**Proceedings of The ASME 2016 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference**
**IDETC/CIE 2016**
**August 21-24, 2016, Charlotte, North Carolina, USA**

**DETC2016-60108**

# EVOLUTION OF FLEXIBLE MULTIBODY DYNAMICS FOR SIMULATION APPLICATIONS SUPPORTING HUMAN SPACEFLIGHT

**An Huynh**
**Thomas A. Brain**
**John R. MacLean**

METECS
Houston, Texas 77058
Email: ahuynh@metecs.com,
jmaclean@metecs.com, tbrain@metecs.com

**Leslie J. Quiocho***

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

## ABSTRACT

*During the course of transition from the Space Shuttle and International Space Station programs to the Orion and Journey to Mars exploration programs, a generic flexible multibody dynamics formulation and associated software implementation has evolved to meet an ever changing set of requirements at the NASA Johnson Space Center (JSC). Challenging problems related to large transitional topologies and robotic free-flyer vehicle capture/release, contact dynamics, and exploration missions concept evaluation through simulation (e.g., asteroid surface operations) have driven this continued development. Coupled with this need is the requirement to oftentimes support human spaceflight operations in real-time. Moreover, it has been desirable to allow even more rapid prototyping of on-orbit manipulator and spacecraft systems, to support less complex infrastructure software for massively integrated simulations, to yield further computational efficiencies, and to take advantage of recent advances and availability of multi-core computing platforms. Since engineering analysis, procedures development, and crew familiarity/training for human spaceflight are fundamental to JSC's charter, there is also a strong desire to share and reuse models in both the non-real-time and real-time domains, with the goal of retaining as much multibody dynamics fidelity as possible. Three specific enhancements are reviewed here: (1) linked list organization to address large transitional topologies, (2) body level model order reduction, and (3) parallel formulation/implementation. This paper provides a detailed overview of these primary updates to JSC's flexible multibody dynamics algorithms as well as a comparison of numerical results to previous formulations and associated software.* [1]

## NOMENCLATURE

| | |
|---|---|
| $\theta_n, \dot{\theta}_n, \ddot{\theta}_n$ | Rigid body states of body $n$ |
| $q_n, \dot{q}_n, \ddot{q}_n$ | Flexible body states of body $n$ |
| $\vec{\omega}_n, \dot{\vec{\omega}}_n$ | Angular velocity and acceleration of body $n$ |
| $M_{rr,n}$ | 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 $n$ |
| $D_{ee,n}$ | Elastic damping matrix of body $n$ |
| $A_h^n, F_h^n$ | Head acceleration and force of body $n$ |
| $r_{ns}$ | Spatial vector from body $n$ to body $s$ |
| $S_{ns}$ | Mode shape/slope at node $s$ wrt head of body $n$ |
| $G_{r,n}$ | Rigid nonlinear terms and external forces (sum) |

*Address all correspondence to this author.

[1] This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. Approved for public release; distribution is unlimited.

| | |
|---|---|
| $G_{e,n}$ | Flexible nonlinear terms and external forces (sum) |
| $A_i^n, F_i^n$ | The $i^{th}$ connection acceleration and force of body $n$ |
| $A_{R,ni}$ | The $i^{th}$ remaining connection acceleration of body $n$ |
| $M_s$ | Mass matrix after eliminating flex DOFs of body $s$ |
| $G_s$ | Body force after eliminating flex DOFs of body $s$ |
| $C_{hk}^s$ | Connection coeff. from head to the $k^{th}$ force of body $s$ |
| $C_{lk}^s$ | Connection coeff. from the $l^{th}$ to $k^{th}$ force of body $s$ |
| $b_l^s$ | Known $l^{th}$ connection vector of body $s$ |
| $P_s$ | Driven DOF projection matrix of body $s$ |

## INTRODUCTION

At the NASA JSC, numerous simulation applications and facilities continue to be used to support robotic and spacecraft vehicle systems within the International Space Station (ISS) and Exploration programs. To support those applications, which included both non-real-time and real-time simulators across a variety of platforms and facilities including desktop, hardware-in-the-loop, and human-in-the loop, a generic multibody dynamics formulation and associated computer algorithms called MBDyn were developed. An overview of the application of these multibody dynamics algorithms at JSC to on-orbit manipulator simulations primarily for the Space Shuttle and International Space Station (ISS) programs was provided in [1]. An example ISS robotics branched tree topology is shown in Figure 1. This formulation was later extended to account for differing boundary equations for flexible multibody spacecraft in [2]. This paper describes the evolution of these multibody dynamics algorithms since that timeframe in response to an ever changing set of challenging requirements at JSC. These changes include increased emphasis on transitional topologies such as robotics free-flyer vehicle capture/release, expansion of contact dynamics scenarios for berthing, docking, and surface terrain applications, and rapid-prototyping for evaluation of exploration vehicle and operations concepts. One example application, concept evaluation of an asteriod mission, is illustrated in Figure 2. Three specific enhancements since the initial overview are reviewed here: (1) linked list organization to address large transitional topologies, (2) body level model order reduction, and (3) parallel formulation/implementation. This paper provides a detailed overview of these primary updates to JSC's flexible multibody dynamics algorithms as well as a comparison of numerical results to previous formulations and associated software.

