes, 15 eigenmodes were retained, while in the case of $\Omega = 10$ the beam was substructured in 8 elements, each one with a length $L = 1$ m.

Figure 10 shows a comparison between the results obtained with the analytic beam model (dashed line) and the FEM model (solid line), as it is apparent, a good agreement was obtained. The simulations based on the analytic beam model took 0.22 s of CPU time for $\Omega = 2$ rad/s and $\Omega = 4$ rad/s, 0.40 s for $\Omega = 10$ rad/s, while the simulations based on the FEM model took 2.06, 2.65, and 3.71 s, respectively.

In a second set of simulation experiments, gravity was added along the negative y -axis, retaining the model data of the previous simulations. As shown in Figure 11, the deflection of the beam tip in the plane of motion is de facto unchanged. On the other hand, the displacement along the y -axis (Figure 12) showed the 3D effects of centrifugal terms,

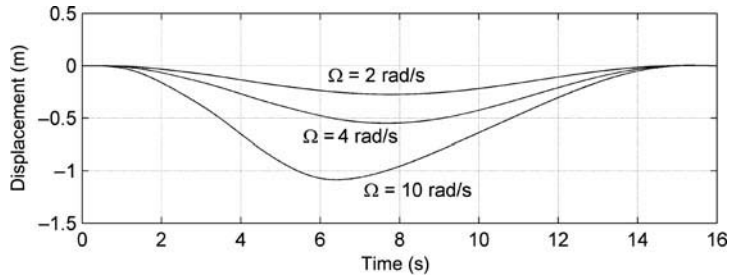


Figure 11. Tip deflection in the plane of motion with gravity at different angular velocities.

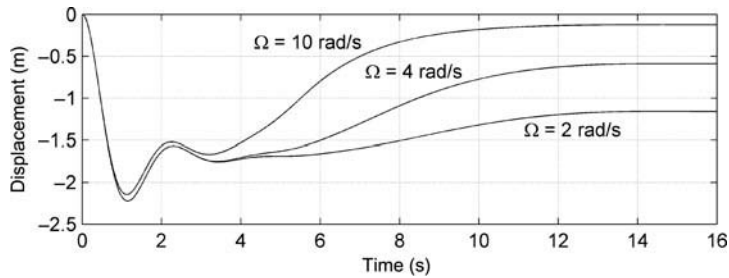


Figure 12. Tip deflection along y -axis with gravity at different angular velocities.

raising up the tip of the beam. The CPU time requested in these simulation experiments resulted longer, namely 6.59, 7.64 and 10.03 s for $\Omega = 2$, $\Omega = 4$ and $\Omega = 10$ rad/s, respectively.

6. Conclusion

A general approach to object-oriented modelling of general flexible multibody systems has been presented, which is fully compatible with the standard Modelica multibody library.

The motion equations of a general flexible body have been derived based on the FFR approach, while identifying the characteristic parameters, in particular, the set of inertia invariants. Exploiting the modular nature of the object-oriented approach, a mixed formulation (finite volume-finite element) can be easily adopted, allowing for the modelling of large deformations, by connecting several FEM components.

Since in general the body data cannot be analytically determined for complex geometries in the full 3D field of deformation, the possibility and convenience of importing the results of a FEM analysis, performed with standard packages, was investigated, and a good agreement has been obtained with analytical models in the case of flexible beams.

Flexible multibody analysis in object-oriented simulation tools can now be applied to real-world cases. In particular, the modelling of machine tools and automotive systems is currently underway, and will be addressed in future works.

Acknowledgements

The authors wish to thank Dr. Giacomo Bianchi and Dr. Nicola Cau, with Institute of Industrial Technologies and Automation (ITIA) – National Research Council (CNR) – for their really valuable help in using FEM tools.

Note

1. The internal representation of the orientation (i.e. the parametrization of the relative rotation matrix) in the Modelica multibody library is completely masked to the user. In addition to that, the orientation 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.

