Chapter 1
An Efficient Treatment of Parameter Identification in the Context of Multibody System Dynamics Using the Adjoint Method

Karim Sherif, Karin Nachbagauer, Stefan Oberpeilsteiner, and Wolfgang Steiner

Abstract In multibody system dynamics, a wide range of parameters can occur where some of them may not be known a priori. Therefore, this work presents an efficient adjoint method for parameter identification that can be utilized in multibody simulation software. Compared to standard system sensitivity based approaches the adjoint method has the major advantage of being independent on the number of parameters to identify. Especially when dealing with large and probably flexible multibody systems this characteristic is crucial. Formulating parameter identification as an automatable procedure, of course, leads to a complicated structure of the involved matrices and equations. However, during a forward simulation of the system many of the matrices needed for solving the so called "adjoint system equations" are already evaluated. Adopting the functionality of the forward solver for the adjoint system solver therefore results in little additional effort. In order to illustrate the performance of the adjoint method two examples are presented. A planar example shows the possibility of identifying non-linear control parameters and a three-dimensional example is presented for identifying time-invariant inertia parameters.

Keywords Adjoint method • Parameter identification • Optimal control • Inverse dynamics • Multibody systems

1.1 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., [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.

Previous work on the adjoint method in multibody dynamics can be found in the work of Bottasso et al. [7], 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 design of the indirect method for solving optimal control problems for multibody systems presented by Bertolazzi et al. [8] seems to be familiar with the idea of the adjoint method. The work by Schaffer [9] presents a numerical algorithm, the piecewise adjoint method, which formulates the coordinate partitioning underlying ordinary differential equations as a boundary value problem, which is solved by multiple shooting methods.

The group around Petzold, Cao, Li and Serban [10–12] 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 [10]. 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.

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

K. Sherif (✉) • K. Nachbagauer • S. Oberpeilsteiner • W. Steiner
University of Applied Sciences Upper Austria, Wels Campus, Stelzhamerstr. 23, 4600 Wels, Austria
e-mail: karim.sherif@fh-wels.at; karin.nachbagauer@fh-wels.at; stefan.oberpeilsteiner@fh-wels.at; wolfgang.steiner@fh-wels.at

M. Mains (ed.), *Topics in Modal Analysis, Volume 10*, Conference Proceedings of the Society for Experimental Mechanics Series, DOI 10.1007/978-3-319-15251-6_1

of the present paper, which is a shortened form of [13], 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 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.

1.2 The Adjoint Method in Multibody Dynamics

In this section we discuss the application of the adjoint gradient computation to multibody dynamics. The adjoint equations for the equations of motion of a multibody system will be derived and a flowchart for embedding the adjoint method in multibody dynamics is illustrated.

1.2.1 Adjoint Equations

In the augmented formulation the dynamics of a multibody system is described by a set of differential-algebraic equations in the following form:

$$\dot{\mathbf{q}} = \mathbf{v} \tag{1.1}$$

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{f}(\mathbf{q}, \mathbf{v}, \mathbf{u}) - \mathbf{C}_{\mathbf{q}}^{\mathsf{T}}\boldsymbol{\lambda} \tag{1.2}$$

$$\mathbf{C}(\mathbf{q}) = \mathbf{0} \tag{1.3}$$

where $\mathbf{q}$ and $\mathbf{v}$ denote the generalized coordinates and velocities. Moreover, $\mathbf{u}$ may either describe a vector of time-invariant parameters or a vector of time-dependent control signals actuating the system. For simplicity we assume that $\mathbf{u}$ appears only in the vector of generalized forces and gyroscopic forces $\mathbf{f}$, which is e.g. the case if $\mathbf{u}$ is a stiffness or damping parameter or an actuating force. The matrix $\mathbf{M}$ represents the symmetric mass matrix of the system. The algebraic constraint equation $\mathbf{C}(\mathbf{q}) = \mathbf{0}$ influences the equations of motion via the Jacobian $\mathbf{C}_{\mathbf{q}}$ and the vector of Lagrange multipliers $\boldsymbol{\lambda}$.