### Previous Formulation

When last described in 2005 [1], the MBDyn formulation was based on system-level equations of motion (EOMs) of the form

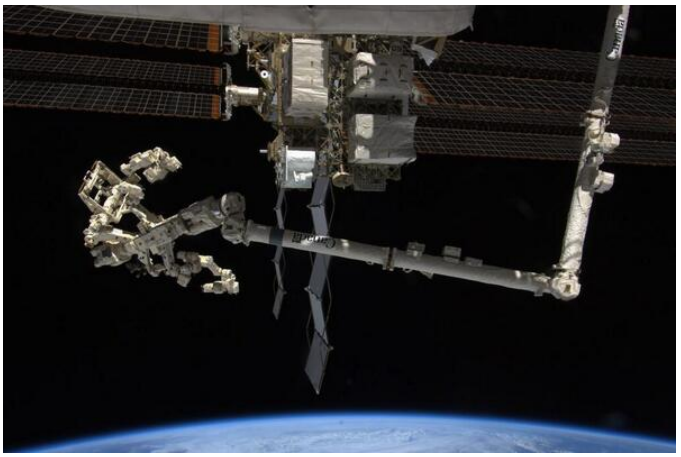

**FIGURE 1**. ISS BRANCHED TREE TOPOLOGY

**FIGURE 2**. EXPLORATION VEHICLE CONCEPT EVALUATION

$$\left.\begin{aligned} M_{rr}\ddot{\theta} + M_{re}\ddot{q} &= \Gamma_n + G_{r,nl} + G_{r,ext} = G_r \\ M_{er}\ddot{\theta} + M_{ee}\ddot{q} + K_{ee}q &= G_{e,nl} + G_{e,ext} = G_e \end{aligned}\right\} \quad (1)$$

The above equations can be solved for $\ddot{\theta}$ and $\ddot{q}$ to yield

$$\begin{pmatrix} \ddot{\theta} \\ \ddot{q} \end{pmatrix} = [M]^{-1} \begin{pmatrix} G_r \\ G_e - K_{ee}q \end{pmatrix} \quad (2)$$

where $M = \begin{bmatrix} M_{rr} & M_{re} \\ M_{er} & M_{ee} \end{bmatrix}$, $K_{ee}$ are system mass and stiffness matrices. The terms $\ddot{\theta}$ and $\ddot{q}$ are rigid and flex degree of freedom (DOF) accelerations. $G_{r,nl}, G_{r,ect}$ are system rigid nonlinear and external forces while $G_{e,nl}, G_{e,ect}$ are the flex equivalents.

Several techniques were employed to achieve real-time performance such as recursive calculation of the system matrices and separation of the system matrices and kinematics transformations into threads executed at slower update rates than the integration stepsize. Parallelism in the overall simulation, including enviromental and vehicle models, was accomplished using tools provided by the Trick Simulation Environment [3], but parallelism in the dynamics itself was limited to the slow update threads. A critical technique used to achieve real-time performance for the systems being simulated was removal of unnecessary high frequency characteristics of the flexible dynamics model prior to integration which avoided the necessity to use extremely small integration time steps. This was accomplished via modal decomposition of the system level equations followed by various options for choice of retained modes [4]. Although this system-level approach proved efficient for supporting real-time simulation of large space manipulators which typically move slowly, evolving requirements significantly increased the system DOFs as well as rates of large motion trajectories. With the increase in DOFs, the CPU time requirements of the $O(n^3)$ system level approach became more dominant and ill-conditioning of the system matrices became of concern. Increases in large motion rates, along with increased requirements for high-fidelity contact dynamics, pushed increases in update rates for kinematics and system matrices. A decision was made to investigate more recent formulations for application to existing and new requirements. Initial evaluation of $O(n)$ algorithms described in [5] showed promise, but to maintain larger integration stepsizes, a method was required to bound high frequency characteristics of the model in the absence of an overall system matrix. In addition, the $O(n)$ algorithm could not take full advantage of our multi-processor simulation host machines. A Divide and Conquer Algorithm (DCA) approach based on [6, 7] was then evaluated. Because the number of cores in our host machines is significantly smaller than the typical number of bodies in on-orbit systems, the DCA algorithm did not perform well compared to the $O(n)$ running on a single processor. A hybrid approach similar to [9,10] which combines $O(n)$ with DCA was then implemented that, along with body-level frequency reduction, provided good performance for real-time simulation without the "slow motion" requirement inherent in the previous formulation.

## Linked-List Organization

A significant reduction in interface software complexity for large multiple-spacecraft simulations was achieved by replacing the previous MBDyn array-based topology definition with an object-oriented interface based on a linked-list architecture. The most fundamental components of the new interface are *Body* and *Joint* definitions. Each Body definition consists of kinematic properties along with optional rigid and flexible properties. Any number of *Nodes* embedded in the body can be defined to represent attachment points, reference frames of interest, and interfaces to external drivers such as environment or flight control system models and contact dynamics engines. Body dynamics properties are expressed with respect to any one of these nodes. Each Joint definition defines an articulated linkage representing from 0 to 6 degrees of freedom along with initial articulation states and mechanisms to collect and apply actuator forces and torques. The Bodies and Joints are organized into *Groups* each representing an independently propagated multibody system. For example, a Group could represent a free-flying space station, a spacecraft, or an asteroid. It could also represent a multibody system attached to the Inertial frame. The topology of each Group is defined using an *Attachment List* which assigns Body Nodes to Joint definitions. This implicitly defines the inboard node of each body which is referred to in this paper as the *Head*. The Attachment List also defines any closed-loop constraints within the Group. Each Group may be propagated using kinematics-only, a combination of kinematics and rigid dynamics, or a combination of kinematics, rigid, and flexible dynamics. Groups may interact with each other through forcing functions such as contact dynamics at any of the Body Nodes. Toplogy transitions may result in a subset of Bodies detaching to form their own Group, or in the addition of Bodies to an existing Group, or from one Group to another. Currently, momentum is conserved for detachments but the attachment process is kinematic and does not guarantee conservation of momentum. This "limitation" has been sufficient for current simulation requirements.

## Body-Level Frequencies

Of fundamental importance to real-time simulation is the ability to limit the highest frequency in the simulation while retaining accuracy in low-frequency behavior. This allows a sufficiently large integration stepsize while maintaining accuracy of response. With an $O(n)$ formulation, the system-level matrices formerly used for modal decomposition are no longer calculated.