References

- [1] G. Ferretti, G. Magnani, and P. Rocco, *Virtual prototyping of mechatronic systems*, IFAC J. Annu. Rev. Control. 28 (2004), pp. 193–206.
- [2] P. Piela, T. Epperly, K. Westerberg, and A. Westerberg, *ASCEND: An object-oriented computer environment for modelling and analysis: The modelling language*, Comput. Chem. Eng. 15 (1991), pp. 53–72.
- [3] S. Mattsson and M. Andersson, *Omola – An Object-Oriented Modeling Language*, in *Recent Advances in Computer Aided Control Systems*, M. Jamshidi and C.J. Herget, eds., Elsevier Science, Amsterdam, 1993, pp. 291–310.
- [4] M. Oh and C. Pantelides, *A modeling and simulation language for combined lumped and distributed parameter systems*, Comput. Chem. Eng. 20 (1996), pp. 611–633.
- [5] C. Maffezzoni and R. Girelli, *MOSES: modular modeling of physical systems in an object-oriented database*, Math. Modelling Syst. 4 (1998), pp. 121–147.
- [6] F. Schiavo, L. Viganò, and G. Ferretti, *Object-oriented modelling of flexible beams*, Multibody Syst Dyn. 15 (2006), pp. 263–286.
- [7] The Modelica Association. *Modelica – A Unified Object-Oriented Language for Physical Systems Modeling*, Language Specification Version 3.1 (2009). Available at www.modelica.org.
- [8] Dynasim AB. *Dymola*, Lund, Sweden (2013). Available at <http://www.3ds.com/products/catia/portfolio/dymola>.
- [9] MathCore AB. *MathModelica*, Linköping, Sweden (2013). Available at <http://www.mathcore.com/>.
- [10] ITI GmbH. *SimulationX*, Dresden, Germany (2013). Available at <http://www.iti.de>.
- [11] PELAB, Linköping University. *OpenModelica*. Sweden (2013). Available at <https://www.openmodelica.org/>.
- [12] M. Otter, H. Elmqvist, and S. Mattsson, *The new Modelica multibody library*, in *3rd Modelica Conference*, November 3–4, Linköping, Sweden, 2003.
- [13] A.A. Shabana, *Dynamics of Multibody Systems*, Cambridge University Press, New York, 1998.
- [14] W. Ritz, *Über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen Physik*, Journal für die Reine und Angewandte Mathematik. 135 (1909), pp. 1–61.
- [15] R.R. Craig and M.C.C. Bampton, *Coupling of substructures for dynamic analyses*, AIAA J. 6 (1968), pp. 1313–1319.
- [16] P. Koutsovasilis and M. Beitelschmidt, *Comparison of model reduction techniques for large mechanical systems*, Multibody Syst Dyn. 20 (2008), pp. 111–128.
- [17] M. Lehner and P. Eberhard, *A two-step approach for model reduction in flexible multibody dynamics*, Multibody Syst Dyn. 17 (2007), pp. 157–176.
- [18] J. Fehr and P. Eberhard, *Simulation process of flexible multibody systems with non-modal model order reduction techniques*, Multibody Syst Dyn. 25 (2011), pp. 313–334.
- [19] C. Nowakowski, J. Fehr, M. Fischer, and P. Eberhard, *Model order reduction in elastic multibody systems using the floating frame of reference formulation*, in *7th Vienna International Conference on Mathematical Modelling – MATHMOD 2012*, February 15–17, Vienna, Austria, 2012.
- [20] U. Lugić, M.A. Naya, A. Luaces, and J. Cuadrado, *Efficient calculation of the inertia terms in floating frame of reference formulations for flexible multibody dynamics*, Proceedings of the Institution of Mechanical Engineers, Part K. J. Multibody Dyn. 223 (2009), pp. 147–157.
- [21] M. Géradin and A. Cardona, *Flexible Multibody Dynamics: A Finite Element Approach*, Wiley, Chichester, Great Britain, 2001.
- [22] S. Wu and E. Haug, *Geometric non-linear substructuring for dynamics of flexible mechanical systems*, Int. J. Numer. Methods Eng. 26 (1988), pp. 2211–2276.

