Proceedings of IDETC/CIE 2005
ASME 2005 International Design Engineering Technical Conferences
& Computers and Information in Engineering Conference
September 24-28, 2005, Long Beach, California USA

DETC2005-85545

APPLICATION OF MULTIBODY DYNAMICS TO ON-ORBIT MANIPULATOR SIMULATIONS

Leslie J. Quiocho
NASA Johnson Space Center
Automation, Robotics, & Simulation Division
Houston, Texas 77058
Email: leslie.j.quiocho@nasa.gov

An Huynh
Jacobs Sverdrup ESCG
Simulation Department
Houston, Texas 77058
Email: an.huynh1@jsc.nasa.gov

Edwin Z. Crues
NASA Johnson Space Center
Automation, Robotics, & Simulation Division
Houston, Texas 77058
Email: edwin.z.crues@nasa.gov

ABSTRACT

This paper discuses a generic multibody dynamics formulation and associated computer algorithm that addresses the variety of manipulator simulation requirements for engineering analysis, procedures development, and crew familiarization/training at the NASA Johnson Space Center (JSC). The formulation is based on body to body relationships with no concept of branched tree topologies. This important notion results in a single recursion pass to construct a system level mass matrix as opposed to the traditional inbound/outbound passes required by the other recursive methods. Moreover, the formulation can be augmented to account for closed loop topologies. The base body of the structure can be fixed or free; each subsequent body, if any, is attached to its parent body via any combination of rotational or translational degrees of freedom (DOFs). Furthermore, each body in the multibody system can be defined as rigid or flexible. The algorithm is designed to partition the data variables and associated computations for multi-frequency or multi-process computation. The resulting algorithm requires approximately one third the computations (in terms of additions and multiplications) of techniques previously used at the NASA JSC.

NOMENCLATURE

K_n Kinetic energy of body n
Π_n Strain energy of body n
q_j The j^{th} generalized DOF of the system
$\dot{q}_j$ First time derivative of q_j
Q_j The j^{th} generalized force
$\vec{v}_n^*$ Absolute velocity of point mass dm inside body n
$\vec{a}_n^*$ Absolute acceleration of dm inside body n
$\vec{x}_n^*$ Position vector of dm in the inertial frame
$\vec{r}_n^*$ Position vector of dm in frame n
$\vec{x}_n$ Position vector of joint n in the inertial frame
$\vec{v}_n$ Absolute velocity of joint n
$\vec{a}_n$ Absolute acceleration of joint n
$\vec{a}_{n,r}^*$ Nonlinear terms of $\vec{a}_n^*$
$\vec{\omega}_n$ Angular velocity of body n
$\dot{\vec{\omega}}_n$ Angular acceleration of body n
φ_n^* Shape function of dm wrt frame n
q_n Flexible DOF of body n
$\dot{q}_n$ First time derivative of q_n
$\ddot{q}_n$ Second time derivative of q_n
A_n Absolute acceleration state of joint n
$M_{rr,n}$ Rigid-rigid mass matrix of body n
$M_{re,n}$ Rigid-elastic mass matrix of body n
$M_{er,n}$ Elastic-rigid mass matrix of body n
$M_{ee,n}$ Elastic-elastic mass matrix of body n
$K_{ee,n}$ Elastic stiffness matrix of body n
F_{np} Force and moment of body p acting on body n
F_{ns} Force and moment of body s acting on body n
r_s Spatial vector from joint n to joint s
$E_{r,n}$ External rigid forces and moments acting on body n
$E_{e,n}$ External elastic forces and moments acting on body n

