Karin Nachbagauer[1]
Faculty of Engineering
and Environmental Sciences,
University of Applied Sciences Upper Austria,
Stelzhamerstrasse 23,
Wels 4600, Austria
e-mail: karin.nachbagauer@fh-wels.at

Stefan Oberpeilsteiner
Faculty of Engineering
and Environmental Sciences,
University of Applied Sciences Upper Austria,
Stelzhamerstrasse 23,
Wels 4600, Austria
e-mail: stefan.oberpeilsteiner@fh-wels.at

Karim Sherif
Faculty of Engineering
and Environmental Sciences,
University of Applied Sciences Upper Austria,
Stelzhamerstrasse 23,
Wels 4600, Austria
e-mail: karim.sherif@fh-wels.at

Wolfgang Steiner
Faculty of Engineering
and Environmental Sciences,
University of Applied Sciences Upper Austria,
Stelzhamerstrasse 23,
Wels 4600, Austria
e-mail: wolfgang.steiner@fh-wels.at

The Use of the Adjoint Method for Solving Typical Optimization Problems in Multibody Dynamics

The present paper illustrates the potential of the adjoint method for a wide range of optimization problems in multibody dynamics such as inverse dynamics and parameter identification. Although the equations and matrices included show a complicated structure, the additional effort when combining the standard forward solver to the adjoint backward solver is kept in limits. Therefore, the adjoint method shows an efficient way to incorporate inverse dynamics to engineering multibody applications, e.g., trajectory tracking or parameter identification in the field of robotics. The present paper studies examples for both, parameter identification and optimal control, and shows the potential of the adjoint method in solving classical optimization problems in multibody dynamics.
[DOI: 10.1115/1.4028417]

1 Introduction

The adjoint method is probably the most efficient way to solve a variety of optimization problems in engineering sciences. Much attention to this approach has been paid recently in the context of continuous systems for sensitivity analysis (see, e.g., Refs. [1–4]). The class of dynamic programming methods for the computation of gradients in an optimization problem includes the adjoint method which has a long history in optimal control theory [5]. In the explanations of Giles and Pierce [6], the adjoint method is seen as a special case of linear duality, in which the dual problem has to solve only a single linear system. Obviously, huge benefits can be achieved by solving the dual formulation. The adjoint method utilizes this powerful aspect of duality to dramatically improve the efficiency of the computation.

The adjoint method is applied in, e.g., aerodynamic shape optimization by Jameson [7], in which the gradient of an objective function is determined indirectly by solving an adjoint equation which has coefficients determined by the solution of the multibody dynamics equations. This directly corresponds to the gradient technique for trajectory optimization pioneered by Bryson and Ho [8]. Once the gradient has been calculated, a descent method can be used to determine the optimal parameters or controls. The fast calculation of the gradients makes optimization computationally feasible even for designs in complex three-dimensional multibody systems. For this purpose, the equations of motion of the multibody system and adjoint equations may either be separately discretized from their representations as differential-algebraic equations (DAEs), or, alternatively, the equations of motion of the multibody system may be discretized first, and the discrete adjoint equations are then derived directly from the discrete multibody equations [8]. In the field of aerodynamic design optimization using the adjoint formulation, the works by Anderson and Venkatakrishnan [9] and Nadarajah and Jameson [10] have to be mentioned. The work of Oberai et al. [11] shows the solution of inverse problems in elasticity imaging using the adjoint method. There, a straightforward calculation of the gradient requires N solves of the forward elasticity problem. This cost is computationally prohibitive for typical values of N ($>=10^3$). To circumvent this difficulty, Ref. [11] presents a new algorithm based on the adjoint elasticity operator which requires only two solves (independent of N) to compute the gradient. Adjoint methods have also been the subject of studies in fluid dynamics research, as e.g., in the work by McNamara et al. [12], in which the adjoint method is used for controlling physics-based fluid simulations through gradient-based nonlinear optimization. Taylor et al. [13] present a hybrid adjoint approach applied to turbulent flow simulations.

Previous work on the adjoint method in multibody dynamics can be found in the work of Bottasso et al. [14], where the solution of inverse dynamics and trajectory optimization problems for multibody systems is reflected by an indirect approach combining optimal control theory with control and adjoint equations and transversality conditions. The inverse solution methodology presented in Ref. [14] has been implemented in the general purpose multibody code ADAMS. Within this implementation, some simple problems that provide a reasonable test and proof of concept of the presented methodology have been investigated [14]. The design of the indirect method for solving optimal control problems for multibody systems presented by Bertolazzi et al. [15] seems to be familiar with the idea of the adjoint method. The work by Schaffer [16] presents a numerical algorithm, the piecewise

[1]Corresponding author.

Contributed by the Design Engineering Division of ASME for publication in the Journal of Computational and Nonlinear Dynamics. Manuscript received May 22, 2014; final manuscript received August 19, 2014; published online April 9, 2015. Assoc. Editor: Dan Negrut.

adjoint method, which formulates the coordinate partitioning underlying ordinary differential equations as a boundary value problem, which is solved by multiple shooting methods. Additionally, convergence analysis of backward differentiation formulas (BDFs) is performed for stabilized differential-algebraic equations of motion in index 1 form and as well for the adjoint differential-algebraic equations for Cartesian noncentroidal multibody systems. Numerical studies in Ref. [16] compare the direct differentiation method, the adjoint method, and the piecewise adjoint method for a slider-crank mechanism and a high mobility wheeled vehicle which revealed the speed-up of multibody systems with a small number of degrees of freedom and the potential speed-up for larger problems are discussed as well.

The group around Petzold, Cao, Li, and Serban [17–19] describe forward and adjoint methods for sensitivity analysis for differential-algebraic equations and partial differential equations and state that the results of sensitivity analysis have wide-ranging applications in science and engineering, including model development, optimization, parameter estimation, model simplification, data assimilation, optimal control, uncertainty analysis, and experimental design [17]. In the work of Eberhard [4], the adjoint method is used for sensitivity analysis in multibody systems interpreted as a continuous, hybrid form of automatic differentiation.

In case of an orientation parameterization of a body in multibody dynamics without using angles, as e.g., described in the absolute nodal coordinate formulation (ANCF) [20], a gradient-based optimization approach using adjoint equations has been presented by Held and Seifried [21]. There, two different objective functions are defined to optimize a flexible slider-crank mechanism. One criterion accounts for the deformation energy of the flexible body, and the second criterion accumulates the squared deviation of the actual and desired position of the slider block. The adjoint method is then derived for the sensitivity analyses of the different objective functions [21]. Due to the structure of the objective functions and of the fact that the ANCF includes a constant mass matrix with vanishing derivative, the equations reduce to a simpler form [21, Eq. (19)]. The framework of the ANCF is as well used in the sensitivity analysis using the adjoint method by Pi et al. [22] within a first order approach, while in Ref. [23] a second order adjoint sensitivity analysis of multibody systems has been presented within the classical multibody formulation. The optimization strategies employing second order sensitivity information show higher accuracy, with the drawback of its complex structure. However, the adjoint method is utilized in Ref. [23] and shows for a large number of design variables dramatically decrease in computational costs, see Table 4 in Ref. [23] for a comparison of the adjoint method to the direct differentiation method.

Despite of the great potential of the adjoint method, in multibody dynamics the adjoint method is rarely applied, since the structure of the equations of motion is usually extremely complicated, in particular if flexible bodies are included, and the effort to obtain the set of adjoint equations seems tremendously high. Hence, dealing with the adjoint method is obviously unattractive for most developers of multibody simulation software, as far as they are familiar with it at all. The main goal of the present paper is to show how the adjoint method can be embedded efficiently to a multibody system described by a system of differential-algebraic equations of index 3 for optimal control problems or parameter identification applications. The body description is, in contrast to the approach in Ref. [21], in which the flexible multibody system is displayed in minimal coordinates, defined by the position of the center of mass and the four redundant Euler parameters for the orientation parameterization. This choice includes an internal constraint for each body which has to be considered also in the constraint Jacobian of the whole system and of course also influences the optimal control equations. The present paper shows the potential of the adjoint method for solving classical optimization problems in multibody dynamics and presents applications for optimal control problems and a parameter identification.

2 Adjoint Gradient Computation

In brief, the key idea of the adjoint method (see Refs. [4] and [17–19] for example) may be summarized as follows. Suppose we have the semi-explicit differential-algebraic system

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}, t, \lambda) \\ \mathbf{C}(\mathbf{x}) &= \mathbf{0} \\ \mathbf{x}(0) &= \mathbf{x}_0, \quad \text{satisfying } \mathbf{C}(\mathbf{x}_0) = \mathbf{0} \end{aligned} \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^{N_x}$ and $\lambda(t) \in \mathbb{R}^{N_\lambda}$ are the vectors of state variables and algebraic variables of a dynamical system such as a multibody system in a redundant formulation. Herein, $\mathbf{u} \in \mathbb{R}^{N_u}$ may either describe a vector of (constant) parameters or a vector of control signals. Our goal is to find $\mathbf{u}$ such that a functional of the form

$$J(\mathbf{u}) = \int_0^T h(\mathbf{x}, \mathbf{u}, t) dt \tag{2}$$

is minimized. For example, it might be of interest to determine a set of controls or parameters $\mathbf{u}$ in a way that the measured signals $\bar{s}_i(t), i = 1, \ldots, N_s$ are best approximated by the system outputs $s_i(\mathbf{x})$. In this case, we could define the root mean square (RMS) error as our functional to be minimized

$$J(\mathbf{u}) = \int_0^T \sum_{i=1}^{N_s} [s_i(\mathbf{x}) - \bar{s}_i(t)]^2 dt \tag{3}$$

Numerous methods are available to compute the argument for which a function or a functional attains a minimum. We just refer to the method of the steepest descent, the conjugate gradient method, the Gauss–Newton method, or quasi Newton methods like the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm when estimating the Hessian or the Davidon–Fletcher–Powell (DFP) algorithm when estimating the inverse of the Hessian. Some authors embed these methods in a homotopy continuation to obtain a global minimum [24]. In any cases, the gradient of $J(\mathbf{u})$ must be determined. For this purpose, several strategies can be pursued again. On the one hand, if $\mathbf{u}$ is a vector of N_u parameters, the sensitivity equations for $\mathbf{x}_\mathbf{u} = \partial \mathbf{x} / \partial \mathbf{u}$ are usually considered (see Refs. [24], [25], or [26] for example). The computational effort for this approach is equal to solving N_u linear sets of equations with the same dimension as Eq. (1). On the other hand, if $\mathbf{u}(t)$ is a vector of control signals, these signals are often discretized in order to transform the problem into a finite dimensional one. The adjoint method is a powerful alternative to compute the gradient of $J(\mathbf{u})$ in both cases.

