Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

Symbolic Equation of Motion and Linear Algebra Models for High-Speed Ground Vehicle Simulations.

By: James. D. Turner, Ph.D., NADS and Simulation Center, 2401 Oakdale Blvd., Iowa City, Iowa, 52242.

Abstract. Synthetic environment modeling for human-centered research requires a high-level of model fidelity to support the illusion of a real-world experience for the simulator operator. This requirement is met by developing mathematical models that capture the critical dynamical behaviors for the modeled ground vehicles. This paper presents a physics-based modeling approach that makes extensive use of computer-aided symbol manipulation capabilities for generating high performance simulation software. A general-purpose Lagrangian method is presented, that exploits innovative sparse partial derivative algorithms for dramatically accelerating the generation of the equations of motion. A major benefit of the Lagrangian method is that all topology-based constraints are eliminated from the problem formulation. Mathematical models are provided for analyzing user-specified time-varying and closed-loop topological constraints. Symbolic methods and data structures are presented for the equations of motion, linear algebra algorithms, and Fortran subroutine generation. The object-oriented symbolic environment consists of: (1) A general-purpose model building tool (position, velocity, orientation, and angular velocity models), (2) sparse partial derivative algorithms, (3) generalized force algorithms, (4) kinetic energy partial derivative algorithms, and (5) sparse application-specific LDL' decomposition symbolic linear algebra algorithms. A script-based tool is presented that builds on the syntax of the object-oriented list-based computer-aided mathematics Macsyma program. Macsyma provides the core capabilities for symbol manipulation, differentiation, vector algebra, and highly optimized Fortran generation. Example applications will be presented for 10-body HMMWV model.

INTRODUCTION

Synthetic environment modeling for human-centered research requires high-fidelity physics-based models for supporting the illusion of a real-world experience for the simulator operator. Expanding needs for high-fidelity demand that vehicle modelers include computationally intense models for vehicle subsystems and complex physics-based tire-soil interactions. These applications, and others, place a premium on minimizing the computational effort devoted to handling the vehicle dynamics part of the synthetic environment computational load.

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

Statement Of The Problem

This paper is concerned with addressing the vehicle dynamics part of the synthetic environment-modeling problem. Traditional multibody modeling approaches have been developed with the idea of supporting general-purpose simulation capabilities. This approach has produced successful commercial multibody products such as DADS, DISCOS, ADAMS, and many other powerful software products. Originally these tools were developed to avoid the time-consuming error-prone process of developing an application-specific modeling tool that must be exhaustively validated and verified. This goal guided the advanced multibody development community during the 1980's and 1990's. Of course, these developers were aware of symbolic tools, but the general consensus among software developers at the time was that supercomputers were required for handling all but academic-scale problems. In the past few years, however, the performance of PC's has increased to the level that application-specific symbolic solution methods are now practical for analysts with access to PC and WorkStation development systems. Symbolic solutions combine the benefits of validated and verified simulation tools with the added advantage of high-performance that can be achieved through the development of an application-specific model. Other symbolic multibody tools exist; however, they do not take advantage of the matrix structure and sparse partial derivative algorithms available for Lagrange's method.

Computational Issues With Topology-Based Constraints

Computationally, as more bodies are modeled in a multibody system, the number of topology-based constraints increases very rapidly, leading to expensive solution algorithms. For example, assuming that one is modeling a simplified ground vehicle

consisting of: a base body with six degrees-of-freedom (DOF) and no constraints, four wheel assembles with eight DOF and 20 constraints, and a steering mechanism with one DOF and 5 constraints. This notional system has a total of 15 DOF and 45 constraints, requiring a solution for 15 acceleration DOF and 45 Lagrange multipliers. For this simple application the number of topology-based constraints is 3x larger than the number of DOF.

Alternatively, the proposed symbolic multibody algorithm completely eliminates the 45 constraints from ever entering the mathematical model. In a conventional multibody algorithm, the requirement for explicitly handling the 45 Lagrange multipliers has a dramatic impact on the computational efficiency of commercial codes such as DADS, DISCOS, and ADAMS. Recursive problem formulations partially address this problem, however, by reformulating the solution for the Lagrange Multipliers to sequentially solve a series of low-order matrix systems. Nevertheless, topology-based Lagrange multipliers remain an essential part of the computational burden. The proposed symbolic tool accelerates the simulation of mechanical systems by completely eliminating the topology-based constraints from the problem model.

A Lagrange multiplier capability, however, is provided for computing user-specified time-varying constraints for evaluating constraint loads, for answering engineering design questions. Capabilities are also provided for handling closed-loop topological structures. As shown in what follows, the full-scale implementation of the classical Lagrange dynamics method has a significant impact on the size and structure of the resulting equations, and the models computational efficiency for simulating the response of linked mechanical systems.

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

Paper Contribution

To this end, this paper presents an object-oriented symbolic multi-body modeling environment that uses a script-based user-defined input for building: (1) model data, (2) processing the model geometry and kinematics, (3) the equations of motion using Lagrange's method, (4) the generalized forces, (5) user-specified time-varying constraint conditions, (6) mass matrix, (7) a sparse symbolic LDL' linear equation for the system accelerations, and (8) transforming all math models to optimized Fortran software for generating the system dynamic response. The goal of the symbolic modeling capability is to create application-specific models that eliminates all: (1) topology-based Lagrange multipliers, (2) mathematical operations leading to zero results, (3) inefficient algorithm structures, (4) logic blocks, (5) do-loops, and (6) all other computer language facilities that act to slow down the simulation performance. An object-oriented list-based computer language is used to develop the software.

