Flexible multibody dynamics using joint coordinates and the Rayleigh-Ritz approximation: the general framework behind and beyond Flex

E. Branlard*

7 February 2019

This is a pre-print version, please refer to the peer-reviewed version:
E. Branlard, *Flexible multibody dynamics using joint coordinates and the Rayleigh-Ritz approximation: the general framework behind and beyond Flex*, Wind Energy. 2019; 1–17. https://doi.org/10.1002/we.2327

1 Introduction

The conventional approach for the simulation of flexible multi-body systems consists in discretizing the bodies in finite elements and solving the constraints equation together with the equations of motion. The discretization and the account of the constraints usually results in a large number of degrees of freedom (DOF). This approach is for instance used by the aero-elastic code *Hawc2* [1] for the simulation of wind turbines. *Flex* [2] is another wide spread tool in the wind energy community which uses a different formulation that results in a very limited number of DOF. The general framework leading to this formulation is the topic of this article. The *ElastoDyn* module of the aero-elastic code *FAST* (now *OpenFAST*) [3, 4] uses a similar formulation, in particular, the early versions of both codes used the same degrees of freedoms to represent a wind turbine. The minor differences between the two will be highlighted in this article: *ElastoDyn* relies on Kane's method instead of the velocity transformation matrix, and *ElastoDyn* accounts directly for the radial shortening of beams.

Flex was originally written by Øye and has been further developed by different wind turbine industries. The equations of motion are set using two main methods: the joint-coordinates approach and the Rayleigh-Ritz (RR) approximation. The joint-coordinates method was described for rigid bodies by Nikravesh [5] and flexible bodies by Book [6]. In this method, a minimal set of DOF is obtained by choosing as coordinates the ones necessary to describe the joints motions. The constraints equation are then automatically satisfied. The Rayleigh-Ritz approximation, or assumed shape function method, consists in projecting the infinite number of DOF necessary to describe a flexible body onto a reduced and converging set of shape functions. The method is well documented in the book of Shabana [7].

The current article relies on the mentioned references and the available documentations of *Flex* [2, 8]. The general framework of the method is described in section 2 to ease the understanding of the specialized implementation of *Flex*. The recursive formulations which are obtained for open-loop systems are presented next. The possibility to couple the equations with finite-element and super-element models is discussed. The 4th section presents the specific treatments required when thin-beam approximations are used, including the possibility to account for beam torsion. The 5th section presents some validation cases and the 6th section introduces a simple wind turbine model as an application to illustrate the method.

2 Formulation of multibody-dynamics equations

2.1 Notations

The multibody system consists of b bodies. The variables related to body i are noted with a superscript. Geometrical points, vectors, and matrices are written with uppercase letters, bold lowercase letters and bold uppercase letters respectively. The vector from the point A to B is written $\boldsymbol{r}_{AB}$. The fixed inertial system is denoted with the index 0 and has its origin at the point $\boldsymbol{O}$. The coordinate system of body i has its origin at the body fixed point $\boldsymbol{O}^i$. The center of gravity (COG) of body i is located at the point $\boldsymbol{G}^i$. The following conventions are introduced to shorten the notations: $\boldsymbol{r}_A \triangleq \boldsymbol{r}_{OA}$, $\boldsymbol{r}^i \triangleq \boldsymbol{r}_{OO^i}$ and $\boldsymbol{s}^i_P \triangleq \boldsymbol{r}_{O^iP^i}$. A given vector $\boldsymbol{u}$ is expressed as a 3×1

*contact: branlard [at] gmail [dot] com

vector of Cartesian coordinates $[u_x;\ u_y;\ u_z]$. Semi-columns and commas are used respectively for vertical and horizontal stacking of scalars, vectors or matrices. The transformation matrix from the body j to the body i is written $\boldsymbol{R}_{ij}$ and its transpose $\boldsymbol{R}_{ij}^t$. A given vector $\boldsymbol{u}$ is expressed in the frame j and i as $\boldsymbol{u}\big|_j = \boldsymbol{R}_{ji}\ \boldsymbol{u}\big|_i$ where the notation "|" is used to explicitly define in which frame the coordinates are expressed. Vectors are assumed to be expressed in the reference inertial frame by default and body variables expressed in their body frame are written with primes: $\boldsymbol{u}'^i \triangleq \boldsymbol{u}^i\big|_i = \boldsymbol{R}_{i0}\ \boldsymbol{u}^i\big|_0 = \boldsymbol{R}_{i0}\boldsymbol{u}^i$. The matrices $\boldsymbol{R}_x$, $\boldsymbol{R}_y$ and $\boldsymbol{R}_z$ are the elemental rotation matrices, with e.g. $\boldsymbol{R}_x(t) = [1,\ 0,\ 0; 0,\ \cos t,\ -\sin t; 0,\ \sin t,\ \cos t]$. Given two vectors $\boldsymbol{u}$ and $\boldsymbol{t}$, the *skew symmetric matrix* $\text{skew}(\boldsymbol{u}) \triangleq \tilde{\boldsymbol{u}}$ is introduced such that $\tilde{\boldsymbol{u}}\ \boldsymbol{t} = \boldsymbol{u} \times \boldsymbol{t}$. The function $\text{diag}(\boldsymbol{A}, \boldsymbol{B})$ is used to form a block diagonal matrix from the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ given as arguments.

2.2 Single body dynamics

Degrees of freedom, velocity and acceleration vectors The position of body i with respect to the inertial frame of reference are described using the 3 Cartesian coordinates of its origin $\boldsymbol{r}^i$. A minimum of 3 coordinates, e.g. Bryant angles, describe its orientation. Their time derivatives are related to the angular velocity of the body $\boldsymbol{\omega}^i$. The velocity and acceleration of $\boldsymbol{O}^i$ are written $\boldsymbol{v}^i$ and $\boldsymbol{a}^i$ respectively. Flexible bodies are governed by an infinite number of DOF that provide the displacement of each point of the body. Under the Rayleigh-Ritz approximation [7], the flexible displacement field of the body is represented using a finite linear combination of *shape functions*. This approximation is valid if the sequence of linear combinations converges to the real displacement field as the number of shape function is increased. Usually a limited number of functions is sufficient when the system is excited below a given frequency. The shape functions of body i are three dimensional fields noted $\boldsymbol{\Phi}'^i_j = [\Phi'^i_{jx},\ \Phi'^i_{jy},\ \Phi'^i_{jz}]^t$ where j is an index going from 1 to the number of selected shape functions, n^i_f. The shape functions define a displacement field for each point of the undeformed body. The shape functions can be combined into a $3 \times n^i_f$ matricial field $\boldsymbol{\Phi}'^i = [\boldsymbol{\Phi}'^i_1,\ \ldots,\ \boldsymbol{\Phi}'^i_{n^i_f}]$. The position of a point P on the body i at a given time t is related to its undeformed position P_0 as:

$$\boldsymbol{r}'_P(t) = \boldsymbol{r}'^i(t) + \boldsymbol{s}'_P(t) = \boldsymbol{r}'^i(t) + \boldsymbol{s}'_{P_0} + \sum_{j=1}^{n^i_f} g^i_j(t)\boldsymbol{\Phi}'^i_j(\boldsymbol{s}'_{P_0}) = \boldsymbol{r}'^i + \boldsymbol{s}'_{P_0} + \boldsymbol{\Phi}'^i\ \boldsymbol{g}^i \tag{1}$$

where $\boldsymbol{g}^i = [g^i_1,\ \ldots,\ g^i_{n^i_f}]^t$ is the vector of *flexible* coordinates defining the amplitudes of the shape functions. In the last equality, the dependency with respect to $\boldsymbol{s}'_{P_0}$ and t have been dropped to shorten the notations. The generalized velocity and acceleration of a body are vectors of dimension $6 + n^i_f$ defined as $\mathbf{v}^i \triangleq [\boldsymbol{v}^i;\ \boldsymbol{\omega}^i;\ \dot{\boldsymbol{g}}^i]$ and $\mathbf{a}^i \triangleq [\dot{\boldsymbol{v}}^i;\ \dot{\boldsymbol{\omega}}^i;\ \ddot{\boldsymbol{g}}^i]$. The transformation matrix $\mathbf{R}_{0i} \triangleq \text{diag}(\boldsymbol{R}_{0i}, \boldsymbol{R}_{0i}, \boldsymbol{I}_{n^i_f})$ is defined to conveniently change the coordinates of these vectors, with $\boldsymbol{I}_n$ the identity matrix of size $n \times n$.

Dynamics equation The superscript i is temporarily dropped in this paragraph. The kinetic energy of a body is defined as $T \triangleq \frac{1}{2}\int \boldsymbol{v}^t_P \cdot \boldsymbol{v}_P\ \text{d}m = \frac{1}{2}\int \boldsymbol{v}'^t_P \cdot \boldsymbol{v}'_P\ \text{d}m$ where the integral is run over the entire volume of the body. The velocity of a body point P is obtained by differentiation of Equation 1, giving: $\boldsymbol{v}'_P = \boldsymbol{v}' + \boldsymbol{\omega}' \times \boldsymbol{s}'_P + \boldsymbol{\Phi}'\ \dot{\boldsymbol{g}} = [\boldsymbol{I}_3,\ -\tilde{\boldsymbol{s}}'_P,\ \boldsymbol{\Phi}'] \cdot \mathbf{v}'$. The kinetic energy is then $T = \frac{1}{2}\mathbf{v}'\mathbf{M}'\mathbf{v}'$ where $\mathbf{M}'$ is the symmetric mass matrix:

$$\mathbf{M}' = \begin{bmatrix} \boldsymbol{M}'_{xx} & \boldsymbol{M}'_{x\theta} & \boldsymbol{M}'_{xg} \\ & \boldsymbol{M}'_{\theta\theta} & \boldsymbol{M}'_{\theta g} \\ \text{sym.} & & \boldsymbol{M}'_{gg} \end{bmatrix} \tag{2}$$

with $\boldsymbol{M}'_{xx} = \int \boldsymbol{I}_3\,\text{d}m = M\boldsymbol{I}_3$, $\boldsymbol{M}'_{x\theta} = -\int \tilde{\boldsymbol{s}}'_P\,\text{d}m$, $\boldsymbol{M}'_{xg} = \int \boldsymbol{\Phi}'\,\text{d}m$, $\boldsymbol{M}'_{\theta\theta} = -\int \tilde{\boldsymbol{s}}'_P\tilde{\boldsymbol{s}}'_P\,\text{d}m$, $\boldsymbol{M}'_{\theta g} = \int \tilde{\boldsymbol{s}}'_P\boldsymbol{\Phi}'\,\text{d}m$, $\boldsymbol{M}'_{gg} = \int \boldsymbol{\Phi}'^t\boldsymbol{\Phi}'\,\text{d}m$, $\boldsymbol{s}'_P = \boldsymbol{s}'_{P_0} + \boldsymbol{\Phi}'\boldsymbol{g}$ and where M is the mass of the body. The block matrices $\boldsymbol{M}'_{xx}$, $\boldsymbol{M}'_{xg}$ and $\boldsymbol{M}'_{gg}$ are constant. The other terms may be expressed as function of shape integrals which are invariants and computed only once [7]. The shape integrals are defined in Appendix A. The terms of $\mathbf{M}'$ that involve $\boldsymbol{g}$ need to be updated at each time step.

The equations of motions are derived using Lagrange's equation. The generalized Newton-Euler equation is then obtained as [7]:

$$\mathbf{M}'\mathbf{a}' = \mathbf{f}' + \mathbf{f}'_v, \qquad \mathbf{f}'_v = -\dot{\mathbf{M}}'\dot{\mathbf{v}}' + \frac{1}{2}\left[\frac{\partial}{\partial \mathbf{x}'}(\mathbf{v}'^t\mathbf{M}'\mathbf{v}')\right]^t \tag{3}$$

where $\mathbf{f}_v$ is the quadratic velocity ("ficticious") force at $\boldsymbol{O}^i$ which arises from the derivation of the kinetic energy and which comprise the contribution from the gyroscopic and Coriolis force. The term $\mathbf{f}' = [\boldsymbol{f}'_x; \boldsymbol{f}'_\theta; \boldsymbol{f}'_g]$ consists of the generalized forces at $\boldsymbol{O}^i$ due to external and internal loads. For instance they are given for a volume force $\boldsymbol{p}'$ as:

$$\boldsymbol{f}'_x = \int \boldsymbol{p}'\left(\boldsymbol{s}'_P\right)\,\mathrm{d}v, \quad \boldsymbol{f}'_\theta = \int \boldsymbol{s}'_P \times \boldsymbol{p}'\left(\boldsymbol{s}'_P\right)\,\mathrm{d}v, \quad \boldsymbol{f}'_g = \int \boldsymbol{\Phi}'^t\left(\boldsymbol{s}'_P\right)\boldsymbol{p}'\left(\boldsymbol{s}'_P\right)\,\mathrm{d}v \tag{4}$$

For a rigid body, Equation 3 reduces to the Newton-Euler equation expressed in the body frame:

$$\begin{bmatrix} M\boldsymbol{I}_3 & -M\tilde{\boldsymbol{\rho}}' \\ M\tilde{\boldsymbol{\rho}}' & \boldsymbol{J}' \end{bmatrix} \cdot \begin{bmatrix} \dot{\boldsymbol{v}}' \\ \dot{\boldsymbol{\omega}}' \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}'_x - M\tilde{\boldsymbol{\omega}}'\tilde{\boldsymbol{\omega}}'\boldsymbol{\rho}' \\ \boldsymbol{f}'_\theta - \tilde{\boldsymbol{\omega}}'\boldsymbol{J}'\tilde{\boldsymbol{\omega}}' \end{bmatrix} \tag{5}$$

where $\boldsymbol{\rho} \triangleq \boldsymbol{r}_{O^iG^i}$ and $\boldsymbol{J}'$ is the inertia tensor of the body at $\boldsymbol{O}^i$, related to the inertia tensor at the COG by: $\boldsymbol{J}' \triangleq -\int \tilde{\boldsymbol{s}}'_P\,\tilde{\boldsymbol{s}}'_P\,\mathrm{d}m = \boldsymbol{J}'_G - M\tilde{\boldsymbol{\rho}}'\tilde{\boldsymbol{\rho}}'$. The terms $\boldsymbol{f}'_x$ and $\boldsymbol{f}'_\theta$ are the external force and moment at $\boldsymbol{O}^i$ exerted on the body.

