

Analyzing and Optimizing Multibody Systems*

Dieter Bestle and Peter Eberhard

UNIVERSITY OF STUTTGART
INSTITUTE B OF MECHANICS
STUTTGART, F.R.G.

ABSTRACT

Optimization of holonomic as well as non-holonomic multibody systems is presented as a nonlinear programming problem that can be solved with general-purpose optimization codes. The adjoint variable approach is used for calculating design derivatives of a rather general integral type performance measure with respect to design parameters. The resulting equations are solved by numerical integration backward in time. A multi-step integration algorithm with order and step-size control is adapted for this application by including an interpolation scheme. Numerical experiments and a comparison to the common approach of approximating the gradient of the performance measure by finite differences show that high efficiency, accuracy, and reliability are achievable.

*Communicated by E. J. Haug.

I. INTRODUCTION

Modeling technical systems with multibody dynamics programs is a well-accepted approach for analyzing the dynamic behavior of such systems. Implicitly, such a modeling procedure involves a parametrization of the mechanical system by design variables such as the total mass and moments of inertia of each body, dimensions of the bodies, and damping and stiffness coefficients of coupling force elements. In recent years, computer codes have been developed for generating the equations of motion for multibody systems automatically in numerical or symbolic form [1]. Although pre- and post-processing of multibody system programs have been improved, there is still no general approach for systematically optimizing the dynamic behavior of multibody systems.

In general, the optimization of a design requires several iterative steps of successively analyzing a given design and finding a better one from this information. Analyzing the dynamic behavior of a multibody system is a very time-consuming task, because it involves numerical integration of ordinary differential equations over a rather long time period. In order to restrict the number of these costly steps, one should choose an optimization algorithm with good convergence properties. Numerical experiments have shown that methods using design derivatives are much more efficient than no-derivative methods. Approximating the design derivatives by finite differences requires even fewer performance evaluations than using no-derivative algorithms. Comparisons of several optimization algorithms have shown the sequential quadratic programming (SQP) algorithm to be one of the most efficient methods [2, 3]. This algorithm, however, has the drawback of using design derivatives of objective and constraint functions with respect to the parameters of the optimization problem. Thus, efficient generation of these sensitivity functions is the missing link between multibody system analysis codes and satisfying optimization algorithms.

In principle, there are two different approaches for generating additional equations for computing design derivatives: the theoretically simple direct differentiation method and the numerically more efficient adjoint variable approach that is applied in this paper. The adjoint variable approach is closely related to the theory of optimal control. It is used extensively in design sensitivity analysis of structural systems [4, 5]. Because of the linearity and symmetry of the equations in structural dynamics, the same algorithms can be applied in computing the nodal displacements and adjoint variables. For nonlinear multibody systems, this is not possible. The structure of the adjoint equations depends on the formulation of the equations of motion and the type of performance measure considered.

Haug [6] has applied the direct differentiation method and the adjoint variable method to multibody systems given by differential-algebraic equations. In the present paper, however, the multibody system kinematics is described by Lagrangian coordinates, which results in equations of motion described by ordinary differential equations only. The performance measure includes the generalized position, velocity, and acceleration coordinates.

In Section 2 of this paper, the general optimization problem is formulated. The dynamic behavior of the multibody system is described by ordinary differential equations. For the objective function, a rather general integral formulation is given. In Section 3, the adjoint variable method is applied to the dynamic system for computing design derivatives with respect to design parameters. In Section 4, algorithms for solving the equations of motion and the differential equations for the adjoint variables are described, and an extrapolation method is provided for checking the gradients by finite differences. Examples that show the efficiency and accuracy of the proposed method are given in Section 5.

II. MULTIBODY DYNAMICS

A multibody system is an idealization of a technical system where bodies are considered as rigid, i.e., nondeformable. These bodies are connected by ideal links and force elements without mass such as springs, dampers, or active elements (Fig. 1).

The equations of motion for multibody systems can be written as

$$\dot{\mathbf{y}} = \mathbf{v}(t, \mathbf{y}, \mathbf{z}, \mathbf{p}) \quad (1)$$

$$\mathbf{M}(t, \mathbf{y}, \mathbf{p})\dot{\mathbf{z}} + \mathbf{k}(t, \mathbf{y}, \mathbf{z}, \mathbf{p}) = \mathbf{q}(t, \mathbf{y}, \mathbf{z}, \mathbf{p}) \quad (2)$$

where the dot indicates differentiation with respect to time [7]. Nomenclature is presented in Table 1.

For mechanical systems, the kinematic relation of Eq. 1 is linear in velocity coordinates \mathbf{z} . If there are holonomic constraints only, the degrees of freedom for position and velocity are equal, i.e., $g = f$, and usually $\mathbf{z} = \dot{\mathbf{y}}$ is taken for the generalized coordinate velocity vector. However, sometimes it is advantageous to introduce new generalized velocity coordinates, even in the holonomic case.

In general, not all parameters of a multibody system must be considered as variable for optimizing the design. Some of them may be kept constant due to given technical restrictions. Thus, the parameter vector $\mathbf{p} \in \mathbf{R}^h$ includes only parameters of the multibody model which can be varied within

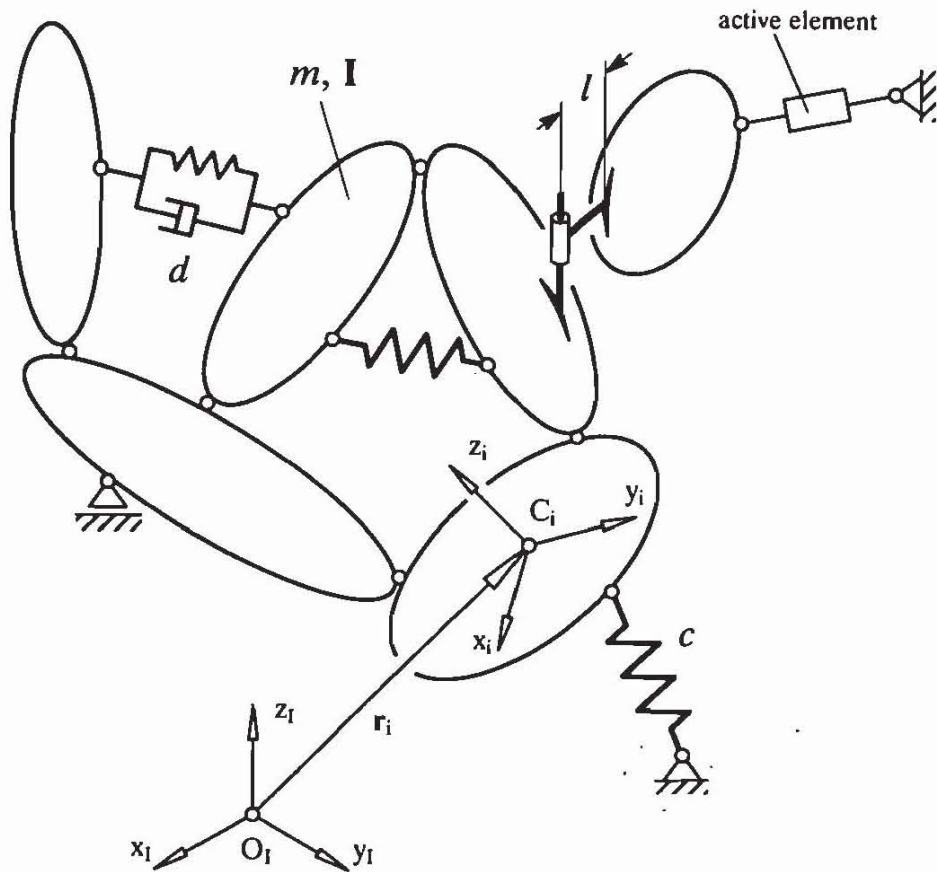


Fig. 1 Multibody system.

TABLE 1
Nomenclature

| | |
|----------------------|--|
| $y \in \mathbb{R}^f$ | generalized coordinate position vector |
| $z \in \mathbb{R}^g$ | generalized coordinate velocity vector |
| f | degrees of freedom for position |
| g | degrees of freedom for velocity |
| v | kinematic relation |
| M | mass matrix |
| k | centrifugal and Coriolis acceleration vector |
| q | generalized external force vector |
| $p \in \mathbb{R}^4$ | parameter vector |

given ranges for optimization purposes; i.e.,

$$p_k^l \leq p_k \leq p_k^u, \quad k = 1 \text{ to } h \quad (3)$$

The lower and upper limits \mathbf{p}^l and \mathbf{p}^u can arise from the physical meaning of the parameters, e.g., $p_k \geq 0$ for stiffness and damping coefficients, or from technical requirements. If the parameters are not independent of each other, implicit algebraic relations

$$\phi(\mathbf{p}) = 0 \quad (4)$$

or combined restrictions such as

$$\phi(\mathbf{p}) \geq 0 \quad (5)$$

must be considered as constraints of the optimization problem.