$B_{r,n}$ Nonlinear rigid inertia force of body n
$B_{e,n}$ Nonlinear elastic inertia force of body n
S_s Mode shape and mode slope of joint s due to body n flex
F_n Force and moment from body p (previous body to n) acting at joint n
F_s Force and moment from body n (previous body to s) acting at joint s
M_{rr} System mass matrix coefficients wrt rigid DOF
M_{re} System mass matrix coefficients wrt flex/rigid DOF
M_{er} System mass matrix coefficients wrt rigid/flex DOF
M_{ee} System mass matrix coefficients wrt flex DOF
K_{ee} System stiffness matrix coefficients of the system
A_0 Absolute acceleration state of base body (index 0)
$\ddot{\theta}$ Rigid DOF of the system with including A_0
$\ddot{q}$ Elastic DOF of the system
τ External joint torques
f_c Joint friction torques
$G_{r,nl}$ Rigid nonlinear generalized force
$G_{e,nl}$ Elastic nonlinear generalized force
$G_{r,ext}$ Rigid external generalized force
$G_{e,ext}$ Elastic external generalized force

INTRODUCTION

At the NASA JSC, numerous manipulator simulation applications and facilities are used to support robotic systems within the Space Shuttle and International Space Station (ISS) programs. One example of this, assembly of the ISS with the Shuttle Remote Manipulator System (SRMS) and the Space Station Remote Manipulator System (SSRMS), is illustrated in Figure 1. These various simulations can be categorized into three primary types: (1) desktop based simulations, (2) hybrid hardware/simulation facilities, and (3) cockpit-in-the-loop simulators.

Desktop based simulations include the:

1. SRMS and SSRMS analytical simulations, for mission planning and post-flight analysis, and
2. SRMS, SSRMS, and Generic Robotic Dynamic Skills Trainers (DSTs), for crew and flight controller skills proficiency and mission specific objective training.

Hybrid hardware/simulation facilities include the:

1. Multi-use Remote Manipulator Development Facility (MRMDF), a *1g* hydraulic trainer that emulates the SSRMS,
2. Neutral Buoyancy Laboratory (NBL) SRMS/SSRMS, underwater manipulators used for EVA crew positioning tasks, and
3. Six-DOF Dynamic Test System (SDTS), a stewart platform used to simulate SRMS/SSRMS docking and berthing mechanism interaction.

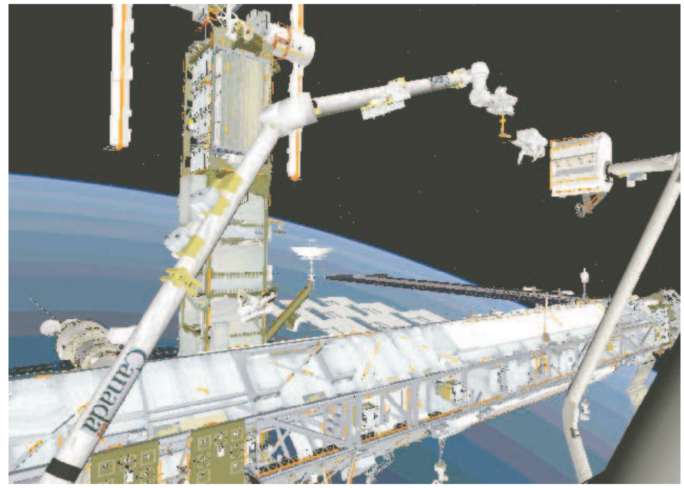

Figure 1. ISS Assembly with SRMS and SSRMS

Integrated end to end cockpit-in-the-loop simulators include the:

1. Space Station Training Facility (SSTF), an integrated ISS systems trainer, and
2. System Engineering Simulator (SES), integrated on-orbit shuttle and ISS cockpit simulators used for crew training, procedures development and engineering evaluations.

Although the above simulations, hardware facilities, and cockpit simulators vary dramatically in terms of requirements, the common thread is their need for multibody equations of motion (EOMs), which is central to simulating any robotic system. Furthermore, due to the real-time nature of the majority of these applications, a computationally efficient multibody dynamics algorithm is of utmost importance (and is especially critical for hardware-in-the-loop).