The internal elastic forces due to the deformation of the body contribute to the potential energy of the flexible body. The virtual work due to the elastic forces is $\delta W = -\int \boldsymbol{\sigma}^t\delta\boldsymbol{\epsilon}\,\mathrm{d}v$ where $\boldsymbol{\sigma}$ is the 6-component stress vector and $\boldsymbol{\epsilon}$ is the 6-component engineering strain vector. The strain-displacement relationship provides the strain as function of the partial derivative of the displacement field $\boldsymbol{\Phi}'\boldsymbol{g}$ (see e.g. [9, p.112]), leading to $\boldsymbol{\epsilon} = \boldsymbol{D}\boldsymbol{\Phi}'\boldsymbol{g}$ where $\boldsymbol{D}$ is the differential operator of the strain-displacement relationship. The stresses are assumed to follow the generalized Hooke's law of elastic materials $\boldsymbol{\sigma} = \boldsymbol{E}\boldsymbol{\epsilon}$ where $\boldsymbol{E}$ is the elasticity matrix which is a function of the constant Lame's coefficient for an homogeneous isotropic material. The virtual work due to elastic deformation reduces to $\delta W = -\boldsymbol{g}^t\boldsymbol{K}_{gg}\delta\boldsymbol{g}$ where $\boldsymbol{K}_{gg} = \int(\boldsymbol{D}\boldsymbol{\Phi}')^t\boldsymbol{E}\boldsymbol{D}\boldsymbol{\Phi}\,\mathrm{d}v$ is the generalized stiffness matrix associated with the elastic DOF $\boldsymbol{g}$. Alternatively, non-linear elastic forces may be included in the theory by using a force term directly instead of a stiffness matrix in the system of equations. In the linear case, if the shape functions are selected as normalized natural mode shapes, the mass matrix $\boldsymbol{M}_{gg}$ is diagonal and the generalized stiffness matrix $\boldsymbol{K}_{gg}$ is also diagonal with its diagonal coefficients proportional to the ones of the mass matrix $K_{gg,jj} = \omega_j^2 M_{gg,jj}$ where ω_j is the natural frequency associated with the mode $\boldsymbol{\Phi}_j$ [10, p. 64]. Geometrical stiffening effects may be combined with this stiffness matrix as discussed in subsection 4.2.

2.3 Multibody formulation

Multibody dynamics The equations are first introduced in the inertial frame of reference and presented in the body-frame afterwards. The multibody dynamics equations are obtained by concatenating the dynamics equations of the individual bodies, written $\mathbf{M}^i\mathbf{a}^i = \mathbf{f}^i$, where the entire right hand side is gathered in the vector $\mathbf{f}^i$ for simplicity. Additional constraint forces are present due to the connections between the different bodies. Notations for joints and bodies will be introduced in subsection 3.1. Details are not relevant for the current development so the reader is referred e.g. to the book of Nikravesh [11]. The relations between the coordinates needed to satisfy the constraints are gathered into a vector of linearly independent equation $\boldsymbol{C} = 0$. The dynamics equations consist of the constraints equation and the equation of motion:

$$\mathbf{M}\mathbf{a} + \mathbf{C}_x\boldsymbol{\lambda} = \mathbf{f}, \quad \text{with} \quad \mathbf{M} = \mathrm{diag}(\mathbf{M}^1, \cdots, \mathbf{M}^b), \quad \mathbf{a} = [\mathbf{a}^1;\ \cdots;\ \mathbf{a}^b], \quad \mathbf{f} = [\mathbf{f}^1;\ \cdots;\ \mathbf{f}^b] \tag{6}$$

where $\mathbf{C}_x$ is the Jacobian of the constraint equation and $\boldsymbol{\lambda}$ is a vector of Lagrangian multipliers.

Generalized coordinates formulation The equations of motion may be expressed using a different set of coordinates than the reference coordinates of the individual bodies. These coordinates are then referred to as *generalized coordinates* and are further written $\mathbf{q}$. Given a set of generalized coordinates, there exists a *velocity transformation matrix* $\mathbf{B}$ that relates the velocity of these coordinates to the Cartesian velocities. The vector of velocities and accelerations are then:

$$\mathbf{v} = \mathbf{B}\dot{\mathbf{q}}, \qquad \mathbf{a} = \mathbf{B}\ddot{\mathbf{q}} + \dot{\mathbf{B}}\dot{\mathbf{q}} \tag{7}$$

The $\mathbf{B}$ matrix is also referred to as the *orthogonal complement array* [12] and the field of robotics use the term *Jacobian* matrix [13]. The elements of the $\mathbf{B}$ matrix may be obtained directly by application of Kane's method [14, 15], which is the approach used in the *ElastoDyn* module of *OpenFAST*, or using the recursive formulation presented in this article. Inserting Equation 7 into Equation 6 leads to $(\mathbf{B}^t\mathbf{M}\mathbf{B})\ddot{\mathbf{q}} + \mathbf{B}^t\mathbf{C}_x\boldsymbol{\lambda} = \mathbf{B}^t\left(\mathbf{f} - \mathbf{M}\dot{\mathbf{B}}\dot{\mathbf{q}}\right)$. Coordinates partitioning consists in selecting the vector $\mathbf{q}$ such that it consists of a minimal set of independent coordinates necessary to describe

the system. This condition is satisfied by choosing: the *joint coordinates* representing the allowed displacements of the joints between the bodies, and the *flexible* coordinates associates with the shape function of the flexible bodies. The constraints are then automatically satisfied [12], i.e. $\mathbf{B}^t\mathbf{C}_x \equiv 0$, and the equations of motion reduce to: $\mathbf{M}_q\ddot{\mathbf{q}} = \mathbf{f}_q$, with $\mathbf{M}_q \triangleq (\mathbf{B}^t\mathbf{M}\mathbf{B})$ and $\mathbf{f}_q = \mathbf{B}^t\left(\mathbf{f} - \mathbf{M}\dot{\mathbf{B}}\dot{\mathbf{q}}\right)$. The variables $\mathbf{M}_q$ and $\mathbf{f}_q$ are referred to as the *general mass matrix* and *generalized forces*. The system has the dimension of $\mathbf{q}$, noted n, which is in many cases significantly smaller than the original system of Equation 6 consisting of $(6 \times b + \sum_b n_f^i)$ equations. The block matrices forming the rows of the $\mathbf{B}$ matrix are written $\mathbf{B}^i$ such that $\mathbf{B} = [\mathbf{B}^1; \cdots ; \mathbf{B}^b]$, with the matrix $\mathbf{B}^i$ having dimensions $(6 + n_f^i) \times n$.

Using joint coordinates greatly simplifies the handling of the constraints compared to conventional methods. The constraints are inherently enforced by the method when the positions of each bodies are computed and when the forces are transferred between bodies, but no explicit treatment is required. Also, the bodies cannot drift away from each other since only allowable displacements are considered. The line-block matrices $\mathbf{B}^i$ are convenient for the numerical implementation. In particular, the generalized mass matrix can be seen as the superposition of mass matrices $\mathbf{M}_q^i$ from the individual bodies $\mathbf{M}_q = \sum_{i=1}^b \mathbf{B}^{i\,t}\mathbf{M}^i\mathbf{B}^i = \sum_{i=1}^b \mathbf{M}_q^i$ with $\mathbf{M}_q^i = \mathbf{B}^{i\,t}\mathbf{M}^i\mathbf{B}^i$.

Body-frame variables The previous developments are now written using variables expressed in the frame of each body: the mass matrix is $\mathbf{M}' \triangleq \mathrm{diag}(\mathbf{M}'^1, \cdots, \mathbf{M}'^b)$ and velocity transformation matrix $\mathbf{B}'$ is defined such that $\mathbf{v}' = \mathbf{B}'\dot{\mathbf{q}}$. The line block matrices of $\mathbf{B}$ and $\mathbf{B}'$ are related by $\mathbf{B}^i = \mathbf{R}_{0i}\mathbf{B}'^i$. It follows that the generalized mass matrix is the same whether expressed using variables in the reference frame or the body-frames:

$$\mathbf{M}_q = \sum_{i=1}^b \mathbf{B}^{i\,t}\mathbf{M}^i\mathbf{B}^i = \sum_{i=1}^b \mathbf{B}'^{i\,t}\mathbf{R}_{0i}^t\mathbf{M}^i\mathbf{R}_{0i}\mathbf{B}'^i = \sum_{i=1}^b \mathbf{B}'^{i\,t}\mathbf{M}'^i\mathbf{B}'^i \tag{8}$$

Similarly, the generalized force does not depend on the frame used:

$$\mathbf{f}_q \triangleq \mathbf{B}^t\mathbf{f} = \sum_{i=1}^b \mathbf{B}^{i\,t}\mathbf{f}^i = \sum_{i=1}^b \mathbf{B}'^{i\,t}\mathbf{R}_{0i}^t\mathbf{f}^i = \sum_{i=1}^b \mathbf{B}'^{i\,t}\mathbf{f}'^i = \mathbf{B}'^t\mathbf{f}' \triangleq \mathbf{f}'_q \tag{9}$$

The implementation may thus indifferently use variables expressed in the reference frame or in the individual body-frames.

Equation of motions The equations of motion to be solved are: $\mathbf{M}_q\ddot{\mathbf{q}} = \mathbf{f}_q$. The stiffness and damping matrices $\mathbf{K}_q$ and $\mathbf{D}_q$ are introduced when linear elastic forces and damping forces are assumed:

$$\mathbf{M}_q\ddot{\mathbf{q}} + \mathbf{D}_q\dot{\mathbf{q}} + \mathbf{K}_q\mathbf{q} = \mathbf{B}(\mathbf{f}_e + \mathbf{f}_v) \tag{10}$$

where $\mathbf{f}_e$ are the external forces and $\mathbf{f}_v$ the forces from the quadratic velocities. The latter are detailled in subsection 3.3. The equation of motions are integrated using a numerical scheme such as a fourth order Runge-Kutta method, or a non linear Newmark method. The damping matrix may be defined using Rayleigh damping $\mathbf{D}_q = \alpha\mathbf{M}_q + \beta\mathbf{K}_q$. Alternatively, when mode shapes are used as shape functions, individual stiffness proportional damping is used for each shape function, such that $\mathrm{D}_{q,jj} = \beta_j \mathrm{K}_{q,jj}$ with $\beta_j \triangleq \frac{2\zeta_j}{\omega_j}$ and ζ_j is the log decrement associated with the DOF j and ω_j its angular frequency.

Alternative derivation using virtual work The equations of motion for a single body or the full body assembly may be equivalently derived using the principle of virtual work. The component j of the generalized force vector associated to a given loading $\boldsymbol{p}$ is calculated such that the work of this force against an infinitesimal displacement δq_j is equal to the work done by the loading against the displacement field $\boldsymbol{\delta u}$ induced by δq_j, that is: $\mathrm{f}_{q,j}\,\delta q_j = \int \boldsymbol{p}\cdot\boldsymbol{\delta u}\,dv$. The column j of the generalized mass matrix $\mathbf{M}_q$ is computed as the generalized force associated with the inertial loading ("$\boldsymbol{p} = m\boldsymbol{a}_j$") obtained from the product of the distributed mass of the structure and the acceleration field $\boldsymbol{a}_j$ corresponding to a unit acceleration of $\ddot{q}_j$ while all the other accelerations are 0 [8, 16].

2.4 Summary

The algorithm of the joint coordinates Rayleigh-Ritz method is as follows. The bodies are initialized independently. The connection points and the nature of the joints between the bodies are set and stored by each parent body. The shape functions are set for each flexible bodies. Optionally, the shape integrals are computed. The indexes of the joint and flexible DOF are set. At each time step, a loop is run from parent bodies to children bodies to compute the body kinematics then $\mathbf{M}'^i$ and $\mathbf{B}'^i$. A loop from children to parent follows, to compute the generalized forces on

each body $\mathbf{f}'^i$, which include the rigid body forces that are passed by the joints to the parent body. The equation of motions are set using Equation 8 and Equation 9, and then solved. A new time-step or iteration follows. More details on the implementation of the method are provided in the next section.