The paper is presented in seven sections. The mathematical model is presented in the first section. This section presents the quasi-coordinate transformations required for building the equations of motion using Lagrange's equations. The second section presents the general form for the constraint equations and the transformations required to map general hinge constraint conditions to generalized coordinate form. The system-level equations of motion are presented in section three. The partial derivative models for evaluating the Lagrangian are presented in the forth section. The generalized force is presented in the fifth section. Results of a 10-body HMMWV are presented in the sixth section. Conclusions are presented in section seven.

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

QUASI-COORDINATE FORMULATION FOR EQUATIONS OF MOTION

The mathematical modeling technique used in this paper is based on the work of Joseph-Louis Lagrange (1736-1813), who published his analytical dynamics method in 1788 in *Mécanique Analytique* (*1*). In this he lays down the law of virtual work, and from that one fundamental principle, by the aid of the calculus of variations, deduces the whole of mechanics, both of solids and fluids. His method remains attractive today for two reasons. First, given the system kinetic energy, the process of generating the equations of motion (EOM) is reduced to performing mechanical differentiation of a scalar function. Second, his method provides an automated way to eliminate topology-based constraint forces and torques. Unlike the earlier force-based methods of Newton and Euler, Lagrange's method focuses attention on the physical displacements of the bodies.

The recipe for constructing the EOM by Lagrange's method consists of four steps. For all bodies in the model one needs to build: the position, velocity, orientation, and angular velocity vectors; the system kinetic energy; the EOM by differentiating the kinetic energy; and the generalized forces. The only weakness of the method is that hand calculations become cumbersome, tedious, and error-prone when many bodies and DOF are involved. This paper presents a symbolic environment for handling the transformational details required generating the EOM and building high-performance Fortran simulation products.

Lagrange's Method

Lagrange's equation is obtained by applying the calculus of variations for Hamilton's principle (*2*,*3*):

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

$$\delta\int_{t_1}^{t_2}(T-V)\,dt=0;\ \delta(t_1)=0;\ \delta(t_2)=0$$

where **q** denotes the vector of generalized coordinates, the time interval end points are assumed to be fixed, and δ denotes the usual variational symbol. Application of Hamilton's principle leads to the lagrangian equations:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right)-\frac{\partial L}{\partial q_i}=Q_i\ ;\quad i=1,\cdots,n \tag{1}$$

where L = T – V denotes the scalar Lagrangian function, $T=T(q,\dot{q})$ denotes the kinetic energy, $V=V(q)$ denotes the potential energy, and from the virtual work $\delta W=\sum_{i=1}^{N}Q_i\delta q_i$ the generalized force is defined by Q_i. Equation (1) is valid for independent generalized coordinates. Many problems, however, are more naturally analyzed by introducing dependent sets of generalized coordinates. The dependent coordinates are referred as quasi-coordinates (*2,3,4,5,6,7*). In deed, for ground-vehicle simulations, it is very convenient to model the chassis with body-fixed quasi-coordinates.

Quasi-Coordinate Transformation Equations

A quasi-coordinate version of Eq. (1), for the base body rotation and translation, is obtained by introducing a change of variables of the form:

$$\xi_s=\psi_{1s}\dot{q}_1+\psi_{2s}\dot{q}_2+\cdots+\psi_{ns}\dot{q}_n;\quad s=1,2,\cdots,n \tag{2}$$

where ψ_{rs} is a known function of the independent generalized coordinates, **q**. The first step consists of generating the following vector versions of Eq. (1) for the base body rotation and translation:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \vec{\alpha}}\right)-\frac{\partial L}{\partial \vec{\theta}}=Q_{\vec{\theta}} \tag{3}$$

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

and

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \vec{v}^{N}}\right)-\frac{\partial L}{\partial \vec{r}^{N}}=Q_{\vec{r}^{N}} \tag{4}$$

where $\vec{\theta}$ denotes an array of Euler angles, $\vec{\alpha}$ denotes an array of Euler angle rates, $\vec{r}^{N}$ denotes the inertial position vector, and $\vec{v}^{N}$ denotes the inertial velocity vector. Equations (3) and (4) are evaluated by introducing a body kinetic energy of the form:

$$T=\frac{1}{2}\left\{\begin{bmatrix}\Gamma & 0\\ 0 & C_{NB}\end{bmatrix}\begin{bmatrix}\vec{\alpha}\\ \vec{v}^{N}\end{bmatrix}\right\}^{t}\begin{bmatrix}\vec{\vec{J}} & \tilde{S}\\ \tilde{S}^{t} & M\end{bmatrix}\begin{bmatrix}\Gamma & 0\\ 0 & C_{NB}\end{bmatrix}\begin{bmatrix}\vec{\alpha}\\ \vec{v}^{N}\end{bmatrix} \tag{5}$$

where $\vec{\vec{J}}$ denotes the body inertia tensor, $\tilde{S}$ denotes center of mass vector expressed the form of a skew symmetric matrix, M denotes the body mass, $\Gamma\left(\vec{\theta}\right)$ denotes a non-orthogonal transformation matrix that relates Euler angle rates and body components of angular velocity (i.e., $\vec{\omega}=\Gamma\vec{\alpha}$), and $C_{NB}\left(\vec{\theta}\right)$[1] denotes the direction cosine matrix that maps inertial vector components into body vector components (i.e., $\vec{v}^{B}=C_{NB}\vec{v}^{N}$). Introducing a quasi-coordinate change of variables in Eq. (5) leads to:

$$T=\frac{1}{2}\begin{bmatrix}\vec{\omega}\\ \vec{v}^{B}\end{bmatrix}^{t}\begin{bmatrix}\vec{\vec{J}} & \tilde{S}\\ \tilde{S}^{t} & M\end{bmatrix}\begin{bmatrix}\vec{\omega}\\ \vec{v}^{B}\end{bmatrix}$$

where only the dynamic reference frame vector components appear.