The key idea of the adjoint method (see [4, 10–12] for example) is to determine the parameter/control $\mathbf{u}$ such that a functional of the form

$$J(\mathbf{u}) = \int_0^T h(\mathbf{q}, \mathbf{v}, \mathbf{u})\, dt + S(\mathbf{q}, \mathbf{v})\Big|_{t=T} \tag{1.4}$$

is minimized, in which the function $h(\mathbf{q}, \mathbf{v}, \mathbf{u})$ and the end point term $S(\mathbf{q}, \mathbf{v})|_T$ have to be defined appropriately. The end point term $S(\mathbf{q}, \mathbf{v})|_T$ is often also denoted as *scrap function*. 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 BFGS algorithm when estimating the Hessian or the DFP algorithm when estimating the inverse of the Hessian. Some authors embed these methods in a homotopy continuation to obtain a global minimum [14]. 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 [15, 16] or [14] for example). The computational effort for this approach is equal to solving N_u linear sets of equations with the same dimension as Eqs. (1.1)–(1.3). On the other hand, if $\mathbf{u}$ represents a vector of time-dependent 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 incorporate Eqs. (1.1)–(1.3) to the integrand

$$J(\mathbf{u}) = \int_0^T \left[h(\mathbf{q}, \mathbf{v}, \mathbf{u}, t) + \mathbf{p}^{\mathsf{T}}(\dot{\mathbf{q}} - \mathbf{v}) + \mathbf{w}^{\mathsf{T}}(\mathbf{M}\dot{\mathbf{v}} - \mathbf{f} + \mathbf{C}_{\mathbf{q}}^{\mathsf{T}}\boldsymbol{\lambda}) + \boldsymbol{\mu}^{\mathsf{T}}\mathbf{C}\right] dt + S(\mathbf{q}, \mathbf{v})\Big|_{t=T} \tag{1.5}$$

no matter how the adjoint variables $\mathbf{p}(t)$, $\mathbf{w}(t)$ and $\boldsymbol{\mu}(t)$ are chosen.

Let us now consider a forward solution $\mathbf{q}(t)$, $\mathbf{v}(t)$ and $\boldsymbol{\lambda}(t)$ of the system Eqs. (1.1)–(1.3) for a set of parameters or control variables $\mathbf{u}$. A variation of $\mathbf{u}$ will result in variations of the variables $\mathbf{q}$, $\mathbf{v}$ and $\boldsymbol{\lambda}$, and, moreover, in a variation of the functional J. Thus, the variation of the functional J can be written in the form

$$\begin{aligned}\delta J = \int_0^T \Big\{ & h_{\mathbf{q}}\delta\mathbf{q} + h_{\mathbf{v}}\delta\mathbf{v} + h_{\mathbf{u}}\delta\mathbf{u} + \mathbf{p}^{\mathsf{T}}(\delta\dot{\mathbf{q}} - \delta\mathbf{v}) + \\ & + \mathbf{w}^{\mathsf{T}}\left[(\mathbf{M}\dot{\mathbf{v}})_{\mathbf{q}}\,\delta\mathbf{q} + \mathbf{M}\delta\dot{\mathbf{v}} - \mathbf{f}_{\mathbf{q}}\delta\mathbf{q} - \mathbf{f}_{\mathbf{v}}\delta\mathbf{v} - \mathbf{f}_{\mathbf{u}}\delta\mathbf{u} + \left(\mathbf{C}_{\mathbf{q}}^{\mathsf{T}}\boldsymbol{\lambda}\right)_{\mathbf{q}}\delta\mathbf{q} + \mathbf{C}_{\mathbf{q}}^{\mathsf{T}}\delta\boldsymbol{\lambda}\right] + \boldsymbol{\mu}^{\mathsf{T}}\mathbf{C}_{\mathbf{q}}\delta\mathbf{q}\Big\}\, dt \\ & + S_{\mathbf{q}}\delta\mathbf{q}\Big|_{t=T} + S_{\mathbf{v}}\delta\mathbf{v}\Big|_{t=T}\end{aligned} \tag{1.6}$$

where $h_{\mathbf{q}}$, $h_{\mathbf{v}}$ and $h_{\mathbf{u}}$ denote the partial derivatives of the function h with respect to the components of $\mathbf{q}$, $\mathbf{v}$ and $\mathbf{u}$.

