Research paper

Struct Multidisc Optim 26, 171–182 (2004)
DOI 10.1007/s00158-003-0336-1

Order-$(n+m)$ direct differentiation determination of design sensitivity for constrained multibody dynamic systems

K.S. Anderson and Y. Hsu

Abstract With the complexity and large dimensionality of many modern multibody dynamic applications, the efficiency of the sensitivity evaluation methods used can greatly impact the overall computation cost and as such can greatly limit the usefulness of the sensitivity information. Most current direct differentiation approaches suffer from prohibitive computational cost, which may be as great as $O(n^4+n^2m^2+nm^3)$ (for systems with n generalized coordinates and m algebraic constraints). This paper presents a concise and computationally efficient sensitivity analysis scheme to facilitate such sensitivity calculations. A unique feature of this scheme is its use of recursive procedures to directly embed the algebraic constraint relations in forming and simultaneously solving a minimal set of equations. This results in far fewer operations than more traditional, or so-called $O(n)$, counterparts. The algorithm determines the derivatives of generalized accelerations in $O(n+m)$ operations overall. The resulting equations are "exact" to integration accuracy and enforce constraints exactly at both the velocity and acceleration levels.

Key words multibody dynamics, design optimization, sensitivity analysis, recursive formulation

$\mathbf{a}^k$	Matrix representation of acceleration of center of mass k^* in the Newtonian reference frame N.
$\mathbf{a}_t^k$	Acceleration remainder term associated with body k in N; includes all terms of $\mathbf{a}^k$ that are not explicit in $\dot{\mathbf{u}}$'s.
$\boldsymbol{\mathcal{A}}^k$	The generalized acceleration matrix of body k in N.
$\bar{\boldsymbol{\mathcal{A}}}^k$	That portion of the generalized acceleration matrix of body k in N that is explicit in the unknown state derivatives $\dot{\mathbf{u}}$.
$\boldsymbol{\mathcal{A}}_t^k$	That portion of the generalized acceleration matrix of body k in N that is not explicit in the unknown state derivatives $\dot{\mathbf{u}}$.
${}^{o_i}\boldsymbol{\mathcal{A}}^k$	The generalized acceleration matrix of k_i in reference frame 0_i that is associated with closed-loop i.
${}^{o_i}\bar{\boldsymbol{\mathcal{A}}}^k$	That portion of the generalized acceleration matrix of k_i in 0_i that is explicit in the unknown state derivatives $\dot{\mathbf{u}}$.
${}^{o_i}\boldsymbol{\mathcal{A}}_t^k$	That portion of the generalized acceleration matrix of k_i in 0_i that is not explicit in the unknown state derivatives $\dot{\mathbf{u}}$.
$\mathbf{C}$	Invertible transformation matrix relating $\dot{\mathbf{q}}$ to $\mathbf{u}$.
$\boldsymbol{\mathcal{C}}^k$	Direction cosine matrix relating the basis vectors fixed in body k to those in proximal body $Pr[k]$.
$Dist[k]$	Distal body set associated with body k.
$\mathbf{D}$	Matrix used in relating $\dot{\mathbf{q}}$ to $\mathbf{u}$ and commonly associated with prescribed motions.
$\boldsymbol{\mathcal{F}}^k$	Recursive generalized force matrix for body k.
$\hat{\boldsymbol{\mathcal{F}}}^k$	Articulated body force associated with body k.
$\mathbf{I}^{k/k^*}$	Central inertia matrix of body k.
$\boldsymbol{\mathcal{I}}^k$	Generalized inertia matrix of body k.
$\hat{\boldsymbol{\mathcal{I}}}^k$	*Articulated body inertia* matrix of body k, associated with acceleration $\bar{\boldsymbol{\mathcal{A}}}^k$.
${}^{o_i}\hat{\boldsymbol{\mathcal{I}}}^k$	*Articulated body inertia* matrix of body k, associated with acceleration ${}^{o_i}\bar{\boldsymbol{\mathcal{A}}}^k$.
$\hat{\boldsymbol{\mathcal{I}}}^{k;o_i}$	*Articulated body inertia* matrix of body k, associated with acceleration $\bar{\boldsymbol{\mathcal{A}}}^{o_i}$.
k	Index representing an arbitrary system body k(global numbering).
k_i	Index representing an arbitrary body k within closed-loop i (local loop numbering).
k^*	Center of mass of body k.
$\mathbf{K}$	Right-hand side of system equations of motion, representing applied forces as well as centripetal and coriolis portions of inertia forces.
m	Total number of independent system constraints.
m_i	Total number of constraints associated with closed-loop i.
$\mathbf{M}$	System mass matrix.

Received: 20 August 2001
Revised manuscript received: 10 April 2002
Published online: 23 December 2003

K.S. Anderson✉,⋆ and Y. Hsu⋆⋆

Department of Mechanical, Aerospace, and Nuclear Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180-3590
e-mail: anderk5@rpi.edu, yhsu@ptc.com

⋆ Associate Professor
⋆⋆ Senior Software Engineer, PTC Co.

$\boldsymbol{\mathcal{M}}_k$ Matrix of terms associated with body k that would be found on the diagonal of a partially triangularized (decomposed) system mass matrix.
n Total number of system generalized coordinates.
n_i Total number of generalized coordinates associated with i-th closed-loop constraint equations.
n_L Total number of closed loops.
N Newtonian reference frame.
0_i Base body (primary reference frame) of closed-loop i
p_i Body containing the highest independent degree of freedom (local loop numbering) within closed-loop i
$\boldsymbol{\mathcal{P}}_r^k$ Partial velocity matrix for body k associated with generalized speeds $\mathbf{u}_r$.
$Pr[k]$ Proximal (parent) body set associated with body k.
$\mathbf{q}$ The $n \times 1$ matrix of generalized coordinates used to describe the configuration of the system.
$\dot{\mathbf{q}}$ The first time derivative of the system generalized coordinates.
$\mathbf{R}^k$ Matrix representation of resultant of all nonconstraint forces acting on body k.
$\boldsymbol{\mathcal{S}}^k$ Basis consistent *shift matrix* that converts the system of forces acting through mass center of k to an equivalent system acting through a point of k instantaneously coincident with the mass center of its proximal body $Pr[k]$.
${}^j\boldsymbol{\mathcal{S}}^k$ *Shift* matrix that transforms a force system acting through mass center of k to an equivalent system acting through a point of k instantaneously coincident with the mass center of body j.
t Time.
$\boldsymbol{\mathcal{T}}^k$ Local *triangularization matrix* associated with body k.
$\mathbf{u}$ The system *generalized speeds* that characterize the motion of the system.
$\dot{\mathbf{u}}$ System generalized accelerations to be determined and temporally integrated.
$\mathbf{U}$ Appropriately dimensioned identity matrix.
$\mathbf{v}^{k^*}$ Velocity of the center of mass of body k in reference frame N.
$\mathbf{v}_r^{k^*}$ Partial velocity of the center of mass of body k in reference frame N associated with u_r.
$\mathbf{v}_t^{k^*}$ *Velocity remainder term* associated with the center of mass of body k in reference frame N.
$\boldsymbol{\mathcal{V}}^k$ The generalized velocity matrix, which relates the velocity of body k to reference frame N.
$\bar{\boldsymbol{\mathcal{V}}}^k$ Portion of the body k generalized velocity matrix that is explicit in the generalized speeds $\mathbf{u}$.
$\boldsymbol{\mathcal{V}}_t^k$ Body k generalized velocity remainder term matrix, which relates the velocity of body k to reference frame $\widetilde{0}$.
${}^{o_i}\boldsymbol{\mathcal{V}}^k$ Generalized velocity matrix, which relates the velocity of k_i mass center to that of loop base body 0_i to reference frame 0_i.
${}^{o_i}\bar{\boldsymbol{\mathcal{V}}}^k$ Portion of the generalized velocity matrix ${}^{o_i}\boldsymbol{\mathcal{V}}^k$, which is explicit in the generalized speeds $\mathbf{u}$.
${}^{o_i}\boldsymbol{\mathcal{V}}_t^k$ That portion of the generalized velocity matrix ${}^{o_i}\boldsymbol{\mathcal{V}}^k$ that is not explicit in the generalized speeds $\mathbf{u}$.
α^k Angular acceleration of body k in Newtonian reference frame N.
α_t^k *Angular acceleration remainder terms* of body k; This represents all terms of α^k that are not explicit in $\dot{\mathbf{u}}$'s.
$\boldsymbol{\delta}^j$ Useful intermediate quantity associated with recursive treatment of dependent generalized speeds and associated state derivatives of the closed loop under consideration.
$\boldsymbol{\Delta}^{k_i}$ Useful intermediate quantity associated with recursive treatment of body k_i of the closed loop under consideration.
$\boldsymbol{\gamma}^k$ Position vector from body $Pr[k]$ mass center to body k mass center.
$\boldsymbol{\Gamma}^k$ Useful intermediate quantity associated with recursive treatment of dependent degrees of freedom of body k within the closed loop under consideration.
$\boldsymbol{\lambda}$ Lagrange multipliers.
$\boldsymbol{\tau}^{k_i}$ Useful intermediate quantity associated with recursive treatment of body k_i of the closed loop under consideration.
$\boldsymbol{\Xi}^k$ Useful intermediate quantity associated with recursive treatment of dependent degrees of freedom of body k within the closed loop under consideration.
$\boldsymbol{\Phi}$ Set of m system algebraic constraint equations.
$\boldsymbol{\Phi}_{,\mathbf{q}}$ System constraint Jacobian.
$\boldsymbol{\chi}^{p+j}$ Useful intermediate quantity associated with recursive treatment of dependent generalized speeds and associated state derivatives $p+j$ of the closed loop under consideration.
$\boldsymbol{\omega}^k$ Angular velocity of body k in reference frame N.
$\boldsymbol{\omega}_r^k$ Partial angular velocity of body k in reference frame N associated with u_r.
$\boldsymbol{\omega}_t^k$ *Angular velocity remainder term* associated with body k in reference frame N.
$\boldsymbol{\omega}_\times^k$ Matrix equivalent to vector cross product $\boldsymbol{\omega}^k\times$.