Rotational Equation Transformation

Equation (3) is transformed by observing that Eq. (4) is a function of Euler angles and Euler angle rates $\left(\vec{\theta},\dot{\vec{\theta}}\right)$. After the transformation, the new variables are Euler angles and body components of angular velocity $\left(\vec{\theta},\vec{\omega}\right)$. Since $\Gamma=\Gamma\left(\vec{\theta}\right)$ and $C_{NB}=C_{NB}\left(\vec{\theta}\right)$ in

Eq. (5), it follows that the transformation must consider both the orientation and velocity variables. For simplicity, we assume that any potential terms have been have been accounted for in the generalized forces, so that only kinetic energy partials must be considered. Equations are required for $\frac{\partial T}{\partial \theta_i}$, $\frac{\partial T}{\partial \dot{\theta}_i}$, and $\frac{d}{dt}\frac{\partial T}{\partial \dot{\theta}_i}$. The chain rule of calculus is used to complete the transformation. To this end, the partial derivative for $\frac{\partial T}{\partial \theta_i}$ follows as:

$$\frac{\partial T}{\partial \theta_i} = \frac{\partial \vec{\omega}^t}{\partial \theta_i}\frac{\partial T}{\partial \vec{\omega}} + \frac{\partial \left(\vec{v}^B\right)^t}{\partial \theta_i}\frac{\partial T}{\partial \vec{v}^B}$$

$$= \vec{\omega}^t \Gamma^{-t}\left(\frac{\partial \Gamma}{\partial \theta_i}\right)^t \frac{\partial T}{\partial \vec{\omega}} + \left(\vec{v}^B\right)^t C_{NB}\left(\frac{\partial C_{NB}}{\partial \theta_i}\right)^t \frac{\partial T}{\partial \vec{v}^B} \tag{6}$$

Similarly, the partial derivative for $\frac{\partial T}{\partial \dot{\theta}_i}$ is given by:

$$\frac{\partial T}{\partial \dot{\theta}_i} = \frac{\partial \vec{\omega}^t}{\partial \dot{\theta}_i}\frac{\partial T}{\partial \vec{\omega}} = \Gamma_{ji}\frac{\partial T}{\partial \omega_j} \tag{7}$$

where j = 1, 2, 3 denotes an implied summation. The time derivative of Eq. (7) is

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{\theta}_i} = \dot{\Gamma}_{ji}\frac{\partial T}{\partial \omega_j} + \Gamma_{ji}\frac{d}{dt}\frac{\partial T}{\partial \omega_j} \tag{8}$$

where the time derivative of i-j[th] element of Γ follows as:

$$\dot{\Gamma}_{ji} = \frac{\partial \Gamma_{ji}}{\partial \vec{\theta}} \cdot \dot{\vec{\theta}} = \frac{\partial \Gamma_{ji}}{\partial \vec{\theta}} \cdot \Gamma^{-1}\vec{\omega} \tag{9}$$

(1 x 3) (3 x 1)

[1] C_{NB} denotes the direction cosine matrix defined by C [From][To], where N is the inertial frame and B denotes the body frame and $C_{BN} = (C_{NB})^T$.

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

where the Euler angle rates have been replaced with body angular velocities. Introducing Eqs. **(6)**, (8), and (9) into Eq.**(3)** , and multiplying through by Γ^{-t}, leads to

$$\frac{d}{dt}\frac{\partial T}{\partial \bar{\omega}} + \Gamma^{-t}\left[\dot{\Gamma}^{t} - \begin{bmatrix} \bar{\omega}^{t}\Gamma^{-t}\dfrac{\partial \Gamma^{t}}{\partial \theta_1} \\ \bar{\omega}^{t}\Gamma^{-t}\dfrac{\partial \Gamma^{t}}{\partial \theta_2} \\ \bar{\omega}^{t}\Gamma^{-t}\dfrac{\partial \Gamma^{t}}{\partial \theta_3} \end{bmatrix}\right]\frac{\partial T}{\partial \bar{\omega}} - \Gamma^{-t}\begin{bmatrix} \left(\bar{\mathrm{v}}^{\mathrm{B}}\right)^{t} C_{NB}\left(\dfrac{\partial C_{NB}}{\partial \theta_1}\right)^{t} \\ \left(\bar{\mathrm{v}}^{\mathrm{B}}\right)^{t} C_{NB}\left(\dfrac{\partial C_{NB}}{\partial \theta_2}\right)^{t} \\ \left(\bar{\mathrm{v}}^{\mathrm{B}}\right)^{t} C_{NB}\left(\dfrac{\partial C_{NB}}{\partial \theta_3}\right)^{t} \end{bmatrix}\frac{\partial T}{\partial \bar{\mathrm{v}}^{\mathrm{B}}} = \Gamma^{-t} Q_{\bar{\theta}} \qquad (10)$$

(3 x 3) (3 x 3)

Rotational Quasi-Coordinate Equation Of Motion After processing Eq. (10) symbolically for all twelve Euler angle sequences, one obtains the transformed rotational EOM:

$$\frac{d}{dt}\frac{\partial T}{\partial \bar{\omega}} + [\tilde{\omega}]\frac{\partial T}{\partial \bar{\omega}} + \left[\tilde{\mathrm{v}}^{\mathrm{B}}\right]\frac{\partial T}{\partial \bar{\mathrm{v}}^{\mathrm{B}}} = \Gamma^{-t} Q_{\bar{\theta}} \qquad (11)$$

where the new matrices have the following form:

$$[\tilde{\omega}] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}; \quad \left[\tilde{\mathrm{v}}^{\mathbf{B}}\right] = \begin{bmatrix} 0 & -\mathrm{v}_3^{\mathrm{B}} & \mathrm{v}_2^{\mathrm{B}} \\ \mathrm{v}_3^{\mathrm{B}} & 0 & -\mathrm{v}_1^{\mathrm{B}} \\ -\mathrm{v}_2^{\mathrm{B}} & \mathrm{v}_1^{\mathrm{B}} & 0 \end{bmatrix}$$

Many authors (*4,5,6,7*) have identified the first matrix of Eq. (11). Typically the second term has been assumed. The necessary condition presented in Eq. (10) for $[\tilde{\mathrm{v}}^{\mathrm{B}}]$ is believed to be an original contribution.