The equations of motion of Eqs. 1 and 2 must be augmented with initial conditions for the generalized position and velocity coordinates at some fixed starting time t^0 . In general, these initial conditions depend implicitly on the parameters \mathbf{p} ; i.e.,

$$\Phi^0(t^0, \mathbf{y}^0, \mathbf{p}) = \mathbf{0}, \quad \Phi^0: \mathbf{R} \times \mathbf{R}^f \times \mathbf{R}^h \rightarrow \mathbf{R}^f, \quad \det \frac{\partial \Phi^0}{\partial \mathbf{y}^0} \neq 0 \quad (6)$$

$$\begin{aligned} \dot{\Phi}^0(t^0, \mathbf{y}^0, \mathbf{z}^0, \mathbf{p}) = \mathbf{0}, \quad \dot{\Phi}^0: \mathbf{R} \times \mathbf{R}^f \times \mathbf{R}^g \times \mathbf{R}^h \rightarrow \mathbf{R}^g, \\ \det \frac{\partial \dot{\Phi}^0}{\partial \mathbf{z}^0} \neq 0 \end{aligned} \quad (7)$$

For determining the initial state \mathbf{y}^0 and \mathbf{z}^0 uniquely, the Jacobians of Φ^0 and $\dot{\Phi}^0$ must be regular.

To evaluate the dynamic behavior of a system, it is necessary to define a measure of performance. A rather general form is given by the functional

$$\psi(\mathbf{p}) = G^1(t^1, \mathbf{y}^1, \mathbf{z}^1, \mathbf{p}) + \int_{t^0}^{t^1} F(t, \mathbf{y}, \mathbf{z}, \dot{\mathbf{z}}, \mathbf{p}) dt \quad (8)$$

The first term includes the case in which special values for the final state \mathbf{y}^1 , \mathbf{z}^1 or a minimum time t^1 must be achieved. The second term is a functional evaluating the trajectories of the generalized position, velocity, and acceleration coordinates within a time interval $[t^0, t^1]$. The final time t^1 may

be fixed or given implicitly by the final state,

$$\begin{aligned} t^1: H^1(t^1, \mathbf{y}^1, \mathbf{z}^1, \mathbf{p}) &= 0, \quad H^1: \mathbf{R} \times \mathbf{R}^f \times \mathbf{R}^g \times \mathbf{R}^h \rightarrow \mathbf{R} \\ \dot{H}^1 &:= \frac{dH^1}{dt^1} = \frac{\partial H^1}{\partial t^1} + \frac{\partial H^1}{\partial y_i^1} v_i^1 + \frac{\partial H^1}{\partial z_j^1} \dot{z}_j^1 \neq 0 \end{aligned} \quad (9)$$

Although the functions G^1 and F depend on state variables, the function ψ is determined entirely by choosing special values for the parameters \mathbf{p} , since Eqs. 1, 2, 6, and 7 can be interpreted as implicit differential-algebraic constraints determining the state variables. Therefore, the problem of optimizing multibody systems can be reduced to a nonlinear programming problem,

$$\text{minimize}_{\mathbf{p} \in \mathbf{R}^h} f(\mathbf{p})$$

subject to constraints

$$\begin{aligned} c_i(\mathbf{p}) &= 0, \quad i \in E \\ c_i(\mathbf{p}) &\geq 0, \quad i \in I \\ p_k^l &\leq p_k \leq p_k^u, \quad k = 1 \text{ to } h \end{aligned} \quad (10)$$

where E is the index set of equality constraints and I is the index set of inequality constraints. The objective function $f(\mathbf{p})$, as well as the constraint functions $c_i(\mathbf{p})$, can be algebraic or of the type given by Eq. 8.

In recent years, many algorithms have been developed for solving such a general programming problem [3]. For good convergence of these optimization algorithms, reliable gradients of the objective function and all constraint functions with respect to parameters \mathbf{p} must be computed. For algebraic functions, the gradient can easily be found as the derivative with respect to the parameters. The sensitivity of the more important function type given by Eq. 8 to changes in the parameters must be considered in more detail.

III. SENSITIVITY ANALYSIS OF A FUNCTIONAL

Ignoring the structure of the function ψ given by Eq. 8 and regarding it as a function of \mathbf{p} only, a linear Taylor approximation yields

$$\psi(\mathbf{p} + \Delta\mathbf{p}) \approx \psi(\mathbf{p}) + \left. \frac{d\psi^T}{d\mathbf{p}} \right|_{\mathbf{p}} \Delta\mathbf{p} = \psi(\mathbf{p}) + \nabla\psi^T(\mathbf{p})\Delta\mathbf{p} \quad (11)$$

From this, the coordinates of the gradient $\nabla\psi$ can be approximated by finite differences

$$\nabla\psi_k \approx \frac{\psi(\mathbf{p} + \Delta p \mathbf{e}_k) - \psi(\mathbf{p})}{\Delta p}, \quad k = 1 \text{ to } h \quad (12)$$

where \mathbf{e}_k is a unit vector with its k -th element equal to one, and Δp is a small parameter perturbation of the k -th parameter. Such finite differences are often used in optimization as a simple way of computing the gradient of a function approximately.

Applying finite differences to dynamic systems has two disadvantages. First, approximating the whole gradient by finite differences requires h additional evaluations of the function ψ for perturbed parameters and each evaluation is a time consuming dynamic analysis, i.e., a numerical integration of the equations of motion. Second, it is not known a priori how to choose Δp . A large parameter perturbation will result in errors due to the linear approximation, and small parameter changes will result in errors due to the limited accuracy in computing the value of function ψ by numerical integration. It is desirable, therefore, to have more analytical information about the gradient.

There are two approaches for computing the gradient with additional equations [6]: the direct differentiation method and the adjoint variable method. In the direct differentiation method, the equations of motion are differentiated with respect to the parameters, resulting in differential equations for sensitivity functions of the state variables. Solving these equations requires about the same computational effort as computing finite differences. Therefore, the adjoint variable method is preferable and will be discussed in this paper.

To derive the desired relationships, it is advantageous to use the variational theory. The parameters p_k are regarded as independent, although there may be relations such as Eq. 4 that need to be considered for optimization only. Further, it is advantageous to use index notation with the summation convention; i.e., if an index occurs twice in an expression, summation over all possible values of that index is implied. The range of possible values will be clear from the context.

Regarding the function ψ again as a function of \mathbf{p} only, i.e., $\psi = \psi(\mathbf{p})$, the first variation is

$$\delta\psi = \frac{d\psi}{dp_k} \delta p_k = \nabla\psi_k \delta p_k \quad (13)$$

On the other hand, considering the structure of the function in more detail, which means considering also the dependence of ψ on the state variables y_i and z_j and the final time which must be varied, the first variation yields

$$\begin{aligned} \delta\psi = & \delta G^1 + F^1 \delta t^1 + \int_{t^0}^{t^1} \delta F dt = \left(\frac{\partial G^1}{\partial t^1} + F^1 \right) \delta t^1 + \frac{\partial G^1}{\partial y_i^1} \delta y_i^1 + \frac{\partial G^1}{\partial z_j^1} \delta z_j^1 \\ & + \frac{\partial G^1}{\partial p_k} \delta p_k + \int_{t^0}^{t^1} \left(\frac{\partial F}{\partial y_i} \delta y_i + \frac{\partial F}{\partial z_j} \delta z_j + \frac{\partial F}{\partial \dot{z}_j} \delta \dot{z}_j + \frac{\partial F}{\partial p_k} \delta p_k \right) dt \end{aligned} \quad (14)$$

where $F^1 := F(t^1, y^1, z^1, \dot{z}^1, \mathbf{p})$. The variation has produced new variations δt^1 , δy_i , δz_j , and $\delta \dot{z}_j$ that depend on the parameter variations δp_k . This dependence is clear from the fact that the state coordinates and the final time are determined by the parameters, i.e., $y = y(t, \mathbf{p})$, $z = z(t, \mathbf{p})$, $t^1 = t^1(\mathbf{p})$, and the definition of the variations,

$$\begin{aligned} \delta t^1 &= \frac{dt^1}{dp_k} \delta p_k, & \delta y_i(t) &= \frac{dy_i}{dp_k} \delta p_k, \\ \delta z_j(t) &= \frac{dz_j}{dp_k} \delta p_k, & \delta \dot{z}_j(t) &= \frac{d\dot{z}_j}{dp_k} \delta p_k \end{aligned} \quad (15)$$