FORMULATION

The following Lagrangian based formulation has evolved over the past fifteen years at the NASA JSC. The initial effort was limited to single chain, rigid body systems, specifically designed to simulate the 6 DOF SRMS and 7 DOF SSRMS operating off of both a full 6 DOF base or a base fixed with respect to an inertial reference frame [1]. This formulation was also limited to non-real-time or batch applications primarily used for analyzing motion (both Cartesian and joint space). As the requirement expanded to include accurate prediction of joint loading as part of pre-flight analytical activities (also for the SRMS and SSRMS), the formulation was extended to account for joint and link flexibility [2]. However, this extension led to increased complexity and difficulties in the symbolic notation associated with the formulation. This problem was eased with the adoption and adaptation of a systematic notation developed during previous multi-body research efforts [3,4]. In addition, as real-time performance became an issue, it was necessary to investigate recursive techniques to improve computational efficiency [5].

Highlights of the current formulation used at the NASA JSC are provided in the following sections.

Body Level Equations of Motion

Lagrange's equation for body n is given as (assuming no potential energy)

$$\frac{d}{dt}\left(\frac{\partial K_n}{\partial \dot{q}_j}\right) - \frac{\partial K_n}{\partial q_j} + \frac{\partial \Pi_n}{\partial q_j} = Q_j \quad (1)$$

where K_n and Π_n are the kinetic and strain energies of body n,Q_j is the j^{th} generalized force acting on body n, and $q_j, \dot{q}_j$ are the j^{th} generalized coordinate and its derivative with respect to (wrt) time.

It can be shown that for a point mass dm in body n that

$$\frac{d}{dt}\left(\frac{\partial K_n}{\partial \dot{q}_j}\right) - \frac{\partial K_n}{\partial q_j} = \oint \frac{\partial (\vec{v}_n^*)^T}{\partial \dot{q}_j} \vec{a}_n^* dm \quad (2)$$

Equation (1) can now be rewritten as

$$\oint \frac{\partial (\vec{v}_n^*)^T}{\dot{q}_j} \vec{a}_n^* dm + \frac{\partial \Pi_n}{q_j} = Q_j \quad (3)$$

This equation is sometimes referred to as the *Modified Lagrangian* equation.

The position of point mass dm wrt the inertial frame is

$$\vec{x}_n^* = \vec{x}_n + \vec{r}_n^* \quad (4)$$

where $\vec{x}_n^*$, $\vec{x}_n$ are position vectors of dm and joint n in the inertial frame, and $\vec{r}_n^*$ is a vector from joint n to a point of mass dm.

The corresponding velocity and acceleration of dm in inertial coordinates are defined as

$$\vec{v}_n^* = \frac{d\vec{x}_n^*}{dt} = \vec{v}_n + \vec{\omega}_n \times \vec{r}_n^* + \varphi_n^* \dot{q}_n \quad (5)$$

$$\vec{a}_n^* = \frac{d\vec{v}_n^*}{dt} = \vec{a}_n + \dot{\vec{\omega}}_n \times \vec{r}_n^* + \varphi_n^* \ddot{q}_n + \vec{a}_{n,r}^* \quad (6)$$

where $\vec{a}_{n,r}^* = \vec{\omega}_n \times (\vec{\omega}_n \times \vec{r}_n^*)$; the term $2\vec{\omega}_n \times \varphi_n^* \dot{q}_n$ is ignored (assumed negligible).

The rigid body EOMs of body n can be expressed in the following matrix form

$$M_{rr,n} A_n + M_{re,n} \ddot{q}_n = F_{np} + \sum_{s \in O_n} r_s F_{ns} + E_{r,n} + B_{r,n} \quad (7)$$

while the flexible or elastic EOMs for body n are

$$M_{er,n} A_n + M_{ee,n} \ddot{q}_n + K_{ee,n} q_n = \sum_{s \in O_n} S_s^T F_{ns} + E_{e,n} + B_{e,n} \quad (8)$$

where

$$A_n = r_n^T A_p + S_n \ddot{q}_p + P_n \ddot{\theta}_n + A_{n,r}$$

$$E_{e,n} = \sum_e [S_n^E]^T F_n^E$$

$$B_{e,n} = -\vec{\omega}_n^T \beta_n \vec{\omega}_n$$

If one defines

$$F_n = F_{np} = \begin{pmatrix} \vec{F}_n \\ \vec{M}_n \end{pmatrix} \quad (9)$$

as the force and moment from body p (previous body of body n) acting on body n at joint n, then

$$F_s = -F_{ns} = -\begin{pmatrix} \vec{F}_s \\ \vec{M}_s \end{pmatrix} \quad (10)$$

as the force and moment from body n (previous body of body s) acting on body s at joint s.