In order to get rid of the variations of $\dot{\mathbf{q}}$ and $\dot{\mathbf{v}}$ we integrate by parts

$$\begin{aligned}\int_0^T \mathbf{p}^{\mathsf{T}}\delta\dot{\mathbf{q}}\, dt &= -\int_0^T \dot{\mathbf{p}}^{\mathsf{T}}\delta\mathbf{q}\, dt + \mathbf{p}^{\mathsf{T}}\delta\mathbf{q}\Big|_{t=T} \\ \int_0^T \mathbf{w}^{\mathsf{T}}\mathbf{M}\delta\dot{\mathbf{v}}\, dt &= -\int_0^T \frac{d}{dt}\left(\mathbf{w}^{\mathsf{T}}\mathbf{M}\right)\delta\mathbf{v}\, dt + \mathbf{w}^{\mathsf{T}}\mathbf{M}\delta\mathbf{v}\Big|_{t=T}\end{aligned} \tag{1.7}$$

where the fact has been used that $\delta\mathbf{q}(0) = 0$ and $\delta\mathbf{v}(0) = 0$ since the initial conditions for the variables $\mathbf{q}$ and $\mathbf{v}$ are given. Using Eq. (1.7) the variation of the functional J given by Eq. (1.6) can be rewritten as

$$\begin{aligned}\delta J = \int_0^T \Big\{ & \left[h_{\mathbf{q}} - \dot{\mathbf{p}}^{\mathsf{T}} + \mathbf{w}^{\mathsf{T}}\left((\mathbf{M}\dot{\mathbf{v}})_{\mathbf{q}} - \mathbf{f}_{\mathbf{q}} + (\mathbf{C}_{\mathbf{q}}^{\mathsf{T}}\boldsymbol{\lambda})_{\mathbf{q}}\right) + \boldsymbol{\mu}^{\mathsf{T}}\mathbf{C}_{\mathbf{q}}\right]\delta\mathbf{q} \\ & + \left[h_{\mathbf{v}} - \mathbf{p}^{\mathsf{T}} - \mathbf{w}^{\mathsf{T}}\mathbf{f}_{\mathbf{v}} - \frac{d}{dt}\left(\mathbf{w}^{\mathsf{T}}\mathbf{M}\right)\right]\delta\mathbf{v} + \left[\mathbf{w}^{\mathsf{T}}\mathbf{C}_{\mathbf{q}}^{\mathsf{T}}\right]\delta\boldsymbol{\lambda} + \left[h_{\mathbf{u}} - \mathbf{w}^{\mathsf{T}}\mathbf{f}_{\mathbf{u}}\right]\delta\mathbf{u}\Big\}\, dt \\ & + \left[S_{\mathbf{q}} + \mathbf{p}^{\mathsf{T}}\right]\delta\mathbf{q}\Big|_{t=T} + \left[S_{\mathbf{v}} + \mathbf{w}^{\mathsf{T}}\mathbf{M}\right]\delta\mathbf{v}\Big|_{t=T}\end{aligned} \tag{1.8}$$

The key question now is how the variation of J is related to the variation of $\mathbf{u}$. In order to obtain this relationship we choose our adjoint variables $\mathbf{p}(t)$, $\mathbf{w}(t)$ and $\boldsymbol{\mu}(t)$ such that the terms involving $\delta\mathbf{q}$, $\delta\mathbf{v}$ and $\delta\boldsymbol{\lambda}$ vanish from the integrand. Thus, we end up with the following system of adjoint equations:

$$\frac{d\mathbf{p}}{dt} = h_{\mathbf{q}}^{\mathsf{T}} + \mathbf{A}\mathbf{w} + \mathbf{C}_{\mathbf{q}}^{\mathsf{T}}\boldsymbol{\mu} \tag{1.9}$$

$$\frac{d}{dt}(\mathbf{M}\mathbf{w}) = h_{\mathbf{v}}^{\mathsf{T}} - \mathbf{p} - \mathbf{B}\mathbf{w} \tag{1.10}$$