- [23] A. Heckmann, *On the choice of boundary conditions for mode shapes in flexible multibody systems*, *Multibody Syst Dyn.* 23 (2010), pp. 141–163.
- [24] A. Heckmann, M. Otter, S. Dietz, and J.D. López, *The DLR FlexibleBodies library to model large motions of beams and of flexible bodies exported from finite element programs*, in *5th Modelica Conference*, September 4–5, Vienna, Austria, 2006.
- [25] Bausch-Gall GmbH. *FlexibleBodies Library*, München, Germany (2013). Available at <http://www.bausch-gall.de>.
- [26] Claytex Services Ltd. *FlexibleBody Library*, Coventry, UK (2013). Available at <http://www.claytex.com>.
- [27] Dassault Systmes Simulia Corp. *Abaqus Analysis Users Manual, Version 6.9* (2009). Available at <http://www.3ds.com/products/simulia/portfolio/>.
- [28] Ansys Release 11.0 Documentation (2009). Available at <http://www.ansys.com>.
- [29] MSC/NASTRAN Reference Manual, Version 68 – Lahey, Miller, et al. (1994). Available at <http://www.mscsoftware.com/products/cae-tools/msc-nastran.aspx>.
- [30] NX Nastran (2012). Available at <http://www.plm.automation.siemens.com/>.
- [31] PERMAS (2013). Available at <http://www.intes.de>.
- [32] SIMPACK AG. *FEMBS – the SIMPACK-Interface to FE-Tools* (2013). Gilching. Available at <http://www.simpack.com/fembs.html>.
- [33] O. Wallrapp, *Standardization of flexible body modeling in multibody system codes, Part I: Definition of standard input data*, *Mech. Based Des. Struct. Mach.* 22 (1994), pp. 283–304.
- [34] R. Schwertassek, O. Wallrapp, and A.A. Shabana, *Flexible multibody simulation and choice of shape functions*, *Nonlinear Dyn.* 20 (1999), pp. 361–380.
- [35] H. Sugiyama, A.A. Shabana, M.A. Omar, and W.Y. Loh, *Development of nonlinear elastic leaf spring model for multibody vehicle systems*, *Comput. Methods Appl. Mech. Eng.* 195 (2006), pp. 6925–6941.
- [36] R. Craig and A. Kurdila, *Fundamentals of Structural Dynamics*, John Wiley & Sons, New York, 2006.
- [37] T. Kurz, P. Eberhard, C. Henninger, and W. Schiehlen, *From Neweul to Neweul-M2: symbolic equations of motion for multibody system analysis and synthesis*, *Multibody Syst Dyn.* 24 (2010), pp. 25–41.
- [38] R. Schwertassek and O. Wallrapp, *Dynamik Flexibler Mehrkörpersysteme*, Vieweg, Wiesbaden, 1999.
- [39] L. Meirovitch, *Analytical Methods in Vibration*, Macmillan Publishing, New York, 1967.
- [40] MSC Software Corporation (2003). *ADAMS/Flex – Theory of flexible bodies*. Available at <http://www.mscsoftware.com>.
- [41] T.F. Project The Fast Lexical Analyzer (2013). Available at <http://ex.sourceforge.net/>.
- [42] Bison – GNU parser generator Bison (2013). Available at <http://www.gnu.org/software/bison/>.
- [43] J. Escalona, H. Hussien, and A. Shabana, *Application of the absolute nodal coordinate formulation to multibody system dynamics*, *J. Sound Vib.* 214 (1998), pp. 833–851.
- [44] G. Ferretti, F. Schiavo, and L. Viganò, *Object-oriented modelling and simulation of flexible multibody thin beams in Modelica with the finite element method*, in *4th Modelica Conference*, March 7–8, Hamburg-Harburg, Germany, 2005.
- [45] J. Mayo and J. Domínguez, *Geometrically non-linear formulation of flexible multibody systems in terms of beam elements: Geometric stiffness*, *Comput. Struct.* 59 (1996), pp. 1039–1050.
- [46] J. Gerstmayr and J. Schöberl, *A 3D finite element method for flexible multibody systems*, *Multibody Syst. Dyn.* 15 (2006), pp. 305–320.
- [47] J.S. Chen and C.L. Huang, *Dynamic analysis of flexible slider-crank mechanisms with non-linear finite element method*, *J. Sound Vib.* 246 (2001), pp. 389–402.
- [48] H. Absy and A. Shabana, *Geometric stiffness and stability of rigid body modes*, *J. Sound Vib.* 207 (1997), pp. 465–496.
- [49] M. Berzeri and A. Shabana, *Study of the centrifugal stiffening effect using the finite element absolute nodal coordinate formulation*, *Multibody Syst. Dyn.* 7 (2002), pp. 357–387.
- [50] J. Mayo, D. Garcia-Vallejo, and J. Domínguez, *Study of the geometric stiffening effect: Comparison of different formulations*, *Multibody Syst. Dyn.* 11 (2004), pp. 321–341.
- [51] M. Pascal, *Some open problems in dynamic analysis of flexible multibody systems*, *Multibody Syst. Dyn.* 5 (2001), pp. 315–334.
- [52] U. Lugris, M.A. Naya, J.A. Pérez, and J. Cuadrado, *Implementation and efficiency of two geometric stiffening approaches*, *Multibody Syst. Dyn.* 20 (2008), pp. 147–161.