The final state y^1 and z^1 depends doubly on the parameters, since both the state itself and the final time depend on \mathbf{p} ; i.e., $y^1 = y(t^1(\mathbf{p}), \mathbf{p})$, $z^1 = z(t^1(\mathbf{p}), \mathbf{p})$. The first variation, with Eq. 1, yields

$$\delta y_i^1 = \left. \frac{dy_i}{dt} \right|_{t^1} \delta t^1 + \left. \frac{dy_i}{dp_k} \right|_{t^1} \delta p_k =: v_i^1 \delta t^1 + \delta^1 y_i \quad (16)$$

$$\delta z_j^1 = \left. \frac{dz_j}{dt} \right|_{t^1} \delta t^1 + \left. \frac{dz_j}{dp_k} \right|_{t^1} \delta p_k =: \dot{z}_j^1 \delta t^1 + \delta^1 z_j \quad (17)$$

where $\delta^1 y := \delta y(t^1)$ and $\delta^1 z := \delta z(t^1)$ are variations of the state taken at the final time t^1 . With these definitions, Eq. 14 simplifies to

$$\begin{aligned} \delta\psi = & (\dot{G}^1 + F^1) \delta t^1 + \frac{\partial G^1}{\partial y_i^1} \delta^1 y_i + \int_{t^0}^{t^1} \frac{\partial F}{\partial y_i} \delta y_i dt + \frac{\partial G^1}{\partial z_j^1} \delta^1 z_j + \int_{t^0}^{t^1} \frac{\partial F}{\partial z_j} \delta z_j dt \\ & + \int_{t^0}^{t^1} \frac{\partial F}{\partial \dot{z}_j} \delta \dot{z}_j dt + \left(\frac{\partial G^1}{\partial p_k} + \int_{t^0}^{t^1} \frac{\partial F}{\partial p_k} dt \right) \delta p_k \end{aligned} \quad (18)$$

where

$$\dot{G}^1 := \frac{dG^1}{dt^1} = \frac{\partial G^1}{\partial t^1} + \frac{\partial G^1}{\partial y_i^1} v_i^1 + \frac{\partial G^1}{\partial z_j^1} \dot{z}_j^1 \quad (19)$$

Equating Eqs. 13 and 18 yields a relationship for computing the gradient $\nabla\psi$, if the dependent variations δt^1 , δy_i , δz_j , and $\delta \dot{z}_j$ can be eliminated. Implicit dependencies of these variations on the independent variations δp_k are given by first variations of the differential equations of motion (Eqs. 1 and 2) and the algebraic equations (Eqs. 9, 6, and 7) for the final time t^1 and the initial conditions,

$$\delta \dot{y} - \delta v = 0 \quad (20)$$

$$M\delta \dot{z} + \delta M\dot{z} + \delta k - \delta q = 0 \quad (21)$$

$$\delta H^1 = 0 \quad (22)$$

$$\delta \Phi^0 = 0 \quad (23)$$

$$\delta \dot{\Phi}^0 = 0 \quad (24)$$

These constraints must be multiplied by Lagrangian multipliers or adjoint variables, respectively, and added to Eq. 18. Choosing special values for the adjoint variables will then eliminate the dependent variations.

Equations 20 and 21 are differential constraints for δy and δz , respectively, which hold for the total time domain $[t^0, t^1]$. Multiplying them by arbitrary adjoint variable vectors $\mu(t)$, $\mu: [t^0, t^1] \rightarrow \mathbf{R}^f$ and $\nu(t)$, $\nu: [t^0, t^1] \rightarrow \mathbf{R}^g$, respectively, and integrating them over the time interval yields

$$\int_{t^0}^{t^1} \left[\mu_i \delta \dot{y}_i - \mu_i \frac{\partial v_i}{\partial y_i} \delta y_i - \mu_i \frac{\partial v_i}{\partial z_j} \delta z_j - \mu_i \frac{\partial v_i}{\partial p_k} \delta p_k \right] dt = 0, \quad (25)$$

for all $\mu_i(t)$

$$\int_{t^0}^{t^1} \left[\nu_m M_{mn} \delta \dot{z}_n + \nu_m \left(\frac{\partial M_{mn}}{\partial y_i} \delta y_i + \frac{\partial M_{mn}}{\partial p_k} \delta p_k \right) \dot{z}_n \right. \\ \left. + \nu_m \frac{\partial (k_m - q_m)}{\partial y_i} \delta y_i + \nu_m \frac{\partial (k_m - q_m)}{\partial z_j} \delta z_j \right. \\ \left. + \nu_m \frac{\partial (k_m - q_m)}{\partial p_k} \delta p_k \right] dt = 0, \quad \text{for all } \nu_m(t) \quad (26)$$

As mentioned above, the equations considered have been regarded as constraints on δy and δz . Thus, the total time derivatives of these variations in the first terms of both integrands must be eliminated by integrating these terms by parts,

$$\begin{aligned} \mu_i^1 \delta^1 y_i - \mu_i^0 \delta^0 y_i - \int_{t^0}^{t^1} \left[\dot{\mu}_i + \mu_i \frac{\partial v_i}{\partial y_i} \right] \delta y_i dt - \int_{t^0}^{t^1} \mu_i \frac{\partial v_i}{\partial z_j} \delta z_j dt \\ - \int_{t^0}^{t^1} \mu_i \frac{\partial v_i}{\partial p_k} dt \delta p_k = 0, \quad \text{for all } \mu_i(t) \end{aligned} \quad (27)$$

$$\begin{aligned} \nu_m^1 M_{mj}^1 \delta^1 z_j - \nu_m^0 M_{mj}^0 \delta^0 z_j - \int_{t^0}^{t^1} \left[\dot{\nu}_m M_{mj} + \nu_m \dot{M}_{mj} - \nu_m \frac{\partial(k_m - q_m)}{\partial z_j} \right] \delta z_j dt \\ + \int_{t^0}^{t^1} \nu_m \left[\frac{\partial M_{mn}}{\partial y_i} \dot{z}_n + \frac{\partial(k_m - q_m)}{\partial y_i} \right] \delta y_i dt \\ + \int_{t^0}^{t^1} \nu_m \left[\frac{\partial M_{mn}}{\partial p_k} \dot{z}_n + \frac{\partial(k_m - q_m)}{\partial p_k} \right] dt \delta p_k = 0, \quad \text{for all } \nu_m(t) \end{aligned} \quad (28)$$

where the definitions $\delta^0 \mathbf{y} := \delta \mathbf{y}(t^0)$ and $\delta^0 \mathbf{z} := \delta \mathbf{z}(t^0)$ are used.

Equation 21 must be considered twice, since it is not only a differential constraint for $\delta \mathbf{z}$ but also an algebraic constraint for $\delta \dot{\mathbf{z}}$. Multiplying it by another adjoint variable vector $\xi(t)$, $\xi: [t^0, t^1] \rightarrow \mathbf{R}^s$, without integrating it by parts, yields a relationship similar to Eq. 26,

$$\begin{aligned} \int_{t^0}^{t^1} \xi_m \left[\frac{\partial M_{mn}}{\partial y_i} \dot{z}_n + \frac{\partial(k_m - q_m)}{\partial y_i} \right] \delta y_i dt + \int_{t^0}^{t^1} \xi_m \frac{\partial(k_m - q_m)}{\partial z_j} \delta z_j dt \\ + \int_{t^0}^{t^1} \xi_m M_{mj} \delta \dot{z}_j dt + \int_{t^0}^{t^1} \xi_m \left[\frac{\partial M_{mn}}{\partial p_k} \dot{z}_n + \frac{\partial(k_m - q_m)}{\partial p_k} \right] dt \delta p_k = 0, \\ \text{for all } \xi_m(t) \end{aligned} \quad (29)$$

Equations 22 to 24 hold only for a fixed time. They can be multiplied by Lagrangian multipliers $\tau^1 \in \mathbf{R}$, $\xi^0 \in \mathbf{R}^f$, and $\eta^0 \in \mathbf{R}^s$, respectively, which

will also be called adjoint variables in the following:

$$\tau^1 \dot{H}^1 \delta t^1 + \tau^1 \frac{\partial H^1}{\partial y_i^1} \delta^1 y_i + \tau^1 \frac{\partial H^1}{\partial z_j^1} \delta^1 z_j + \tau^1 \frac{\partial H^1}{\partial p_k} \delta p_k = 0, \quad \text{for all } \tau^1 \quad (30)$$

$$\zeta_i^0 \frac{\partial \Phi_l^0}{\partial y_i^0} \delta^0 y_i + \zeta_i^0 \frac{\partial \Phi_l^0}{\partial p_k} \delta p_k = 0, \quad \text{for all } \zeta_i^0 \quad (31)$$

$$\eta_m^0 \frac{\partial \dot{\Phi}_m^0}{\partial y_i^0} \delta^0 y_i + \eta_m^0 \frac{\partial \dot{\Phi}_m^0}{\partial z_j^0} \delta^0 z_j + \eta_m^0 \frac{\partial \dot{\Phi}_m^0}{\partial p_k} \delta p_k = 0, \quad \text{for all } \eta_m^0 \quad (32)$$

For deriving these relations, consider Eqs. 16 and 17, $\delta \mathbf{y}^0 = \delta^0 \mathbf{y}$, and $\delta \mathbf{z}^0 = \delta^0 \mathbf{z}$, which is implied by assuming that initial time t^0 is fixed and does not depend on the parameters.