3 Open-loop systems and recursive formulations

3.1 Notations

Bodies and joints The system considered is assumed to be an open-loop. Such system has a tree structure where each body i is connected to a unique parent body p and may have several children. A scheme with the notations is presented in Figure 1. The term *open chain* refers to the special case where the parent of $i+1$ is i, for $i = 1..b$. The

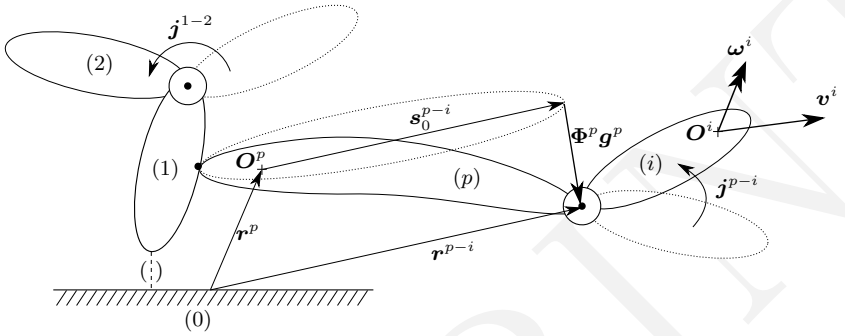

Figure 1: Notations for the tree system of bodies and joints. Dotted shapes represent the bodies shapes when the joint coordinates are 0 and the bodies are undeformed. Bodies (1) and (p) are rigidly attached. The joint rigid-body motion between bodies (p) and (i) is associated with the coordinates $\boldsymbol{j}^{p-i}$. Body (p) is flexible and its deformation is represented with the shape functions $\boldsymbol{\Phi}^p$ associated with the generalized coordinates $\boldsymbol{g}^p$.

bodies may be rigidly connected, like 1 and p on the figure, or connected via a joint, like p and i. The joints allows for rigid body motions between two bodies. The DOF associated with the joint motions between bodies p and i are written $\boldsymbol{j}^{p\text{-}i}$, of dimension $n^{p\text{-}i}$. On the figure, body p is flexible, and, according to Equation 1, the deformation of the connection point between body i and p is $\boldsymbol{\Phi}^p(\boldsymbol{s}_0^{p\text{-}i})\boldsymbol{g}^p$ where $\boldsymbol{s}_0^{p\text{-}i}$ is the vector from $\boldsymbol{O}^p$ to the joint p-i when the body is undeformed.

Ordering The bodies are ordered based on their depth in the tree, the root being the first body. The generalized coordinates are ordered accordingly. Using Figure 1 as an example, and assuming the first body is attached to the ground, the DOF are ordered as $\mathbf{q} = [\boldsymbol{g}^1;\ \boldsymbol{j}^{1\text{-}2};\ \boldsymbol{g}^2;\ \boldsymbol{g}^p;\ \boldsymbol{j}^{p\text{-}i};\ \boldsymbol{g}^i]$. If the first body is not rigidly attached to the ground, its translation and rotation coordinates are placed at the beginning of the generalized coordinates vector. For each body i, the number n^i is introduced such that the set of generalized coordinates $\mathbf{q}^i \triangleq [q_1;\ \ldots;\ q_{n^i}]$ is the minimal set sufficient to describe its position and orientation. The relation between $\mathbf{q}^i$ and $\mathbf{q}^p$ is $\mathbf{q}^i = [\mathbf{q}^p;\ \boldsymbol{g}^p;\ 0 \cdots\ ;\ \boldsymbol{j}^{p\text{-}i}]$ where the intermediate zeros account for other potential children of body p. The following recursive formula is obtained in the special case of an open chain: $\mathbf{q}^i = [\mathbf{q}^{i\text{-}1};\ \boldsymbol{g}^{i\text{-}1};\ \boldsymbol{j}^{(i-1)\text{-}i}]$ for $i = 2..n$. With the assumed ordering of DOF, the line block velocity matrices presented in subsection 2.3 are written as follows:

$$\begin{bmatrix} \boldsymbol{v}^i \\ \boldsymbol{\omega}^i \\ \dot{\boldsymbol{g}}^i \end{bmatrix} \triangleq \mathbf{B}^i\dot{\mathbf{q}} \triangleq \begin{bmatrix} \boldsymbol{B}_x^i & 0 & 0 \\ \boldsymbol{B}_\theta^i & 0 & 0 \\ 0 & \boldsymbol{I}^i & 0 \end{bmatrix} \dot{\mathbf{q}}, \qquad \text{viz.} \qquad \begin{bmatrix} \boldsymbol{v}^i \\ \boldsymbol{\omega}^i \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}_x^i \\ \boldsymbol{B}_\theta^i \end{bmatrix} \dot{\mathbf{q}}^i \tag{11}$$

where $\boldsymbol{B}_x^i$ and $\boldsymbol{B}_\theta^i$ have the dimensions $3 \times n^i$ and $\boldsymbol{I}^i$ has n_f^i rows and a number of columns which varies depending on how the DOF are ordered. The following paragraph will detail the expression of the velocity transformation matrices.

3.2 Velocity transformation matrices

This section presents the formalism of the velocity transformation matrices. The elements of these matrices corresponds to the partial velocities introduced in Kane's method [14, 15]. The recursive relations between the matrices of different bodies are established in this section for the case of an open-chain.

Rotational velocity transformation matrix The matrix $\boldsymbol{B}_\theta^i$ provides the rotational velocity of body i with respect to the ground, $\boldsymbol{\omega}_0^i$, as function of $\dot{\mathbf{q}}^i$, that is, the velocities of all the DOF leading to body i. Writing p the parent body of body i, the composition of the rotational velocities $\boldsymbol{\omega}_0^i = \boldsymbol{\omega}_0^p + \boldsymbol{\omega}_p^i$ leads to $\boldsymbol{B}_\theta^i \mathbf{q}^i = \boldsymbol{B}_\theta^p \mathbf{q}^p + \boldsymbol{\omega}_p^i = \boldsymbol{B}_\theta^p \mathbf{q}^p + \boldsymbol{R}_{0p} \hat{\boldsymbol{B}}_\theta^i [\dot{\boldsymbol{g}}^p; \dot{\boldsymbol{j}}^{p\text{-}i}]$ where $\hat{\boldsymbol{B}}_\theta^i$ provides $\boldsymbol{\omega}_p^i\big|_p$, the rotational velocity of body i with respect to body p expressed in the frame of body p, as function of the velocities of the DOF of body p and the joint between body p and i: $[\dot{\boldsymbol{g}}^p; \dot{\boldsymbol{j}}^{p\text{-}i}]$. To obtain $\boldsymbol{\omega}_p^i\big|_i$, the following matrices are introduced: $\tilde{\boldsymbol{B}}_\theta^i = \boldsymbol{R}_{ip} \hat{\boldsymbol{B}}_\theta^i$. The $\hat{\boldsymbol{B}}_\theta$ or $\tilde{\boldsymbol{B}}_\theta$ matrices are readily obtained for a spherical joint. An example is now given. If the DOF between body p and i are φ_y, φ_z and φ_x corresponding to three successive rotations, then the rotation matrix from the system i to the system p is $\boldsymbol{R}_{pi} = \boldsymbol{R}_y(\varphi_y)\boldsymbol{R}_z(\varphi_z)\boldsymbol{R}_x(\varphi_x)$. The composition of angular velocities with this assumed order of rotations leads to:

$$\hat{\boldsymbol{B}}_\theta^{y\to z\to x} = \begin{bmatrix} 0 & \boldsymbol{R}_y \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} & \boldsymbol{R}_y \boldsymbol{R}_z \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\ 1 & & \\ 0 & & \end{bmatrix}, \quad \tilde{\boldsymbol{B}}_\theta^{y\to z\to x} = \begin{bmatrix} \boldsymbol{R}_x^t \boldsymbol{R}_z^t \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} & \boldsymbol{R}_x^t \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} & 1 \\ & & 0 \\ & & 0 \end{bmatrix} \tag{12}$$

giving $\boldsymbol{\omega}_p^i\big|_p = \hat{\boldsymbol{B}}_\theta^i [\dot{\varphi}_y;\ \dot{\varphi}_z;\ \dot{\varphi}_x]$ for that example. In the general case, the rotation angles are a linear combination of the flexible and joint DOF. Once the local matrices $\hat{\boldsymbol{B}}_\theta^i$ or $\tilde{\boldsymbol{B}}_\theta$ are set, the following recursive formula is used to compute the $\boldsymbol{B}_\theta$ matrix: $\boldsymbol{B}_\theta^i = [\hat{\boldsymbol{B}}_\theta^1,\ \boldsymbol{R}_{01}\hat{\boldsymbol{B}}_\theta^2,\ \cdots,\ \boldsymbol{R}_{0p}\hat{\boldsymbol{B}}_\theta^i] = [\boldsymbol{B}_\theta^p,\ \boldsymbol{R}_{0p}\hat{\boldsymbol{B}}_\theta^i]$ which is equivalently expressed using the body-frame variables as $\boldsymbol{B}_\theta^{i\prime} \triangleq \boldsymbol{B}_\theta^i\big|_i \triangleq \boldsymbol{R}_{i0}\boldsymbol{B}_\theta^i = \boldsymbol{R}_{i0}[\boldsymbol{B}_\theta^p,\ \boldsymbol{R}_{0p}\hat{\boldsymbol{B}}_\theta^i] = [\boldsymbol{R}_{ip}\boldsymbol{B}_\theta^{p\prime},\ \tilde{\boldsymbol{B}}_\theta^i]$.

Translational velocity transformation matrix The velocity of body i is readily expressed as function of the velocity of body p. Using Equation 11, introducing $\boldsymbol{r}_{pi}$ the vector between the origins of body p and i, and defining $\boldsymbol{v}_p^i$ the relative velocity of body i with respect to p yields:

$$\boldsymbol{v}^i = \boldsymbol{v}^p + \boldsymbol{\omega}^p \times \boldsymbol{r}_{pi} + \boldsymbol{v}_p^i \quad \Rightarrow \quad \boldsymbol{B}_x^i \dot{\mathbf{q}}^i = \boldsymbol{B}_x^p \dot{\mathbf{q}}^p + (\boldsymbol{B}_\theta^p \dot{\mathbf{q}}^p) \times \boldsymbol{r}_{pi} + \boldsymbol{v}_p^i \tag{13}$$

The velocity $\boldsymbol{v}_p^i$ is a function solely of the flexible DOF of the parent body, $\boldsymbol{g}^p$, and the DOF of the joint $\boldsymbol{j}^{p\text{-}i}$. It is usually straight-forward to express the vector $\boldsymbol{r}_{ip}$ in the coordinate system p as function of these DOF. The velocity is then expressed as:

$$\boldsymbol{v}_p^i\Big|_p = \frac{d\,(\boldsymbol{r}_{ip})|_p}{dt} = \begin{bmatrix} \dfrac{\partial\,(\boldsymbol{r}_{ip})|_p}{\partial \boldsymbol{g}^p} & \dfrac{\partial\,(\boldsymbol{r}_{ip})|_p}{\partial \boldsymbol{j}^{p\text{-}i}} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{g}}^p \\ \dot{\boldsymbol{j}}^{p\text{-}i} \end{bmatrix} \triangleq \hat{\boldsymbol{B}}_x^i \begin{bmatrix} \dot{\boldsymbol{g}}^p \\ \dot{\boldsymbol{j}}^{p\text{-}i} \end{bmatrix} \tag{14}$$

where the matrix $\hat{\boldsymbol{B}}_x^i$ has been introduced to shorten notations. It corresponds to the Jacobian of the coordinates of the vector $\boldsymbol{r}_{ip}|_p$ with respect to the generalized coordinates involved between the parent of body i and body i. Inserting Equation 14 into Equation 13 leads to

$$\boldsymbol{B}_x^i \dot{\mathbf{q}}^i = \boldsymbol{B}_x^p \dot{\mathbf{q}}^p + (\boldsymbol{B}_\theta^p \dot{\mathbf{q}}^p) \times \boldsymbol{r}_{pi} + \boldsymbol{R}_{0p} \hat{\boldsymbol{B}}_x^i \begin{bmatrix} \dot{\boldsymbol{g}}^p \\ \dot{\boldsymbol{j}}^{p\text{-}i} \end{bmatrix} \tag{15}$$

Introducing a general formalism would be too lengthy since it requires to keep track of the DOF indexes. For simplicity, it is assumed that the DOF p and i are ordered next to each other, as in an open chain. In that case, Equation 15 leads to the following recursive relation (with $p = i-1$): $\boldsymbol{B}_x^i = [\boldsymbol{B}_x^p + \boldsymbol{B}_\theta^p \otimes \boldsymbol{r}_{pi},\ \boldsymbol{R}_{0p}\hat{\boldsymbol{B}}_x^i]$ where $\otimes$ is an operator that performs the individual cross product between each columns of the $\boldsymbol{B}_\theta^p$ matrix with the vector $\boldsymbol{r}_{pi}$. The matrix $\boldsymbol{B}_x^i$ has dimension $3 \times n^i$. The relation is expressed as follows in the body-frame: $\boldsymbol{B}_x^{i\prime} \triangleq \boldsymbol{R}_{i0}\boldsymbol{B}_x^i = [\boldsymbol{R}_{ip}\left(\boldsymbol{B}_x^{p\prime} + \boldsymbol{R}_{p0}(\boldsymbol{B}_\theta^p \otimes \boldsymbol{r}_{pi})\right),\ \boldsymbol{R}_{ip}\hat{\boldsymbol{B}}_x^i]$. The recursive relations are convenient for the numerical implementation. Yet, if an expression for the vector $\boldsymbol{r}_i$ is known analytically, the matrix $\hat{\boldsymbol{B}}_x^i$ is obtained by evaluation of the Jacobian matrix: $\boldsymbol{B}_x^i \equiv \frac{\partial \boldsymbol{r}_i}{\partial \mathbf{q}^i}$.