The resulting EOMs for body n then become (again, in matrix form)

$$M_{rr,n}A_n + M_{re,n}\ddot{q}_n = F_n - \sum_{s\in O_n} r_s F_s + E_{r,n} + B_{r,n} \qquad (11)$$

$$M_{er,n}A_n + M_{ee,n}\ddot{q}_n + K_{ee,n}q_n = -\sum_{s\in O_n} S_s^T F_s + E_{e,n} + B_{e,n} \qquad (12)$$

System Level Equations of Motion

The system level EOMs expressed in physical coordinates are of the following form

$$M_{rr}\ddot{\theta} + M_{re}\ddot{q} = (\tau - f_c) + G_{r,nl} + G_{r,ext} \qquad (13)$$

$$M_{er}\ddot{\theta} + M_{ee}\ddot{q} + K_{ee}q = G_{e,nl} + G_{e,ext} \qquad (14)$$

Equations (13) and (14) can be combined into matrix form as

$$\begin{bmatrix} M_{rr} & M_{re} \\ M_{er} & M_{ee} \end{bmatrix} \begin{pmatrix} \ddot{\theta} \\ \ddot{q} \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_{ee} \end{bmatrix} \begin{pmatrix} \theta \\ q \end{pmatrix} = \begin{pmatrix} \Gamma \\ 0 \end{pmatrix} + \begin{pmatrix} G_{r,nl} \\ G_{e,nl} \end{pmatrix} + \begin{pmatrix} G_{r,ext} \\ G_{e,ext} \end{pmatrix} \qquad (15)$$

or reduced to a simpler matrix form as

$$M\ddot{X} + KX = G_j + G_{nl} + G_{ext} \qquad (16)$$

where M represents the system level mass matrix, K the system level stiffness matrix, G_j the external system torques (e.g., servo and/or joint friction), G_{nl} the generalized nonlinear force and G_{ext} the generalized external force.

Recursive Algorithm

As was shown in [5], a recursive algorithm is utilized to build M, K, G_j, G_{nl}, G_{ext} developed in Equation (16). This algorithm relies on the proven positive definite symmetric nature of the the system level mass matrix to construct only the lower triangular portions of M_{rr}, M_{re}, M_{er}, and M_{ee}. Moreover, it is based on the individual EOMs of the current body n and kinematic relationships with the previous body p to work on a body by body basis without regard to any specific tree topology structure. Highlights of the recursive algorithm are provided here.

Assuming outbound bodies of body n (i.e., for $s > n$), the EOMs provided in Equations (11) and (12) can be modified as

$$M^*_{rr,n}A_n + M^*_{re,n}\ddot{q}_n + f_{r,n}(\ddot{\theta}_s, \ddot{q}_s) = F_n + E^*_{r,n} + B^*_{r,n} \qquad (17)$$

$$M^*_{er,n}A_n + M^*_{ee,n}\ddot{q}_n + f_{e,n}(\ddot{\theta}_s, \ddot{q}_s) + K_{ee,n}q_n = E^*_{e,n} + B^*_{e,n} \qquad (18)$$

where

$$M^*_{rr,n} = M_{rr,n} + r_s M^*_{rr,s} r_s^T$$

$$M^*_{re,n} = M_{re,n} + r_s M^*_{rr,s} S_s$$

$$M^*_{ee,n} = M_{ee,n} + S_s^T M^*_{rr,s} S_s$$

Due to mass matrix symmetry, $M^*_{er,n} = [M^*_{re,n}]^T$.