$$\mathbf{0} = \mathbf{C}_{\mathbf{q}}\mathbf{w} \tag{1.11}$$

$$\mathbf{0} = S_{\mathbf{q}}^{\mathsf{T}} + \mathbf{p}(T) \tag{1.12}$$

$$\mathbf{0} = S_{\mathbf{v}}^{\mathsf{T}} + \mathbf{M}(\mathbf{q}(T))\mathbf{w}(T) \tag{1.13}$$

with the following abbreviations

$$\mathbf{A} = (\mathbf{M}\dot{\mathbf{v}})_{\mathbf{q}}^{\mathsf{T}} - \mathbf{f}_{\mathbf{q}}^{\mathsf{T}} + (\mathbf{C}_{\mathbf{q}}^{\mathsf{T}}\boldsymbol{\lambda})_{\mathbf{q}}^{\mathsf{T}} \tag{1.14}$$

$$\mathbf{B} = \mathbf{f}_{\mathbf{v}}^{\mathsf{T}} \tag{1.15}$$

Using this set of adjoint equations, Eq. (1.8) reduces to

$$\delta J = \int_0^T \left[h_{\mathbf{u}} - \mathbf{w}^{\mathsf{T}} \mathbf{f}_{\mathbf{u}}\right] \delta \mathbf{u}\, dt \tag{1.16}$$

which directly relates the independent variation $\delta\mathbf{u}$ to the variation of the objective function. In order to find the minimum of J, we walk along its negative gradient for a finite distance and update the gradient form time to time until we reach a point where the gradient gets zero. If $\mathbf{u}$ is a control signal, it is easy to show that the largest infinitesimal increase of δJ is obtained, if

$$\delta \mathbf{u}(t) = -\kappa \left[h_{\mathbf{u}} - \mathbf{w}^{\mathsf{T}} \mathbf{f}_{\mathbf{u}}\right]^{\mathsf{T}} \quad \text{(control optimization)} \tag{1.17}$$

where κ is an infinitesimal positive factor. Hence Eq. (1.17) is the gradient formula for the cost functional we are looking for.

Moreover, if $\mathbf{u}$ is a vector of time-invariant parameters, Eq. (1.16) can be written as

$$\delta J = \left(\int_0^T \left[h_{\mathbf{u}} - \mathbf{w}^{\mathsf{T}} \mathbf{f}_{\mathbf{u}}\right] dt\right) \delta \mathbf{u} = \nabla J^{\mathsf{T}} \delta \mathbf{u} \tag{1.18}$$

where the gradient of the functional J is defined as $\nabla J = \int_0^T \left[h_{\mathbf{u}} - \mathbf{w}^{\mathsf{T}} \mathbf{f}_{\mathbf{u}}\right]^{\mathsf{T}} dt$. Thus, the variation of $\mathbf{u}$ can be written for the case of time-invariant parameters as

$$\delta \mathbf{u} = -\kappa \nabla J = -\kappa \left(\int_0^T \left[h_{\mathbf{u}} - \mathbf{w}^{\mathsf{T}} \mathbf{f}_{\mathbf{u}}\right]^{\mathsf{T}} dt\right) \quad \text{(parameter optimization)} \tag{1.19}$$

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

The adjoint equations (1.9)–(1.13) have to be solved backwards in the physical time. Therefore it is advantageous to introduce a new time coordinate τ by the following transformation

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

For the numerical solution of the adjoint equations 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 $\tau_n, \tau_{n-1}, \ldots, \tau_{n-k}$. The backward differentiation formula (BDF) reads

$$\left.\frac{dF}{d\tau}\right|_{\tau_n} \approx \frac{1}{\gamma} \sum_{i=0}^{k} \alpha_i F(\tau_{n-i}) \tag{1.20}$$

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 [17, p. 349].

A closer look at the adjoint equations (1.9)–(1.13) shows that the boundary condition (1.13) of the variable $\mathbf{w}$ is in general incompatible with the adjoint constraint equation (1.11). 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 $\mathbf{w}(T) = \mathbf{0}$. For the case that $S_{\mathbf{v}} \neq 0$ a consistent boundary condition approach has to be applied, see [13] for more details on this issue.

