Computers and Structures 213 (2019) 82–99

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

Design optimization of dynamic flexible multibody systems using the discrete adjoint variable method

Mehran Ebrahimi *, Adrian Butscher, Hyunmin Cheong, Francesco Iorio

Autodesk Research, 661 University Avenue, Toronto, ON M5G 1M1, Canada

ARTICLE INFO

Article history:
Received 18 July 2018
Accepted 13 December 2018
Available online 21 December 2018

Keywords:
Flexible multibody systems
Design optimization
Sensitivity analysis
Adjoint variable method
Discrete adjoint variable method
Geometrical design variables

ABSTRACT

The design space of dynamic multibody systems (MBSs), particularly those with flexible components, is considerably large. Consequently, having a means to efficiently explore this space and find the optimum solution within a feasible time-frame is crucial. It is well-known that for problems with several design variables, sensitivity analysis using the adjoint variable method extensively reduces the computational costs. This paper presents the novel extension of the discrete adjoint variable method to the design optimization of dynamic flexible MBSs. The extension involves deriving the adjoint equations directly from the discrete, rather than the continuous, equations of motion. This results in a system of algebraic equations that is computationally less demanding to solve compared to the system of differential algebraic equations produced by the continuous adjoint variable method. To describe the proposed method, it is integrated with a numerical time-stepping algorithm based on geometric variational integrators. The developed technique is then applied to the optimization of MBSs composed of springs, dampers, beams and rigid bodies, considering both geometrical (e.g., positions of joints) and non-geometrical (e.g., mechanical properties of components) design variables. To validate the developed methods and show their applicability, three numerical examples are provided.

1. Introduction

Multibody systems are mechanical assemblies composed of several interconnected parts and have numerous applications in automotive, aerospace, robotics and many other industries. Generally, these systems have a large number of degrees of freedom, show dynamic behaviors, and their components may undergo both large overall motion and large deformation, thus making the optimal design of such systems a challenging task.

The optimization of an MBS involves finding the optimum values of the system's parameters, also called *design variables*, that minimize (or maximize) a desired objective function and satisfy a set of constraints. Design variables, in general, could be classified into control parameters and design parameters. The former are the focus of optimal control problems and their examples include time-dependent input forces and torques. Design parameters, on the other hand, are handled in design optimization problems and are related to mechanical and physical properties of a system and its components (e.g., material properties, cross-sectional areas, lengths). In both optimization scenarios, for an assembly with many coupled flexible and rigid bodies, the parametric design space is considerably vast. Therefore, an efficient algorithm is required to search this space and find an optimum design within a reasonable timescale.

Accordingly, gradient-based optimizers have been shown to outperform meta-heuristic optimization techniques and have a better convergence rate (see e.g. [1]). In order to utilize gradient-based approaches, sensitivity analysis should be performed, which requires computing the derivative of the objective and constraint functions with respect to the design variables. In the majority of problems, these functions depend not only explicitly on the design variables, but also implicitly on them through the *state variables*, which describe the state of an MBS. Hence, to obtain their total derivatives, one needs to calculate the sensitivity values of the state variables to the design variables as well. This becomes a laborious task, particularly in large-scale and time-dependent problems.

One way of computing the gradients is to calculate them numerically using the finite difference method. Although this approach is easy to implement, it provides only an approximation to the actual gradient values, and the perturbation step of each design variable is not known *a priori*. Additionally, to obtain the sensitivities, one additional simulation per design variable is required. Consequently, for computationally expensive problems,

* Corresponding author.
E-mail address: mehran.ebrahimi@autodesk.com (M. Ebrahimi).

https://doi.org/10.1016/j.compstruc.2018.12.007
0045-7949/© 2018 Elsevier Ltd. All rights reserved.

this method is highly inefficient and prohibitively time-consuming [2,3]. A more practical and accurate approach is to apply analytical techniques that compute the exact gradients.

The two most widely used methods in the analytical category are the *direct differentiation method* and the *adjoint variable method*. In the former, first, the motion equations of a given MBS are differentiated with respect to each design variable. This results in a system of differential-algebraic equations (DAEs), in which the gradients of the state variables and the Lagrange multipliers are unknowns. Then, by solving this system and implementing the solution values into the sensitivity equations of the objective and constraint functions, the desired gradients are evaluated [3–9]. Since for each design variable a new set of DAEs needs to be solved, the computational cost of this approach becomes significant for complex assemblies with a large number of design variables.

An alternative strategy is to exploit the adjoint variable method [8,10–14]. In this technique, the explicit computation of the gradients of state variables and Lagrange multipliers, as required in the direct differentiation method, is avoided by introducing a set of auxiliary variables called *adjoint variables*. To compute the adjoint variables, the equations of motion are first solved (*forward simulation*), followed by integrating a system of linear DAEs known as adjoint equations backward in time (*backward simulation*). The main advantage of the adjoint variable method is that the generated linear system needs to be solved only once at each iteration of the optimization process, while eliminating entirely the need for computing the derivatives of state variables and Lagrange multipliers with respect to the design variables.

In one view, the adjoint variable method can further be split into two categories: the continuous adjoint variable method (CAVM) and the discrete adjoint variable method (DAVM). In the former, the continuous equations of motion (i.e., the equations of motion prior to discretization) are used to derive the continuous adjoint equations — a system of linear DAEs. This system should be then discretized in time and solved to find the adjoint variables. On the other hand, in the DAVM, the discrete adjoint equations are directly derived from the discretized equations of motion. This approach yields the exact gradients of the discrete objective and constraint functions. More importantly, the equations generated by the DAVM no longer need an additional level of discretization in time and form a system of linear algebraic equations, which are computationally easier to solve compared to the DAEs that arise via the CAVM.

The DAVM has been previously used in optimal control and design optimization problems [15,16], however, its application in large-scale MBSs with coupled flexible and rigid components is not yet well studied. This paper introduces the novel application of the DAVM to the design optimization of flexible MBSs. Without loss of generality, to describe the implementation details and solve the equation of motion, the proposed sensitivity analysis scheme is applied to a symplectic-momentum preserving geometric variational integrator proposed by Leyendecker et al. [17]. The same method can be applied to other numerical time-stepping solvers.

In the current study, both *geometrical* and *non-geometrical* design variables are taken into account. To the authors' knowledge, the inclusion of geometrical design variables in optimizing dynamic MBSs has not been investigated before. Geometrical design variables concern the parameters defining the shape of a multibody configuration for example the global initial positions of joints and components' length. Non-geometrical variables include other mechanical and physical properties of the bodies such as spring constants, damping coefficients, masses, cross-sections and Young's moduli. Considering geometrical parameters would increase the dimension of the design space and provides the optimization routine with more flexibility. There are many applications, for instance designing an MBS whose components should follow certain trajectories in space, where incorporating only the non-geometrical parameters would not lead to a solution, while including the geometrical variables is essential to finding the desired design.

To demonstrate how to apply the proposed methods, without loss of generality, the relevant sensitivity equations for assemblies made of springs, dampers, beams and rigid bodies connected via spherical (pin) and welded (fixed) joints are derived. The presented approach can be extended to include other component and joint types. The remainder of the paper is organized as follows. Section 2 introduces the general problem definition and basics of gradient-based optimization algorithms. Section 3 presents the equations of motion and how to solve them using the selected geometrical variational integrator. In Section 4, the detailed derivation of the DAVM is provided. In Section 5, first, the equations involved in describing the dynamics of interconnected springs, dampers, beams and rigid bodies are discussed. Then, the equations for computing the sensitivities with respect to geometrical design variables are developed. Finally, in the last section, three numerical examples are presented.

2. General problem definition

Consider an unconstrained optimization problem for the vector $\vec{\mathbf{a}} \in \mathbb{R}^n$ of *design variables* of a multibody system in the following form:

$$\min_{\vec{\mathbf{a}}} \phi\left(\vec{\mathbf{q}}\left(\vec{\mathbf{a}}\right), \dot{\vec{\mathbf{q}}}\left(\vec{\mathbf{a}}\right), \vec{\lambda}\left(\vec{\mathbf{a}}\right), \vec{\mathbf{a}}\right) \tag{1}$$

In this equation, ϕ is the objective function, $\vec{\mathbf{q}}$ represents the vector of *state variables*, $\dot{\vec{\mathbf{q}}}$ is its *time derivative*, and $\vec{\lambda}$ denotes the vector of *Lagrange multipliers* associated with the joint (constraint) equations. These quantities are time-dependent and are the solutions of the multibody dynamic equations, to be described in Section 3. As the behavior of an MBS is affected by the choices of the design variables, $\vec{\mathbf{q}}, \dot{\vec{\mathbf{q}}}$ and $\vec{\lambda}$ are also dependent on $\vec{\mathbf{a}}$. It is assumed that ϕ is at least a twice-differentiable function of its arguments. Moreover, It is assumed that ϕ has the following generic form

$$\phi = F\left(\vec{\mathbf{q}}_0, \dot{\vec{\mathbf{q}}}_0, \vec{\mathbf{q}}_T, \dot{\vec{\mathbf{q}}}_T, \vec{\mathbf{a}}\right) + \int_0^T H\left(\vec{\mathbf{q}}, \dot{\vec{\mathbf{q}}}, \vec{\lambda}, \vec{\mathbf{a}}\right)dt \tag{2}$$

where F is a function defined on the two ends of the simulation duration, $t = 0$ and $t = T$, and H is described over its entire duration.

A gradient-based algorithm for finding a locally minimizing solution of Eq. (1) proceeds by iterative improvement of the optimization objective function: if $k \geqslant 0$ denotes the iteration counter, then one strives to improve the current values of the design variables $\vec{\mathbf{a}}^k$ by taking an appropriate small step, i.e., $\vec{\mathbf{a}}^{k+1} := \vec{\mathbf{a}}^k + \delta\vec{\mathbf{a}}^k$. To determine $\delta\vec{\mathbf{a}}^k, \phi$ is represented as the following Taylor series near $\vec{\mathbf{a}}^k$

$$\begin{aligned}
&\phi\left(\vec{\mathbf{q}}, \dot{\vec{\mathbf{q}}}, \vec{\lambda}, \vec{\mathbf{a}}^k + \delta\vec{\mathbf{a}}^k\right) = \phi\left(\vec{\mathbf{q}}, \dot{\vec{\mathbf{q}}}, \vec{\lambda}, \vec{\mathbf{a}}^k\right) \\
&\quad + \frac{d}{d\vec{\mathbf{a}}}\phi\left(\vec{\mathbf{q}}, \dot{\vec{\mathbf{q}}}, \vec{\lambda}, \vec{\mathbf{a}}^k\right) \cdot \delta\vec{\mathbf{a}}^k + \mathrm{o}\left(\left\|\delta\vec{\mathbf{a}}^k\right\|^2\right) \\
&\text{with} \lim_{\left\|\delta\vec{\mathbf{a}}^k\right\| \to 0} \frac{\mathrm{o}\left(\left\|\delta\vec{\mathbf{a}}^k\right\|^2\right)}{\left\|\delta\vec{\mathbf{a}}^k\right\|} = 0
\end{aligned} \tag{3}$$

where $d\phi/d\vec{\mathbf{a}}$ is the *sensitivity* of ϕ with respect to the design variables. To improve the value of ϕ to first order, $\delta\vec{\mathbf{a}}^k$ must be along a *descent direction* such that the second term in Eq. (3) is negative. One possible choice for such a direction is

$$\delta\vec{\mathbf{a}}^k = -\epsilon D_{\vec{\mathbf{a}}}\phi\left(\vec{\mathbf{q}},\dot{\vec{\mathbf{q}}},\vec{\lambda},\vec{\mathbf{a}}^k\right) \tag{4}$$

in which $D_{\vec{\mathbf{a}}}\phi = \frac{d\phi}{d\vec{\mathbf{a}}}$ and $\epsilon > 0$ is a small descent step. Putting Eq. (4) in Eq. (3) leads to

$$\phi\left(\vec{\mathbf{q}},\dot{\vec{\mathbf{q}}},\vec{\lambda},\vec{\mathbf{a}}^k+\delta\vec{\mathbf{a}}^k\right) \approx \phi\left(\vec{\mathbf{q}},\dot{\vec{\mathbf{q}}},\vec{\lambda},\vec{\mathbf{a}}^k\right) - \epsilon\left\|D_{\vec{\mathbf{a}}}\phi\left(\vec{\mathbf{q}},\dot{\vec{\mathbf{q}}},\vec{\lambda},\vec{\mathbf{a}}^k\right)\right\|^2 \tag{5}$$

which guarantees the decrease of ϕ at iteration k of the optimization process. This procedure can be easily extended to constrained optimization problems, using for example the Augmented Lagrangian method. Thus, in order to move along the descent direction in Eq. (4), the derivative of ϕ with respect to $\vec{\mathbf{a}}$ is desired. It is expressed as

$$\frac{d\phi}{d\vec{\mathbf{a}}} = \frac{\partial\phi}{\partial\vec{\mathbf{a}}} + \frac{\partial\phi}{\partial\vec{\mathbf{q}}}\frac{\partial\vec{\mathbf{q}}}{\partial\vec{\mathbf{a}}} + \frac{\partial\phi}{\partial\dot{\vec{\mathbf{q}}}}\frac{\partial\dot{\vec{\mathbf{q}}}}{\partial\vec{\mathbf{a}}} + \frac{\partial\phi}{\partial\vec{\lambda}}\frac{\partial\vec{\lambda}}{\partial\vec{\mathbf{a}}} \tag{6}$$

To compute $\frac{d\phi}{d\vec{\mathbf{a}}}$, the values of $\frac{\partial\vec{\mathbf{q}}}{\partial\vec{\mathbf{a}}}, \frac{\partial\dot{\vec{\mathbf{q}}}}{\partial\vec{\mathbf{a}}}$ and $\frac{\partial\vec{\lambda}}{\partial\vec{\mathbf{a}}}$, as well as $\frac{\partial\phi}{\partial\vec{\mathbf{a}}}$, for different types of design variables are required. This is the main focus of the present paper and is addressed in the forthcoming sections. But first, the following section presents the equations of motions for an MBS and the method chosen to solve them.

3. Solving the equations of motion

According to *Hamilton's principle* (a.k.a. the *least action principle*), it is possible to re-formulate Newton's Law of Motion in such a way that the trajectories of a dynamic system themselves satisfy a type of optimization problem. Namely, the trajectory that an MBS takes to move between two positions in space minimizes a quantity known as the *action integral*. Therefore, the motion of an MBS can be characterized by finding the stationary solutions of its action integral. This leads to a useful method of discretizing the equations of motion known as *geometric variational integration*, which is outlined below.

Suppose $\vec{\mathbf{q}} \in \mathbb{R}^m$ is the vector of all degrees of freedom of an MBS and $\dot{\vec{\mathbf{q}}}$ is their time derivative. Also, let $\vec{\lambda} \in \mathbb{R}^l$ denotes the vector of Lagrange multipliers associated with l *holonomic* constraints (joints) between different bodies in the assembly. Assuming a *conservative* system, its continuous action integral for time $t \in [0, T]$ is expressed by

$$S\left(\vec{\mathbf{q}}\right) = \int_0^T \left(L\left(\vec{\mathbf{q}},\dot{\vec{\mathbf{q}}}\right) - \vec{\lambda}\cdot\vec{\mathbf{g}}\left(\vec{\mathbf{q}}\right)\right)dt \tag{7}$$

where $L\left(\vec{\mathbf{q}},\ \dot{\vec{\mathbf{q}}}\right)$ and $\vec{\mathbf{g}}\left(\vec{\mathbf{q}}\right)$ are the Lagrangian and constraint equations, respectively. The Lagrangian function L is defined as the *kinetic energy* minus the *potential energy* of the system

$$L\left(\vec{\mathbf{q}},\dot{\vec{\mathbf{q}}}\right) = T\left(\dot{\vec{\mathbf{q}}}\right) - U\left(\vec{\mathbf{q}}\right) = \frac{1}{2}\dot{\vec{\mathbf{q}}}^{\mathrm{T}}\mathbf{M}\dot{\vec{\mathbf{q}}} - U\left(\vec{\mathbf{q}}\right) \tag{8}$$

in which $\mathbf{M}$ is the positive-definite *mass matrix* of the system, and U is the total potential energy, due to for example gravity and the deformation of springs and flexible bodies. Following Hamilton's principle, two classes of time-stepping algorithms have come to exist: (1) computing the stationary values of the continuous action integral and then discretizing the resultant dynamic equations in time, or (2) first discretizing the continuous action integral in time and then deriving the discrete dynamic equations directly from the discretized action integral [18]. The latter approach forms the basis of geometric variational integration and is pursued in this paper.

3.1. Geometric variational integrators

Geometric variational integrators belong to the class of *symplectic-momentum* preserving techniques. They are applicable to a wide spectrum of dynamic problems and have been proven superior to traditional approaches [19,20]. Traditional time-stepping algorithms (such as explicit Euler, implicit Euler, Runge-Kutta) suffer from numerical instabilities and artificial dissipation, which makes them incapable of capturing the true dynamic behavior of a system, particularly in the long-duration problems [19]. This can have significant consequences on the accuracy of *forward* (solving the dynamic equations) and *backward* (solving the adjoint equations) simulations for the system of interest. In the current work, a specific type of variational integrators proposed by Leyendecker et al. [17] is adopted to solve the equations of motion and further describe the proposed sensitivity computation scheme.

