A Practical Approach for the Linearization of the Constrained Multibody Dynamics Equations

Dan Negrut[*]
Department of Mechanical Engineering
The University of Wisconsin
Madison, WI-53706
negrut@wisc.edu

Jose L. Ortiz
MSC.Software
Ann Arbor, Michigan 48105
jose.ortiz@mscsoftware.com

February 24, 2006

Abstract

The paper presents an approach to linearize the set of index 3 nonlinear Differential Algebraic Equations (DAE) that govern the dynamics of constrained mechanical systems. The proposed method handles heterogeneous systems that might contain flexible bodies, friction, control elements (user-defined differential equations), and non-holonomic constraints. Analytically equivalent to a state-space formulation of the system dynamics in Lagrangian coordinates, the proposed method augments the governing equations and then computes a set of sensitivities that provide the linearization of interest. The attributes associated with the method are the ability to handle large heterogeneous systems, ability to linearize the system in terms of arbitrary user-defined coordinates, and straightforward implementation. The proposed approach has been released in the 2005 version of the MSC.ADAMS/Solver(C++) and compares favorably with a reference method previously available. The approach was also validated against MSC.NASTRAN and experimental results.

INTRODUCTION

The state of a multibody system at the position level is represented herein by an array $\mathbf{q} = [q_1, \dots, q_p]^T$ of generalized coordinates. The velocity of the system is most often described by the array of generalized velocities $\dot{\mathbf{q}} = [\dot{q}_1, \dots, \dot{q}_p]^T$, although, as discussed later, this condition does not necessarily have to hold. The set of generalized coordinates and velocities can be selected in a multitude of ways [1, 6, 7, 9]. The generalized coordinates used here are Cartesian coordinates for position and Euler angles for orientation of body centroidal reference frames. For each body i, its position is described by the vector $\mathbf{r}_i = [x_i, y_i, z_i]^T$, while its orientation is given by the array of local 3-1-3 Euler angles $\mathbf{e}_i = [\psi_i, \theta_i, \phi_i]^T$ (see, for instance, [19]). Consequently, for a mechanical system containing n_b bodies, $\mathbf{q} = \left[\mathbf{r}_1^T, \mathbf{e}_1^T, \dots, \mathbf{r}_{n_b}^T, \mathbf{e}_{n_b}^T\right]^T \in \mathbf{R}^p,\ p = 6n_b$. The set of position generalized coordinates is augmented with deformation modes when flexible bodies are present in the model.

In any constrained mechanical system, joints connecting bodies restrict their relative motion and impose constraints on the generalized coordinates. These kinematic constraints are formulated as time dependent algebraic expressions involving generalized coordinates,

$$\mathbf{C}(\mathbf{q}, t) = \left[\begin{array}{ccc} C_1(\mathbf{q}, t) & \dots & C_m(\mathbf{q}, t) \end{array}\right]^T = \mathbf{0} \tag{1a}$$

where m is the total number of constraint equations that must be satisfied by the generalized coordinates throughout the simulation. It is assumed here that the m constraint equations are independent. Although the implementation of the proposed method handles non-holonomic constraints, in order to keep the presentation simple, the case of

[*] Address all correspondence to this author.

holonomic constraints is assumed in what follows. Where necessary, short remarks will be provided to indicate how the method accommodates the non-holonomic case. Differentiating Eq. (1a) with respect to time leads to the velocity kinematic constraint equation

$$\mathbf{C_q}(\mathbf{q}, t)\,\dot{\mathbf{q}} + \mathbf{C}_t(\mathbf{q}, t) = \mathbf{0} \tag{1b}$$

where the over-dot denotes differentiation with respect to time and the subscript denotes partial differentiation, $[\mathbf{C_q}]_{ij} = \left[\frac{\partial C_i}{\partial q_j}\right]$, for $1 \le i \le m$, and $1 \le j \le p$. The acceleration kinematic constraint equations are obtained by differentiating Eq. (1b) with respect to time,

$$\mathbf{C_q}(\mathbf{q}, t)\,\ddot{\mathbf{q}} + \left(\mathbf{C_q}(\mathbf{q}, t)\dot{\mathbf{q}}\right)_{\mathbf{q}}\dot{\mathbf{q}} + 2\mathbf{C}_{\mathbf{q}t}(\mathbf{q}, t)\,\dot{\mathbf{q}} + \mathbf{C}_{tt}(\mathbf{q}, t) = \mathbf{0} \tag{1c}$$

Equations (1a)–(1c) characterize the admissible motion of the mechanical system.

The state of the mechanical system changes in time under the effect of applied forces. The time evolution of the system is governed by the Lagrange multiplier form of the constrained equations of motion (see, for instance, [9], [17]),

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}^T_{\mathbf{q}}(\mathbf{q})\lambda = \mathbf{Q}\left(\dot{\mathbf{q}}, \mathbf{q}, t\right) \tag{1d}$$

where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{p \times p}$ is the generalized mass and $\mathbf{Q}\left(\dot{\mathbf{q}}, \mathbf{q}, t\right) \in \mathbb{R}^p$ is the action (as opposed to the reaction $\mathbf{C}^T_{\mathbf{q}}(\mathbf{q})\lambda$) force acting on the generalized coordinates $\mathbf{q} \in \mathbb{R}^p$. These equations are neither linear nor ordinary differential. Additionally, the solution $\mathbf{q}(t)$ must also satisfy the kinematic constraint equations in Eq. (1a). These constraint equations lead, in Eq. (1d), to the presence of the reaction force $\mathbf{C}^T_{\mathbf{q}}(\mathbf{q})\lambda$, where $\lambda \in \mathbb{R}^m$ is the Lagrange multiplier associated with the kinematic constraints.

Equations (1a) and (1d) represent a set of index 3 DAE [4]. While the analytical solution of this problem automatically satisfies the velocity and acceleration constraint equations (Eqs. (1b) and (1c)), this ceases to be the case for the numerical solution of this highly nonlinear problem. The index 3 problem can be numerically solved in many ways, including a direct approach [16], a state-space reduction (SSR) [11, 12, 21, 18], an index reduction method [8], and a stabilization approach [3]. While these approaches are designed for finding the time evolution of a mechanical system (dynamic analysis), and their advantages and weaknesses are mostly well understood, when it comes to a linear analysis of the index 3 DAE associated with a mechanical system, the picture is not that clear. The approach proposed in this paper draws on a well-established approach for dynamic analysis, namely the SSR method. In this method, if a Lagrangian set of coordinates [17], that is, a minimal set of generalized coordinates, is chosen, locally the dynamics of the system is represented as a set of second-order ordinary differential equations (ODE), rather than the index 3 DAE introduced above. The SSR-based DAE-to-ODE transformation enables the linear analysis of the problem to be carried out in the ODE framework, even though the problem is originally formulated as an index 3 DAE. The challenge with this analytically sound approach remains to render the SSR tractable and efficient for complex mechatronic systems.

The state-space representation of a system is not unique; and the coordinate partitioning [21], the null space [12], and a $\mathbf{QR}$-based approach [11] are only some of the methodologies used in the past to obtain a set of Lagrangian coordinates. In the coordinate partitioning-based SSR, which is the approach that this work builds on, the starting point is a partitioning of $\mathbf{q}$ in independent coordinates (the Lagrangian coordinates), $\mathbf{y} \in \mathbf{R}^z$, $z = n - m$, and dependent coordinates, $\mathbf{x} \in \mathbf{R}^m$. The partitioning is based on two mappings $\nu : \{1, 2, \ldots, z\} \to S_{indep}$ and $\mu : \{1, 2, \ldots, m\} \to S_{dep}$, with $S_{indep} \cup S_{dep} = \{1, 2, \ldots, p\}$ and $S_{indep} \cap S_{dep} = \emptyset$.

$$\mathbf{y}[i] = \mathbf{q}[\nu(i)]\ ,\ \mathit{1} \le i \le z,\ \text{and}\ \mathbf{x}[j] = \mathbf{q}[\mu(j)]\ ,\ \mathit{1} \le j \le m \tag{2}$$

As pointed out in [21], the partitioning is such that the sub-Jacobian of the constraints with respect to $\mathbf{x}$ is nonsingular:

$$\det\left(\mathbf{C_x}(\mathbf{q})\right) \neq 0 \tag{3}$$

Such a partitioning can be found starting with a set of consistent generalized coordinates $\mathbf{q}_0$, that is, which satisfy Eq. (1a). The constraint Jacobian matrix $\mathbf{C_q}$ is evaluated and numerically factored by using the Gauss-Jordan algorithm with full pivoting [2]:

$$\mathbf{C}_{\mathbf{q}}(\mathbf{q}_0) \mapsto (\mathrm{Gauss} - \mathrm{Jordan}) \mapsto [\mathbf{C}_{\mathbf{x}}(\mathbf{q}_0)|\mathbf{C}_{\mathbf{y}}(\mathbf{q}_0)] \tag{4}$$