First, we notice that $J(\mathbf{u})$ does not change if we add Eq. (1) to the integrand

$$J(\mathbf{u}) = \int_0^T \left[h + \mathbf{y}^\mathsf{T} (\mathbf{f} - \dot{\mathbf{x}}) + \mu^\mathsf{T} \mathbf{C} \right] dt \tag{4}$$

no matter how the functions $\mathbf{y}(t)$ and $\mu(t)$ are chosen. By introducing the Hamiltonian

$$H(\mathbf{x}, \mathbf{y}, \lambda, \mu, \mathbf{u}, t) = h(\mathbf{x}, \mathbf{u}, t) + \mathbf{y}^\mathsf{T} \mathbf{f}(\mathbf{x}, \lambda, \mathbf{u}, t) + \mu^\mathsf{T} \mathbf{C}(\mathbf{x}) \tag{5}$$

Equation (4) becomes

$$J(\mathbf{u}) = \int_0^T \left[H - \mathbf{y}^\mathsf{T} \dot{\mathbf{x}} \right] dt \tag{6}$$

Let us now consider a forward solution $\mathbf{x}(t)$ and $\lambda(t)$ of the system equations (1) for a set of parameters or control variables $\mathbf{u}$. A variation of $\mathbf{u}$ about $\delta \mathbf{u}$ will result in variations of $\mathbf{x}(t)$ and $\lambda(t)$ about $\delta \mathbf{x}(t)$ and $\delta \lambda(t)$ and, moreover, in a variation of the functional J about δJ. Up to a first order approximation δJ is given by

$$\delta J = \int_0^T \left[H_{\mathbf{u}} \delta \mathbf{u} + H_{\mathbf{x}} \delta \mathbf{x} + H_{\lambda} \delta \lambda - \mathbf{y}^{\mathsf{T}} \delta \dot{\mathbf{x}} \right] dt \tag{7}$$

where $H_{\mathbf{u}}, H_{\mathbf{x}}$, and H_{λ} denote the row vectors with the partial derivatives of H with respect to the components of $\mathbf{u}$, $\mathbf{x}$, and λ, e.g.,

$$H_{\mathbf{x}} = \left[\frac{\partial H}{\partial x_1} \quad \cdots \quad \frac{\partial H}{\partial x_n} \right] \tag{8}$$

Integration by parts in the last term of Eq. (7) yields

$$\begin{aligned} \delta J &= \int_0^T \left[H_{\mathbf{u}} \delta \mathbf{u} + H_{\mathbf{x}} \delta \mathbf{x} + H_{\lambda} \delta \lambda + \dot{\mathbf{y}}^{\mathsf{T}} \delta \mathbf{x} \right] dt - \mathbf{y}^{\mathsf{T}} \delta \mathbf{x} |_0^T \\ &= \int_0^T \left[H_{\mathbf{u}} \delta \mathbf{u} + \left(H_{\mathbf{x}} + \dot{\mathbf{y}}^{\mathsf{T}} \right) \delta \mathbf{x} + H_{\lambda} \delta \lambda \right] dt - \mathbf{y}^{\mathsf{T}} \delta \mathbf{x} |_T \end{aligned} \tag{9}$$

since $\delta \mathbf{x}(0) = 0$ as the initial state is assumed to be prescribed. It is the key idea of the adjoint method that the computation of $\delta \mathbf{x}$ and $\delta \lambda$ can be circumvented if the adjoint variables $\mathbf{y}(t)$ and $\mu(t)$ are chosen such that

$$\dot{\mathbf{y}}^{\mathsf{T}} = -H_{\mathbf{x}}, \quad H_{\lambda} = 0 \quad \text{and} \quad \mathbf{y}(T) = 0 \tag{10}$$

or including an other known fixed value for the boundary condition $\mathbf{y}(T)$. This set of semi-explicit differential-algebraic equations is called the adjoint system of Eq. (1). It may be solved backwards in time starting at time $t = T$, once the original equations have been solved forward for $t \in [0, T]$. Then, with $\mathbf{x}(t), \lambda(t), \mathbf{y}(t)$, and $\mu(t)$ from Eqs. (1) and (10), the variation of J according to Eq. (9) is readily given by

$$\delta J = \int_0^T H_{\mathbf{u}} \delta \mathbf{u} dt \tag{11}$$

If $\mathbf{u}$ is a control signal, the largest possible increase of δJ is obtained, if $\delta \mathbf{u}(t)$ is chosen in the direction of $H_{\mathbf{u}}^{\mathsf{T}}$. Hence, $H_{\mathbf{u}}^{\mathsf{T}}$ may be considered as the gradient of the functional $J(\mathbf{u})$.

Moreover, if $\mathbf{u}$ is a vector of parameters, Eq. (11) becomes

$$\delta J = \left(\int_0^T H_{\mathbf{u}} dt \right) \delta \mathbf{u} = \nabla J^{\mathsf{T}} \delta \mathbf{u} \tag{12}$$

and, hence, the gradient of the function $J(\mathbf{u})$ is given by $\nabla J = \int_0^T H_{\mathbf{u}}^{\mathsf{T}} dt$. To find a minimum of J, we could simply walk a short distance along the negative gradient of J. In this case, we must choose a (small) number $\kappa > 0$ and determine the increment of $\mathbf{u}$ from

$$\delta \mathbf{u}(t) = -\kappa H_{\mathbf{u}}^{\mathsf{T}} \quad \text{(control optimization)} \tag{13}$$

or from

$$\delta \mathbf{u} = -\kappa \int_0^T H_{\mathbf{u}}^{\mathsf{T}} dt \quad \text{(parameter optimization)} \tag{14}$$

If κ is sufficiently small, the updated control/parameter $\mathbf{u} + \delta \mathbf{u}$ will always reduce J.

Note that only two systems of DAEs must be integrated for computing the direction of the gradient. The main difficulty with multibody dynamics results from the complexity of the original system (1). Hence, many authors focused on two-dimensional examples or rather general aspects (e.g., Refs. [25], [27], and [28]). However, based on highly redundant formulations the adjoint equations (10) for a multibody system are relatively simple. In Sec. 3, we show how these equations can be implemented in a MBS-code under the assumption that the rigid body rotations are described by Euler parameters.

3 The Adjoint Method in Multibody Dynamics

In this section, we discuss the application of the adjoint gradient computation to multibody dynamics. First, we derive the adjoint equations for the equations of motion of a multibody system and discuss the structure of the boundary conditions, in particular, if an endpoint term is included in the objective function. Second, we present a new time discretization method for solving the adjoint equations numerically. Finally, we sum up the complete process for the adjoint gradient computation.

3.1 Adjoint Equations. A mechanical system consisting of rigid and flexible bodies, forces and constraints acting between these bodies can be described by equations of motion in the following form:

$$\begin{aligned} \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} &= \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}) - \mathbf{C}_{\mathbf{q}}^{\mathsf{T}}(\mathbf{q}) \lambda \\ \mathbf{C}(\mathbf{q}) &= \mathbf{0} \end{aligned} \tag{15}$$

Here, $\mathbf{q}$ denotes a vector of redundant generalized coordinates. They are subject to the holonomic constraints $\mathbf{C}(\mathbf{q}) = \mathbf{0}$, which enter the equations of motion via the Jacobian $\mathbf{C}_{\mathbf{q}}$ and the vector of Lagrange multipliers λ representing the constraint forces in the system. Moreover, the system may incorporate a control $\mathbf{u}(t)$ or a vector of parameters $\mathbf{u}$. For simplicity, we suppose that $\mathbf{u}$ appears only in the vector $\mathbf{f}$, which is true, if $\mathbf{u}$ is an actuating force, a stiffness, or a damping parameter. However, the subsequent derivations may be extended to the case, where $\mathbf{u}$ appears in the constraint equations, too. (This might be the case, if $\mathbf{u}$ is a parameter to describe the position or orientation of a joint or $\mathbf{u}$ is included in the initial conditions.)

Using the additional variables $\mathbf{v} = \dot{\mathbf{q}}$, the equations of motion can be reformulated as a first order system of equations reading

$$\dot{\mathbf{q}} = \mathbf{v} \tag{16}$$

$$\mathbf{M} \dot{\mathbf{v}} = \mathbf{f}(\mathbf{q}, \mathbf{v}, \mathbf{u}) - \mathbf{C}_{\mathbf{q}}^{\mathsf{T}} \lambda \tag{17}$$

$$\mathbf{C}(\mathbf{q}) = \mathbf{0} \tag{18}$$

The state vector $\mathbf{x}$ introduced in Sec. 2 now consists of $\mathbf{q}$ and $\mathbf{v}$. It is our goal to determine the parameter/control which minimizes a functional of the form

$$J = \int_0^T h(\mathbf{q}, \mathbf{v}, \mathbf{u}) dt + S(\mathbf{q}, \mathbf{v})|_T \tag{19}$$

in which $S(\mathbf{q}, \mathbf{v})|_T$ introduces an additional end point term, usually denoted as scrap function. For example, S could describe an end point RMS-error of a sensor variable similar to Eq. (3). In this means, a scrap function may also be used to replace possible boundary conditions for the state variables at $t = T$ by an optimization condition.