Following the idea of variational integrators, if the time domain $[0, T]$ is split into N intervals as $[t_n, t_n + h_n](n = 0, 1, \ldots, N-1)$, Eq. (7) can be rewritten as

$$\int_0^T \left(L\left(\vec{\mathbf{q}},\dot{\vec{\mathbf{q}}}\right) - \vec{\lambda}\cdot\vec{\mathbf{g}}\left(\vec{\mathbf{q}}\right)\right)dt = \sum_{n=0}^{N-1}\left(\int_{t_n}^{t_{n+1}} \left(L\left(\vec{\mathbf{q}},\dot{\vec{\mathbf{q}}}\right) - \vec{\lambda}\cdot\vec{\mathbf{g}}\left(\vec{\mathbf{q}}\right)\right)dt\right) \tag{9}$$

The following quadrature approximations can be applied to the integral summands:

$$\begin{aligned}\int_{t_n}^{t_{n+1}} \left(L\left(\vec{\mathbf{q}},\dot{\vec{\mathbf{q}}}\right)\right)dt &\approx h_n L\left((1-\alpha)\vec{\mathbf{q}}_n + \alpha\vec{\mathbf{q}}_{n+1}, \frac{\vec{\mathbf{q}}_{n+1}-\vec{\mathbf{q}}_n}{h_n}\right)\\ &= L_d\left(\vec{\mathbf{q}}_n, \vec{\mathbf{q}}_{n+1}\right)\\ \int_{t_n}^{t_{n+1}} \left(\vec{\lambda}\cdot\vec{\mathbf{g}}\left(\vec{\mathbf{q}}\right)\right)dt &\approx \frac{h_n}{2}\left(\vec{\lambda}_n\cdot\vec{\mathbf{g}}\left(\vec{\mathbf{q}}_n\right) + \vec{\lambda}_{n+1}\cdot\vec{\mathbf{g}}\left(\vec{\mathbf{q}}_{n+1}\right)\right)\\ &= \frac{1}{2}\left(\vec{\lambda}_n\cdot\vec{\mathbf{g}}_d\left(\vec{\mathbf{q}}_n\right) + \vec{\lambda}_{n+1}\cdot\vec{\mathbf{g}}_d\left(\vec{\mathbf{q}}_{n+1}\right)\right)\end{aligned} \tag{10}$$

where $\alpha \in [0, 1]$. If $\alpha = 0.5$, the approximations are second-order accurate, otherwise they are of linear accuracy. Higher order of accuracies can be achieved by improving the quadrature rule. Using Eqs. (9) and (10), the discretized action integral is given by

$$S_d\left(\vec{\mathbf{q}}\right) = \sum_{n=0}^{N-1}\left(L_d\left(\vec{\mathbf{q}}_n,\vec{\mathbf{q}}_{n+1}\right) - \frac{1}{2}\left(\vec{\lambda}_n\cdot\vec{\mathbf{g}}_d\left(\vec{\mathbf{q}}_n\right) + \vec{\lambda}_{n+1}\cdot\vec{\mathbf{g}}_d\left(\vec{\mathbf{q}}_{n+1}\right)\right)\right) \tag{11}$$

Taking the variation of this equation, re-indexing it, and setting $\delta\vec{\mathbf{q}}_0 = \delta\vec{\mathbf{q}}_N = 0$, lead to

$$\delta S_d\left(\vec{\mathbf{q}}\right) = \sum_{n=1}^{N-1}\left(\frac{\partial L_d\left(\vec{\mathbf{q}}_n,\vec{\mathbf{q}}_{n+1}\right)}{\partial\vec{\mathbf{q}}_n} + \frac{\partial L_d\left(\vec{\mathbf{q}}_{n-1},\vec{\mathbf{q}}_n\right)}{\partial\vec{\mathbf{q}}_n} - \left[\frac{\partial\vec{\mathbf{g}}_d\left(\vec{\mathbf{q}}_n\right)}{\partial\vec{\mathbf{q}}_n}\right]^{\mathrm{T}}\vec{\lambda}_n\right)\cdot\delta\vec{\mathbf{q}}_n \tag{12}$$

The stationary values of Eq. (12), along with the constraint equations, provide the *discrete Euler-Lagrange* equation

$$\begin{aligned}&\frac{\partial L_d\left(\vec{\mathbf{q}}_n,\vec{\mathbf{q}}_{n+1}\right)}{\partial\vec{\mathbf{q}}_n} + \frac{\partial L_d\left(\vec{\mathbf{q}}_{n-1},\vec{\mathbf{q}}_n\right)}{\partial\vec{\mathbf{q}}_n} - \left[\frac{\partial\vec{\mathbf{g}}_d\left(\vec{\mathbf{q}}_n\right)}{\partial\vec{\mathbf{q}}_n}\right]^{\mathrm{T}}\vec{\lambda}_n = \vec{\mathbf{0}}\\ &\vec{\mathbf{g}}\left(\vec{\mathbf{q}}_{n+1}\right) = \vec{\mathbf{0}}\end{aligned} \tag{13}$$

Eq. (13) is a system of nonlinear algebraic equations that can be solved iteratively, using for example the Newton-Raphson method, to find $\vec{\mathbf{q}}_{n+1}$ and $\vec{\lambda}_n$ for $n = 1, 2, \ldots, N-1$. Note: in the presence of *non-conservative* external forces (e.g. due to dampers or friction), Eq. (13) is modified to the *discrete Lagrange-d'Alembert* equation [19]

$$\begin{aligned}&\frac{\partial L_d\left(\vec{\mathbf{q}}_n,\vec{\mathbf{q}}_{n+1}\right)}{\partial\vec{\mathbf{q}}_n}+\frac{\partial L_d\left(\vec{\mathbf{q}}_{n-1},\vec{\mathbf{q}}_n\right)}{\partial\vec{\mathbf{q}}_n}-\left[\frac{\partial\vec{\mathbf{g}}_d\left(\vec{\mathbf{q}}_n\right)}{\partial\vec{\mathbf{q}}_n}\right]^{\mathrm{T}}\vec{\lambda}_n\\&\quad+\vec{\mathbf{f}}_d^{-}\left(\vec{\mathbf{q}}_n,\vec{\mathbf{q}}_{n+1}\right)+\vec{\mathbf{f}}_d^{+}\left(\vec{\mathbf{q}}_{n-1},\vec{\mathbf{q}}_n\right)=\vec{\mathbf{0}}\\&\vec{\mathbf{g}}\left(\vec{\mathbf{q}}_{n+1}\right)=\vec{\mathbf{0}}\end{aligned}\tag{14}$$

In order to initiate solving Eqs. (13) and (14), both $\vec{\mathbf{q}}_0$ and $\vec{\mathbf{q}}_1$ are required. In most applications, however, only initial conditions ($\vec{\mathbf{q}}_0$ and $\dot{\vec{\mathbf{q}}}_0$) are provided. To form an equation for $\vec{\mathbf{q}}_1$, the discrete version of the *Legendre transform* can be used. The Legendre transform is a way to switch between the Lagrangian formulation of a dynamic system, which is in the $\left(\vec{\mathbf{q}},\ \dot{\vec{\mathbf{q}}}\right)$ space, to its Hamiltonian formulation in the $\left(\vec{\mathbf{q}},\vec{\mathbf{p}}\right)$ space, where $\vec{\mathbf{p}}$ is the system's momentum. If $\vec{\mathbf{p}}^{-}_{n,n+1}$ and $\vec{\mathbf{p}}^{+}_{n-1,n}$ are the so-called *pre-* and *post-momenta* at time t_n, for a *constrained dissipative* system, the *discrete Legendre transform* reads as [17]

$$\begin{aligned}\vec{\mathbf{p}}^{-}_{n,n+1}&=-\frac{\partial L_d\left(\vec{\mathbf{q}}_n,\vec{\mathbf{q}}_{n+1}\right)}{\partial\vec{\mathbf{q}}_n}+\frac{1}{2}\left[\frac{\partial\vec{\mathbf{g}}_d\left(\vec{\mathbf{q}}_n\right)}{\partial\vec{\mathbf{q}}_n}\right]^{\mathrm{T}}\vec{\lambda}_n-\vec{\mathbf{f}}_d^{-}\left(\vec{\mathbf{q}}_n,\vec{\mathbf{q}}_{n+1}\right),\\\vec{\mathbf{p}}^{+}_{n-1,n}&=\frac{\partial L_d\left(\vec{\mathbf{q}}_{n-1},\vec{\mathbf{q}}_n\right)}{\partial\vec{\mathbf{q}}_n}-\frac{1}{2}\left[\frac{\partial\vec{\mathbf{g}}_d\left(\vec{\mathbf{q}}_n\right)}{\partial\vec{\mathbf{q}}_n}\right]^{\mathrm{T}}\vec{\lambda}_n+\vec{\mathbf{f}}_d^{+}\left(\vec{\mathbf{q}}_{n-1},\vec{\mathbf{q}}_n\right)\end{aligned}\tag{15}$$

The numerical schemes in Eqs. (14) and (13) are regarded as *momentum matching*, meaning that at every t_n

$$\vec{\mathbf{p}}_n=\vec{\mathbf{p}}^{+}_{n-1,n}=\vec{\mathbf{p}}^{-}_{n,n+1}\tag{16}$$

Thus, as $\vec{\mathbf{p}}_0=\mathbf{M}\dot{\vec{\mathbf{q}}}_0$ at t_0 and given $\vec{\mathbf{q}}_0$ and $\dot{\vec{\mathbf{q}}}_0$ as inputs, solving the following system of nonlinear equations, with $\mathbf{M}$ as the mass matrix, provides $\vec{\mathbf{q}}_1$ and $\vec{\lambda}_0$.

$$\begin{aligned}&\vec{\mathbf{p}}_0=-\frac{\partial L_d\left(\vec{\mathbf{q}}_0,\vec{\mathbf{q}}_1\right)}{\partial\vec{\mathbf{q}}_0}+\frac{1}{2}\left[\frac{\partial\vec{\mathbf{g}}_d\left(\vec{\mathbf{q}}_0\right)}{\partial\vec{\mathbf{q}}_0}\right]^{\mathrm{T}}\vec{\lambda}_0-\vec{\mathbf{f}}_d^{-}\left(\vec{\mathbf{q}}_0,\vec{\mathbf{q}}_1\right)=\mathbf{M}\dot{\vec{\mathbf{q}}}_0,\\&\vec{\mathbf{g}}\left(\vec{\mathbf{q}}_1\right)=\vec{\mathbf{0}}\end{aligned}\tag{17}$$

The solutions computed for the equations of motions can now be used in sensitivity analysis, which is detailed in the next section.

4. Sensitivity analysis using the discrete adjoint variable method

As mentioned earlier, suppose the optimization goal is to minimize the objective function in Eq. (2). Adopting the idea of variational integrators explained in the previous section and assuming a constant time-step size (for simplicity), Eq. (2) can be discretized using one-point quadrature rule as

$$\begin{aligned}\phi&=F\left(\vec{\mathbf{q}}_0,\dot{\vec{\mathbf{q}}}_0,\vec{\mathbf{q}}_T,\dot{\vec{\mathbf{q}}}_T,\vec{\mathbf{a}}\right)+\sum_{n=0}^{N-1}\int_{t_n}^{t_{n+1}}H\left(\vec{\mathbf{q}},\dot{\vec{\mathbf{q}}},\vec{\lambda},\vec{\mathbf{a}}\right)dt\\&=F\left(\vec{\mathbf{q}}_0,\dot{\vec{\mathbf{q}}}_0,\vec{\mathbf{q}}_N,\frac{\vec{\mathbf{q}}_N-\vec{\mathbf{q}}_{N-1}}{h},\vec{\mathbf{a}}\right)\\&\quad+\sum_{n=0}^{N-1}\left(hH\left((1-\alpha)\vec{\mathbf{q}}_n+\alpha\vec{\mathbf{q}}_{n+1},\frac{\vec{\mathbf{q}}_{n+1}-\vec{\mathbf{q}}_n}{h}\right),\vec{\lambda}_n,\vec{\mathbf{a}}\right)\\&=\Phi\left(\vec{\mathbf{q}}_0,\vec{\mathbf{q}}_1,\ldots c,\vec{\mathbf{q}}_N,\dot{\vec{\mathbf{q}}}_0,\vec{\lambda}_0,\vec{\lambda}_1,\ldots c,\vec{\lambda}_{N-1},\vec{\mathbf{a}}\right)\end{aligned}\tag{18}$$

To find a local minimum of this function using a gradient-based approach, its derivative with respect to each design variable is required. For a component a_i of $\vec{\mathbf{a}}$, this is given by

$$\begin{aligned}\frac{d\Phi}{da_i}&=\frac{\partial\Phi}{\partial a_i}+\left[\frac{\partial\Phi}{\partial\dot{\vec{\mathbf{q}}}_0}\right]^{\mathrm{T}}\frac{\dot{\vec{\mathbf{q}}}_0}{\partial a_i}+\sum_{n=0}^{N}\left(\left[\frac{\partial\Phi}{\partial\vec{\mathbf{q}}_n}\right]^{\mathrm{T}}\frac{\vec{\mathbf{q}}_n}{\partial a_i}\right)\\&\quad+\sum_{n=0}^{N-1}\left(\left[\frac{\partial\Phi}{\partial\vec{\lambda}_n}\right]^{\mathrm{T}}\frac{\vec{\lambda}_n}{\partial a_i}\right)\end{aligned}\tag{19}$$

Thus, to compute Eq. (19), one needs to know the derivative of the discrete state variables and Lagrange multipliers with respect to each design variable, namely $\frac{\vec{\mathbf{q}}_n}{\partial a_i}$ and $\frac{\vec{\lambda}_n}{\partial a_i}$. This can be done through the *discrete adjoint variable method* (DAVM). In this section, the detailed derivation of the equations involved in the DAVM and how to solve them are presented.

Based on Eqs. (14) and (17), the set of motion equations to be solved are of the form

$$\begin{aligned}&\vec{\mathbf{c}}_0\left(\vec{\mathbf{q}}_0,\vec{\mathbf{q}}_1,\vec{\lambda}_0,\dot{\vec{\mathbf{q}}}_0,\vec{\mathbf{a}}\right)=\vec{\mathbf{0}}\\&\vec{\mathbf{g}}_1=\vec{\mathbf{g}}\left(\vec{\mathbf{q}}_1,\vec{\mathbf{a}}\right)=\vec{\mathbf{0}}\\&\vec{\mathbf{c}}_n\left(\vec{\mathbf{q}}_{n-1},\vec{\mathbf{q}}_n,\vec{\mathbf{q}}_{n+1},\vec{\lambda}_n,\vec{\mathbf{a}}\right)=\vec{\mathbf{0}}\quad\text{for } n=1,2,\ldots c,N-1\\&\vec{\mathbf{g}}_{n+1}=\vec{\mathbf{g}}\left(\vec{\mathbf{q}}_{n+1},\vec{\mathbf{a}}\right)=\vec{\mathbf{0}}\quad\text{for } n=1,2,\ldots c,N-1\end{aligned}\tag{20}$$

Also, at $t=0$, the following equations stand

$$\begin{aligned}&\vec{\mathbf{g}}_0=\vec{\mathbf{g}}\left(\vec{\mathbf{q}}_0,\vec{\mathbf{a}}\right)=\vec{\mathbf{0}}\\&\frac{\partial\vec{\mathbf{g}}\left(\vec{\mathbf{q}}_0,\vec{\mathbf{a}}\right)}{\partial\vec{\mathbf{q}}_0}\dot{\vec{\mathbf{q}}}_0=\vec{\mathbf{0}}\end{aligned}\tag{21}$$

Differentiating each of these six equations with respect to a_i leads to

$$\begin{aligned}&\frac{\partial\vec{\mathbf{g}}_0}{\partial a_i}+\frac{\partial\vec{\mathbf{g}}_0}{\partial\vec{\mathbf{q}}_0}\frac{\partial\vec{\mathbf{q}}_0}{\partial a_i}=\vec{\mathbf{0}}\\&\frac{\partial}{\partial a_i}\left(\frac{\partial\vec{\mathbf{g}}_0}{\partial\vec{\mathbf{q}}_0}\right)\dot{\vec{\mathbf{q}}}_0+\frac{\partial\vec{\mathbf{g}}_0}{\partial\vec{\mathbf{q}}_0}\frac{\partial\dot{\vec{\mathbf{q}}}_0}{\partial a_i}+\frac{\partial}{\partial\vec{\mathbf{q}}_0}\left(\frac{\partial\vec{\mathbf{g}}_0}{\partial\vec{\mathbf{q}}_0}\right)\frac{\partial\vec{\mathbf{q}}_0}{\partial a_i}\dot{\vec{\mathbf{q}}}_0=\vec{\mathbf{0}}\\&\frac{\partial\vec{\mathbf{c}}_0}{\partial a_i}+\frac{\partial\vec{\mathbf{c}}_0}{\partial\vec{\mathbf{q}}_0}\frac{\partial\vec{\mathbf{q}}_0}{\partial a_i}+\frac{\partial\vec{\mathbf{c}}_0}{\partial\vec{\mathbf{q}}_1}\frac{\partial\vec{\mathbf{q}}_1}{\partial a_i}+\frac{\partial\vec{\mathbf{c}}_0}{\partial\vec{\lambda}_0}\frac{\partial\vec{\lambda}_0}{\partial a_i}+\frac{\partial\vec{\mathbf{c}}_0}{\partial\dot{\vec{\mathbf{q}}}_0}\frac{\partial\dot{\vec{\mathbf{q}}}_0}{\partial a_i}=\vec{\mathbf{0}}\\&\frac{\partial\vec{\mathbf{g}}_1}{\partial a_i}+\frac{\partial\vec{\mathbf{g}}_1}{\partial\vec{\mathbf{q}}_1}\frac{\partial\vec{\mathbf{q}}_1}{\partial a_i}=\vec{\mathbf{0}}\\&\frac{\partial\vec{\mathbf{c}}_n}{\partial a_i}+\frac{\partial\vec{\mathbf{c}}_n}{\partial\vec{\mathbf{q}}_{n-1}}\frac{\partial\vec{\mathbf{q}}_{n-1}}{\partial a_i}+\frac{\partial\vec{\mathbf{c}}_n}{\partial\vec{\mathbf{q}}_n}\frac{\partial\vec{\mathbf{q}}_n}{\partial a_i}+\frac{\partial\vec{\mathbf{c}}_n}{\partial a_i}+\frac{\partial\vec{\mathbf{c}}_n}{\partial\vec{\mathbf{q}}_{n+1}}\frac{\partial\vec{\mathbf{q}}_{n+1}}{\partial a_i}\\&\quad+\frac{\partial\vec{\mathbf{c}}_n}{\partial\vec{\lambda}_n}\frac{\partial\vec{\lambda}_n}{\partial a_i}=\vec{\mathbf{0}}\\&\frac{\partial\vec{\mathbf{g}}_{n+1}}{\partial a_i}+\frac{\partial\vec{\mathbf{g}}_{n+1}}{\partial\vec{\mathbf{q}}_{n+1}}\frac{\partial\vec{\mathbf{q}}_{n+1}}{\partial a_i}=\vec{\mathbf{0}}\end{aligned}\tag{22}$$

Since the geometrical design variables are also considered in this study, the constraint equations $\vec{\mathbf{g}}_n$ are dependent on the design variables, as will be seen in the forthcoming sections. Hence, their derivative with respect to a_i is not zero.