1.2.2 Flowchart of the Adjoint Method Embedded in Multibody Systems

At this point it should be mentioned that only two systems of DAEs must be integrated for computing the direction of the gradient, that is Eqs. (1.1)–(1.3) and (1.9)–(1.13). Subsequently, we give a flowchart of those steps, which are needed for the computation of 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 flowchart describes how one can successively decrease the cost function J until a user-defined limit is achieved by varying $\mathbf{u}$.

1. Solve the equations of motion Eqs. (1.1)–(1.3) forward in time $t \in [0, T]$ yielding $\mathbf{q}(t)$, $\mathbf{v}(t)$ and $\boldsymbol{\lambda}(t)$. This may be done e.g. by choosing the Hilbert-Hughes-Taylor (HHT) integration scheme, as proposed in [18] and its application for a differential algebraic system given in an index three formulation in [19].
2. Compute the objective function J by inserting $\mathbf{q}(t)$, $\mathbf{v}(t)$ and $\mathbf{u}(t)$ into Eq. (1.4). Note, that the integration must be done numerically. If J is smaller than a user defined minimum then stop.
3. Along the forward simulation of the equations of motion compute the mass matrix $\mathbf{M}$, the constraint Jacobian $\mathbf{C_q}$ and the Jacobian matrices $\mathbf{A}$, $\mathbf{B}$ from Eqs. (1.14) and (1.15).
4. Determine the consistent boundary conditions at $\tau = 0$ for the adjoint variables $\mathbf{w}$, $\mathbf{p}$ as suggested in [13].
5. Solve the adjoint equations (1.9)–(1.13) for $\mathbf{p}(\tau)$, $\mathbf{w}(\tau)$ and $\boldsymbol{\mu}(\tau)$, where $\tau = T - t$.
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 $h_\mathbf{u}$ and $\mathbf{f_u}$ along the forward simulation.
7. Compute $\mathbf{u} = \mathbf{u} + \delta\mathbf{u}$. Use Eq. (1.17) for the computation of $\delta\mathbf{u}$ if $\mathbf{u}$ is a control signal. Otherwise, if $\mathbf{u}$ is a set of time-invariant parameters, use Eq. (1.19) for the update. A sufficiently small number $\kappa > 0$ has to be chosen for that purpose. Go to step 1.

For the sake of overview, it should be mentioned at this point that the Jacobian matrices $(\mathbf{M}\dot{\mathbf{v}})_\mathbf{q}$, $\mathbf{f_q}$, $\mathbf{f_v}$ and $(\mathbf{C}_\mathbf{q}^\top\boldsymbol{\lambda})_\mathbf{q}$ which are needed in the adjoint equations may be required already for the simulation of the multibody system if an implicit integration scheme such as, e.g., the HHT-algorithm [18, 19] is applied.

1.3 Numerical Examples

Two numerical examples will be presented to illustrate 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 parametrized 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. A time history of the identified forces and moments as well as convergence analyses for the cost functionals and the inertia parameters are presented.

1.3.1 Planar Overhead Crane

The overhead crane presented in [20, 21] is a classical 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. 1.1a. 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. 1.1b. The goal of this example is to compute the drive signals F and M, depicted in Fig. 1.1b, such that the point mass D follows a given trajectory defined by a linear path from a given starting point $(x_{m0}, y_{m0}) = (0/4)$ to a given end point $(x_{mf}, y_{mf}) = (5/1)$ within 3 s, see Fig. 1.2. The control force F and control torque M are identified within the time period $t \in [0, 3]$. For the computation we set the mass of the cart and the hoisting drum $A + B$ to 10 kg, the mass of D is set to 100 kg, the moment of inertia of the hoisting drum is defined as 0.1 kg m^2 and the radius of the hoisting drum is given by 0.1 m. As starting point of our drive signals we choose the static solution, i.e. $M_0 = 100 \cdot 9.81 \cdot 0.1 = 98.1$ N m and $F_0 = 0$ N. The results in Fig. 1.3 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. 1.4.