Translational Equation Transformation

Equation **(4)** is transformed by observing that Eq. **(5)** is only a function of the translational velocity. Equations are required for $\frac{\partial T}{\partial r_i^N}$, $\frac{\partial T}{\partial \mathrm{v}_i^N}$, and $\frac{d}{dt}\frac{\partial T}{\partial \mathrm{v}_i^N}$. The chain

rule of calculus is used to complete the transformation. To this end, the transformation for $\frac{\partial T}{\partial r_i^N}$ follows as:

$$\frac{\partial T}{\partial r_i^N} = \frac{\partial \left(\vec{\mathrm{r}}^{\mathrm{B}}\right)^t}{\partial r_i^N} \frac{\partial T}{\partial \vec{\mathrm{r}}^{\mathrm{B}}} = \frac{\partial \left(\mathrm{C}_{\mathrm{NB}} r_i^N\right)^t}{\partial r_i^N} \frac{\partial T}{\partial \vec{\mathrm{r}}^{\mathrm{B}}} = \left(\mathrm{C}_{\mathrm{NB}}\right)^t \frac{\partial T}{\partial \vec{\mathrm{r}}^{\mathrm{B}}} \tag{12}$$

Similarly, the partial derivative of the inertial velocity vector is given by:

$$\frac{\partial T}{\partial \vec{\mathrm{v}}^{\mathrm{N}}} = \frac{\partial \left(\vec{\mathrm{v}}^{\mathrm{B}}\right)^t}{\partial \vec{\mathrm{v}}^{\mathrm{N}}} \frac{\partial T}{\partial \vec{\mathrm{v}}^{\mathrm{B}}} = \frac{\partial \left(\mathrm{C}_{\mathrm{NB}} \vec{\mathrm{v}}^{\mathrm{N}}\right)^t}{\partial \vec{\mathrm{v}}^{\mathrm{N}}} \frac{\partial T}{\partial \vec{\mathrm{v}}^{\mathrm{B}}} = \left(\mathrm{C}_{\mathrm{NB}}\right)^t \frac{\partial T}{\partial \vec{\mathrm{v}}^{\mathrm{B}}} \tag{13}$$

from which it follows that**:**

$$\frac{d}{dt}\frac{\partial T}{\partial \vec{\mathrm{v}}^{\mathrm{N}}} = \left(\dot{\mathrm{C}}_{\mathrm{NB}}\right)^t \frac{\partial T}{\partial \vec{\mathrm{v}}^{\mathrm{B}}} + \left(\mathrm{C}_{\mathrm{NB}}\right)^t \frac{d}{dt}\frac{\partial T}{\partial \vec{\mathrm{v}}^{\mathrm{B}}} \tag{14}$$

Introducing Eqs. **(12)** and **(14)** into Eq. **(4)**, and multiplying the equation through by $\left(C_{NB}\right)^{-t}$, leads to the transformed equation

$$\frac{d}{dt}\frac{\partial T}{\partial \vec{\mathrm{v}}^{\mathrm{B}}} + \left(\mathrm{C}_{\mathrm{NB}}\right)^{-t}\left(\dot{\mathrm{C}}_{\mathrm{NB}}\right)^t \frac{\partial T}{\partial \vec{\mathrm{v}}^{\mathrm{B}}} - \left(\mathrm{C}_{\mathrm{NB}}\right)^{-t}\left(\mathrm{C}_{\mathrm{NB}}\right)^t \frac{\partial T}{\partial \vec{\mathrm{r}}^{\mathrm{B}}} = \left(\mathrm{C}_{\mathrm{NB}}\right)^{-t} Q_{\vec{\mathrm{r}}^{\mathrm{N}}} \tag{15}$$

This equation is further simplified by using the kinematic identity for the time derivative of the direction cosine matrix (*2,3,5*) given by

$$\dot{C}_{NB} = -\left[\tilde{\omega}\right] C_{NB} \tag{16}$$

where $\left[\tilde{\omega}\right]$ is defined following Eq. **(11)**.

Translational Quasi-Coordinate Equation of Motion Introducing Eq. **(16)** into Eq. **(15)**, one obtains the transformed translational EOM:

$$\frac{d}{dt}\frac{\partial T}{\partial \vec{\mathrm{v}}^{\mathrm{B}}} + \left[\tilde{\omega}\right]\frac{\partial T}{\partial \vec{\mathrm{v}}^{\mathrm{B}}} - \frac{\partial T}{\partial \vec{\mathrm{r}}^{\mathrm{B}}} = \left(\mathrm{C}_{\mathrm{NB}}\right)^{-t} Q_{\vec{\mathrm{r}}^{\mathrm{N}}} \tag{17}$$

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

Equations (11) and (17) represent the desired quasi-coordinate forms for Lagrange's equation, for the base body rotation and translation.