- [53] I. Sharf, *Geometric stiffening in multibody dynamics formulations*, J. Guid. Control Dyn. 18 (1995), pp. 882–890.
- [54] P. Shi, J. McPhee, and G. Heppler, *A deformation field for Euler-Bernoulli beams with applications to flexible multibody dynamics*, Multibody Syst. Dyn. 5 (2001), pp. 79–104.
- [55] R.E. Valembois, P. Fiset, and J.C. Samin, *Comparison of various techniques for modelling flexible beams in multibody dynamics*, Nonlinear Dyn. 12 (1997), pp. 367–397.

Appendix A. Inertia invariants

Defining $\mathbf{S} = [\mathbf{s}_1 \quad \mathbf{s}_2 \quad \dots \quad \mathbf{s}_M]$ then, from the motion equations as given in [6]:

$$m = \int_V \rho \, dV = I^1, \quad (\text{A1})$$

$$m\ddot{\mathbf{d}}_C = \int_V \rho \ddot{\mathbf{u}} \, dV = \int_V \rho \ddot{\mathbf{u}}_0 \, dV + \int_V \rho (\widetilde{\mathbf{S}}\mathbf{q}) \, dV = \int_V \rho \ddot{\mathbf{u}}_0 \, dV + \sum_{i=1}^M \left(\int_V \rho \tilde{\mathbf{s}}_i \, dV \right) q_i = \tilde{\mathbf{I}}^2 + \sum_{i=1}^M \tilde{\mathbf{I}}_i^3 q_i, \quad (\text{A2})$$

$$\tilde{\mathbf{C}}_t^T = \int_V \rho \mathbf{S} \, dV = \int_V \rho [\mathbf{s}_1 \quad \mathbf{s}_2 \quad \dots \quad \mathbf{s}_M] \, dV = [\mathbf{I}_1^3 \quad \mathbf{I}_2^3 \quad \dots \quad \mathbf{I}_M^3], \quad (\text{A3})$$

$$\begin{aligned} \tilde{\mathbf{C}}_r^T &= \int_V \rho \ddot{\mathbf{u}} \mathbf{S} \, dV = \int_V \rho \ddot{\mathbf{u}}_0 \mathbf{S} \, dV + \int_V \rho (\widetilde{\mathbf{S}}\mathbf{q}) \mathbf{S} \, dV \\ &= \int_V \rho \ddot{\mathbf{u}}_0 \mathbf{S} \, dV + \sum_{i=1}^M \left(\int_V \rho \tilde{\mathbf{s}}_i \mathbf{S} \, dV \right) q_i = \mathbf{I}^4 + \sum_{i=1}^M \mathbf{I}_i^5 q_i, \end{aligned} \quad (\text{A4})$$

$$\mathbf{M}_e = \int_V \rho \mathbf{S}^T \mathbf{S} \, dV = \mathbf{I}^6, \quad (\text{A5})$$

$$\begin{aligned} \tilde{\mathbf{J}} &= \int_V \rho \ddot{\mathbf{u}}^T \ddot{\mathbf{u}} \, dV = \int_V \rho \ddot{\mathbf{u}}_0^T \ddot{\mathbf{u}}_0 \, dV + \int_V \rho \ddot{\mathbf{u}}_0^T (\widetilde{\mathbf{S}}\mathbf{q}) \, dV + \int_V \rho (\widetilde{\mathbf{S}}\mathbf{q})^T \ddot{\mathbf{u}}_0 \, dV + \int_V \rho (\widetilde{\mathbf{S}}\mathbf{q})^T (\widetilde{\mathbf{S}}\mathbf{q}) \, dV \\ &= \int_V \rho \ddot{\mathbf{u}}_0^T \ddot{\mathbf{u}}_0 \, dV - \sum_{i=1}^M \left(\int_V \rho \tilde{\mathbf{s}}_i^T \ddot{\mathbf{u}}_0 \, dV \right) q_i - \sum_{i=1}^M \left(\int_V \rho \ddot{\mathbf{u}}_0^T \tilde{\mathbf{s}}_i \, dV \right) q_i - \sum_{i=1}^M \sum_{j=1}^M \left(\int_V \rho \tilde{\mathbf{s}}_i^T \tilde{\mathbf{s}}_j \, dV \right) q_i q_j \\ &= \mathbf{I}^7 - \sum_{i=1}^M (\mathbf{I}_i^{8T} + \mathbf{I}_i^8) q_i - \sum_{i=1}^M \sum_{j=1}^M \mathbf{I}_{ij}^9 q_i q_j. \end{aligned} \quad (\text{A6})$$