1.3.2 Single Rigid Body Parametrized with Euler Parameters

The second example fall in the category of parameter identification, i.e. $\mathbf{u}$ is time-invariant. The goal of the present example is to identify the components of the inertia tensor (I_{11}, I_{22}, I_{33}, I_{12}, I_{13} and I_{23}) of a single three-dimensional rigid body. The position of the rigid body is described by the coordinates of the center of mass $\mathbf{x} = (x, y, z)$ and for the description of

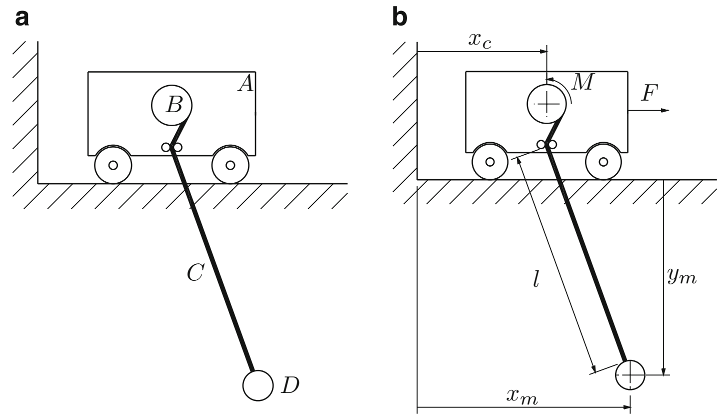

Fig. 1.1 Geometry description of the planar, rigidly modeled overhead crane (**a**) Planar overhead crane (**b**) Parameters of planar overhead crane

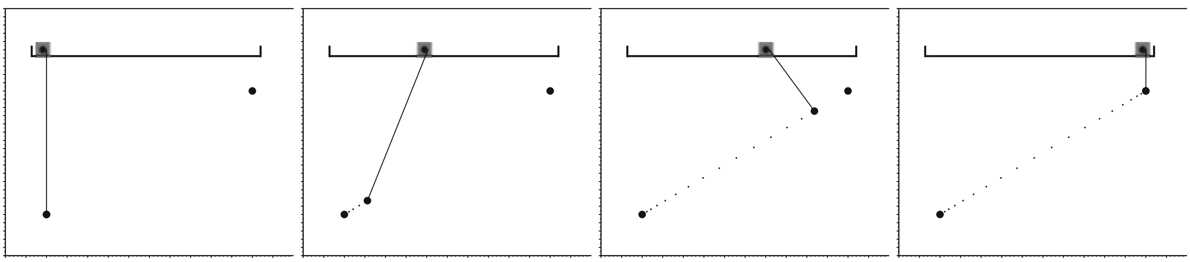

Fig. 1.2 The point mass has to follow a linear trajectory from a specified starting point to a fixed end point

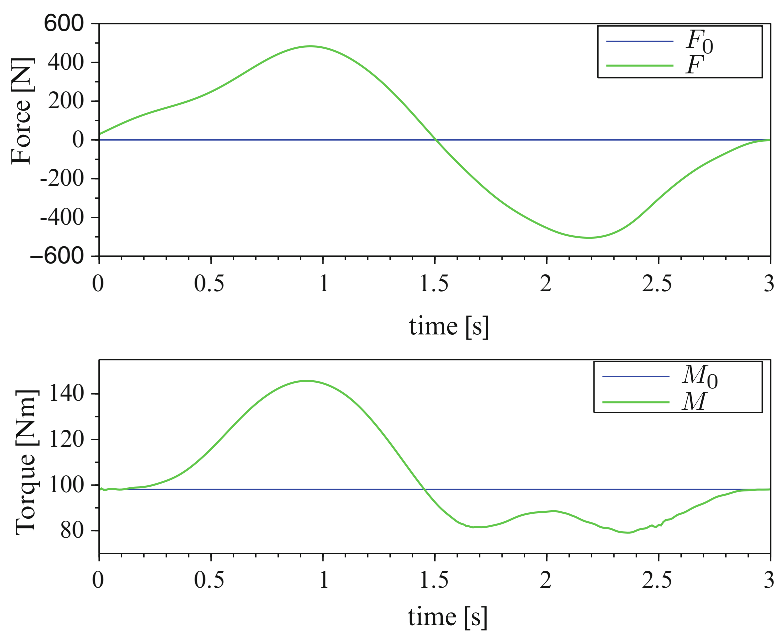

Fig. 1.3 Time history of identified force F and torque M after 300 iterations and initial settings for the first iteration for Sect. 1.3.1. The initial input for the force is set to $F_0 = 0\,\mathrm{N}$ for the first iteration. The initial input for the torque is defined as the static torque $M_0 = 98.1\,\mathrm{N\,m}$