We want to derive a gradient formula like Eqs. (13) and (14) for the multibody system now. Without changing the functional, we may therefore augment J by the system equations in the following way:

$$\begin{aligned} J = \int_0^T \left[h(\mathbf{q}, \mathbf{v}, \mathbf{u}, t) + \mathbf{p}^{\mathsf{T}} (\dot{\mathbf{q}} - \mathbf{v}) + \mathbf{w}^{\mathsf{T}} (\mathbf{M} \dot{\mathbf{v}} - \mathbf{f} + \mathbf{C}_{\mathbf{q}}^{\mathsf{T}} \lambda) + \mu^{\mathsf{T}} \mathbf{C} \right] dt \\ + S(\mathbf{q}, \mathbf{v})|_{t=T} \end{aligned} \tag{20}$$

At this point, the variables $\mathbf{p}(t)$, $\mathbf{w}(t)$, and $\mu(t)$ may be chosen arbitrarily. The variation of the functional J is given by

$$\delta J = \int_0^T \Big\{ h_{\mathbf{q}}\delta\mathbf{q} + h_{\mathbf{v}}\delta\mathbf{v} + h_{\mathbf{u}}\delta\mathbf{u} + \mathbf{p}^{\mathsf{T}}(\delta\dot{\mathbf{q}} - \delta\mathbf{v}) + \mathbf{w}^{\mathsf{T}}\Big[(\mathbf{M}\dot{\mathbf{v}})_{\mathbf{q}}\delta\mathbf{q} + \mathbf{M}\delta\dot{\mathbf{v}} - \mathbf{f}_{\mathbf{q}}\delta\mathbf{q} - \mathbf{f}_{\mathbf{v}}\delta\mathbf{v} - \mathbf{f}_{\mathbf{u}}\delta\mathbf{u} + \left(\mathbf{C}_{\mathbf{q}}^{\mathsf{T}}\lambda\right)_{\mathbf{q}}\delta\mathbf{q} + \mathbf{C}_{\mathbf{q}}^{\mathsf{T}}\delta\lambda\Big] + \mu^{\mathsf{T}}\mathbf{C}_{\mathbf{q}}\delta\mathbf{q}\Big\}dt + S_{\mathbf{q}}\delta\mathbf{q}|_{t=T} + S_{\mathbf{v}}\delta\mathbf{v}|_{t=T} \tag{21}$$

Integration by parts for the terms including $\delta\dot{\mathbf{q}}$ and $\delta\dot{\mathbf{v}}$ is computed by

$$\int_0^T \mathbf{p}^{\mathsf{T}}\delta\dot{\mathbf{q}}dt = -\int_0^T \dot{\mathbf{p}}^{\mathsf{T}}\delta\mathbf{q}dt + \mathbf{p}^{\mathsf{T}}\delta\mathbf{q}|_{t=T}$$
$$\int_0^T \mathbf{w}^{\mathsf{T}}\mathbf{M}\delta\dot{\mathbf{v}}dt = -\int_0^T \frac{d}{dt}(\mathbf{w}^{\mathsf{T}}\mathbf{M})\delta\mathbf{v}dt + \mathbf{w}^{\mathsf{T}}\mathbf{M}\delta\mathbf{v}|_{t=T} \tag{22}$$

with $\delta\mathbf{q}(0) = \delta\mathbf{v}(0) = 0$ since initial conditions for the adjoint variables are prescribed as $\mathbf{q}(0) = \mathbf{q}_0$ and $\mathbf{v}(0) = \mathbf{v}_0$. Using Eq. (22) and collecting the terms multiplied with $\delta\mathbf{q}$, $\delta\mathbf{v}$, $\delta\lambda$, and $\delta\mathbf{u}$, the variation of the functional J given by Eq. (19) can be rewritten as

$$\delta J = \int_0^T \Big\{ \Big[h_{\mathbf{q}} - \dot{\mathbf{p}}^{\mathsf{T}} + \mathbf{w}^{\mathsf{T}}\Big((\mathbf{M}\dot{\mathbf{v}})_{\mathbf{q}} - \mathbf{f}_{\mathbf{q}} + \left(\mathbf{C}_{\mathbf{q}}^{\mathsf{T}}\lambda\right)_{\mathbf{q}}\Big) + \mu^{\mathsf{T}}\mathbf{C}_{\mathbf{q}}\Big]\delta\mathbf{q} + \Big[h_{\mathbf{v}} - \mathbf{p}^{\mathsf{T}} - \mathbf{w}^{\mathsf{T}}\mathbf{f}_{\mathbf{v}} - \frac{d}{dt}(\mathbf{w}^{\mathsf{T}}\mathbf{M})\Big]\delta\mathbf{v} + \Big[\mathbf{w}^{\mathsf{T}}\mathbf{C}_{\mathbf{q}}^{\mathsf{T}}\Big]\delta\lambda + \big[h_{\mathbf{u}} - \mathbf{w}^{\mathsf{T}}\mathbf{f}_{\mathbf{u}}\big]\delta\mathbf{u}\Big\}dt + \big[S_{\mathbf{q}} + \mathbf{p}^{\mathsf{T}}\big]\delta\mathbf{q}|_{t=T} + \big[S_{\mathbf{v}} + \mathbf{w}^{\mathsf{T}}\mathbf{M}\big]\delta\mathbf{v}|_{t=T} \tag{23}$$

To eliminate the terms involving $\delta\mathbf{q}$, $\delta\mathbf{v}$ the adjoint variables $\mathbf{p}$, $\mathbf{w}$, and μ may now be defined by equating the respective expressions in square brackets to zero. After transposing these expressions, one obtains the following system of adjoint equations:

$$\frac{d\mathbf{p}}{dt} = h_{\mathbf{q}}^{\mathsf{T}} + \mathbf{A}\mathbf{w} + \mathbf{C}_{\mathbf{q}}^{\mathsf{T}}\mu \tag{24}$$

$$\frac{d}{dt}(\mathbf{M}\mathbf{w}) = h_{\mathbf{v}}^{\mathsf{T}} - \mathbf{p} - \mathbf{B}\mathbf{w} \tag{25}$$

$$\mathbf{0} = \mathbf{C}_{\mathbf{q}}\mathbf{w} \tag{26}$$

$$\mathbf{0} = S_{\mathbf{q}}^{\mathsf{T}} + \mathbf{p}(T) \tag{27}$$

$$\mathbf{0} = S_{\mathbf{v}}^{\mathsf{T}} + \mathbf{M}(T)\mathbf{w}(T) \tag{28}$$

where the abbreviations

$$\mathbf{A} = (\mathbf{M}\dot{\mathbf{v}})_{\mathbf{q}}^{\mathsf{T}} - \mathbf{f}_{\mathbf{q}}^{\mathsf{T}} + (\mathbf{C}_{\mathbf{q}}^{\mathsf{T}}\lambda)_{\mathbf{q}}^{\mathsf{T}} \tag{29}$$

$$\mathbf{B} = \mathbf{f}_{\mathbf{v}}^{\mathsf{T}} \tag{30}$$

and the symmetry of the mass matrix $\mathbf{M} = \mathbf{M}^{\mathsf{T}}$ have been used. If Eqs. (24)–(28) are satisfied Eq. (23) reduces to

$$\delta J = \int_0^T \big[h_{\mathbf{u}} - \mathbf{w}^{\mathsf{T}}\mathbf{f}_{\mathbf{u}}\big]\delta\mathbf{u}dt \tag{31}$$

which directly relates the independent variation $\delta\mathbf{u}$ to the variation of the objective function. Hence, we may conclude that the gradient of J is given by $h_{\mathbf{u}}^{\mathsf{T}} - \mathbf{f}_{\mathbf{u}}^{\mathsf{T}}\mathbf{w}$ if $\mathbf{u}$ is a control or by $\int_0^T [h_{\mathbf{u}}^{\mathsf{T}} - \mathbf{f}_{\mathbf{u}}^{\mathsf{T}}\mathbf{w}]dt$ if $\mathbf{u}$ denotes a parameter.

3.2 Consistent Boundary Conditions for the Adjoint System. One may have noticed that the boundary condition (28) for the adjoint variable $\mathbf{w}$ is generally incompatible with the constraint equation (26) at $t = T$. Only when $S_{\mathbf{v}} = 0$, i.e., when the scrap function does not depend on $\mathbf{v}$, all equations are satisfied by setting $\mathbf{p}(T) = -S_{\mathbf{q}}^{\mathsf{T}}$ and $\mathbf{w}(T) = \mathbf{0}$. However, we run into trouble if $S_{\mathbf{v}} \neq 0$. To circumvent this problem, we apply an approach similar to the idea of Gear–Gupta–Leimkuhler [29] and consider the constraint equations $\mathbf{C}(\mathbf{q}) = \mathbf{0}$ of the original system Eq. (15) at velocity level yielding

$$\mathbf{C}_{\mathbf{q}}\dot{\mathbf{q}} = \mathbf{C}_{\mathbf{q}}(\mathbf{q})\mathbf{v} = \mathbf{0} \tag{32}$$

Its variation with respect to $\mathbf{v}$ and $\mathbf{q}$ results in

$$\mathbf{C}_{\mathbf{q}}\delta\mathbf{v} + \left(\mathbf{C}_{\mathbf{q}}\mathbf{v}\right)_{\mathbf{q}}\delta\mathbf{q} = \mathbf{0} \tag{33}$$

We may therefore consider this relation at time $t = T$ and multiply it with an arbitrary vector of numbers ξ to obtain

$$\xi^{\mathsf{T}}\Big(\mathbf{C}_{\mathbf{q}}\delta\mathbf{v} + \left(\mathbf{C}_{\mathbf{q}}\mathbf{v}\right)_{\mathbf{q}}\delta\mathbf{q}\Big)\Big|_T = \mathbf{0} \tag{34}$$

Since the expression on the left side is always zero we may add it to Eq. (23) without modifying the variation of our objective functional. Hereby, we end up with two additional boundary terms in Eq. (23) transforming Eqs. (27) and (28) into

$$\mathbf{S}_{\mathbf{q}}^{\mathsf{T}} + \mathbf{p} + \left(\mathbf{C}_{\mathbf{q}}\mathbf{v}\right)_{\mathbf{q}}^{\mathsf{T}}\xi = \mathbf{0} \quad \ldots\text{at } t = T \tag{35}$$

$$\mathbf{S}_{\mathbf{v}}^{\mathsf{T}} + \mathbf{M}\mathbf{w} + \mathbf{C}_{\mathbf{q}}^{\mathsf{T}}\xi = \mathbf{0} \quad \ldots\text{at } t = T \tag{36}$$

By involving the still undetermined variable ξ we can now compute $\mathbf{p}(T)$, $\mathbf{w}(T)$ such that also the constraint equation (26) is satisfied at $t = T$. For that purpose, the following system of equations has to be solved for $\mathbf{p}(T)$, $\mathbf{w}(T)$, and ξ:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \left(\mathbf{C}_{\mathbf{q}}\mathbf{v}\right)_{\mathbf{q}}^{\mathsf{T}} \\ \mathbf{0} & \mathbf{M} & \mathbf{C}_{\mathbf{q}}^{\mathsf{T}} \\ \mathbf{0} & \mathbf{C}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{w} \\ \xi \end{bmatrix} = \begin{bmatrix} -\mathbf{S}_{\mathbf{q}}^{\mathsf{T}} \\ -\mathbf{S}_{\mathbf{v}}^{\mathsf{T}} \\ \mathbf{0} \end{bmatrix} \quad \ldots\text{at } t = T \tag{37}$$

In practice, one may first solve

$$\begin{bmatrix} \mathbf{M} & \mathbf{C}_{\mathbf{q}}^{\mathsf{T}} \\ \mathbf{C}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \xi \end{bmatrix} = \begin{bmatrix} -\mathbf{S}_{\mathbf{v}}^{\mathsf{T}} \\ \mathbf{0} \end{bmatrix} \quad \ldots\text{at } t = T \tag{38}$$

yielding ξ and $\mathbf{w}(T)$, and compute then $\mathbf{p}(T)$ from

$$\mathbf{p} = -\left(\mathbf{C}_{\mathbf{q}}\mathbf{v}\right)_{\mathbf{q}}^{\mathsf{T}}\xi - \mathbf{S}_{\mathbf{q}}^{\mathsf{T}} \quad \ldots\text{at } t = T \tag{39}$$

Once $\mathbf{w}(T)$ and $\mathbf{p}(T)$ has been determined in this way, the differential-algebraic set of adjoint equations (24)–(26) may be solved to obtain $\mathbf{w}$, $\mathbf{p}$, and μ at every time instant in the interval $[0, T]$.