3.3 Accelerations and quadratic velocity forces

The relations between accelerations are needed to assess the quadratic velocity forces. The quadratic velocity vector consists of the terms $\mathbf{M}\dot{\mathbf{B}}\dot{\mathbf{q}}$ arising from Equation 7. The following notations are introduced to clarify the decomposition of the accelerations: $\boldsymbol{a}^i \triangleq \boldsymbol{a}_a^i + \boldsymbol{a}_v'^i$ and $\dot{\boldsymbol{\omega}}^i \triangleq \dot{\boldsymbol{\omega}}_a^i + \dot{\boldsymbol{\omega}}_v^i$ with $\boldsymbol{a}_a^i \triangleq \boldsymbol{B}_x^i \ddot{\mathbf{q}}^i$, $\boldsymbol{a}_v^i \triangleq \dot{\boldsymbol{B}}_x{}^i \dot{\mathbf{q}}^i$, $\dot{\boldsymbol{\omega}}_a^i \triangleq \boldsymbol{B}_\theta^i \ddot{\mathbf{q}}^i$ and

$\dot{\boldsymbol{\omega}}_v^i \triangleq \dot{\boldsymbol{B}}_\theta{}^i \dot{\mathbf{q}}^i$. Given a parent body p and child body i which origins are separated by a vector $\boldsymbol{s}_p^i = \boldsymbol{r}_{O^pO^i}$, the following recursive relations are obtained:

$$\boldsymbol{a}_v^i = \boldsymbol{a}_v^p + \boldsymbol{\omega}^p \times (\boldsymbol{\omega}^p \times \boldsymbol{s}_p^i) + 2\boldsymbol{\omega}^p \times \boldsymbol{v}_p^i + \dot{\boldsymbol{\omega}}_v^p \times \boldsymbol{s}_p^i \qquad \boldsymbol{a}_a^i = \boldsymbol{a}_a^p + \dot{\boldsymbol{\omega}}_a^p \times \boldsymbol{s}_p^i \tag{16}$$

$$\dot{\boldsymbol{\omega}}_v^i = \dot{\boldsymbol{\omega}}_v^p + \dot{\boldsymbol{\omega}}_{vp}{}^i + \boldsymbol{\omega}^p \times \boldsymbol{\omega}_p^i \qquad \dot{\boldsymbol{\omega}}_a^i = \dot{\boldsymbol{\omega}}_a^p + \dot{\boldsymbol{\omega}}_{ap}{}^i \tag{17}$$

where $\boldsymbol{\omega}_p^i$ is the angular velocity of body i with respect to body p. Given n successive rotations, Equation 17 gives: $\dot{\boldsymbol{\omega}}_v^n = \sum_{j=1}^{n-1} \boldsymbol{\omega}_{j-1}^j \times \sum_{i=j+1}^{n} \boldsymbol{\omega}_{i-1}^i$. This operation can be carried out using the successive columns of the $\boldsymbol{B}_\theta$ matrix from right to left since each of these columns corresponds to an elementary rotation. The recursive "v"-accelerations are convenient to avoid the computation of $\dot{\mathbf{B}}$. The relations for the "a"-accelerations above are not used since these components are obtained by the product $\mathbf{B}\ddot{\mathbf{q}}$. The full acceleration and v-acceleration of a point P belonging to body i are

$$\boldsymbol{a}_P^i = \boldsymbol{a}^i + \boldsymbol{\omega}^i \times (\boldsymbol{\omega}^i \times \boldsymbol{s}_P) + 2\boldsymbol{\omega}^i \times \dot{\boldsymbol{s}}_P + \dot{\boldsymbol{\omega}}^p \times \boldsymbol{s}_p^i + \ddot{\boldsymbol{s}}_P, \qquad \boldsymbol{a}_{vP}^i = \boldsymbol{a}_v^i + \boldsymbol{\omega}^i \times (\boldsymbol{\omega}^i \times \boldsymbol{s}_P) + 2\boldsymbol{\omega}^i \times \dot{\boldsymbol{s}}_P + \dot{\boldsymbol{\omega}}_v^p \times \boldsymbol{s}_p^i \tag{18}$$

with $\boldsymbol{s}'_P = \boldsymbol{s}'_{P_0} + \boldsymbol{\Phi}_i \boldsymbol{g}^i$, $\dot{\boldsymbol{s}}'_P = \boldsymbol{\Phi}_i \dot{\boldsymbol{g}}^i$ and $\ddot{\boldsymbol{s}}'_P = \boldsymbol{\Phi}_i \ddot{\boldsymbol{g}}^i$.

The quadratic velocity forces of a body i are the inertial loads from the v-accelerations at all points of the body. They are obtained using Equation 4 with $\boldsymbol{p} = -\rho \boldsymbol{a}_{vP}^i$ where ρ is the material density. The negative sign arises due to the placement of the inertial loads on the right hand side of Equation 10. Since gravity acts similarly on the mass distribution, the variable $\boldsymbol{g}_{\text{grav},v} = \boldsymbol{g}_{\text{grav}} - \boldsymbol{a}_v$ may be defined to integrate the influence of the gravity and quadratic velocities in a single computation using $\boldsymbol{p} = \rho \boldsymbol{g}_{\text{grav},v}$ in Equation 4.

3.4 Coupling with a finite element formulation with potential superelement reduction

The formalism described above is compatible with a finite element method (FEM). This is here illustrated by using two bodies rigidly connected at a point B where body a is described using the FEM and body b is represented using the Rayleigh-Ritz method. The method is similarly applicable when joints are present between the bodies. The DOF of body a are written $\mathbf{q}^a = [\mathbf{q}^0; \boldsymbol{g}^a]$ where $\boldsymbol{g}^a$ consists of n^a DOF representing the displacements and rotations of B. The vector $\mathbf{q}^0$ may be empty (Guyan reduction), may consist of other FEM-DOF, or a combination of Craig-Bampton DOF and FEM-DOF where the master DOF of the Craig-Bampton reduction contain at least $\boldsymbol{g}^a$. Body b consists of n^b flexible DOF, written $\mathbf{q}^b$. The velocity of point B, assumed to be the origin of body b, is a simple function of $\boldsymbol{g}^a$ and does not depend on $\mathbf{q}^0$. This implies that the first rows of $\mathbf{B}^b$ are zeros. Since the size of $\mathbf{q}^0$ may be large in a FEM representation, the application of Equation 8 may be simplified to avoid large matricial products and focus only on the effective products. The FEM mass matrix in the ground coordinate system is written as a 4×4 block matrix $\mathbf{M}^a = [\boldsymbol{M}_{00},\ \boldsymbol{M}_{0a}; \boldsymbol{M}_{a0},\ \boldsymbol{M}_{aa}]$. The velocity matrix up to body b is written $\mathbf{B}'^b = [\mathbf{0},\ \mathbf{0}; \mathbf{B}'_{ab},\ \mathbf{0}; \mathbf{0},\ \boldsymbol{I}_{n^b}]$ where the size of $\mathbf{B}'_{ab}$ is $6 \times n^a$. Last, the mass matrix of body b in its own coordinate system is given by Equation 2. It is written $\mathbf{M}'^b = [\boldsymbol{M}'_{rr},\ \boldsymbol{M}'_{rg}; \boldsymbol{M}'_{gr},\ \boldsymbol{M}'_{gg}]$ where the subscript "r" is used to combine the rigid body DOF "x" and "θ". The application of Equation 8 leads to:

$$\mathbf{M}_q = \begin{bmatrix} \boldsymbol{M}_{00} & \boldsymbol{M}_{0a} & \mathbf{0} \\ & \boldsymbol{M}_{aa} + \mathbf{B}'^t_{ab} \boldsymbol{M}'_{rr} \mathbf{B}'_{ab} & \mathbf{B}'^t_{ab} \boldsymbol{M}'_{rg} \\ \text{sym.} & & \boldsymbol{M}'_{gg} \end{bmatrix} \tag{19}$$

Similar operations are performed for the $\mathbf{K}_q$, $\mathbf{D}_q$ and $\mathbf{f}_q$. An example of application is provided in subsection 5.5. Despite the possibility to combine the relative and absolute coordinates formulations, the system of equation becomes stiffer and may require dedicated time-integration solvers or significantly smaller time steps. In the current study, a Newmark scheme was used when the system was coupled to a full FEM model, while the Runge-Kutta method was used when coupled with a Craig-Bampton reduction of a FEM model.

4 Specific treatments for thin-beams

The previous formalism is applied to beams. The equations are simplified since the volume integrals are replaced by line integrals over the mean line of the beam. The thin-beam theory assumptions introduce further simplifications, at the cost of requiring ad-hoc corrections. These corrections are discussed in this section.

4.1 Thin-beam body: notations and assumptions

A single body approximated as a thin beam is considered. The superscript i is dropped in this section. All vectors are expressed in the body frame and the prime notation is also dropped. The beam of length L has the following properties along its span: cross section A [m^2], planar second moment of area I_y [m^4], polar second moment of area I_x [m^4], torsion constant K_t [m^4], Young modulus E [N m^{-2}], shear modulus G [N m^{-2}], density ρ [kg m^{-3}] and linear mass $m = \rho A$ [kg m^{-1}]. The notations used for the beam deflection are illustrated in Figure 2. Small deflections

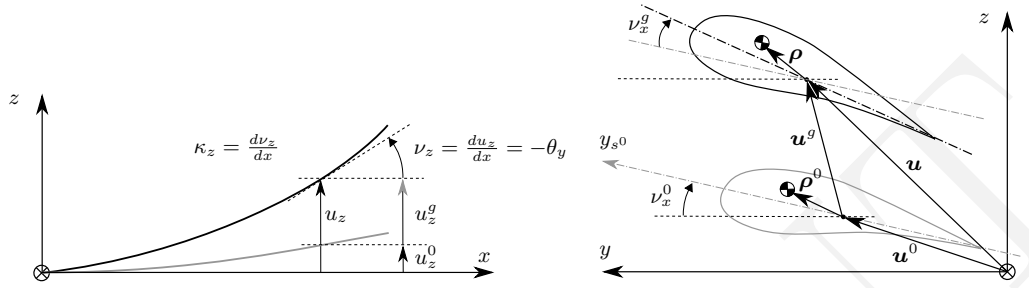

Figure 2: Notations used for a thin-beam. Left: mean-line of the beam in the $x-z$ plane. The deflection, slope and curvature are respectively noted u, ν and κ. Right: undeflected and deflected cross section in the case where the center of gravity is outside of the mean beam line ($\rho^0 \neq 0$) and where torsion is present ($\nu_x^g \neq 0$). The undeflected mean line is shifted by a vector $\boldsymbol{u}^0$ compared to the reference x-axis.

are assumed (i.e. $|\theta_y| \lesssim 10^\circ$, $|\theta_z| \lesssim 10^\circ$), and the mean line of the beam is parametrized using the coordinate x, referred to as the *axial* direction. The deflection, slope and curvature of the mean line of the beam in the z-direction are written u_z, v_z and κ_z respectively. Similar notations are used for the y-direction. The vector $\boldsymbol{u}$ is formed by introducing $u_x = x$. The undeflected mean line of the beam is defined by $\boldsymbol{s}_{P_0} \triangleq \boldsymbol{u}^0 = [x,\ u_y^0,\ u_z^0]^t$. The undeflected COG of the beam is assumed to be at $\boldsymbol{\rho}^0 = [0,\ \rho_y^0,\ \rho_z^0]^t$ from the mean line, i.e. : $\boldsymbol{s}_{G_0} \triangleq \boldsymbol{u}^0 + \boldsymbol{\rho}^0$. For a wind turbine blade the function $\boldsymbol{u}^0$ would correspond to the pre-bend and pre-sweep shapes. The shape function j at the mean line of the beam is noted $\boldsymbol{u}^j \triangleq \boldsymbol{\Phi}_j(\boldsymbol{s}_{P_0})$. They are computed using an iterative procedures [16] or a FEM which may account for material couplings, the direction of the principal axes and the initial mean line of the beam. The position of the deflected mean line is $\boldsymbol{s}_P \triangleq \boldsymbol{u} = \boldsymbol{s}_{P_0} + g_j \boldsymbol{\Phi}_j(\boldsymbol{s}_{P_0}) = \boldsymbol{u}^0 + \boldsymbol{u}^g$ where $\boldsymbol{u}^g = g_j \boldsymbol{u}^j$ and where summation over the repeated index is assumed. The COG is located at: $\boldsymbol{s}_G = \boldsymbol{s}_{G_0} + g_j \boldsymbol{\Phi}_j(\boldsymbol{s}_{G_0}) = \boldsymbol{u}^0 + \boldsymbol{u}^g + \boldsymbol{\rho}$. The functions $\boldsymbol{\rho}$ and $\boldsymbol{\rho^0}$ are identical when torsion is neglected. Torsion will be discussed in subsection 4.3. The slope and curvature of the mean line are similarly defined as function of the initial values and the shape functions: $\boldsymbol{\nu} = \boldsymbol{\nu}^0 + g_j \boldsymbol{\nu}^j$ and $\boldsymbol{\kappa} = \boldsymbol{\kappa}^0 + g_j \boldsymbol{\kappa}^j$. The component ν_x^0 is the structural twist defined as the angle of the principal axis y_s^0 of the section about the reference x-axis. The mass matrix is readily obtained by evaluating the integrals of Equation 2 , replacing the volume integrals with line-integrals over the beam span and along the COG, e.g.: $\boldsymbol{M}_{x\theta} = -\int_0^L \tilde{\boldsymbol{s}}_G \, m \, dx$. The stiffness matrix is obtained by integration of the product of the local curvatures and the beam stiffness:

$$K_{gg,ij} = \int_0^L \left[EI_{z_s^0} \kappa_{y_s^0}^i \kappa_{y_s^0}^j + EI_{y_s^0} \kappa_{z_s^0}^i \kappa_{z_s^0}^j + GK_t \kappa_{x_s^0}^i \kappa_{x_s^0}^j \right] dx \tag{20}$$

where the notation s^0 indicates that the values are expressed in the local coordinate system aligned with the undeformed principal axes of the cross section. The transformation matrix from the body coordinate system to the local cross section is assumed to be $\boldsymbol{R}_{s^0 B} = \boldsymbol{R}_x^t(\nu_x^0) \boldsymbol{R}_z^t(-\nu_z^0) \boldsymbol{R}_y^t(\nu_y^0)$. The order of the last two rotations is irrelevant given the small angle approximation. If mode shapes are used, the stiffness matrix is diagonal with its coefficients equal to: $K_{gg,jj} = \omega_j^2 M_{gg,jj}$. In practice, $\mathbf{K}_{gg}$ is kept constant during the simulation.