1 Introduction

The theory of multibody dynamic systems has been used in an ever broadening variety of applications and has now become an essential element of the design and analysis processes for many mechanical systems. Evidence of this is seen in a very broad range of practical applications including robotics, biomechanics, molecular systems, electromechanics, spacecraft, and microscale dynamic devices. Within this context, multibody systems are described by a finite number of interconnected rigid and/or flexible bodies with chain, tree, or closed-loop geometric configurations moving under the influence of applied and constraint forces. The advancements realized in

multibody dynamics modeling and analysis methods over the past few decades have opened a new era of simulation- and model-based engineering in the 21st century.

One of the profound needs exhibited in modern multibody dynamics applications is that of sensitivity analysis. Sensitivity analysis requires evaluations of sensitivities of various multibody dynamics quantities with respect to parameters and design variables $\mathbf{p}$ of interest. These sensitivities can, in various forms, play important roles associated with multibody computational problems such as implicit integration schemes, design optimization, linearized dynamics, optimal control, etc. Due to the complexity of equations of motion, the determination of these sensitivity terms have traditionally been regarded as a formidable (prohibitive) task. Significant computational effort is required to reveal the implicit relationships between system generalized accelerations and design parameters (Haug and Arora 1979) and to solve for sensitivities of generalized accelerations through explicit formulation of sensitivity governing equations (Hsu 2000; Tak 1990).

Resorting to a numerical finite difference approximation may represent a *simple* alternative. However, this (perhaps the most broadly used technique) usually possesses some very undesired features. Specifically, numerical solutions are sensitive to the perturbation size (Bestle and Eberhard 1992; Hsu and Anderson 2001); the *optimal* perturbation size of a set of design variables differs from that associated with generalized coordinates (Hsu 2000), and additional costs are necessary to determine the perturbed effect of each variable (Bischof 1996).

Analytical methods do not suffer from these aforementioned undesirable characteristics. The adjoint variable method (Bestle and Seybold 1992; Eberhard 1996; Haug et al. 1984) and the direct differentiation method (Chang and Nikravesh 1985; Jain and Rodriguez 1999; Serban and Haug 1998; Tak 1990) are two such very competitive approaches that may be applied in performing sensitivity analysis. Additionally, the automatic differentiation technique has also shown its potential in this regard (Barthelemy and Hall 1995; Bischof 1996).

The adjoint variable method is a sophisticated approach that introduces a set of adjoint variables so as to avoid explicit calculations of state sensitivities. Solving a sequence of adjoint relationships yields the sensitivity vector corresponding to the sensitivity of the performance criteria in terms of parameter variations. Unfortunately, the implementation of the adjoint variable method is often complicated and requires a significant amount of I/O operations (Chang and Nikravesh 1985; Pagalday and Aranburu 1996) due to the extremely large amount of data, which must be stored, associated with all state information for the entire forward problem simulation. This I/O can severely hobble the performance (speed) of the method for large-scale problems. Additionally, a possible source of error in this approach may arise from the backward temporal integration required in solving for the adjoint variables. This error can occur because the time steps in the forward and backward integration will not generally coincide when using variable step size temporal integration schemes, so an interpolation model must then be employed.

The direct differentiation method has received increasing attention recently, with many researchers reconsidering its benefits and potential. This approach possesses many desired features in dealing with complex sensitivity analysis: (1) it is arguably the most conceptually straightforward method, being a direct application of the differentiation *chain rule*; (2) it often offers higher numerical stability than many competing approaches; (3) the solution accuracy is insensitive to parameter perturbation; (4) the general approach is easily adapted to problem structures; (5) the key quantities desired from the analysis are directly available. Current implementations of direct differentiation methods appearing in the literature (Chang and Nikravesh 1985; Dias and Pereira 1997; Serban and Haug 1998) often involve Newton–Euler with Cartesian coordinate formulations in forming the sensitivity governing equations. Although this usage exhibits ease of formulation for a highly constrained system, this combination often suffers from the production of a *differential algebraic* set of sensitivity equations that are of high dimension, which can be prohibitively costly to produce and solve.

Formulating and subsequently solving these equations may become the main source of computational burden if the problem structures are not carefully exploited or a brute force approach is adopted to compute state derivatives. Specifically, a substantial amount of work is needed to form the required state derivatives, which often results in the phenomenon termed *derivative swell*, leading to an overall computational load as high as $O(n^4 + n^2m^2 + nm^3)$ (Hsu 2000) per member of $\mathbf{p}$, where n is the number of generalized coordinates used in an unconstrained problem and m is the number of independent algebraic constraints to be enforced. Addressing the computational efficiency, and thereby reducing the overall computational cost associated with the sensitivity analysis, represents perhaps the most important issue associated with the use of the direct differentiation approach.

This paper presents a novel first-order sensitivity analysis algorithm that circumvents many of the drawbacks exhibited in more conventional direct differentiation implementations. Its foundation can in part be traced to various recursive dynamic formulations (Anderson 1990; Rosenthal 1990; Pradhan et al. 1997) that have demonstrated numerical efficiency for large-scale dynamic forward problem analysis and simulation. The developed method provides a way to make sensitivity computations analytically and numerically tractable through its state space full recursive procedure, which directly embeds algebraic constraint relations, simultaneously solving the minimal set of first-order sensitivity equations as they are being formed. The method presented herein will greatly reduce computational inefficiency relative to alternative "exact" first-order sensitivity approaches, particularly

when applied to modestly and heavily constrained multi-rigid-body dynamic systems involving a large number of design parameters and/or generalized coordinates. The algorithm is able to achieve a significantly improved bilinear (state and design variables) $O(n+m)$ computational performance for each design variable p_j in $\mathbf{p}$. This represents a considerable computational savings relative to that offered by more conventional implementations of direct differentiation for constrained systems that can yield $O(n^4+n^3m+n^2m^2+nm^3)$ overall performance. Additionally, the presented algorithm delivers considerable performance gains relative to the authors' prior $O(n)$-based sensitivity algorithms (Hsu and Anderson 2002) that provided $O(n+nm+nm^2+m^3)$ overall performance when applied to highly constrained systems. The reduced overall computational cost of the presented algorithm should offer considerable performance advantages over competing methods, particularly when applied to large-scale (n and/or $m \gg 1$) multibody electromechanical dynamic systems.

2 Sensitivity problem formulations

Calculation of state sensitivities with respect to the desired quantities (e.g., design or control variables) under consideration represent perhaps the most arduous task involved in sensitivity analysis. This is often due to the implicit relationships that exist between state variables and these parameters and the associated need for the extraction of state information from the corresponding forward problem analysis. For a general multibody system, the system's dynamic properties can be fully characterized by a set of n generalized coordinates $\mathbf{q}$, the generalized speeds $\mathbf{u}$, and the generalized accelerations $\dot{\mathbf{u}}$. In this formulation, the generalized speeds (Kane and Levinson 1985) $\mathbf{u}$ are a user-specified invertible combination of the $\dot{\mathbf{q}}$'s such that

$$\dot{\mathbf{q}} = \mathbf{C}(\mathbf{q}, t; \mathbf{p})\,\mathbf{u} + \mathbf{D}(\mathbf{q}, t; \mathbf{p}). \tag{1}$$

Computation of values associated with the derivatives of each of these generalized quantities with respect to the parameters $\mathbf{p}$ are thus required. However, these derivatives need not be performed independently since the relationships Eq. 1 and

$$\frac{\mathrm{d}\mathbf{q}_r}{\mathrm{d}\mathbf{p}_j} = \int_{t=0}^{t=\tau} \frac{\mathrm{d}\dot{\mathbf{q}}_r}{\mathrm{d}\mathbf{p}_j}\,\mathrm{d}t + \left.\frac{\mathrm{d}\mathbf{q}_r}{\mathrm{d}\mathbf{p}_j}\right|_{t=0}\Bigg]_{\mathbf{p}}, \tag{2}$$

$$\frac{\mathrm{d}\mathbf{u}_r}{\mathrm{d}\mathbf{p}_j} = \int_{t=0}^{t=\tau} \frac{\mathrm{d}\dot{\mathbf{u}}_r}{\mathrm{d}\mathbf{p}_j}\,\mathrm{d}t + \left.\frac{\mathrm{d}\mathbf{u}_r}{\mathrm{d}\mathbf{p}_j}\right|_{t=0}\Bigg]_{\mathbf{p}}, \tag{3}$$

exist.

Equations 2 and 3 enable the total derivatives of $\mathrm{d}\mathbf{q}_i/\mathrm{d}\mathbf{p}_j$ and $\mathrm{d}\mathbf{u}_i/\mathrm{d}\mathbf{p}_j$ to be obtained from the temporal integration of $\mathrm{d}\dot{\mathbf{u}}_i/\mathrm{d}\mathbf{p}_j$ evaluated at the current design point $\mathbf{p}$. This property implies that the determination of the derivatives used in the sensitivity analysis can now be reduced to the problem of determining key values of $\mathrm{d}\dot{\mathbf{u}}_i/\mathrm{d}\mathbf{p}_j$ for the entire time interval of interest.