Consider the flexible body $n$ shown in Figure 3 which models a single articulated link in the multibody system to be simulated. This body could, for example, be generated by MSC Nastran, or by a lumped mass-spring modeling tool. The body has an inboard connection, $h$, and $s$ possible outboard connections.

$$M_{rr,n}A_h^n + M_{re,n}\ddot{q}_n = F_h^n - \sum_s r_{ns}F_h^s + G_{r,n} \tag{3}$$

$$M_{er,n}A_h^n + M_{ee,n}\ddot{q}_n + K_{ee,n}q_n = -\sum_s S_{ns}^T F_h^s + G_{e,n} \tag{4}$$

Typically for our systems, due to structural modeling techniques, the body level model may contain very high frequencies.

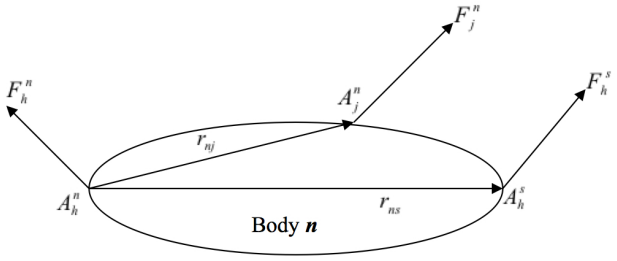


**FIGURE 3**. FLEXIBLE BODY N

These high frequency vibrations are usually not of interest, however, the flexibility associated with these frequencies at the body level may affect low frequency behavior when the body is combined in the larger system. We aim to retain the low-frequency behavior for accuracy while bounding the highest frequency in the multibody system for stability. The objective is to modify $M_{ee,n}$ to reduce the highest frequency $\omega_{n,max}$ of the body $n$ to a desired maximum frequency $\omega_{max}$ while retaining the stiffness matrix $K_{ee,n}$.

The system is transposed into a decoupled set of modal coordinates such that

$$\ddot{\eta}_n + < \omega_n^2 > \eta_n = g_n \tag{5}$$

where $< \omega_n >$ is a diagonal matrix containing the body frequencies. A new set of coordinates $\eta_i^*$ is defined such that for $\omega_i \leq \omega_{max}$,

$$\ddot{\eta}_i^* + 2\varsigma\omega_i\dot{\eta}_i^* + \omega_i^2\eta_i^* = g_i \tag{6}$$

where for most simulations, $\varsigma_i$ is chosen conservatively small. If $\omega_i > \omega_{max}$, then

$$\ddot{\eta}_i^* + 2\varsigma\omega_{max}\dot{\eta}_i^* + \omega_{max}^2\eta_i^* = \frac{\omega_{max}^2}{\omega_i^2}g_i \tag{7}$$

Transforming back to physical coordinates one obtains

$$M_{rr,n}A_h^n + M_{re,n}\ddot{q}_n = F_h^n - \sum_s r_n s F_h^s + G_{r,n} \tag{8}$$

$$M_{er,n}A_h^n + M_{ee,n}^*\ddot{q}_n + D_{ee,n}\dot{q}_n + K_{ee,n}q_n = -\sum_s S_n^T s F_h^s + G_{e,n} \tag{9}$$

It can be shown that the maximum frequency of a resulting multibody system is bounded by $\omega_{max}$.

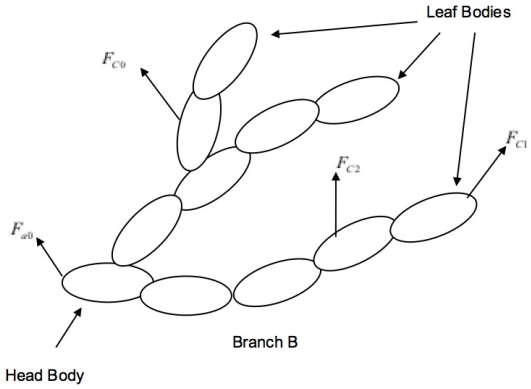


**FIGURE 4**. BRANCH TOPOLOGY EXAMPLE

## Parallel Formulation

Negrut et. al. discussed the why, when, and how questions of using parallelization for multibody system dynamics in [8]. However, the aspects of real-time simulation which are important in our applications were outside the scope of that paper. A detailed overview of the parallel formulation suitable for real-time use is now provided.

Based on the number of CPUs assigned to simulate a Group, the topology of the Group is cut into *Branches* so that there is up to one Branch per available CPU. Each Branch is a tree-topology potentially containing closed loops. The Body Nodes corresponding to the cut points are referred to in this paper as *Connection* Nodes. Each body has one *Head* Node but may have multiple Connection Nodes. An $O(n)$ algorithm is used to simulate the flexible multibody dynamics of each Branch in parallel. The cut points are chosen to distribute the load as evenly as possible between the CPUs. A DCA approach is then used to assemble and disassemble the Branch results. The resulting implementation balances the efficiency of the $O(n)$ algorithm with the parallel capability of the DCA approach.

## Branch Equations

Consider the Branch $B$ topology shown in Figure 4. $F_{a0}$ is the force on the Head Node of Body which is also the root body of the Branch. This force could be due to a Connection with another Branch or it could represent the force at the root Node of the Group. $F_{C0}$ and $F_{C2}$ represent Connection forces at Nodes on the inner bodies while $F_{C1}$ is a Connection force on a Leaf Body of the topology.

The EOMs of a generic Body $n$ are

$$M_{rr,n}A_h^n + M_{re,n}\ddot{q}_n = F_h^n + \sum_j^{I_n} r_{nj}F_j^n - \sum_s r_{ns}F_h^s + G_{r,n} \tag{10}$$

$$M_{er,n}A_h^n + M_{ee,n}\ddot{q}_n + D_{ee,n}\dot{q}_n + K_{ee,n}q_n = \sum_j^{I_n} S_{nj}^T F_j^n - \sum_s S_{ns}^T F_h^s + G_{e,n} \tag{11}$$

$$A_i^n = r_{ni}^T A_h^n + S_{ni}\ddot{q}_n + A_{R,ni} \qquad \forall i \in C_n \tag{12}$$

If Body $n$ is the one of leaf bodies, then the terms $\sum_s r_{ns}F_h^s$ and $\sum_s S_{ns}^T F_h^s$ are dropped due to no outer forces acting on the leaf bodies.