Generalized Coordinates After the quasi-coordinate transformation has been completed, the time derivative of the generalized coordinate vector is

$$\dot{q} = \left(\omega_1, \omega_2, \omega_3, \mathrm{v}_1^{\mathrm{B}}, \mathrm{v}_2^{\mathrm{B}}, \mathrm{v}_3^{\mathrm{B}}, \dot{q}_7, \cdots, \dot{q}_n\right) \tag{18}$$

CONSTRAINT EQUATIONS

Capabilities are provided for supporting user-specified fixed and time-varying constraints. A system-level velocity constraint is defined by:

$$BV = b \tag{19}$$

where B denotes the global constraint matrix, V denotes the system velocity vector, and b denotes the vector of user-specified constraint rates. This equation is valid for all of the velocities in the system model. The system-level velocity vector, for an n-body system, is defined by

$$V = [\omega_1, v_1, \omega_2, v_2, ..., \omega_n, v_n] \tag{20}$$

where the angular velocity and translational velocities are provided for each body. The structure of B is presented in the hinge kinematic section.

To be useful for the quasi-coordinate variables of Eq. (18), one needs a velocity transformation that relates Eqs. (18) and (20). The transformation equation is defined by

$$V = T\,\dot{q} \tag{21}$$

Given Eqs. (18) and (20) where $\omega_j = \omega_j\left(q, \dot{q}\right)$ and $v_j = v_j\left(q, \dot{q}\right)$, the transformation matrix, *T*, is obtained using standard symbolic utilities, so that no special programming is required. Introducing Eq. (21) into Eq. (19), leads to the generalized coordinate constraint matrix

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

$$BT\dot{q} = \bar{B}\dot{q} = b \tag{22}$$

where $\bar{B} = BT$. Generally B and T have a very complicated structure, however, the product BT is very compact and simple.

Hinge Kinematics

The general velocity constraint matrix of Eq. (19) is obtained by considering the constrained DOF at each hinge. A 6x1 kinematic equation for the time derivative of the hinge rates follows as

$$\begin{pmatrix} \dot{\theta} \\ v \end{pmatrix}_h = B_q V_n + B_p V_m \tag{23}$$

where $\dot{\theta}$ denotes the Euler angle hinge rates, v denotes the hinge translational velocities, $V_n = [\omega_n, v_n]$ denotes the outboard body velocity state, $V_m = [\omega_m, v_m]$ denotes the inboard body velocity state, B_p denotes the kinematic transformation matrix that maps the outboard body velocities to the hinge frame, and B_q denotes the kinematic transformation matrix that maps the inboard body velocities to the hinge frame. Equation (23) provides differential equations for the free hinge variables and constraint equations for Eq. (22).

Constrained Hinge Rates

Assuming that the constrained DOT of Eq. (23) have been identified, one obtains a notional system-level constraint equation of the form:

$$B = \begin{bmatrix} B_{p,h_1} & B_{q,h_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & B_{p,h_2} & \cdots & B_{q,h_2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & B_{p,h_m} & B_{q,h_m} \end{bmatrix} \tag{24}$$

where each row has only two matrix inputs, the h_k subscript denotes the matrix block consisting of the rows corresponding to the constrained hinge DOF, and m denotes the

number of hinges (m $\geq$ n-1 because of the possibility of closed-loops existing in the model). The *B*, *T*, and $\bar{B}$ matrices are symbolically generated during the pre-processing step of the model-building algorithm. The time derivative of Eq. (24) is not formulated symbolically, because the product of matrices defined by Eq. (22) represents a relatively small n_c x n_q matrix that is easily differentiated symbolically, n_c denotes the number of user-defined constraints (typically small,) and n_q denotes the number of generalized coordinates in Eq. (18).

Free Hinge Rates

Assuming that the free DOF of Eq. (23) have been identified, kinematic differential equations of the form

$$(\ )_{h,f} = B_{q,f} V_n + B_{p,f} Vm \tag{25}$$

are defined, where (*) denotes an array of free generalized coordinate rates, $h_{,f}$ denotes the free hinge DOF, f denotes the matrix block consisting of the rows corresponding to the free hinge DOF. During symbolic processing, Eq. (25) is saved as a string variable(s) for writing the Fortran subroutine for the system kinematics equations.

Closed-Loop Hinge Kinematics

Closed loops are detected in the input topology data by defining an inboard body list as

$$\begin{array}{c} b_1\ b_2\ b_3 \cdots\ b_n \\ \text{InB:= [0, j , k, ... , p]} \end{array} \tag{26}$$

where b_r denotes the r^{th} body, the integers denote the inboard bodies, and 0 denotes the inertial frame for the base body (b_1). For example, using Eq. (26), the inboard body for b_3 is InB(b_3) = k. This array allows the entire system topology to be examined.

Assuming that hinge σ has been identified as a cut-joint hinge, where the bodies linking the hinge are the outboard body α and inboard body δ; one obtains the cut-loop version of Eq. (23) given by

$$\begin{pmatrix} \dot{\theta} \\ v \end{pmatrix}_{\sigma} = B_{q(\alpha)} V_{\alpha} + B_{p(\delta)} V_{\delta} \tag{27}$$

Equation (26) is used to detect a shared body along the inboard path of bodies for Eq. (27). Assuming that the common body has been identified, say π, then Eq. (23) is used to define a generic body velocity transformation of the form

$$V_n = B^{-1}_{q(n)} \left(\begin{pmatrix} \dot{\theta} \\ v \end{pmatrix}_{h} - B_{p(\,\mathrm{InB(n)}\,)} V_{\mathrm{InB(n)}} \right) \tag{28}$$

where InB(n) is defined by Eq. (26). By repeatedly applying Eq. (28) for Eq. (27), Eq. (27) is transformed so that $V_{\alpha} \Rightarrow V_{\pi}$ and $V_{\delta} \Rightarrow V_{\pi}$. Collecting the intermediate results during the transformation process for Eq. (27) leads to

$$\begin{pmatrix} \dot{\theta} \\ v \end{pmatrix}_{\sigma} = (\ )_{\Sigma} + B_{\Sigma} V_{\pi} \tag{29}$$

where the subscript Σ denotes a collection of all the terms obtained in the transformation. The structure of Eq. (29) differs from Eq. (23) because only one body velocity appears, namely V_{π}.