Pre-multiplying both sides of the Equation (17) by P_n^T results in

$$P_n^T M^*_{rr,n}A_n + P_n^T M^*_{re,n}\ddot{q}_n + P_n^T f_{r,n}(\ddot{\theta}_s, \ddot{q}_s) = \tau_n + P_n^T E^*_{r,n} + P_n^T B^*_{r,n} \qquad (19)$$

$$M^*_{er,n}A_n + M^*_{ee,n}\ddot{q}_n + f_{e,n}(\ddot{\theta}_s, \ddot{q}_s) + K_{ee,n}q_n = E^*_{e,n} + B^*_{e,n} \qquad (20)$$

where the torque at joint n is $\tau_n = P_n^T F_n$. Equation (18) is reintroduced here as Equation (20) for convenience.

The acceleration of joint n can be expressed as

$$\begin{aligned} A_n = {} & r_{1n}^T A_0 + r_{2n}^T P_1 \ddot{\theta}_1 + r_{3n}^T P_3 \ddot{\theta}_3 + \cdots + P_n \ddot{\theta}_n + r_{2n}^T S_1 \ddot{q}_0 \\ & + r_{3n}^T S_2 \ddot{q}_1 + \cdots + S_n \ddot{q}_{n-1} + A^*_{r,n} \end{aligned} \qquad (21)$$

where $A^*_{r,n} = A_{r,n} + r_n^T A^*_{r,n-1}$.

Introducing A_n into Equations (19) and (20) allows the extraction of mass matrix coefficients $M^*_{rr,n}, M^*_{re,n}, M^*_{ee,n}$ for each body n. If one starts with the outer most body N and decrements by 1 until the base body (i.e., index 0) is reached, it can be shown that the number of operations required by the algorithm are linear and proportional to N. In [5], it was demonstrated that this order N algorithm requires approximately one third (1/3) of the additions and multiplications than either Newton-Euler or projection based recursive techniques previously used at the NASA JSC (both of order $\frac{N(N+1)}{2}$).

Closed Kinematic Chains

Some on-orbit simulation scenarios require the use of closed kinematics chains. Examples of this are the grappling of a stowed payload in the orbiter cargo bay by the SRMS for deployment or the handoff of a payload between the SRMS and SSRMS during assembly operations. It is possible to augment Equations (13) and (14) to account for such closed loop topologies.

Consider the following

$$M_{rr}\ddot{\theta} + M_{re}\ddot{q} + J^T_{r,C}F_C = \tau + G_{r,nl} + G_{r,ext} \tag{22}$$

$$M_{er}\ddot{\theta} + M_{ee}\ddot{q} + J^T_{e,C}F_C + K_{ee}q = G_{e,nl} + G_{e,ext} \tag{23}$$

$$J_{r,C}\ddot{\theta} + J_{e,C}\ddot{q} = \sigma^*_C \tag{24}$$

where

$$J^T_{r,C} = \begin{bmatrix} [0] \\ P^T_1 r_{2C_1} \\ P^T_2 r_{3C_1} \\ \cdots \\ P^T_{N-1} r_{NC_1} \\ P^T_N r_{C_1} \end{bmatrix}, \qquad J^T_{e,C} = \begin{bmatrix} [[S_1]^T r_{2C_1} - S^T_{C_2}] \\ S^T_2 r_{3C_1} \\ S^T_3 r_{4C_1} \\ \cdots \\ S^T_N R_{C_1} \\ S^T_{C_1} \end{bmatrix} \tag{25}$$

and

$$\sigma^*_C = \sigma_C - K_{err}\Delta X_C - D_{err}\Delta\dot{X}_C \tag{26}$$

Equations (22), (23), and (24) can be written more simply as

$$M\ddot{X} + J^T_C F_C = G \tag{27}$$

$$J_C\ddot{X} = \sigma^*_C \tag{28}$$

If one assumes that the common point of constraint is given by C, and C_1 is the point on a body and C_2 the connecting point on an adjacent body, then

$$\begin{pmatrix} \vec{a}_{C1} \\ \vec{\omega}_{C1} \end{pmatrix} = \begin{pmatrix} \vec{a}_{C2} \\ \vec{\omega}_{C2} \end{pmatrix} \implies A_{C1} - A_{C2} = 0 \tag{29}$$