3.3 Remarks on the Computation of the Jacobian Matrices A and B. For the adjoint equations of a multibody system, the matrices $\mathbf{M}$, $\mathbf{C}_{\mathbf{q}}$, $\mathbf{A}$, and $\mathbf{B}$ from Eqs. (29) and (30) are required along a forward simulation of the equations of motion. Whereas $\mathbf{M}$, $\mathbf{C}_{\mathbf{q}}$ also appear in the equations of motion and are therefore available, the determination of the Jacobian matrices $\mathbf{A}$ and $\mathbf{B}$ requires additional computational effort. Basically, the Jacobians can be computed in three ways: First, the derivatives are computed exactly by implementing explicit formulas in the MBS-code. Second, one may compute the derivatives numerically by substituting the derivatives by finite difference quotients. Finally, the derivatives could also be determined by the technique of automatic differentiation, see, e.g., Ref. [4]. For complex multibody systems, the first way seems expensive and susceptible to

programming errors. However, if a highly redundant formulation of the equations of motion is used, this way becomes attractive. We therefore recommend to assign full rotational and translational degrees of freedom to every body of the system and to describe the kinematic coupling between the bodies by constraint equations. Moreover, the use of Euler parameters for the rotational motions simplifies the system matrices such that programming explicit formulas for $\mathbf{A}$ and $\mathbf{B}$ become the most efficient strategy. It should also be noted that the Jacobian matrices $(\mathbf{M}\dot{\mathbf{v}})_{\mathbf{q}}, \mathbf{f}_{\mathbf{q}}, \mathbf{f}_{\mathbf{v}}$, and $(\mathbf{C}_{\mathbf{q}}^{\mathrm{T}}\lambda)_{\mathbf{q}}$ may be required already for the simulation of the multibody system if an implicit integration scheme such as, e.g., the Hilbert–Hughes–Taylor (HHT)-algorithm [30,31] is applied. Hence, an efficient computation of these matrices is crucial also for solving the equations of motion in forward direction.

3.4 A Backward Differentiation Scheme for the Adjoint System. Since Eqs. (24)–(26) must be solved backwards in the physical time $t \in [0, T]$, it is advantageous to introduce a new time coordinate τ running also from $\tau = 0$ to $\tau = T$, before a time discretization scheme is developed. The transformation

$$\tau = T - t, \quad \tau \in [0, T], \quad \frac{d}{dt} = \frac{d}{d\tau}\frac{d\tau}{dt} = -\frac{d}{d\tau} \tag{40}$$

converts the adjoint equations (24)–(26) into

$$\begin{aligned} \frac{d\mathbf{p}}{d\tau} &= -h_{\mathbf{q}}^{\mathrm{T}}(\tau) - \mathbf{A}(\tau)\mathbf{w} - \mathbf{C}_{\mathbf{q}}^{\mathrm{T}}(\tau)\mu \\ \frac{d}{d\tau}(\mathbf{M}(\tau)\mathbf{w}) &= -h_{\mathbf{v}}^{\mathrm{T}}(\tau) + \mathbf{p} + \mathbf{B}(\tau)\mathbf{w} \\ \mathbf{0} &= \mathbf{C}_{\mathbf{q}}(\tau)\mathbf{w} \end{aligned} \tag{41}$$

where $\mathbf{M}(\tau), \mathbf{A}(\tau), \mathbf{B}(\tau), \mathbf{C}_{\mathbf{q}}(\tau), h_{\mathbf{q}}(\tau)$, and $h_{\mathbf{v}}(\tau)$ have to be computed for $\mathbf{q}(\tau) = \mathbf{q}(T - t), \mathbf{v}(\tau) = \mathbf{v}(T - t)$ and $\lambda(\tau) = \lambda(T - t)$, resulting from a forward simulation of the equations of motion (15). The boundary condition for $\mathbf{p}$ and $\mathbf{w}$ is now given by Eqs. (38) and (39) at $\tau = 0$.

For the numerical solution of Eq. (41) at the time instances $\tau_n = n\gamma$, $n = 1,\ldots, N_t$, $\gamma = T/N_t$, we propose a backward differentiation scheme which approximates the derivative of a function $F(\tau)$ at a time instant τ_n by using the function values at τ_n, $\tau_{n-1},\ldots,$ τ_{n-k}. The BDF reads

$$\left.\frac{dF}{d\tau}\right|_{\tau_n} \approx \frac{1}{\gamma}\sum_{i=0}^{k}\alpha_i F(\tau_{n-i}) \tag{42}$$

The coefficients α_i result from differentiating an interpolation polynomial through $F(\tau_n),\ldots, F(\tau_{n-k})$ and are chosen as the standard coefficients presented, e.g., in Ref. [32, p. 349]. Considering Eq. (41) at $\tau = \tau_n$ and inserting the BDF-approximation for $d\mathbf{p}/d\tau$ and $d(\mathbf{Mw})/d\tau$, the following set of algebraic equations for $\mathbf{p}(\tau_n)$, $\mathbf{v}(\tau_n)$, and $\lambda(\tau_n)$ is obtained:

$$\frac{1}{\gamma}\left(\alpha_0\mathbf{p}(\tau_n) + \sum_{i=1}^{k}\alpha_i\mathbf{p}(\tau_{n-i})\right) = -h_{\mathbf{q}}^{\mathrm{T}}(\tau_n) - \mathbf{A}(\tau_n)\mathbf{w}(\tau_n) - \mathbf{C}_{\mathbf{q}}^{\mathrm{T}}(\tau_n)\mu(\tau_n) \tag{43}$$

$$\frac{1}{\gamma}\left(\alpha_0\mathbf{M}(\tau_n)\mathbf{w}(\tau_n) + \sum_{i=1}^{k}\alpha_i\mathbf{M}(\tau_{n-i})\mathbf{w}(\tau_{n-i})\right) = -h_{\mathbf{v}}^{\mathrm{T}}(\tau_n) + \mathbf{p}(\tau_n) + \mathbf{B}(\tau_n)\mathbf{w}(\tau_n) \tag{44}$$

$$\mathbf{C}_{\mathbf{q}}(\tau_n)\mathbf{w}(\tau_n) = \mathbf{0} \tag{45}$$

where $\mathbf{p}(\tau_{n-i})$, $\mathbf{v}(\tau_{n-i})$, and $\lambda(\tau_{n-i})$ is supposed to be known for $i > 0$ from previous integration steps. Notice that it is not necessary to differentiate the term $d(\mathbf{Mw})/d\tau$ by using the product rule, which would require the additional term $d\mathbf{M}/d\tau$ to be computed from the forward simulation.

Since the adjoint system (41) is linear in $\mathbf{p}$, $\mathbf{v}$, and λ, the discretized system given by Eqs. (43)–(45) is also linear in $\mathbf{p}(\tau_n)$, $\mathbf{v}(\tau_n)$, and $\lambda(\tau_n)$. Moreover, $\mathbf{p}(\tau_n)$ may be eliminated by solving Eq. (43) for

$$\mathbf{p}(\tau_n) = -\frac{\gamma}{\alpha_0}\left(h_{\mathbf{q}}^{\mathrm{T}}(\tau_n) + \mathbf{A}(\tau_n)\mathbf{w}(\tau_n) + \mathbf{C}_{\mathbf{q}}^{\mathrm{T}}(\tau_n)\mu(\tau_n)\right) - \frac{1}{\alpha_0}\sum_{i=1}^{k}\alpha_i\mathbf{p}(\tau_{n-i}) \tag{46}$$

Inserting into Eq. (44) yields

$$\begin{aligned} &\frac{1}{\gamma}\left(\alpha_0\mathbf{M}(\tau_n)\mathbf{w}(\tau_n) + \sum_{i=1}^{k}\alpha_i\mathbf{M}(\tau_{n-i})\mathbf{w}(\tau_{n-i})\right) \\ &= -h_{\mathbf{v}}^{\mathrm{T}}(\tau_n) - \frac{\gamma}{\alpha_0}\left(h_{\mathbf{q}}^{\mathrm{T}}(\tau_n) + \mathbf{A}(\tau_n)\mathbf{w}(\tau_n) + \mathbf{C}_{\mathbf{q}}^{\mathrm{T}}(\tau_n)\mu(\tau_n)\right) \\ &\quad - \frac{1}{\alpha_0}\sum_{i=1}^{k}\alpha_i\mathbf{p}(\tau_{n-i}) + \mathbf{B}(\tau_n)\mathbf{w}(\tau_n) \end{aligned} \tag{47}$$

or after rearranging and multiplying with $\alpha_0\gamma$

$$\begin{aligned} &\left(\alpha_0^2\mathbf{M}(\tau_n) + \gamma^2\mathbf{A}(\tau_n) - \alpha_0\gamma\mathbf{B}(\tau_n)\right)\mathbf{w}(\tau_n) + \gamma^2\mathbf{C}_{\mathbf{q}}^{\mathrm{T}}(\tau_n)\mu(\tau_n) \\ &= -\alpha_0\gamma h_{\mathbf{v}}^{\mathrm{T}}(\tau_n) - \gamma^2 h_{\mathbf{q}}^{\mathrm{T}}(\tau_n) - \sum_{i=1}^{k}\alpha_i\left(\gamma\mathbf{p}(\tau_{n-i}) + \alpha_0\mathbf{M}(\tau_{n-i})\mathbf{w}(\tau_{n-i})\right) \end{aligned} \tag{48}$$