Production of this set of key quantities can be achieved by considering a general form of the equations of motion

$$\boldsymbol{\mathcal{M}}(\mathbf{q}, t; \mathbf{p}) \cdot \dot{\mathbf{u}} + \boldsymbol{\Phi}_{,\mathbf{q}}^{T}(\mathbf{q}, t; \mathbf{p}) \cdot \boldsymbol{\lambda} = \boldsymbol{\mathcal{K}}(\mathbf{q}, \mathbf{u}, t; \mathbf{p}), \tag{4}$$

$$\boldsymbol{\Phi}(\mathbf{q}, t; \mathbf{p}) = \mathbf{0}, \tag{5}$$

where matrix $\boldsymbol{\mathcal{M}}(\mathbf{q}, t; \mathbf{p})$ is often referred to as the system mass matrix and matrix $\boldsymbol{\mathcal{K}}(\mathbf{q}, \mathbf{u}, t; \mathbf{p})$ contains all applied forces as well as the inertia force terms associated with coriolis and centripetal accelerations. The quantity $\boldsymbol{\Phi}_{,\mathbf{q}}$ is the constraint Jacobian associated with the partial derivative of the m independent algebraic constraints Eq. 5 with respect to $\mathbf{q}$. In Eqs.4 and 5, some of the state variables may be redundant. If a redundancy exists, then generalized constraint "forces" $\boldsymbol{\lambda}$ must be applied to the system equations of motion (Eq. 4) to enforce the algebraic constraint equations (Eq. 5).

Taking the first total derivative of Eqs. 4 and 5 with respect to a nominal parameter $\mathbf{p}_j$ and rearranging the quantities yields

$$\begin{aligned}
&\boldsymbol{\mathcal{M}}_{,\mathbf{p}_j}\frac{\mathrm{d}\dot{\mathbf{u}}}{\mathrm{d}\mathbf{p}_j} = \boldsymbol{\mathcal{K}}_{,\mathbf{p}_j} + \boldsymbol{\mathcal{K}}_{,\mathbf{q}_r}\frac{\mathrm{d}\mathbf{q}_r}{\mathrm{d}\mathbf{p}_j} + \boldsymbol{\mathcal{K}}_{,\mathbf{u}_r}\frac{\mathrm{d}\mathbf{u}_r}{\mathrm{d}\mathbf{p}_j} - \\
&\left(\boldsymbol{\mathcal{M}}_{,\mathbf{p}_j} + \boldsymbol{\mathcal{M}}_{,\mathbf{q}_r}\frac{\mathrm{d}\mathbf{q}_r}{\mathrm{d}\mathbf{p}_j}\right)\dot{\mathbf{u}} - \\
&\left(\boldsymbol{\Phi}_{,\mathbf{q}}^{T}\cdot\boldsymbol{\lambda}\right)_{,\mathbf{p}_j} - \left(\boldsymbol{\Phi}_{,\mathbf{q}}^{T}\cdot\boldsymbol{\lambda}\right)_{,\mathbf{q}_r}\frac{\mathrm{d}\mathbf{q}_r}{\mathrm{d}\mathbf{p}_j}
\end{aligned} \tag{6}$$

and

$$\boldsymbol{\Phi}_{,\mathbf{p}_j} + \boldsymbol{\Phi}_{,\mathbf{q}_r}\frac{\mathrm{d}\mathbf{q}_r}{\mathrm{d}\mathbf{p}_j} = \mathbf{0}, \tag{7}$$

where the use of a comma "," indicates a partial derivative with respect to the indicated parameter, $r = 1, \ldots, n$, and summation is implied over the repeated index within a term. Equations 6 and 7 are the governing equations for yielding $\mathrm{d}\dot{\mathbf{u}}/\mathrm{d}\mathbf{p}_j$ in the first-order sensitivity analysis of a constrained system. To solve for $\mathrm{d}\dot{\mathbf{u}}/\mathrm{d}\mathbf{p}_j$, the sensitivities of the Lagrange multiplier $\boldsymbol{\lambda}$ associated with the constraint forces must be computed. This is performed by first expressing $\mathrm{d}\dot{\mathbf{u}}/\mathrm{d}\mathbf{p}_j$ using Eq. 6 and then substituting this expression into Eq. 7 at the acceleration level, yielding

$$\begin{aligned}
&\frac{\mathrm{d}\boldsymbol{\lambda}}{\mathrm{d}\mathbf{p}_j} = \left[\boldsymbol{\Phi}_{,\mathbf{q}_r}\boldsymbol{\mathcal{M}}^{-1}\boldsymbol{\Phi}_{,\mathbf{q}}^{T}\right]^{-1}\left[\frac{\mathrm{d}\boldsymbol{\Phi}_{,\mathbf{q}_r}}{\mathrm{d}\mathbf{p}_j}\dot{\mathbf{u}} + \frac{\mathrm{d}\mathbf{B}}{\mathrm{d}\mathbf{p}_j} + \right. \\
&\left.\boldsymbol{\Phi}_{,\mathbf{q}_r}\boldsymbol{\mathcal{M}}^{-1}\left(\frac{\mathrm{d}\boldsymbol{\mathcal{K}}}{\mathrm{d}\mathbf{p}_j} - \frac{\mathrm{d}\boldsymbol{\mathcal{M}}}{\mathrm{d}\mathbf{p}_j}\dot{\mathbf{u}} - \frac{\mathrm{d}\boldsymbol{\Phi}_{,\mathbf{q}}^{T}}{\mathrm{d}\mathbf{p}_j}\boldsymbol{\lambda}\right)\right],
\end{aligned} \tag{8}$$

where $\mathbf{B}$ contains all terms that are not explicit in $\dot{\mathbf{u}}$ when Eq. 5 is differentiated to generate constraint equations at

the acceleration level. Substituting Eq. 8 into Eq. 6 produces the desired quantities $\mathrm{d}\dot{\mathbf{u}}/\mathrm{d}\mathbf{p}_j$, namely,

$$\frac{\mathrm{d}\dot{\mathbf{u}}}{\mathrm{d}\mathbf{p}_j} = \quad \boldsymbol{\mathcal{M}}^{-1}\left[\frac{\mathrm{d}\boldsymbol{\mathcal{K}}}{\mathrm{d}\mathbf{p}_j} - \frac{\mathrm{d}\boldsymbol{\mathcal{M}}}{\mathrm{d}\mathbf{p}_j}\dot{\mathbf{u}} - \frac{\mathrm{d}\boldsymbol{\Phi}_{,\mathbf{q}}^{T}}{\mathrm{d}\mathbf{p}_j}\boldsymbol{\lambda} - \boldsymbol{\Phi}_{,\mathbf{q}}^{T}\frac{\mathrm{d}\boldsymbol{\lambda}}{\mathrm{d}\mathbf{p}_j}\right]. \tag{9}$$

For any nominal quantity β involved in the above calculations, its total derivative is defined as

$$\frac{\mathrm{d}\beta}{\mathrm{d}\mathbf{p}_j} \equiv \beta_{,\mathbf{p}_j} + \beta_{,\mathbf{q}_r}\frac{\mathrm{d}\mathbf{q}_r}{\mathrm{d}\mathbf{p}_j} + \beta_{,\mathbf{u}_r}\frac{\mathrm{d}\mathbf{u}_r}{\mathrm{d}\mathbf{p}_j}\ , \tag{10}$$

where $r = 1, \ldots, n$. The highest cost of forming Eqs. 8 and 9 is associated with $\mathrm{d}\boldsymbol{\mathcal{M}}/\mathrm{d}\mathbf{p}_j$, where $O(n^4)$ operations overall may be required for an $O(n^3)$-based formulation (Hsu and Anderson 2001). This cost is further increased by $O(n^2)$ matrix multiplications, an $O(n^3)$ expense for the decomposition of the system mass matrix, and additional $O(n^2m^2 + nm^3)$ costs associated with the enforcement of constraints. This results in a prohibitively high computational effort of $O(n^4 + n^3 + n^2m^2 + nm^3)$ when using a direct differentiation method based on a traditional $O(n^3)$ forward dynamics formulation. To circumvent these deficiencies, a more efficient strategy must be developed. A new fully recursive dynamics formulation that does not suffer from computational cost penalties for either large numbers of system generalized coordinates n or constraints m offers a means to resolve these issues.

3
Recursive sensitivity analysis formulation

As was previously shown in Hsu and Anderson (2002), a significant reduction in the overall computational effort associated with direct differentiation methods applied to multibody dynamic systems can be accomplished through the use of efficient recursive formalisms. For such formalisms, generalized coordinates that describe the relative orientation and position of adjacent bodies have thus far shown themselves to be well suited for the generation of state space recursive relationships (Anderson 1990; Featherstone 1983; Hollerbach 1980; Rosenthal 1990). Similarly, recursive relationships that arise as part of the sensitivity analysis can also be performed in a like, very efficient, manner. Within this context, the key quantities of state sensitivities are determined through a set of procedures that are very much of the same form as those used in the forward problem simulation. These procedures are: (i) A kinematic sweep in which all kinematic quantities are determined involving both independent and dependent generalized speeds; (ii) recursive elimination of redundant quantities at both the velocity and acceleration level; (iii) a triangularization sweep in which the equations of motion and governing sensitivity equations are effectively decomposed as they are formed; and (iv) back substitution. In the following sections, each of these principal topics will be discussed.