Now, Eqs. 27 to 32 can be subtracted from Eq. 18, without changing the value of $\delta \psi$, for any choice of the adjoint variables, to obtain

$$\begin{aligned} \delta \psi = & [\dot{G}^1 + F^1 - \tau^1 \dot{H}^1] \delta t^1 \\ & + \left[\frac{\partial G^1}{\partial y_i^1} - \tau^1 \frac{\partial H^1}{\partial y_i^1} - \mu_i^1 \right] \delta^1 y_i + \left[\mu_i^0 - \zeta_i^0 \frac{\partial \Phi_l^0}{\partial y_i^0} - \eta_m^0 \frac{\partial \dot{\Phi}_m^0}{\partial y_i^0} \right] \delta^0 y_i \\ & + \int_{t^0}^{t^1} \left[\frac{\partial F}{\partial y_i} + \dot{\mu}_i + \mu_i \frac{\partial v_i}{\partial y_i} - (v_m + \xi_m) \left(\frac{\partial M_{mn}}{\partial y_i} \dot{z}_n + \frac{\partial (k_m - q_m)}{\partial y_i} \right) \right] \delta y_i dt \\ & + \left[\frac{\partial G^1}{\partial z_j^1} - \tau^1 \frac{\partial H^1}{\partial z_j^1} - \nu_m^1 M_{mj}^1 \right] \delta^1 z_j + \left[\nu_m^0 M_{mj}^0 - \eta_m^0 \frac{\partial \dot{\Phi}_m^0}{\partial z_j^0} \right] \delta^0 z_j \\ & + \int_{t^0}^{t^1} \left[\frac{\partial F}{\partial z_j} + \mu_i \frac{\partial v_i}{\partial z_j} + \dot{\nu}_m M_{mj} + \nu_m \dot{M}_{mj} - (v_m + \xi_m) \frac{\partial (k_m - q_m)}{\partial z_j} \right] \delta z_j dt \\ & + \int_{t^0}^{t^1} \left[\frac{\partial F}{\partial \dot{z}_j} - \xi_m M_{mj} \right] \delta \dot{z}_j dt + \left\{ \frac{\partial G^1}{\partial p_k} - \tau^1 \frac{\partial H^1}{\partial p_k} - \zeta_i^0 \frac{\partial \Phi_l^0}{\partial p_k} - \eta_m^0 \frac{\partial \dot{\Phi}_m^0}{\partial p_k} \right. \\ & \left. + \int_{t^0}^{t^1} \left[\frac{\partial F}{\partial p_k} + \mu_i \frac{\partial v_i}{\partial p_k} - (v_m + \xi_m) \left(\frac{\partial M_{mn}}{\partial p_k} \dot{z}_n + \frac{\partial (k_m - q_m)}{\partial p_k} \right) \right] dt \right\} \delta p_k \end{aligned} \quad (33)$$

The aim of this procedure is to eliminate all dependent variations. This can be achieved by choosing the adjoint variables in such a way that the respective coefficients vanish. From Eq. 33,

$$\tau^1 = \frac{\dot{G}^1 + F^1}{H^1} \quad (34)$$

$$\mu_i^1 = \frac{\partial G^1}{\partial y_i^1} - \tau^1 \frac{\partial H^1}{\partial y_i^1} \quad (35)$$

$$\nu_m^1 M_{mj}^1 = \frac{\partial G^1}{\partial z_j^1} - \tau^1 \frac{\partial H^1}{\partial z_j^1} \quad (36)$$

$$\xi_m M_{mj} = \frac{\partial F}{\partial \dot{z}_j} \quad (37)$$

$$\dot{\mu}_i = -\mu_i \frac{\partial v_i}{\partial y_i} + (\nu_m + \xi_m) \left(\frac{\partial M_{mn}}{\partial y_i} \dot{z}_n + \frac{\partial(k_m - q_m)}{\partial y_i} \right) - \frac{\partial F}{\partial y_i} \quad (38)$$

$$\begin{aligned} \dot{\nu}_m M_{mj} = & -\mu_i \frac{\partial v_i}{\partial z_j} - \nu_m \left(\frac{\partial M_{mj}}{\partial t} + \frac{\partial M_{mj}}{\partial y_i} v_i \right) \\ & + (\nu_m + \xi_m) \frac{\partial(k_m - q_m)}{\partial z_j} - \frac{\partial F}{\partial z_j} \end{aligned} \quad (39)$$

$$\eta_m^0 \frac{\partial \dot{\Phi}_m^0}{\partial z_j^0} = \nu_m^0 M_{mj}^0 \quad (40)$$

$$\zeta_i^0 \frac{\partial \dot{\Phi}_i^0}{\partial y_i^0} = \mu_i^0 - \eta_m^0 \frac{\partial \dot{\Phi}_m^0}{\partial y_i^0} \quad (41)$$

With this choice, Eq. 33 can be equated to Eq. 13. Since the remaining parameter variations δp_k are independent, this yields the desired relationship for computing the gradient of ψ with respect to \mathbf{p} ,

$$\begin{aligned} \nabla \psi_k = & \frac{d\psi}{dp_k} = \frac{\partial G^1}{\partial p_k} - \tau^1 \frac{\partial H^1}{\partial p_k} - \zeta_i^0 \frac{\partial \dot{\Phi}_i^0}{\partial p_k} - \eta_m^0 \frac{\partial \dot{\Phi}_m^0}{\partial p_k} \\ & + \int_{t^0}^{t^1} \left[\frac{\partial F}{\partial p_k} + \mu_i \frac{\partial v_i}{\partial p_k} - (\nu_m + \xi_m) \left(\frac{\partial M_{mn}}{\partial p_k} \dot{z}_n + \frac{\partial(k_m - q_m)}{\partial p_k} \right) \right] dt \end{aligned} \quad (42)$$

Assuming symmetry of the mass matrix, these equations can also be written in the more convenient matrix notation,

$$\tau^1 = \frac{\dot{G}^1 + F^1}{\dot{H}^1} \quad (43)$$

$$\boldsymbol{\mu}^1 = \frac{\partial G^1}{\partial \mathbf{y}^1} - \tau^1 \frac{\partial H^1}{\partial \mathbf{y}^1} \quad (44)$$

$$\mathbf{M}^1 \mathbf{v}^1 = \frac{\partial G^1}{\partial \mathbf{z}^1} - \tau^1 \frac{\partial H^1}{\partial \mathbf{z}^1} \quad (45)$$

$$\mathbf{M} \boldsymbol{\xi} = \frac{\partial F}{\partial \dot{\mathbf{z}}} \quad (46)$$

$$\dot{\boldsymbol{\mu}} = - \frac{\partial \mathbf{v}^T}{\partial \mathbf{y}} \boldsymbol{\mu} + \frac{\partial (\mathbf{M} \dot{\mathbf{z}} + \mathbf{k} - \mathbf{q})^T}{\partial \mathbf{y}} (\mathbf{v} + \boldsymbol{\xi}) - \frac{\partial F}{\partial \mathbf{y}} \quad (47)$$

$$\mathbf{M} \dot{\mathbf{v}} = - \frac{\partial \mathbf{v}^T}{\partial \mathbf{z}} \boldsymbol{\mu} - \dot{\mathbf{M}} \mathbf{v} + \frac{\partial (\mathbf{k} - \mathbf{q})^T}{\partial \mathbf{z}} (\mathbf{v} + \boldsymbol{\xi}) - \frac{\partial F}{\partial \mathbf{z}} \quad (48)$$

$$\frac{\partial \dot{\Phi}^{0T}}{\partial \mathbf{z}^0} \boldsymbol{\eta}^0 = \mathbf{M}^0 \mathbf{v}^0 \quad (49)$$

$$\frac{\partial \Phi^{0T}}{\partial \mathbf{y}^0} \boldsymbol{\zeta}^0 = \boldsymbol{\mu}^0 - \frac{\partial \dot{\Phi}^{0T}}{\partial \mathbf{y}^0} \boldsymbol{\eta}^0 \quad (50)$$

$$\begin{aligned} \nabla \psi = \frac{d\psi}{d\mathbf{p}} &= \frac{\partial G^1}{\partial \mathbf{p}} - \tau^1 \frac{\partial H^1}{\partial \mathbf{p}} - \frac{\partial \Phi^{0T}}{\partial \mathbf{p}} \boldsymbol{\zeta}^0 - \frac{\partial \dot{\Phi}^{0T}}{\partial \mathbf{p}} \boldsymbol{\eta}^0 \\ &+ \int_{\rho}^{t^1} \left[\frac{\partial F}{\partial \mathbf{p}} + \frac{\partial \mathbf{v}^T}{\partial \mathbf{p}} \boldsymbol{\mu} - \frac{\partial (\mathbf{M} \dot{\mathbf{z}} + \mathbf{k} - \mathbf{q})^T}{\partial \mathbf{p}} (\mathbf{v} + \boldsymbol{\xi}) \right] dt \quad (51) \end{aligned}$$

The set of equations must be solved in the order given above. First, the algebraic equations (Eqs. 43–45) must be solved for τ^1 , $\boldsymbol{\mu}^1$, and \mathbf{v}^1 . Then, the differential equations for $\boldsymbol{\mu}$ and \mathbf{v} can be integrated backward in time, and Eq. 46 must be solved simultaneously, since the adjoint variables $\boldsymbol{\xi}_m$ play the role of substitution variables. Finally, $\boldsymbol{\eta}^0$ and $\boldsymbol{\zeta}^0$ can be computed from Eqs. 49 and 50, successively.