Using the abbreviations

$$\mathbf{W}(\tau_n) = \alpha_0^2\mathbf{M}(\tau_n) + \gamma^2\mathbf{A}(\tau_n) - \alpha_0\gamma\mathbf{B}(\tau_n) \tag{49}$$

$$\mathbf{r}(\tau_n) = -\alpha_0\gamma h_{\mathbf{v}}^{\mathrm{T}}(\tau_n) - \gamma^2 h_{\mathbf{q}}^{\mathrm{T}}(\tau_n) - \sum_{i=1}^{k}\alpha_i\left(\gamma\mathbf{p}(\tau_{n-i}) + \alpha_0\mathbf{M}(\tau_{n-i})\mathbf{w}(\tau_{n-i})\right) \tag{50}$$

Equations (48) and (45) may be summarized in the following matrix equation for $\mathbf{w}(\tau_n)$ and $\mu(\tau_n)$:

$$\begin{bmatrix} \mathbf{W}(\tau_n) & \gamma^2\mathbf{C}_{\mathbf{q}}^{\mathrm{T}}(\tau_n) \\ \mathbf{C}_{\mathbf{q}}(\tau_n) & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{w}(\tau_n) \\ \mu(\tau_n) \end{bmatrix} = \begin{bmatrix} \mathbf{r}(\tau_n) \\ \mathbf{0} \end{bmatrix} \tag{51}$$

After solving this equation, the second adjoint variable $\mathbf{p}(\tau_n)$ may be computed from Eq. (46). The integration process can be started by choosing the integration order $k = 1$ at first.

3.5 Summary: The Adjoint Gradient Computation. We finally summarize the steps to compute the gradient of the objective function J with the adjoint method. Let $\mathbf{u}$ denote a given vector of controls or parameters of a multibody system. The following process describes how one obtains the direction of a control or parameter increment $\delta\mathbf{u}$ which causes the maximum decrease of J (with respect to a first order approximation of J).

(1) Solve the equations of motion Eq. (15) forward in time $t \in [0, T]$ yielding $\mathbf{q}(t)$, $\mathbf{v}(t)$, and $\lambda(t)$. This may be done, e.g., by choosing the HHT integration scheme, as proposed in Ref. [30] and its application for a differential algebraic system given in an index three formulation in Ref. [31].
(2) Just for information, we may then already compute the objective function J by inserting $\mathbf{q}(t)$, $\mathbf{v}(t)$, and $\mathbf{u}(t)$ into Eq. (19). Note that the integration must be done numerically.
(3) Along the forward simulation of the equations of motion compute the mass matrix $\mathbf{M}(t)$, the constraint Jacobian

$\mathbf{C_q}(t)$, and the Jacobian matrices $\mathbf{A}(t)$, $\mathbf{B}(t)$ from Eqs. (29) and (30). For lack of computer memory, it might be impossible to store these matrices at a sufficiently high number of time instances. In this case, one must provide formulas to compute these matrices from $\mathbf{q}$, $\mathbf{v}$, and $\mathbf{u}$ on demand.

(4) Determine the consistent boundary conditions at $\tau = 0$ for the adjoint variables $\mathbf{w}$ and $\mathbf{p}$ from Eqs. (38) and (39). Note that the additional Jacobian $\left(\mathbf{C_q v}\right)^{\mathsf{T}}_{\mathbf{q}}$ must be computed at $t = T$ for that purpose.

(5) Solve the adjoint equations (41) for $\mathbf{p}(\tau)$, $\mathbf{w}(\tau)$, and $\mu(\tau)$, where $\tau = T - t$. A numerical solution at time instances τ_n can be computed from Eqs. (51) and (46), where the integration order k must be chosen equal to one in the first step and may be increased afterwards.

(6) Compute the adjoint variables as functions of the original time by setting $\mathbf{p}(t) = \mathbf{p}(\tau = T - t)$ and $\mathbf{w}(t) = \mathbf{w}(\tau = T - t)$. Moreover, determine $h_{\mathbf{u}}(t)$ and $\mathbf{f_u}(t)$ along the forward simulation.

(7) From Eq. (31), one may finally derive the direction of the steepest descent. If $\mathbf{u}$ is a vector of parameters, set

$$\delta\mathbf{u} = -\kappa \int_0^T \left(h_{\mathbf{u}}^{\mathsf{T}} - \mathbf{f}_{\mathbf{u}}^{\mathsf{T}}\mathbf{w}\right) dt \tag{52}$$

If $\mathbf{u}$ is a vector of controls, set

$$\delta\mathbf{u}(t) = -\kappa\left(h_{\mathbf{u}}^{\mathsf{T}}(t) - \mathbf{f}_{\mathbf{u}}^{\mathsf{T}}(t)\mathbf{w}(t)\right) \tag{53}$$

A sufficiently small number $\kappa > 0$ has to be chosen for that purpose.

4 Optimization Strategies

Based on the adjoint gradient computation outlined above, we may now search for a control or a parameter $\mathbf{u}$ which minimizes the objective function J. Generally speaking, we can pursue two different strategies: First, we could always walk a certain distance along the gradient until we end up in a local minimum of J (gradient method). Second, we could solve the problem of finding $\mathbf{u}$ such that the gradient becomes zero by applying a quasi-Newton method. We describe both approaches in this section.

4.1 The Gradient Method. The gradient method uses directly Eq. (52) or Eq. (53) to compute an update of $\mathbf{u}$. The resulting change δJ of the objective function can be evaluated by inserting Eq. (52) or Eq. (53) into Eq. (31) yielding

$$\delta J = -\kappa \left| \int_0^T \left(h_{\mathbf{u}}^{\mathsf{T}} - \mathbf{f}_{\mathbf{u}}^{\mathsf{T}}\mathbf{w}\right) dt \right|^2 \quad \text{if } \mathbf{u} \text{ is a parameter} \tag{54}$$

or, respectively,

$$\delta J = -\kappa \int_0^T \left| h_{\mathbf{u}}^{\mathsf{T}} - \mathbf{f}_{\mathbf{u}}^{\mathsf{T}}\mathbf{w}\right|^2 dt \quad \text{if } \mathbf{u} \text{ is a control} \tag{55}$$

We notice that $\delta J \leq 0$ in both cases if $\kappa > 0$. Hence, starting from an initial guess for $\mathbf{u}$, the method should converge to a local minimum of J after several iteration steps. However, Eqs. (54) and (55) only hold for small increments $\delta\mathbf{u}$, i.e., for a small number κ, since Eq. (31) was obtained from linearizing the objective function J at $\mathbf{u}$. Therefore, the key problem is to choose the factor κ appropriately. On the one hand, if κ is very small, the objective function decreases reliably after each iteration step, but the convergence rate will be very small, too. On the other hand, if κ is a larger number, the updated control or parameter $\mathbf{u} + \delta\mathbf{u}$ will cause an unpredictable change of J, which is not determined by Eqs. (54) and (55).

Basically, there are two ways to choose κ [33]:

(1) One may prescribe that the change δJ becomes a certain fraction ε of J, i.e., $|\delta J| = \varepsilon |J|$. For example, a 5% change of J would result from setting $\varepsilon = 0.05$. The factor κ is then given by

$$\kappa = \varepsilon |J| \left| \int_0^T \left(h_{\mathbf{u}}^{\mathsf{T}} - \mathbf{f}_{\mathbf{u}}^{\mathsf{T}}\mathbf{w}\right) dt \right|^{-2} \quad \text{if } \mathbf{u} \text{ is a parameter} \tag{56}$$

or, respectively,

$$\kappa = \varepsilon |J| \left(\int_0^T \left| h_{\mathbf{u}}^{\mathsf{T}} - \mathbf{f}_{\mathbf{u}}^{\mathsf{T}}\mathbf{w}\right|^2 dt \right)^{-1} \quad \text{if } \mathbf{u} \text{ is a control} \tag{57}$$

Of course, the integrals must be computed numerically, but no additional simulation of the system equations is required to determine κ. However, this strategy does not necessarily result in a decrease of the objective function.

(2) Finding number κ such that J is absolutely reduced may require several simulations of the system equations. For that purpose, the increments given by Eq. (52) or Eq. (53) are considered as functions of κ. After solving the equations of motion with $\mathbf{u} + \delta\mathbf{u}$ as inputs also the objective function J becomes ultimately a function of κ. By means of a line search algorithm, one may then find a number κ in a predefined interval $[0, \kappa_{\max}]$ which minimizes J. In practice, it is sufficient to apply an inexact line search algorithm for that purpose.

4.2 Quasi-Newton Methods. It is well known that the convergence rate of the gradient method is rather slow and that Newton's method provides an alternative approach to find a minimum of the function $J(\mathbf{u})$. The basic idea is the following one: If $\mathbf{u}$ is a vector of N_u numbers, the minimizing vector $\mathbf{u}$ is defined by a zero gradient, i.e., by the equations:

$$\nabla J = \left[\frac{\partial J}{\partial u_1}, \ldots, \frac{\partial J}{\partial u_{N_u}}\right]^{\mathsf{T}} = \mathbf{0} \tag{58}$$

which can be solved by Newton's method. However, the Hesse matrix $\mathbf{H} = (\nabla J)_{\mathbf{u}}$ is required for that purpose. To avoid the full computation of $\mathbf{H}$, which would be extremely time consuming in multibody dynamics, several quasi-Newton methods have been developed. They all approximate the Hessian by using the gradients of successive Newton-iterations. For example, the Hessian or even its inverse can be estimated efficiently by the well-known DFP formula [34, Chap. 6.14, p. 354] or by the classical BFGS algorithm [34, Chap. 6.15, p. 360], respectively. To describe the methods, we have to distinguish between parameter and control optimization problems.