Two more inertia invariants can be introduced in order to compute the vector \mathbf{h}_ω^f in (Equation (2)), given by [13]

$$\mathbf{h}_\omega^f = - \int_V \rho \mathbf{S}^T \tilde{\omega}^2 \ddot{\mathbf{u}} \, dV - 2 \int_V \rho \mathbf{S}^T \tilde{\omega} \mathbf{S} \dot{\mathbf{q}} \, dV = \left(\int_V \rho \left[-(\tilde{\omega} \times \ddot{\mathbf{u}}) \mathbf{S} \right]^T \, dV \right) \tilde{\omega} - 2 \left(\int_V \rho \mathbf{S}^T \tilde{\omega} \mathbf{S} \, dV \right) \dot{\mathbf{q}}. \quad (\text{A7})$$

Defining $\mathbf{S}^T = [\hat{\mathbf{s}}_1^T \quad \hat{\mathbf{s}}_2^T \quad \hat{\mathbf{s}}_3^T]$ one obtains

$$\begin{aligned} \mathbf{h}_\omega^f &= \left(\int_V \rho \begin{bmatrix} \tilde{\omega}_1 \ddot{\mathbf{u}}^T \mathbf{S} - \tilde{\omega}^T \tilde{u}_1 \mathbf{S} \\ \tilde{\omega}_2 \ddot{\mathbf{u}}^T \mathbf{S} - \tilde{\omega}^T \tilde{u}_2 \mathbf{S} \\ \tilde{\omega}_3 \ddot{\mathbf{u}}^T \mathbf{S} - \tilde{\omega}^T \tilde{u}_3 \mathbf{S} \end{bmatrix}^T \, dV \right) \tilde{\omega} - 2 \left(\tilde{\omega}_1 \int_V \rho (\hat{\mathbf{s}}_3^T \hat{\mathbf{s}}_2 - \hat{\mathbf{s}}_2^T \hat{\mathbf{s}}_3) \, dV \right) \dot{\mathbf{q}} \\ &\quad - 2 \left(\tilde{\omega}_2 \int_V \rho (\hat{\mathbf{s}}_1^T \hat{\mathbf{s}}_3 - \hat{\mathbf{s}}_3^T \hat{\mathbf{s}}_1) \, dV \right) \dot{\mathbf{q}} - 2 \left(\tilde{\omega}_3 \int_V \rho (\hat{\mathbf{s}}_2^T \hat{\mathbf{s}}_1 - \hat{\mathbf{s}}_1^T \hat{\mathbf{s}}_2) \, dV \right) \dot{\mathbf{q}}, \end{aligned} \quad (\text{A8})$$

which yields (Equation (12)) defining

$$\mathbf{D}_i = \begin{bmatrix} \mathbf{d}_{i1} \\ \mathbf{d}_{i2} \\ \mathbf{d}_{i3} \end{bmatrix} = \int_V \rho \tilde{u}_i \mathbf{S} \, dV, \quad \mathbf{I}_{ij}^{11} = \int_V \rho \hat{\mathbf{s}}_i^T \hat{\mathbf{s}}_j \, dV \quad i, j = 1, 2, 3. \quad (\text{A9})$$