Note: Eq. (22) provides a system of algebraic equations with $\frac{\partial\dot{\vec{\mathbf{q}}}_0}{\partial a_i}$, $\frac{\partial\vec{\mathbf{q}}_n}{\partial a_i}$ and $\frac{\partial\vec{\lambda}_n}{\partial a_i}$ as unknowns. Having done the forward simulation, by solving this system and implementing the computed values in Eq. (19), the sensitivity of the objective function can be calculated. This way of performing the sensitivity analysis is referred to as the *discrete direct differentiation method*. The main issue with this technique is that Eq. (22) needs to be solved separately for all design variables. Therefore, for complex problems with a large number of design variables, it becomes computationally too expensive. For such cases, the DAVM can be utilized, which eliminates the need for computing $\frac{\partial \vec{\mathbf{q}}_n}{\partial a_i}$ and $\frac{\partial \vec{\boldsymbol{\lambda}}_n}{\partial a_i}$ in Eq. (22).

One can introduce *adjoint vectors* $\vec{\boldsymbol{\mu}}_n \in \mathbb{R}^m \, (n = 0, 1, \dots c, N-1)$ and $\vec{\boldsymbol{\eta}}_n \in \mathbb{R}^l \, (n = 1, \dots c, N)$ associated with the dynamic and constraint equations at each time step (m: number of degrees of freedom of the MBS and l: number of holonomic constraints). Since the expressions in Eq. (22) are all equal to zero, multiplying them by the transpose of the adjoint vectors and subtracting them from Eq. (19) does not change the value of $\frac{d\Phi}{da_i}$, regardless of the values of $\vec{\boldsymbol{\mu}}_n$ and $\vec{\boldsymbol{\eta}}_n$. Thus,

$$\begin{aligned}
\frac{d\Phi}{da_i} = {} & \frac{\partial\Phi}{\partial a_i} + \left[\frac{\partial\Phi}{\partial \dot{\vec{\mathbf{q}}}_0}\right]^{\mathrm{T}} \frac{\partial \dot{\vec{\mathbf{q}}}_0}{\partial a_i} + \sum_{n=0}^{N}\left(\left[\frac{\partial\Phi}{\partial \vec{\mathbf{q}}_n}\right]^{\mathrm{T}} \frac{\partial \vec{\mathbf{q}}_n}{\partial a_i}\right) + \sum_{n=0}^{N-1}\left(\left[\frac{\partial\Phi}{\partial \vec{\boldsymbol{\lambda}}_n}\right]^{\mathrm{T}} \frac{\partial \vec{\boldsymbol{\lambda}}_n}{\partial a_i}\right) \\
& - \sum_{n=1}^{N-1}\left(\vec{\boldsymbol{\mu}}_n^{\mathrm{T}}\left[\frac{\partial \vec{\mathbf{c}}_n}{\partial a_i} + \frac{\partial \vec{\mathbf{c}}_n}{\partial \vec{\mathbf{q}}_{n-1}}\frac{\partial \vec{\mathbf{q}}_{n-1}}{\partial a_i} + \frac{\partial \vec{\mathbf{c}}_n}{\partial \vec{\mathbf{q}}_n}\frac{\partial \vec{\mathbf{q}}_n}{\partial a_i} + \frac{\partial \vec{\mathbf{c}}_n}{\partial a_i}\right.\right. \\
& \left.\left. + \frac{\partial \vec{\mathbf{c}}_n}{\partial \vec{\mathbf{q}}_{n+1}}\frac{\partial \vec{\mathbf{q}}_{n+1}}{\partial a_i} + \frac{\partial \vec{\mathbf{c}}_n}{\partial \vec{\boldsymbol{\lambda}}_n}\frac{\partial \vec{\boldsymbol{\lambda}}_n}{\partial a_i}\right]\right) - \sum_{n=1}^{N}\left(\vec{\boldsymbol{\eta}}_n^{\mathrm{T}}\left[\frac{\partial \vec{\mathbf{g}}_n}{\partial a_i} + \frac{\partial \vec{\mathbf{g}}_n}{\partial \vec{\mathbf{q}}_n}\frac{\partial \vec{\mathbf{q}}_n}{\partial a_i}\right]\right) \\
& - \vec{\boldsymbol{\mu}}_0^{\mathrm{T}}\left[\frac{\partial \vec{\mathbf{c}}_0}{\partial a_i} + \frac{\partial \vec{\mathbf{c}}_0}{\partial \vec{\mathbf{q}}_0}\frac{\partial \vec{\mathbf{q}}_0}{\partial a_i} + \frac{\partial \vec{\mathbf{c}}_0}{\partial \vec{\mathbf{q}}_1}\frac{\partial \vec{\mathbf{q}}_1}{\partial a_i} + \frac{\partial \vec{\mathbf{c}}_0}{\partial \vec{\boldsymbol{\lambda}}_0}\frac{\partial \vec{\boldsymbol{\lambda}}_0}{\partial a_i} + \frac{\partial \vec{\mathbf{c}}_0}{\partial \dot{\vec{\mathbf{q}}}_0}\frac{\partial \dot{\vec{\mathbf{q}}}_0}{\partial a_i}\right]
\end{aligned} \tag{23}$$

Re-indexing the terms and putting the similar ones together lead to

$$\begin{aligned}
\frac{d\Phi}{da_i} = {} & \frac{\partial\Phi}{\partial a_i} - \sum_{n=0}^{N-1}\left(\vec{\boldsymbol{\mu}}_n^{\mathrm{T}}\frac{\partial \vec{\mathbf{c}}_n}{\partial a_i}\right) - \sum_{n=1}^{N}\left(\vec{\boldsymbol{\eta}}_n^{\mathrm{T}}\frac{\partial \vec{\mathbf{g}}_n}{\partial a_i}\right) \\
& + \sum_{n=1}^{N-2}\left(\left[\left[\frac{\partial\Phi}{\partial \vec{\mathbf{q}}_n}\right]^{\mathrm{T}} - \vec{\boldsymbol{\mu}}_{n+1}^{\mathrm{T}}\frac{\partial \vec{\mathbf{c}}_{n+1}}{\partial \vec{\mathbf{q}}_n} - \vec{\boldsymbol{\mu}}_n^{\mathrm{T}}\frac{\partial \vec{\mathbf{c}}_n}{\partial \vec{\mathbf{q}}_n}\right.\right. \\
& \left.\left. - \vec{\boldsymbol{\mu}}_{n-1}^{\mathrm{T}}\frac{\partial \vec{\mathbf{c}}_{n-1}}{\partial \vec{\mathbf{q}}_n} - \vec{\boldsymbol{\eta}}_n^{\mathrm{T}}\frac{\partial \vec{\mathbf{g}}_n}{\partial \vec{\mathbf{q}}_n}\right]\frac{\partial \vec{\mathbf{q}}_n}{\partial a_i}\right) \\
& + \sum_{n=0}^{N-1}\left(\left[\left[\frac{\partial\Phi}{\partial \vec{\boldsymbol{\lambda}}_n}\right]^{\mathrm{T}} - \vec{\boldsymbol{\mu}}_n^{\mathrm{T}}\frac{\partial \vec{\mathbf{c}}_n}{\partial \vec{\boldsymbol{\lambda}}_n}\right]\frac{\partial \vec{\boldsymbol{\lambda}}_n}{\partial a_i}\right) \\
& + \left[\left[\frac{\partial\Phi}{\partial \vec{\mathbf{q}}_{N-1}}\right]^{\mathrm{T}} - \vec{\boldsymbol{\mu}}_{N-1}^{\mathrm{T}}\frac{\partial \vec{\mathbf{c}}_{N-1}}{\partial \vec{\mathbf{q}}_{N-1}} - \vec{\boldsymbol{\mu}}_{N-2}^{\mathrm{T}}\frac{\partial \vec{\mathbf{c}}_{N-2}}{\partial \vec{\mathbf{q}}_{N-1}} - \vec{\boldsymbol{\eta}}_{N-1}^{\mathrm{T}}\frac{\partial \vec{\mathbf{g}}_{N-1}}{\partial \vec{\mathbf{q}}_{N-1}}\right] \\
& \times \frac{\partial \vec{\mathbf{q}}_{N-1}}{\partial a_i} + \left[\left[\frac{\partial\Phi}{\partial \vec{\mathbf{q}}_N}\right]^{\mathrm{T}} - \vec{\boldsymbol{\mu}}_{N-1}^{\mathrm{T}}\frac{\partial \vec{\mathbf{c}}_{N-1}}{\partial \vec{\mathbf{q}}_N} - \vec{\boldsymbol{\eta}}_N^{\mathrm{T}}\frac{\partial \vec{\mathbf{g}}_N}{\partial \vec{\mathbf{q}}_N}\right]\frac{\partial \vec{\mathbf{q}}_N}{\partial a_i} \\
& + \left[\left[\frac{\partial\Phi}{\partial \vec{\mathbf{q}}_0}\right]^{\mathrm{T}} - \vec{\boldsymbol{\mu}}_1^{\mathrm{T}}\frac{\partial \vec{\mathbf{c}}_1}{\partial \vec{\mathbf{q}}_0} - \vec{\boldsymbol{\mu}}_0^{\mathrm{T}}\frac{\partial \vec{\mathbf{c}}_0}{\partial \vec{\mathbf{q}}_0}\right]\frac{\partial \vec{\mathbf{q}}_0}{\partial a_i} \\
& + \left[\left[\frac{\partial\Phi}{\partial \dot{\vec{\mathbf{q}}}_0}\right]^{\mathrm{T}} - \vec{\boldsymbol{\mu}}_0^{\mathrm{T}}\frac{\partial \vec{\mathbf{c}}_0}{\partial \dot{\vec{\mathbf{q}}}_0}\right]\frac{\partial \dot{\vec{\mathbf{q}}}_0}{\partial a_i}
\end{aligned} \tag{24}$$

As this equation holds for any values of $\vec{\boldsymbol{\mu}}_n$ and $\vec{\boldsymbol{\eta}}_n$, it is possible to choose these in such a way that bypasses the computation of the gradients of the state variables and Lagrange multipliers in the sensitivity analysis. That is, one can choose $\vec{\boldsymbol{\mu}}_n$ and $\vec{\boldsymbol{\eta}}_n$ so that the coefficients of $\frac{\partial \vec{\mathbf{q}}_n}{\partial a_i}$ and $\frac{\partial \vec{\boldsymbol{\lambda}}_n}{\partial a_i}$ in Eq. (24) all become zero. To do so, first, $\vec{\boldsymbol{\mu}}_{N-1}$, $\vec{\boldsymbol{\mu}}_{N-2}$, $\vec{\boldsymbol{\eta}}_N$ and $\vec{\boldsymbol{\eta}}_{N-1}$ are found from the following two systems of linear algebraic equations

$$\begin{aligned}
& \begin{cases}
\left[\frac{\partial \vec{\mathbf{c}}_{N-1}}{\partial \vec{\mathbf{q}}_N}\right]^{\mathrm{T}} \vec{\boldsymbol{\mu}}_{N-1} + \left[\frac{\partial \vec{\mathbf{g}}_N}{\partial \vec{\mathbf{q}}_N}\right]^{\mathrm{T}} \vec{\boldsymbol{\eta}}_N = \frac{\partial\Phi}{\partial \vec{\mathbf{q}}_N} \\
\left[\frac{\partial \vec{\mathbf{c}}_{N-1}}{\partial \vec{\boldsymbol{\lambda}}_{N-1}}\right]^{\mathrm{T}} \vec{\boldsymbol{\mu}}_{N-1} = \frac{\partial\Phi}{\partial \vec{\boldsymbol{\lambda}}_{N-1}}
\end{cases} \\
& \begin{cases}
\left[\frac{\partial \vec{\mathbf{c}}_{N-2}}{\partial \vec{\mathbf{q}}_{N-1}}\right]^{\mathrm{T}} \vec{\boldsymbol{\mu}}_{N-2} + \left[\frac{\partial \vec{\mathbf{g}}_{N-1}}{\partial \vec{\mathbf{q}}_{N-1}}\right]^{\mathrm{T}} \vec{\boldsymbol{\eta}}_{N-1} = \frac{\partial\Phi}{\partial \vec{\mathbf{q}}_{N-1}} - \left[\frac{\partial \vec{\mathbf{c}}_{N-1}}{\partial \vec{\mathbf{q}}_{N-1}}\right]^{\mathrm{T}} \vec{\boldsymbol{\mu}}_{N-1} \\
\left[\frac{\partial \vec{\mathbf{c}}_{N-2}}{\partial \vec{\boldsymbol{\lambda}}_{N-2}}\right]^{\mathrm{T}} \vec{\boldsymbol{\mu}}_{N-2} = \frac{\partial\Phi}{\partial \vec{\boldsymbol{\lambda}}_{N-2}}
\end{cases}
\end{aligned} \tag{25}$$

Then, $\vec{\boldsymbol{\mu}}_{n-1}$ and $\vec{\boldsymbol{\eta}}_n (n = N-2, N-3, \dots c, 1)$ can be computed by solving the system of linear algebraic equations below

$$\begin{cases}
\left[\frac{\partial \vec{\mathbf{c}}_{n-1}}{\partial \vec{\mathbf{q}}_n}\right]^{\mathrm{T}} \vec{\boldsymbol{\mu}}_{n-1} + \left[\frac{\partial \vec{\mathbf{g}}_n}{\partial \vec{\mathbf{q}}_n}\right]^{\mathrm{T}} \vec{\boldsymbol{\eta}}_n & = \frac{\partial\Phi}{\partial \vec{\mathbf{q}}_n} - \left[\frac{\partial \vec{\mathbf{c}}_{n+1}}{\partial \vec{\mathbf{q}}_n}\right]^{\mathrm{T}} \vec{\boldsymbol{\mu}}_{n+1} - \left[\frac{\partial \vec{\mathbf{c}}_n}{\partial \vec{\mathbf{q}}_n}\right]^{\mathrm{T}} \vec{\boldsymbol{\mu}}_n \\
\left[\frac{\partial \vec{\mathbf{c}}_{n-1}}{\partial \vec{\boldsymbol{\lambda}}_{n-1}}\right]^{\mathrm{T}} \vec{\boldsymbol{\mu}}_{n-1} & = \frac{\partial\Phi}{\partial \vec{\boldsymbol{\lambda}}_{n-1}}
\end{cases} \tag{26}$$

Note: Eqs. (25) and (26) are solved in a backward manner from $n = N$ to $n = 1$ with $\vec{\boldsymbol{\mu}}_{n-1}$ and $\vec{\boldsymbol{\eta}}_n$ as unknowns. That is why the process of solving the adjoint equations is sometimes called *backward simulation*. Unlike Eq. (22) in the discrete direct differentiation method, systems of equations in Eqs. (25) and (26) need to be solved only once for each iteration of optimization. Once $\vec{\boldsymbol{\mu}}_n$ and $\vec{\boldsymbol{\eta}}_n$ are computed and $\frac{\partial \vec{\mathbf{q}}_0}{\partial a_i}$ and $\frac{\dot{\vec{\mathbf{q}}}_0}{\partial a_i}$ are calculated using the first two expressions in Eq. (22), the sensitivity Eq. (24) reduces to

$$\begin{aligned}
\frac{d\Phi}{da_i} = {} & \frac{\partial\Phi}{\partial a_i} - \sum_{n=0}^{N-1}\left(\vec{\boldsymbol{\mu}}_n^{\mathrm{T}}\frac{\partial \vec{\mathbf{c}}_n}{\partial a_i}\right) - \sum_{n=1}^{N}\left(\vec{\boldsymbol{\eta}}_n^{\mathrm{T}}\frac{\partial \vec{\mathbf{g}}_n}{\partial a_i}\right) \\
& + \left[\left[\frac{\partial\Phi}{\partial \vec{\mathbf{q}}_0}\right]^{\mathrm{T}} - \vec{\boldsymbol{\mu}}_1^{\mathrm{T}}\frac{\partial \vec{\mathbf{c}}_1}{\partial \vec{\mathbf{q}}_0} - \vec{\boldsymbol{\mu}}_0^{\mathrm{T}}\frac{\partial \vec{\mathbf{c}}_0}{\partial \vec{\mathbf{q}}_0}\right]\frac{\partial \vec{\mathbf{q}}_0}{\partial a_i} \\
& + \left[\left[\frac{\partial\Phi}{\partial \dot{\vec{\mathbf{q}}}_0}\right]^{\mathrm{T}} - \vec{\boldsymbol{\mu}}_0^{\mathrm{T}}\frac{\partial \vec{\mathbf{c}}_0}{\partial \dot{\vec{\mathbf{q}}}_0}\right]\frac{\partial \dot{\vec{\mathbf{q}}}_0}{\partial a_i}
\end{aligned} \tag{27}$$

This is the final equation for the derivative of the objective function with respect to design variable a_i. The same process can be applied to compute the gradients of optimization constraint functions.

5. Sensitivity with respect to geometrical design variables

As aforementioned, geometrical design variables are those affecting the shape of MBS configurations, such as the dimensions of components and joint positions. Unlike non-geometrical parameters whose values influence only one component of an assembly, geometrical parameters could be related to multiple bodies that share that variable. For instance, changing the position of a joint would impact all the components connected to it. For the sake of brevity and to narrow the premise, this section focuses on MBSs composed of springs, dampers, beams and rigid bodies interacting via spherical (pin) and welded (fixed) joints. The same procedure can be extended to handle more complicated bodies and joints. In this section, first, a brief overview of *rotation-free* formulations for beam and rigid body dynamics, constraint equations between them and the equations of springs and dampers is provided. Then, as an example, the sensitivity analysis of a simple rigid-flexible multibody considering geometrical design variables is described. The avoidance of rotation parameters leads to a *constant* mass matrix for each body. This turns out to be highly beneficial toward having a conserving numerical time-stepping solver [21,22], like the one used in this paper, but also toward simplifying the equations of motion, thus facilitating the required gradient computations with respect to geometrical and non-geometrical design variables.

5.1. Absolute nodal coordinate formulation for beams

Initially proposed by Shabana [23], the *absolute nodal coordinate formulation* (ANCF) uses the *positions* and *slopes* of nodes in the global inertial frame as the generalized coordinates for beams. This results in a constant mass matrix for beams, and subsequently cancels the nonlinear terms of centrifugal and Coriolis inertia forces in the equations of motion. It also leads to simple expressions for the constraint equations, which is of a great benefit for defining different types of joints and including geometrical design variables in the optimization routine. The ANCF has been widely implemented in modeling deformable objects, such as beams, plates and solids, in dynamic problems. It is capable of describing the rigid body modes and accurately solving large deformation problems [21,24–27]. In this paper, the basics of this theory for a two-noded gradient-deficient Euler–Bernoulli beam element with uniform cross-section is provided [28].