Assume that EOMS of the outer bodies $s$ are

$$M_s A_h^s = F_h^s + \sum_k^{Cs} C_{hk}^s F_k^s + G_s \tag{13}$$

$$A_l^s = C_{lh}^s A_h^s + \sum_k^{Cs} C_{lk}^s F_k^s + b_l^s \qquad \forall l \in C_s \tag{14}$$

The kinematic relationship between $A_h^n$ and $A_h^s$ is

$$A_n^s = r_{ns}^T A_h^n + S_{ns}\ddot{q}_n + P_s\ddot{\theta}_s + A_{R,ns} \tag{15}$$

Substituting $A_n^s$ from (15) into (13) and (14) yields

$$M_s r_{ns}^T A_h^n + M_s S_{ns}\ddot{q}_n + M_s P_s\ddot{\theta}_s = F_h^s + \sum_k^{Cs} C_{hk}^s F_k^s + G_{s,x} \tag{16}$$

$$A_l^s = C_{lh}^s r_{ns}^T A_h^n + C_{lh}^s S_{ns}\ddot{q}_n + C_{lh}^s P_s\ddot{\theta}_s + \sum_k^{Cs} C_{lk}^s F_k^s + b_{l,x}^s \tag{17}$$

If one pre-multiplies both sides of Equation (16) by $P_s^T$ and lets $P_s^T F_h^S = \tau_s$, then $\ddot{\theta}_s$ can be solved as

$$\ddot{\theta}_s = \mu_s^{-1}\left\{-P_s^T M_s r_{ns}^T A_h^n - P_s^T M_s S_{ns}\ddot{q}_n + P_s^T \sum_k^{Cs} C_{hk}^s F_k^s + \tau_{s,x}\right\} \tag{18}$$

Substituting $\ddot{\theta}_s$ from (18) into (16) and (17) and arranging terms, one then obtains

$$\beta_s M_s r_{ns}^T A_h^n + \beta_s M_s S_{ns}\ddot{q}_n = F_h^s + \sum_k^{Cs} \phi_{hk}^n F_k^s + \hat{G}_s \tag{19}$$

$$A_l^s = \phi_{lh}^n r_{ns}^T A_h^n + \phi_{lh}^n S_{ns}\ddot{q}_n + \sum_k^{Cs} \hat{C}_{lk}^n F_k^s + \hat{b}_l^n \qquad \forall l \in C_s \tag{20}$$

where

$$G_{s,x} = G_s - M_s A_{R,ns} \tag{21}$$

$$b_{l,x}^s = b_l^s + C_{lh}^s A_{R,ns} \tag{22}$$

$$\tau_{s,x} = \tau_s + P_s^T G_{s,x} \tag{23}$$

$$\hat{G}_s = G_{s,x} - M_s P_s \mu_s^{-1}\tau_{s,x} \tag{24}$$

$$\mu_s = P_s^T M_s P_s \tag{25}$$

$$\beta_s = [1] - M_s P_s \mu_s^{-1} P_s^s \tag{26}$$

$$\phi_{hk}^n = \beta_s C_{hk}^s \tag{27}$$

$$\hat{C}_{lk}^n = C_{lk}^s + C_{lh}^s P_s \mu_s^{-1} P_s^T C_{hk}^s \tag{28}$$

$$\hat{b}_l^n = b_{l,x}^s + C_{lh}^s P_s \mu_s^{-1}\tau_{s,x} \tag{29}$$

From Equation (19) one can solve for $F_h^s$

$$F_h^s = \beta_s M_s r_{ns}^T A_h^n + \beta_s M_s S_{ns}\ddot{q}_n - \sum_k^{Cs} \phi_{hk}^n F_k^s - \hat{G}_s \tag{30}$$

Now if $F_h^s$ is substituted from (30) into (10) and (11), together with (12) and (20),

$$M_{rr,n}^* A_h^n + M_{re,n}^*\ddot{q}_n = F_h^n + \sum_j^{I_n} r_{nj}F_j^n + \sum_s\sum_k^{Cs} r_{ns}\phi_{hk}^n F_k^s + G_{r,n}^* \tag{31}$$

$$M_{er,n}^* A_h^n + M_{ee,n}^*\ddot{q}_n + D_{ee,n}\dot{q}_n + K_{ee,n}q_n = \sum_j^{I_n} S_{nj}^T F_j^n + \sum_s\sum_k^{Cs} S_{ns}^T \phi_{hk}^n F_k^s + G_{e,n}^* \tag{32}$$

$$A_i^n = r_{ni}^T A_h^n + S_{ni}\ddot{q}_n + A_{R,ni} \qquad \forall i \in I_n \tag{33}$$

$$A_l^s = \phi_{lh}^n r_{ns}^T A_h^n + \phi_{lh}^n S_{ns}\ddot{q}_n + \sum_k^{Cs} \hat{C}_{lk}^n F_k^s + \hat{b}_l^n \qquad \forall l \in C_s \tag{34}$$

where

$$M_{rr,n}^* = M_{rr,n} + \sum_s r_{ns}\beta_s M_s r_{ns}^T \tag{35}$$

$$M_{re,n}^* = M_{re,n} + \sum_s r_{ns}\beta_s M_s S_{ns} \tag{36}$$

$$M_{er,n}^* = M_{er,n} + \sum_s S_{ns}^T \beta_s M_s r_{ns}^T \tag{37}$$

$$M_{ee,n}^* = M_{ee,n} + \sum_s S_{ns}^T \beta_s M_s S_{ns} \tag{38}$$

$$G_{r,n}^* = G_{r,n} + \sum_s r_{ns}\hat{G}_s \tag{39}$$

$$G_{e,n}^* = G_{e,n} + \sum_s S_{ns}^T \hat{G}_s \tag{40}$$

The terms $\sum_s \sum_r^{Cs} r_{ns}\phi_{hk}^n F_k^s$ and $\sum_s \sum_r^{Cs} S_{ns}^T \phi_{hk}^n F_k^s$ can be simplified by considering $C_s^* = \sum_k^{Cs}$,

$$\sum_s \sum_k^{Cs} r_{ns}\phi_{hk}^n F_k^s = \sum_r^{C_s^*} r_{ns}\phi_{hr}^n F_r^s \tag{41}$$

$$\sum_s \sum_k^{Cs} S_{ns}^T \phi_{hk}^n F_k^s = \sum_r^{C_s^*} S_{ns}^T \phi_{hr}^n F_r^s \tag{42}$$