The thin beam theory assumption, referred to as the Euler-Bernoulli beam theory, assumes that the plane cross-sections of the beam, which are normal to the undeformed centroidal axis, remain plane after bending and remain normal to the deformed axis. No axial loads are usually included in the classical thin-beam theory. Also, no axial-displacements of the sections are assumed (i.e. $u_x = u_x^0 = x$), which implies that even if an axial load is present, no work is done. Corrections are introduced in the next paragraphs to account for the work done by axial loads and correctly assess the kinetic and potential energy of a beam.

4.2 Thin-beam theory with axial loading

A beam deflected along z is used in this section but the results apply to deflections in both the z and y directions.

Equivalent axial load A beam with axial and transverse loading is considered. A distributed transverse loading p_z is assumed for simplicity in this paragraph. The axial loading T along the beam consists of a distributed loading p_x and n_x point-wise forces T_k acting at the location x_k for $k = 1..n_x$. The equilibrium equations in the z and x direction are obtained similarly to the classical Euler-Bernoulli beam theory equations to give:

$$\frac{\partial^2}{\partial x^2}\left[EI_y\kappa_z^2\right] + m\frac{\partial^2 u_z}{\partial t^2} = p_z + \frac{\partial}{\partial x}\left[T\nu_z\right], \qquad \frac{\partial T}{\partial x} = -p_x - \sum_{k=1}^{n_x} T_k\delta(x - x_k) \tag{21}$$

By comparison with the equation without axial loading (i.e. $T = 0$), it is seen that the axial loading contributes to the transverse loading with an apparent transverse load given by:

$$p_z^T \triangleq \frac{\partial}{\partial x}\left[T\nu_z\right] = T\kappa_z - \nu_z p_x - \nu_z \sum_{k=1}^{n_x} T_k\delta(x - x_k) \tag{22}$$

The superscript T is used for variables related to the axial loading. The equation on the right of Equation 21 is integrated as $T = \int_x^L p_x g\, dx' + \sum_{x_k \geq x} T_k$ where a non-zero value of $T(L)$ is accounted for by T_{n_x} with $x_{n_x} = L$.

Formulation in term of moment and shear force At a given axial position x, the moment around y due to the axial loading is: $M_y^T = \int_x^L \left[p_x + \sum_{k=1}^{n_x} T_k\delta(x' - x_k)\right]\left[u_z(x') - u_z\right] dx'$. The shear force corresponding to this moment is obtained as $S_z^T = \frac{\partial M_y^T}{\partial x} = -\nu_z \int_x^L \left[p_x + \sum_{k=1}^{n_x} T_k\delta(x' - x_k)\right] dx'$ where a careful derivation under the integral sign is required. It is seen that the axial loading has a restoring effect for a positive axial loading. Similar to a spring, it tends to deflect the beam towards a zero-slope. The distributed transverse load balancing this shear force is:

$$p_z^T = -\frac{\partial S_z^T}{\partial x} = \kappa_z\left[\int_x^L p_x\, dx' + \sum_{x_k \geq x} T_k\right] - \nu_z\left[p_x + \sum_{k=1}^{n_x} T_k\delta(x - x_k)\right] \tag{23}$$

where a careful derivation under the integral sign is again required. The above result is seen to be equivalent to Equation 22.

Integration of the equivalent load The expression of the kinetic and potential energy associated with the axial load will require the determination of integrals of the form $\int_0^L f\, p_z^T\, dx$ where f is an arbitrary differentiable function. The integral can be expressed indifferently as:

$$\int_0^L f p_z^T\, dx = \int_0^L f\left[T\kappa_z - \nu_z p_x\right] dx = \int_0^L f\frac{\partial[T\nu_z]}{\partial x}\, dx = \left[f\nu_z T\right]_0^L - \int_0^L f'\nu_z T\, dx \tag{24}$$

The boundary term $[-]_0^L$ often vanishes due to the values taken at the boundaries.

Generalized force from axial loading and geometrical stiffness matrix The additional generalized force, $\boldsymbol{f}_g^T$, associated with the axial loading (i.e. the equivalent loading p_z^T) is defined by Equation 4 with $\boldsymbol{p}^T = p_z^T \boldsymbol{e}_z$:

$$f_g^{T,i} = \int \boldsymbol{\Phi}^i\left(\boldsymbol{s}_P\right) \cdot \boldsymbol{p}^T\left(\boldsymbol{s}_P\right)\, \mathrm{d}v = \int_0^L u_z^i p_z^T\, dx = \int_0^L u_z^i \frac{\partial}{\partial x}\left[\nu_z T\right] dx \tag{25}$$

This force is commonly referred to as the *centrifugal stiffening* force when the axial loading is due to the centrifugal inertia of a rotating beam, or *beam softening* when the axial loading is due to the self-weight of a beam and an axial force due to a structure located on top of a beam. In general, for any axial loading, the generalized force correction derived above needs to be included in the calculation. Such correction is required because the beam is here assumed not to move in the axial direction and hence the work contribution from this axial force is otherwise missing. For instance, the self weight would provide $p_x = \pm mg$ and the top-mass weight $T_L = \pm M_{\text{top}} g$ where g is the acceleration of gravity and M_{top} the overhead mass. Using the integration by part from Equation 24 and expanding $\nu_z = \sum_j \nu_z^j q_j$ gives:

$$f_g^{T,i} = \left[u_z^i \nu_z T\right]_0^L - \int_0^L \nu_z^i \nu_z T\, dx = -\sum_j K_{ij}^T q_j, \qquad K_{ij}^T = \int_0^L \nu_z^i \nu_z^j T\, dx - \left[u_z^i \nu_z^j T\right]_0^L \tag{26}$$

The generalized force can thus be expressed by the matricial product $-\boldsymbol{K}^T\boldsymbol{q}$ where $\boldsymbol{K}^T$ is the stiffness matrix corresponding to the axial force contribution, commonly referred to as the *geometrical stiffness* matrix. This matrix is to be added to $\boldsymbol{K}_{gg}$ defined in Equation 2.2. Similar terms in y are simply added to Equation 26 if the beam is also deflected along y. This approximation follows from the thin-beam assumption and non-linear coupling terms are effectively neglected. The expression of the geometrical stiffness matrix may be obtained from the potential energy. As mentioned earlier, the axial load T acts like a spring constant restoring the beam slope ν_z to its equilibrium. The potential energy of the beam due to the axial loading T is then

$$V^T = \frac{1}{2}\int_0^L T\nu_z(x')^2\,dx' = \frac{1}{2}\sum_i\sum_j \int_0^L T\nu_z^i(x')\nu_z^j(x')\,dx'\;q_iq_j \approx \frac{1}{2}\sum_i\sum_j K_{ij}^T q_i q_j \tag{27}$$

Generalized force from axial inertia and mass matrix A deflection of the beam will imply a rotation of the mass of its cross section by an angle $-v_z$ compared to its undeflected position. This effect will have an impact on the first line and first column of the mass matrix. The effect is captured by the volume integrals of Equation 2 but is not captured if the cross-section of the beam is represented by a concentrated mass since this concentrated mass is assumed not to be displaced axially in the thin-beam theory, and hence, not to work in the axial direction. An axial acceleration of the beam yet implies an axial inertial load along the beam. This load does not produce any work axially due to the thin-beam assumption, but it can be accounted for by using the transverse load equivalent to this axial load (i.e. Equation 22). The first column of the mass matrix is then computed using the principle of virtual work. A unit axial acceleration creates an axial inertial field $p_x = m \times 1$, which is equivalently represented by p_z^T and p_y^T. The generalized force associated with this inertial field are computed according to Equation 4. They provide the elements of the body mass matrix:

$$M_{15} = -\int_0^L x p_z^T\,dx = \int_0^L \nu_z T\,dx - [x\nu_z T]_0^L, \qquad M_{16} = \int_0^L x p_y^T\,dx = -\int_0^L \nu_y\,T\,dx + [x\nu_y T]_0^L \tag{28}$$

$$M_{1(6+j)} = \int_0^L \boldsymbol{u}^j\cdot\boldsymbol{p}^T\,dx = -\int_0^L \boldsymbol{\nu}^j\cdot\boldsymbol{\nu}\,T\,dx + [\boldsymbol{u}^j\cdot\boldsymbol{\nu}T]_0^L \tag{29}$$

with $T = \int_x^L m dx' + \sum_{x_k\geq x} M_k$ and $p_\bullet^T = T\kappa_\bullet - \nu_\bullet m - \nu_\bullet \sum_{k=1}^{n_x} M_k\delta(x-x_k)$. These values are used instead of the ones that would be obtained from a direct application of Equation 2 for a thin-beam.

4.3 Beam torsion

Torsion is accounted for by the formalism presented in section 2. Care is yet required when standard beam assumptions are used. In the current implementation the shape functions are obtained using a finite element procedure that accounts for material couplings and the position of the shear center of the cross section. The shape function j is characterized by the displacements of the mean line u_y^j and u_z^j and the torsion ν_x^j about the mean line. The total torsion angle is $\nu_x = \nu_x^0 + \nu_x^g = \nu_x^0 + g_j\nu_x^j$ where ν_x^0 is the initial twist angle and ν_x^g is the torsion angle due to the deformation of the body. The notations are illustrated on the right of Figure 2.

Position of the center of mass Due to the rotation ν_x^g the COG initially at position $\boldsymbol{\rho}^0$ compared to the mean line is displaced to $\boldsymbol{\rho} = \boldsymbol{R}_{\boldsymbol{x}}(\nu_x^g)\boldsymbol{\rho}^0$. The COG location is then $\boldsymbol{s}_G = \boldsymbol{u}^0 + g_j\boldsymbol{u}^j + \boldsymbol{R}_{\boldsymbol{x}}(\nu_x^g)\boldsymbol{\rho}^0$. This formula may be used to compute the mass matrix. Alternatively, if use is to be made of the shape integrals, it is possible to introduce an approximation for small torsion angle $\cos\nu_x^g \approx 1$ and $\sin\nu_x^g \approx g_j\nu_x^j$, leading to the following displacement field:

$$\boldsymbol{s}_G - \boldsymbol{s}_{G_0} \approx g_j \begin{bmatrix} 0 \\ u_y^j - \rho_z^0\nu_x^j \\ u_z^j + \rho_y^0\nu_x^j \end{bmatrix} = g_j\tilde{\boldsymbol{\Phi}}_j(s_{G_0}) \tag{30}$$

The displacement of the COG is then obtained as a linear combination of the newly introduced shape functions $\tilde{\boldsymbol{\Phi}}_j$.

Moment of inertia As mentioned in subsection 4.1, the volume integrals are conveniently replaced by line integrals over the COG line in the thin-beam formulation. An account of the polar moment of inertia is yet required when torsion is considered. The polar moment of inertia of the beam cross sections about the COG is noted j_G [kg m]. The following terms need to be accounted for in the mass matrix when line integrals are used:

$$M_{44}: \int_0^L j_G\,dx, \qquad M_{4(6+j)}: \int_0^L j_G\,\nu_x^j\,dx, \qquad M_{(6+i)(6+j)}: \int_0^L \nu_x^i\,j_G\,\nu_x^j\,dx \tag{31}$$

Non-linear correction Similar to the correction presented in subsection 4.2, any bending moments acting around the y and z axis of the beam will add an extra contribution to the torsional load p_x. This extra contribution is obtained as $p_x^{\mathrm{M}} = -\kappa_y M_y - \kappa_z M_z$. These corrections are applied as an extra force term, for any bending moment from inertial or external loads.