Generalized Closed-Loop Velocity Constraint Equation

The user-specified time-varying constraint of Eq. (19) is generalized to account for closed-loop constraints of the form of Eq. (29), by identifying the constrained DOF in Eq. (29) and developing the modified system-level constraint matrix

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

$$B_{cl}\,V = b_{cl} \text{ where}$$

$$B_{cl} = \begin{bmatrix} B \\ \left[B_{\Sigma}\right]_c \end{bmatrix} \text{ and } b_{cl} = \begin{pmatrix} b \\ \left(\left(\ \right)_{\sigma,c} - \left(\ \right)_{\Sigma,c}\right) \end{pmatrix} \tag{30}$$

and the constraint block matrix is given by

$$\begin{array}{c} \quad b_1 \quad \cdots \quad b_{\pi} \quad \cdots \quad b_{n-1}\ b_n \\ \left[\left[B_{\Sigma}\right]_c\right] = \left[0 \quad \cdots \quad B_{\Sigma,c} \quad \cdots \quad 0 \quad 0\right] \end{array} \tag{31}$$

where the b_j denote the body numbers and only one block is none-zero, and $B_{\Sigma,c}$ denotes the matrix block matrix of Eq. (29) that corresponds to the constrained DOF for hinge σ.

Introducing the velocity transformation matrix, T, of Eq. (21) the generalized coordinate version of Eq. (30) follows as

$$\bar{B}_{cl}\dot{q} = b_{cl} \quad \text{where} \quad \bar{B} = B_{cl}T \tag{32}$$

Kinematic Differential Equation for the Cut-Loop Hinge Free DOF

The kinematic differential equation for the free cut-loop DOF of hinge σ in Eq. (29), are given by

$$\begin{pmatrix} \dot{\theta} \\ v \end{pmatrix}_{\sigma,f} = \left(\ \right)_{\Sigma,f} + \left(B_{\Sigma}V_{\pi}\right)_{,f} \tag{33}$$

where the subscript f denotes the free DOF. Equation (33) is saved as a string variable(s) for the symbolic generation of the kinematic differential equations.

SYSTEM-LEVEL EQUATIONS OF MOTION

The quasi-coordinate form for the EOM are assembled as follows:

$$\frac{d}{dt}\frac{\partial T}{\partial \vec{\omega}} + [\tilde{\omega}]\frac{\partial T}{\partial \vec{\omega}} + \left[\tilde{\mathrm{v}}^{\mathrm{B}}\right]\frac{\partial T}{\partial \vec{\mathrm{v}}^{\mathrm{B}}} = \Gamma^{-t} Q_{\vec{\theta}}$$

$$\frac{d}{dt}\frac{\partial T}{\partial \vec{\mathrm{v}}^{\mathrm{B}}} + [\tilde{\omega}]\frac{\partial T}{\partial \vec{\mathrm{v}}^{\mathrm{B}}} - \frac{\partial T}{\partial \vec{\mathrm{r}}^{\mathrm{B}}} = \left(\mathrm{C}_{\mathrm{NB}}\right)^{-t} Q_{\vec{\mathrm{r}}^{\mathrm{N}}} \tag{34}$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} = Q_i \; ; \quad i = 7, \cdots, n$$

The symbolic processor evaluates and transforms these equations to simpler forms. The potential energy is assumed to be zero. All loads are assumed to be provided by user-specified force and torque models.

Matrix Form Of The Equations Of Motion

After processing Eq. (34), one has an EOM and constraint equations of the form:

$$M\,\ddot{q} = F_q - \bar{B}^t \lambda \quad \text{and} \quad \bar{B}\dot{q} = b \tag{35}$$

where λ denotes the constraint Lagrange multiplier for both time varying and/or closed-loop constraint calculations. The solution for the constrained generalized coordinates is obtained by differentiating the constraint matrix as

$$\dot{\bar{B}}\dot{q} + \bar{B}\ddot{q} = \dot{b} \tag{36}$$

and combining Eqs. (35) and (36) in the form

$$\begin{bmatrix} M & \bar{B}^t \\ \bar{B} & 0 \end{bmatrix}\begin{pmatrix} \ddot{q} \\ \lambda \end{pmatrix} = \begin{pmatrix} F_q \\ \dot{b} - \dot{\bar{B}}\dot{q} \end{pmatrix} \tag{37}$$

This equation represents the classic descriptor form for the equations of motion. Since the number of time-varying/closed-loop constraints is typically small, this equation is easily inverted symbolically using a sparse $\mathrm{LDL}^{\mathrm{t}}$ (*8*) algorithm, which rigorously eliminates all zero operations through out the linear algebra reduction process.

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

LAGRANGIAN SPARSE PARTIAL DERIVATIVE ALGORITHM

Computing partial derivatives of the system kinetic energy generates the equations of motion for Eq. (34). Two steps are required. First, the kinetic energy partials must be defined. Second, sparse algorithms are introduced for accelerating the generation of Eq. (34).