Equations (1a) through (1d) can be rewritten in the associated partitioned form [9],

$$\mathbf{M}^{\mathbf{yy}}(\mathbf{x},\mathbf{y})\ddot{\mathbf{y}} + \mathbf{M}^{\mathbf{yx}}(\mathbf{x},\mathbf{y})\ddot{\mathbf{x}} + \mathbf{C}_{\mathbf{y}}^T(\mathbf{x},\mathbf{y})\lambda = \mathbf{Q}^{\mathbf{y}}(\dot{\mathbf{x}},\dot{\mathbf{y}},\mathbf{x},\mathbf{y},t) \tag{5a}$$

$$\mathbf{M}^{\mathbf{xy}}(\mathbf{x},\mathbf{y})\ddot{\mathbf{y}} + \mathbf{M}^{\mathbf{xx}}(\mathbf{x},\mathbf{y})\ddot{\mathbf{x}} + \mathbf{C}_{\mathbf{x}}^T(\mathbf{x},\mathbf{y})\lambda = \mathbf{Q}^{\mathbf{x}}(\dot{\mathbf{x}},\dot{\mathbf{y}},\mathbf{x},\mathbf{y},t) \tag{5b}$$

$$\mathbf{C}(\mathbf{x},\mathbf{y},t) = \mathbf{0} \tag{5c}$$

$$\mathbf{C}_{\mathbf{x}}(\mathbf{x},\mathbf{y},t)\dot{\mathbf{x}} + \mathbf{C}_{\mathbf{y}}(\mathbf{x},\mathbf{y},t)\dot{\mathbf{y}} + \mathbf{C}_t(\mathbf{x},\mathbf{y},t) = \mathbf{0} \tag{5d}$$

$$\mathbf{C}_{\mathbf{x}}(\mathbf{x},\mathbf{y},t)\ddot{\mathbf{x}} + \mathbf{C}_{\mathbf{y}}(\mathbf{x},\mathbf{y},t)\ddot{\mathbf{y}} = \tau(\dot{\mathbf{x}},\dot{\mathbf{y}},\mathbf{x},\mathbf{y},t) \tag{5e}$$

where τ is defined as the sum of all the terms in the left side of Eq. (1c) that do not depend on acceleration $\ddot{\mathbf{q}}$. The partitioning of Eqs. (5a)–(5e) is induced by the partitioning of the generalized coordinates in Eq. (2). For instance, $\mathbf{M}^{\mathbf{yx}}[i,j] = \mathbf{M}[\nu(i),\mu(j)]$, for $1 \leq i \leq z, 1 \leq j \leq m$, while $\mathbf{Q}^{\mathbf{x}}[j] = \mathbf{Q}[\mu(j)]$, for $1 \leq j \leq m$. Likewise, a partial with respect to the dependent set of coordinates $\mathbf{x}$ is obtained by gathering the columns $\mu(1)$ through $\mu(m)$ of the derivative with respect to the generalized coordinates $\mathbf{q}$. The remaining z columns provide the partial with respect to the independent coordinates $\mathbf{y}$.

Based on the nonsingularity condition of Eq. (3), a second-order state-space ODE (SSODE) can be determined for the time evolution of $\mathbf{y}$ [21]:

$$\widehat{\mathbf{M}}(\mathbf{y})\,\ddot{\mathbf{y}} = \widehat{\mathbf{Q}}(\dot{\mathbf{y}},\mathbf{y},t) \tag{6a}$$

where

$$\widehat{\mathbf{M}} = \mathbf{M}^{\mathbf{yy}} - \mathbf{M}^{\mathbf{yx}}\mathbf{C}_{\mathbf{x}}^{-1}\mathbf{C}_{\mathbf{y}} - \mathbf{C}_{\mathbf{y}}^T\mathbf{C}_{\mathbf{x}}^{-T}\left[\mathbf{M}^{\mathbf{xy}} - \mathbf{M}^{\mathbf{xx}}\mathbf{C}_{\mathbf{x}}^{-1}\mathbf{C}_{\mathbf{y}}\right] \tag{6b}$$

$$\widehat{\mathbf{Q}} = \mathbf{Q}^{\mathbf{y}} - \mathbf{M}^{\mathbf{yx}}\mathbf{C}_{\mathbf{x}}^{-1}\tau - \mathbf{C}_{\mathbf{y}}^T\mathbf{C}_{\mathbf{x}}^{-T}\left[\mathbf{Q}^{\mathbf{x}} - \mathbf{M}^{\mathbf{xx}}\mathbf{C}_{\mathbf{x}}^{-1}\tau\right] \tag{6c}$$

The SSODE in Eq. (6a) is well defined because the matrix $\widehat{\mathbf{M}}(\mathbf{y})$ is positive definite, and it is subsequently linearized to understand the local behavior of the mechanical system. If one defines $\mathbf{w} = \dot{\mathbf{y}}$, and $\alpha = \left[\mathbf{w}^T\ ,\ \mathbf{y}^T\right]^T \in \mathbb{R}^{2z}$, Eq. (6a) becomes $\dot{\alpha} = \mathbf{g}(\alpha)$, where $\mathbf{g}(\alpha) = \left[\left(\widehat{\mathbf{M}}^{-1}(\mathbf{y})\,\widehat{\mathbf{Q}}(\mathbf{w},\mathbf{y},t)\right)^T\ ,\ \mathbf{w}^T\right]^T$. The linearization of this equivalent first-order differential equation leads to

$$\delta\dot{\alpha} = \begin{bmatrix} \widehat{\mathbf{M}}^{-1}(\mathbf{y})\,\widehat{\mathbf{Q}}_{\mathbf{w}}(\mathbf{w},\mathbf{y},t) & \left(\widehat{\mathbf{M}}^{-1}(\mathbf{y})\,\widehat{\mathbf{Q}}(\mathbf{w},\mathbf{y},t)\right)_{\mathbf{y}} \\ \mathbf{I}_z & \mathbf{0}_{z\times z} \end{bmatrix} \delta\alpha = \begin{bmatrix} \ddot{\mathbf{y}}_{\dot{\mathbf{y}}} & \ddot{\mathbf{y}}_{\mathbf{y}} \\ \mathbf{I}_z & \mathbf{0}_{z\times z} \end{bmatrix} \delta\alpha \tag{7}$$

where $\mathbf{I}_z$ is the identity matrix of dimension z. Linearization in this form is of limited practical interest however, and both in control theory and in linear analysis of structures the representation sought is

$$\hat{\mathbf{M}}\ddot{\mathbf{y}} + \mathbf{C}\dot{\mathbf{y}} + \mathbf{K}\mathbf{y} = \hat{\mathbf{Q}} \tag{8}$$

Here the $z \times z$ matrices $\hat{\mathbf{M}}$, $\mathbf{C}$, $\mathbf{K}$, and $\hat{\mathbf{Q}} \in \mathbb{R}^z$ are constant, and correspond to a linearization of the dynamic system at time t_0 in a consistent configuration $\mathbf{q}_0$, $\dot{\mathbf{q}}_0$, $\ddot{\mathbf{q}}_0$. More precisely, based on Eqs. (6b) and (6c) $\hat{\mathbf{M}} = \widehat{\mathbf{M}}(\mathbf{y}_0)$, and $\hat{\mathbf{Q}} = \widehat{\mathbf{Q}}(\dot{\mathbf{y}}_0,\mathbf{y}_0,t_0)$, while $\mathbf{C}$ and $\mathbf{K}$ are determined such that if this linear second-order differential equation is reduced to first order, it will produce the same sensitivities as the fully nonlinear case. Based on this condition,

$$\mathbf{K} = -\hat{\mathbf{M}}^{-1}\ddot{\mathbf{y}}_{\mathbf{y}} \qquad \mathbf{C} = -\hat{\mathbf{M}}^{-1}\ddot{\mathbf{y}}_{\dot{\mathbf{y}}} \tag{9}$$

Because of the extremely contrived form of $\widehat{\mathbf{M}}(\mathbf{y})$ and $\widehat{\mathbf{Q}}(\dot{\mathbf{y}},\mathbf{y},t)$, linearization in this form is difficult (particularly evaluating the partial derivative $\ddot{\mathbf{y}}_{\mathbf{y}}$). Specifically, the linearization of Eq. (6a) is laborious because of the level of matrix manipulation required by this approach. Moreover, the direct approach is impractical for a general-purpose mechanical simulation package such as MSC.ADAMS [13] when the index 3 DAE system is usually augmented with additional equations that further complicate the problem:

1. *Ordinary differential equations* that, in the most general case, are provided in implicit form:

$$\mathbf{d}(\dot{\mathbf{X}}, \mathbf{X}, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \lambda, \mathbf{V}, \mathbf{F}, t) = \mathbf{0}_{n_d} \tag{10a}$$

   This type of differential equations is encountered, for example, when controls are active in the system, such as cars with antilock braking system (ABS) or active suspension control. The state of the controller is $\mathbf{X}$, its time derivative is $\dot{\mathbf{X}}$, and the assumption is that in its implicit form Eq. (10a) properly and uniquely defines $\dot{\mathbf{X}}$ as a function of the state of the system through the user-specified function $\mathbf{d}$. If n_d represents the number of differential states, then $\mathbf{X}, \mathbf{d} \in \mathbb{R}^{n_d}$.

2. *User-defined variables*, which can technically be regarded as aliases or definition equations. A set of n_v user defined variables $\mathbf{V} \in \mathbb{R}^{n_v}$ is typically specified through an equation of the form

$$\mathbf{V} - \mathbf{v}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{X}, \lambda, \mathbf{V}, \mathbf{F}, t) = \mathbf{0}_{n_v} \tag{10b}$$

   which, during the solution sequence, is solved (or, rather, evaluated) simultaneously with the equations of motion and the kinematic constraint equations. Here, $\mathbf{v} \in \mathbb{R}^{n_v}$ is a user-defined function that depends on other system states.

3. *External force definition*, $\mathbf{F}$, which allows the user to define the set of n_f applied forces $\mathbf{F} \in \mathbb{R}^{n_f}$ that act on the system. This is the mechanism through which a complex tire model can be interfaced to a vehicle model, that supports user-defined bushing elements, custom nonlinear damper and friction models, and the like.

$$\mathbf{F} - \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{X}, \dot{\mathbf{X}}, \lambda, \mathbf{V}, \mathbf{F}, t) = \mathbf{0}_{n_f} \tag{10c}$$

Proposed here is a *different but equivalent* method that simultaneously handles the set of equations (1) and (10). In doing so, the method augments the governing equations and relies on sparse matrix factorization followed by the solution of a multiple right-side linear system to compute the quantities of interest.

LINEARIZING THE EQUATIONS OF MOTION

Sensitivities as Induced by an Algebraic Equation. The Augmented Approach.

Consider an equation of the form $\psi(x, y) = 0$, with $\psi(x, y) = x^2 - 4y$; given a value for the "independent" variable $y > 0$, a value for x can be readily computed. The question is, how does the defining equation $\psi(x, y) = 0$ match the change in x to a small perturbation in y? Accepting that for the "dependent" variable $x = x(y)$, and introducing a new function $\phi(y) = \psi(x(y), y)$, taking the partial derivative of ϕ with respect to the "independent" variable y produces the following (a subscript indicates partial derivative with that quantity).

$$\phi_y = \psi_x \, x_y + \psi_y \tag{11}$$

Since necessarily $\phi_y = 0$, as long as

$$\psi_x \neq 0 \tag{12}$$

the sensitivity of interest at a consistent operation point $\mathbf{q}_0 = [x_0, y_0]^T$, that is, for which $\psi(\mathbf{q}_0) = 0$, is computed as $x_y = -[\psi_x]^{-1} \, \psi_y$.