4.2.1 Optimal Parameter. From $\delta J = \nabla J^{\mathsf{T}} \delta\mathbf{u}$ and Eq. (31), the gradient ∇J is readily identified as

$$\nabla J = \int_0^T \left(h_{\mathbf{u}}^{\mathsf{T}} - \mathbf{f}_{\mathbf{u}}^{\mathsf{T}}\mathbf{w}\right) dt \tag{59}$$

if $\mathbf{u}$ is a parameter vector. If $\tilde{\mathbf{H}}^{-1}$ is an approximation of the inverse Hessian computed from the DFP or the BFGS algorithm, an increment $\delta\mathbf{u}$ is given by

$$\delta\mathbf{u} = -\tilde{\mathbf{H}}^{-1} \nabla J \tag{60}$$

4.2.2 Optimal Control. If $\mathbf{u}$ is a control vector, J becomes a functional mapping a function $\mathbf{u}(t)$ on a real number J. Hence, a straight forward application of the BFGS-method is not possible. But since we are dealing with a discretized version of the adjoint

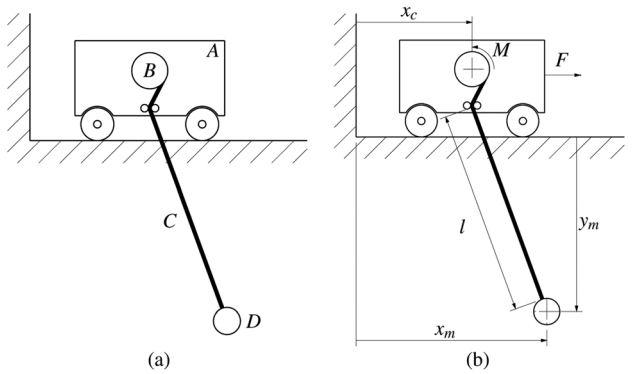


Fig. 1 Geometry description of the planar, rigidly modeled overhead crane

equations, $\mathbf{u}$ is only computed at time instances $t_n = T - \tau_n = T - \gamma n$, see Sec. 3.4. Hence, in the discrete system J can be considered as a function $\hat{J}$ of the vector

$$\hat{\mathbf{u}} = \left[\mathbf{u}^{\mathrm{T}}(t_1), \mathbf{u}^{\mathrm{T}}(t_2), \ldots, \mathbf{u}^{\mathrm{T}}(t_{N_t})\right]^{\mathrm{T}} \tag{61}$$

Moreover, the integral in Eq. (31) is also approximated by

$$\delta\hat{J} = \gamma \sum_{n=1}^{N_t} \left[h_{\mathbf{u}}(t_n) - \mathbf{w}^{\mathrm{T}}(t_n)\mathbf{f}_{\mathbf{u}}(t_n)\right]\delta\mathbf{u}(t_n) \tag{62}$$

Since the gradient $\nabla\hat{J}$ of the function $\hat{J}(\hat{\mathbf{u}})$ is again defined by $\delta\hat{J} = \nabla\hat{J}^{\mathrm{T}}\delta\hat{\mathbf{u}}$, we identify

$$\nabla\hat{J} = \gamma\left[h_{\mathbf{u}}(t_n) - \mathbf{w}^{\mathrm{T}}(t_n)\mathbf{f}_{\mathbf{u}}(t_n)\right]^{\mathrm{T}} \tag{63}$$

From the DFP or the BFGS algorithm, we may now again compute an approximation $\tilde{\mathbf{H}}^{-1}$ of the inverse of the Hessian $(\nabla\hat{J})_{\hat{\mathbf{u}}}$. Then, an increment $\delta\hat{\mathbf{u}}$ of the discretized control is given by

$$\delta\hat{\mathbf{u}} = -\tilde{\mathbf{H}}^{-1}\nabla\hat{J} \tag{64}$$

Note that it is strongly recommended to use a quasi-Newton method which directly approximates $\tilde{\mathbf{H}}^{-1}$. Otherwise, if the original Hessian $\tilde{\mathbf{H}}$ is computed, a very large and dense matrix must be inverted, since the number of components of J given by $N_t \times N_u$ might become large.

5 Numerical Examples

Three numerical examples are presented here in order to show the application of the adjoint method in typical multibody systems. As a first example, a planar overhead crane is considered as an example of an underactuated mechanical system which follows a given trajectory and the optimization process identifies the control force and the control torque in a specific time domain. The second example incorporates a single rigid body which is parameterized by the four redundant Euler parameters. A point of the body is excited in order to follow a specific trajectory and the optimization process identifies all entries of the inertia tensor. The goal of the third example is to find the excitation force which is required to swing up an inverse triple pendulum chain into the upper rest position. Here, the numerical results are computed without any desired trajectory, but specifying the end points of all states and corresponding velocities only. A time history of the identified forces and moments as well as convergence analyses for the cost functionals and the inertia parameters are presented.

5.1 Planar Overhead Crane. A planar overhead crane, as e.g., presented in Refs. [35] and [36], is considered as an example of an underactuated mechanical system consisting of the cart A, the hoisting drum B, the cable C, and a mass D mounted at the end of the cable, see Fig. 1(*a*). The generalized coordinates of this two-dimensional problem are chosen as the position of the cart in x-direction x_{c}, the length of the cable l, and the position of the mass in x- and y-direction, x_{m} and y_{m}, respectively, see Fig. 1(*b*). The optimal control problem is defined such that the point mass D follows a given trajectory defined by a linear path from a given starting point $(x_{\mathrm{m0}},\ y_{\mathrm{m0}}) = (0/4)$ to a given end point $(x_{\mathrm{mf}}, y_{\mathrm{mf}}) = (5/1)$ which is depicted in Fig. 2. The control force F and control torque M depicted in Fig. 1(*b*) are identified within the time period $t \in [0, 3]$. Following Ref. [36], a smooth rest-to-rest maneuver is realized by defining the trajectory in a parameterized form in $\tau = (t - t_0)/(t_0 - t_{\mathrm{f}})$ as follows:

$$x_{\mathrm{m}} = 5\left(70\tau^9 - 315\tau^8 + 540\tau^z - 420\tau^6 + 126\tau^5\right) \tag{65}$$

$$y_{\mathrm{m}} = 4 - 3\left(70\tau^9 - 315\tau^8 + 540\tau^z - 420\tau^6 + 126\tau^5\right) \tag{66}$$

The mass of the cart and the hoisting drum $A + B$ is set to 10 kg, the mass of D is set to 100 kg, the moment of inertia of the hoisting drum is 0.1 kg m^2, and the radius of the hoisting drum is defined as 0.1 m. For this setting, the static torque acting at the hoisting drum M, see Fig. 1(*b*), is given by $M_0 = 100 \times 9.81 \times 0.1 = 98.1$ N m. The results in Fig. 3 are computed with a constant step size and show the initial settings for the first iteration and the identified control force F and control torque M after 300 iterations. The optimization process reduces the costs to a factor 10^{-7} within 300 iterations, see Fig. 4.

5.2 Single Rigid Body Parameterized With Euler Parameters. A three-dimensional, single rigid body is studied which is defined by the position of the center of mass $\mathbf{x} = (x, y, z)$, and the orientation of the body is described by the four redundant Euler parameters $\theta = (\theta_0, \theta_1, \theta_2, \theta_3)$. Figure 5(*a*) shows an arbitrarily shaped rigid body for which the inertia parameters describing the inertia tensor are not known in advance and the goal of the study is to identify the moments of inertia I_{11}, I_{22}, and I_{33} and the entries in the off-diagonal I_{12}, I_{13}, and I_{23}. The point S of the body follows a prescribed motion realized by a constraint, and the velocity of the point P is measured (in global coordinates) in order to identify the entries of the inertia tensor. For the computation, the body shown in Fig. 5(*b*) is considered. The geometry used for generating the inertia data is a cuboid with one diagonal congruent with the z-axis. The excitation for the identification process is applied in the point S situated at the origin at time $t = 0$. Velocity

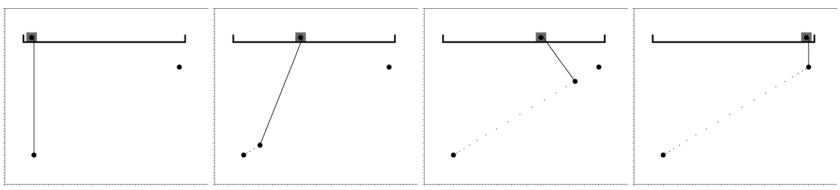


Fig. 2 The point mass has to follow a linear trajectory from a specified starting point to a fixed end point

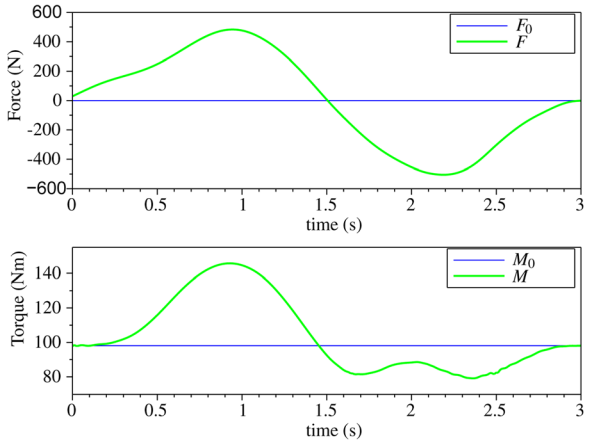


Fig. 3 Time history of identified force *F* and torque *M* after 300 iterations and initial settings for the first iteration for Ex. 5.1. The initial input for the force is set to $F_0 = 0$ N for the first iteration. The initial input for the torque is defined as the static torque $M_0 = 98.1$ N m.

measurements are taken in the point P situated in the opposite direction to S along the z-axis. The mass of the body is set to 692.747 kg. The positions of the point S, the center of mass of the body $\mathbf{x}$, and the point P are chosen (in meters) as follows: $S = (0, 0, 0)$, $\mathbf{x} = (0, 0, -0.916)$, and $P = (0, 0, 0, -1.832)$. The desired trajectory $\overline{s_i}$ of the point S used for the optimization process is generated with the correct parameters of the body under consideration and follows the excitation:

$$\overline{s_i} = 0.05\left(\left(1 + \sin\left(\frac{6\pi}{T}\left(t - \frac{T}{12}\right)\right)\right)\right) \tag{67}$$

for $t \in [0, T]$ with $T = 3$ s in x-direction for the time interval $0 \le t < (T/3)$, in y-direction for the time interval $(T/3) \le t < (2T/3)$, and in z-direction for the time interval $(2T/3) \le t < T$. In order to start the optimization, initial values for the parameters to identify have to be defined. Table 1 shows the correct, the initial values for the first iteration and the final identified values for the moments of inertia after 126 iterations, for which the values are assumed to be converged, see Fig. 6. The convergence analysis of the cost functional shows that the optimization process reduces the costs already within the first 100 iterations to a factor 10^{-8}, see again Fig. 6. Exemplary, the convergence analysis of the moment of inertia I_{11} is shown in Fig. 7. It has to be mentioned here that in case of identifying only the diagonal entries of the inertia tensor, I_{11}, I_{22}, and I_{33}, the results show higher accordance to the correct values as in case of identifying the whole inertia tensor also incorporating the deviation moments of inertia.