Consider a beam element as depicted in Fig. 1. For this element, the position vector $\vec{\mathbf{r}}$ of an arbitrary point in the *global* Cartesian frame can be written as

$$\vec{\mathbf{r}}(x,t) = \mathbf{S}(x)\vec{\mathbf{q}}(t) \tag{28}$$

where x is the local coordinate of that point along the centerline in the *undeformed* configuration. In Eq. (28), $\vec{\mathbf{q}}(t)$ is the *time-dependent* vector of nodal degrees of freedom and $\mathbf{S}$ is the *time-independent* shape function matrix expressed via

$$\begin{aligned}
&\vec{\mathbf{q}}(t) = \begin{bmatrix} \vec{\mathbf{r}}_i^{\mathrm{T}} & \frac{\partial \vec{\mathbf{r}}_i^{\mathrm{T}}}{\partial x} & \vec{\mathbf{r}}_j^{\mathrm{T}} & \frac{\partial \vec{\mathbf{r}}_j^{\mathrm{T}}}{\partial x} \end{bmatrix}^{\mathrm{T}} \\
&\mathbf{S}(x) = [s_1\mathbf{I}_{3\times3} \quad s_2\mathbf{I}_{3\times3} \quad s_3\mathbf{I}_{3\times3} \quad s_4\mathbf{I}_{3\times3}] \\
&s_1 = 1 - 3\xi^2 + 2\xi^3 \\
&s_2 = l(\xi - 2\xi^2 + \xi^3) \\
&s_3 = 3\xi^2 - 2\xi^3 \\
&s_4 = l(\xi^3 - \xi^2)
\end{aligned} \tag{29}$$

in which $\mathbf{I}_{3\times3}$ is the identity matrix, l is the element's length in the undeformed configuration and $\xi = x/l$. To solve the equations of motion described in Section 3, the element's mass matrix and potential energy are required. The former reads as (details are provided in Appendix A)

$$\mathbf{M}_{\mathrm{element}} = \frac{\rho A l}{420}\begin{bmatrix} 156\mathbf{I}_{3\times3} & 22l\mathbf{I}_{3\times3} & 54\mathbf{I}_{3\times3} & -13l\mathbf{I}_{3\times3} \\ 22l\mathbf{I}_{3\times3} & 4l^2\mathbf{I}_{3\times3} & 13l\mathbf{I}_{3\times3} & -3l^2\mathbf{I}_{3\times3} \\ 54\mathbf{I}_{3\times3} & 13l\mathbf{I}_{3\times3} & 156\mathbf{I}_{3\times3} & -22l\mathbf{I}_{3\times3} \\ -13l\mathbf{I}_{3\times3} & -3l^2\mathbf{I}_{3\times3} & -22l\mathbf{I}_{3\times3} & 4l^2\mathbf{I}_{3\times3} \end{bmatrix} \tag{30}$$

where ρ, V and A are the density, volume and cross-sectional area of the element, respectively. The potential energy of the element is due to its elastic deformation and gravity. It is given by

$$\begin{aligned} U_{\mathrm{element}} &= U_{\mathrm{longitudinal}} + U_{\mathrm{transverse}} + U_{\mathrm{gravity}} \\ &= \frac{1}{2}\int_V (EA\varepsilon^2 + EI\kappa^2)dV + \int_V -\rho\vec{\mathbf{v}}^{\mathrm{T}}\vec{\mathbf{r}}\,dV \end{aligned} \tag{31}$$

in which E, I and $\vec{\mathbf{v}}$ are, respectively, Young's modulus, the second moment of area and the gravity vector. In Eq. (31), ε and κ are the element's longitudinal strain and spatial curvature formulated as

$$\varepsilon = \frac{1}{2}\left(\frac{\partial \vec{\mathbf{r}}^{\mathrm{T}}}{\partial x}\frac{\partial \vec{\mathbf{r}}}{\partial x} - 1\right), \quad \kappa = \frac{\left\| \frac{\partial \vec{\mathbf{r}}}{\partial x} \times \frac{\partial^2 \vec{\mathbf{r}}}{\partial x^2} \right\|}{\left\| \frac{\partial \vec{\mathbf{r}}}{\partial x} \right\|^3} \tag{32}$$

In case of small axial deformation and assuming constant longitudinal strain, both expressions in Eq. (32) can be simplified extensively as follows

$$\varepsilon = \frac{d}{l} - 1 = \frac{1}{l}\left\|\vec{\mathbf{r}}_j - \vec{\mathbf{r}}_i\right\| - 1, \quad \kappa = \left\|\frac{\partial^2 \vec{\mathbf{r}}}{\partial x^2}\right\| \tag{33}$$

To perform the forward simulation, as in Section 3, one needs to compute the derivative of element's potential energy, Eq. (31), with respect to $\vec{\mathbf{q}}$, also called the vector of *generalized forces*. Different assumptions and formulas have been proposed for this quantity, some of which could be found in [28,29]. Assuming small axial deformations and using Eqs. 28, 31 and 33, the generalized elastic forces take a linear form as

$$\begin{aligned} \frac{\partial U_{\mathrm{elastic}}}{\partial \vec{\mathbf{q}}} &= \frac{\partial U_{\mathrm{longitudinal}}}{\partial \vec{\mathbf{q}}} + \frac{\partial U_{\mathrm{transverse}}}{\partial \vec{\mathbf{q}}} \\ &= \mathbf{K}_{\mathrm{longitudinal}}\vec{\mathbf{q}} + \mathbf{K}_{\mathrm{transverse}}\vec{\mathbf{q}} \end{aligned} \tag{34}$$

Expressions of $\mathbf{K}_{\mathrm{longitudinal}}$ and $\mathbf{K}_{\mathrm{transverse}}$ are provided in Appendix A. The generalized gravity force vector can be computed by

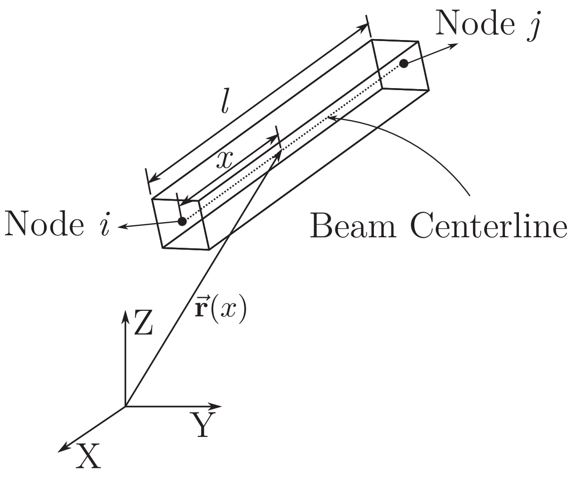

Fig. 1. Two-noded Euler-Bernoulli beam element.

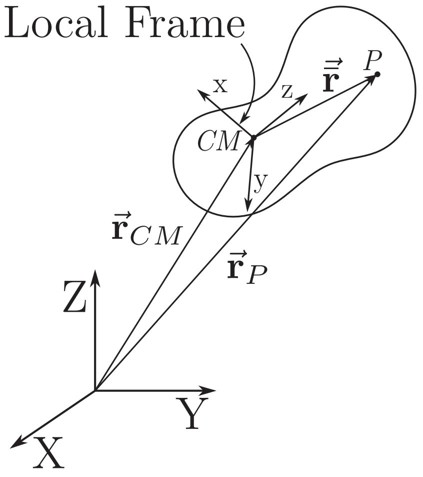

Fig. 2. Rigid body.

$$\frac{\partial U_{\text{gravity}}}{\partial \vec{\mathbf{q}}} = -\left[\int_V \left[\rho \vec{\mathbf{v}}^{\mathrm{T}} \mathbf{S}\right] dV\right] = -\frac{1}{12}\rho A l [6\mathbf{I}_{3\times3} \quad l\mathbf{I}_{3\times3} \quad 6\mathbf{I}_{3\times3} \quad -l\mathbf{I}_{3\times3}]^{\mathrm{T}} \vec{\mathbf{v}} \tag{35}$$

5.2. Natural coordinates for rigid bodies

Natural coordinates, also known as *basic coordinates* or *Cartesian coordinates*, achieve the rotation-free expression for dynamic rigid bodies by introducing a set of redundant degrees of freedom and additional constraints into the equations of motion [30–33]. Similar to ANCF for beams, natural coordinates formulation also results in a constant mass matrix for rigid bodies and extensively simplifies the dynamic and sensitivity equations.

Let $\vec{\mathbf{r}}_{CM}$ and $\mathbf{R}$ denote the global position of the body's center of mass (CM) and the local frame attached to it, respectively, as shown in Fig. 2. If $\vec{\bar{\mathbf{r}}} = [\bar{x} \quad \bar{y} \quad \bar{z}]^{\mathrm{T}}$ is the local position vector of an arbitrary point P on the body, its global position vector is given by

$$\vec{\mathbf{r}}_P = \vec{\mathbf{r}}_{CM} + \mathbf{R}\vec{\bar{\mathbf{r}}} = \vec{\mathbf{r}}_{CM} + \mathbf{R}\begin{Bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{Bmatrix} \tag{36}$$

Suppose $\vec{\mathbf{e}}_1$, $\vec{\mathbf{e}}_2$ and $\vec{\mathbf{e}}_3$ are the unit basis vectors of the local frame. Then, Eq. (36) can be rewritten as

$$\vec{\mathbf{r}}_P(\bar{x},\bar{y},\bar{z},t) = [\mathbf{I}_{3\times3} \quad \bar{x}\mathbf{I}_{3\times3} \quad \bar{y}\mathbf{I}_{3\times3} \quad \bar{z}\mathbf{I}_{3\times3}]\begin{Bmatrix} \vec{\mathbf{r}}_{CM} \\ \vec{\mathbf{e}}_1 \\ \vec{\mathbf{e}}_2 \\ \vec{\mathbf{e}}_3 \end{Bmatrix} = \mathbf{S}(\bar{x},\bar{y},\bar{z})\vec{\mathbf{q}}(t) \tag{37}$$

where $\vec{\mathbf{q}}$ is the *time-dependent* vector of the rigid body's degrees of freedom (generalized coordinates). As only 6 degrees of freedom are enough to fully configure a rigid body in the 3D space and $\vec{\mathbf{q}}$ has 12 elements, a set of additional constraints, called *internal constraints*, are required. They can be stated as

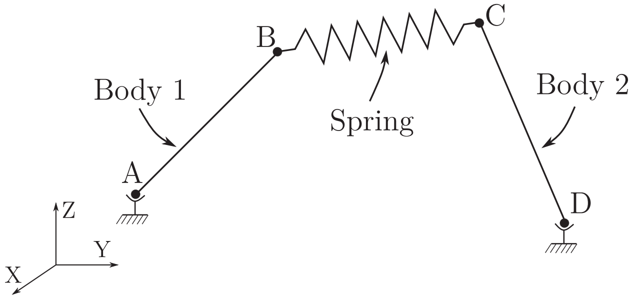

Fig. 3. A rigid-spring-beam assembly.

$$\vec{\mathbf{g}}_{\text{internal}}\left(\vec{\mathbf{q}}\right) = \begin{Bmatrix} \vec{\mathbf{e}}_1 \cdot \vec{\mathbf{e}}_1 - 1 \\ \vec{\mathbf{e}}_2 \cdot \vec{\mathbf{e}}_2 - 1 \\ \vec{\mathbf{e}}_3 \cdot \vec{\mathbf{e}}_3 - 1 \\ \vec{\mathbf{e}}_1 \cdot \vec{\mathbf{e}}_2 \\ \vec{\mathbf{e}}_1 \cdot \vec{\mathbf{e}}_3 \\ \vec{\mathbf{e}}_2 \cdot \vec{\mathbf{e}}_3 \end{Bmatrix} = \vec{\mathbf{0}}_6 \tag{38}$$

Internal constraints, in fact, enforce the *orthonormality* of the rigid body's local coordinate system. If the local frame's axes are aligned with the body's principal axes, incorporating the procedure described in Appendix B, the rigid body's mass matrix takes the following diagonal form

$$\mathbf{M}_{\text{rigid}} = \begin{bmatrix} m\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \frac{1}{2}(I_2+I_3-I_1)\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \frac{1}{2}(I_1+I_3-I_2)\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \frac{1}{2}(I_1+I_2-I_3)\mathbf{I}_{3\times3} \end{bmatrix} \tag{39}$$

in which, m and $I_i(i = 1,2,3)$ are the body's mass and principal moments of inertia, respectively. Since there is no time-term in the mass matrix equation, it remains constant during the forward simulation. To solve the equations described in Section 3, the derivative of body's potential energy, which is only due to gravity, with respect to $\vec{\mathbf{q}}$ is needed. This parameter is called the *generalized gravity force vector* and reads as

$$\frac{\partial U_{\text{gravity}}}{\partial \vec{\mathbf{q}}} = -\left[\int_V \left[\rho \vec{\mathbf{v}}^{\mathrm{T}} \mathbf{S}\right] dV\right] = -m[\nu_x \quad \nu_y \quad \nu_z \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^{\mathrm{T}} \tag{40}$$

where $\vec{\mathbf{v}} = [\nu_x \quad \nu_y \quad \nu_z]^{\mathrm{T}}$ is the gravity vector,

5.3. Constraint equations using ANCF and natural coordinates

A key benefit of utilizing these two rotation-free theories is the simplification they make on the constraint equations. For example, a spherical joint between a point A on a rigid body and an end point B on a beam is expressed as

$$\begin{aligned} \vec{\mathbf{g}}_{\text{spherical}} &= \vec{\mathbf{r}}_{\mathrm{A}} - \vec{\mathbf{r}}_{\mathrm{B}} = \mathbf{S}_{\text{rigid}}\big|_{\mathrm{A}}\vec{\mathbf{q}}_{\text{rigid}} - \mathbf{S}_{\text{beam}}\big|_{\mathrm{B}}\vec{\mathbf{q}}_{\text{beam}} \\ &= \left[\mathbf{S}_{\text{rigid}}\big|_{\mathrm{A}} \quad -\mathbf{S}_{\text{beam}}\big|_{\mathrm{B}}\right]\begin{Bmatrix} \vec{\mathbf{q}}_{\text{rigid}} \\ \vec{\mathbf{q}}_{\text{beam}} \end{Bmatrix} = \vec{\mathbf{0}}_3 \end{aligned} \tag{41}$$

and if instead, this connection is welded, Eq. (41) is augmented by these three more equations

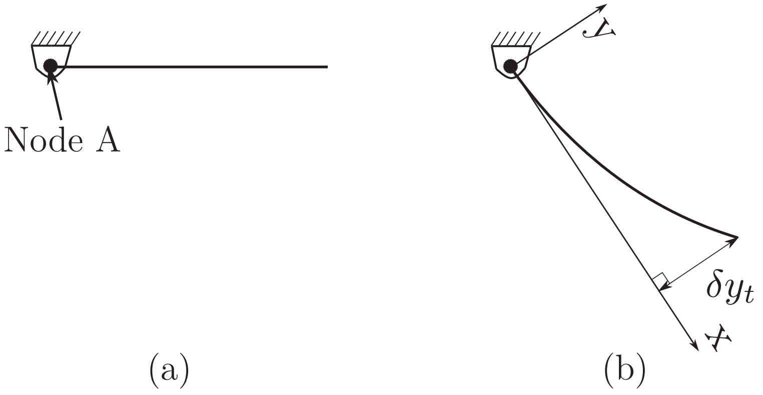

Fig. 4. (a) Pendulum at $t = 0$. (b) Deflected pendulum at time instance t.

$$\vec{\mathbf{g}}_{\text{welded}} = \mathbf{R}^{\mathrm{T}}_{\text{rigid}}\Big|_{t_{\text{current}}} \frac{\partial \vec{\mathbf{r}}_{\mathrm{B}}}{\partial x}\Big|_{t_{\text{current}}} - \mathbf{R}^{\mathrm{T}}_{\text{rigid}}\Big|_{t_0} \frac{\partial \vec{\mathbf{r}}_{\mathrm{B}}}{\partial x}\Big|_{t_0} = \vec{\mathbf{0}}_3 \tag{42}$$

Other types of constraints can be defined in the same fashion [22,34]. Using ANCF and natural coordinates, the constraint Jacobian matrix can be analytically evaluated in a straightforward manner.

5.4. Spring and damper equations

Assume a spring and a damper, between a point C of a rigid body and an end point D of a beam. If k is the spring constant and c denotes the damping coefficient, the *generalized spring* and *damper force* vectors are expressed via

$$\frac{\partial U_{\text{spring}}}{\partial \vec{\mathbf{q}}} = k\,\frac{l_{t_{\text{current}}} - l_0}{l_{t_{\text{current}}}} \begin{bmatrix} \mathbf{S}^{\mathrm{T}}_{\text{rigid}}|_{\mathrm{C}}\mathbf{S}_{\text{rigid}}|_{\mathrm{C}} & -\mathbf{S}^{\mathrm{T}}_{\text{rigid}}|_{\mathrm{C}}\mathbf{S}_{\text{beam}}|_{\mathrm{D}} \\ -\mathbf{S}^{\mathrm{T}}_{\text{beam}}|_{\mathrm{D}}\mathbf{S}_{\text{rigid}}|_{\mathrm{C}} & \mathbf{S}^{\mathrm{T}}_{\text{beam}}|_{\mathrm{D}}\mathbf{S}_{\text{beam}}|_{\mathrm{D}} \end{bmatrix} \begin{Bmatrix} \vec{\mathbf{q}}_{\text{rigid}} \\ \vec{\mathbf{q}}_{\text{beam}} \end{Bmatrix}$$

$$\vec{\mathbf{f}}_{\text{damper}} = -c \begin{bmatrix} \mathbf{S}^{\mathrm{T}}_{\text{rigid}}|_{\mathrm{C}}\mathbf{S}_{\text{rigid}}|_{\mathrm{C}} & -\mathbf{S}^{\mathrm{T}}_{\text{rigid}}|_{\mathrm{C}}\mathbf{S}_{\text{beam}}|_{\mathrm{D}} \\ -\mathbf{S}^{\mathrm{T}}_{\text{beam}}|_{\mathrm{D}}\mathbf{S}_{\text{rigid}}|_{\mathrm{C}} & \mathbf{S}^{\mathrm{T}}_{\text{beam}}|_{\mathrm{D}}\mathbf{S}_{\text{beam}}|_{\mathrm{D}} \end{bmatrix} \begin{Bmatrix} \dot{\vec{\mathbf{q}}}_{\text{rigid}} \\ \dot{\vec{\mathbf{q}}}_{\text{beam}} \end{Bmatrix} \tag{43}$$

where $l_{t_{\text{current}}}$ and l_0 are the current and initial lengths of the spring, respectively. The derivation of Eq. (43) is provided in Appendix C.