For linear holonomic time-variant systems

$$\dot{\mathbf{y}} = \mathbf{z} \quad (52)$$

$$\mathbf{M}(t, \mathbf{p}) \dot{\mathbf{z}} + \mathbf{P}(t, \mathbf{p}) \mathbf{z} + \mathbf{Q}(t, \mathbf{p}) \mathbf{y} = \mathbf{h}(t, \mathbf{p}) \quad (53)$$

Eqs. 47, 48, and 51 simplify to

$$\dot{\boldsymbol{\mu}} = \mathbf{Q}^T(\mathbf{v} + \boldsymbol{\xi}) - \frac{\partial F}{\partial \mathbf{y}} \quad (54)$$

$$\mathbf{M}\dot{\mathbf{v}} = -\boldsymbol{\mu} - \frac{\partial \mathbf{M}}{\partial t} \mathbf{v} + \mathbf{P}^T(\mathbf{v} + \boldsymbol{\xi}) - \frac{\partial F}{\partial \mathbf{z}} \quad (55)$$

$$\begin{aligned} \nabla \psi = & \frac{\partial G^1}{\partial \mathbf{p}} - \tau^1 \frac{\partial H^1}{\partial \mathbf{p}} - \frac{\partial \Phi^{0T}}{\partial \mathbf{p}} \boldsymbol{\zeta}^0 - \frac{\partial \dot{\Phi}^{0T}}{\partial \mathbf{p}} \boldsymbol{\eta}^0 \\ & + \int_{t^0}^{t^1} \left[\frac{\partial F}{\partial \mathbf{p}} - \frac{\partial (\mathbf{M}\dot{\mathbf{z}} + \mathbf{P}\mathbf{z} + \mathbf{Q}\mathbf{y} - \mathbf{h})^T}{\partial \mathbf{p}} (\mathbf{v} + \boldsymbol{\xi}) \right] dt \quad (56) \end{aligned}$$

IV. COMPUTATIONAL ALGORITHMS

Forward integration of the equations of motion and backward integration of the adjoint equations must be considered as a unit. Since the adjoint equations depend on the state coordinates, enough information must be stored during forward integration for reconstructing the trajectories of the state coordinates when integrating the adjoint equations backward in time.

A. Forward Integration of the Equations of Motion

Forward integration is started by computing the initial values of the state from the initial conditions in Eqs. 6 and 7. If they cannot be solved explicitly for \mathbf{y}^0 and \mathbf{z}^0 , an iterative algorithm must be applied. Subsequently, the equations of motion can be solved by numerical integration algorithms. It is advantageous to evaluate the functional in Eq. 8 simultaneously as the solution of the following initial-value problem:

$$\begin{aligned} \dot{\boldsymbol{\psi}} &= F(t, \mathbf{y}, \mathbf{z}, \dot{\mathbf{z}}, \mathbf{p}), \quad \bar{\boldsymbol{\psi}}(t^0) = 0, \quad t \in [t^0, t^1], \\ \boldsymbol{\psi}(\mathbf{p}) &= G^1(t^1, \mathbf{y}^1, \mathbf{z}^1, \mathbf{p}) + \bar{\boldsymbol{\psi}}(t^1) \quad (57) \end{aligned}$$

Thus, evaluating the function ψ once implies the numerical integration of $f + g + 1$ ordinary differential equations over $[t^0, t^1]$.

In general, the equations of motion for technical systems are complex and, therefore, it is time consuming to evaluate the right sides of the equations. Multi-step integration algorithms have been shown to use the least number of function evaluations of all common algorithms for intermediate to high accuracy requirements. For this reason, the integration algorithm STEP described by Shampine and Gordon [8] is applied. Controlling the integration error is only possible by using step-size control. Thus, the forward integration will generate a mesh with unequally spaced mesh points, and the state coordinates will be known at these time points only. For local error control, the user must provide tolerances ε_{rel} and ε_{abs} for relative and absolute errors in the state coordinates, respectively. Then, the local error e_i in the i -th state coordinate x_i will be limited to

$$|e_i| \leq |x_i| \varepsilon_{\text{rel}} + \varepsilon_{\text{abs}} \quad (58)$$

The final time t^1 is found by checking the sign of H^1 after each step. If H^1 changes sign during a step, there must be a solution of the final condition (Eq. 9) in this last step interval. An approximation for t^1 is computed by applying a root solver, and the final state is approximated by interpolation [8].

B. Backward Integration of the Adjoint Equations

Applying the same integration algorithm to the adjoint equations will generate a mesh that does not coincide with the mesh given by forward integration. Although it can be shown that the adjoint equations with backward integration and the equations of motion locally have the same eigenvalues, the mesh will be different, due to the different terms on the right sides of the respective equations. Thus, interpolation of the trajectories of the state coordinates cannot be circumvented.

Due to this interpolation, multi-step integration methods also have an edge on single-step algorithms. Multi-step methods already use an interpolating polynomial supported by values given at the mesh points and the optimal order of the polynomial is estimated from error calculations, whereas single-step methods use intermediate points that cannot be used for interpolation purposes.

In order to combine the forward and backward integration needed for sensitivity analysis with the adjoint variable approach, the following information from forward integration must be stored:

$$[t^{(i)}, \mathbf{y}^{(i)}, \mathbf{z}^{(i)}, \dot{\mathbf{z}}^{(i)}, o^{(i)}], \quad t^{(0)} = t^0, \quad i = 0, 1, 2, \dots \quad (59)$$

where $o^{(i)}$ is the order of the interpolating polynomial for computing the state at mesh point $t^{(i)}$. If the state for a given time point $t \in [t^{(i-1)}, t^{(i)}]$ must be reconstructed, it is good to use a polynomial of same order for interpolation. The Aitken-Neville interpolation algorithm has proven to be efficient and reliable, and therefore is used in this application. The polynomials of order $o^{(i)}$ for $\mathbf{y}(t)$, $\mathbf{z}(t)$, and $\dot{\mathbf{z}}(t)$ are computed from the values of the state coordinates at $o^{(i)} + 1$ mesh points $\{t^{(i)}, t^{(i-1)}, \dots, t^{(i-o^{(i)})}\}$.

To be more precise, let t_i , $i = 0$ to n , be a mesh for interpolating a scalar function $f(t)$ at a given point $t = \tau$. Then, the algorithm successively computes higher order polynomial approximations, starting from the function values at the mesh points,

$$P_{ik} = P_{i,k-1} + \frac{t_i - \tau}{t_{i-k} - t_i} (P_{i,k-1} - P_{i-1,k-1}), \quad i = 0 \text{ to } n, \quad k = 1 \text{ to } i \quad (60)$$

where $P_{i0} := f(t_i)$ are polynomials of zero order and the result P_{nn} is an approximation of order n for $f(\tau)$. To avoid cancellation errors, the formula is used in a slightly altered form [9].

Similar to the combination of forward integration of the equations of motion and performance evaluation, the gradient of ψ can also be computed simultaneously with the backward integration of Eqs. 47 and 48. The gradient (Eq. 51) can be written as

$$\nabla\psi = \overline{\nabla\psi}(t^0) - \frac{\partial\Phi^{0T}}{\partial\mathbf{p}} \boldsymbol{\zeta}^0 - \frac{\partial\dot{\Phi}^{0T}}{\partial\mathbf{p}} \boldsymbol{\eta}^0 \quad (61)$$

where

$$\begin{aligned} \overline{\nabla\psi}(t) &= \frac{\partial G^1}{\partial\mathbf{p}} - \tau^1 \frac{\partial H^1}{\partial\mathbf{p}} \\ &+ \int_t^{t^1} \left[\frac{\partial F}{\partial\mathbf{p}} + \frac{\partial\mathbf{v}^T}{\partial\mathbf{p}} \boldsymbol{\mu} - \frac{\partial(\mathbf{M}\dot{\mathbf{z}} + \mathbf{k} - \mathbf{q})^T}{\partial\mathbf{p}} (\mathbf{v} + \boldsymbol{\xi}) \right] d\tau \\ &\overline{\nabla\psi} : [t^0, t^1] \rightarrow \mathbf{R}^h \end{aligned} \quad (62)$$

Differentiating $\overline{\nabla\psi}(t)$ with respect to time yields a final value problem,

$$\begin{aligned} \dot{\overline{\nabla\psi}} &= -\frac{\partial F}{\partial\mathbf{p}} - \frac{\partial\mathbf{v}^T}{\partial\mathbf{p}} \boldsymbol{\mu} + \frac{\partial(\mathbf{M}\dot{\mathbf{z}} + \mathbf{k} - \mathbf{q})^T}{\partial\mathbf{p}} (\mathbf{v} + \boldsymbol{\xi}), \\ \overline{\nabla\psi}(t^1) &= \frac{\partial G^1}{\partial\mathbf{p}} - \tau^1 \frac{\partial H^1}{\partial\mathbf{p}} \end{aligned} \quad (63)$$

which can be solved by backward integration similar to that used in the adjoint equations. With the resulting value $\overline{\nabla\psi}(t^0)$, the gradient can be computed from Eq. 61.