the orientation of the body Euler parameters $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3)$ are used. Figure 1.5 shows the arbitrarily shaped rigid body for which the inertia parameters describing the inertia tensor are not known in advance. 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. 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 measurements are taken in the point P situated in the opposite direction to S along the z-axis. In order to start the optimization, *initial* values for the parameters to identify have to be defined. Table 1.1 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. 1.6. The convergence analysis of the cost functional shows that the optimization process

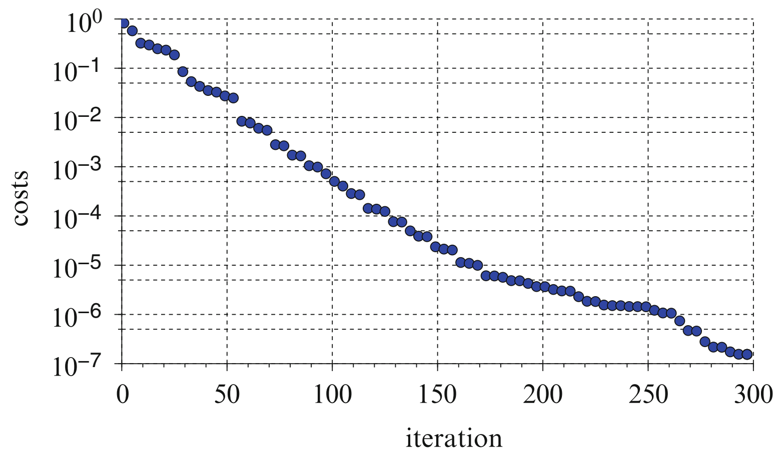

Fig. 1.4 The convergence analysis of the cost functional shows that the optimization process reduces the costs to a factor of 10^{-7} within 300 iterations

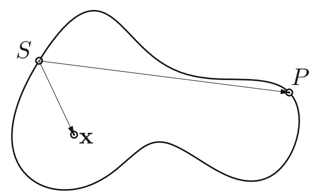

Fig. 1.5 A single rigid body is studied for which the moments of inertia parameters describing the inertia tensor are not known. Point S follows a specified motion and the velocity of point P is measured in order to identify the entries of the inertia tensor

Table 1.1 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

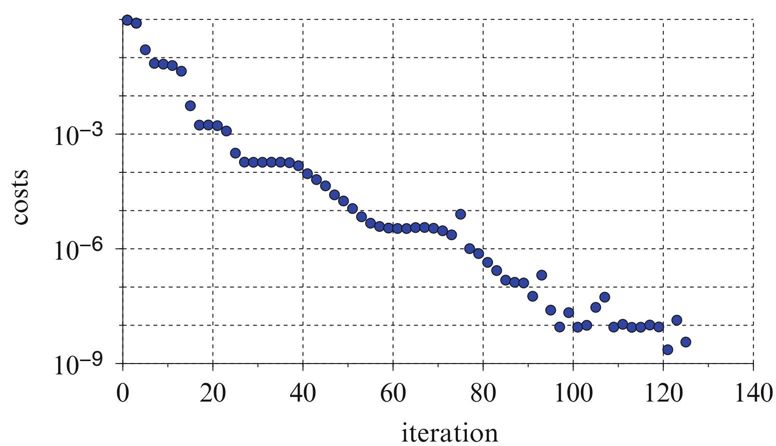

Fig. 1.6 The convergence analysis of the cost functional shows that the optimization process reduces the costs already tremendously within the first 100 iterations. It has to be mentioned that only the costs of every second iteration are depicted here

reduces the costs already within the first 100 iterations to a factor 10^{-8}, see Fig. 1.6. It has to be mentioned 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.

1.4 Conclusions

The adjoint method has been embedded to multibody system dynamics. The presented method can be used for inverse dynamic problems as well as for parameter identification in multibody systems. The computation of the direction of the gradient of the cost functional is based on solving two DAEs. On the one hand, the equations of motion of the multibody system have to be integrated forward in time, and, on the other hand, the adjoint equations have to be integrated backwards. 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- 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 problems.