System Kinetic Energy

For a collection of linked rigid bodies, the system kinetic energy is defined by:

$$T = \frac{1}{2}\sum_{i=1}^{N_B}(M_i\vec{v}_i \cdot \vec{v}_i + \vec{\omega}_i \cdot \vec{\vec{J}}_i \cdot \vec{\omega}_i) \tag{38}$$

where $(\cdot)$ denotes the vector dot product, M_i denotes the mass of the i^{th} body, $\vec{\vec{J}}_i$ denotes the inertia tensor for the i^{th} body with components (referenced to the center of mass), and $\vec{v}_i$ and $\vec{\omega}_i$ denote the i^{th} body translational and rotational velocities. Partial derivatives of Eq. (38) are required for generating the EOM. For engineering-scale applications, building the kinetic energy and computing every partial derivative is very inefficient, because many of the partial derivatives vanish. Great computational efficiency is realized by identifying the functional dependences for the translational and rotational velocities as a pre-processing step. This allows a sparse partial derivative algorithm to be developed for handling these calculations.

Kinetic Energy for a Single Body

For individual bodies the kinetic energy partial derivative calculations are simplified by defining the kinetic energy in the momentum form:

$$T = \left(\vec{\omega} \cdot \vec{h} + \vec{v} \cdot \vec{p}\right)/2 \tag{39}$$

where the body angular momentum is defined by $\vec{h} = \vec{\vec{J}} \cdot \vec{\omega}$ and the body linear momentum is defined by $\vec{p} = M\vec{v}$.

First-Order Single Body Kinetic Energy Partials The partial derivatives of the kinetic energy are defined as $T_{,\xi} = \dfrac{\partial T}{\partial \xi}$. From Eq. (39), the first-order partial derivative w.r.t. q_i follows as

$$T_{,q_i} = \vec{\omega}_{,q_i} \cdot \vec{h} + \vec{v}_{,q_i} \cdot \vec{p} \tag{40}$$

Similarly, the partial derivative of Eq. (39) with respect to $\dot{q}_i$ is given by

$$T_{,\dot{q}_i} = \vec{\omega}_{,\dot{q}_i} \cdot \vec{h} + \vec{v}_{,\dot{q}_i} \cdot \vec{p} \tag{41}$$

Time Derivative of $T_{,\dot{q}}$ Taking the time derivative of Eq. (41) leads to

$$\frac{d}{dt}\left(T_{,\dot{q}_i}\right) = T_{,\dot{q}_i,q_j}\,\dot{q}_j + T_{,\dot{q}_i,\dot{q}_j}\,\ddot{q}_j \tag{42}$$

where the first term of Eq. (42) is given by

$$\begin{aligned} T_{,\dot{q}_i,q_j}.\dot{q}_j = &\left(\vec{\omega}_{,\dot{q}_i,q_j}.\dot{q}_j\right)\cdot\vec{h} + \left(\vec{v}_{,\dot{q}_i,q_j}.\dot{q}_j\right)\cdot\vec{p} + \\ &\vec{\omega}_{,\dot{q}_i}\cdot\left(\vec{h}_{,q_j}.\dot{q}_j\right) + \vec{v}_{,\dot{q}_i}\cdot\left(\vec{p}_{,q_j}.\dot{q}_j\right) \end{aligned} \tag{43}$$

and the second term in Eq. (42) is given by

$$T_{,\dot{q}_i,q_j}.\dot{q}_j = \vec{\omega}_{,\dot{q}_i}\cdot\left(\vec{h}_{,\dot{q}_j}.\ddot{q}_j\right) + \vec{v}_{,\dot{q}_i}\cdot\left(\vec{p}_{,\dot{q}_j}.\ddot{q}_j\right) \tag{44}$$

where the kinematic partial derivative identity $\vec{\omega}_{,\dot{q}_i,\dot{q}_j} = \vec{V}_{,\dot{q}_i,\dot{q}_j} = 0$ has been used, and momentum partials are defined by

$$\vec{h}_{,\xi} = \vec{\vec{J}}.\vec{\omega}_{,\xi} \quad \text{and} \quad \vec{p}_{,\xi} = M\,\vec{v}_{,\xi} \quad \text{for} \quad \xi = q \text{ or } \dot{q} \tag{45}$$

The bracketed terms in Eqs. (40) through (44) indicate that a vector-inner product is evaluated for the j^{th} variables.

Avoiding An Exponential Explosion in the Size of the Symbolic Equations

A key point to observe in Eqs. (40) through (44) is that the linear and angular momentum appear (high-lighted in red). Symbolically, this creates an opportunity to define body level equations. The issue is that $\vec{h}$ and $\vec{p}$ can be defined in two ways during a symbolic computation. First, the symbolically generated expressions for $\vec{\omega}$ and $\vec{v}$ are used, where $\vec{\omega} = f\left(q, \dot{q}\right)$ and $\vec{v} = g\left(q, \dot{q}\right)$. The problem with this approach is that each vector can consist of tens to hundreds of symbolic terms in real-world applications. The problem becomes intractable when many products of symbolic expressions are multiplied together. Second, $\vec{h}$ and $\vec{p}$ can be treated as "numbers", when the partials of Eqs. (40) through (44) are symbolically evaluated and saved for generating optimized Fortran models. The second option is the best choice. The second option acts to reduce an "exponential explosion" in the number to terms appearing in the symbolically generated equations of motion.

During a multibody simulation, that uses Eqs. (40) and (44), the linear and angular momentum vectors are numbers—not symbolic expressions. This observation represents one of the key issues that separates numerical algorithms from symbolic computational strategies; opportunities of this type must be exploited at possible every step in an algorithm.