3.1
Mathematical preliminaries

To aid in the following mathematical development, consider the notation associated with the description of an arbitrary set of interconnected rigid bodies shown in Fig. 1. For this system, proximal (parent) body $Pr[k]$ is connected to its child body k through joint-k, via joint points k^- and k^+, which reside in bodies $Pr[k]$ and k, respectively. Similarly, the *distal* (child) bodies of body k are given as members of the set of bodies $Dist[k]$. The position vector $\mathbf{s}^k$ locates joint-k relative to the mass center of body $Pr[k]$, while the position vector $\mathbf{r}^k$ locates the mass center of body k with respect to the outboard end of this same joint. It will also prove convenient to describe the position of child mass center k relative to proximal mass center $Pr[k]$ by the vector $\boldsymbol{\gamma}^k$. Finally, the generalized speeds u_k to be used in the recursive relations are

$$u_k = \dot{q}_k \qquad (k = 1, 2, \ldots, n)\ . \tag{11}$$

Thus the angular velocity of any body k with respect to the Newtonian reference frame N and the velocity of its associated mass center k^* may in turn be written in terms of these generalized speeds as

$$\boldsymbol{\omega}^k = \sum_{r=1}^{n}\left(\boldsymbol{\omega}_r^k\right)^T\mathbf{u}_r + \boldsymbol{\omega}_t^k \tag{12}$$

and

$$\mathbf{v}^{k^*} = \sum_{r=1}^{n}\mathbf{v}_r^{k^*}\,\mathbf{u}_r + \mathbf{v}_t^{k^*}. \tag{13}$$

In these expressions, $\boldsymbol{\omega}_r^k$ and $\mathbf{v}_r^{k^*}$ are termed the *r-th partial angular velocity matrix of body k* and the *r-th partial*

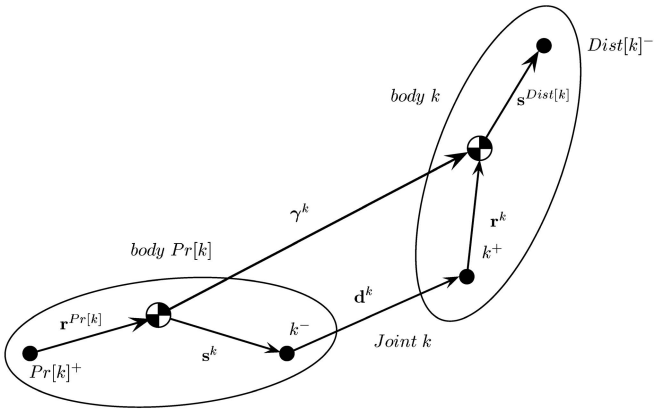

Fig. 1 A multibody system and associated set of subsystems (subchains)

velocity matrix of pointk^* in N, respectively. These quantities may be thought of as basis vectors for the space of admissible system velocities and angular velocities, while the associated generalized speeds are the velocity space measure numbers. Additionally, the terms $\boldsymbol{\omega}_t^k$ and $\mathbf{v}_t^{k^*}$ appearing in Eqs. 12 and 13 are referred to as the *angular velocity remainder term* matrix of body k and *velocity remainder term* matrix of point k^* in N, respectively.

With these quantities so defined, it is convenient to introduce generalized (considering both rotation and translation) *velocity*, *partial velocity*, and *velocity remainder term* matrices as

$$\boldsymbol{\mathcal{V}}^k = \begin{bmatrix} \boldsymbol{\omega}^k \\ \mathbf{v}^{k^*} \end{bmatrix}, \quad \boldsymbol{\mathcal{P}}_r^k = \begin{bmatrix} \boldsymbol{\omega}_r^k \\ \mathbf{v}_r^{k^*} \end{bmatrix}, \quad \text{and} \quad \boldsymbol{\mathcal{V}}_t^k = \begin{bmatrix} \boldsymbol{\omega}_t^k \\ \mathbf{v}_t^{k^*} \end{bmatrix}. \tag{14}$$

With these matrices so defined, Eqs. 12 and 13) may be represented by

$$\boldsymbol{\mathcal{V}}^k = \bar{\boldsymbol{\mathcal{V}}}^k + \boldsymbol{\mathcal{V}}_t^k = \sum_{r=1}^{n} \boldsymbol{\mathcal{P}}_r^k u_r + \boldsymbol{\mathcal{V}}_t^k. \tag{15}$$

One can similarly express the generalized acceleration matrix of an arbitrary body k as defined in previous works (Anderson 1990, 1992) as

$$\boldsymbol{\mathcal{A}}^k = \begin{bmatrix} {}^N\boldsymbol{\alpha}^k \\ {}^N\mathbf{a}^{k^*} \end{bmatrix}, \tag{16}$$

where $\boldsymbol{\mathcal{A}}^k$ may also be divided into two portions. One is $\bar{\boldsymbol{\mathcal{A}}}^k$, which contains all terms that are explicit in the unknown state derivatives of $\dot{\mathbf{u}}$, and the other is *acceleration remainder term* $\boldsymbol{\mathcal{A}}_t^k$, which represent all of the other acceleration terms, giving

$$\boldsymbol{\mathcal{A}}^k = \bar{\boldsymbol{\mathcal{A}}}^k + \boldsymbol{\mathcal{A}}_t^k. \tag{17}$$

3.2 Kinematic relationships

With the generalized velocity, generalized acceleration, and generalized acceleration remainder term matrices represented in this manner, it is possible to compactly represent the recursive relationships necessary for determining all system kinematic quantities (Anderson 1990, 1992; Hsu and Anderson 2001). Specifically,

$$\boldsymbol{\mathcal{V}}^k = \left[(\boldsymbol{\mathcal{S}}^k)^T \bar{\boldsymbol{\mathcal{V}}}^{Pr[k]} + \boldsymbol{\mathcal{P}}_k^k \mathbf{u}_k \right] + \boldsymbol{\mathcal{V}}_t^k, \tag{18}$$

and

$$\boldsymbol{\mathcal{A}}^k = \left[(\boldsymbol{\mathcal{S}}^k)^T \bar{\boldsymbol{\mathcal{A}}}^{Pr[k]} + \boldsymbol{\mathcal{P}}_k^k \dot{\mathbf{u}}_k \right] + \boldsymbol{\mathcal{A}}_t^k, \tag{19}$$

where

$$\boldsymbol{\mathcal{V}}_t^k = (\boldsymbol{\mathcal{S}}^k)^T \boldsymbol{\mathcal{V}}_t^{Pr[k]} + \begin{bmatrix} \text{simple expressions dependent} \\ \text{on joint type of } k \end{bmatrix}, \tag{20}$$

and

$$\boldsymbol{\mathcal{A}}_t^k = (\boldsymbol{\mathcal{S}}^k)^T \boldsymbol{\mathcal{A}}_t^{Pr[k]} + \begin{bmatrix} \text{simple expressions involving} \\ \text{elements of } \boldsymbol{\mathcal{V}}^k, \mathbf{r}^k, \text{ and } \mathbf{s}^k \end{bmatrix}. \tag{21}$$

The quantity $\boldsymbol{\mathcal{S}}^k$ appearing in Eqs. 18–21 is the basis consistent linear transformation matrix

$$\boldsymbol{\mathcal{S}}^k = \begin{bmatrix} \boldsymbol{\mathcal{C}}^k & \boldsymbol{\mathcal{C}}^k \boldsymbol{\gamma}_\times^k \\ \mathbf{0} & \boldsymbol{\mathcal{C}}^k \end{bmatrix}_{6\times 6}. \tag{22}$$

Within this expression $\boldsymbol{\mathcal{C}}^k \equiv {}^{Pr[k]}\boldsymbol{\mathcal{C}}^k$ is the direction cosine matrix that relates the body k basis vectors to those fixed in its parent body $Pr[k]$, $\mathbf{0}$ are simply 3×3 zero matrices, and $\boldsymbol{\gamma}_\times^k$ is the skew symmetric matrix equivalent to the vector cross product operation $\boldsymbol{\gamma}^k\times$. The transformation $\boldsymbol{\mathcal{S}}^k$ converts a system of forces and moments acting through the center of mass of k to an equivalent force system, acting through a point of k that is instantaneously coincident with the center of mass of $Pr[k]$.

At this time, it is convenient to define the body k *generalized inertia* $\boldsymbol{\mathcal{I}}^k$ and the body k *generalized force* $\boldsymbol{\mathcal{F}}^k$ matrices

$$\boldsymbol{\mathcal{I}}^k = \begin{bmatrix} \mathbf{I}^{k/k^*} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^k \end{bmatrix}_{6\times 6}, \tag{23}$$

$$\boldsymbol{\mathcal{F}}^k = \mathbf{F}^k - \boldsymbol{\mathcal{I}}^k \boldsymbol{\mathcal{A}}_t^k + \begin{bmatrix} \boldsymbol{\omega}_\times^k \mathbf{I}^{k/k^*} \boldsymbol{\omega}^k \\ \mathbf{0} \end{bmatrix}_{6\times 1}. \tag{24}$$

Within these expressions, $\mathbf{I}^{k/k^*}$ is the 3×3 central inertia matrix of body k and $\mathbf{M}^k$ is the diagonal translational mass matrix of this same body. By comparison, the term $\mathbf{F}^k$ represents the resultant force system of all moments and forces acting on body k.