Rewritting Equations (31, 32) as

$$M_{rr,n}^* A_h^n + M_{re,n}^* \ddot{q}_n = F_h^n + \sum_j^{I_n} r_{nj} F_j^n + \sum_r^{C_s^*} r_{ns}\phi_{hr}^n F_r^s + G_{r,n}^* \tag{43}$$

$$M_{er,n}^* A_h^n + M_{ee,n}^* \ddot{q}_n + D_{ee,n}\dot{q}_n + K_{ee,n} q_n = \sum_j^{I_n} S_{nj}^T F_j^n + \sum_r^{C_s^*} S_{ns}^T \phi_{hr}^n F_r^s + G_{e,n}^* \tag{44}$$

the terms $\sum_j^{I_n} r_{nj} F_j^n + \sum_r^{C_s^*} r_{ns}\phi_{hr}^n F_r^s$ and $\sum_j^{I_n} S_{nj}^T F_j^n + \sum_r^{C_s^*} S_{ns}^T \phi_{hr}^n F_r^s$ can be consolidated.

Let $C_n = I_n + C_s^*$, and

$$\phi_{hj}^n = [1], \qquad \hat{C}_{ij}^n = 0, \qquad \forall i, j \in I_n \tag{45}$$

Then

$$\sum_j^{I_n} r_{nj} F_j^n + \sum_r^{C_s^*} r_{ns}\phi_{hr}^n F_r^s = \sum_j^{C_n} r_{ns}\phi_{hj}^n F_j^n \tag{46}$$

$$\sum_j^{I_n} S_{nj}^T F_j^n + \sum_r^{C_s^*} S_{ns}^T \phi_{hr}^n F_r^s = \sum_j^{C_n} S_{ns}^T \phi_{hj}^n F_j^s \tag{47}$$

Equations (33, 34) can be rewritten as

$$A_i^n = \phi_{ih}^n r_{ni}^T A_h^n + \phi_{ih}^n S_{ni}\ddot{q}_n + \sum_j^{Cn} \hat{C}_{ij}^n F_j^n + \hat{b}_j^n \qquad \forall i \in C_n \tag{48}$$

and Equations (31, 32, 33, 34) can be written as

$$M_{rr,n}^* A_h^n + M_{re,n}^* \ddot{q}_n = F_h^n + \sum_j^{Cn} r_{nj}\phi_{hj}^n F_j^n + G_{r,n}^* \tag{49}$$

$$M_{er,n}^* A_h^n + M_{ee,n}^* \ddot{q}_n + D_{ee,n}\dot{q}_n + K_{ee,n} q_n = \sum_j^{Cn} S_{nj}^T \phi_{hj}^n F_j^n + G_{e,n}^* \tag{50}$$

$$A_i^n = \phi_{ih}^n r_{ni}^T A_h^n + \phi_{ih}^n S_{ni}\ddot{q}_n + \sum_j^{Cn} \hat{C}_{ij}^n F_j^n + \hat{b}_i^n, \qquad \forall i \in C_n \tag{51}$$

Solving for $\ddot{q}_n$ from Equation (50)

$$\ddot{q}_n = -X_{ee,n} M_{er,n}^* A_h^n + \sum_j^{Cn} X_{ee,n} S_{nj}^T \phi_{hj}^n F_j^n + Q_n \tag{52}$$

Substituting $\ddot{q}_n$ from (52) into (49, 51) yields

$$M_n A_h^n = F_h^n + \sum_j^{Cn} C_{hj}^n F_j^n + G_n \tag{53}$$

$$A_i^n = C_{ih}^n A_h^n + \sum_j^{Cn} C_{ij}^n F_j^n + b_i^n, \qquad \forall i \in C_n \tag{54}$$

where

$$X_{ee,n} = [M_{ee,n}^*]^{-1} \tag{55}$$

$$M_n = M_{rr,n}^* - M_{re,n}^* X_{ee,n} M_{er,n}^* \tag{56}$$

$$C_{hj}^n = (r_{nj} - M_{re,n}^* X_{ee,n} S_{nj}^T)\phi_{hj}^n \tag{57}$$

$$C_{ih}^n = \phi_{ih}^n (r_{ni}^T - S_{ni} X_{ee,n} M_{er,n}^*) \tag{58}$$

$$C_{ij}^n = \hat{C}_{ij}^n + \phi_{ih}^n S_{ni} X_{ee,n} S_{nj}^T \phi_{hj}^n \tag{59}$$

$$Q_n = X_{ee,n}(G_{e,n}^* - D_{ee,n}\dot{q}_n - K_{ee,n} q_n) \tag{60}$$

$$G_n = G_{r,n}^* - M_{re,n}^* Q_n \tag{61}$$

$$b_i^n = \hat{b}_i^n + \phi_{ih}^n S_{ni} Q_n \tag{62}$$

The Equations (53) and (54) are identical in form to Equations (13) and (14). Continuing in this manner inward to the Head or root body of the branch and then propagating the body accelerations outward to the leaf bodies forms the basis of the $O(n)$ formulation for the solution of the Branch dynamics.

### Combining Branches

Combination bodies are then assembled two at a time using the DCA approach in [9, 10], starting with a *Parent* branch and

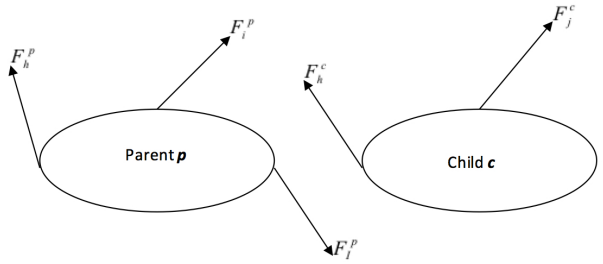


**FIGURE 5**. PARENT AND CHILD BRANCH COMBINATION

a *Child* branch which is outboard of the Parent branch. This is illustrated in Figure 5.