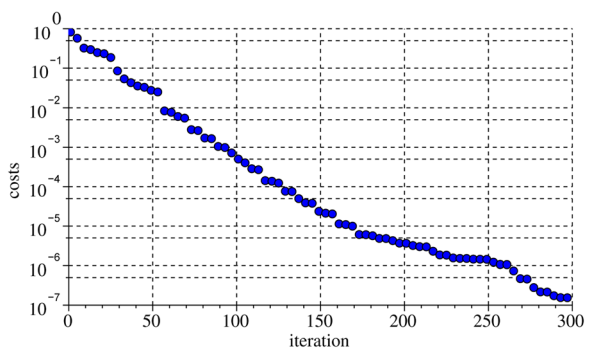


Fig. 4 The convergence analysis of the cost functional for Ex. 5.1 shows that the optimization process reduces the costs to a factor of 10^{-7} within 300 iterations

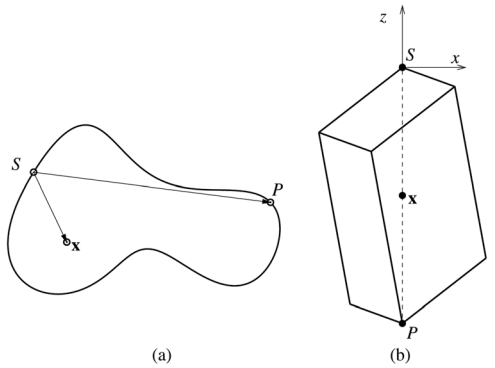


Fig. 5 A single rigid body is studied for which the moments of inertia parameters describing the inertia tensor are not known. Point *S* follows a specified motion, and the velocity of point *P* is measured in order to identify the entries of the inertia tensor.

Table 1 Moments of inertia: the correct values, the initial values for the first iteration, and the final identified values after 126 iterations, for which the values are assumed to be converged

Parameter	Correct value	Initial value	Final identified value	Unit
I_{11}	132.285	1.0	133.441	kg m^2
I_{22}	162.054	1.0	162.073	kg m^2
I_{33}	92.809	1.0	96.815	kg m^2
I_{12}	22.382	0.0	22.198	kg m^2
I_{13}	−42.342	0.0	−44.593	kg m^2
I_{23}	2.689	0.0	3.011	kg m^2

5.3 Inverse Triple Pendulum. An inverse triple pendulum is studied consisting of a cart and three rigid bodies connected by revolute joints. Figure 8(*a*) shows the geometric description of the triple pendulum, which has been considered by various other authors, e.g., by Ref. [37] considering a nonlinear feedforward controller and an optimal feedback controller and its experimental realization. The cart is only allowed to move along the x-axis leading to a two-dimensional motion of the pendulum, see again Fig. 8(*a*). The redundant generalized coordinates are chosen to be the cart position x_c, the position of the center of mass of each pendulum (x_i, y_i), and the absolute pendulum angles ϕ_1, ϕ_2, and ϕ_3. Linear friction torque/force is considered for each revolute joint and between the ground and the cart defined by friction coefficients d_1, d_2, d_3, and d_c. The distances of the center of gravity of

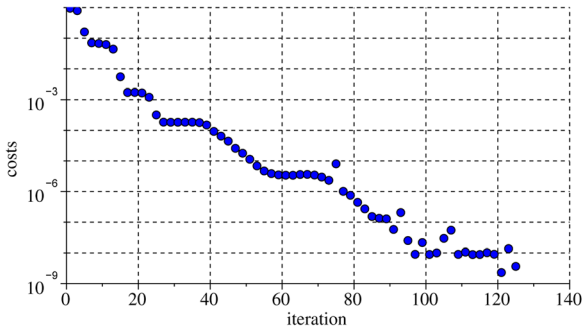


Fig. 6 The convergence analysis of the cost functional for Ex. 5.2 shows that the optimization process reduces the costs already tremendously within the first 100 iterations. It has to be mentioned that only the costs of every second iteration are depicted here.

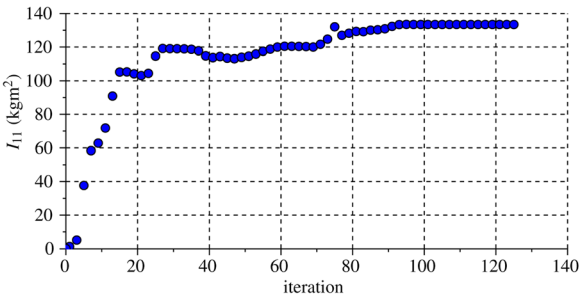
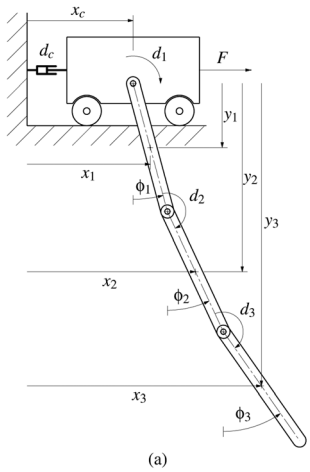


Fig. 7 Exemplary, the convergence analysis of the moment of inertia I_{11} considered in Ex. 5.2 is shown here for 126 iterations. Only the costs of every second iteration are depicted here.



parameter	value	unit
mass of cart	2.0	kg
mass of pendulum 1	0.876	kg
mass of pendulum 2	0.938	kg
mass of pendulum 3	0.553	kg
length of pendulum 1	0.323	m
length of pendulum 2	0.419	m
length of pendulum 3	0.484	m
s_{p_1}	0.215	m
s_{p_2}	0.269	m
s_{p_3}	0.226	m
d_c	0.1	Ns/m
d_1	0.215	Nms
d_2	0.002	Nms
d_3	0.002	Nms

Fig. 8 A triple inverse pendulum is studied for which the excitation force F is identified which leads to a swing up maneuver into the rest position with $\phi_1 = \phi_2 = \phi_3 = \pi$. (*a*) Geometric description of the inverse pendulum and (*b*) definition of the necessary parameters for the numerical simulation.

each pendulum to the according joint are abbreviated by s_{p_1}, s_{p_2}, and s_{p_3} and defined following the values in table in Fig. 8(*b*). The goal of the optimization is to find the excitation force $F(t)$ which is required to swing up the pendulum chain into the upper rest position with all pendulum angles being $\phi_1 = \phi_2 = \phi_3 = \pi$, see Fig. 9. As explained in Sec. 3.1, it is possible to incorporate a scrap function, as introduced in Eq. (19), for computing the final values for the adjoint variables. Hence, it is not necessary to provide a trajectory describing the movement of the system during the swing up motion. The presented numerical results are computed without any desired trajectory, while specifying the end

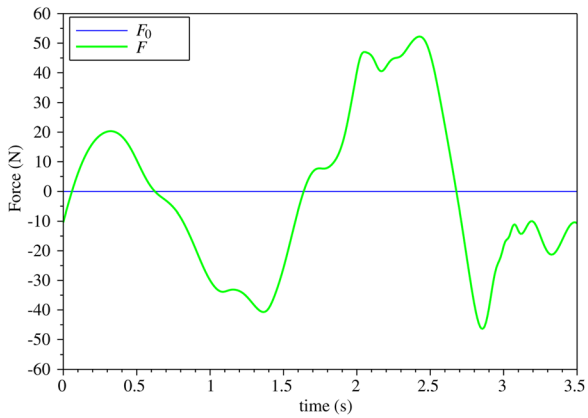


Fig. 10 Time history of identified force F for the inverse triple pendulum in Ex. 5.3 after 353 iterations. The initial input for the force is set to $F_0 = 0$ N for the first iteration.

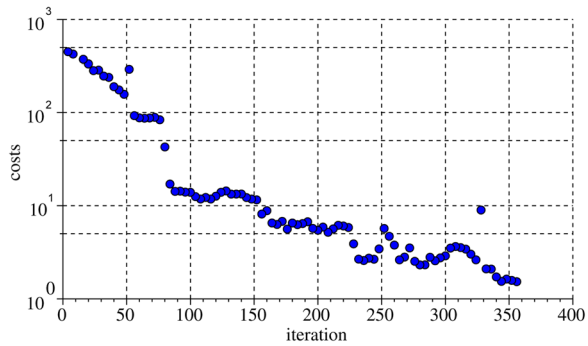


Fig. 11 The costs according to the end point error considered in Ex. 5.3 decrease to the limit of the value 1.5 after 353 iterations

points of all states $\mathbf{q}$ and corresponding velocities $\mathbf{v}$ only. In order to obtain good convergence performance, the error in positions is weighted with a factor 5. The cart position x_c is constrained in a meaningful way by a penalty function considered in Eq. (2). In order to start the optimization, an initial value for the excitation force F has to be defined, here $F_0 = 0$, and a constant step size is chosen. Figure 10 shows the initial setting $F_0 = 0$ for the first iteration and the identified excitation force F after 353 iterations. The optimization process reduces the costs defined by the end point error to a factor which satisfies the optimal control requirements within 353 iterations, see Fig. 11. The time history of the three angles ϕ_1, ϕ_2, and ϕ_3 in the revolute joints are depicted in Fig. 12.