Summarizing the computational effort for evaluating the gradient of a functional ψ , in addition to integrating the equations of motion, a system of $f + g + h$ ordinary differential equations, where h is the dimension of the parameter space, must be integrated backward in time. Thus, a complete analysis and sensitivity analysis by the adjoint variable approach takes the time of a numerical integration of $2(f + g) + (h + 1)$ first-order differential equations over the time domain $[t^0, t^1]$. For approximating the gradient by finite differences, $(h + 1)(f + g + 1)$ first-order differential equations must be solved. From the point of view of computational effort, the adjoint variable method must be preferred if many parameters are to be varied.

C. Computing a Reference Gradient

Before using an optimization algorithm, it is always advisable to check the gradient in such a way that the check is independent of additional calculations necessary for the adjoint variable approach. Moreover, in this paper reliable reference values are required for estimating the errors of the adjoint variable approach. As will be seen from the examples, using finite differences as a zero-order approximation of the gradient is not sufficiently reliable for this purpose.

From a Taylor series

$$\psi(\mathbf{p} + \Delta p \mathbf{e}_k) = \psi(\mathbf{p}) + \left. \frac{d\psi}{dp_k} \right|_{\mathbf{p}} \Delta p + \frac{1}{2} \left. \frac{d^2\psi}{dp_k^2} \right|_{\mathbf{p}} \Delta p^2 + \dots \quad (64)$$

the finite difference

$$D(\Delta p) := \frac{\psi(\mathbf{p} + \Delta p \mathbf{e}_k) - \psi(\mathbf{p})}{\Delta p} = \nabla\psi_k + D_1\Delta p + D_2\Delta p^2 + \dots \quad (65)$$

is obtained, being a function of Δp with $D(0) = \nabla\psi_k$, where

$$D_i := \frac{1}{(i + 1)!} \left. \frac{d^{i+1}\psi}{dp_k^{i+1}} \right|_{\mathbf{p}}, \quad i = 1, 2, \dots \quad (66)$$

Thus, approximating $D(\Delta p)$ by a polynomial of order n and extrapolating it to $\Delta p = 0$ will give an approximation for $\nabla\psi_k$, with an error of order $n + 1$. The approximating polynomial can be found from a mesh of $n + 1$ pa-

parameter perturbations $\Delta p^{(i)}$, $i = 0$ to n , and $D(0)$ can be computed with the Aitken-Neville algorithm for $\tau = 0$. Choosing $\Delta p^{(i)} = \Delta p^{(0)}/(i + 1)$,

$$P_{ik} = P_{i,k-1} + \frac{1 + i - k}{k} (P_{i,k-1} - P_{i-1,k-1}), \quad i = 0 \text{ to } n, \quad k = 1 \text{ to } i \quad (67)$$

where $\Delta p^{(0)}$ is a user-defined maximum perturbation and the zero-order approximations $P_{i0} := D(\Delta p^{(i)})$ must be computed as finite differences. The order n can be increased, beginning with one, until a stopping criterion such as

$$|P_{nn} - P_{n,n-1}| \leq \varepsilon^* \quad (68)$$

is satisfied, where $\varepsilon^* > 0$ is a given error tolerance. The maximum order should be restricted to avoid infinite iteration if the algorithm cannot converge because of numerical errors or because $\Delta p^{(0)}$ is too large.

V. NUMERICAL EXAMPLES

The proposed method is applied to two dynamic systems, a simple oscillator with one degree of freedom and a model of a manipulator with five degrees of freedom. The sensitivity of the solution to changes in several parameters is calculated and the results are compared to results computed by finite differences. All numerical experiments are carried out on an APOLLO DN3000 workstation with a roundoff error of about 10^{-16} .

A. Simple Oscillator

The oscillator of Fig. 2 is described by linear differential equations

$$\dot{y} = z, \quad m\dot{z} = -cy \quad (69)$$

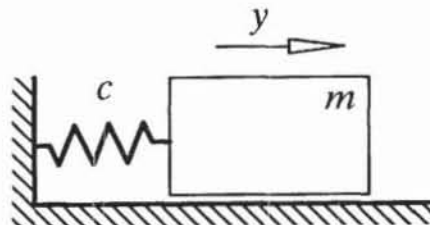


Fig. 2 Simple oscillator.

and initial conditions

$$y(0) = 0, \quad z(0) = z_0 \quad (70)$$

The objective function is the position of the oscillator for a given final time $t^1 = \pi/2$; i.e. [6],

$$\psi := y^1 \quad (71)$$

The stiffness coefficient c and the initial velocity z_0 are chosen to be variable; i.e., $\mathbf{p} = [c, z_0]^T \in \mathbf{R}^2$. The mass m may be a constant, $m = 1$.

An analytical solution is available for this simple oscillator. Thus, the example is well-suited to testing the accuracy of both the adjoint variable approach and the approximation by finite differences. With the solution for the position,

$$y(t) = z_0 \sqrt{\frac{m}{c}} \sin\left(\sqrt{\frac{c}{m}} t\right) \quad (72)$$

the sensitivity of the objective function of Eq. 71 is

$$\nabla\psi_1 = \frac{d\psi}{dc} = \frac{z_0}{2c} \left[\frac{\pi}{2} \cos\left(\sqrt{\frac{c}{m}} \frac{\pi}{2}\right) - \sqrt{\frac{m}{c}} \sin\left(\sqrt{\frac{c}{m}} \frac{\pi}{2}\right) \right] \quad (73)$$

$$\nabla\psi_2 = \frac{d\psi}{dz_0} = \sqrt{\frac{m}{c}} \sin\left(\sqrt{\frac{c}{m}} \frac{\pi}{2}\right) \quad (74)$$

The deviations of the results computed by the adjoint variable method and the finite difference approach from these exact solutions are considered as errors,

$$e_{\text{adj}k} := |\nabla\psi_{\text{adj}k} - \nabla\psi_k| \quad \text{and} \quad e_{\text{fin}k} := |\nabla\psi_{\text{fin}k} - \nabla\psi_k|, \quad k = 1, 2 \quad (75)$$

respectively. The reference parameter point for numerical experiments is $\mathbf{p} = [1, 0.5]^T$, for which the exact solutions are $\psi = 0.5$, $\nabla\psi_1 = -0.25$, and $\nabla\psi_2 = 1$.

Figure 3 shows the errors of all three proposed methods for a fixed integration tolerance $\varepsilon_{\text{rel}} = \varepsilon_{\text{abs}} = 10^{-9}$. As expected, sensitivity analysis with finite differences (Eq. 12) has an optimal parameter perturbation where er-

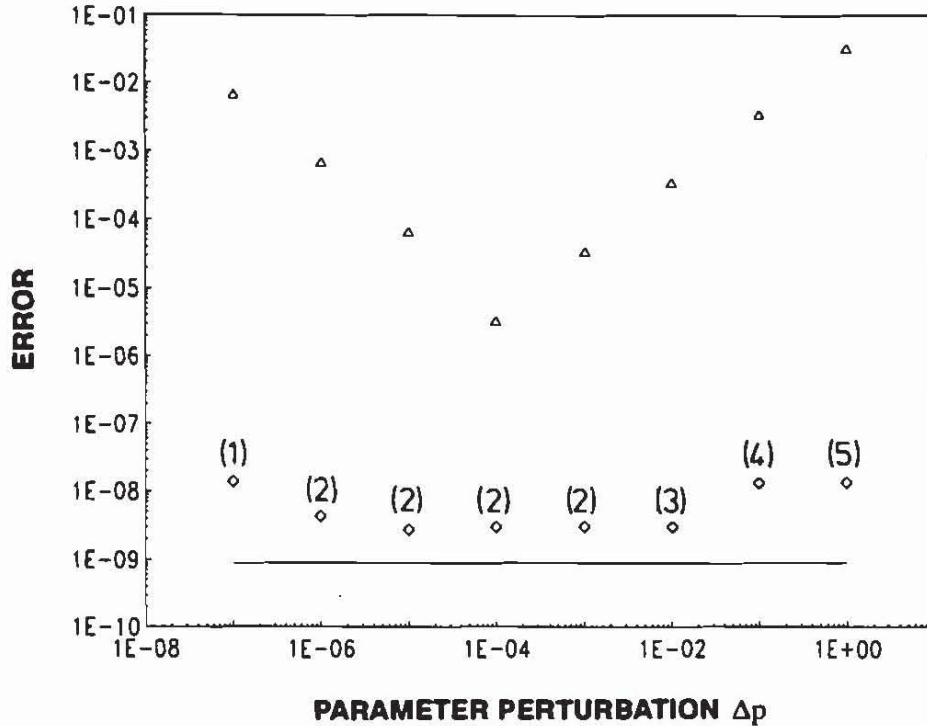


Fig. 3 Errors of sensitivity analysis for simple oscillator; deviations from $\nabla\psi_1$ are shown.

- adjoint variable approach
- \triangle finite differences
- \diamond extrapolation procedure

errors are minimal. However, even for this optimal perturbation, the error is much higher than the error e_{adj1} of the adjoint variable approach, which is of the same order as the integration tolerance. For the extrapolation procedure described in section IV.C, the user-defined starting perturbation $\Delta p^{(0)}$ is chosen to be Δp . The numbers in brackets indicate the order of the approximating polynomial for $D(\Delta p)$. The tolerance for the stopping criterion (Eq. 68) is $\varepsilon^* = 10^{-8}$. The error is nearly independent of $\Delta p^{(0)}$ and almost as low as the error made by the adjoint variable approach. Thus, the extrapolation procedure seems to give a reliable reference gradient for checking the adjoint variable approach.