The augmented approach that the proposed linearization method relies on is slightly different. Thus, the problem of computing the first-order (linear) sensitivity of x with respect to y is equivalent to that of computing the first-order (linear) sensitivity of x with respect to a "virtual" state $\bar{y}$ that the problem is augmented with

$$\begin{cases} \psi(x, y) &= 0 \\ y &= \bar{y} \end{cases}$$

or equivalently,

$$\left\{ \begin{array}{lcl} \psi(\mathbf{q}) & = & 0 \\ \mathbf{B}\,\mathbf{q} & = & \bar{y} \end{array} \right. \tag{13}$$

where $\mathbf{B} = [0\ \ 1]$ is a Boolean matrix used to indicate the "independent" coordinate. In anticipation of future notation, this system of equations is expressed as

$$\mathcal{F}(\mathbf{q};\bar{y}) = \mathbf{0}$$

where $\bar{y}$ can be regarded as a parameter. The condition

$$\frac{\partial \mathcal{F}}{\partial \bar{y}} = \mathbf{0}$$

leads to first-order sensitivity linear system

$$\begin{bmatrix} \frac{\partial \psi}{\partial x} & \frac{\partial \psi}{\partial y} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial \bar{y}} \\ \frac{\partial y}{\partial \bar{y}} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

No low-level manipulations, that is, partitioning of $\psi_{\mathbf{q}}$, are required to compute the sensitivities; and all the partitioning information is encapsulated in the matrix $\mathbf{B}$. The determinant of the coefficient matrix above is $\frac{\partial \psi}{\partial x}$, and the same sufficient condition of Eq. (12) renders the augmented approach well defined. It is easy to see that the sensitivity of interest in this case is $\frac{\partial x}{\partial \bar{y}}$.

The Proposed Augmented Approach to Linearization

The method proposed was implemented in a general-purpose simulation software package, and as such the formulation of the equations of motion was brought to a slightly more general form. First, the level-one set of generalized coordinates $\mathbf{u}$, the generalized velocities, is not assumed to be the straight time derivative of the position coordinates $\mathbf{q}$. Rather,

$$\mathbf{u} = \mathbf{T}(\mathbf{q})\dot{\mathbf{q}} \tag{14}$$

The old case is obtained if $\mathbf{T}(\mathbf{q}) = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix of appropriate dimension. Nonetheless, when Euler angles are used as the level-zero generalized coordinates and, for instance, the set of angular velocities ω is used for level-one generalized coordinates, this leads to a nonconstant transformation matrix $\mathbf{T}(\mathbf{q})$ (see, for instance, [17], [19]).

Second, since the approach allows for handling of Pfaffian non-holonomic constraints [17], the projection of the Lagrange multipliers onto the equations of motion in the most general case is carried out by means of a projection operator, $\mathbf{P} = \mathbf{P}(\mathbf{q})$ (in the case of exclusively holonomic constraints, $\mathbf{P} = \mathbf{C}_{\mathbf{q}}^{T}(\mathbf{q})$). With this, the set of equations governing the dynamics of the mechanical system becomes

$$\begin{array}{rcl} \mathbf{M}(\mathbf{q})\,\dot{\mathbf{u}} + \mathbf{P}(\mathbf{q})\,\lambda - \mathbf{Q}(\mathbf{u},\mathbf{q},t) & = & \mathbf{0} \quad (15a) \\ \mathbf{T}(\mathbf{q})\dot{\mathbf{q}} - \mathbf{u} & = & \mathbf{0} \quad (15b) \\ \ddot{\mathbf{C}}(\dot{\mathbf{u}},\mathbf{u},\mathbf{q},t) & = & \mathbf{0} \quad (15c) \\ \dot{\mathbf{C}}(\mathbf{u},\mathbf{q},t) & = & \mathbf{0} \quad (15d) \\ \mathbf{C}(\mathbf{q},t) & = & \mathbf{0} \quad (15e) \\ \mathbf{B}_1\mathbf{u} & = & \bar{\mathbf{u}} \quad (15f) \\ \mathbf{B}_0\mathbf{q} & = & \bar{\mathbf{q}} \quad (15g) \end{array}$$

Equations (15c) and (15d) indicate in conjunction with Eq. (15e) that the set of constraints considered are holonomic. This assumption is not necessary but is made in order to keep the presentation simpler. If non-holonomic constraints are present, Eq. (15d) would be rewritten as $\tilde{\mathbf{C}}(\mathbf{u}, \mathbf{q}, t) = \mathbf{0}$, where $\tilde{\mathbf{C}}$ includes the expressions of $\dot{\mathbf{C}}$ and the strictly non-holonomic constraints. This would carry over into Eq. (15c), which becomes $\dot{\tilde{\mathbf{C}}}(\dot{\mathbf{u}}, \mathbf{u}, \mathbf{q}, t) = \mathbf{0}$. Another simplification was made in that the set of equations (10) was not added. These three sets of equations do not bring anything qualitatively new into the picture. In the augmented approach, the introduction of the virtual variables $\bar{\mathbf{q}} \in \mathbb{R}^{z_0}$ and $\bar{\mathbf{u}} \in \mathbb{R}^{z_1}$ has already been discussed for the simple algebraic equation, and it is based on the partitioning of the generalized coordinates $\mathbf{q}$ and $\mathbf{u}$ into dependent and independent, at both position and velocity levels. The process is carried out as discussed previously in conjunction with Eqs. (2) through (4). The factorization in Eq. (4) might have to be applied twice if non-holonomic constraints are present or when $\mathbf{T}(\mathbf{q}) \neq \mathbf{I}$. Based on Eq. (2), the important outcome of the partitioning is the set of two Boolean matrices (containing only ones and zeros) $\mathbf{B}_0 \in \mathbb{R}^{z_0 \times p}$ and $\mathbf{B}_1 \in \mathbb{R}^{z_1 \times p}$. Typically $z_0 = z_1$ (which is not the case in the presence of non-holonomic constraints), and the number of degrees of freedom is defined as $z = z_0$. The key fact is the inclusion of the first and second derivatives of the constraint equations in order to make the system solvable in terms of $\bar{\mathbf{u}}$ and $\bar{\mathbf{q}}$. The nonlinear system above can be expressed as

$$\mathcal{F}(\dot{\mathbf{u}}, \dot{\mathbf{q}}, \lambda, \mathbf{u}, \mathbf{q}, t; \bar{\mathbf{u}}, \bar{\mathbf{q}}) = \mathbf{0}$$

Given $\bar{\mathbf{q}}$, Eq. (15g) provides the set of independent positions, which in turn are used in Eq. (15e) to retrieve the set of dependent positions (this action is possible because locally a condition of type Eq. (3) holds as a requisite of the partitioning process). Similarly, Eq. (15f) provides the set of independent velocities, which in turn are used in Eq. (15d) to retrieve the set of dependent velocities. Once $\mathbf{u}$ is available, $\dot{\mathbf{q}}$ is solved based on Eq. (15b), and finally $\dot{\mathbf{u}}$ and λ are solved for simultaneously, as the solution of the Eq. (15a) and (15c). Therefore, since both $\bar{\mathbf{u}}$ and $\bar{\mathbf{q}}$ are arbitrary, the following must hold:

$$\frac{\partial \mathcal{F}}{\partial \bar{\mathbf{u}}} = \mathbf{0} \qquad \frac{\partial \mathcal{F}}{\partial \bar{\mathbf{q}}} = \mathbf{0} \tag{16}$$

With the notation

$$\Upsilon_{\mathbf{u}} = \left[\mathbf{P}(\mathbf{q})\,\lambda - \mathbf{Q}(\mathbf{u}, \mathbf{q}, t)\right]_{\mathbf{u}}$$

$$\Upsilon_{\mathbf{q}} = \left[\mathbf{M}(\mathbf{q})\,\dot{\mathbf{u}} + \mathbf{P}(\mathbf{q})\,\lambda - \mathbf{Q}(\mathbf{u}, \mathbf{q}, t)\right]_{\mathbf{q}}$$

the computation of the sensitivities proceeds by using the chain rule on the conditions of Eq. (16). This yields a multiple right-side system of linear equations that provides, among several first-order (linear) sensitivities, the quantities of interest, namely, $\dot{\mathbf{u}}_{\bar{\mathbf{u}}}$, $\dot{\mathbf{u}}_{\bar{\mathbf{q}}}$, $\dot{\mathbf{q}}_{\bar{\mathbf{u}}}$, and $\dot{\mathbf{q}}_{\bar{\mathbf{q}}}$.