Let Node $I$ be the node on the Parent branch where the child branch is connected. Then accelerations for the Parent branch are

$$A_h^p = \Phi_{hh}^p F_h^p + \sum_{r\neq I}^{N_p} \Phi_{hr}^p F_r^p + \Phi_{hI}^p F_I^p + b_h^p \tag{63}$$

$$A_i^p = \Phi_{ih}^p F_h^p + \sum_{r\neq I}^{N_p} \Phi_{ir}^p F_r^p + \Phi_{iI}^p F_I^p + b_i^p \tag{64}$$

$$A_I^p = \Phi_{Ih}^p F_h^p + \sum_{r\neq I}^{N_p} \Phi_{Ir}^p F_r^p + \Phi_{II}^p F_I^p + b_I^p \tag{65}$$

Solving for $F_I^p$ from (65)

$$F_I^p = x_{II}^p A_I^p - x_{II}^p \Phi_{Ih}^p F_h^p - x_{II}^p \sum_{r\neq I}^{N_p} \Phi_{Ir}^p F_r^p - x_{II}^p b_I^p \tag{66}$$

where $x_{II}^p = [\Phi_{II}^p]^{-1}$.

Similarly, accelerations for the Child branch are given by

$$A_h^c = \Phi_{hh}^c F_h^c + \sum_s^{N_c} \Phi_{hs}^c F_s^c + b_h^c \tag{67}$$

$$A_j^c = \Phi_{jh}^c F_h^c + \sum_s^{N_c} \Phi_{js}^c F_s^c + b_j^c \tag{68}$$

Solving for $F_h^c$ from (67)

$$F_h^c = x_{hh}^c A_h^c - x_{hh}^c \sum_s^{N_c} \Phi_{hs}^c F_s^c - x_{hh}^c b_h^c \tag{69}$$

where $x_{hh}^c = [\Phi_{hh}^c]^{-1}$.

The kinematic relationship between $A_h^c$ and $A_I^p$ is given by

$$A_h^c = A_I^p + P_c \ddot{\theta}_c + A_{R,h}^c \tag{70}$$

Substituting $A_h^c$ from (70) into (69)

$$F_h^c = x_{hh}^c A_I^p + x_{hh}^c P_c \ddot{\theta}_c - x_{hh}^c \sum_s^{N_c} \Phi_{hs}^c F_s^c - y_1^c \tag{71}$$

If one pre-multiplies both sides of (71) by $P_c^T$ and lets $P_c^T F_h^c = \tau_c$, $\ddot{\theta}_c$ can then be solved for as

$$\ddot{\theta}_c = \mu_c^{-1} \left[ \tau_{c,x} + P_c^T x_{hh}^c \left( \sum_s^{N_c} \Phi_{hs}^c F_s^c - A_I^p \right) \right] \tag{72}$$

Substituting $\ddot{\theta}_c$ from (72) into (71), and solving for $F_h^c$,

$$F_h^c = \beta_c x_{hh}^c A_I^p - \beta_c x_{hh}^c \sum_s^{N_c} \Phi_{hs}^c F_s^c - y_2^c \tag{73}$$

where

$$\mu_c = P_c^T x_{hh}^c P_c \tag{74}$$

$$\beta_c = [1] - x_{hh}^c P_c \mu_c^{-1} P_c^T \tag{75}$$

$$A_{R,h}^c = \begin{pmatrix} \vec{0} \\ \tilde{\omega}_c \hat{p}_c \dot{\theta}_c \end{pmatrix} \tag{76}$$

$$y_1^c = x_{hh}^c (b_h^c - A_{R,h}^c) \tag{77}$$

$$\tau_{c,x} = \tau_c + P_c^T y_1^c \tag{78}$$

$$y_2^c = y_1^c - x_{hh}^c P_c \mu_c^{-1} \tau_{c,x} \tag{79}$$

Given

$$F_I^p = -F_h^c \tag{80}$$

Substituting $F_I^p$ from (66) and $F_h^c$ from (73) and solving for $A_I^p$

yields

$$A_I^p = m_{II}^{-1} x_{II}^p \Phi_{Ih}^p F_h^p + m_{II}^{-1} x_{II}^p \sum_{r \neq I}^{N_p} \Phi_{Ir}^p F_r^p + m_{II}^{-1} [\beta_c x_{hh}^c] \sum_{s}^{N_c} \Phi_{hs}^c F_s^c + y_1^p \tag{81}$$

Now substituting $A_I^p$ from (81) into (66) one obtains

$$F_I^p = z_{II}^p \Phi_{Ih}^p F_h^p + z_{II}^p \sum_{r \neq I}^{N_p} \Phi_{Ir}^p F_r^p + w_{pc}^c \sum_{s}^{N_c} \Phi_{hs}^c F_s^c + y_2^p \tag{82}$$

where

$$m_{II}^p = x_{II}^p + [\beta_c x_{hh}^c] \tag{83}$$
$$m_{II}^{-1} = [m_{II}^p]^{-1} \tag{84}$$
$$z_{II}^p = x_{II}^p m_{II}^{-1} x_{II}^p - x_{II}^p \tag{85}$$
$$w_{pc}^c = x_{II}^p m_{II}^{-1} [\beta_c x_{hh}^c] \tag{86}$$
$$y_1^p = m_{II}^{-1} (y_2^c + x_{II}^p b_I^p) \tag{87}$$
$$y_2^p = x_{II}^p (y_1^p - b_1^p) \tag{88}$$