Since this constraint only guarantees that accelerations are maintained, the position and velocity states at C_1 and C_2 can drift due to numerical integration error,

$$\Delta X_C = \begin{pmatrix} \vec{r}_{1C_1} - \vec{r}_{C_2} \\ e_{C_1/C_2} \end{pmatrix}, \qquad \Delta\dot{X}_C = \begin{pmatrix} \vec{v}_{C_1} - \vec{v}_{C_2} \\ \vec{\omega}_{C_1} - \vec{\omega}_{C_2} \end{pmatrix} \tag{30}$$

It is therefore necessary to introduce correction coefficients of K_{err} and D_{err}. These coefficents must be selected such that the solution of Equations (27) and (28) remains stable.

IMPLEMENTATION

Once the formulation and corresponding algorithm had been derived, the design requirements for implementation had to be considered. In general, the following high-level objectives were kept in mind during implementation:

1. common and reusable across all supported robotic simulation applications described in the Introduction,
2. generic for any multibody system tree topology (including closed loops),
3. configurable for either rigid or flexible body inputs,
4. recursive to minimize computations while still producing a system, level mass matrix (for modal reduction and corresponding frequency analysis),
5. modular to allow pre/post processing of system specific characteristics (e.g., SRMS, SSRMS, combined SRMS/SSRMS handoff scenarios, Japanese Experimental Module RMS (in work), or Special Purpose Dextrous Manipulator (future)), and

6. structured for interfacing to external models dependent on dynamics parameters such as contact and orbital dynamics.

The resulting multibody dynamics implementation, entitled "MBDYN", meets the above requirements and more. For example, the base body of the given multibody system can be either fixed or free. Each subsequent body, if any, can be attached to its parent body via joints characterized by any combination of rotational or tranlational DOFs. Each body in the system can be partially or completely rigid or flexible in one or all three dimensions.

In order to increase the computational efficiency of MBDYN, several 'tricks' were employed, including the partitioning of data and associated computations. In other words, MBDYN includes a set of subroutines that can be executed independently, at difference frequencies, to optimize the computational performance in a given application. All subroutines work together to produce a complete dynamic solution, but computation time can be reduced (at the cost of fidelity) by lowering the frequencies of execution of some subroutines (or by not calling subroutines at all).

The number of operations to compute the coefficients of the mass matrix and forcing function terms has also been reduced significantly through recursion in MBDYN. Moreover, taking advantage of the symmetric nature of the mass matrix, an efficient Cholesky decomposition [6] is performed on the upper left portion of the matrix; elements pertaining to rigid/elastic coupling, which are more difficult to compute, remain in lower left portion of the matrix. This relationship is essential to accomplishing the single-pass recursion technique previously introduced.

MBDYN has been coded in 'C' and implemented utilizing the Trick Simulation Environment [7], a simulation development and operations "toolkit" used extensively throughout the JSC community for robotic, orbital dynamics, and vehicle guidance, navigations, and control applications.

VERIFICATION AND VALIDATION

As with most software implementations based on complex mathematical formulations, verification of the implementation and validation of the associated results can be a difficult task. The MBDYN development team took a three prong approach to address this issue:

1. Comparison to simple analytical solutions,
2. Verification of momentum and energy using conservation principles, and
3. Validation against other accepted tools

For analytic comparisons, MBDYN was configured to the given simplified system (e.g., rigid two link pendulum) and then compared to the known analytic results [8]. Once these types of comparisons were established, more complex systems such as the 6 DOF and 7 DOF SRMS and SSRMS were formulated and checked to ensure that both linear and angular momentum as well as total system energy were conserved [9].