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

References

1. Haug E, Ehle P (1982) Second-order design sensitivity analysis of mechanical system dynamics. Int J Numer Methods Eng 18(11):1699–1717
2. Haug E, Wehage R, Mani N (1984) Design sensitivity analysis of large-scaled constrained dynamic mechanical systems. Trans ASME 106: 156–162
3. Bestle D, Eberhard P (1992) Analyzing and optimizing multibody systems. Mech Struct Mach 20:67–92
4. Eberhard P (1996) Adjoint variable method for sensitivity analysis of multibody systems interpreted as a continuous, hybrid form of automatic differentiation. In: Proceedings of the 2nd international workshop on computational differentiation, Santa Fe. SIAM, Philadelphia, pp 319–328
5. Lions J (1971) Optimal control of systems governed by partial differential equations. Springer, New York
6. Giles M, Pierce N (2000) An introduction to the adjoint approach to design. Flow Turbul Combust 65:393–415
7. Bottasso C, Croce A, Ghezzi L, Faure P (2004) On the solution of inverse dynamics and trajectory optimization problems for multibody systems. Multibody Sys Dyn 11:1–22
8. Bertolazzi E, Biral F, Lio MD (2005) Symbolic-numeric indirect method for solving optimal control problems for large multibody systems. Multibody Sys Dyn 13:233–252
9. Schaffer A (2005) On the adjoint formulation of design sensitivity analysis of multibody dynamics. Dissertation, University of Iowa. http://ir.uiowa.edu/etd/93
10. Petzold L, Li S, Cao Y, Serban R (2006) Sensitivity analysis for differential-algebraic equations and partial differential equations. Comput Chem Eng 30:1553–1559
11. Cao Y, Li S, Petzold L (2002) Adjoint sensitivity analysis for differential-algebraic equations: algorithms and software. J Comput Appl Math 149:171–191
12. Cao Y, Li S, Petzold L, Serban R (2003) Adjoint sensitivity analysis for differential-algebraic equations: the adjoint DAE system and its numerical solution. SIAM J Sci Comput 24(3):1076–1089
13. Nachbagauer K, Oberpeilsteiner S, Sherif K, Steiner W (2014) The use of the adjoint method for solving typical optimization problems in multibody dynamics. J Comput Nonlinear Dyn. doi: 10.1115/1.4028417
14. Vyasarayani CP, Uchida T, McPhee J (2011) Parameter identification in multibody systems using lie series solutions and symbolic computation. J Comput Nonlinear Dyn 6(4):041011. doi: 10.1115/1.4003686
15. Ding JY, Pan ZK, Chen LQ (2012) Parameter identification of multibody systems based on second order sensitivity analysis. Int J Non-Linear Mech 47:1105–1110
16. Özyurt DB, Barton PI (2005) Cheap second order directional derivatives of stiff ODE embedded functionals. SIAM J Sci Comput 26(2): 1725–1743
17. Süli E, Mayers D (2003) An introduction to numerical analysis. Cambridge University Press, Cambridge
18. Hilbert H, Hughes T, Taylor R (1977) Improved numerical dissipation for time integration algorithms in structural dynamics. Earthq Eng Struct Dyn 5:283–292
19. Negrut D, Rampalli R, Ottarsson G, Sajdak A (2005) On the use of the HHT method in the context of index 3 differential algebraic equations of multibody dynamics. In: Goicolea JM, Cuadrado J, Garcia Orden JC (eds) Proceedings of the ECCOMAS conference on advances in computational multibody dynamics, Madrid
20. Blajer W, Kołodziejczyk K (2004) A geometric approach to solving problems of control constraints: theory and a DAE framework. Multibody Sys Dyn 11(4):343–364
21. Betsch P, Uhlar S, Quasem M (2009) Numerical integration of mechanical systems with mixed holonomic and control constraints. In: Arczewski K, Frączek J, Wojtyra M (eds) Proceedings of the ECCOMAS thematic conference on multibody dynamics, Warsaw University of Technology, 29th June–2nd July