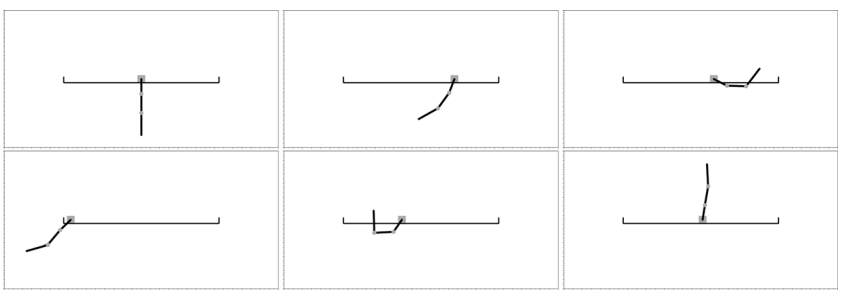


Fig. 9 Simulation results of the swing up maneuver of the inverse triple pendulum at six time steps: $t = 0.0, 0.7, 1.4, 2.3$, and 3.0 s

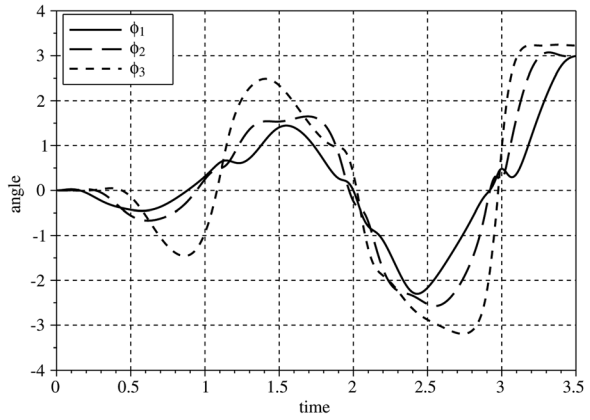


Fig. 12 Time history of the three angles ϕ_1, ϕ_2, and ϕ_3 in the revolute joints considered in Ex. 5.3 after 353 iterations, where the optimization is stopped since the costs according to the end point error decrease below a prescribed limit value

6 Conclusions

The proposed method shows that embedding the adjoint method to multibody system dynamics is straightforward and illustrates its potential for inverse dynamics and parameter identification on account of its linear structure. Depending on the chosen forward time integration scheme, almost all necessary matrices for the backward time integration can be reused from the forward time integration and therefore a time-efficient and memory-efficient simulation tool for inverse dynamics in the field of multibody dynamics can be constructed. Therefore, the adjoint method shows an efficient way to incorporate inverse dynamics to flexible multibody system applications arising from modern engineering or biomechanical problems.

Acknowledgment

This research has been funded by the European Regional Development Fund and the government of Upper Austria via a Regio 13 project.

References

[1] Haug, E., and Ehle, P., 1982, "Second-Order Design Sensitivity Analysis of Mechanical System Dynamics," Int. J. Numer. Methods Eng., **18**(11), pp. 1699–1717.
[2] Haug, E., Wehage, R., and Mani, N., 1984, "Design Sensitivity Analysis of Large-Scaled Constrained Dynamic Mechanical Systems," Trans. ASME, **106**(2), pp. 156–162.
[3] Bestle, D., and Eberhard, P., 1992, "Analyzing and Optimizing Multibody Systems," Mech. Struct. Mach., **20**(1), pp. 67–92.
[4] Eberhard, P., 1996, "Adjoint Variable Method for Sensitivity Analysis of Multibody Systems Interpreted as a Continuous, Hybrid Form of Automatic Differentiation," *Computational Differentiation Techniques, Applications, and Tools*, M. Berz, C. Bischof, G. Corliss, and A. Griewank, A., eds., SIAM, Philadelphia, PA, pp. 319–328.
[5] Lions, J., 1971, *Optimal Control of Systems Governed by Partial Differential Equations*, Springer-Verlag, New York.
[6] Giles, M., and Pierce, N., 2000, "An Introduction to the Adjoint Approach to Design," Flow, Turbul. Combust., **65**(3–4), pp. 393–415.
[7] Jameson, A., 2003, "Aerodynamic Shape Optimization Using the Adjoint Method" (VKI Lecture Series on Aerodynamic Drag Prediction and Reduction), von Karman Institute of Fluid Dynamics, Rhode St. Genese, Belgium, pp. 3–7.
[8] Bryson, A. E., and Ho, Y. C., 1975, *Applied Optimal Control*, Hemisphere, Washington, DC.
[9] Anderson, W., and Venkatakrishnan, V., 1999, "Aerodynamic Design Optimization on Unstructured Grid With a Continuous Adjoint Formulation," Comput. Fluid, **28**(4–5), pp. 443–480.
[10] Nadarajah, S., and Jameson, A., 2000, "A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimisation," AIAA Paper No. 2000-0067.
[11] Oberai, A., Gokhale, N., and Feijóo, G., 2003, "Solution of Inverse Problems in Elasticity Imaging Using the Adjoint Method," Inverse Probl., **19**(2), pp. 297–313.
[12] McNamara, A., Treuille, A., Popović, Z., and Stam, J., 2004, "Fluid Control using the Adjoint Method," ACM Trans. Graphics (TOG) - Proc ACM SIGGRAPH, **23**(3), pp. 449–456.
[13] Taylor, T., Palacios, F., Duraisamy, K., and Alonso, J., 2013, "A Hybrid Adjoint Approach Applied to Turbulent Flow Simulations," 21st AIAA Computational Fluid Dynamics Conference, San Diego, June 24–27.
[14] Bottasso, C., Croce, A., Ghezzi, L., and Faure, P., 2004, "On the Solution of Inverse Dynamics and Trajectory Optimization Problems for Multibody Systems," Multibody Syst. Dyn., **11**(1), pp. 1–22.
[15] Bertolazzi, E., Biral, F., and Lio, M. D., 2005, "Symbolic-Numeric Indirect Method for Solving Optimal Control Problems for Large Multibody Systems," Multibody Syst. Dyn., **13**(2), pp. 233–252.
[16] Schaffer, A., 2005, "On the Adjoint Formulation of Design Sensitivity Analysis of Multibody Dynamics," Ph.D. dissertation, University of Iowa, Iowa City, IA, http://ir.uiowa.edu/etd/93
[17] Petzold, L., Li, S., Cao, Y., and Serban, R., 2006, "Sensitivity Analysis for Differential-Algebraic Equations and Partial Differential Equations," Comput. Chem. Eng., **30**(10–12), pp. 1553–1559.
[18] Cao, Y., Li, S., and Petzold, L., 2002, "Adjoint Sensitivity Analysis for Differential-Algebraic Equations: Algorithms and Software," J. Comput. Appl. Math., **149**(1), pp. 171–191.
[19] Cao, Y., Li, S., Petzold, L., and Serban, R., 2003, "Adjoint Sensitivity Analysis for Differential-Algebraic Equations: The Adjoint DAE System and Its Numerical Solution," SIAM J. Sci. Comput., **24**(3), pp. 1076–1089.
[20] Shabana, A., 1997, "Definition of the Slopes and the Finite Element Absolute Nodal Coordinate Formulation," Multibody Syst. Dyn., **1**(3), pp. 339–348.
[21] Held, A., and Seifried, R., 2013, "Gradient-Based Optimization of Flexible Multibody Systems Using the Absolute Nodal Coordinate Formulation," Proceedings of the ECCOMAS Thematic Conference Multibody Dynamics 2013, Zagreb, Croatia, July 1–4.
[22] Pi, T., Zhang, Y., and Chen, L., 2012, "First Order Sensitivity Analysis of Flexible Multibody Systems Using the Absolute Nodal Coordinate Formulation," Multibody Syst. Dyn., **27**(2), pp. 153–171.
[23] Ding, J.-Y., Pan, Z.-K., and Chen, L.-Q., 2007, "Second Order Adjoint Sensitivity Analysis of Multibody Systems Described by Differential-Algebraic Equations," Multibody Syst. Dyn., **18**(4), pp. 599–617.
[24] Vyasarayani, C., Uchida, T., and McPhee, J., 2012, "Nonlinear Parameter Identification in Multibody Systems Using Homotopy Continuation," ASME J. Comput. Nonlinear Dyn., **7**(1), p. 011012.
[25] Ding, J.-Y., Pan, Z.-K., and Chen, L.-Q., 2012, "Parameter Identification of Multibody Systems Based on Second Order Sensitivity Analysis," Int. J. Nonlinear Mech., **47**(10), pp. 1105–1110.
[26] Özyurt, D. B., and Barton, P. I., 2005, "Cheap Second Order Directional Derivatives of Stiff ODE Embedded Functionals," SIAM J. Sci. Comput., **26**(2), pp. 1725–1743.
[27] Steiner, W., and Reichl, S., 2011, "A contribution to Inverse Dynamic Problems in Multibody Systems," Multibody Dynamics 2011, ECCOMAS Thematic Conference, Brussels, Belgium, July 4–7.
[28] Steiner, W., and Reichl, S., 2012, "The Optimal Control Approach to Dynamical Inverse Problems," ASME J. Dyn. Syst., Meas., Control, **134**(2), p. 021010.
[29] Gear, C., Gupta, G., and Leimkuhler, B., 1985, "Automatic Integration of the Euler-Lagrange Equations With Constraints," J. Comp. Appl. Math., **12/13**, pp. 77–90.
[30] Hilbert, H., Hughes, T., and Taylor, R., 1977, "Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics," Earthquake Eng. Struct. Dyn., **5**(3), pp. 283–292.
[31] Negrut, D., Rampalli, R., Ottarsson, G., and Sajdak, A., 2005, "On the Use of the HHT Method in the Context of Index 3 Differential Algebraic Equations of Multibody Dynamics," ASME Paper No. DETC2005-85096.
[32] Süli, E., and Mayers, D., 2003, *An Introduction to Numerical Analysis*, Cambridge University Press, Cambridge, UK.
[33] Kirk, D., 2004, *Optimal Control Theory*, Dover Publications, Mineola, NY.
[34] Rao, S., 2009, *Engineering Optimization: Theory and Practice*, 4th ed., Wiley, Hoboken, NJ.
[35] Blajer, W., and Kołodziejczyk, K., 2004, "A Geometric Approach to Solving Problems of Control Constraints: Theory and a DAE Framework," Multibody Syst. Dyn., **11**(4), pp. 343–364.
[36] Betsch, P., Uhlar, S., and Quasem, M., 2009, "Numerical Integration of Mechanical Systems With Mixed Holonomic and Control Constraints," Proceedings of the ECCOMAS Thematic Conference on Multibody Dynamics, K. Arczewski, J. Fraczek, and M. Wojtyra, eds., Warsaw University of Technology, Poland, June 29–July 2.
[37] Glück, T., Eder, A., and Kugi, A., 2013, "Swing-Up Control of a Triple Pendulum on a Cart With Experimental Validation," Automatica, **49**(3), pp. 801–808.