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}) & \mathbf{0} & \mathbf{P}(\mathbf{q}) & \Upsilon_{\mathbf{u}} & \Upsilon_{\mathbf{q}} \\ \mathbf{0} & \mathbf{T}(\mathbf{q}) & \mathbf{0} & -\mathbf{I} & [\mathbf{T}(\mathbf{q})\dot{\mathbf{q}}]_{\mathbf{q}} \\ \ddot{\mathbf{C}}_{\dot{\mathbf{u}}} & \mathbf{0} & \mathbf{0} & \ddot{\mathbf{C}}_{\mathbf{u}} & \ddot{\mathbf{C}}_{\mathbf{q}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dot{\mathbf{C}}_{\mathbf{u}} & \dot{\mathbf{C}}_{\mathbf{q}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{C}_{\mathbf{q}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_0 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}}_{\bar{\mathbf{u}}} & \dot{\mathbf{u}}_{\bar{\mathbf{q}}} \\ \dot{\mathbf{q}}_{\bar{\mathbf{u}}} & \dot{\mathbf{q}}_{\bar{\mathbf{q}}} \\ \lambda_{\bar{\mathbf{u}}} & \lambda_{\bar{\mathbf{q}}} \\ \mathbf{u}_{\bar{\mathbf{u}}} & \mathbf{u}_{\bar{\mathbf{q}}} \\ \mathbf{q}_{\bar{\mathbf{u}}} & \mathbf{q}_{\bar{\mathbf{q}}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{17}$$

In fact, this linear system provides sensitivities of both the independent and dependent quantities (accelerations, velocities) with respect to the independent coordinates. In this method, of interest are only the sensitivities of the independent quantities with respect to the independent coordinates, which are readily available and computed by using the Boolean partitioning matrices $\mathbf{B}_0$ and $\mathbf{B}_1$. For the more common case when there are no non-holonomic constraints and $\mathbf{T}(\mathbf{q}) = \mathbf{I}$, assuming that each body contributes six entries in $\mathbf{q}$, $\mathbf{u}$, $\dot{\mathbf{q}}$, and $\dot{\mathbf{u}}$, the total number of unknowns is $24n_b + m$. The total number of equations is obtained as follows: $6n_b$ equations of motion (Eq. (15a)), $6n_b$ kinematic relations (Eq. (15b)), $3m$ constraint equations (Eqs. (15c) to (15e)), and $2z$ equations given by the coordinate partition (Eqs. (15f) and (15g)). Summing up, the number of equations matches the number of unknowns. Note that this tally was done with no flexible bodies in the problem, but the conclusion carries over to the case in which $\mathbf{q}$ includes modal coordinates.

Equation (17) is a generalization of Eq. (13). The partials required to set up the sensitivity linear system of Eq. (17) are readily available because they are the typical derivatives otherwise required by the implicit numerical integration-based time-domain analysis of the system. The partitioning information is again entirely encapsulated in the Boolean matrices $\mathbf{B}_0$ and $\mathbf{B}_1$ and the old SSR process and subsequent linearization of Eq. (6a) is delegated to the linear algebra solver. The proposed method is significantly simpler to implement; it is more general in that it can handle non-holonomic constraints, as well as controls, ordinary differential equations, user-defined variables, and the like, as defined in Eq. (10); and there is no need to partition the mass or constraint Jacobian, or for that matter the generalized force vector; there is no requirement to explicitly invert any matrix (as $\mathbf{C}_{\mathbf{x}}^{-1}$ (see Eqs. (6b) and (6c)) and subsequently compute its linearization. These attractive features by far outweigh an increase in the dimension of the linear algebra problem. In terms of CPU time required for solution, the numerical experiments carried out with the new method compare favorably with the previous MSC.ADAMS method, a fact credited to the inherent sparsity associated with the augmented approach. In this context, the efficiency of the method is directly linked to the performance of the underlying solver considered for the solution of the linear system in Eq. (17). This was a sparse solver, which did not take into account however any underlying topology of the problem, if it existed (such as it would be the case in finite element analysis, for instance). After a local search, the sparse linear solver considered selects a new pivot based on its magnitude and the amount of fill-in generated by this choice. Finally, although currently not implemented, the multiple right-side system is very amenable to a multithreaded approach.

LINEARIZATION IN USER-DEFINED STATES

A general-purpose software package formulates the equations of motions associated with a constrained mechanical systems in terms of a set of internal coordinates $\mathbf{q}$ and $\mathbf{u}$; the user has no control over this predefined selection. For example, in MSC.ADAMS [13], the internal states are the Cartesian positions and 3-1-3 local Euler angles. In the case of a simple pendulum, the user might be interested in linearizing the dynamics at an operation point using the pendulum equilibrium deviation angle θ, a coordinate that the software package does not use in formulating the equations of motion. In an augmented framework, this section introduces by means of a simple example the linearization process as carried out in terms of user-defined coordinates. To this end, consider the differential equation

$$\dot{y} = f(y) \tag{18}$$

where f is nonlinear and the linearized equation is

$$\delta\dot{y} = \left[\frac{\partial f}{\partial y}\right] \delta y$$

Suppose that the user indicates by means of a function g a new coordinate for the purpose of reformulating the dynamics equation (assumed here to be given by Eq. (18)) in the new state s:

$$s = g(y)$$

Locally, the nonlinear mapping above is assumed one-to-one, which is the case as soon as the Jacobian $\mathbf{G} = [\frac{\partial g}{\partial y}]$ is nonsingular. For the linearization problem, the interest is in finding $\frac{\partial \dot{g}}{\partial s}$, or equivalently $\frac{\partial \dot{s}}{\partial s}$. Typically, the quantities $\frac{\partial g}{\partial y}$, and $\frac{\partial \dot{g}}{\partial y}$ are readily available because these derivatives with respect to the internal coordinates can be computed without any difficulty. The problem is augmented by the introduction of a virtual state $\bar{s}$:

$$\begin{aligned} \dot{y} - f(y) &= 0 \\ s - g(y) &= 0 \\ \dot{s} - \dot{g}(y, \dot{y}) &= 0 \\ s &= \bar{s} \end{aligned}$$

Using the chain rule to take the partials with respect to $\bar{s}$, and then rearranging, one obtains the sensitivity linear system:

$$\begin{bmatrix} 1 & -f_y & 0 & 0 \\ 0 & -g_y & 0 & 1 \\ -\dot{g}_{\dot{y}} & -\dot{g}_y & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{y}_{\bar{s}} \\ y_{\bar{s}} \\ \dot{s}_{\bar{s}} \\ s_{\bar{s}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{19}$$

Solving this system produces the quantities of interest, namely, $\frac{\partial \dot{s}}{\partial s} = \dot{s}_{\bar{s}}$. The observation that carries over to the general multidimensional case of differential-algebraic equations of motion is that without low-level manipulations, a new set of user coordinates can be introduced, and, using an augmented approach and exclusively internal coordinate type partials, one can compute the linearization in terms of the user-defined coordinates as the solution of a sensitivity linear system.

Selection of the Linearization State S

Assume that a set of $\hat{z}$ user-defined states $\hat{\mathbf{S}}$

$$\hat{\mathbf{S}} = \hat{\mathbf{s}}(\mathbf{q}, t) \tag{20}$$

are defined and indicated to be the user's choice of coordinates for carrying out the linearization. For a complex real-life system that contains flexible bodies, it is unreasonable to expect the user to provide a full set of states equal to the number of model degrees of freedom, which typically is unknown a priori. Consequently, in an attempt to provide a Lagrangian set of coordinates one can over- or underprescribe the set of states. To select Lagrangian coordinates, consider the $m + \hat{z}$ equations in the $p + \hat{z}$ variables $\mathbf{q}$ and $\hat{\mathbf{S}}$,

$$\mathbf{C}(\mathbf{q}, t) = \mathbf{0} \tag{21a}$$

$$\hat{\mathbf{S}} - \hat{\mathbf{s}}(\mathbf{q}) = \mathbf{0} \tag{21b}$$

The state selection algorithm proceeds by building the Jacobian matrix

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{C}(\mathbf{q},t)}{\partial \mathbf{q}} & \mathbf{0} \\ -\frac{\partial \hat{\mathbf{s}}(\mathbf{q})}{\partial \mathbf{q}} & \mathbf{I} \end{bmatrix} \tag{22}$$

The selection of the Lagrangian states $\mathbf{S} \in \mathbb{R}^z$, which contain either a subset or all of the states $\hat{\mathbf{S}}$ defined by Eq. (21b), continues with an $\mathbf{LU}$ factorization with full pivoting. Since necessarily $m < p$, at the end of the factorization a nonsingular submatrix will span the columns corresponding to partials with respect to the *dependent coordinates*. The value of z is subsequently defined as the count of remaining columns in the factored Jacobian, and these rightmost columns define the set of independent coordinates $\mathbf{S}$. Note that each of the s right-most states may belong to either $\mathbf{q}$ or $\hat{\mathbf{S}}$. On the one hand, if too many user states have been proposed, the ones that don't find their way into the set of independent states are dropped from the problem. On the other hand, if too few states are selected, some of the internal states $q_i \in \mathbf{q}$ are promoted to become Lagrangian coordinates. In retrospect, this is what happens with coordinate-partitioning SSR, when no user states are specified and the algorithm selects the Lagrangian coordinates $\mathbf{S}$ among the internal states $\mathbf{q}$.

Regardless of the number of user-defined coordinates in $\hat{\mathbf{S}}$, the algorithm favors the selection of coordinates $\mathbf{S}$ from this set. The selection is accomplished by means of a weighting strategy in which each row is assigned a weight that biases the factorization algorithm toward preferring a user-defined coordinate over an internal coordinate from $\mathbf{q}$. Moreover, in order to bias the column pivoting, the identity matrix shown in Eq. (22) can be replaced by a diagonal matrix of small values.