5 Simple validations

5.1 Effect of top mass and self-weight (tower/monopile)

A uniform vertical beam with a top mass M_t is considered. A static analysis is used to illustrate the effect of the self-weight and top-mass correction on the tower and/or monopile frequency. The following values are used for the beam: $L = 100$m, $EI = 2 \times 10^{12}$ Nm^2 and $m = 9 \times 10^3$ kg m^{-1}. Only the flexible DOF of the beam are considered, not the rigid body translation and rotation.

Simple Rayleigh Ritz-approximation The first mode of a uniform tower is well approximated by $u_0 = (1 - \cos(\tilde{x}\pi/2))$, which corresponds to the mode shape for a massless beam with a top-mass, and where $\tilde{x} = x/L$. The values of $\nu_0 = \frac{\partial u_0}{\partial x}$ and $\kappa_0 \frac{\partial \nu_0}{\partial x}$ are obtained. This Rayleigh-Ritz approximation leads to the following quantities $M_q = \int_0^L m u_0^2 dx = mL\frac{3\pi-8}{2\pi}$, $K_q = \int_0^L EI\kappa_0^2 dx = \frac{EI\pi^4}{32L^3}$ and $v_2 = \int_0^L \nu_0^2 dx = \frac{\pi^2}{8L}$. The stiffening of the tower due to a top mass M_t is obtained by inserting $T = -M_t g$ and ν_0 into Equation 26, giving: $K_{MT}^T = -M_t g \int_0^L \nu_0^2\, dx = -M_t g \frac{\pi^2}{8L}$, $\frac{K_{MT}^T}{K_q} = -\frac{M_t g}{P_{cr}}$ where $P_{cr} = \frac{\pi^2 EI}{4L^2}$ is the critical buckling load of the massless beam, introduced for scaling. The stiffening due to the self weight of the tower is: obtained by using $T = \int_x^L -mg\, dx'$ leading to: $K_{\mathrm{SW}}^T = \int_0^L \nu_0^2 \int_x^L -mg\, dx'\, dx = Mg\left(-\frac{\pi^2}{8L} + \frac{4+\pi^2}{16L}\right)$ and thus $\frac{K_{\mathrm{SW}}^T}{K_q} = -\frac{Mg}{P_{cr}}\left(\frac{1}{2} - \frac{2}{\pi^2}\right)$.

Numerical Rayleigh-Ritz approximation The numerical implementation of the method is tested by using as shape functions the bending modes of a beam with a top-mass:

$$u_z^j = S_j\left[\cosh(\beta_j\tilde{x}) - \cos(\beta_j\tilde{x})\right] - C_j\left[\sinh(\beta_j\tilde{x}) - \sin(\beta_j\tilde{x})\right], \quad \text{with} \quad S_j = \sinh\beta_j + \sin\beta_j, \quad C_j = \cosh\beta_j + \cos\beta_j \tag{32}$$

where β_j is the solution of $1+\cosh\beta_j\cos\beta_j - \beta_j M_t/M(\sin\beta_j\cosh\beta_j - \cos\beta_j\sinh\beta_j) = 0$ the closest to $(2j-1)\pi/2$. The angular frequency of the bending mode is $\omega_j = (\beta_j/L)^2\sqrt{EI_y/(\rho A)}$. The matrices $\mathbf{M}_q$, $\mathbf{K}_q$ and $\mathbf{K}^T$ are determined numerically using the modes from Equation 32 as shape functions.

Results The importance of the stiffness correction is measured by evaluating the relative difference between the eigenfrequency with and without correction, respectively noted f_c and f_{nc}. For a single DOF system these frequencies are:

$$f_{nc} = \frac{1}{2\pi}\sqrt{\frac{K_q}{M_q}}, \qquad f_c = \frac{1}{2\pi}\sqrt{\frac{K_q + K^T}{M_q}}, \qquad \frac{f_c - f_{nc}}{f_{nc}} = \sqrt{1 + \frac{K^T}{K_q}} - 1 \approx -\frac{1}{2}\frac{K^T}{K_q} \tag{33}$$

Results from the two Rayleigh-Ritz (RR) approximations of the two previous paragraphs are shown in Figure 3. They are compared to results from a beam FEM code which accounts for geometrical non-linearities. The simple theory is not expected to represent well large self-weight and large top-mass since the shape function used is approximative. The results would yet be improved if more shape functions were used. The use of a correct mode shape provides results in agreement with the heavier FEM implementation. The FEM implementation uses 200 DOF compared to the Rayleigh-Ritz method which uses here only 1 DOF and 50 radial positions for the numerical integration.

For a typical wind turbine, the effect of the self-weight and top-mass will account for a frequency shift of about -1% and will affect the dynamics of the turbine due to the lower stiffness. Only the effect of the top-mass was investigated here, but it is important to note that other effects need to be accounted for in a wind turbine simulation. The effect of the rotational inertia of the rotor-nacelle assembly has a great impact on the frequencies of the structure - in particular on the second mode. This impact would be directly captured in the current formalism by building the matrices for the full system as done for instance in section 6. Also, in a time series calculation, not-only the top-mass but the entire axial loading from inertial and external loads need to be accounted for by means of the generalized force correction presented in subsection 4.2.

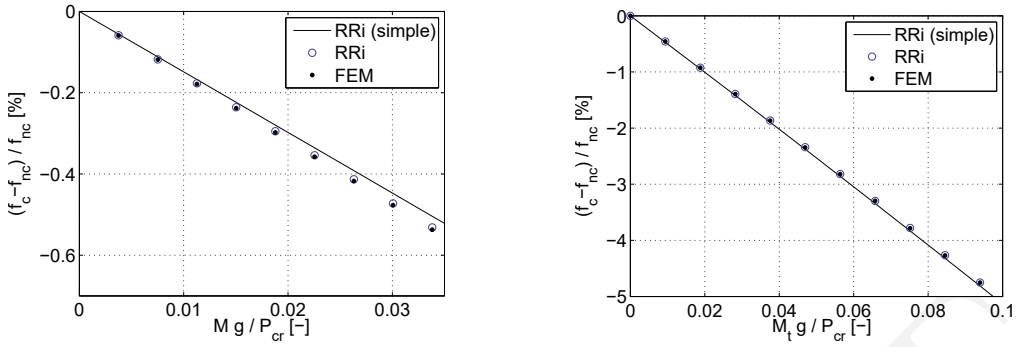

Figure 3: Effect of self-weight (left) and top-mass (right) corrections on the first eigenfrequency of a uniform vertical clamped-free beam. The corrected frequency f_c is compared to the non-corrected frequency f_{nc}. The effect is computed using the simple analytical Rayleigh-Ritz results presented in this paragraph (RRi simple), a numerical Rayleigh-Ritz method (RRi) and a Finite Element method (FEM) based on beam elements. The correction is of prime importance when the overall mass of the beam M or the top-mass M_{top} increases.

5.2 Centrifugal stiffening (blade)

The stiffness correction presented in subsection 4.2 was validated using static analyses in subsection 5.1. A dynamic analysis is here presented to further validate the current implementation. The stiffness correction is relevant for blades where the main axial force is due to centrifugal loads. The numerical experiment of Kane [17] is used as a test case. Kane's driver experiment consists in rotating a beam at its clamped extremity with the angular acceleration $\ddot{\theta} = \frac{2}{5}(1 - \cos(\pi t/7.5))$ from $t = 0$ to $t = 15$s and no acceleration afterwards. The beam properties are $L = 10$m, $EI = 14\text{kNm}^2$, $m = 1.2\text{kg}m^{-1}$. The shape functions of Equation 32 are used for the Rayleigh-Ritz approximation of the elastic deformation of the beam, with $M_t = 0$. Analytical mode shapes of a thin rotating beam can also be used, but little impact on the results was found for this numerical experiment.

The results of the tip-deflection of the beam using three shape functions are shown on the left of Figure 4. The results are validated against a lumped mass numerical code implemented based on the article of Huston and Wang [18]. The lumped-mass method includes geometrical non-linearities. The relative root mean square error between the

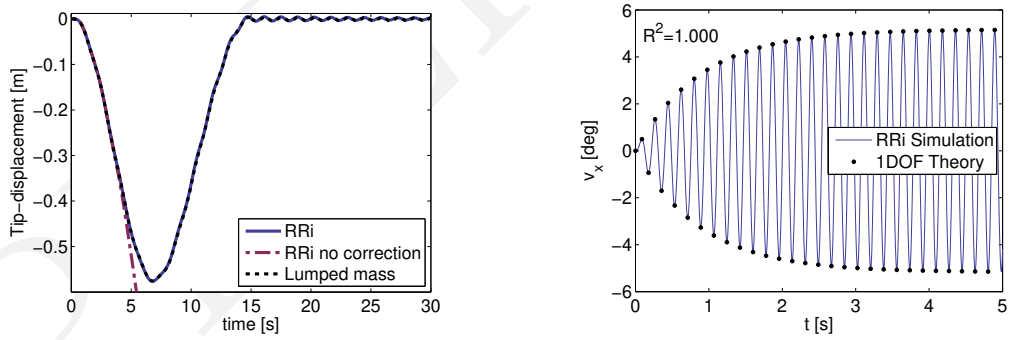

Figure 4: (Left) Tip displacement for the rotational acceleration of a beam (Kane's driver experiment [17]) as computed with a lumped mass and spring algorithm (also found in the work of Huston et al. [18]) and with the current Rayleigh-Ritz (RRi) method. The former includes geometrical non-linearity effects and the latter is here successfully corrected to account for them. The absence of correction lead to a diverging tip motion. (Right) Torsional response of a uniform beam to harmonic oscillations as computed with the RRi method and compared with the analytical solution of a 1DOF system provided in this paragraph.

Rayleigh-Ritz and the lumped mass simulation is 0.5% and 0.3% using 1 or 3 shape functions respectively. The method produces unphysical results when the stiffness correction is not included. The computational time for the current implementation is 5s compared to 15min for the lumped mass method.

5.3 Torsion (blade/tower)

The torsional model is validated using a simple harmonic excitation of a clamped-free beam. A straight uniform beam of hollow tubular section is considered with: $L = 100$m, $G = 79.3$GPa, $\rho = 7850$kg/m^3. The diameter and thickness of the section are $D = 8$m and $t = 45$mm, leading to $I_x = \frac{\pi}{32}(D^4 - (D-2t)^4) = 17.8\text{m}^4$, $K_t = I_x/2$ (hollow disk) and $j_G \triangleq \rho I_x$. The shape functions used for this test case are the pure torsional modes of a uniform beam: $v_x^j = \sin(\beta_j \tilde{x})$ with $\beta_j = \frac{\pi}{2} + (j-1)\pi$, $\omega_j = \frac{c}{L}\beta_j$, $c = \sqrt{\frac{GK_t}{\rho I_x}}$. The generalized mass and stiffness matrix obtained with these shape function are diagonal with coefficients $M_{q,jj} = \int_0^L \rho I_x (\nu_x^j)^2\, dx = \rho I_x L/2$ and $K_{q,jj} = \int_0^L GK_t(\kappa_x^j)^2\, dx = M_{q,jj}\omega_j^2$. The damping is set using stiffness proportional damping with a logarithmic decrement of $\zeta = 1/(10\pi)$. The beam is excited with the moment $\mathcal{M}_x = \mathcal{M}_0 \sin(\Omega t)$ at its free end, with $\mathcal{M}_0 = 50$GNm and $\tilde{\Omega} \triangleq \Omega/\omega_1 = 1.0002$. The numerical results are compared with the transient response of a single DOF with forced harmonic oscillations: $v_x(t) = Ae^{-\zeta\omega_1 t}\sin(\omega_1 t + \psi) + H_0 \sin(\Omega t - \phi)$ with $\psi = \text{atan}\,\frac{\sqrt{1-\zeta^2}}{\zeta - \tilde{\Omega}\frac{\cos\phi}{\sin\phi}}$, $A = H_0\frac{\sin\phi}{\sin\psi}$ where A and ψ were determined from the initial conditions $v_x(0) = \dot{v_x}(0) = 0$ and with: $\phi = \text{atan}\left(\frac{2\zeta\tilde{\Omega}}{1-\tilde{\Omega}^2}\right)$ and $H_0 = \mathcal{M}_0/K_{q,11}/\sqrt{\left(1-\tilde{\Omega}^2\right)^2 + \left(2\zeta\tilde{\Omega}\right)^2}$. The comparison is shown on the right of Figure 4 where 4 shape functions were used. A perfect fit is found validating the numerical implementation of the torsional model for this simple case.

5.4 Rotation of a pre-bended beam

Kane's driver experiment presented in subsection 5.2 is here used with a pre-bended beam in order to induce a torsional motion during the acceleration of the beam. The beam is bended in the "flap-wise" direction, i.e. along the axis of rotation z, according to the function: $z_0 = -0.5\left[1 - \cos(\frac{x}{L}\frac{\pi}{2})\right]$. The material properties are the same as subsection 5.2 but supplemented with the following quantities for the torsion: $j_G = 2$kgm and $GK_t = 200\text{Nm}^2$. The simulation is performed using the current tool and the module *BeamDyn* of *OpenFAST* [19], a Legendre-spectral-FE implementation of geometrically exact beam theory. The tip-displacement results obtained using both codes are presented in Figure 5. The edge-wise tip-displacement follows a similar shape as the non-bended case presented in

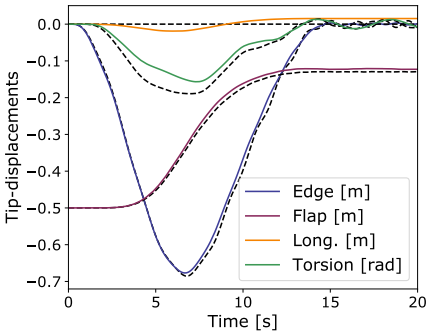

Figure 5: Rotational acceleration of a pre-bended beam as computed with *BeamDyn* (plain lines) and the current Rayleigh-Ritz approach (dashed lines). The rotation of the beam induces displacements in the edge-wise, flap-wise, longitudinal and torsional directions. The edgewise and torsional motions dampens out after the initial 15s of acceleration.