In turn, matrix \mathbf{D}_i is given by

$$\begin{aligned} \mathbf{D}_i &= \int_V \rho \bar{u}_i \mathbf{S} dV = \int_V \rho (\bar{\mathbf{u}}_0 + \mathbf{S}\mathbf{q})_i \mathbf{S} dV = \int_V \rho \bar{u}_{0i} \mathbf{S} dV + \int_V \rho (\mathbf{S}\mathbf{q})_i \mathbf{S} dV \\ &= \begin{bmatrix} \mathbf{I}_{i1}^{10} \\ \mathbf{I}_{i2}^{10} \\ \mathbf{I}_{i3}^{10} \end{bmatrix} + \int_V \rho \hat{\mathbf{s}}_i \mathbf{q} \mathbf{S} dV = \begin{bmatrix} \mathbf{I}_{i1}^{10} \\ \mathbf{I}_{i2}^{10} \\ \mathbf{I}_{i3}^{10} \end{bmatrix} + \int_V \rho \mathbf{q}^T \hat{\mathbf{s}}_i^T \mathbf{S} dV = \begin{bmatrix} \mathbf{I}_{i1}^{10} \\ \mathbf{I}_{i2}^{10} \\ \mathbf{I}_{i3}^{10} \end{bmatrix} + \begin{bmatrix} \mathbf{q}^T \mathbf{I}_{i1}^{11} \\ \mathbf{q}^T \mathbf{I}_{i2}^{11} \\ \mathbf{q}^T \mathbf{I}_{i3}^{11} \end{bmatrix} \quad i = 1, 2, 3, \end{aligned} \quad (\text{A10})$$

where $(\mathbf{v})_i$ is the i th element of vector \mathbf{v} and

$$\begin{bmatrix} \mathbf{I}_{i1}^{10} \\ \mathbf{I}_{i2}^{10} \\ \mathbf{I}_{i3}^{10} \end{bmatrix} = \int_V \rho \bar{u}_{0i} \mathbf{S} dV \quad i = 1, 2, 3. \quad (\text{A11})$$

Invariants \mathbf{I}_{ij}^{10} and \mathbf{I}_{ij}^{11} may be however computed from invariants \mathbf{I}_i^8 and \mathbf{I}_{ij}^9 in fact, recalling that

$$\mathbf{I}_i^8 = \int_V \rho \begin{bmatrix} -\bar{u}_{03} s_{3i} - \bar{u}_{02} s_{2i} & \bar{u}_{02} s_{1i} & \bar{u}_{03} s_{1i} \\ \bar{u}_{01} s_{2i} & -\bar{u}_{03} s_{3i} - \bar{u}_{01} s_{1i} & \bar{u}_{03} s_{2i} \\ \bar{u}_{01} s_{3i} & \bar{u}_{02} s_{3i} & -\bar{u}_{02} s_{2i} - \bar{u}_{01} s_{1i} \end{bmatrix} dV, \quad (\text{A12})$$

$$\mathbf{I}_{ij}^9 = \int_V \rho \begin{bmatrix} -s_{3i} s_{3j} - s_{2i} s_{2j} & s_{2i} s_{1j} & s_{3i} s_{1j} \\ s_{1i} s_{2j} & -s_{3i} s_{3j} - s_{1i} s_{1j} & s_{3i} s_{2j} \\ s_{1i} s_{3j} & s_{2i} s_{3j} & -s_{2i} s_{2j} - s_{1i} s_{1j} \end{bmatrix} dV, \quad (\text{A13})$$

one obtains

$$(\mathbf{I}_{ij}^{10})_k = \begin{cases} (\mathbf{I}_k^8)_{ji} & i \neq j = 1, 2, 3 & k = 1, \dots, M \\ \frac{1}{2} [(\mathbf{I}_k^8)_{ii} - (\mathbf{I}_k^8)_{ll} - (\mathbf{I}_k^8)_{mm}] & i = j = 1, 2, 3 \\ & l, m = 1, 2, 3 & k = 1, \dots, M \end{cases} \quad (\text{A14})$$

$$(\mathbf{I}_{ij}^{11})_{kl} = \begin{cases} (\mathbf{I}_{kl}^9)_{ji} & i \neq j = 1, 2, 3 & k, l = 1, \dots, M \\ \frac{1}{2} [(\mathbf{I}_{kl}^9)_{ii} - (\mathbf{I}_{kl}^9)_{mm} - (\mathbf{I}_{kl}^9)_{nn}] & i = j = 1, 2, 3 \\ & m, n = 1, 2, 3 & k, l = 1, \dots, M. \end{cases} \quad (\text{A15})$$

Appendix B. Analytical computation of beam data

An example of analytical computation of the data reported in [Table 1](#) is here briefly discussed, with reference to the case of a beam. The beam is subdivided into N elements, each one in turn considered as a thin beam connected to the preceding and following ones. A local abscissa $\xi = x/\ell$ is defined, where x is the longitudinal local coordinate and ℓ is the length of a single element, and for each element a coordinate system is defined associated with the deformation degrees of freedom $\mathbf{q}_i = [q_{j1} \ q_{j2} \ q_{j3} \ q_{j4} \ q_{j5} \ q_{j6}]^T$, as shown in [Figure B1](#).