The General Case of the Multibody Equations of Motion

Along with the selected state $\mathbf{S} \in \mathbb{R}^z$ go a set of equations $\mathbf{S} = \mathbf{s}(\mathbf{q})$ defining these states and a set of velocity states $\mathbf{W} = \dot{\mathbf{S}}$. The augmented set of equations assumes the form

$$\begin{aligned}
\mathbf{M}(\mathbf{q})\,\dot{\mathbf{u}} + \mathbf{P}(\mathbf{q})\,\lambda - \mathbf{Q}(\mathbf{u},\mathbf{q},t) &= \mathbf{0} && \text{(23a)}\\
\mathbf{T}(\mathbf{q})\dot{\mathbf{q}} - \mathbf{u} &= \mathbf{0} && \text{(23b)}\\
\ddot{\mathbf{C}}(\dot{\mathbf{u}},\mathbf{u},\mathbf{q},t) &= \mathbf{0} && \text{(23c)}\\
\dot{\mathbf{C}}(\mathbf{u},\mathbf{q},t) &= \mathbf{0} && \text{(23d)}\\
\mathbf{C}(\mathbf{q},t) &= \mathbf{0} && \text{(23e)}\\
\mathbf{S} - \mathbf{s}(\mathbf{q}) &= \mathbf{0} && \text{(23f)}\\
\dot{\mathbf{S}} - \mathbf{W} &= \mathbf{0} && \text{(23g)}\\
\mathbf{W} - \dot{\mathbf{s}}(\mathbf{u},\mathbf{q},t) &= \mathbf{0} && \text{(23h)}\\
\dot{\mathbf{W}} - \ddot{\mathbf{s}}(\dot{\mathbf{u}},\mathbf{u},\mathbf{q},t) &= \mathbf{0} && \text{(23i)}\\
\mathbf{W} &= \bar{\mathbf{W}} && \text{(23j)}\\
\mathbf{S} &= \bar{\mathbf{S}} && \text{(23k)}
\end{aligned}$$

which is equivalently expressed as

$$\mathcal{F}(\dot{\mathbf{u}},\dot{\mathbf{q}},\dot{\mathbf{W}},\dot{\mathbf{S}},\lambda,\mathbf{u},\mathbf{q},\mathbf{W},\mathbf{S},t;\bar{\mathbf{W}},\bar{\mathbf{S}}) = \mathbf{0} \tag{24}$$

In the augmented approach, the dependent states are $\dot{\mathbf{u}}$, $\dot{\mathbf{q}}$, $\dot{\mathbf{W}}$, $\dot{\mathbf{S}}$, λ, $\mathbf{u}$, $\mathbf{q}$, $\mathbf{W}$, and $\mathbf{S}$. The Lagrangian coordinates $\bar{\mathbf{S}}$ and their time derivatives $\bar{\mathbf{W}}$ can assume arbitrary values, and locally all the dependent coordinates can be uniquely computed. Thus, based on the value of $\bar{\mathbf{S}}$, Eq. (23k) is used to compute $\mathbf{S}$. Because of the way in which $\mathbf{S}$ was selected,

$$\det \begin{vmatrix} \frac{\partial \mathbf{C}(\mathbf{q},t)}{\partial \mathbf{q}} & \mathbf{0} \\ -\frac{\partial \mathbf{s}(\mathbf{q},t)}{\partial \mathbf{q}} & \mathbf{I} \end{vmatrix} \neq 0$$

Therefore, at a certain time $t = t_{linearize}$, based on the implicit function theorem [5], $\mathbf{q}$ can be locally computed based on Eqs. (23e) and (23f). Likewise, based on the value of $\bar{\mathbf{W}}$, Eq. (23j) is used to compute $\mathbf{W}$, which is then used in conjunction with Eqs. (23h) and (23d) to compute $\mathbf{u}$. Equation (23b) provides $\dot{\mathbf{q}}$, Eqs. (23a) and (23c) are simultaneously solved for λ and $\dot{\mathbf{u}}$, and Eq. (23i) is used to compute $\dot{\mathbf{W}}$. Therefore, necessarily,

$$\frac{\partial \mathcal{F}}{\partial \bar{\mathbf{W}}} = \mathbf{0} \qquad \frac{\partial \mathcal{F}}{\partial \bar{\mathbf{S}}} = \mathbf{0} \tag{25a}$$

which leads to

$$\mathbf{A}\,\mathbf{X} = \mathbf{B} \tag{25b}$$

This is a multiple right-side linear system (see Table 1 for the expression of $\mathbf{A}$, $\mathbf{X}$, and $\mathbf{B}$), which provides for the quantities of interest, namely, $\frac{\partial \dot{\mathbf{S}}}{\partial \bar{\mathbf{S}}}$, $\frac{\partial \dot{\mathbf{S}}}{\partial \bar{\mathbf{W}}}$, $\frac{\partial \dot{\mathbf{W}}}{\partial \bar{\mathbf{S}}}$, and $\frac{\partial \dot{\mathbf{W}}}{\partial \bar{\mathbf{W}}}$. In the derivation it was assumed that the constraints were holonomic. The method easily accommodates the case of non-holonomic constraints, with the caveat that Eq. (23d) should be replaced by a new set of equations that contains $\dot{\mathbf{C}}$, plus the strictly non-holonomic constraints. Likewise, the selection of states $\mathbf{S}$ and $\mathbf{W}$ should be carried out independently, as $\mathbf{S} \in \mathbb{R}^{z_0}$, and $\mathbf{W} \in \mathbb{R}^{z_1}$. Furthermore, the traditional concept of number of degrees of freedom ceases to be well defined, as $z_1 < z_0$. These technicalities do not take away from the generality of the augmented approach, which remains a direct and simple way of conducting linearization of the nonlinear differential-algebraic equations of motion in terms of user-defined coordinates.

Table 1: $\mathbf{A}$, $\mathbf{X}$, and $\mathbf{B}$ Sensitivity Matrices

$$\mathbf{A} = \begin{bmatrix} \mathbf{M}(\mathbf{q}) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}(\mathbf{q}) & \boldsymbol{\Upsilon}_{\mathbf{u}} & \boldsymbol{\Upsilon}_{\mathbf{q}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}(\mathbf{q}) & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & [\mathbf{T}(\mathbf{q})\dot{\mathbf{q}}]_{\mathbf{q}} & \mathbf{0} & \mathbf{0} \\ \ddot{\mathbf{C}}_{\dot{\mathbf{u}}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddot{\mathbf{C}}_{\mathbf{u}} & \ddot{\mathbf{C}}_{\mathbf{q}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dot{\mathbf{C}}_{\mathbf{u}} & \dot{\mathbf{C}}_{\mathbf{q}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{C}_{\mathbf{q}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{s}_{\mathbf{q}} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\dot{\mathbf{s}}_{\mathbf{u}} & -\dot{\mathbf{s}}_{\mathbf{q}} & \mathbf{I} & \mathbf{0} \\ -\ddot{\mathbf{s}}_{\dot{\mathbf{u}}} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & -\ddot{\mathbf{s}}_{\mathbf{u}} & -\ddot{\mathbf{s}}_{\mathbf{q}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} \dot{\mathbf{u}}_{\bar{\mathbf{W}}} & \dot{\mathbf{u}}_{\bar{\mathbf{S}}} \\ \dot{\mathbf{q}}_{\bar{\mathbf{W}}} & \dot{\mathbf{q}}_{\bar{\mathbf{S}}} \\ \dot{\mathbf{W}}_{\bar{\mathbf{W}}} & \dot{\mathbf{W}}_{\bar{\mathbf{S}}} \\ \dot{\mathbf{S}}_{\bar{\mathbf{W}}} & \dot{\mathbf{S}}_{\bar{\mathbf{S}}} \\ \lambda_{\bar{\mathbf{W}}} & \lambda_{\bar{\mathbf{S}}} \\ \mathbf{u}_{\bar{\mathbf{W}}} & \mathbf{u}_{\bar{\mathbf{S}}} \\ \mathbf{q}_{\bar{\mathbf{W}}} & \mathbf{q}_{\bar{\mathbf{S}}} \\ \mathbf{W}_{\bar{\mathbf{W}}} & \mathbf{W}_{\bar{\mathbf{S}}} \\ \mathbf{S}_{\bar{\mathbf{W}}} & \mathbf{S}_{\bar{\mathbf{S}}} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

The MCK Representation

Both in control theory and in linear analysis of structures, a linear representation of the form

$$\hat{\mathbf{M}}\ddot{\mathbf{S}} + \mathbf{C}\dot{\mathbf{S}} + \mathbf{K}\mathbf{S} = \hat{\mathbf{Q}} \tag{26}$$

is preferred. In the configuration in which the linearization is carried out the formalism proposed provides for the quantities $-\hat{\mathbf{M}}^{-1}\mathbf{K}$, and $-\hat{\mathbf{M}}^{-1}\mathbf{C}$, which are computed as $\frac{\partial \dot{\mathbf{W}}}{\partial \bar{\mathbf{S}}}$, and $\frac{\partial \dot{\mathbf{W}}}{\partial \bar{\mathbf{W}}}$, respectively. According to d'Alembert's principle

$$\delta \mathbf{q}^T \left[\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{Q}(\mathbf{u}, \mathbf{q}, t)\right] = \widehat{\mathbf{Q}}(t) \tag{27a}$$

where the virtual displacement satisfied the condition

$$\mathbf{C}_{\mathbf{q}}(\mathbf{q}, t)\delta\mathbf{q} = \mathbf{0} \tag{27b}$$

The set of generalized coordinates $\mathbf{q}$ can be expressed in terms of the Lagrangian coordinates $\mathbf{S}$:

$$\mathbf{q} = \mathbf{\Psi}(\mathbf{S}) \qquad \mathbf{\Psi} : \mathbb{R}^z \rightarrow \mathbb{R}^n$$

It follows that $\mathbf{\Psi}$ is such that $\mathbf{C}(\mathbf{\Psi}(\mathbf{S}), t) = \mathbf{0}$. Starting with Eq. (27a) and expressing the virtual displacement $\delta\mathbf{q}$ and acceleration $\ddot{\mathbf{q}}$ in terms of the Lagrangian coordinates leads to

$$\delta \mathbf{S}^T \left[\hat{\mathbf{M}}\ddot{\mathbf{S}} - \hat{\mathbf{Q}}\right] = \mathbf{0} \tag{28a}$$

where

$$\begin{aligned} \hat{\mathbf{M}} \quad &= \mathbf{\Psi}_{\mathbf{S}}^T\, \mathbf{M}(\mathbf{q})\, \mathbf{\Psi}_{\mathbf{S}} && (28b) \\ \hat{\mathbf{Q}} \quad &= \mathbf{\Psi}_{\mathbf{S}}^T\, \mathbf{Q} - \mathbf{\Psi}_{\mathbf{S}}^T \mathbf{M}(\mathbf{q})(\mathbf{\Psi}_{\mathbf{S}}\dot{\mathbf{S}})_{\mathbf{S}}\dot{\mathbf{S}} && (28c) \end{aligned}$$