Mass Matrix Calculation

From Eq. (42) the system mass matrix is defined as

$$M_{ij} = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\vec{\omega}_{,\dot{q}_i} \cdot \vec{h}_{,\dot{q}_j} + \vec{V}_{,\dot{q}_i} \cdot \vec{p}_{,\dot{q}_j} \right) \tag{46}$$

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

where the momentum partial derivatives are defined by Eq. (45). The non-vanishing elements of i, j, and M_{ij} are saved as string variables for symbolic processing. A Fortran mass matrix is written that only updates the time-varying terms—all zero operations are by passed.

Sparse Partial Derivative Calculations

The kinetic energy partial derivatives require calculations of the general form:

$$\psi_{,\xi} \quad \text{where } \xi = \dot{q} \text{ or } \ddot{q}$$

These calculations are simplified by defining the function relationships for ψ as a pre-processing step. The pre-processing step consists of scanning V in Eq. (20), and developing a list-based data structure for storing the sub-lists that identify the functional dependencies for each variable in V, leading to

$$\begin{matrix} b_1 & b_2 & \cdots & b_n \end{matrix}$$
$$\Theta = [[\Upsilon_{\omega_1}, \Upsilon_{v_1}], [\Upsilon_{\omega_2}, \Upsilon_{v_2}], \cdots, [\Upsilon_{\omega_n}, \Upsilon_{v_n}]]$$

where b_j denotes the j^{th} body and Υ denotes a list containing the generalized coordinate rates that appear in each variable. For example, assuming that the rotational and translational variables for the 3nd body have the following functional dependencies

$$\vec{\omega}_3 = \vec{\omega}_3(\vec{\omega}_1, \dot{\theta}_1, \dot{\theta}_3) \text{ and } \vec{v}_3 = \vec{v}_3(\vec{\omega}_1, \vec{v}_1, \dot{\theta}_1, \dot{\theta}_3, \dot{l}_3)$$

then the list functions for $\Upsilon_{\vec{\omega}_3}$ and $\Upsilon_{\vec{v}_3}$ are defined by

$$\Upsilon_{\omega_3} = [\vec{\omega}_1, \dot{\theta}_1, \dot{\theta}_3] \text{ and } \Upsilon_{v_3} = [\vec{\omega}_1, \vec{v}_1, \dot{\theta}_1, \dot{\theta}_3, \dot{l}_3]$$

A pre-processing algorithm also determines the global storage locations for the data contained in Θ, defined as Ω. The structure of Ω is identical to Θ, except that the variables are replaced by integer locations. This data formation is computed one time and used everywhere in the symbolic processing algorithm. The information contained in

Θ permits sparse partial derivatives to be computed. The information contained in Ω permits indirect data storage algorithms to be developed. This data permits a great reduction in the number of partial derivatives that need to be computed for evaluating Eq. (34). As an example, these data-structures are used to reduce computations, consider the follow vector dot product that is summed over all bodies

$$c = [c_1, c_2, \cdots, c_n] = \sum_{i=1}^{n} \vec{a}_i . \vec{b}_{i,\dot{q}} = \sum_{i=1}^{n} \sum_{r \in \Upsilon_i}^{length(\Theta_i)} \vec{a}_{\Theta_i(r)} * \vec{b}_{\Theta_i(r),\dot{q}_r}$$

which replaces the n inner summations with length(Θ_r) summations, while computing the only non-vanishing partial derivatives.

GENERALIZED FORCE

The generalized forces and torques are computed using the following equation:

$$Q_i = \sum_{k=1}^{N_B} \left(\vec{F}_k \cdot \frac{\partial \vec{v}_k}{\partial \dot{q}_i} + \vec{\mathrm{T}}_k \cdot \frac{\partial \vec{\omega}_k}{\partial \dot{q}_i} \right) \tag{47}$$

where Q_i denotes the i^{th} generalized force, $\vec{F}_k$ denotes the k^{th} body force acting at the center of mass, $\vec{\mathrm{T}}_k$ denotes the k^{th} body torque acting at the center of mass. The sparse partial derivatives used for computing the kinetic energy calculations are re-used for the generalized for calculations.

These equations are saved as string variables for generating Fortran Software.

10-Body HMMWV MODEL AND SIMULATION RESULTS

TBD, Data definition, Timing Results, Timing Comparisons for other multibody tools and linear algebra computations, and simulation results

CONCLUSIONS

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

TBD

REFERENCES

1. Lagrange, J.L., “Méchanique Analytique” (Paris, 1788). Many later Additions.

2. Whittaker, E.T., “Analytical Dynamics of Particles and Rigid Bodies”, Dover Publications, New York, 1944, pp. 41-44.

3. Pars, L.A., “A Treatise on Analytical Dynamics”, Ox Bow Press, 1979.

4. Meirovitch, L., “Methods of Analytical Dynamics”, McGraw-Hill Publishing Company, New York, NY (1970), pp. 157-162.

5. Likins, P.W., “Analytical Dynamics and Nonrigid Spacecraft Simulation”, Technical Report 32-1593, July 15, 1974, Jet Propulsion Lab, Pasadena, CA.

6. Meirovitch, L. and Quinn, R.D., “Equations of Motion for Maneuvering Flexible Spacecraft”, Journal of Guidance and Control, Vol. 10, No. 5, Sept.-Oct. 1987, pp. 453-465.

7. Quinn, R. D., “Equations of Motion for Structures in Terms of Quasi-Coordinates”, J. Of Applied Mechanics, Vol. 57, No. 3, pp. 745-749, Sept. 1990.

8. Engeln-Müllges, G, and Uhlig, F., Numerical Algorithms with Fortran, Springer Verlag, 1996.