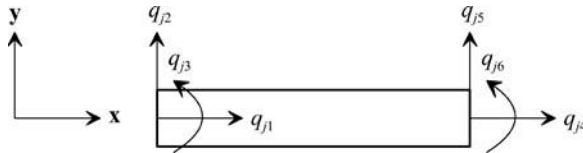


Figure B1. Element coordinate systems.

Accordingly, the generic position $\bar{\mathbf{u}}_j$ of a point on element j can be expressed as

$$\bar{\mathbf{u}}_j = \bar{\mathbf{u}}_{0j} + \mathbf{S}\mathbf{q}_j = [\xi_j \ell + (j-1)\ell] \mathbf{x} + \mathbf{S}\mathbf{B}_j \mathbf{q}, \quad (\text{B1})$$

where $\bar{\mathbf{u}}_{0j}$ is the undeformed position of the root of the j th element, \mathbf{x} is the unit vector along the beam axis, \mathbf{S} is the shape function matrix and \mathbf{B}_j is the so-called *connectivity matrix* used to relate the vector \mathbf{q} , containing the $M = 3N + 3$ deformation degrees of freedom for the whole beam, to the corresponding j th element [13,44]:

$$\mathbf{S} = \begin{bmatrix} 1 - \xi & 0 & 0 & \xi & 0 & 0 \\ 0 & 1 - 3\xi^2 + 2\xi^3 & \ell(\xi - 2\xi^2 + \xi^3) & 0 & 3\xi^2 - 2\xi^3 & \ell(\xi^3 - \xi^2) \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (\text{B2})$$

$$\mathbf{B}_j = [\mathbf{O}_{6 \times 3(j-1)} \quad \mathbf{I}_6 \quad \mathbf{O}_{6 \times 3(N-j)}], \quad j = 1, \dots, N. \quad (\text{B3})$$

If m and L are, respectively, the mass and the length of the whole beam and $\ell = L/N$ and $\mathbf{S}\mathbf{B}_j = [\mathbf{s}_{i1} \ \mathbf{s}_{i2} \ \dots \ \mathbf{s}_{i(3N+3)}]$ the inertia invariants can be computed as

$$\mathbf{I}^1 = m, \quad (\text{B4})$$

$$\mathbf{I}^2 = m [L/2 \quad 0 \quad 0]^T, \quad (\text{B5})$$

$$\mathbf{I}^3 = \frac{m}{N} \sum_{j=1}^N \int_0^1 \mathbf{S}\mathbf{B}_j \, d\xi, \quad (\text{B6})$$

$$\mathbf{I}^4 = \frac{m}{N} \sum_{j=1}^N \int_0^1 \tilde{\mathbf{u}}_{0j} \mathbf{S}\mathbf{B}_j \, d\xi, \quad (\text{B7})$$

$$\mathbf{I}_i^5 = \frac{m}{N} \sum_{j=1}^N \int_0^1 \tilde{\mathbf{s}}_{ij} \mathbf{S}\mathbf{B}_j \, d\xi, \quad (\text{B8})$$

$$\mathbf{I}^6 = \frac{m}{N} \sum_{j=1}^N \int_0^1 \mathbf{B}_j^T \mathbf{S}^T \mathbf{S}\mathbf{B}_j \, d\xi, \quad (\text{B9})$$

$$\mathbf{I}^7 = \frac{m}{N} \sum_{j=1}^N \int_0^1 \tilde{\mathbf{u}}_{0j}^T \tilde{\mathbf{u}}_{0j} \, d\xi, \quad (\text{B10})$$

$$\mathbf{I}_i^8 = \frac{m}{N} \sum_{j=1}^N \int_0^1 \tilde{\mathbf{u}}_{0j} \tilde{\mathbf{s}}_{ij} \, d\xi, \quad (\text{B11})$$

$$\mathbf{I}_{ij}^9 = \frac{m}{N} \sum_{k=1}^N \int_0^1 \tilde{\mathbf{s}}_{ik} \tilde{\mathbf{s}}_{jk} \, d\xi, \quad (\text{B12})$$