Thus, the matrix $\hat{\mathbf{M}}$ can be obtained as soon as $\mathbf{\Psi}_{\mathbf{S}}$ is available. Note that requiring $\mathbf{\Psi}$, that is, the explicit dependence of $\mathbf{q}$ on $\mathbf{S}$, is unnecessary and that $\mathbf{\Psi}_{\mathbf{S}} = \mathbf{q}_{\bar{\mathbf{S}}}$, that is, the quantity of interest, becomes available as a byproduct of the overall sensitivity computation in Eq. (25b). Therefore,

$$\begin{aligned} \mathbf{C} \quad &= \quad -\mathbf{q}_{\bar{\mathbf{S}}}^T\, \mathbf{M}(\mathbf{q})\, \mathbf{q}_{\bar{\mathbf{S}}}\, \dot{\mathbf{W}}_{\bar{\mathbf{W}}} && (29a) \\ \mathbf{K} \quad &= \quad -\mathbf{q}_{\bar{\mathbf{S}}}^T\, \mathbf{M}(\mathbf{q})\, \mathbf{q}_{\bar{\mathbf{S}}}\, \dot{\mathbf{W}}_{\bar{\mathbf{S}}} && (29b) \end{aligned}$$

which allows for the formulation of the problem in the desired form of Eq. (26). Note that in the simple case of the generalized coordinates partitioning approach, $\mathbf{S} \equiv \mathbf{y}$, and

$$\mathbf{\Psi}_{\mathbf{S}} = \begin{bmatrix} -\mathbf{C}_{\mathbf{x}}^{-1}\, \mathbf{C}\mathbf{y} \\ \mathbf{I} \end{bmatrix}$$

Using Eq. (28b), one obtains precisely the generalized mass matrix of Eq. (6b). Finally, when the level 1 and level 0 user-defined states are not the same, it is easy to show that the mass matrix becomes (using the notation of Table 1)

$$\hat{\mathbf{M}} = \mathbf{u}_{\bar{\mathbf{W}}}^T \mathbf{M}(\mathbf{q})\mathbf{u}_{\bar{\mathbf{W}}} \tag{30}$$

which is a generalization of the mass matrix of Eq. (28b).

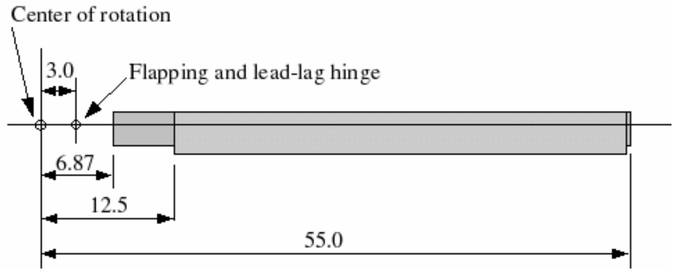


Figure 1: A non uniform helicopter blade.

NUMERICAL EXPERIMENTS

The proposed linearization algorithm was validated in three ways. First, to test the cases when no user-defined coordinates were present and the linearization was carried out exclusively in terms of internal coordinates, a complete suite of examples ranging from simple to complex models containing several hundred states (including flexible bodies) was considered, and results were compared against the MSC.ADAMS/Solver(F77), a FORTRAN-based solver in which the solution is based on SSR followed by linearization of Eq. (6) as described in [20]. Second, a small suite of simple mechanical systems was analyzed by hand and by a symbolic manipulator, and linearization matrices were computed explicitly. This approach was taken to test linearizations carried out in terms of unusual choices of user-defined coordinates. Third, a numerical differentiation tool was implemented to linearize in terms of user-defined coordinates larger systems when the results could not be validated by using either symbolic tools or the MSC.ADAMS/Solver(F77) (which cannot accommodate linearization in terms of user-defined coordinates). In what follows, four models are presented, and the proposed method is compared with the method available in the MSC.ADAMS/Solver(F77) or validated against both experimental and simulation results produced by MSC.NASTRAN [14].

A Nonuniform Helicopter Blade

Figure 1 shows a sketch of a nonuniform helicopter blade tested by NASA. Experimental results along with numerical results obtained by using the FEA code MSC.NASTRAN are presented in [22]. The blade corresponds to an articulated helicopter blade with a hinge at x=3.0 in. A rotational spring is located at the hinge and oriented to control the lead lag motion. The hinge provides no stiffness to prevent flapping. The finite-element analysis with MSC.NASTRAN was done in two steps. First, the model was subjected to a nonlinear pseudo static analysis by applying centripetal forces to simulate the steady-state rotational motion. The instantaneous stiffness matrix and mass matrix were saved when the analysis was finished. Second, a reload analysis was performed, followed by an eigensolution. The second step required a DMAP modification to include additional centrifugal softening effects and Coriolis terms into the damping matrix.

The linearization algorithm proposed requires one single run to generates results. However, the MSC.ADAMS model creation required to export 20 subsections of the MSC.NASTRAN model which were later imported into MSC.ADAMS as flexible-body objects. The complete MSC.ADAMS process was as follows:

1. Divide the original finite-element blade model into 20 sections. This process required refining the grid mesh in some parts.

2. Export the 20 sections by using the mnf (modal neutral format) utility. Care was taken to always include flexural, torsional, and elongation modes.

3. Import all of the 20 mnf files to create corresponding flexible-bodies.

4. Stitch all of the flexible-bodies, using fixed joints in order to reconstruct the blade.

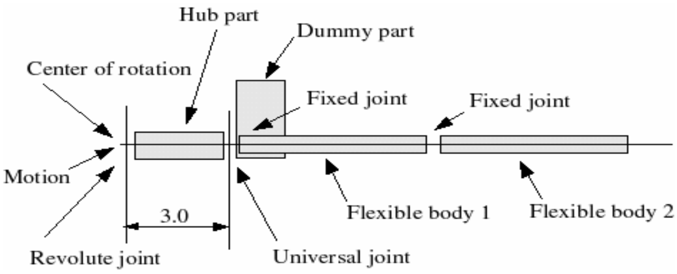


Figure 2: MSC.ADAMS model of the non uniform helicopter blade.

Table 2: Results for rotor speed of 0 rpm.

Mode	Experimental	MSC.NASTRAN	MSC.ADAMS(C++)
1^{st} flap	10.7	11.53	11.53
2^{nd} flap	32.6	36.38	36.36
1^{st} chord	41.0	42.44	42.41
3^{rd} flap	67.8	76.80	76.77
1^{st} torsion	110.3	102.05	102.05

5. Attach dummy part to the first flexible-body at the hinge location.
6. Use hub part (body) to connect the hinge to the center of rotation. The hub part is driven by a rotational motion.
7. Model lead-lag/flapping hinge by using a universal joint connected to the above dummy part and the hub part.
8. Use scalar force element to simulate the rotational spring located at the hinge.

Figure 2 shows a sketch of the resulting MSC.ADAMS model. Each of the MSC.ADAMS runs was done by first setting the blade at the operating speed, by using a motion set to a *step* function to gradually spin the blade until it reached the desired rotational speed. All dynamic simulations were done with the MSC.ADAMS/Solver(C++) using the HHT integrator [15] set to default values. Once the blade was at the required steady-state speed, an eigensolution was computed by using relative coordinates with respect to a reference frame. This process was repeated for the speeds 0, 150, 250, 350, 450, 550, and 660 rpm. Very good agreement between the two codes was found for all rotational speeds. Tables 2 and 3 show the results in Hz for 0 rpm and 660 rpm. MSC.ADAMS/Solver(C++) results also show rigid body frequencies that were verified manually by using a rigid body model of the blade found in the literature [10]. Again, good agreement was found. Figure 3 shows the second flapping mode when the blade is at rest. For this model, on a Pentium4 Linux machine, the reference solution was found in 9.37 seconds, while the new approach produced the solution in 7.61 seconds.

Table 3: Results for rotor speed of 660 rpm.

Mode	Experimental	MSC.NASTRAN	MSC.ADAMS(C++)
1^{st} flap	28.80	29.62	29.56
2^{nd} flap	51.75	55.69	55.51
1^{st} chord	49.10	49.29	49.21
3^{rd} flap	91.50	96.52	96.45
1^{st} torsion	115.00	102.53	102.37

Washing Machine

The washing machine model shown schematically in Fig. 4 contains 15 rigid bodies, five revolute joints, four spherical joints, four translational joints, four universal joints, and one prescribed motion. It involves ten degrees of freedom, with a total number of 90 coordinates describing the system. Linearization followed by eigenvalue analysis was performed first with the MSC.ADAMS/Solver(F77), which uses for linearization a combination of standard numerical differentiation and matrix manipulations, and then with the proposed method. No user-defined coordinates were used, in order to have matching states in both runs. Table 4 shows some of the resulting eigenvalues (17 of them). All units are in `hertz` and negligible differences (less then 0.5%) were observed in the complete range of results. Note that, in this and the following tables, for each pair of eigenvalues only the positive value of the imaginary part is presented; the values for $\Re$ (the real part component) and $\Im$ (the imaginary part component) provided in the tables are used to generated the complex conjugate of each listed eigenvalue. For this simple model, on a Pentium4 Linux machine, the reference solution was found in 0.16 seconds, while the new approach produced the solution in 0.37 seconds. This was the only model where the numerical experiments revealed a slow down of the simulation. This is attributed to the initialization overhead associated with the proposed method, which becomes irrelevant as soon as more complex models (see NASA blade, and All Terrain Vehicle) are considered.

Table 4: Washer model results.

Undmp. Freq.	Damping Ratio	$\Re$	$\Im$
7.316E-3	1.000E0	-7.316E-3	0.000E0
7.404E-3	1.000E0	-7.404E-3	0.000E0
8.440E-3	1.000E0	-8.440E-3	0.000E0
8.508E-3	1.000E0	-8.508E-3	0.000E0
3.216E-1	1.000E0	-3.216E-1	0.000E0
3.447E-1	1.000E0	-3.447E-1	0.000E0
3.449E-1	1.000E0	-3.449E-1	0.000E0
1.395E+1	1.000E0	-1.395E01	0.000E0
1.398E+1	1.000E0	-1.398E01	0.000E0
1.439E+1	1.000E0	-1.439E01	0.000E0
1.702E+1	1.000E0	-1.702E01	0.000E0
2.706E+1	1.000E0	-2.706E01	0.000E0
1.020E+2	1.000E0	-1.020E02	0.000E0
1.323E+2	1.000E0	-1.323E02	0.000E0
7.233E-1	3.744E-2	-2.708E-2	7.228E-1
7.237E-1	3.753E-2	-2.716E-2	7.232E-1
2.712E0	1.021E-1	-2.771E-1	2.698E0

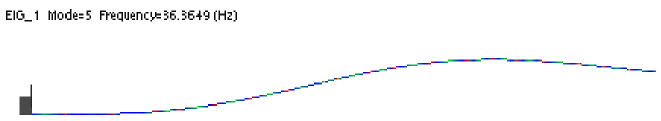


Figure 3: Second flapping mode for 0 rpm.