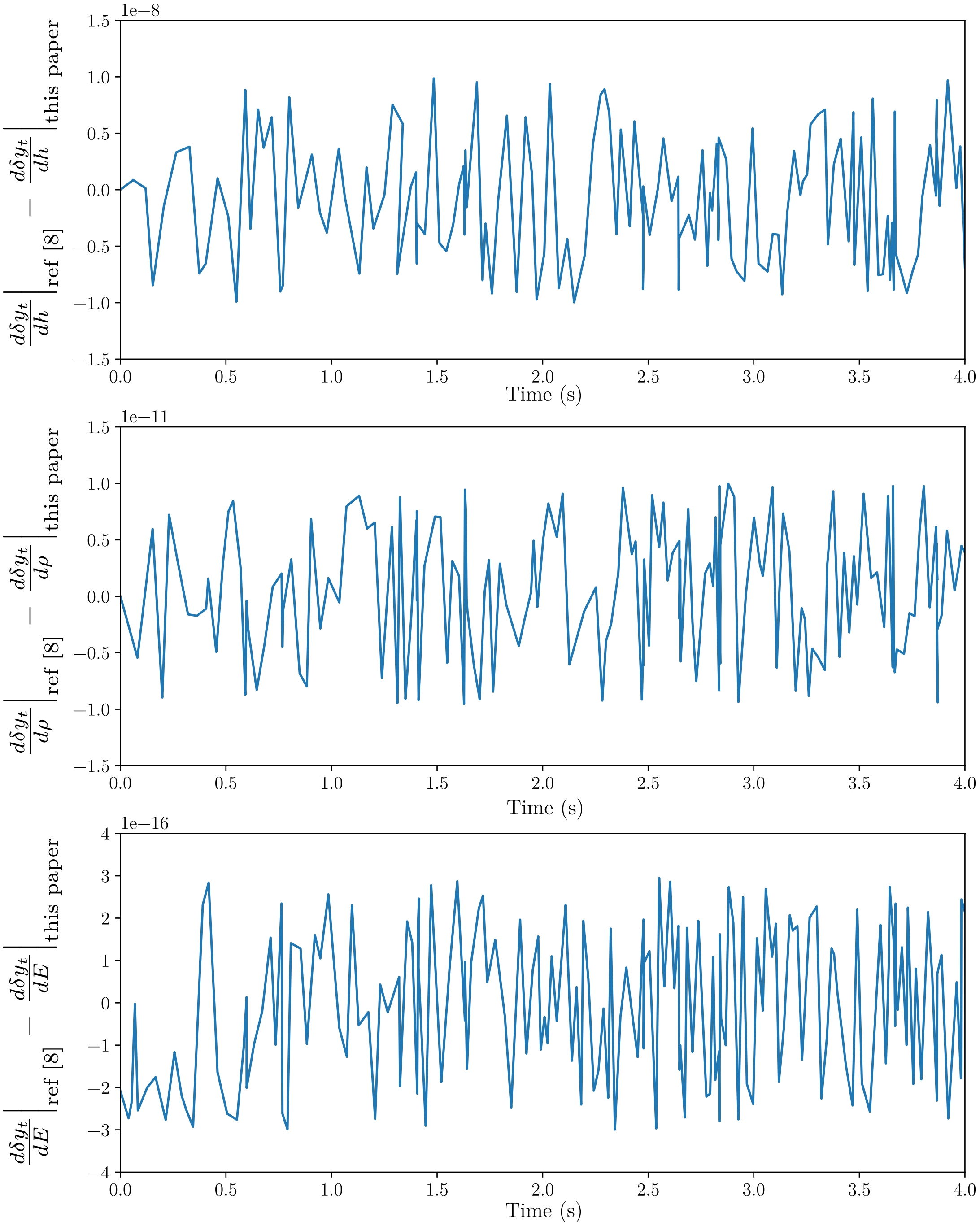

Fig. 5. Difference between the sensitivity values with respect to h, ρ and E from [8] and the proposed DAVM.

5.5. Example: sensitivity analysis for a simple rigid-spring-beam assembly

To demonstrate the techniques developed so far, the sensitivity analysis of a simple rigid-spring-beam multibody is presented. The same computations can be employed for more complex assemblies. Suppose the aim is to minimize the squared norm of the displacement of Node B for the mechanism depicted in Fig. 3. The objective function in this case is

$$\phi = \int_0^T \left\| \vec{\mathbf{d}}_{\mathrm{B}} \right\|^2 dt \tag{44}$$

where $\vec{\mathbf{d}}_{\mathrm{B}}$ represents the time-dependent displacement of Node B. The vector of design variables contains three geometrical variables: the initial global position of Node D along X, Y and Z axes. Body 1 is a rigid bar and Body 2 is a beam discretized by one element with length l and constant cross section A. The assembly is connected to the ground via two spherical joints at Points A and D. If $X_{\mathrm{B}}, Y_{\mathrm{B}}$ and Z_{B} denote the initial global position of Node B at $t = 0$, $\vec{\mathbf{d}}_{\mathrm{B}}$ reads as

$$\vec{\mathbf{d}}_{\mathrm{B}} = \mathbf{S}_{\mathrm{Body1}}|_{\mathrm{B}} \vec{\mathbf{q}}_{\mathrm{Body1}}(t) - \begin{Bmatrix} X_{\mathrm{B}} \\ Y_{\mathrm{B}} \\ Z_{\mathrm{B}} \end{Bmatrix} \tag{45}$$

Following Eq. (18), Eq. (44) can be transformed to

$$\begin{aligned} &\Phi\left(\vec{\mathbf{q}}_{0_{\mathrm{Body1}}}, \vec{\mathbf{q}}_{1_{\mathrm{Body1}}}, \ldots, \vec{\mathbf{q}}_{N_{\mathrm{Body1}}}\right) \\ &\quad = \sum_{n=0}^{N-1} \left(h \left\| \vec{\mathbf{d}}_{\mathrm{B}} \left((1-\alpha) \vec{\mathbf{q}}_{n_{\mathrm{Body1}}} + \alpha \vec{\mathbf{q}}_{(n+1)_{\mathrm{Body1}}} \right) \right\|^2 \right) \end{aligned} \tag{46}$$

Based on the notation used in Eq. (20) and pursuing the equations derived in Sections 3–5, with $\alpha = 0.5$, the set of motion equations for this assembly is

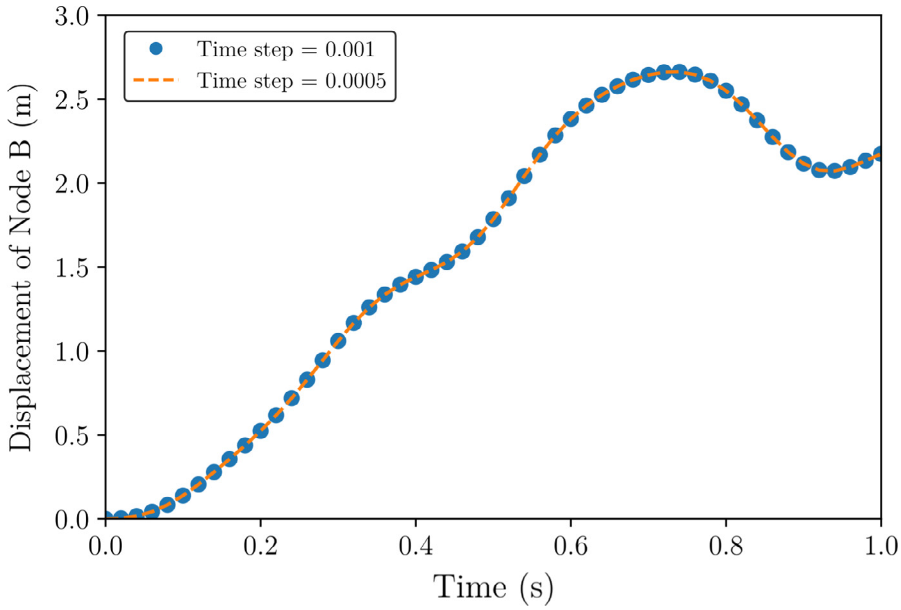

Fig. 6. Displacement of Node B for the two time-steps.

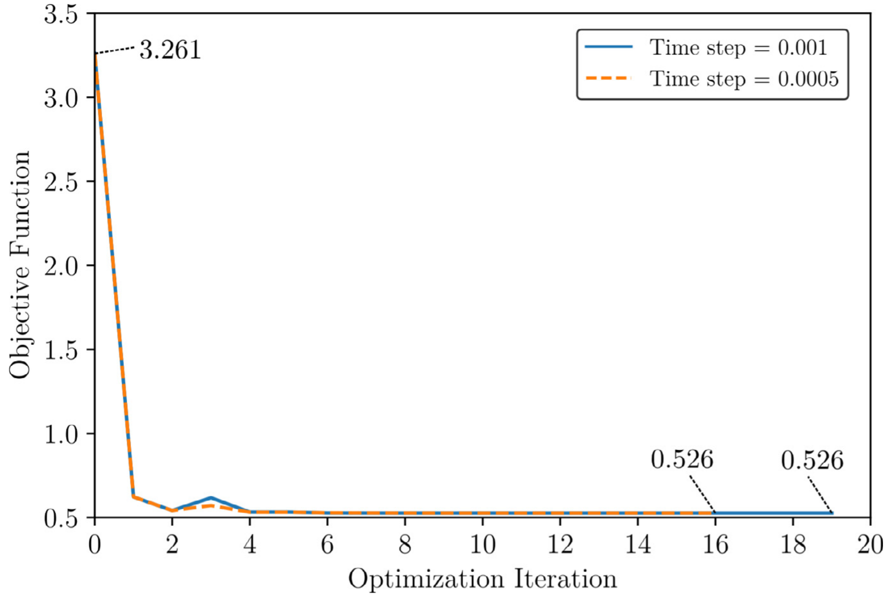

Fig. 7. Evolution of objective function for the two time-steps.

$$\begin{aligned}
\vec{\mathbf{c}}_0 &:= \frac{1}{h}\mathbf{M}\left(\vec{\mathbf{q}}_1 - \vec{\mathbf{q}}_0\right) + \frac{h}{2}\frac{\partial}{\partial \vec{\mathbf{q}}_0}U\left(\frac{\vec{\mathbf{q}}_0 + \vec{\mathbf{q}}_1}{2}\right) \\
&\quad + \frac{h}{2}\left[\frac{\partial \vec{\mathbf{g}}\left(\vec{\mathbf{q}}_0\right)}{\partial \vec{\mathbf{q}}_0}\right]^{\mathrm{T}} \vec{\boldsymbol{\lambda}}_0 - \mathbf{M}\dot{\vec{\mathbf{q}}}_0 = \vec{\mathbf{0}} \\
\vec{\mathbf{c}}_n &:= \frac{1}{h}\mathbf{M}\left(2\vec{\mathbf{q}}_n - \vec{\mathbf{q}}_{n+1} - \vec{\mathbf{q}}_{n-1}\right) \\
&\quad - \frac{h}{2}\left(\frac{\partial}{\partial \vec{\mathbf{q}}_n}U\left(\frac{\vec{\mathbf{q}}_n + \vec{\mathbf{q}}_{n+1}}{2}\right) + \frac{\partial}{\partial \vec{\mathbf{q}}_n}U\left(\frac{\vec{\mathbf{q}}_{n-1} + \vec{\mathbf{q}}_n}{2}\right)\right) \\
&\quad - \frac{h}{2}\left[\frac{\partial \vec{\mathbf{g}}\left(\vec{\mathbf{q}}_n\right)}{\partial \vec{\mathbf{q}}_n}\right]^{\mathrm{T}} \vec{\boldsymbol{\lambda}}_n = \vec{\mathbf{0}} \quad \text{for } n = 1, 2, \ldots c, N-1
\end{aligned} \tag{47}$$

where

$$\vec{\mathbf{q}}_n = \left\{ \begin{array}{c} \vec{\mathbf{q}}_{n_{\mathrm{Body1}}} \\ \vec{\mathbf{q}}_{n_{\mathrm{Body2}}} \end{array} \right\}, \quad \mathbf{M} = \begin{bmatrix} \mathbf{M}_{\mathrm{Body1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\mathrm{Body2}} \end{bmatrix}$$

$$\vec{\mathbf{g}}_n = \vec{\mathbf{g}}\left(\vec{\mathbf{q}}_n\right) = \begin{bmatrix} \mathbf{S}_{\mathrm{Body1}}|_{\mathrm{A}} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{\mathrm{Body2}}|_{\mathrm{D}} \end{bmatrix} \vec{\mathbf{q}}_n - \left\{ \begin{array}{c} X_{\mathrm{A}} \\ Y_{\mathrm{A}} \\ Z_{\mathrm{A}} \\ X_{\mathrm{D}} \\ Y_{\mathrm{D}} \\ Z_{\mathrm{D}} \end{array} \right\} = \vec{\mathbf{0}} \tag{48}$$

$$\frac{\partial U}{\partial \vec{\mathbf{q}}_n} = \frac{\partial U_{\mathrm{gravity}}}{\partial \vec{\mathbf{q}}_n} + \frac{\partial U_{\mathrm{spring}}}{\partial \vec{\mathbf{q}}_n} + \frac{\partial U_{\mathrm{elastic}}}{\partial \vec{\mathbf{q}}_n}$$

These equations, along with the adjoint equations (Eqs. (25) and (26)), as well as Eqs. (28)–(65) can be utilized to find adjoint variables $\vec{\boldsymbol{\mu}}_n$ and $\vec{\boldsymbol{\eta}}_n$.To compute the total derivative based on

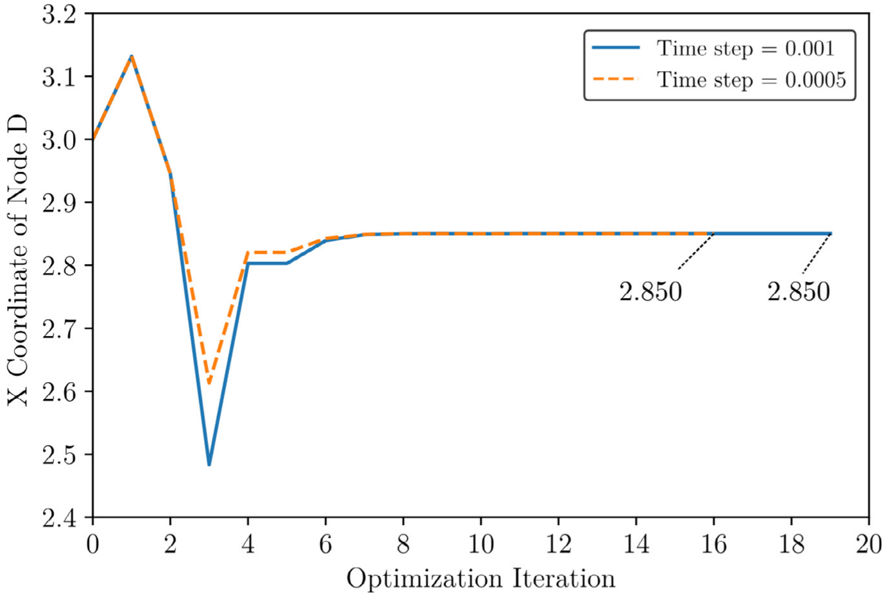

Fig. 8. Evolution of X coordinate of Node D for the two time-steps.

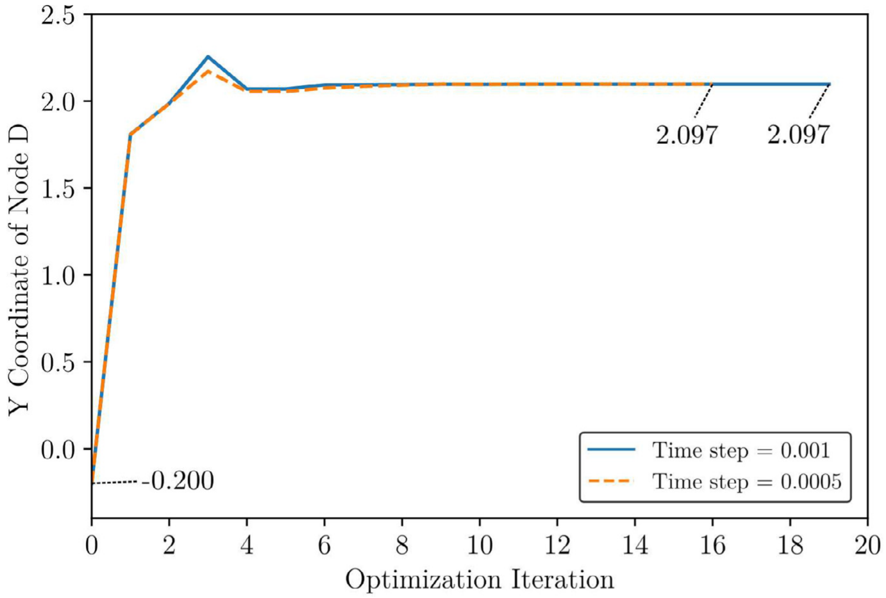

Fig. 9. Evolution of Y coordinate of Node D for the two time-steps.