It is also interesting to note the dependence of the errors on the integration tolerance, $\varepsilon := \varepsilon_{rel} = \varepsilon_{abs}$. From Fig. 4, it can be seen that the error e_{adjk} of the adjoint variable approach is always lower than the integration tolerance, whereas even the best finite difference approximation for a certain given integration tolerance is worse than e_{adjk} . Increasing Δp in Fig. 4b seems to decrease the error in the finite difference approximation of $\nabla\psi_2$. This unusual

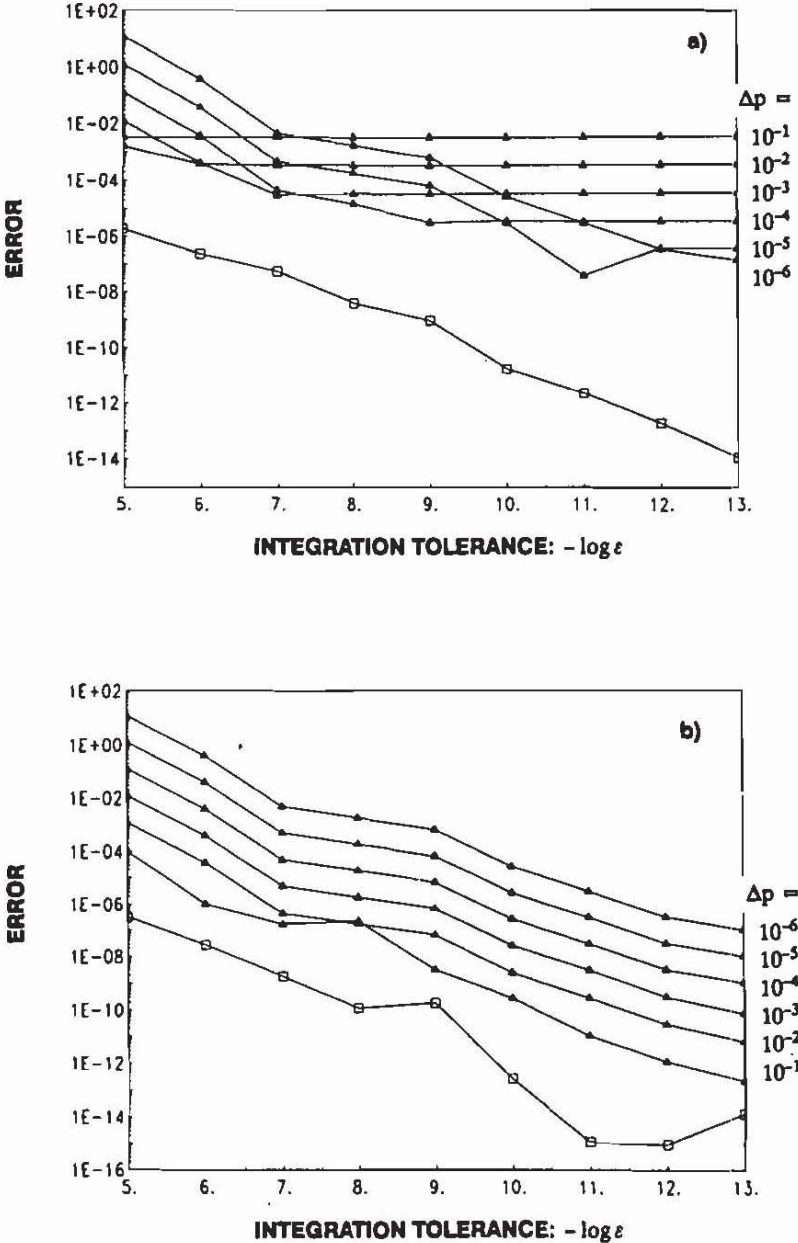


Fig. 4 Sensitivity analysis of simple oscillator with different error tolerances for $\nabla\psi_1$ (a) and $\nabla\psi_2$ (b).
□ adjoint variable approach
△ finite difference approximation for several parameter perturbations

behavior is due to the fact that the function ψ depends linearly on the parameter z_0 and, consequently, the zero-order approximation is exact. Thus, increasing Δp reduces the influence of errors in ψ on the finite differences.

Further numerical experiments have shown comparable results for several different performance measures that also include velocity and acceleration coordinates. In no case has loss of accuracy been observed with the adjoint variable approach.

B. Manipulator

As a second example, the nonlinear model of a manipulator with five degrees of freedom is considered. The model shown in Fig. 5 is already too complex for generating the equations of motion and the adjoint equations with pencil and paper. Therefore, the program NEWEUL [10] was used for computing the equations of motion symbolically and symbolic computation of the adjoint equations was supported by the program MAPLE [11].

The position of the robot is described by five generalized coordinates; i.e., $\mathbf{y} = [Z1, GA1, Y2, BE2, AL3]^T$. The generalized coordinate velocity vector is chosen to be $\mathbf{z} = \dot{\mathbf{y}}$. The manipulator is described in detail in Ref. 1 as a test example for comparing several multibody system codes. Deviating from that model, in this paper the applied forces are chosen to be time invariant; i.e., $F2Y = L2Y = L3X = 0$, $F1Z$ and $L1Z$ being variable param-

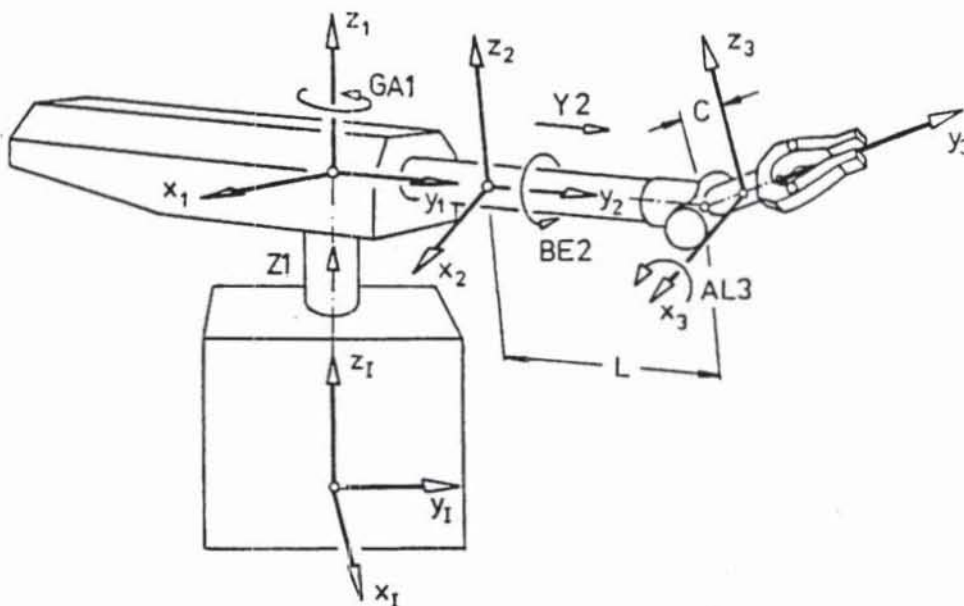


Fig. 5 Manipulator.

eters. Furthermore, the final time t^1 , dimension L , mass $M2$ of body two, and the moment of inertia $I3Z$ are chosen as variable parameters of the system; i.e. $\mathbf{p} = [F1Z, L1Z, t^1, L, M2, I3Z]^T$. The initial conditions are set to $\mathbf{y}_0 = \mathbf{z}_0 = \mathbf{0}$.

The objective is to find the sensitivity of the motion to changes in several parameters of the system. Let \mathbf{r}_3^1 be the end position of the center of body three for given parameters \mathbf{p} , and let \mathbf{r}_3^{1*} be a given reference position that is achieved for parameters $\mathbf{p}^* = [5000, 200, 0.5, 0.5, 150, 4.3]^T$. Then, the objective function is defined by

$$\psi := \|\mathbf{r}_3^1 - \mathbf{r}_3^{1*}\|_2^2 = (\mathbf{r}_3^1 - \mathbf{r}_3^{1*})^T (\mathbf{r}_3^1 - \mathbf{r}_3^{1*}) \quad (76)$$

An interesting point is the accuracy of the adjoint variable approach for this complex nonlinear dynamic system compared to the finite difference approximation. The reference values of the gradient are computed by the adjoint variable method and the extrapolation procedure with smallest error tolerances possible. For the parameters $\mathbf{p} = [4500, 220, 0.5, 0.5, 155, 4.2]^T$,

$$\nabla\psi = \begin{bmatrix} -0.6597 \dots \times 10^{-4} \pm 0.6 \times 10^{-15} \\ 0.9200 \dots \times 10^{-5} \pm 0.5 \times 10^{-15} \\ 0.1438 \dots \pm 0.1 \times 10^{-12} \\ 0.6811 \dots \times 10^{-2} \pm 0.4 \times 10^{-11} \\ 0.5922 \dots \times 10^{-3} \pm 0.5 \times 10^{-13} \\ -0.1837 \dots \times 10^{-4} \pm 0.6 \times 10^{-14} \end{bmatrix} \quad (77)$$

where the confidence interval results from the differences of both computations. Figure 6 shows the same qualitative dependence of the deviations on the error tolerance as for the simple oscillator.