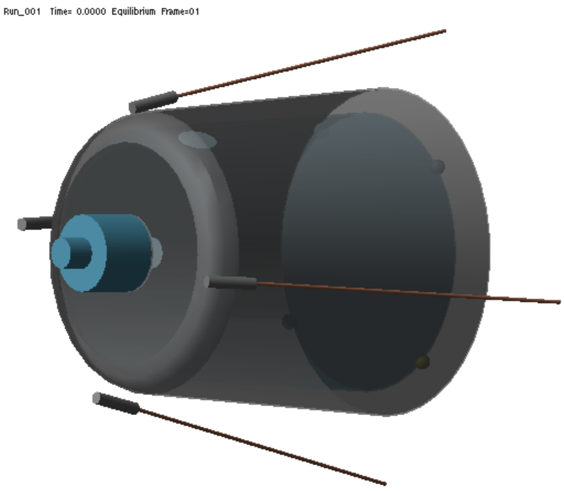


Figure 4: A washing machine example.

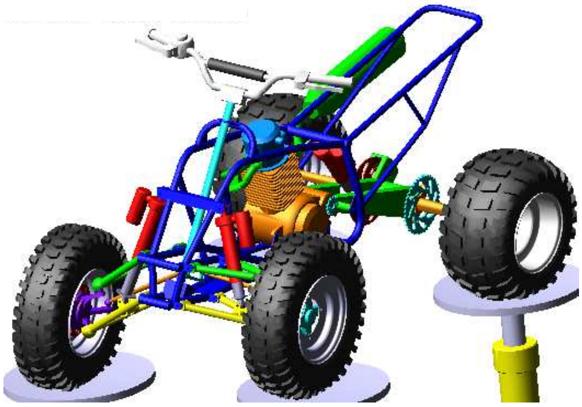


Figure 5: All terrain vehicle (ATV).

All Terrain Vehicle (ATV)

The ATV model presented in Fig. 5 has 28 rigid bodies, one flexible body, one cylindrical joint, three revolute joints, 21 spherical joints, seven translational joints, four universal joints, six fixed joints, one inline primitive, five prescribed motions, 132 degrees of freedom, and a total of 314 coordinates describing the system (140 modal coordinates were used to model the flexible body).

While the reference MSC.ADAMS/Solver(F77) and proposed MSC.ADAMS/Solver(C++) methods almost always produce similar results, this is not always true. The results in Table 5 agree well, with the exception of two sets of eigenvalues. The source of these differences is a series of simplifying assumptions embedded in the previous MSC.ADAMS solver and, to a lesser degree, the fact that the linearization was performed at the end of a dynamic simulation. The dynamic simulation was carried out with the legacy MSC.ADAMS/Solver(F77) in one case and with the new MSC.ADAMS/Solver(C++) in the other case. Given the complexity of the model, the system configuration at the end of the dynamic analysis is slightly different in the two solvers, and this contributes to the difference between the two sets of eigenvalues. Nonetheless, in the high-frequency range good agreement is observed between the reference (F77) and new (C++) implementations. For this model, on a Pentium4 Linux machine, the reference solution was found in 4.78 seconds, while the new approach produced the solution in 3.76 seconds.

Table 5: ATV eigensolution for selected values. Proposed method results and % error.

$\Re$	$\Im$	ERR-$\Re$[%]	ERR-$\Im$[%]
-1.426E0	5.145E1	17.80	1.13
-1.244E0	6.331E1	21.98	2.78
-4.272E1	3.966E2	0.20	0.00
-4.915E1	4.236E2	1.11	0.80
-8.835E2	1.083E3	0.86	0.04
-6.279E3	2.265E3	0.08	0.09
-6.279E3	2.266E3	0.00	0.06
-3.827E3	3.234E3	0.05	0.15
-4.338E3	4.221E3	0.04	0.04
-8.848E3	5.021E3	0.00	0.06

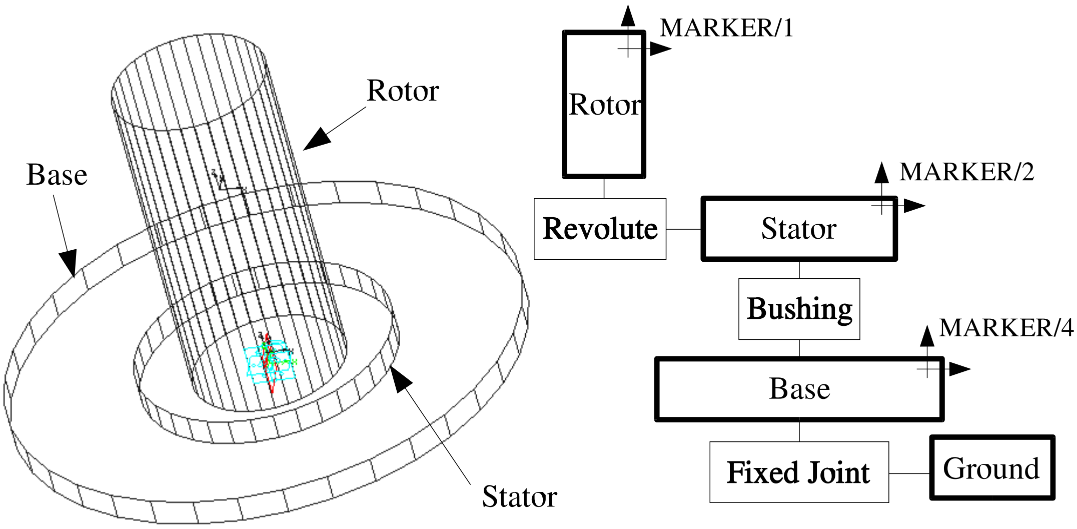
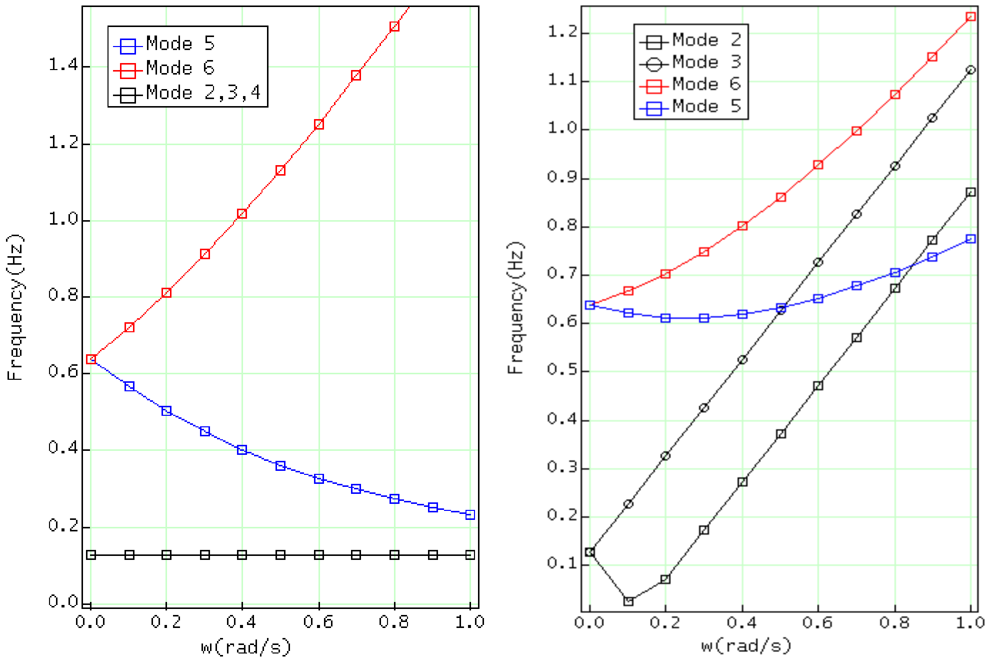


Figure 6: A Rotor-stator model.

Rotor-stator Model

Figure (6) shows a rotor-stator model for which a linearization of the equations of motion was carried out, followed by an eignesolution computation. The rotor is connected to the stator by a revolute joint with no friction and no compliance elements between them. The stator is connected to the base by a bushing element, and the base is fixed to the ground. The physical properties of the rotor are $m = 1300$ kg, $I_x = I_y = 500$ kg m^2, and $I_z = 1000$ kg m^2; and the center of mass is located at 0.75 m above the origin. The stator properties are $m = 200$ kg, $I_x = I_y = 50$ kg m^2, and $I_z = 10$ kg m^2. The bushing has translational constant $K = 1000$ N/m and $C = 1$ Ns/m in all directions; the rotational constants about the x and y axes are $K_t = 10000$ N m and $C_t = 10$ N m s, and the rotational constants about the z axis are $K_t = 100000$ N m and $C_t = 100$ N m s. The model has seven degrees of freedom and was linearized for different values of the initial angular speed of the rotor. The angular speed of the rotor has a big influence on the eigenvalues of the system. Moreover, given that the eigenvalues depend on the choice of coordinates, two reference frames were selected to first linearize and then compute the eigensolutions.

The first choice was MARKER/4 (global reference) located on the base element. Results are shown in Fig. (7)(a) for initial angular speeds of 0 Hz to 1 Hz. Notice that the plot shows only five eigenvalues of interest. Modes 5 and 6 correspond to the backward and forward rocking modes, respectively, while modes 2, 3,



(a) (b)

Figure 7: Eigenvalues for speeds 0 Hz to 1 Hz. Global (a) and local (b) reference frames.

and 4 correspond to the translational modes. One mode has zero frequency (because of the free revolute joint), and another frequency corresponds to the rotation about a vertical axis (not shown for clarity). The second choice was MARKER/1 (local reference), located on the rotor. Figure (7)(b) shows the corresponding results. Mode 5 corresponds to the forward rocking mode, mode 6 corresponds to the backward rocking mode, and modes 2 and 3 correspond to the horizontal translational modes.