In the case where a Parent branch is combined with a combination body, one can substitute $F_I^p$ from (82) into (63), and (64) to obtain

$$A_h^p = \Phi_{hh}^{p*} F_h^p + \sum_{r \neq I}^{N_p} \Phi_{hr}^{p*} F_r^p + \sum_{s}^{N_c} \Phi_{hs}^{p*} F_s^c + b_h^{p*} \tag{89}$$

$$A_i^p = \Phi_{ih}^{p*} F_h^p + \sum_{r \neq I}^{N_p} \Phi_{ir}^{p*} F_r^p + \sum_{s}^{N_c} \Phi_{is}^{p*} F_s^c + b_i^{p*} \tag{90}$$

where

$$\Phi_{hh}^{p*} = \Phi_{hh}^p + [\Phi_{hI}^p z_{II}^p] \Phi_{Ih}^p \tag{91}$$
$$\Phi_{hr}^{p*} = \Phi_{hr}^p + [\Phi_{hI}^p z_{II}^p] \Phi_{Ir}^p \tag{92}$$
$$\Phi_{hs}^{p*} = [\Phi_{hI}^p w_{pc}^c] \Phi_{hs}^c \tag{93}$$
$$\Phi_{ir}^{p*} = \Phi_{ir}^p + [\Phi_{iI}^p z_{II}^p] \Phi_{Ir}^p \tag{94}$$
$$\Phi_{is}^{p*} = [\Phi_{iI}^p w_{pc}^c] \Phi_{hs}^c \tag{95}$$
$$b_h^{p*} = b_h^p + \Phi_{hI}^p y_2^p \tag{96}$$
$$b_i^{p*} = b_i^p + \Phi_{iI}^p y_2^p \tag{97}$$

In the case where an inboard combination body is combined with a Child branch, one can substitute $A_I^p$ from (81) into (73) to solve for $F_h^c$

$$F_h^c = w_{pc}^T \Phi_{Ih}^p F_h^p + w_{pc}^T \sum_{r \neq I}^{N_p} \Phi_{Ir}^p F_r^p + z_{hh}^c \sum_{s}^{N_c} \Phi_{hs}^c F_s^c + y_{12}^c \tag{98}$$

Substitute $F_h^c$ from (98) into (68) to solve for $A_j^c$

$$A_j^c = \Phi_{jh}^{p*} F_h^p + \sum_{r \neq I}^{N_p} \Phi_{jr}^{p*} F_r^p + \sum_{s}^{N_c} \Phi_{js}^{p*} F_s^c + b_j^{p*} \tag{99}$$

where

$$z_{hh}^c = [\beta_c x_{hh}^c] m_{II}^{-1} [\beta_c x_{hh}^c] - [\beta_c x_{hh}^c] \tag{100}$$
$$\Phi_{js}^{p*} = \Phi_{js}^c + [\Phi_{jh}^c z_{hh}^c] \Phi_{hs}^c \tag{101}$$
$$y_{12}^c = [\beta_c x_{hh}^c] y_1^p - y_2^c \tag{102}$$
$$b_j^{p*} = b_j^c + \Phi_{jh}^c y_{12}^c \tag{103}$$

Finally, in the case where two combination bodies are combined, let

$$A_h^{p*} = A_h^p, \qquad F_h^{p*} = F_h^p \tag{104}$$
$$A_i^{p*} = A_i^p, \qquad F_r^{p*} = F_r^p \tag{105}$$
$$A_j^{p*} = A_j^c, \qquad F_s^{p*} = F_s^c \tag{106}$$

Then Equations (89), (90), and (99) become

$$A_h^{p*} = \Phi_{hh}^{p*} F_h^{p*} + \sum_{r \neq I}^{N_p} \Phi_{hr}^{p*} F_r^{p*} + \sum_{s}^{N_c} \Phi_{hs}^{p*} F_s^{p*} + b_h^{p*} \tag{107}$$

$$A_i^{p*} = \Phi_{ih}^{p*} F_h^{p*} + \sum_{r \neq I}^{N_p} \Phi_{ir}^{p*} F_r^{p*} + \sum_{s}^{N_c} \Phi_{is}^{p*} F_s^{p*} + b_i^{p*} \tag{108}$$

$$A_j^{p*} = \Phi_{jh}^{p*} F_h^{p*} + \sum_{r \neq I}^{N_p} \Phi_{jr}^{p*} F_r^{p*} + \sum_{s}^{N_c} \Phi_{js}^{p*} F_s^{p*} + b_j^{p*} \tag{109}$$

This forms the basis for assembly of branches as per the Divide and Conquer Algorithm.

## Validation and Performance

CPU timing results were generated with the previous system-level MBDyn implementation for two test topologies and

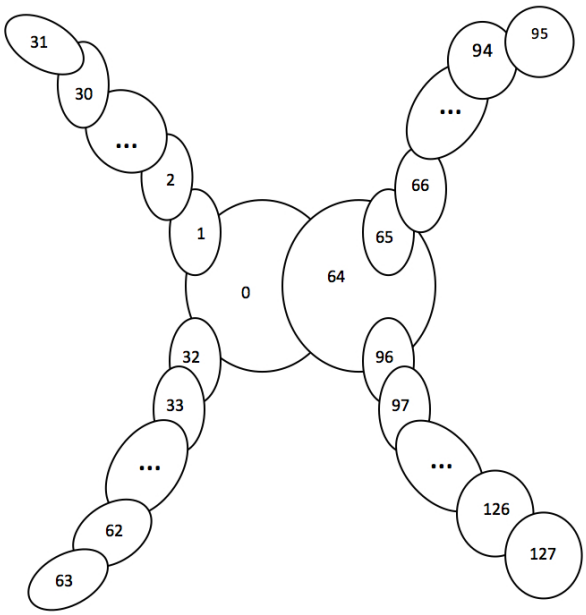


**FIGURE 6**. 128 BODY TOPOLOGY TEST CASE

varying periodic update intervals of the system matrices and transformations. In both test cases, the integration stepsize was 1 ms with a frequency bound of 200 Hz. The tests were executed using a single processor of a mid-level eight processor simulation host computer. The ratio of CPU time to simulation time was calculated not including initialization processing. Values above 1.0 indicate that the multibody dynamics did not run fast enough for it to be used in a real-time simulation application on this particular simulation host. Results are tabulated in Table 1.