Figure 4. The centrifugal force brings the beam closer to the plane of rotation, from its initial pre-bended position of -0.5m to about -0.13m. *BeamDyn* models the elongation of the beam via the longitudinal stiffness taken here as $EA = 2e7$N. This elongation is not taken into account in the current Rayleigh-Ritz model. The acceleration of the beam induces a torsional motion of the beam, negative about the axis of the beam directed from root to tip. The current model is in fair agreement with the more advanced FE-model. The torsional motion appears to be over-predicted by the method, revealing that additional correction terms may be required to increase the fidelity of the model.

5.5 Mixed FEM-Rayleigh-Ritz formulation

The beam of subsection 5.1 is split into two bodies to validate the mixed FEM-Rayleigh-Ritz formulation. The notations from subsection 3.4 are adopted. A 2D formulation is used and the nodal DOF at point B are written $\mathbf{q}^a = [u_z,\ \theta_y]$ respectively for the translation and rotation. This leads to $\boldsymbol{v}^b\big|_a = \dot{u}_z \boldsymbol{e}_z^a$, and $\boldsymbol{\omega}^b\big|_a = -\dot{\theta}_y \boldsymbol{e}_y^a$ and thus

$$\mathbf{B}_{ab} = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^t, \qquad \mathbf{B}'_{ab} \triangleq \mathbf{R}_{ba} \cdot \mathbf{B}_{ab} = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 & 0 & 0 \end{bmatrix}^t \tag{34}$$

Equation 19 is then used at each time step to obtain $\mathbf{M}_q$, $\mathbf{K}_q$ and $\mathbf{f}_q$. The procedure is first validated by comparing the obtained mode shapes and frequencies with the analytical solution from Equation 32. Result from 7 configurations are shown in Table 1. The case RR^4 models the full beam using 4 shape-functions. The other cases use a mixed

Table 1: Relative errors δ and mean relative errors $\overline{\delta}$ in % obtained for different mixed FEM-Rayleigh-Ritz models of a beam. Frequencies and shapes are compared to the analytical solution. The displacements are compared to a reference simulation. The simulation times are relative to the longest simulation.

Case	**RR^4**	**FEM^{50}-RR^2**	**FEM^{50}-RR^4**	**GY^2-RR^2**	**GY^2-RR^4**	**GY^2-RR^{4}c**	**CB^{2+2}-RR^4**
Mode frequency $\delta\mathbf{f_2}$	0.00	0.02	-0.01	0.71	0.68	0.14	-0.01
Mode frequency $\delta\mathbf{f_4}$	0.00	1.18	-0.03	50.08	35.28	12.06	0.11
Mode shape $\overline{\delta\mathbf{u_2}}$	0.00	0.18	0.01	0.29	0.32	0.07	0.01
Mode shape $\overline{\delta\mathbf{u_4}}$	0.00	4.80	0.22	15.89	18.48	9.47	0.23
Top-displacement $\overline{\delta\mathbf{d}}$	1.30	1.29	1.22	1.30	1.25	1.23	1.22
Simulation time δt	6.00	96.20	100.00	8.70	9.40	9.30	9.20

formulation where the first part of the beam is discretized using 50 FEM-beam-elements and the second part using 2 or 4 shape functions discretized with 50 spanwise sections. In the 4 last cases, the FEM model is reduced with the Guyan (GY) and Craig-Bampton (CB) techniques. Both reductions use the nodal DOF at point B as master DOF. The case CB^{2+2}-RR^4 uses 2 CB modes and 2 Guyan modes for the first body and 4 shape functions for the second. The analytical mode shapes are used as shape functions for all the RR cases. The beam is split at its middle for all cases except GY^2-RR^{4}c where the beam is split at 0.4% of its length. All the models are run on the same machine and solved using a Newmark time scheme with a constant time step about 250 time smaller than the period of the first mode of the structure. The highest frequency modeled varies depending on the model and it was chosen to use a consistent time step between the different models. The first mode is well captured by all models and is thus not studied here. The $2nd$ and $4th$ modal frequencies and mode shapes are compared to the analytical values in Table 1. The relative error in the mode shape is computed with respect to the maximum deflection and is only computed along the second body to avoid penalizing the GY method. The results show that: a sufficient number of RR shape functions are needed to capture a given number of mode shapes of the full structure; the Guyan method cannot capture modes higher than 2 accurately; the results depend on the location where the two beams are split; the CB reduction offers a great compromise between total number of DOF and accuracy. Theses results are expected since the accuracy of the RR method depends on the number of shape functions used but also on the quality of the shape function used.

Unsteady simulations similar to the ones performed in subsection 5.3 are also run using an harmonic force exciting the second mode of the structure. The results are compared to a reference simulation performed with 8 RR shape functions. In all models presented in Table 1, the mean relative error compared to that simulation is less than 2%, validating the unsteady implementation of the method. The simulation times from these unsteady simulations show that great gains in computational time may be obtained by using the reduction techniques.

The original *Flex* implementation uses a Guyan representations for the foundation which is not suitable when higher modes are to be captured or complex structures such as jackets need to be modelled. The mixed FEM-RR or CB-RR implementation are then preferred.

5.6 Computational time

Computational efficiency is one of the key motivation for the use of a reduced set of coordinates. To quantify the efficiency gain, a 10 min startup simulation of the NREL 5MW onshore turbine [20] is performed and compared with

the tools *FAST7*, *OpenFAST2* [21] and *HAWC2* [1]. No controller, nor turbulence is used in the simulation. The number of calculation points for the tower and the blades are 11 and 49 respectively for all the tools, and the time step set to 0.01s. *HAWC2* uses a finite element representation based on Timoshenko beam elements, with a Newmark solver, converging under less than 2 iterations for most time steps of this simulation. *HAWC2* uses 5 elements for the shaft. The *FAST* family tools use an Adams-Bashforth-Moulton integration scheme, and the *ElastoDyn* module is here used since its formulation is similar to the one presented in this study. Compared to *FAST7*, *OpenFAST2* consists of a highly versatile modular framework. The current tool use a 4th order Runge-Kutta scheme, and consists of 15 DOF, chosen such as to match the ones of *ElastoDyn*. The average simulations time over 10 runs were obtained as: 25s for *FAST7* (single precision), 56s for the current tool (double precision), 74s for *OpenFAST2* (single precision), 171s for *HAWC2* (double precision), 188s for *OpenFAST2* (double precision). For the case investigated here, a three-fold factor is observed between the simulation times of the current tool and *HAWC2*. A more through investigation would be required to draw further trends. The computational time is not the only metric of relevance, the modularity and accuracy of the tool are also crucial. These topics will be discussed in section 7.

6 Application to a simple wind turbine model

Model description A simple wind turbine model of 4 DOF is derived in this section to illustrate the method. A sketch is given in Figure 6. The model consists of 5 bodies: tower, nacelle, shaft, blade$_1$ and blade$_2$. Each body is labelled using their first letter as a superscript. The origin of the bodies are the points T, N, S and B. The ground origin is at point T, with the x_t axis pointing upwards and the z_t axis pointing downstream. The tower and blades are modelled as flexible beams. One shape function is used for each of them, representing the fore-aft motion of the tower, u_z^t, associated with the flexible DOF q_1 and the flapping motion of the blades $u_z^{b_1} \equiv u_z^{b_2}$ associated with q_3 and q_4. The nacelle and shaft are rigid bodies of given mass and inertias. The nacelle is rigidly attached to the tower-top at point N. The nacelle and shaft bodies are connected at point S by a revolute-joint, which rotation is represented by the DOF q_2. The blades are rigidly attached, perpendicular to the shaft at point B, and evenly distributed along the azimuth.

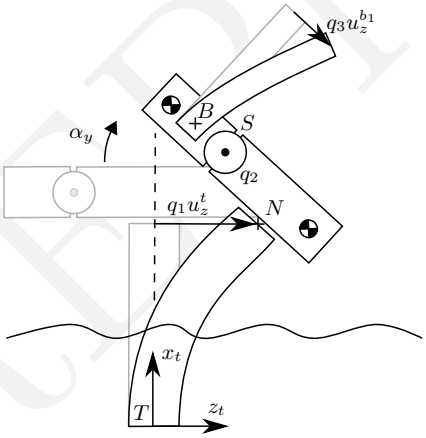

Figure 6: Sketch of the 4DOF system consisting of 5 bodies: tower (flexible), nacelle (rigid), combined shaft and hub (rigid), blade 1 (flexible) and blade 2 (not shown). The light gray sketch represents the undeflected structure when all the DOF q_i are zero. The DOF q_i are respectively associated with: the fore-aft tower motion, the shaft rotation, and the two flap motions of the blades.

Geometry, material properties and shape functions The distances between each points are: $r_{TN} = 100$m, $r_{NS} = 10$m, $r_{SB} = 0$. The nacelle mass, COG and inertia at point N are: $M^n = 400$t, $\boldsymbol{\rho}^n = 2\boldsymbol{e}_z^n$ and $J_N^n = \text{diag}(7, 3, 1)e^6$ kg/m^2. The shaft mass, COG and inertia are: $M^s = 100$t, $\boldsymbol{\rho}^s = \mathbf{0}$, and $J_S^s = \text{diag}(2, 2, 3)e^5$ kg/m^2. The tower and blades are modelled as uniform beams of respective properties: $L^t = 100$m, $EI_y^t = 2e^{12}$N/m^2, $m^t = 9$t/m and $L^b = 75$m, $EI_y^b = 2e^{10}$N/m^2, $m^b = 500$kg/m and $j_G^b = 1e^5$kg/m. Their total mass are $M^t = 900$t and $M^b = 30$t. The shape functions u_z^t, $u_z^{b_1}$ and $u_z^{b_2}$ are taken as the beam mode shapes given in Equation 32. The overhead mass $M_t = M^n + M^s + 2M^b$ is used for the tower.

Body matrices The mass matrix of each individual bodies are obtained using Equation 2, including the correction from Equation 29 for the flexible bodies. The mass matrices of the rigid bodies are readily obtained from M, $\boldsymbol{\rho}$ and $\boldsymbol{J}'$. For validation purposes the mass matrices of the flexible bodies are given below, for a *unit deflection* of the shapes, i.e. $q_1 = 1$ and $q_3 = q_4 = 1$:

$$\boldsymbol{M}'^t = \begin{bmatrix} 9.0e^5 & 0 & 0 & 0 & 3.4e^5 & 0 & 0 \\ & 9.0e^5 & 0 & -3.4e^5 & 0 & 4.5e^7 & 0 \\ & & 9.0e^5 & 0 & -4.5e^7 & 0 & 3.4e^5 \\ & & & 2.2e^5 & 0 & -2.5e^7 & 0 \\ & & & & 3.0e^9 & 0 & -2.5e^7 \\ & & & & & 3.0e^9 & 0 \\ \text{sym} & & & & & & 2.2e^5 \end{bmatrix}, \boldsymbol{M}'^b = \begin{bmatrix} 3.0e^4 & 0 & 0 & 0 & 1.2e^4 & 0 & -2.0e \\ & 3.0e^4 & 0 & -1.2e^4 & 0 & 9.0e^5 & 0 \\ & & 3.0e^4 & 0 & -9.0e^5 & 0 & 1.2e^4 \\ & & & 6.0e^6 & 0 & -5.1e^5 & 0 \\ & & & & 3.6e^7 & 0 & -5.1e \\ & & & & & 3.6e^7 & 0 \\ \text{sym} & & & & & & 7.5e^3 \end{bmatrix}$$

The stiffness matrices are obtained using Equation 20 and Equation 26. All coefficients are zero except for $\boldsymbol{K}'^t_{77} = 6.0e^6$, $\boldsymbol{K}'^t_{g,77} = -1.0e^5$, $\boldsymbol{K}'^b_{77} = 2.9e^5$, $\boldsymbol{K}'^b_{g,77} = -1.9e^4$.