Eq. (27), $\frac{\partial \Phi}{\partial a_i}$, $\frac{\partial \vec{\mathbf{c}}_n}{\partial a_i}$ and $\frac{\partial \vec{\mathbf{g}}_n}{\partial a_i}$ are also required. If $X_{\mathrm{D}}, Y_{\mathrm{D}}$ and Z_{D} are the initial global position of Node D at $t = 0$, since there is no explicit dependency in Φ on the given design variables, $\frac{\partial \Phi}{\partial X_{\mathrm{D}}} = \frac{\partial \Phi}{\partial Y_{\mathrm{D}}} = \frac{\partial \Phi}{\partial Z_{\mathrm{D}}} = 0$. As for the constraint equations, there are two spherical joints in this example, only one of which is a function of $X_{\mathrm{D}}, Y_{\mathrm{D}}$ and Z_{D}. Using Eq. (48)

$$\frac{\partial \vec{\mathbf{g}}_n}{\partial X_{\mathrm{D}}} = -\begin{Bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{Bmatrix}, \quad \frac{\partial \vec{\mathbf{g}}_n}{\partial Y_{\mathrm{D}}} = -\begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{Bmatrix}, \quad \frac{\partial \vec{\mathbf{g}}_n}{\partial Z_{\mathrm{D}}} = -\begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{Bmatrix} \tag{49}$$

In the equations of $\vec{\mathbf{c}}_0$ and $\vec{\mathbf{c}}_n$, Eq. (47), only the terms related to Body 2 are explicitly dependent on $X_{\mathrm{D}}, Y_{\mathrm{D}}$ and Z_{D}. They are

- $\mathbf{M}_{\text{Body2}}$
  According to Eq. (30)

$$\frac{\partial \mathbf{M}_{\text{Body2}}}{\partial a_i} = \frac{\partial \mathbf{M}_{\text{Body2}}}{\partial l} \frac{\partial l}{\partial a_i} \tag{50}$$

- $\partial U_{\text{gravity}_{\text{Body2}}} / \partial \vec{\mathbf{q}}_n$
  Denoting $\vec{\mathbf{v}}$ as the gravity vector and following Eq. (35)

$$\frac{\partial}{\partial a_i}\left[\frac{\partial U_{\text{gravity}_{\text{Body2}}}}{\partial \vec{\mathbf{q}}_n}\right] = -\frac{\partial}{\partial l}\left[\int_V \left[\rho \vec{\mathbf{v}}^{\mathrm{T}} \mathbf{S}\right] dV\right] \frac{\partial l}{\partial a_i} \tag{51}$$

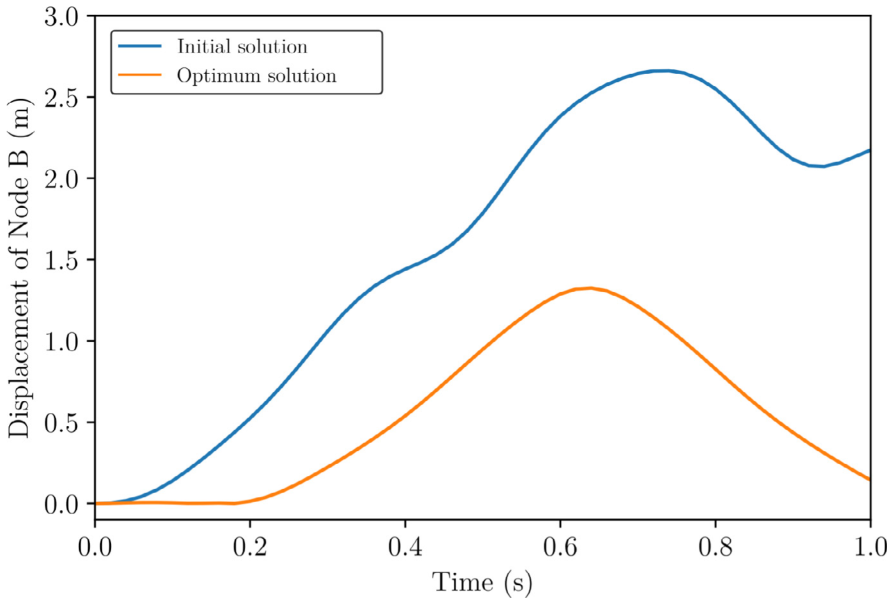

Fig. 10. Displacement of Node B in the initial and optimum configurations.

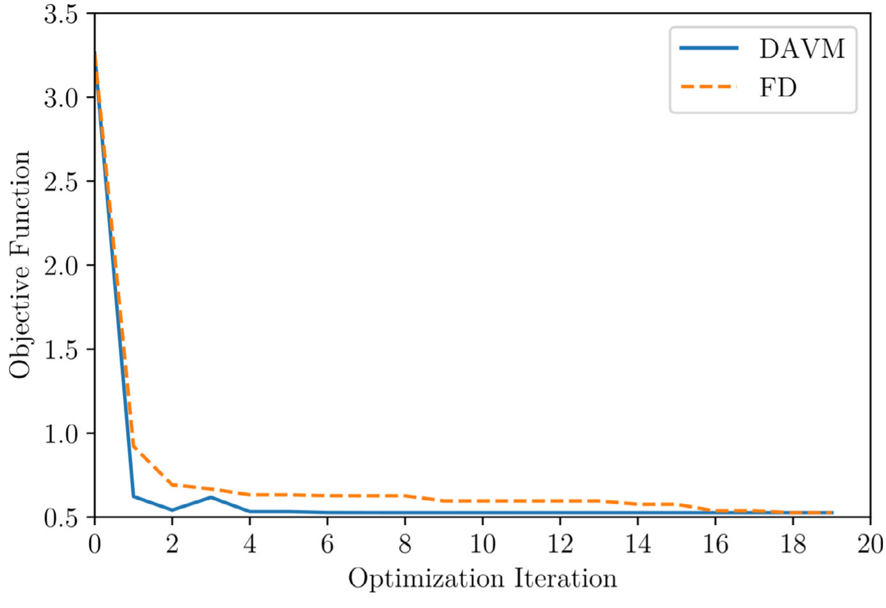

Fig. 11. Evolution of objective function using DAVM and FD for $\Delta t = 0.001$ s.

- $\partial U_{\text{longitudinal}_{\text{Body2}}} / \partial \vec{\mathbf{q}}_n$ and $\partial U_{\text{transverse}_{\text{Body2}}} / \partial \vec{\mathbf{q}}_n$
Assuming small longitudinal deformation and utilizing Eq. (34)

$$\begin{aligned} \frac{\partial}{\partial a_i}\left[\frac{\partial U_{\text{longitudinal}_{\text{Body2}}}}{\partial \vec{\mathbf{q}}_n}\right] &= \frac{\partial \mathbf{K}_{\text{longitudinal}}}{\partial l}\frac{\partial l}{\partial a_i} \\ \frac{\partial}{\partial a_i}\left[\frac{\partial U_{\text{longitudinal}_{\text{Body2}}}}{\partial \vec{\mathbf{q}}_n}\right] &= \frac{\partial \mathbf{K}_{\text{transverse}}}{\partial l}\frac{\partial l}{\partial a_i} \end{aligned} \tag{52}$$

In Eqs. (50)–(52), l is the beam's length in the undeformed configuration, which is calculated by

$$l = \sqrt{(X_{\text{D}} - X_{\text{C}})^2 + (Y_{\text{D}} - Y_{\text{C}})^2 + (Z_{\text{D}} - Z_{\text{C}})^2} \tag{53}$$

and its derivatives are

$$\frac{\partial l}{\partial X_{\text{D}}} = \frac{X_{\text{D}} - X_{\text{C}}}{l}, \quad \frac{\partial l}{\partial Y_{\text{D}}} = \frac{Y_{\text{D}} - Y_{\text{C}}}{l}, \quad \frac{\partial l}{\partial Z_{\text{D}}} = \frac{Z_{\text{D}} - Z_{\text{C}}}{l} \tag{54}$$

Having computed all these terms and substituting them into Eq. (27), the sensitivity of the given objective function is thus found.

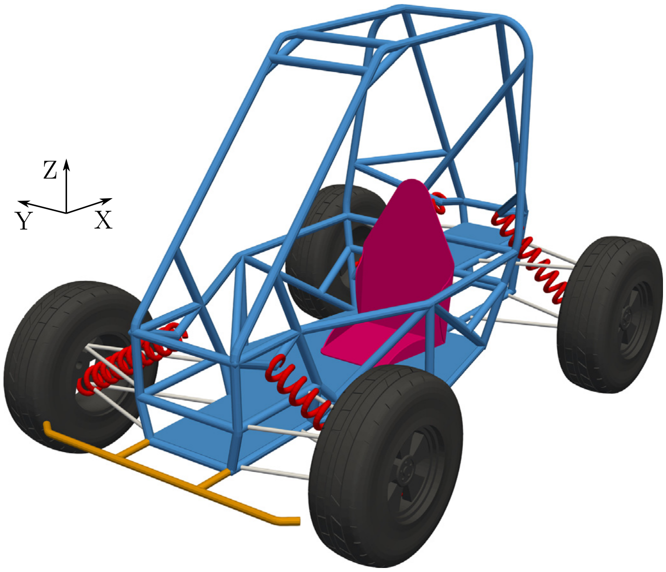

Fig. 12. The car with double-wishbone suspension system.

For cases with a higher number of components and other types of geometrical design variables, derivatives are obtainable in a similar manner.

6. Numerical examples

Three numerical test cases are provided in this section. The first example is to validate the proposed sensitivity analysis technique based on the study reported in [8]. The other two are to compare its performance with the finite difference method (FD) and assay its applicability to the cases containing geometrical and non-geometrical design variables. In all examples, a Newton-Raphson scheme is used to solve the linear and nonlinear equations of forward and backward simulations. Once the required gradients are computed, the NLopt library [35], which is an open-source nonlinear optimization library, is utilized to find the optimum solutions. To handle the optimization constraints, the Augmented Lagrangian method is exploited. In all examples, α is set to 0.5 in the forward and backward simulations. The optimization scheme is stopped when the objective function improvement is less than 10^{-6} for five consecutive iterations or the maximum number of iterations, set to 60, is reached.

6.1. Sensitivity analysis of a flexible pendulum

Consider the planar flexible pendulum shown in Fig. 4. It is made of a material with the density $\rho = 4000\,\text{kg/m}^3$, Young's modulus $E = 10^7\,\text{N/m}^2$ and Poisson's ratio 0.3. The length of the beam is 1.2 m, discretized by 5 equi-length elements, and the width (h) of its square cross-section is 0.05 m. The duration of the problem is 4 s and the beam is initially at rest horizontally along the global X axes.

Suppose δy_t is the deflection of the tip at time t, and the goal is to compute the sensitivity of this parameter at different time instances with respect to h, ρ and E. δy_t is measured along the y axis of the local coordinate system whose origin is at Node A and its x axis is tangent to the beam's centerline at that node. Fig. 5 shows the difference between the values reported in [8] and those computed using the developed DAVM in this paper. The results are almost identical, and the small differences are due to the different time-stepping methods for solving the forward and backward simulation equations.

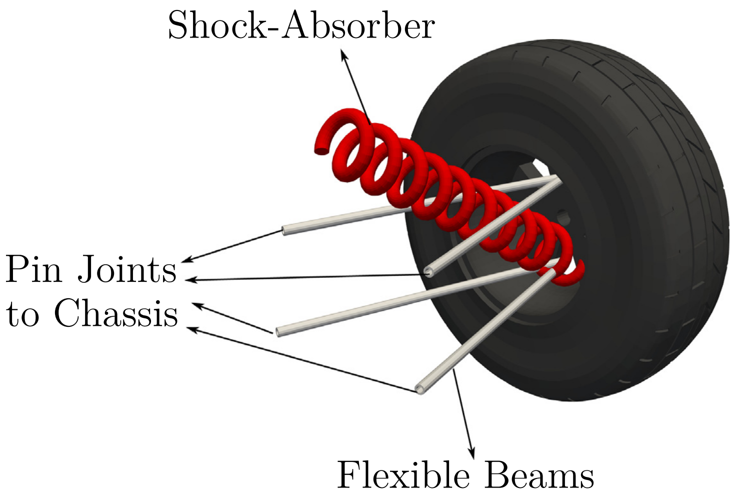

Fig. 13. A suspension system module.

6.2. Optimization of a rigid-spring-beam assembly

For the second example, the two-dimensional version of the assembly discussed in Section 5 and shown in Fig. 3 is considered. The system is connected to the ground via two spherical joints at Nodes A and D. Body 1 is a rigid cylinder connecting Nodes A and B. Its mass and cross-sectional diameter are 1.6 kg and 0.01 m, respectively, and its center of mass is located in the midpoint between the two ends. Body 2 is a flexible beam with Young's modulus of 70 GPa, Poisson's ratio of 0.3 and density of $3700\,\text{kg/m}^3$. Its cross-section is a solid square with 0.01 m width. A linear spring connects Nodes B and C. The spring constant is $100\,\text{N/m}^3$ and its initial undeformed length is equal to the distance between Nodes B and C at time zero.

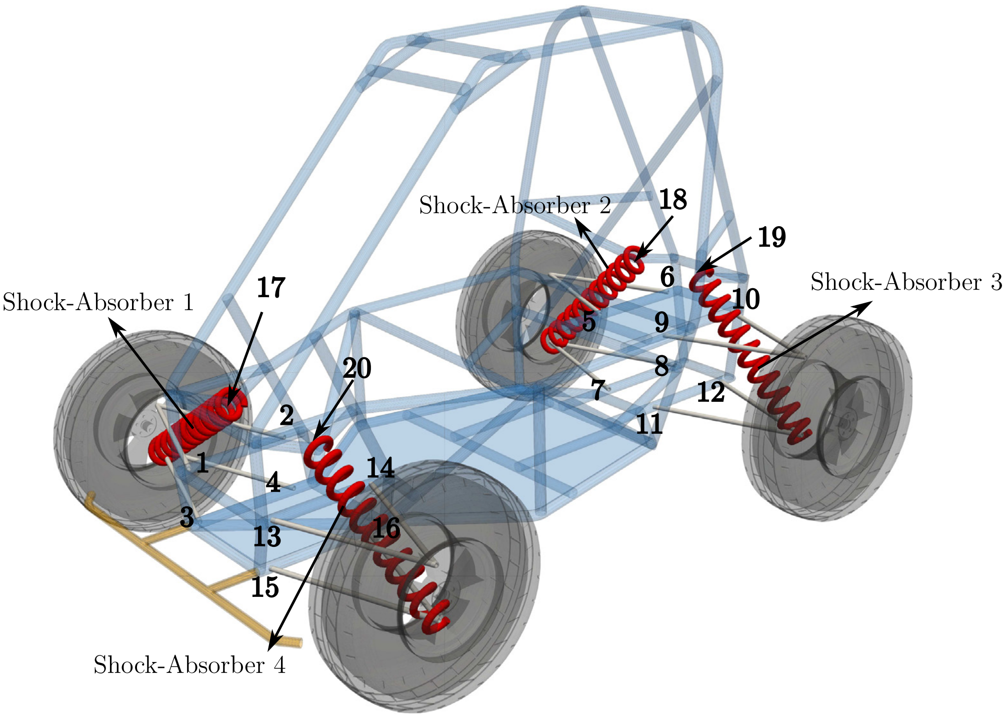

Fig. 14. Node IDs associated with design variables.

Table 1
Initial values and range of design variables for suspension system example.

Design variables (a_i)	Initial values	Lower bounds (lb_i)	Upper bounds (ub_i)
(X_1, Y_1, Z_1)	$(-1.270, 0.210, 0.150)$ m	$(-1.350, 0.100, 0.100)$	$(-1.100, 0.350, 0.300)$
(X_2, Y_2, Z_2)	$(-0.960, 0.210, 0.150)$ m	$(-1.100, 0.100, 0.100)$	$(-0.800, 0.350, 0.300)$
(X_3, Y_3, Z_3)	$(-1.270, 0.180, 0)$ m	$(-1.350, 0.100, -0.100)$	$(-1.100, 0.350, 0.100)$
(X_4, Y_4, Z_4)	$(-0.960, 0.180, 0)$ m	$(-1.100, 0.100, -0.100)$	$(-0.800, 0.350, 0.100)$
(X_5, Y_5, Z_5)	$(0.210, 0.140, 0.250)$ m	$(0.100, 0.100, 0.100)$	$(0.350, 0.300, 0.300)$
(X_6, Y_6, Z_6)	$(0.520, 0.140, 0.250)$ m	$(0.400, 0.100, 0.100)$	$(0.650, 0.300, 0.300)$
(X_7, Y_7, Z_7)	$(0.210, 0.100, 0)$ m	$(0.100, 0.100, -0.100)$	$(0.350, 0.300, 0.100)$
(X_8, Y_8, Z_8)	$(0.520, 0.100, 0)$ m	$(0.400, 0.100, -0.100)$	$(0.650, 0.300, 0.100)$
(X_9, Y_9, Z_9)	$(0.210, -0.140, 0.250)$ m	$(0.100, -0.300, 0.100)$	$(0.350, -0.100, 0.300)$
(X_{10}, Y_{10}, Z_{10})	$(0.520, -0.140, 0.250)$ m	$(0.400, -0.300, 0.100)$	$(0.650, -0.100, 0.300)$
(X_{11}, Y_{11}, Z_{11})	$(0.210, -0.100, 0)$ m	$(0.100, -0.300, -0.100)$	$(0.350, -0.100, 0.100)$
(X_{12}, Y_{12}, Z_{12})	$(0.520, -0.100, 0)$ m	$(0.400, -0.300, -0.100)$	$(0.650, -0.100, 0.100)$
(X_{13}, Y_{13}, Z_{13})	$(-1.270, -0.210, 0.150)$ m	$(-1.350, -0.350, 0.100)$	$(-1.100, -0.100, 0.300)$
(X_{14}, Y_{14}, Z_{14})	$(-0.960, -0.210, 0.150)$ m	$(-1.100, -0.350, 0.100)$	$(-0.800, -0.100, 0.300)$
(X_{15}, Y_{15}, Z_{15})	$(-1.270, -0.180, 0)$ m	$(-1.350, -0.350, -0.100)$	$(-1.100, -0.100, 0.100)$
(X_{16}, Y_{16}, Z_{16})	$(-0.960, -0.180, 0)$ m	$(-1.100, -0.350, -0.100)$	$(-0.800, -0.100, 0.100)$
Z_{17}	0.3 m	0.15	0.6
Z_{18}	0.4 m	0.15	0.6
Z_{19}	0.4 m	0.15	0.6
Z_{20}	0.3 m	0.15	0.6
k_1	10000 N/m	100	30000
k_2	10000 N/m	100	30000
k_3	10000 N/m	100	30000
k_4	10000 N/m	100	30000
c_1	50 N.s/m	20	1000
c_2	500 N.s/m	20	1000
c_3	500 N.s/m	20	1000
c_4	50 N.s/m	20	1000

In this example, the duration is 1 s, and the problem is run by two different time-step sizes of 0.001 s and 0.0005 s. To drive the mechanism an initial angular velocity of −5 rad/s about the positive Z axis is applied to Body 2. The objective function is the squared norm of Node B's displacement over the entire simulation duration. A lower bound constraint is imposed on the beam's length to prevent producing a beam with zero length. This optimization problem can be formulated as

$$
\begin{aligned}
& \min_{X_{\mathrm{D}},Y_{\mathrm{D}}} \int_0^1 \left\| \vec{\mathbf{d}}_{\mathrm{B}} \right\|^2 dt \\
\text{subject to} \quad & \\
& l_{\mathrm{Body2}} \geqslant 0.001\mathrm{m}, \\
& -10\mathrm{m} \leqslant X_{\mathrm{D}}, Y_{\mathrm{D}} \leqslant 10\mathrm{m}
\end{aligned}
\tag{55}
$$

Fig. 6 shows the displacement of Node B in time for the initial configuration for the two given time-step sizes. As can be seen, the plots for the two time-steps coincide and so the forward simulation phase is insensitive to these two time-step sizes.