The 49 body test case described in [11] illustrates use of periodic system updates to achieve real-time performance. This system includes 41 rigid DOFs and 64 flexible DOFs (for a total of 105 DOFs). As shown in Table 1, system matrix updates at the integration stepsize of 1 ms did not achieve fast enough performance for real-time. Periodic updates at 5 ms and greater achieved real-time performance. At periodic updates of greater than 1 s, the system-level calculations take up an insignificant portion of the CPU time so further increases are of limited return. For comparison, typical update periods for open loop ISS simulations range between 0.5 s and 2.0 s depending on the application. ISS simulations involving contact dynamics or systems with larger articulation rates frequently require smaller update periods to maintain accuracy, thus driving the requirement for the newer formuation.

The 128 body test case shown in Figure 6 illustrates the limitations of the previous approach with increasing system size. This system includes 133 rigid DOFs and 508 flexible DOFs (for a total of 641 DOFs). As shown in Table 1, for this simulation host, increasing the periodic update rate is not sufficient to achieve real-time performance, even without considering its potential negative affect on accuracy.

**TABLE 1**. SYSTEM-LEVEL MBDYN TIMING RESULTS

| System | Update Interval | CPU to Simulation Time Ratio |
|---|---|---|
| 49 Body | 1 ms | 1.35 |
| 49 Body | 5 ms | 0.372 |
| 49 Body | 10 ms | 0.250 |
| 49 Body | 100 ms | 0.139 |
| 49 Body | 1 s | 0.128 |
| 128 Body | 10 ms | 26.1 |
| 128 Body | 100 ms | 5.62 |
| 128 Body | 1 s | 3.56 |
| 128 Body | 2 s | 3.46 |

Table 2 shows results of the test cases with the updated parallel MBDyn implementation. For the 49 body test case, real-time performance is achieved on a single CPU (equivalent to $O(n)$). This is an improvement over the previous formulation because there was no loss in accuracy due to periodic updates. No significant improvement was observed for this system by splitting the topology into parallel branches.

**TABLE 2**. PARALLEL MBDYN TIMING RESULTS

| System | Number of Branches | CPU to Simulation Time Ratio |
|---|---|---|
| 49 Body | 1 | 0.256 |
| 49 Body | 2 | 0.231 |
| 49 Body | 4 | 0.256 |
| 128 Body | 1 | 1.25 |
| 128 Body | 2 | 0.665 |
| 128 Body | 4 | 0.388 |

For the 128 body test case, however, real-time performance is not achieved on a single CPU. In this case, real-time performance is realized by splitting the topology into parallel branches. These results demonstrate the advantage of the parallel formulation for large systems.

## Concluding Remarks

Changing simulation requirements within NASA's human spaceflight program have necessitated the ongoing evolution of multibody dynamics simulations at JSC. Software updates to better handle multiple spacecraft and transient topology scenarios have significantly simplified large complex simulations and enhanced rapid prototyping capabilities to meet the needs of both current ISS and exploration programs. A parallel flexible multibody dynamics formulation has successfully utilized multiple-processor simulation hosts to address the simultaneous needs of higher numbers of degrees of freedom and faster articulation rates. Comparisons against a previously validated implementation show that these improvements have been achieved with no loss of fidelity.

## ACKNOWLEDGMENT

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

## REFERENCES

[1] Quiocho, L.J., Huynh, A., and Crues, E.Z, 2005, "Application of Multibody Dynamics to On-Orbit Manipulator Simulations", *ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, DETC 2005-85545, Long Beach, CA.

[2] MacLean, J.R., Huynh, A., and Quiocho, L.J., 2007, "Investigation of Boundary Conditions for Flexible Multibody Spacecraft Dynamics", *ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, DETC 2007-35511, Las Vegas, NV.

[3] 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.

[4] Hughes, P.C., and Skelton, R.E, 1981, "Modal Truncation for Flexible Spacecraft", *Journal of Guidance and Control*, Vol. 4, No 3, pp. 291-297.

[5] Jain, A., "Unified Formulation of Dynamics for Serial Rigid Multibody Systems", *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 3, pp. 531-542.

[6] Featherstone, R., 1999, "A Divide-and-Conquer Articulated Body Algorithm for Parallel $O(log(n))$ Calculation of Rigid Body Dynamics, Part 1: Basic Algorithm", *International Journal of Robotics Research*, Vol. 18, No. 9, pp. 867-875.

[7] Featherstone, R., 1999, "A Divide-and-Conquer Articulated Body Algorithm for Parallel $O(log(n))$ Calculation of Rigid Body Dynamics, Part 2: Trees, Loops, and Accuracy", *International Journal of Robotics Research*, Vol. 18, No. 9, pp. 876-892.

[8] Negrut, D., Serban, R., Mazhar, H., and Heyn, T., 2014, "Parallel Computing in Multibody System Dynamics: Why, When, and How", *Journal of Computational and Nonlinear Dynamics*, Vol. 9, No. 4, pp. 041007:1-12.

[9] Critchley, J.H., Anderson, K.S., and Binani, A., 2009, "An Efficient Multibody Divide and Conquer Algorithm and Implementation", *Journal of Computational and Nonlinear Dynamics*, Vol. 4, No. 2, pp. 021004:1-10.

[10] Laflin, J.J., Anderson, K.S., Khan, I.M., and Poursina, M., 2014, "Advances in the Application of the Divide-and-Conquer Algorithm to Multibody System Dynamics", *Journal of Computational and Nonlinear Dynamics*, Vol. 9, No. 4, pp. 041003:1-8.

[11] Ghosh, T.K., and Quiocho, L.J., 2013, "Development and Evaluation of an Order-N Formulation for Multi-flexible Body Space Systems", *EUROSIS 11th Annual Industrial Simulation Conference*, Ghent, Belgium.