Rotation and velocity transformation matrix The deflection of the tower induces a rotation of the nacelle of angle $\alpha_y = -\sum q_j^t \Phi_{z,j}^t(\boldsymbol{s}_S) = -q_1 u_z^t(L^t) = q_1 C_y$, where C_y is then the coupling coefficient between the angle α_y and the first shape function of the tower. The rotation matrices between the different systems are: $\boldsymbol{R}_{tn} = \boldsymbol{R}_y(\alpha_y)$, $\boldsymbol{R}_{ns} = \boldsymbol{R}_z(q_2)$, $\boldsymbol{R}_{sb_1} = \boldsymbol{R}_z(0)$ and $\boldsymbol{R}_{sb_2} = \boldsymbol{R}_z(\pi)$. The velocity transformation matrices defined in Equation 11 are given below. Explicit formulae are provided but in practice an automated routine is used to derive them as function of the connection points and the nature of the constraints/joints between the bodies. The matrices for the tower and nacelle are $\boldsymbol{B}_x^t = \varnothing$, $\boldsymbol{B}_\theta^t = \varnothing$, $\boldsymbol{I}^t = 1$, $\boldsymbol{B}_x^n = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^t$, $\boldsymbol{B}_\theta^n = \begin{bmatrix} 0 & C_y & 0 \end{bmatrix}^t$, $\boldsymbol{I}^n = 1$. For the shaft and blades:

$$\boldsymbol{B}_x^s = \boldsymbol{B}_x^{b\bullet} = \begin{bmatrix} C_y z_{NS} & 0 & 1 - C_y x_{NS} \\ 0 & 0 & 0 \end{bmatrix}^t, \quad \boldsymbol{B}_\theta^s = \boldsymbol{B}_\theta^{b\bullet} = \begin{bmatrix} 0 & C_y & 0 \\ \sin\alpha_y & 0 & \cos\alpha_y \end{bmatrix}^t, \quad \boldsymbol{I}^s = i\varnothing, \quad \boldsymbol{I}^{b_1} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \boldsymbol{I}^{b_2} = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{35}$$

The $\boldsymbol{B}_x$ and $\boldsymbol{B}_\theta$ matrices were expressed in the reference frame for convenience, but in practice they are expressed in each individual body frames. The mass matrix is obtained according to Equation 8 as $\boldsymbol{M}_q = \sum_{i=1}^b \mathbf{B}'^{it} \mathbf{M}'^i \mathbf{B}'^i$ and the stiffness matrix $\mathbf{K}_q = \sum_{i=1}^b \mathbf{B}'^{it} \mathbf{M}'^i \mathbf{B}'^i$. For validation purposes, these matrices are given below for $\boldsymbol{q} = [1\ 1\ 1\ 1]$:

$$\boldsymbol{M}_q = \begin{bmatrix} 7.86e^5 & 0 & 7.71e^3 & 1.58e^4 \\ & 7.23e^7 & 0 & 0 \\ & & 7.50e^3 & 0 \\ \text{sym} & & 0 & 7.50e^3 \end{bmatrix}, \quad \mathbf{K}_q = \begin{bmatrix} 6.01e^6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ & & 2.86e^5 & 0 \\ \text{sym} & & 0 & 2.86e^6 \end{bmatrix} \tag{36}$$

Stiffness proportional damping or Rayleigh-Damping based on the initial mass matrix may be used for $\mathbf{D}_q$. The external and quadratic velocity forces on each bodies expressed in their own system are also converted to generalized forces using the $\mathbf{B}'$ matrices. The body forces are transferred by the joints from the child to the parents. The equation of motions Equation 10 are then fully defined at a given time. The scripts to reproduce these results are provided on the GitHub repository of the author [22].

7 Discussions

The principles behind the formulation presented in this article may be attributed to the work of Kane [14] and Nikravesh [5] nearly six decades ago. Wind turbine aeroelastic codes which rely on these principles started to be developed as early as the beginning of the 1980's. The most common tools in the wind energy community which use this formulation are *FAST* (now *OpenFAST*) and *Flex*, but different institutions have implemented similar codes. Despite the long legacy behind these tools, the documentation is scarce. It is hoped that the current article bridges some of these knowledge gaps and may help the development of future implementations. This article focused on the systematic development of the equations using the formalism of the velocity transformation matrix. Such formalism may ease some of the understanding and implementation, but it can also be replaced by using alternative formulations such as Kane's partial velocity method, or the application of the virtual work principle.

The Rayleigh-Ritz approximation with joint coordinates is an elegant and effective solution that focuses on the key physical behaviors of a system. One drawback is that it requires expertise on the system to carefully select the shape functions and degrees of freedom, but a direct advantage is that it results in a model with significantly less number of degrees of freedom and with computational times several times faster than conventional tools. Yet, as

the computational power increases and the engineering work pushes the systems to higher complexity, the need for accuracy over computational time becomes essential. The current method presents several limitations on this aspect.

The thin-beam and small-deflection approximations used in section 4 and 5 of the current article are crude assumptions. They appear reasonable for the modelling of the support structure, though the relatively small thickness of modern monopiles make the beam-approximation questionable. On the other hand, these assumptions are greatly challenged for the simulation of modern rotors with highly-flexible blades, both in bending and torsion.

The assumption that the beams do not deflect axially can easily be lifted, as done in *OpenFAST* for instance. The account for large deflections and geometrical non-linearity is less trivial. The re-computation of the system matrices at each time step using the deflected geometry of the structure may account for an important part of the non-linearity, but the accuracy of the method is yet to be investigated and proven. Alternatively, the small-deflection approximation may be lifted by dividing the bodies into sub-bodies. Unfortunately the complexity involved may not justify the use of such formulation over a beam finite-element approach.

An important part of the accuracy of the method relies on the determination of the shape functions. The original version of *Flex* used a simple iterative procedure suitable for straight beams with uncoupled flapwise and edgewise shapes. Pre-bend of the beam could then be modelled as elastic equilibrium points by projecting the pre-bend shapes and expressing them as a linear combinations of the shape functions. The original tool is easily extended by using the general formalism presented in section 1. The shape functions are determined using a finite element formulation that accounts for the true mean-line of the body and make use of the so called "6 by 6" stiffness matrices of each cross section of the body. For further refinement, the shape functions should be determined for a given operating point of the turbine, to better fit the behavior of the structure at a given rotational speed and about a given deflected state.

The recursive methodology presented in section 3 only applies for an open-loop system. Unfortunately, the modelling of advanced support structures, detailed nacelle frame and drive-train components requires the introduction of kinematic closed loop. Methods to address these issues exists [23] and the specific treatments for wind energy tools such as *Flex* may be addressed in future work.

8 Conclusion

The framework presented allows for the systematic development of structural models for open-loop systems with a minimal set of degrees of freedom. A limited set of shape functions is required for systems that are mainly excited below a given frequency. In that case, the models consist of a small number of degrees of freedom and numerical implementations will result in reduced computational times compared to conventional approaches.

Wind turbine models such as the one used in the aero-elastic code *Flex* are readily implemented. The original *Flex* model may be improved using the developments discussed in this article, e.g. by: adding more shape functions, introducing torsion, and coupling with a FEM model. A simple analytical model of a wind turbine was derived and may be used for fundamental studies or fast performance estimates.

Future work will focus on further developing and validating the method and investigate the limitations of the method regarding its linear approach and its potential shortcomings in modelling large deformations.

A Shape integrals

Shape integral functions are introduced to compute the constant terms that appears in the mass matrix. They are defined as:

$$\boldsymbol{\sigma} = \int \boldsymbol{s}'_{P_0}\,\mathrm{d}m, \quad \sigma_{kl} = \int s_{0k}'\, s_{0l}'\,\mathrm{d}m, \quad \boldsymbol{\Sigma}_{kl} = \int s_{0k}'\,\boldsymbol{\Phi}_l'\,\mathrm{d}m, \quad \boldsymbol{\Psi} = \int \boldsymbol{\Phi}'\,\mathrm{d}m, \quad \boldsymbol{\Upsilon}_{kl} = \int \boldsymbol{\Phi}_k'^{\,t}\boldsymbol{\Phi}_l'\,\mathrm{d}m, \quad k,l = 1,2,3 \tag{37}$$

where s_{0k}' are the coordinates of $\boldsymbol{s}'_{P_0}$. The nine vectors $\boldsymbol{\Sigma}_{kl}$ have dimension $1 \times n_f$, the matrix $\boldsymbol{\Psi}$ is $3 \times n_f$ and the nine matrices $\boldsymbol{\Upsilon}_{kl}$ are $n_f \times n_f$, with $\boldsymbol{\Upsilon}_{kl} = \boldsymbol{\Upsilon}_{lk}^t$. The elements of the mass matrix given in Equation 2 are expressed as function of the shape integrals as follows [7]: $\boldsymbol{M}'_{x\theta} = -\tilde{\boldsymbol{\sigma}} - \mathrm{Skew}\left(\boldsymbol{\Psi}\boldsymbol{g}\right)$, $\boldsymbol{M}'_{xg} = \boldsymbol{\Psi}$, $\boldsymbol{M}'_{gg} = \boldsymbol{\Upsilon}_{11} + \boldsymbol{\Upsilon}_{22} + \boldsymbol{\Upsilon}_{33}$.

$$\boldsymbol{M}'_{\theta\theta} = \begin{bmatrix} u_{22}+u_{33} & -u_{12} & -u_{13} \\ -u_{12} & u_{11}+u_{33} & -u_{23} \\ -u_{13} & -u_{23} & u_{11}+u_{22} \end{bmatrix}, \qquad \boldsymbol{M}'_{\theta g} = \begin{bmatrix} \boldsymbol{g}^t(\boldsymbol{\Upsilon}_{23}-\boldsymbol{\Upsilon}_{32}) + \boldsymbol{\Sigma}_{23} - \boldsymbol{\Sigma}_{32} \\ \boldsymbol{g}^t(\boldsymbol{\Upsilon}_{31}-\boldsymbol{\Upsilon}_{13}) + \boldsymbol{\Sigma}_{31} - \boldsymbol{\Sigma}_{13} \\ \boldsymbol{g}^t(\boldsymbol{\Upsilon}_{12}-\boldsymbol{\Upsilon}_{21}) + \boldsymbol{\Sigma}_{12} - \boldsymbol{\Sigma}_{21} \end{bmatrix} \tag{38}$$

where $u_{kl} = \sigma_{kl} + (\boldsymbol{\Sigma}_{kl} + \boldsymbol{\Sigma}_{lk})\boldsymbol{g} + \boldsymbol{g}^t\boldsymbol{\Upsilon}_{lk}\boldsymbol{g}$. The computation of the shape integrals is done in a preprocessing step which only requires the initial geometry of the structure and the shape functions.

References

[1] T.J. Larsen and A.M. Hansen. *HAWC2 - User manual*. DTU-Risø-R-1597, 2007.

[2] S. Øye. *Fix Dynamisk, aeroelastisk beregning af vindmøllevinger*. Report AFM83-08, Fluid Mechanics, DTU, 1983.

[3] J. M. Jonkman and M. L. Buhl. Fast user's guide. Technical Report NREL/EL-500-38230, National Renewable Energy Laboratory, 2005.

[4] OpenFAST. Available at `http://github.com/OpenFAST/OpenFAST/`.

[5] Parviz E. Nikravesh. Systematic reduction of multibody equations of motion to a minimal set. *International Journal of Non-Linear Mechanics*, 25(2):143 – 151, 1990.

[6] W. J. Book. Recursive lagrangian dynamics of flexible manipulator arms. *The International Journal of Robotics Research*, 3(3):87–101, 1984.

[7] A.A. Shabana. *Dynamics of Multibody Systems*. Dynamics of Multibody Systems. Cambridge University Press, 2013.

[8] S. Øye. Flex4 metoder, 1990.

[9] T. H.H. Pian, R. L. Bisplinghoff, and J. W. Mar. *Statics of Deformable Solids*. Dover, 1965.

[10] R.L. Bisplinghoff and H. Ashley. *Principles of Aeroelasticity*. Dover Books on Engineering Series. Dover Publications, 1962.

[11] P.E. Nikravesh. *Computer-aided analysis of mechanical systems*. Prentice-Hall, 1988.

[12] R.L. Huston. *Multibody Dynamics*. Butterworth-Heinemann, 1990.

[13] B. Siciliano and O. Khatib. *Springer Handbook of Robotics*. Springer Handbooks. Springer International Publishing, 2016.

[14] T. R. Kane and C. F. Wang. On the derivation of equations of motion. *Journal of the Society for Industrial and Applied Mathematics*, 13(2):487–492, 1965.

[15] F. Amirouche. Fundamentals of multibody dynamics: Theory and applications. *Fundamentals of Multibody Dynamics: Theory and Applications*, pages 1–684, 2006.

[16] M.O.L. Hansen. *Aerodynamics of Wind Turbines - Second Edition*. Earthscan, 2008.

[17] T. R. Kane, R. Ryan, and A. K. Banerjee. Dynamics of a cantilever beam attached to a moving base. *Journal of Guidance, Control, and Dynamics*, 10(2):139–151, 1987.

[18] R. L. Huston and Y. Wang. Flexibility effects in multibody systems. *Computer-Aided Analysis of Rigid and Flexible Mechanical Systems*, pages 351–376, 1994.

[19] Qi Wang, Nick Johnson, Michael A. Sprague, and Jason Jonkman. Beamdyn: A high-fidelity wind turbine blade solver in the fast modular framework. *33rd Wind Energy Symposium*, pages 17 pp., 17 pp., 2015.

[20] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5mw reference wind turbine for offshore system development. Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, 2009.

[21] J. Jonkman. The new modularization framework for the fast wind turbine cae tool. *51st Aiaa Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition 2013*, 2013.

[22] E. Branlard. Yams github repository `http://github.com/ebranlard/YAMS/`.

[23] Gim G. Nikravesh PE. Systematic construction of the equations of motion for multibody systems containing closed kinematic loops. *ASME. J. Mech. Des.*, 115(1):143–149, 1993.