Table 2
Front and rear wheels excitation.

Front wheels	$d_z = 0.050\sin(2\pi t)$
Rear wheels	$d_z = 0.075\sin(4\pi t)$

Table 3
Physical and material properties for suspension system example.

Parameters	Values
mass of chassis	300 kg
mass of wheels	20 kg
Young's modulus of beams	200 GPa
Poisson's ratio of beams	0.3
outer radius of beams' cross-section	0.015 m
inner radius of beams' cross-section	0.01 m
density of beams	7800 kg/m^3
location of chassis's center of mass	$(0.243, 0, 0.365)$ m
location of front-left wheel's center of mass	$(-1.110, -0.680, 0.075)$ m
location of front-right wheel's center of mass	$(-1.110, 0.680, 0.075)$ m
location of rear-left wheel's center of mass	$(0.370, -0.560, 0.125)$ m
location of rear-right wheel's center of mass	$(0.370, 0.560, 0.125)$ m

Figs. 7–9 present the evolution of the objective function and design variables for both time-step sizes. Even though the optimization procedure proceeds slightly differently for the two time-steps, the final optimum solutions are almost exactly identical. These figures confirm that not only the forward simulation phase, but also the backward simulation and consequently the sensitivity values and optimization results are independent of the time-step size. This is, however, not the case in the approximation methods, such as the equivalent static loads (ESL). In ESL, the sensitivity values are computed for a sequence of equivalent linear static models which have the same response fields as those in the actual dynamic problem at a selected number of time instances. Depending on which time-steps are taken into account and for how many of them the equivalent model is constructed, the optimum solutions could be entirely different (e.g., the numerical examples in [36]). It has been even shown that ESL does not in general lead to optimal designs [37], whereas the proposed sensitivity analysis technique as explained in Section 2 guarantees moving along the descent direction (or ascent direction in maximization problems) and subsequently improving the defined objective function.

Fig. 10 shows the displacement of Node B in the initial and optimum configurations. According to this figure, the displacement of Node B in the optimum solution is considerably smaller than that in the initial solution for all time steps, therefore leading to a significantly better performance in terms of the defined criterion in Eq. (55).

To further investigate the performance of the proposed sensitivity analysis technique, this problem is run using FD as well. Fig. 11 depicts the optimization evolution for the developed DAVM and FD with 1% perturbation value. Even though both approaches converge to the same optimum solution, their computation time is considerably different. On the same computer, the DAVM executes about 4.5 times faster than FD. This is mainly due to the fact that in FD to compute the sensitivities for each design variable, an additional round of simulation is required. The difference is even more significant for more complex examples, which makes FD almost impractical in those cases.

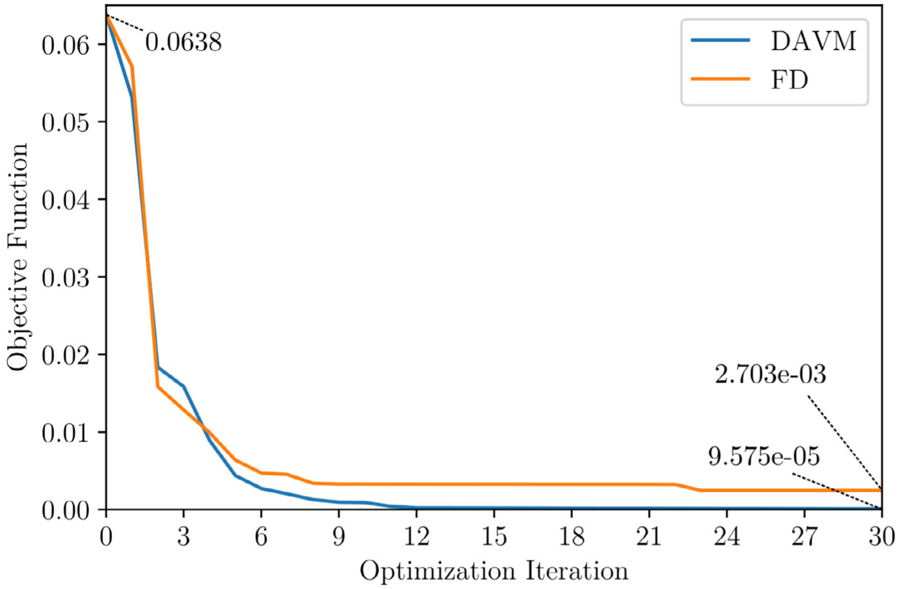

Fig. 15. Evolution of objective function for the suspension system example.

6.3. Optimization of an automotive double-wishbone suspension system

To examine the developed methods on a large-scale engineering case study, they are applied to optimize the double-wishbone suspension system of the car depicted in Fig. 12. There are four suspension system modules, as portrayed in Fig. 13, each consists of two wish-bone like structures and a shock-absorber. The wishbones are composed of two hollow circular flexible beams which are welded together from one end and connected to the wheels from another end through spherical joints. The shock-absorbers are modeled by a spring and a damper, connected to the wheels and the chassis. The rest of the bodies are considered as rigid bodies. No symmetry is imposed on the design in order to give the optimization scheme the freedom to find the optimum values of design variables independently for each of the four suspension system modules.

The objective function is the squared magnitude of displacement of the chassis's center of mass over time. There are 52 geometrical and 8 non-geometrical design variables in this problem. The geometrical variables are the position of the connection points between the beams and chassis (X, Y and Z coordinates of each point) and those between the shock-absorbers and chassis along the global Z axis (see Fig. 14). The non-geometrical variables are

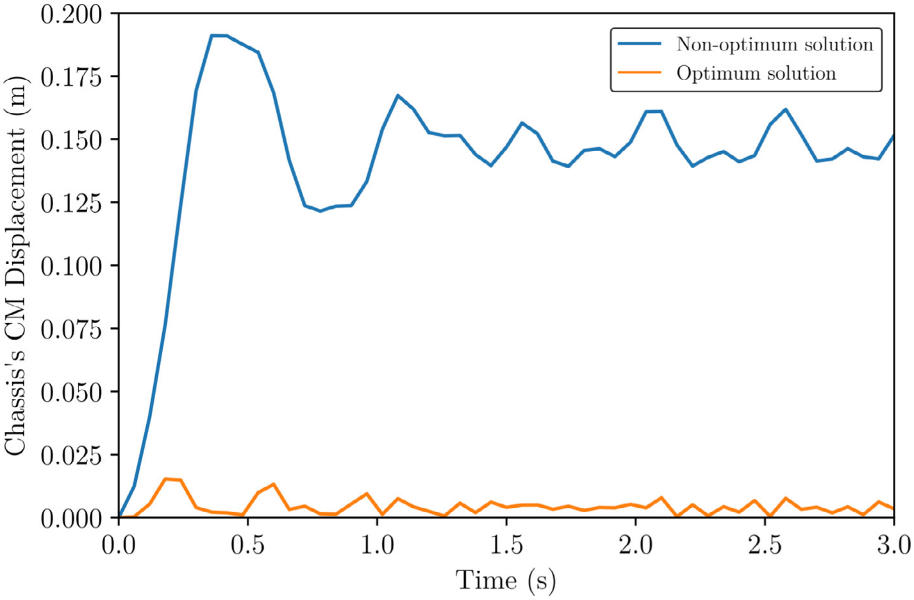

Fig. 16. Displacement of chassis's center of mass in the non-optimum (initial) and optimum configurations.

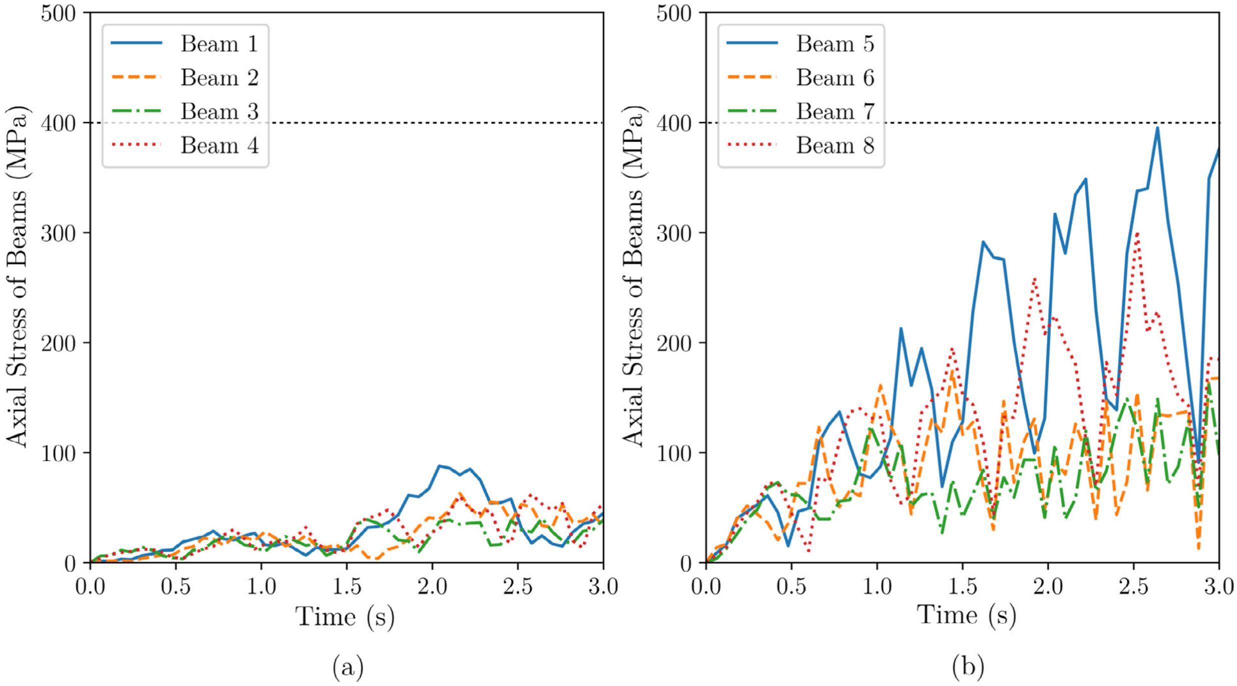

Fig. 17. Maximum axial stress of beams in the optimum solution (a) Front beams. (b) Rear beams.

Table 4
Optimum values of design variables for suspension system example

Design variables	Optimum values (DAVM)	Optimum values (FD)
(X_1, Y_1, Z_1)	$(-1.27028, 0.20972, 0.14977)$ m	$(-1.27051, 0.20937, 0.14930)$ m
(X_2, Y_2, Z_2)	$(-0.96033, 0.21016, 0.15049)$ m	$(-0.96044, 0.21009, 0.15107)$ m
(X_3, Y_3, Z_3)	$(-1.27009, 0.18006, 0.00561)$ m	$(-1.27005, 0.17924, 0.01282)$ m
(X_4, Y_4, Z_4)	$(-0.96021, 0.18013, 0.00234)$ m	$(-0.96011, 0.17940, 0.00849)$ m
(X_5, Y_5, Z_5)	$(0.21030, 0.13594, 0.26894)$ m	$(0.21081, 0.13529, 0.27875)$ m
(X_6, Y_6, Z_6)	$(0.51827, 0.14297, 0.23080)$ m	$(0.51602, 0.14415, 0.21930)$ m
(X_7, Y_7, Z_7)	$(0.20898, 0.11675, -0.03229)$ m	$(0.20936, 0.11558, -0.02645)$ m
(X_8, Y_8, Z_8)	$(0.51810, 0.12266, -0.04791)$ m	$(0.51797, 0.12282, -0.04917)$ m
(X_9, Y_9, Z_9)	$(0.210301, -0.13596, 0.26894)$ m	$(0.21081, -0.13529, 0.27874)$ m
(X_{10}, Y_{10}, Z_{10})	$(0.51827, -0.14299, 0.23080)$ m	$(0.51602, -0.14415, 0.21930)$ m
(X_{11}, Y_{11}, Z_{11})	$(0.20898, -0.11675, -0.03229)$ m	$(0.20935, -0.11558, -0.02644)$ m
(X_{12}, Y_{12}, Z_{12})	$(0.51810, -0.12265, -0.04791)$ m	$(0.51797, -0.12282, -0.04917)$ m
(X_{13}, Y_{13}, Z_{13})	$(-1.27028, -0.20972, 0.14977)$ m	$(-1.27051, -0.20938, 0.14930)$ m
(X_{14}, Y_{14}, Z_{14})	$(-0.96033, -0.21016, 0.15049)$ m	$(-0.96044, -0.21009, 0.15107)$ m
(X_{15}, Y_{15}, Z_{15})	$(-1.27009, -0.18006, 0.00561)$ m	$(-1.27005, -0.17924, 0.01282)$ m
(X_{16}, Y_{16}, Z_{16})	$(-0.96021, -0.18014, 0.00234)$ m	$(-0.96011, -0.17941, 0.00850)$ m
Z_{17}	0.18536 m	0.1500 m
Z_{18}	0.28891 m	0.27918 m
Z_{19}	0.28891 m	0.27918 m
Z_{20}	0.18536 m	0.1500 m
k_1	7552.080 N/m	3780.350 N/m
k_2	21048.100 N/m	21058.600 N/m
k_3	21048.100 N/m	21058.600 N/m
k_4	7552.080 N/m	3780.350 N/m
c_1	102.625 N.s/m	122.734 N.s/m
c_2	457.101 N.s/m	364.194 N.s/m
c_3	457.101 N.s/m	364.194 N.s/m
c_4	102.625 N.s/m	122.734 N.s/m

spring constants and damping coefficients of each shock-absorber. The lower and upper bounds and initial values of each parameter are reported in Table 1.

To mimic the driving condition on a rugged terrain, the front and rear wheels are subject to two different harmonic excitations along the global Z axis as described in Table 2. For this problem, the final time T and the time-step size are 3 s and 0.001 s, respectively. Other parameters are summarized in Table 3.

The optimization problem in this case is written as

$$\min_{a_i \in \mathbb{R}^{60}} \int_0^3 \left\| \vec{\mathbf{d}}_{CM} \right\|^2 dt$$

subject to

$$\begin{aligned} & lb_i \leqslant a_i \leqslant ub_i \quad (i = 1, \ldots, 60), \\ & -400 \text{ MPa} \leqslant \sigma_{beam_j} \leqslant 400 \text{ MPa} \quad (j = 1, \ldots, 16), \\ & l_{beam_j} \geqslant 0.001\text{m} \quad (j = 1, \ldots, 16) \end{aligned} \tag{56}$$

Similar to the previous example, for the sensitivity analysis, FD with 1% perturbation value is employed as well. Fig. 15 shows the evolution of the objective function using the proposed DAVM and FD. Both methods converge to an optimum design, however, the one found by the DAVM has a better objective function value. This could be due to approximating the sensitivity values in FD, as opposed to having the exact values using the DAVM. Errors in computing the gradients drive the optimization procedure along either a wrong direction (i.e., deteriorating the performance) or the one that does not improve the performance the most (i.e., slowing down the convergence). In this example, both methods eventually lead to the same solution, if they are run for 60 iterations. In terms of the computation time, on the same computer, the DAVM runs about 40 times faster than FD for 60 optimization iterations.

Fig. 16 is the plot of total displacement of the chassis's center of mass for the initial and optimum designs utilizing the DAVM. It well confirms that the optimum solution outperforms the initial design in terms of the defined performance criterion.

Fig. 17 shows the variation of beams' maximum axial stress for the optimum solution. Since the model is symmetric with respect to XZ plane, only the values for one half of the car are plotted. It is clear that all stress constraints are satisfied. Also, as the center of mass of the chassis is closer to the rear of the car, beams belonging to the rear suspension system module (Beams 5, 6, 7, 8) are subject to larger forces and subsequently have higher stress values.

Optimum values of the objective function and design variables using the proposed DAVM and FD are reported in Table 4. Although no symmetry constraint is imposed on the front and rear suspension modules, the optimum values suggest a symmetric solution for each pair, as expected.

7. Conclusion

The current paper makes the following contributions in the area of design optimization of dynamic flexible MBSs. First, it presents the novel application of the DAVM for computing the sensitivities in design optimization of flexible dynamic multibody systems. In order to derive the adjoint equations, the discrete, instead of continuous, dynamic equations are used directly. This leads to a system of linear algebraic equations, which is in general less computationally expensive to solve compared to the system of differential algebraic equations produced by the CAVM. It also yields the exact values of gradients of discrete objective and constraint functions, thus ensuring the move along the actual descent or ascent directions in the optimization problems. Although the relevant adjoint equations are generated using a specific type of geometric variational integrators for the forward simulation, the proposed sensitivity analysis technique can be integrated, with minor modifications, with other types of numerical time-stepping methods as well.

Another contribution is to demonstrate how to handle geometrical, as well as non-geometrical, design variables. From the design perspective, geometrical variables such as initial joint positions and linkage lengths are challenging yet significant part of the design embodiment process. The inclusion of the geometrical variables increases the design space dimensionality, and subsequently assists the designer in finding the optimum solution. A key consideration in dealing with geometrical design variables is the feasibility of the generated solutions. To prevent creating infeasible designs, a proper set of constraints should be determined for the optimization routine. For example, in the design of a rigid four-bar linkage where the crank is needed to make a full revolution, relevant dimensional constraints (according to Grashof's rules) must be devised to satisfy this requirement.

Lastly, this paper shows how rigid and flexible bodies (e.g., beams) can be effectively incorporated by utilizing natural coordinates and ANCF. The use of rotation-free formulations results in constant mass matrices for the bodies, which extensively simplifies the equations involved in both forward and backward simulations. In addition, it facilitates differentiating the equations with respect to geometrical and non-geometrical design variables, hence serving as an essential component for the contributions described above. It is worth noting that the proposed sensitivity analysis technique guarantees that the optimization routine finds an optimum solution and moves toward improving the defined performance criteria, as proved in Section 2 and shown by the three numerical examples. It is well-known that if the initial solution is far away from the global optimum, it is possible that the optimization scheme falls into a local optimum. This is an intrinsic characteristic of all gradient-based optimizers and has been addressed extensively in the mathematics and engineering literature [38,39]. A possible remedy to this issue is to start the optimization procedure from different initial designs, thus increasing the chance of finding the global optimum.