Kinematic term derivatives

To facilitate the sensitivity analysis, the total derivatives associated with Eqs.20–24 are required. An examination of these equations shows that they may be easily determined via the chain rule as

$$\frac{\mathrm{d}\boldsymbol{\mathcal{V}}_t^k}{\mathrm{d}\mathbf{p}_j} = \left(\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j} \boldsymbol{\mathcal{S}}^k \right)^T \boldsymbol{\mathcal{V}}_t^{Pr[k]} + (\boldsymbol{\mathcal{S}}^k)^T \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j} \boldsymbol{\mathcal{V}}_t^{Pr[k]} + \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j} \begin{bmatrix} \text{geometric expressions} \\ \text{local to } k \end{bmatrix} \tag{25}$$

$$\frac{\mathrm{d}\boldsymbol{\mathcal{A}}_t^k}{\mathrm{d}\mathbf{p}_j} = \left(\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j} \boldsymbol{\mathcal{S}}^k \right)^T \boldsymbol{\mathcal{A}}_t^{Pr[k]} + (\boldsymbol{\mathcal{S}}^k)^T \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j} \boldsymbol{\mathcal{A}}_t^{Pr[k]} + \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j} \begin{bmatrix} \text{expressions involving} \\ \text{terms from } \boldsymbol{\mathcal{V}}_t^k \end{bmatrix} \tag{26}$$

$$\frac{\mathrm{d}\boldsymbol{\mathcal{F}}^k}{\mathrm{d}\mathbf{p}_j} = \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j}\left(\mathbf{F}^k - \boldsymbol{\mathcal{I}}^k\boldsymbol{\mathcal{A}}_t^k + \begin{bmatrix}\boldsymbol{\omega}_\times^k\,\mathbf{I}^{k/k^*}\,\boldsymbol{\omega}^k \\ \mathbf{0}\end{bmatrix}\right)_{6\times 1}, \quad (27)$$

with

$$\frac{\mathrm{d}\boldsymbol{\mathcal{S}}^k}{\mathrm{d}\mathbf{p}_j} = \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j}\begin{bmatrix}\boldsymbol{\mathcal{C}}^k & \boldsymbol{\mathcal{C}}^k\boldsymbol{\gamma}_\times^k \\ \mathbf{0} & \boldsymbol{\mathcal{C}}^k\end{bmatrix}. \quad (28)$$

and

$$\frac{\mathrm{d}\boldsymbol{\mathcal{I}}^k}{\mathrm{d}\mathbf{p}_j} = \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j}\left[\begin{array}{c|c}\mathbf{I}^{k/k^*} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{M}^k\end{array}\right]_{6\times 6}. \quad (29)$$

The derivatives of the inertia mass matrix $\boldsymbol{\mathcal{I}}^k$ and the shifting matrix $\boldsymbol{\mathcal{S}}^k$ are strictly local quantities, being affected only by body and/or joint k. These quantities can be expressed as simple combinations of what this paper terms *derivative primitives*. These derivative primitives are the most basic form of derivative expression and are as a rule simple in form, local in nature, and form the basis on which all other derivative expressions can be built.

From Eqs. 25–29 it is apparent that all indicated quantities are either local to a specific body k and thus can be easily determined at fixed cost or recursively determined from previously ascertained quantities.

3.3 Enforcement of constraints

The primary computational efficiency gains associated with the new algorithm presented here relative to other so-called $O(n)$ approaches (Hsu and Anderson 2002) with respect to systems with closed loops are realized through the manner in which the loops are treated and the associated algebraic constraints enforced. In the authors' prior work, the constraints were enforced through the application of unknown constraint forces and an associated companion set of algebraic constraint equations. This prior approach was shown (Hsu and Anderson 2002) to determine first-order design sensitivities for general constrained multi-rigid-body system in $O(n+nm+nm^2+m^3)$ operations overall per temporal integration step. This represented a considerable improvement over the $O(n^4+n^2m^2+nm^3)$ performance generally realized when using direct differentiation with more traditional multibody formulations. Unfortunately, this $O(n+nm+nm^2+m^3)$ expense could still prove prohibitively high in situations involving large-scale heavily constrained systems, i.e., $(m \sim n, \text{and } m \gg 1)$. The goal is to further significantly reduce this computational expense by directly embedding the constraints in such a manner that the calculations associated with dependent degrees of freedom can be removed and many of the numerical problems often associated with such system of *differential algebraic equations* avoided. This is accomplished by using recursive loop closure relationships to eliminate dependent velocity and acceleration coordinates from the equations.

Consider an arbitrary i-th closed loop of a complex multibody system, as indicated in Fig. 2. The process begins by selecting a body of the closed loop that will act as the subsystem's primary reference frame. This body is locally referred to as the *loop base body* and is given the local body number 0_i. This is most easily accomplished by making the body that branches to form the closed loop the loop's base body. The bodies of this loop are then locally numbered consecutively, up through body local number $n_i - 1$, where n_i is the number of bodies that form this i-th closed loop, with the dependent coordinates defined as those associated with the highest joint (body) numbers. Body p_i is then the highest numbered body within the loop that can be fully described using only independent coordinates.

It can be shown (Stejskal and Valášek 1996; Anderson 2001a,b) that, proceeding as summarized above, it is possible to describe all velocity and acceleration level unknowns associated with dependent variables in terms of dependent variables. Specifically,

$${}^{o_i}\bar{\boldsymbol{\mathcal{V}}}^k = \boldsymbol{\Gamma}^{k\ o_i}\bar{\boldsymbol{\mathcal{V}}}^{p_i} + \boldsymbol{\Xi}^{k\ o_i}\boldsymbol{\mathcal{V}}_t^{n_i}, \quad (30)$$

and

$${}^{o_i}\bar{\boldsymbol{\mathcal{A}}}^k = \boldsymbol{\Gamma}^{k\ o_i}\bar{\boldsymbol{\mathcal{A}}}^{p_i} + \boldsymbol{\Xi}^{k\ o_i}\boldsymbol{\mathcal{A}}_t^{n_i}, \quad (31)$$

where all velocity and acceleration terms indicated are given relative to the primary loop reference frame fixed in body 0_i. The terms $\boldsymbol{\Gamma}^k$ and $\boldsymbol{\Xi}^k$ in Eqs. 30 and 31 and their derivatives with respect to $\mathbf{p}_j$ are determined through a chain of recursively determined intermediate quantities

$${}^{n_i}\boldsymbol{\mathcal{S}}^{n_i} = \mathbf{U} \quad \Rightarrow \quad {}^{k-1}\boldsymbol{\mathcal{S}}^{n_i} = \boldsymbol{\mathcal{S}}^{k\ k}\boldsymbol{\mathcal{S}}^{n_i}, \quad (32)$$

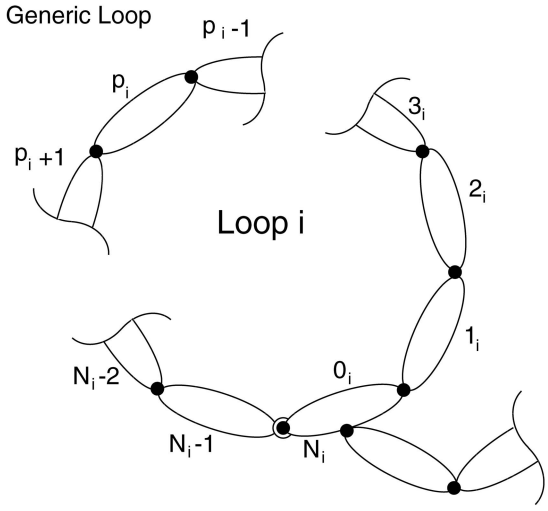

Fig. 2 Local numbering scheme of generic loop

$$\boldsymbol{\tau}^{n_i+1} = \mathbf{U}$$

$$\boldsymbol{\tau}^k = \left[\boldsymbol{\tau}^{k+1} - \left({}^k\boldsymbol{\mathcal{S}}^{n_i}\right)^T \boldsymbol{\mathcal{P}}_k^k \left(\Delta^k\right)^{-1} \left(\boldsymbol{\mathcal{P}}_k^k\right)^T {}^k\boldsymbol{\mathcal{S}}^{n_i}\boldsymbol{\tau}^k\right] \quad (33)$$

$$\Delta^k = \left(\boldsymbol{\mathcal{P}}_k^k\right)^T {}^k\boldsymbol{\mathcal{S}}^{n_i}\boldsymbol{\tau}^{k+1}\left({}^k\boldsymbol{\mathcal{S}}^{n_i}\right)^T \boldsymbol{\mathcal{P}}_k^k, \quad (34)$$

$$\chi^k = -\boldsymbol{\mathcal{P}}_k^k\left(\Delta^k\right)^{-1}\left(\boldsymbol{\mathcal{P}}_k^k\right)^T {}^k\boldsymbol{\mathcal{S}}^{n_i}\boldsymbol{\tau}^{k+1} \quad (35)$$

and

$$\Delta^k = \left({}^k\boldsymbol{\mathcal{S}}^{n_i}\right)^T + \chi^k \left({}^k\boldsymbol{\mathcal{S}}^{n_i}\right)^T, \quad (36)$$

which yield for $p < k \leq n_i$

$$\boldsymbol{\Gamma}^p = \mathbf{U}, \qquad \text{and} \qquad \boldsymbol{\Gamma}^{k+1} = \Delta^{k+1}\boldsymbol{\Gamma}^k, \quad (37)$$

and

$$\boldsymbol{\Xi}^p = \mathbf{0}, \qquad \text{and} \qquad \boldsymbol{\Xi}^{k+1} = \Delta^{k+1}\boldsymbol{\Xi}^k + \boldsymbol{\chi}^{k+1}. \quad (38)$$

The required derivatives of each of the intermediate quantities defined by Eq. 32–38 are then determined by simply applying the chain rule of differentiation to each of these defining expressions. It is apparent from Eqs. 32–38 that the resulting expressions for these derivative terms will be composed only of derivative terms that have been previously evaluated and that are fixed in number (i.e., the number of these terms to be determined is the same for each body).

These quantities are in turn used to construct values for the generalized inertia and force matrices associated with the virtual body p_i, which has all the inertia and applied force information associated with bodies p through n_i implicitly embedded within it.