In the case of systems involving applied forces and torques, comparisons to third party COTS applications such as Treetops and AutoLev were relied upon. In addition, extensive integrated manipulator simulation comparisons (often referred to as 'simulation to simulation validation' by the on-orbit robotics community) were also performed against accepted simulation applications used by both SPAR Aerospace and the Canadian Space Agency as the hardware providers of the SRMS and SSRMS [10–12]. When differences arose as part of these correlation activities, mass matrix coefficients and nonlinear generalized forces were often checked numerically term by term for correctness.

CONCLUDING REMARKS

To summarize, there are several novel or unique features of this dynamics formulation and its associated algorithms. First, it is based upon a clean and systematic notation which help simplify and illuminate the overall dynamics equations. Second, there is a body by body emphasis, with no particular reference to topological structure. Third, it is computationally efficient since it utilizes a single recursion pass to construct the system level mass matrix and generalized force terms. Fourth, it leverages upon mass matrix symmetry to successfully decompose computations. Fifth, it relies on the separation of computations to run at different frequencies or on different processors. Finally, it can be augmented to account for closed kinematic chains.

ACKNOWLEDGMENT

The work described in this paper was performed entirely within the Simulation and Graphics Branch of the Automation, Robotics, and Simulation Division of the NASA JSC Engineering Directorate.

REFERENCES

[1] Quiocho, L.J., and Bailey, R.W., 1990, "On the Dynamics Modeling and Simulation of Robotic Manipulators for Free Space Applications", Technical Memo JSC-26452, NASA Lyndon B. Johnson Space Center, Houston, TX.
[2] Ghosh, T.K., and Huynh, A., 1998, "Multibody Dynamics for Rigid and Flexible Bodies in a Tree Topology for Space Robotic Applications", Technical Memo LMSMS&S-32634, Lockheed-Martin, Houston, TX.
[3] Sincarsin, G.B., and Hughes, P.C., 1989, "Dynamics of an Elastic Multibody Chain: Part A - Body Motion Equations", Dynamics and Stability of Systems, **4**, pp. 209-226.

[4] Hughes, P.C., and Sincarsin, G.B., 1989, ”Dynamics of an Elastic Multibody Chain: Part B - Global Dynamics”, Dynamics and Stability of Systems, **4**, pp. 227-244.

[5] Huynh, A., 2000, ”Formulation and Algorithm for Multi-Flexible-Body Dynamics”, NASA Tech Briefs MSC-22814, Associated Business Publications, New York, NY.

[6] Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P., 1992, *Numerical Recipes in C - The Art of Scientific Computing*, Cambridge University Press, Cambridge, MA.

[7] Paddock, E.J., Lin, A., Vetter, K., and Crues, E.Z., 2003, ”Trick: A Simulation Development Toolkit”, *AIAA Modeling and Simulation Technologies Conference and Exhibit*, AIAA 2003-5809, Austin, TX.

[8] Craig, J.J., 1989, *Introduction to Robotics: Mechanics and Control*, Addison-Wesley, Reading, MA.

[9] Roberson, R.E., Schwertassek, R., 1988, *Dynamics of Multibody Systems*, Springer-Verlag, Berlin.

[10] Nguyen, P., Ravindran, K., Carr, R., Gossain, D.M., Doetsch, K.H., 1982, ”Structural Flexibility of the Shuttle Remote Manipulator System Mechanical Arm”, *AIAA Guidance and Control Conference*, AIAA 1982-1536, San Diego, CA.

[11] Hunter, J.A., Ussher, T.H., and Gossain, D.M., 1982, ”Structural Dynamic Design Considerations of the Shuttle Remote Manipulator System”, *AIAA Structures, Structural Dynamics and Materials Conference*, AIAA 1982-762, New Orleans, LA.

[12] Ma, O., Buhariwala, K., Roger, N., MacLean, J., and Carr, R., 1997, ”MDSF - A Generic Development and Simulation Facility for Flexible, Complex Robotic Systems”, Robotica, **15**, pp. 49-62.