As a closing remark, the authors would like to emphasize that with the growing interests in generative design (i.e., design automation) in engineering applications, it is imperative to develop simulation and numeral techniques that can be effective in searching high-dimensional design spaces. The current work has highlighted some of the challenges in this endeavor and contributed an effective solution in the domain of dynamic multibody systems.

Appendix A. Beam's mass and stiffness matrices

Differentiating Eq. (29) with respect to time results in the global velocity vector below for a beam element

$$\dot{\overrightarrow{\mathbf{r}}}(x) = \mathbf{S}(x)\dot{\overrightarrow{\mathbf{q}}}(t) \tag{57}$$

Using this and the kinetic energy equation, the element's mass matrix can be obtained as

$$\begin{aligned} T_{\text{element}} &= \frac{1}{2}\int_V \left(\rho\left[\dot{\overrightarrow{\mathbf{r}}}^{\mathrm{T}}\dot{\overrightarrow{\mathbf{r}}}\right]\right)dV = \frac{1}{2}\dot{\overrightarrow{\mathbf{q}}}^{\mathrm{T}}\left[\int_V \left(\rho\left[\mathbf{S}^{\mathrm{T}}\mathbf{S}\right]\right)dV\right]\dot{\overrightarrow{\mathbf{q}}} \\ &= \frac{1}{2}\dot{\overrightarrow{\mathbf{q}}}^{\mathrm{T}}\mathbf{M}_{\text{element}}\dot{\overrightarrow{\mathbf{q}}} \\ \mathbf{M}_{\text{element}} &= \frac{\rho A l}{420}\begin{bmatrix} 156\mathbf{I}_{3\times3} & 22l\mathbf{I}_{3\times3} & 54\mathbf{I}_{3\times3} & -13l\mathbf{I}_{3\times3} \\ 22l\mathbf{I}_{3\times3} & 4l^2\mathbf{I}_{3\times3} & 13l\mathbf{I}_{3\times3} & -3l^2\mathbf{I}_{3\times3} \\ 54\mathbf{I}_{3\times3} & 13l\mathbf{I}_{3\times3} & 156\mathbf{I}_{3\times3} & -22l\mathbf{I}_{3\times3} \\ -13l\mathbf{I}_{3\times3} & -3l^2\mathbf{I}_{3\times3} & -22l\mathbf{I}_{3\times3} & 4l^2\mathbf{I}_{3\times3} \end{bmatrix} \end{aligned} \tag{58}$$

$\mathbf{K}_{\text{longitudinal}}$ and $\mathbf{K}_{\text{transverse}}$ for the element are

$$\begin{aligned} \mathbf{K}_{\text{longitudinal}} &= \tfrac{EA}{l}\varepsilon\begin{bmatrix} \tfrac{6}{5}\mathbf{I}_{3\times3} & \tfrac{l}{10}\mathbf{I}_{3\times3} & -\tfrac{6}{5}\mathbf{I}_{3\times3} & \tfrac{l}{10}\mathbf{I}_{3\times3} \\ \tfrac{l}{10}\mathbf{I}_{3\times3} & \tfrac{2l^2}{15}\mathbf{I}_{3\times3} & -\tfrac{l}{10}\mathbf{I}_{3\times3} & -\tfrac{l^2}{30}\mathbf{I}_{3\times3} \\ -\tfrac{6}{5}\mathbf{I}_{3\times3} & -\tfrac{l}{10}\mathbf{I}_{3\times3} & \tfrac{6}{5}\mathbf{I}_{3\times3} & -\tfrac{l}{10}\mathbf{I}_{3\times3} \\ \tfrac{l}{10}\mathbf{I}_{3\times3} & -\tfrac{l^2}{30}\mathbf{I}_{3\times3} & -\tfrac{l}{10}\mathbf{I}_{3\times3} & \tfrac{2l^2}{30}\mathbf{I}_{3\times3} \end{bmatrix} \\ \mathbf{K}_{\text{transverse}} &= \tfrac{EA}{l^3}\varepsilon\begin{bmatrix} 12\mathbf{I}_{3\times3} & 6l\mathbf{I}_{3\times3} & -12\mathbf{I}_{3\times3} & 6l\mathbf{I}_{3\times3} \\ 6l\mathbf{I}_{3\times3} & 4l^2\mathbf{I}_{3\times3} & -6l\mathbf{I}_{3\times3} & 2l^2\mathbf{I}_{3\times3} \\ -12\mathbf{I}_{3\times3} & -6l\mathbf{I}_{3\times3} & 12\mathbf{I}_{3\times3} & -6l\mathbf{I}_{3\times3} \\ 6l\mathbf{I}_{3\times3} & 2l^2\mathbf{I}_{3\times3} & -6l\mathbf{I}_{3\times3} & 4l^2\mathbf{I}_{3\times3} \end{bmatrix} \end{aligned} \tag{59}$$

Appendix B. Rigid body's mass matrix

Using the kinetic energy equation,

$$\begin{aligned} T_{\text{rigid}} &= \frac{1}{2}\int_V \left(\rho\left[\dot{\overrightarrow{\mathbf{r}}}^{\mathrm{T}}\dot{\overrightarrow{\mathbf{r}}}\right]\right)dV = \frac{1}{2}\dot{\overrightarrow{\mathbf{q}}}^{\mathrm{T}}\left[\int_V \left(\rho\left[\mathbf{S}^{\mathrm{T}}\mathbf{S}\right]\right)dV\right] \\ &= \frac{1}{2}\dot{\overrightarrow{\mathbf{q}}}^{\mathrm{T}}\mathbf{M}_{\text{rigid}}\dot{\overrightarrow{\mathbf{q}}} \end{aligned} \tag{60}$$

and $\mathbf{S}$ in Eq. (37), the body's mass matrix becomes

$$\mathbf{M}_{\text{rigid}} = \int_V \rho \begin{bmatrix} \mathbf{I}_{3\times3} & \bar{x}\mathbf{I}_{3\times3} & \bar{y}\mathbf{I}_{3\times3} & \bar{z}\mathbf{I}_{3\times3} \\ \bar{x}\mathbf{I}_{3\times3} & \bar{x}^2\mathbf{I}_{3\times3} & \bar{x}\bar{y}\mathbf{I}_{3\times3} & \bar{x}\bar{z}\mathbf{I}_{3\times3} \\ \bar{y}\mathbf{I}_{3\times3} & \bar{x}\bar{y}\mathbf{I}_{3\times3} & \bar{y}^2\mathbf{I}_{3\times3} & \bar{y}\bar{z}\mathbf{I}_{3\times3} \\ \bar{z}\mathbf{I}_{3\times3} & \bar{x}\bar{z}\mathbf{I}_{3\times3} & \bar{y}\bar{z}\mathbf{I}_{3\times3} & \bar{z}^2\mathbf{I}_{3\times3} \end{bmatrix} dV \tag{61}$$

If the origin of the body's local frame is located at its center of mass and the local frame's axes coincide with the body's principal axes, the mass matrix takes a diagonal form.

Appendix C. Spring and damper's generalized force vectors

The potential energy of the spring depicted in Fig. 3 is defined by

$$U_{\text{spring}} = \frac{1}{2}k(l_{t_{\text{current}}} - l_0)^2 \tag{62}$$

in which l_0 and $l_{t_{\text{current}}}$ are, respectively, the spring's lengths at zero and current time-step. Considering Eqs. (28) and (37), Eq. (62) can be rewritten as

$$\begin{aligned} U_{\text{spring}} &= \tfrac{1}{2}k\left(\sqrt{\left(\overrightarrow{\mathbf{r}}_{\mathrm{C}} - \overrightarrow{\mathbf{r}}_{\mathrm{D}}\right)^{\mathrm{T}}\left(\overrightarrow{\mathbf{r}}_{\mathrm{C}} - \overrightarrow{\mathbf{r}}_{\mathrm{D}}\right)} - l_0\right)^2 \\ &= \tfrac{1}{2}k\left(\sqrt{\overrightarrow{\bar{\mathbf{q}}}^{\mathrm{T}}\begin{bmatrix} \mathbf{S}^{\mathrm{T}}_{\text{rigid}}|_{\mathrm{C}}\mathbf{S}_{\text{rigid}}|_{\mathrm{C}} & -\mathbf{S}^{\mathrm{T}}_{\text{rigid}}|_{\mathrm{C}}\mathbf{S}_{\text{beam}}|_{\mathrm{D}} \\ -\mathbf{S}^{\mathrm{T}}_{\text{beam}}|_{\mathrm{D}}\mathbf{S}_{\text{rigid}}|_{\mathrm{C}} & \mathbf{S}^{\mathrm{T}}_{\text{beam}}|_{\mathrm{D}}\mathbf{S}_{\text{beam}}|_{\mathrm{D}} \end{bmatrix}\overrightarrow{\bar{\mathbf{q}}}} - l_0\right)^2 \end{aligned} \tag{63}$$

where $\overrightarrow{\bar{\mathbf{q}}} = \left[\overrightarrow{\mathbf{q}}^{\mathrm{T}}_{\text{rigid}} \quad \overrightarrow{\mathbf{q}}^{\mathrm{T}}_{\text{beam}}\right]^{\mathrm{T}}$. The *generalized spring force* vector, thus, becomes

$$\frac{\partial U_{\text{spring}}}{\partial \overrightarrow{\bar{\mathbf{q}}}} = k\frac{l_{t_{\text{current}}} - l_0}{l_{t_{\text{current}}}}\begin{bmatrix} \mathbf{S}^{\mathrm{T}}_{\text{rigid}}|_{\mathrm{C}}\mathbf{S}_{\text{rigid}}|_{\mathrm{C}} & -\mathbf{S}^{\mathrm{T}}_{\text{rigid}}|_{\mathrm{C}}\mathbf{S}_{\text{beam}}|_{\mathrm{D}} \\ -\mathbf{S}^{\mathrm{T}}_{\text{beam}}|_{\mathrm{D}}\mathbf{S}_{\text{rigid}}|_{\mathrm{C}} & \mathbf{S}^{\mathrm{T}}_{\text{beam}}|_{\mathrm{D}}\mathbf{S}_{\text{beam}}|_{\mathrm{D}} \end{bmatrix}\begin{Bmatrix} \overrightarrow{\mathbf{q}}_{\text{rigid}} \\ \overrightarrow{\mathbf{q}}_{\text{beam}} \end{Bmatrix} \tag{64}$$

Following a similar procedure, the vector of *generalized damper forces* is stated via

$$\overrightarrow{\mathbf{f}}_{\text{damper}} = -c\begin{bmatrix} \mathbf{S}^{\mathrm{T}}_{\text{rigid}}|_{\mathrm{C}}\mathbf{S}_{\text{rigid}}|_{\mathrm{C}} & -\mathbf{S}^{\mathrm{T}}_{\text{rigid}}|_{\mathrm{C}}\mathbf{S}_{\text{beam}}|_{\mathrm{D}} \\ -\mathbf{S}^{\mathrm{T}}_{\text{beam}}|_{\mathrm{D}}\mathbf{S}_{\text{rigid}}|_{\mathrm{C}} & \mathbf{S}^{\mathrm{T}}_{\text{beam}}|_{\mathrm{D}}\mathbf{S}_{\text{beam}}|_{\mathrm{D}} \end{bmatrix}\begin{Bmatrix} \dot{\overrightarrow{\mathbf{q}}}_{\text{rigid}} \\ \dot{\overrightarrow{\mathbf{q}}}_{\text{beam}} \end{Bmatrix} \tag{65}$$

References

[1] Tromme E, Brüls O, Emonds-Alt J, Bruyneel M, Virlez G, Duysinx P. Discussion on the optimization problem formulation of flexible components in multibody systems. Struct Multidiscipl Optim 2013;48(6):1189–206.

[2] Greene WH, Haftka RT. Computational aspects of sensitivity calculations in transient structural analysis. Comput Struct 1989;32(2):433–43.

[3] Maly T, Petzold LR. Numerical methods and software for sensitivity analysis of differential-algebraic systems. Appl Numer Math 1996;20(1–2):57–79.

[4] Mukherjee RM, Bhalerao KD, Anderson KS. A divide-and-conquer direct differentiation approach for multibody system sensitivity analysis. Struct Multidiscipl Optim 2008;35(5):413–29.

[5] Dias JMP, Pereira MS. Sensitivity analysis of rigid-flexible multibody systems. Multibody Syst Dyn 1997;1(3):303–22.

[6] Wang X, Haug EJ, Pan W. Implicit numerical integration for design sensitivity analysis of rigid multibody systems. Mech Based Des Struct Mach 2005;33 (1):1–30.

[7] Haug EJ, Ehle PE. Second-order design sensitivity analysis of mechanical system dynamics. Int J Numer Meth Eng 1982;18(11):1699–717.

[8] Pi T, Zhang Y, Chen L. First order sensitivity analysis of flexible multibody systems using absolute nodal coordinate formulation. Multibody Syst Dyn 2012;27(2):153–71.

[9] Tromme E, Tortorelli D, Brüls O, Duysinx P. Structural optimization of multibody system components described using level set techniques. Struct Multidiscipl Optim 2015;52(5):959–71.

[10] Schaffer AS. On the adjoint formulation of design sensitivity analysis of multibody dynamics. Theses Dissert 2005:93.

[11] Ding J-Y, Pan Z-K, Chen L-Q. Second order adjoint sensitivity analysis of multibody systems described by differential–algebraic equations. Multibody Syst Dyn 2007;18(4):599–617.

[12] Zhu Y, Dopico D, Sandu C, Sandu A. Dynamic response optimization of complex multibody systems in a penalty formulation using adjoint sensitivity. J Comput Nonlinear Dyn 2015;10(3):31009.

[13] Cao Y, Li S, Petzold L. Adjoint sensitivity analysis for differential-algebraic equations: algorithms and software. J Comput Appl Math 2002;149 (1):171–91.

[14] Li S, Petzold L, Zhu W. Sensitivity analysis of differential–algebraic equations: a comparison of methods on a special problem. Appl Numer Math 2000;32 (2):161–74.

[15] Ober-Blöbaum S, Junge O, Marsden JE. Discrete mechanics and optimal control: an analysis. ESAIM: Control Optim Cal Var 2011;17(2):322–52.

[16] Lauß T, Oberpeilsteiner S, Steiner W, Nachbagauer K. The discrete adjoint method for parameter identification in multibody system dynamics. Multibody Syst Dyn 2017:1–14.

[17] Leyendecker S, Marsden JE, Ortiz M. Variational integrators for constrained dynamical systems. ZAMM–J Appl Math Mech/Z Angew Math Mech 2008;88 (9):677–708.

[18] Marsden JE, West M. Discrete mechanics and variational integrators. Acta Numer 2001;10:357–514.

[19] Kane C, Marsden JE, Ortiz M, West M. Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems. Int J Numer Meth Eng 2000;49(10):1295–325.

[20] Stern A, Desbrun M. Discrete geometric mechanics for variational time integrators. In: ACM SIGGRAPH 2006 courses. ACM; 2006. p. 75–80.

[21] Shabana AA, Yakoub RY. Three dimensional absolute nodal coordinate formulation for beam elements: theory. J Mech Des 2001;123(4):606–13.

[22] Uhlar S, Betsch P. A rotationless formulation of multibody dynamics: modeling of screw joints and incorporation of control constraints. Multibody Syst Dyn 2009;22(1):69–95.

[23] Shabana AA. An absolute nodal coordinate formulation for the large rotation and deformation analysis of flexible bodies. Technical Report. Department of Mechanical Engineering, University of Illinois at Chicago; 1996.

[24] Gerstmayr J, Shabana AA. Analysis of thin beams and cables using the absolute nodal co-ordinate formulation. Nonlinear Dyn 2006;45(1):109–30.

[25] Garcia-Vallejo D, Escalona JL, Mayo J, Dominguez J. Describing rigid-flexible multibody systems using absolute coordinates. Nonlinear Dyn 2003;34 (1):75–94.

[26] Gerstmayr J, Sugiyama H, Mikkola A. Review on the absolute nodal coordinate formulation for large deformation analysis of multibody systems. J Comput Nonlinear Dyn 2013;8(3):31016.

[27] Yakoub RY, Shabana AA. Three dimensional absolute nodal coordinate formulation for beam elements: implementation and applications. J Mech Des 2001;123(4):614–21.

[28] Berzeri M, Shabana AA. Development of simple models for the elastic forces in the absolute nodal co-ordinate formulation. J Sound Vib 2000;235(4):539–65.

[29] Takahashi Y, Shimizu N. Study on elastic forces of the absolute nodal coordinate formulation for deformable beams. In: ASME proceedings of design engineering technical conference; 1999. p. 33–40.

[30] de Jalon JG, Unda J, Avello A, Jiménez JM. Dynamic analysis of three-dimensional mechanisms in "natural" coordinates. J Mech Transm Autom Des 1987;109(4):460–5.

[31] De Jalon JG, Bayo E. Kinematic and dynamic simulation of multibody systems: the real-time challenge. Springer Science & Business Media; 2012.

[32] Betsch P, Steinmann P. Constrained integration of rigid body dynamics. Comput Meth Appl Mech Eng 2001;191(3):467–88.

[33] de Jalón JG. Twenty-five years of natural coordinates. Multibody Syst Dyn 2007;18(1):15–33.

[34] Garcia-Vallejo D, Mayo J, Escalona JL, Dominguez J. Three-dimensional formulation of rigid-flexible multibody systems with flexible beam elements. Multibody Syst Dyn 2008;20(1):1–28.

[35] Johnson SG. The nlopt nonlinear-optimization package; 2014.

[36] Sun J, Tian Q, Hu H. Structural optimization of flexible components in a flexible multibody system modeled via ANCF. Mech Mach Theory 2016;104:59–80.

[37] Stolpe M, Verbart A, Rojas-Labanda S. The equivalent static loads method for structural optimization does not in general generate optimal designs. Struct Multidiscipl Optim 2018;58(1):139–54.

[38] Nocedal J, Wright SJ. Numerical optimization. 2nd ed. New York, NY, USA: Springer; 2006.

[39] Gill PE, Murray W, Wright MH. Practical optimization; 1981.