A numerical experiment was set up to verify the results, as shown in Fig. (8). The base element is now connected to the ground by a motion constraint that imposes a random vibration. A second motion constraint was added to the revolute joint to speed the rotor with an angular speed of $f(t) = 0.0005(2\pi t)$. Also, a small mass (0.001 kg) was attached to the rotor in order to create an eccentric excitation. A dynamic simulation was undertaken from $t = 0$ to $t = 2500$ (the final prescribed speed of the rotor was 1.25 Hz), and the signals shown in Fig. 8 were collected and processed with an FFT filter. Figure (9)(a) shows the frequency content of signal DX(2,4) (the distance between markers 2 and 4). Notice the excellent match with the global computation shown in Fig. (7)(a).

Figure (9)(b) shows the frequency content of signal AY(4,1) (the small angle rotation of marker 1 about marker 4). Notice the excellent match with the frequencies shown in Fig. (7)(b). The FFT plot of signal AY(4,1) highlights the rocking frequencies; to highlight the translational frequencies, the FFT plot of signal DX(4,1,1) (the distance between markers 1 and 4, measured in the reference frame associated with marker 1), shown in Fig. (9)(c) was also generated. This numerical experiment validates the proposed linearization implemented in MSC.ADAMS/Solver(C++) and shows that the eigenvalues of a nonlinear system depend on the choice of coordinates used to linearized the system.

CONCLUSIONS

This paper presents a practical approach for computing at an operating point the linearization of the index 3 DAEs associated with the time evolution of a constrained mechanical system. The method proposed is equivalent to a state-space reduction followed by linearization of the resulting set of second-order state space ordinary differential

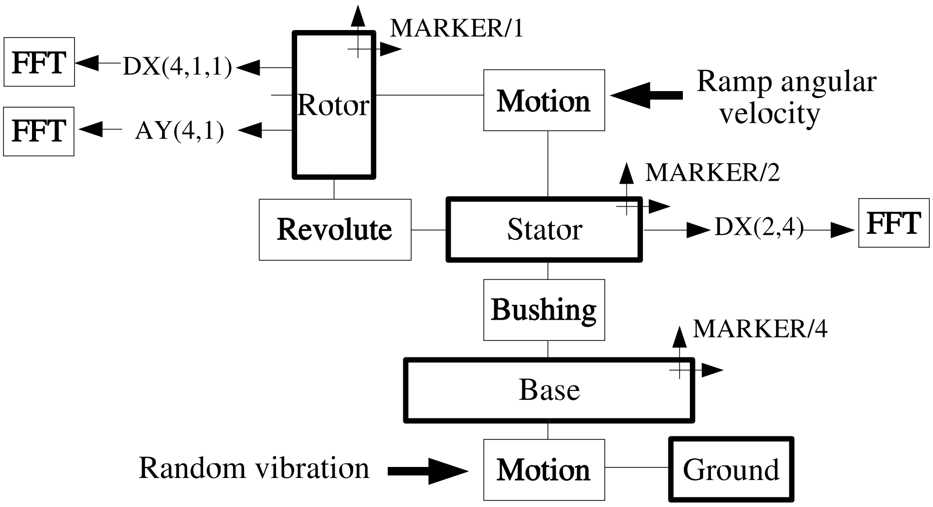


Figure 8: Rotor-stator model setup for numerical experiment.

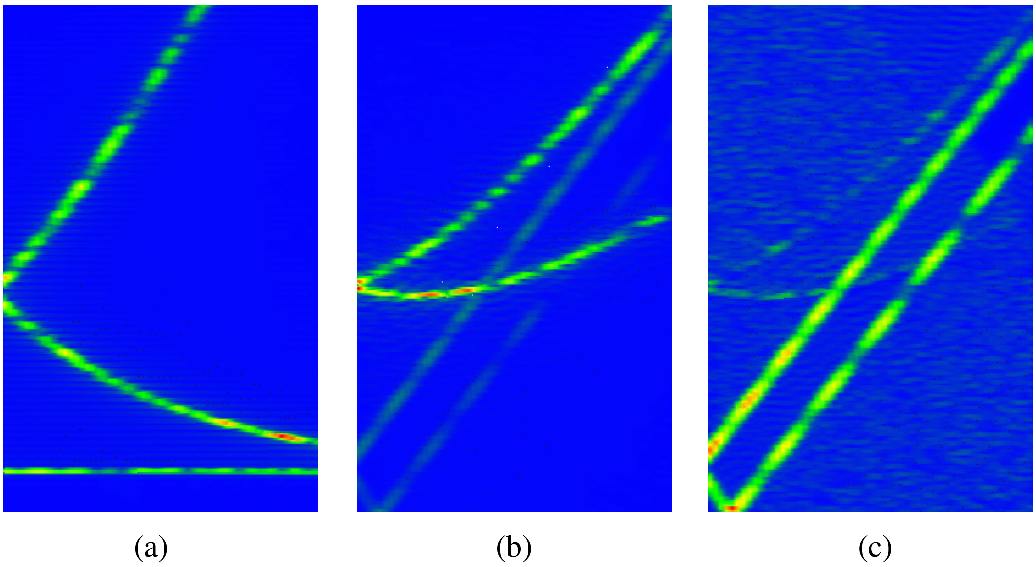


Figure 9: FFT plots of signals (a) DX(2,4), (b) AY(4,1), and (c) DX(4,1,1).

equations of motion. However, the approach does not explicitly reduce the problem to second-order differential form; instead, it augments the set of equations of motion and computes the quantities of interest as the solution of a multiple right-side sensitivity linear system. The proposed method is simple and versatile; it handles user-defined differential equations resulting from controls, arbitrary user-defined holonomic/non-holonomic constraints, and arbitrary user forces (including contact and friction). A second attractive attribute of the method is its ability to allow the user to define a specific set of coordinates (user states) in which the linearization will be carried out. If the user states specified are in deficit, the method completes this set with internal coordinates. If specified in excess, an appropriate number of user states are dropped. A third advantage is that the method is straightforward to implement. Four numerical experiments with real-life models validate the accuracy of the proposed method. The method has been implemented and released in the 2005 version of MSC.ADAMS/Solver(C++) simulation package.

ACKNOWLEDGMENTS

The authors thank Dr. Koichi Ohtomi of Toshiba Corporation for providing the stator-rotor model, Anthony Sajdak of MSC.Software for providing the ATV model, and Michael Collingridge of MSC.Software for his help with MSC.NASTRAN to prepare the NASA blade example. The first author was supported by the U.S. Department of Energy under Contract W-31-109-ENG-38.

References

[1] K.S. Anderson and S. Duan. Highly parallelizable low order dynamics algorithm for complex multi-rigid-body systems. *AIAA Journal of Guidance, Control and Dynamics*, 23(2):355–364, 2000.

[2] K. E. Atkinson. *An Introduction to Numerical Analysis*. John Wiley & Sons Inc., New York, second edition, 1989.

[3] J. Baumgarte. Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 1:1–16, 1972.

[4] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. North-Holland, New York, 1989.

[5] L. J. Corwin and R. H. Szczarba. *Multivariable Calculus*. Marcel Dekker, New York, 1982.

[6] J. Garcia de Jalon and E. Bayo. *Kinematic and Dynamic Simulation of Multibody Systems. The Real-Time Challenge*. Springer-Verlag, Berlin, 11994.

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

[8] C. W. Gear, G. Gupta, and B. Leimkuhler. Automatic integration of the eulerlagrange equations with constraints. *J. Comp. Appl. Math.*, 12:77–90, 1985.

[9] E. J. Haug. *Computer-Aided Kinematics and Dynamics of Mechanical Systems*. Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[10] W. Johnson. *Helicopter Theory*. Dover Publications, Inc., New York, 1980.

[11] S. S. Kim and M. J. Vanderploeg. QR decomposition for state space representation of constrained mechanical dynamic systems. *ASME Journal of Mechanism, Transmission, and Automation in Design*, 108:183–188, 1986.

[12] C. D. Liang and G. M. Lance. A differentiable null-space method for constrained dynamic analysis. *ASME Journal of Mechanism, Transmission, and Automation in Design*, 109:405–410, 1987.

[13] MSCsoftware. ADAMS User Manual. Also available online at http://www.mscsoftware.com, 2005.

[14] MSC.Software. Nastran User Manual. Also available online at http://www.mscsoftware.com, 2005.

[15] D. Negrut, R. Rampalli, G. Ottarsson, and A. Sajdak. On the use of the HHT method in the context of index 3 Differential Algebraic Equations of Multibody Dynamics. *International Journal for Numerical Methods in Engineering*, submitted, 2005.

[16] N. Orlandea, M. A. Chace, and D. A. Calahan. A sparsity-oriented approach to the dynamic analysis and design of mechanical systems – part I. *Transactions of the ASME Journal of Engineering for Industry*, pages 773–779, 1977.

[17] L. A. Pars. *A Treatise on Analytical Dynamics*. John Wiley & Sons, New York, 1965.

[18] F. A. Potra. Implementation of linear multistep methods for solving constrained equations of motion. *SIAM. Numer. Anal.*, 30(3), 1993.

[19] A. A. Shabana. *Dynamics of Multibody Systems*. Cambridge University Press, second edition, 1998.

[20] V. N. Sohoni and J. Whitesell. Automatic linearization of constrained dynamical models. *ASME Journal of Mechanisms, Transmissions and Automation in Design*, 108(8):300–304, 1986.

[21] R. A. Wehage and E. J. Haug. Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems. *J. Mech. Design*, 104:247–255, 1982.

[22] W.K. Wilkie, P.H. Mirick, and C.W. Langston. Rotating shake test and modal analysis of a model helicopter rotor blade. Computing Science Technical Report NASA Technical Memorandum 4760, Vehicle Technology Center, U.S. Army Research Laboratory, Langley Research Center, Hampton, Virginia, 1997.