while the stiffness matrix \mathbf{K}_e is computed as

$$\mathbf{K}_e = \sum_{j=1}^N \mathbf{B}_j^T \mathbf{K} \mathbf{B}_j, \quad (\text{B13})$$

where \mathbf{K} is the stiffness matrix of a single element [13]. Considering only two connectors at the beam ends ($N_c = 2$) and placing the reference frame in a connector, say connector 1, the undeformed position of the other connector is $\bar{\mathbf{u}}_{02} = [L \quad 0 \quad 0]^T$ while, for the sake of simplicity, it can be assumed $\bar{\mathbf{A}}_2 = \mathbf{I}_3$.

Finally, the slices of the shape matrix \mathbf{S}_2 and $\hat{\mathbf{S}}_2$ are given by $\mathbf{S}_2 = \mathbf{S}_c \mathbf{B}_N$ and $\hat{\mathbf{S}}_2 = \hat{\mathbf{S}}_c \mathbf{B}_N$ where:

$$\mathbf{S}_c = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \hat{\mathbf{S}}_c = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (\text{B14})$$

Appendix C. Modelica code of motion equations

```

...
dq=der(q); //elastic coordinates velocities
ddq=der(ddq); //elastic coordinates accelerations
[data.inv1*identity(3), transpose(mDcbartilde), transpose(data.Ct)] // (2)
* [aa_a - g_0; za_a; ddq] = h_w_r+h_e_r; // (2)
[mDcbartilde, Jbar, transpose(Cr)] * [aa_a - g_0; za_a; ddq] // (2)
= h_w_theta+h_e_theta; // (2)
[data.Ct, Cr, data.Me] * [aa_a - g_0; za_a; ddq] = h_w_f+h_e_f -matrix(data.Ke * q) // (2)
- matrix((d/100)*(alpha*data.Me+beta*data.Ke) * dq); // (2)
mDcbar = data.inv2+ sum(data.inv3[:,i]*q[i] for i in 1:M); // (4)
mDcbartilde=skew(mDcbar); // (4)
transpose(Cr) = data.inv4+sum(data.inv5[i,,:]*q[i] for i in 1:M); // (6)
Jbar = data.inv7 // (8)
- sum((transpose(data.inv8[i,,:])+data.inv8[i,,:])*q[i] for i in 1:M) // (8)
- sum(sum(data.inv9[i,j,,:]*q[j] for j in i:M)*q[i] for i in 1:M); // (8)
h_w_r = matrix(-cross(wa_a, cross(wa_a, mDcbar)) // (9)
- 2*cross(wa_a, (transpose(data.Ct) * dq))); // (9)
h_w_theta = matrix(-cross(wa_a, Jbar * wa_a) - (der(Jbar) * wa_a) // (10)
- cross(wa_a, (transpose(Cr) * dq))); // (10)
h_w_f = matrix(transpose(hwfint)*wa_a // (12)
-2*[wa_a[1]*(data.inv11[3,2,,:]-data.inv11[2,3,,:]) // (12)
+wa_a[2]*(data.inv11[1,3,,:]-data.inv11[3,1,,:]) // (12)
+wa_a[3]*(data.inv11[2,1,,:]-data.inv11[1,2,,:])]*dq); // (12)
for i in 1:3 loop // (13)
for j in 1:3 loop // (13)
D[i,j,:] =data.inv10[i,j,:]+data.inv11[i,j,,:]*q; // (13)
end for; // (13)
end for; // (13)
sumd=sum(D[i,i,:]) for i in 1:3); // (13)
for i in 1:3 loop // (13)
hwfint[i,:]=wa_a[i]*sumd-wa_a*D[i,,:]; // (13)
end for; // (13)
h_e_r = matrix(sum(fb_a[i,:] for i in 1:data.Nc)); // (14)
h_e_theta = matrix(sum((taub_a[i,:]) // (15)
+ cross((data.r0b[i,:]+data.Sb[i,,:]*q), fb_a[i,:]) for i in 1:data.Nc)); // (15)
h_e_f = sum(matrix(transpose(data.Sb[i,,:])*fb_a[i,:]) // (16)
+ transpose(data.SbCap[i,,:])*taub_a[i,:]) for i in 1:data.Nc); // (16)
...

```