The resulting inertia and force matrices are given by

$${}^{o_i}\hat{\boldsymbol{\mathcal{I}}}^{p_i} = \sum_{j=0}^{m_i}\left\{\left(\boldsymbol{\Gamma}^{p_i+j}\right)^T \boldsymbol{\mathcal{I}}^{p_i+j} \left({}^{0_i}\boldsymbol{\mathcal{S}}^{p_i+j}\right)^T\right\}, \quad (39)$$

$$\hat{\boldsymbol{\mathcal{I}}}^{p_i;o_i} = \sum_{j=0}^{m_i}\left\{\left(\boldsymbol{\Gamma}^{p_i+j}\right)^T \boldsymbol{\mathcal{I}}^{p_i+j} \boldsymbol{\Gamma}_{p_i+j}\right\}, \quad (40)$$

and

$$\hat{\boldsymbol{\mathcal{F}}}_i^p = \sum_{j=0}^{m_i}\left\{\left(\boldsymbol{\Gamma}^{p_i+j}\right)^T \boldsymbol{\mathcal{F}}^{p_i+j}\right\} + \sum_{j=0}^{m_i}\left\{\left(\boldsymbol{\Gamma}^{p_i+j}\right)^T \boldsymbol{\mathcal{I}}^{p_i+j} \boldsymbol{\Xi}^{p_i+j}\right\} {}^{o_i}\boldsymbol{\mathcal{A}}_t^{n_i}, \quad (41)$$

requiring $O(\sum_i^{n_L} m_i) = O(m)$ operations overall.

3.4 Triangularization of equations

It has been shown (Featherstone 1983; Rosenthal 1990; Anderson 1990) that the use of relative generalized coordinates allows the equations of motion to effectively be triangularized, as part of an LU decomposition, as they are formed. Indeed, $\tilde{\mathcal{F}}^{p_i}$, as well as the inertia terms ${}^{o_i}\hat{\boldsymbol{\mathcal{I}}}^{p_i}$ and $\hat{\boldsymbol{\mathcal{I}}}^{p_i;o_i}$ (in association with ${}^{o_i}\bar{\boldsymbol{\mathcal{A}}}^k$, the acceleration of k with respect to loop reference frame 0_i, and $\bar{\boldsymbol{\mathcal{A}}}^{0_i}$, the acceleration of 0_i in the Newtonian reference frame N) may be recursively triangularized in much the same manner as with comparable terms in the standard state space $O(n)$ (Anderson 1990) algorithms for unconstrained systems. Specifically, for $0_i < k \leq p_i$

$$\left(\boldsymbol{\mathcal{P}}_k^k\right)^T \left[\hat{\boldsymbol{\mathcal{I}}}^{k;o_i}\bar{\boldsymbol{\mathcal{A}}}^{0_i} + {}^{o_i}\hat{\boldsymbol{\mathcal{I}}}^k {}^{o_i}\bar{\boldsymbol{\mathcal{A}}}^k + \tilde{\mathcal{F}}^k\right] = \mathbf{0}, \quad (42)$$

with the recursive relationships

$$\hat{\boldsymbol{\mathcal{I}}}^{k-1;o_i} = \boldsymbol{\mathcal{I}}^{k-1}\left({}^{o_i}\boldsymbol{\mathcal{S}}^{k-1}\right)^T + \boldsymbol{\mathcal{T}}^k \hat{\boldsymbol{\mathcal{I}}}^{k;o_i}, \quad (43)$$

$${}^{o_i}\hat{\boldsymbol{\mathcal{I}}}^{k-1} = \boldsymbol{\mathcal{I}}^{k-1} + \boldsymbol{\mathcal{T}}^k {}^{o_i}\hat{\boldsymbol{\mathcal{I}}}^k \left(\boldsymbol{\mathcal{S}}^k\right)^T, \quad (44)$$

$$\tilde{\boldsymbol{\mathcal{F}}}^{k-1} = \mathcal{F}^{k-1} + \boldsymbol{\mathcal{T}}^k \tilde{\boldsymbol{\mathcal{F}}}^k, \quad (45)$$

where $\boldsymbol{\mathcal{T}}^k$ is our triangularization operation matrix

$$\boldsymbol{\mathcal{T}}^k = \boldsymbol{\mathcal{S}}^k\left[\mathbf{U} - {}^{o_i}\hat{\boldsymbol{\mathcal{I}}}^k \boldsymbol{\mathcal{P}}_k^k \left(\boldsymbol{\mathcal{M}}_k\right)^{-1}\left(\boldsymbol{\mathcal{P}}_k^k\right)^T\right] \quad (46)$$

and

$$\boldsymbol{\mathcal{M}}_k = \left(\boldsymbol{\mathcal{P}}_k^k\right)^T {}^{o_i}\hat{\boldsymbol{\mathcal{I}}}^k \boldsymbol{\mathcal{P}}_k^k. \quad (47)$$

The process then continues recursively until body 0_i is reached, at which point ${}^{o_i}\bar{\boldsymbol{\mathcal{A}}}^{o_i} = \mathbf{0}$ and the associated inertia term are dropped. At this point, $\hat{\boldsymbol{\mathcal{I}}}^{k;o_i}$ and $\tilde{\boldsymbol{\mathcal{F}}}^k$ become synonymous with $\hat{\boldsymbol{\mathcal{I}}}^k$ and $\hat{\boldsymbol{\mathcal{F}}}^k$, respectively, for $k = 0_i$. The procedure then continues as with a totally unconstrained system.

Recursively proceeding inward from the terminal bodies p_i to the system base body, one obtains a general triangularization procedure to evaluate the total derivatives associated with ${}^{o_i}\hat{\boldsymbol{\mathcal{I}}}^k$, $\hat{\boldsymbol{\mathcal{I}}}^{k;o_i}$ $\hat{\boldsymbol{\mathcal{F}}}^k$, and $\boldsymbol{\mathcal{M}}_k$, namely,

$$\frac{\mathrm{d}\hat{\boldsymbol{\mathcal{I}}}^{k;o_i}}{\mathrm{d}\mathbf{p}_j} = \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j}\left[\boldsymbol{\mathcal{I}}^k\left({}^{o_i}\boldsymbol{\mathcal{S}}^k\right)^T\right] + \sum_{l\,\in\, Dist[k]}\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j}\left[\boldsymbol{\mathcal{T}}^l \hat{\boldsymbol{\mathcal{I}}}^{l;o_i}\right], \quad (48)$$

$$\frac{\mathrm{d}\,{}^{o_i}\hat{\boldsymbol{\mathcal{I}}}^k}{\mathrm{d}\mathbf{p}_j} = \frac{\mathrm{d}\boldsymbol{\mathcal{I}}^k}{\mathrm{d}\mathbf{p}_j} + \sum_{l\,\in\, Dist[k]}\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j}\left[\boldsymbol{\mathcal{T}}^l \hat{\boldsymbol{\mathcal{I}}}^{l;o_i}\left(\boldsymbol{\mathcal{S}}^l\right)^T\right], \quad (49)$$

$$\tilde{\boldsymbol{\mathcal{F}}}^k = \frac{\mathrm{d}\boldsymbol{\mathcal{F}}^k}{\mathrm{d}\mathbf{p}_j} + \sum_{l\,\in\, Dist[k]}\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j}\left(\boldsymbol{\mathcal{T}}^l \tilde{\boldsymbol{\mathcal{F}}}^l\right), \quad (50)$$

$$\frac{\mathrm{d}\boldsymbol{\mathcal{M}}_k}{\mathrm{d}\mathbf{p}_j} = \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j}\left[\boldsymbol{\mathcal{P}}_k^k \; {}^{o_i}\hat{\boldsymbol{\mathcal{I}}}^k \; \boldsymbol{\mathcal{P}}_k^k\right] , \tag{51}$$

and

$$\frac{\mathrm{d}\boldsymbol{\mathcal{M}}_k^{-1}}{\mathrm{d}\mathbf{p}_j} = \boldsymbol{\mathcal{M}}_k^{-1}\frac{\mathrm{d}\boldsymbol{\mathcal{M}}_k}{\mathrm{d}\mathbf{p}_j}\boldsymbol{\mathcal{M}}_k^{-1}. \tag{52}$$

An examination of Eqs. 48–52 clearly indicates that matrix manipulations of known quantities are the sole computations necessary to produce the required total derivatives. This desirable property substantially reduces the differentiation complexity encountered relative to that of more traditional direct differentiation approaches. Since the numbers of matrices involved are fixed for each body, and their dimensions are also fixed (either 6×6 or 6×1), these matrix manipulations can be performed at a fixed cost, which can be further reduced using special matrix manipulation classes. The entire triangularization procedure therefore leads to an efficient $O(n-m)$ operation overall.

The key purpose of using the presented triangularization procedure is to avoid explicitly formulating each element of $\boldsymbol{\mathcal{M}}$, $\boldsymbol{\mathcal{K}}$, and their associated derivatives. When the recursive triangularization is accomplished, all off-diagonal elements of $\mathbf{M}$ are implicitly computed and embedded in the values $\boldsymbol{\mathcal{M}}_k$. Proper manipulation (i.e., recursive back substitution) of these elements and the modified $\boldsymbol{\mathcal{K}}$ matrix generates the solutions for the sensitivities of the generalized accelerations $\dot{\mathbf{u}}_i$ $(i=1,\ldots,n)$. Specifically, the generalized accelerations and their associated sensitivities can be carried out in an orderly manner through the equations

$$\dot{\mathbf{u}}_k = \boldsymbol{\mathcal{M}}_k^{-1}\left(\boldsymbol{\mathcal{P}}_k^k\right)^T\left[\tilde{\boldsymbol{\mathcal{F}}}^k - \hat{\boldsymbol{\mathcal{I}}}^k\left({}^k\boldsymbol{\mathcal{S}}^k\right)^T\bar{\boldsymbol{\mathcal{A}}}^{Pr[k]}\right] , \tag{53}$$

$$\bar{\boldsymbol{\mathcal{A}}}^k = \left[\boldsymbol{\mathcal{S}}^k\bar{\boldsymbol{\mathcal{A}}}^{Pr[k]} + \boldsymbol{\mathcal{P}}_k^k\dot{\mathbf{u}}_k\right] , \tag{54}$$