A second interesting comparison can be made regarding numerical efficiency. For this purpose, the number of variable parameters is increased from one to six, adding variable parameters in the sequence given above. Figure 7 shows execution times for completing an analysis and computing the sensitivity information. The execution time for analysis of dynamic behavior is taken to be one time unit. As expected, approximating the gradient by finite differences takes one time unit for every parameter to be varied. Small deviations are due to the slightly changed dynamic behavior for perturbed parameters. In contrast, the increase in execution time due to consideration of more variable parameters is much less than one time unit per parameter for the adjoint variable approach. However, it should be noted

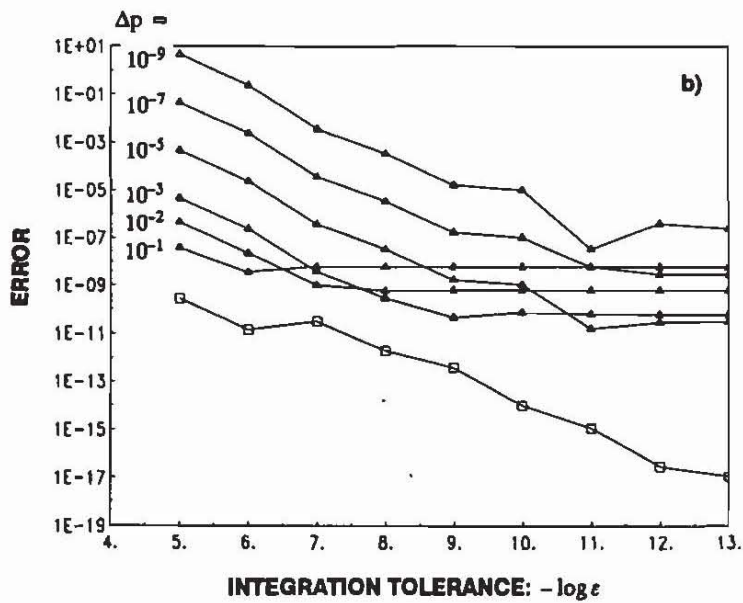
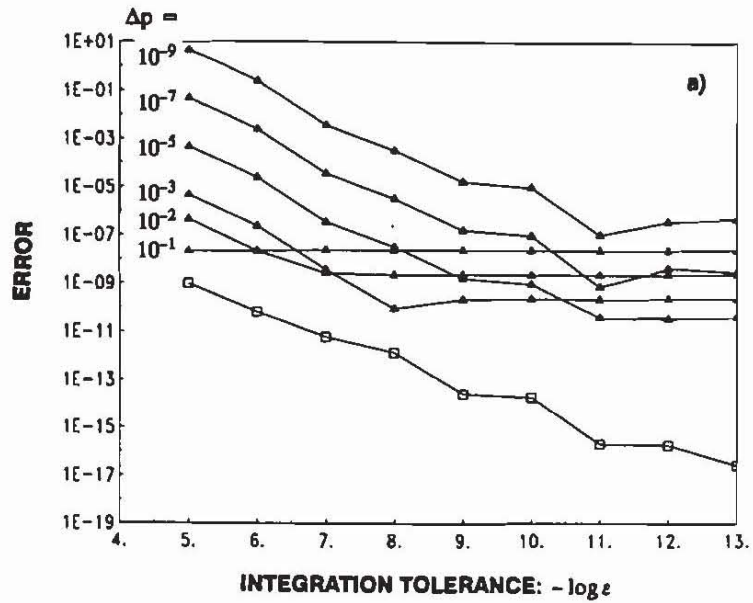


Fig. 6 Sensitivity analysis of manipulator with different error tolerances for $\nabla\psi_1$ (a) and $\nabla\psi_2$ (b).

□ adjoint variable approach

△ finite difference approximation for several parameter perturbations

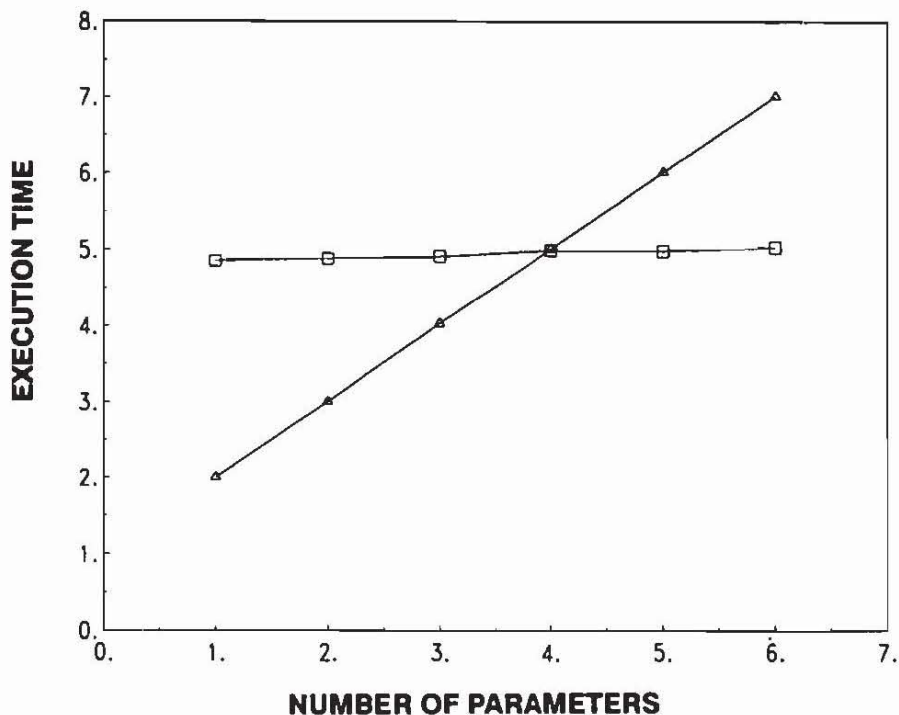


Fig. 7 Execution time for sensitivity analysis of manipulator.

- adjoint variable approach
- △ finite difference approximation

that the solution of the adjoint equations already takes a considerable amount of execution time, independent of the number of parameters.

VI. CONCLUSIONS

The optimization of multibody systems is formulated as a nonlinear programming problem under rather general assumptions. The missing link between multibody dynamic analysis codes for computing dynamic behavior and optimization codes is the sensitivity analysis of the objective and constraint functions with respect to parameter perturbations. For this purpose, the adjoint variable approach is applied to the first-order differential equations of motion for holonomic as well as non-holonomic multibody systems. With this method, an additional set of ordinary differential equations is generated, which can be solved by numerical backward integration with little more computational effort than the analysis of the dynamic behavior requires.

The combined forward and backward integration is performed by a multi-step integration algorithm with order and step-size control. Interpolations are carried out by an Aitken-Neville interpolation algorithm.

Applications of the proposed method to two examples, a simple oscillator and a robot system, show much higher accuracy of the adjoint variable method than can be achieved with the finite difference approximation. Although the method has been tested only for two examples, it seems to compute reliable gradients, which are necessary for good convergence of optimization algorithms.

REFERENCES

1. W. Schiehlen (ed.), *Multibody Systems Handbook*, Springer-Verlag, Berlin, 1990.
2. R. Fletcher, *Practical Methods of Optimization*, Wiley, Chichester, 1987.
3. K. Schittkowski, *Nonlinear Programming Codes: Information, Test, Performance*, Springer-Verlag, Berlin, 1980.
4. E. J. Haug and J. S. Arora, *Applied Optimal Design*, Wiley, New York, 1979.
5. E. J. Haug, K. K. Choi and V. Komkov, *Design Sensitivity Analysis of Structural Systems*, Academic Press, Orlando, Florida, 1986.
6. E. J. Haug, "Design Sensitivity Analysis of Dynamic Systems," in *Computer Aided Design: Structural and Mechanical Systems*, (C. A. Mota-Soares, ed.), Springer-Verlag, Berlin, 1987.
7. W. Schiehlen, *Technische Dynamik*, Teubner, Stuttgart, 1986.
8. L. F. Shampine and M. K. Gordon, *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*, Freeman, San Francisco, 1975.
9. R. Bulirsch and H. Rutishauser, "Interpolation und genäherte Quadratur," in *Mathematische Hilfsmittel des Ingenieurs* (vol. 3), (R. Sauer and I. Szabo, eds.), Springer-Verlag, Berlin, 1968.
10. E. Kreuzer and G. Leister, *Programmsystem NEWEUL '88*, Anleitung AN-21, University of Stuttgart, Stuttgart, 1989.
11. B. W. Char et al., *MAPLE—Reference Manual*, Waterloo Maple, Waterloo, Canada, 1990.

Received October 1990