and

$$\frac{\mathrm{d}\dot{\mathbf{u}}_k}{\mathrm{d}\mathbf{p}_j} = \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j}\left\{\boldsymbol{\mathcal{M}}_k^{-1}\left(\boldsymbol{\mathcal{P}}_k^k\right)^T\left[\tilde{\boldsymbol{\mathcal{F}}}^k - \hat{\boldsymbol{\mathcal{I}}}^k\left(\boldsymbol{\mathcal{S}}^k\right)^T\bar{\boldsymbol{\mathcal{A}}}^{Pr[k]}\right]\right\} ,$$

$$\frac{\mathrm{d}\bar{\boldsymbol{\mathcal{A}}}^k}{\mathrm{d}\mathbf{p}_j} = \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j}\left\{\boldsymbol{\mathcal{S}}^k\bar{\boldsymbol{\mathcal{A}}}^{Pr[k]} + \boldsymbol{\mathcal{P}}_k^k\dot{\mathbf{u}}_k\right\} \tag{55}$$

for the situation where k lies inboard of body 0_i (i.e. $0<k\leq 0_i$). In this case Eqs. 53–55 may be applied recursively as the sequence of calculations proceeds inward to the system base body 0, where

$$\bar{\boldsymbol{\mathcal{A}}}^0 = \mathbf{0}. \tag{56}$$

For situations where the body k lies between bodies 0_i and p_i within a closed loop i (i.e., $0_i<k\leq p_i$), these back-substitution relations are modified to

$$\dot{\mathbf{u}}_k = \boldsymbol{\mathcal{M}}_k^{-1}\left(\boldsymbol{\mathcal{P}}_k^k\right)^T\left(\hat{\boldsymbol{\mathcal{F}}}^k - \left[\hat{\boldsymbol{\mathcal{I}}}^{k;o_i}\bar{\boldsymbol{\mathcal{A}}}^{0_i} + {}^{o_i}\hat{\boldsymbol{\mathcal{I}}}^k\left(\boldsymbol{\mathcal{S}}^k\right)^T\,{}^{o_i}\bar{\boldsymbol{\mathcal{A}}}^{Pr[k]}\right]\right) , \tag{57}$$

$${}^{o_i}\bar{\boldsymbol{\mathcal{A}}}^k = \left(\boldsymbol{\mathcal{S}}^k\right)^T\,{}^{o_i}\bar{\boldsymbol{\mathcal{A}}}^{Pr[k]} + \boldsymbol{\mathcal{P}}_k^k\dot{\mathbf{u}}_k \tag{58}$$

and

$$\frac{\mathrm{d}\dot{\mathbf{u}}_k}{\mathrm{d}\mathbf{p}_j} = \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j}\left\{\boldsymbol{\mathcal{M}}_k^{-1}\left(\boldsymbol{\mathcal{P}}_k^k\right)^T\left(\hat{\boldsymbol{\mathcal{F}}}^k - \left[\hat{\boldsymbol{\mathcal{I}}}^{k;o_i}\bar{\boldsymbol{\mathcal{A}}}^{0_i} + {}^{o_i}\hat{\boldsymbol{\mathcal{I}}}^k\left(\boldsymbol{\mathcal{S}}^k\right)^T\,{}^{o_i}\bar{\boldsymbol{\mathcal{A}}}^{Pr[k]}\right]\right)\right\} , \tag{59}$$

$$\frac{\mathrm{d}\,{}^{o_i}\bar{\boldsymbol{\mathcal{A}}}^k}{\mathrm{d}\mathbf{p}_j} = \frac{\mathrm{d}}{\mathrm{d}\mathbf{p}_j}\left\{\left(\boldsymbol{\mathcal{S}}^k\right)^T\,{}^{o_i}\bar{\boldsymbol{\mathcal{A}}}^{Pr[k]} + \boldsymbol{\mathcal{P}}_k^k\dot{\mathbf{u}}_k\right\} . \tag{60}$$

To this end, the key quantities $\mathrm{d}\dot{\mathbf{u}}_k/\mathrm{d}\mathbf{p}_j$ in the first-order sensitivity analysis are determined in a fully recursive manner. Inspecting the back-substitution procedure as indicated in Eqs. 53–60 one observes that neither the number of terms nor the required number of operations per state derivative increases as the procedures ensue. This implies, again, that a constant number of operations is required to solve for $\mathrm{d}\dot{\mathbf{u}}_k/\mathrm{d}\mathbf{p}_j$ as one moves recursively from one generalized acceleration to the next. Consequently, the back-substitution process yields $O(n)$ performance overall for one design variable. For an entire p design variable, the proposed solution scheme is able to achieve an $O(\mathrm{p}n)$ operation overall for obtaining all solutions associated with the p sets of sensitivity governing equations.

4 Examples

Three mechanical systems are presented that are used to demonstrate and validate the presented algorithms. In each of the cases presented, the system being considered consists of identical interconnected bodies, with the single design variable chosen to be the length of each body. Defining the problems in this (albeit artificial) manner represents a "worst case" (from the point of view of computational expense) given that a change in this single design variable will affect the kinematic and kinetic terms associated with every body of each of these systems.

4.1 Simple chain system

The first case, a simple spatial unconstrained chain system, is comprised of n identical uniform massive rods, each connected to one another by single-degree-of-freedom revolute joints. The sensitivity of the system

kinetic energy to the above-described common length design variable was then determined using the present recursive algorithm as well as more traditional $O(n^3)$ based algorithms. Figure 3 presents a comparison of the CPU time required to determine this sensitivity using each of these approaches. In the case of the $O(n^3)$-based approach (which actually produced $O(n^4)$ overall performance), the equations of motion and sensitivity were developed symbolically and then coded using efficient coding practices in an effort to produce the best possible performance given the underlying algorithm. It is clear from Fig. 3 that for systems involving more than a few unconstrained degrees of freedom n, the presented algorithm is much faster. Indeed, for a system involving 12 generalized coordinates, the presented algorithm was able to determine the desired first-order sensitivity information by a factor in excess of 4000 more quickly than the traditional $O(n^3)$-based aapproach.

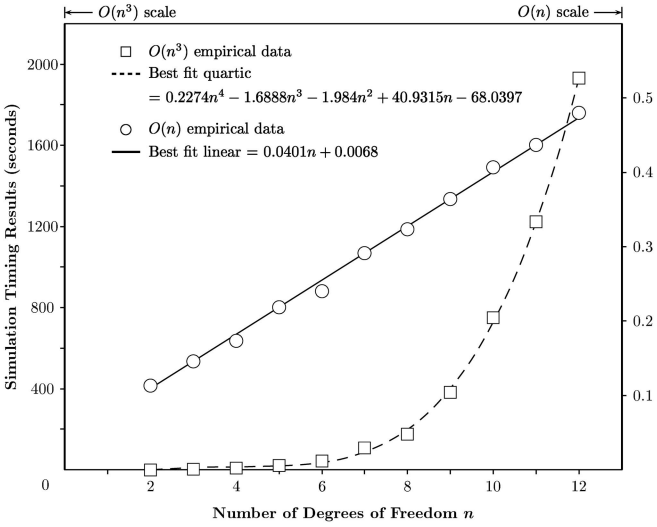

Fig. 3 CPU time required for derivative operation by $O(n)$-based and $O(n^3)$-based methodologies

4.2 Simple closed chain system

The second case consists of the simple closed-loop system shown in Fig. 4 involving a simple chain of identical interconnected bodies constrained at each end. As with

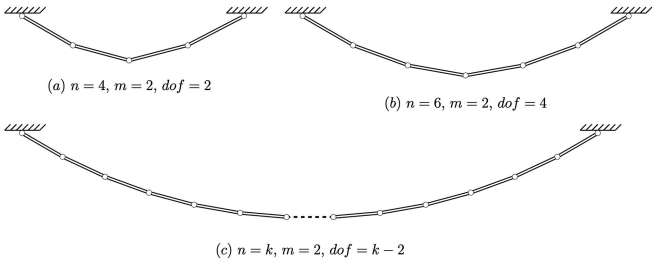

Fig. 4 Schematic of closed-loop chain system

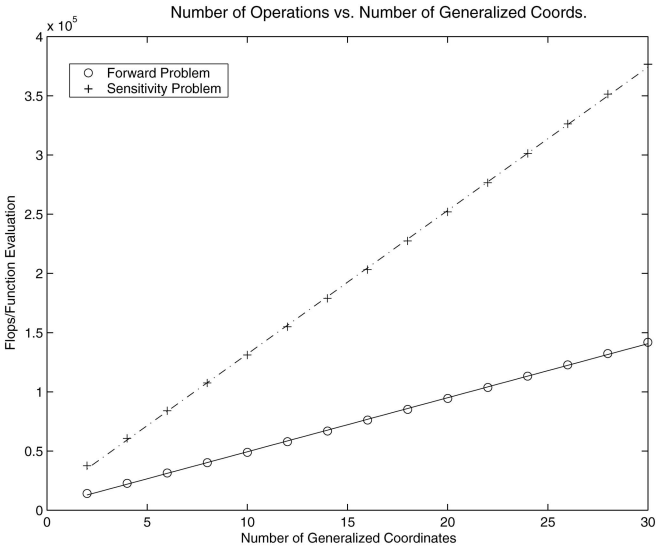

Fig. 5 Number of flops associated with forward problem simulation and sensitivity determination of closed chain

case 1, each body consisted of a uniform massive rod. For this closed chain case the computational cost in floating point operations (flops) per function evaluation as a function of the number of bodies n is presented in Fig. 5 as determined for the simulation and sensitivity problems. This figure demonstrates that the computational cost associated with this "worst case" sensitivity analysis is linear in the number of generalized coordinates and roughly twice as expensive as performing the forward problem analysis.

4.3 Heavily constrained system

The final case consists of the heavily constrained system of identical bodies connected in a ladder-type formation as shown in Fig. 6. In the situation where all joints used are two-degree-of-freedom Hooke's joints, the system of L independent closed loops will involve $N = 2L + 1$ bodies, $n = 2N = 4L + 2$ generalized coordinates, $m = 4L$ independent algebraic constraints, and only 2 degrees of freedom. This system represents a situation where previously developed sensitivity algorithms are likely to pay a high computational price both for the number of generalized coordinates n and the number of algebraic constraints m.

Figure 7 indicates the projected performance characteristics for the analysis of such a heavily constrained problem: (i) If the system constraints are not considered; (ii) the system constraints are considered, but a more conventional so-called $O(n)$ approach is used; and (iii) the $O(n+m)$ algorithm presented here is used. Figure 7 indicates that in situations such as this, where the number of constraints m is of the same order as the number of generalized coordinates n, then the so-called "order-n" algorithms actually offer $O(n^3)$ performance and generally may not perform even as well as well-written $O(n^3)$-

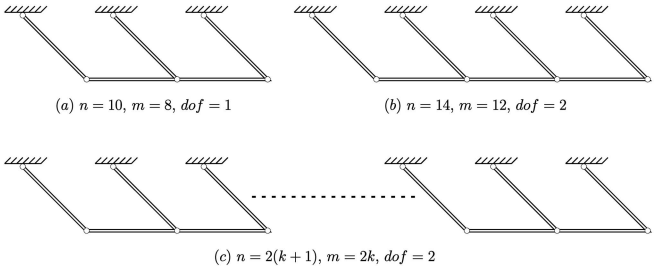

Fig. 6 Schematic of heavily constrained system

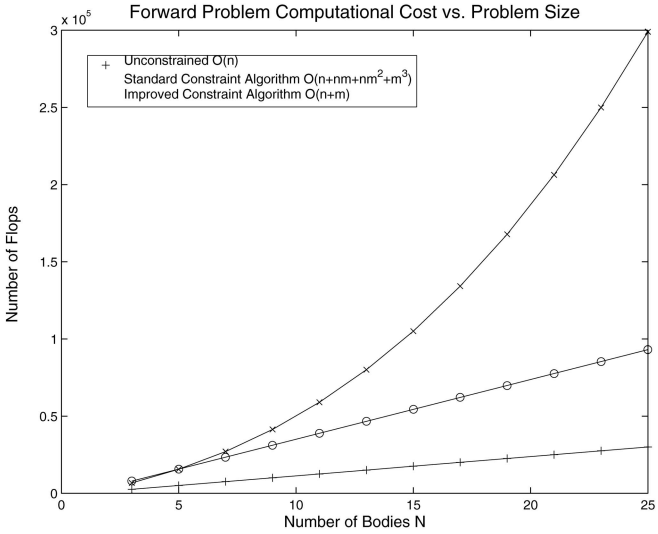

Fig. 7 Projected CPU time required for forward problem simulation of this system

based algorithms. However, the presented algorithm does not pay such a price, being linear in both the number of generalized coordinates n and the number of algebraic constraints m.

5 Conclusions

A fully recursive sensitivity analysis for modestly to heavily constrained multi-rigid-body dynamic systems is presented. The use of recursive relationships coupled with the local representations of dynamic quantities permits a great reduction in the total number of matrix manipulation and derivative operations required in performing sensitivity analysis for complex multibody systems relative to that encountered in traditional direct differentiation approaches. The presented $O(n)$-based approach has been shown to yield first-order sensitivity information in at most $O(n+m)$ operations overall. This represents a significant improvement relative to the best performance possible using more traditional $O(n^3)$-based dynamic formations that would yield roughly $O(n^4+n^2m^2+nm^3)$ performance or previously developed low-order sensitivity algorithms offering $O(n+nm+nm^2+m^3)$ performance when applied to the same pervasive design variable example problems.

The present formulation is quite straightforward with regard to both analytical formulation and numerical implementation, making a general computer code easy to develop and use. Further, the entire procedure can be (and is best) performed simultaneously with the forward dynamics problem, which avoids costly postprocessing that would be associated with a totally separate sensitivity analysis. This method is very well suited for sensitivity analysis of large-scale multibody dynamic systems and should perform extremely well relative to competing methods for systems where n and/or m is greater than 5 or 6. Indeed, for more realistic choices of design variables (i.e., variables that only directly affect a single body or joint), the computational savings that can be realized through the presented method and its local derivative property (Hsu and Anderson 2001, 2002) are even greater.

As such, the presented method presents a powerful, easily implementable approach that should now make it possible to rigorously perform sensitivity analysis on complex multibody dynamic systems that were previously so large, complex, and expensive as to be intractable.

Acknowledgements The authors are grateful for the support of this work through NSF Grant 9733684.

References

Anderson, K.S. 1990: Recursive Derivation of Explicit Equations of Motion for Efficient Dynamic/Control Simulation of Large Multibody Systems, *Ph.D. Dissertation*, Stanford University, CA

Anderson, K.S. 1992: An order-n formulation for motion simulation of general constrained multi-rigid-body systems. *Computers and Structures* **43**(3), 565–572

Anderson, K.S. 2001: Improved order-n Performance Algorithm for the Simulation of Constrained Multi-Rigid-Body Dynamic Systems. *Proceedings ASME Design Engineering Technical Conference (DETC 2001)*, Pittsburgh Sept. 9–13, 2001. Paper DETC2001/VIB-21334

Anderson, K.S.; Critchley, J.H. 2003: Improved Order-n Performance Algorithm for the Simulation of Constrained Multi-Rigid-Body Systems. *Multibody System Dynamics* **9**, 185–212

Barthelemy, J.-F.M.; Hall, L.E. 1995: Automatic differentiation as a tool in engineering design. *Structural Optimization* **9**, 76–82

Bestle, D.; Eberhard, P. 1992: Analyzing and optimizing multibody systems. *Mechanics of Structures and Machines* **20**(1), 67–92

Bestle, D.; Seybold, J. 1992: Sensitivity analysis of constrained multibody systems. *Archive of Applied Mechanics* **62**, 181–190

Bischof, C.H. 1996: On the automatic differentiation of computer programs and an application to multibody systems. *IUTAM Symposium on Optimization of Mechanical Systems*, Kluwer Academic, pp. 41–48

Chang, C.O.; Nikravesh, P.E. 1985:Optimal design of mechanical systems with constraint violation stabilization method. *Journal of Mechanisms, Transmissions, and Automation in Design* **107**, 493–498

Dias, J.M.P.; Pereira, M.S. 1997: Sensitivity analysis of rigid-flexible multibody systems. *Multibody System Dynamics* **1**(3), 303–322

Eberhard, P. 1996: Adjoint variable method for sensitivity analysis of multibody systems interpreted as a continuous, hybrid form of automatic differentiation. *Proceedings of the 2nd International Workshop on Computational Differentiation*, Santa Fe, NM, USA, 12–14 Feb, pp. 319–328

Featherstone, R. 1983: The calculation of robot dynamics using articulated-body inertias. *International Journal of Robotics Research* **2**(1), 13–30

Hollerbach, J.M. 1980: A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity. *IEEE Transaction on Systems, Man, and Cybernetics* **SMC-10**(11), 730–736

Haug, E.J.; Arora, J.S. 1979: *Applied Optimal Design*. New York: John Wiley & Sons, Inc.

Haug, E.J.; Wehage, R.A.; Mani, N.K. 1984: Design sensitivity analysis of large-scaled constrained dynamic mechanical systems. *Transactions of the ASME* **106**, 156–162

Hsu, Y. 2001: Order-n Analytic Sensitivity Analysis for Multibody Dynamic Systems Optimization. *Ph.D. Dissertation* Department of Mechanical Engineering, Aeronautical Engineering and Mechanics. Rensselaer Polytechnic Institute, Nov.

Hsu, Y.; Anderson, K.S. 2001: Low operational order analytic sensitivity analysis for tree-type multibody dynamic systems. *Journal of Guidance, Control, and Dynamics* **24**(6), 1133–1143

Hsu, Y.; Anderson, K.S. 2002: Recursive Sensitivity Analysis for Constrained Multi-rigid-body Dynamic Systems Design Optimization. *Structural and Multidisciplinary Optimization* **24**(4), 312–324

Jain, A.; Rodriguez, G. 1999: Sensitivity analysis for multibody systems using spatial operators. *International Conference on Scientific Computation and Differential Equations* (SciCADE'99), Aug., Fraser Island, Queensland, Australia.

Kane, T.R.; Levinson, D.A. 1985: *Dynamics: Theory and Applications*. New York: McGraw-Hill, Inc.

Pagalday, J.M.; Aranburu, I. 1996: Multibody dynamics optimization by direct differentiation methods using object oriented programming. *IUTAM Symposium on Optimization of Mechanical Systems*, Kluwer Academic, pp. 213–220

Pradhan, S.; Modi, V.J.; Misra, A.K. 1997: Order n formulation for flexible multibody systems in tree topology: lagrangian approach. *Journal of Guidance, Control, and Dynamics* **20**(4), 665–672

Rosenthal, D.E. 1990: An order n formulation for robotic systems. *Journal of Astronautical Sciences* **38**(4), 511–529

Schaechter, D.B.; Levinson, D.A.; Kane, T.R. 1997: *AUTOLEV User's Manual Version 3.2*, Online Dynamics, Inc., Sunnyvale, CA

Serban, R.; Haug, E.J. 1998: Analytical derivatives for multibody system analyses. *Mechanics of Structures and Machines* **26**(2), 145–173

Tak, T. 1990: A recursive approach to design sensitivity analysis of multibody systems using direct differentiation. *Ph.D. Dissertation*, The University of Iowa

Stejskal, V.; Valášek, M. 1996: *Kinematics and Dynamics of Machinery*. New York: Marcel Dekker, Inc.

Wehage, R.A.; Haug, E.J. 1982: Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems. *Transaction of ASME, Journal of Machine Design* **104**, 247–255