Momentum-Based Dynamics for Spacecraft with Chained Revolute Appendages

Steven Queen
NASA Goddard Space Flight Center
Flight Dynamics Analysis Branch
Greenbelt, Maryland

Ken London and Marcelo Gonzalez
Swales Aerospace Inc.
Guidance, Navigation and Control Group
Beltsville, Maryland

ABSTRACT

An efficient formulation is presented for a sub-class of multi-body dynamics problems that involve a six degree-of-freedom base body and a chain of N *rigid* linkages connected in series by single degree-of-freedom revolute joints. This general method is particularly well suited for simulations of spacecraft dynamics and control that include the modeling of an *orbiting* platform with or without internal degrees of freedom such as reaction wheels, dampers, and/or booms. In the present work, particular emphasis is placed on dynamic simulation of multi-linkage *robotic manipulators*. The differential equations of motion are explicitly given in terms of linear and angular *momentum states*, which can be evaluated recursively along a serial chain of linkages for an efficient real-time solution on par with the best of the $\mathcal{O}\left(N^3\right)$ methods.

INTRODUCTION

The solution to multi-body dynamics problems, and specifically real-time robotic simulation, has remained an intellectually and computationally challenging problem for the past 35 years. A multitude of formulations of the dynamic simulation (forward dynamics) problem exist in the literature, each with its own particular strengths. Traditionally, the methods have been grouped into formulations with *dependent coordinates* and *independent coordinates*. The former requires solutions of m ordinary differential equations (ODE) along with an associated n number of algebraic equations of constraint, where the total number of degrees of freedom (N) is equal to $m-n$. The *dependent coordinate* methods tend to be very general but more difficult to solve and are the basis of commercial packages such as ADAMS© or DADS©. Formulations based on a minimal set of *independent coordinates* are more specific to an application, but can be correspondingly more efficient. The methods used to derive the equations of motion also vary considerably (e.g. Euler-Lagrange[1], D'Alembert[2] and Newton-Euler[3]) but ultimately the resulting equations are equivalent[4]. Further delineation between methods can be made based on whether the final treatment of the equations of motion are *explicit* or *recursive*. *Explicit* solutions (sometimes referred to as *closed-form* [5]—a misnomer since numerical integration is still required for a time-domain solution) tend to be exceedingly complex expressions, even for a fairly small number of degrees-of-freedom, and typically are generated using symbolic manipulation programs such as MATHEMATICA©, MAPLE© or AUTOLEV©. Furthermore, the *explicit* solution is specific to a particular problem/configuration and for that reason also has the potential to be the most computationally efficient [5, 6]. A sub-category of *independent coordinate* methods, that applies to serial or branched manipulator systems, uses a *recursive* algorithm to solve the equations of motion. *Recursive* solvers are historically very fast, especially for the small number of degrees of freedom ($N < 10$) typical to aerospace applications [7].

In the present work, a *recursive* form applicable to open-loop, chain-like, serial manipulators with *rigid* linkages is considered. The form of the equations presented differs from both the Euler-Lagrange method as well as from the traditional Newton-Euler approach[3]. The former method that uses ODE's derived from energy and Lagrange multipliers for elimination of constraints. The latter method solves for relative joint *accelerations* given the input torques at a particular trajectory point by inverting a (mass) matrix that is assembled via multiple passes of a (forward-backward) recursive *inverse dynamics* algorithm[10] (i.e., a

method of obtaining joint torques given the link velocities and accelerations). Instead, the method presented here is an extension of work by P. C. Hughes[8] which uses a topology analogous to the *direct-path method* of Ho[9]. This method is written in terms of first-order differential equations for the linear and angular momenta. In choosing Hughes' *momentum-state dynamics* (MSD) formulation we gain advantages over traditional Newton-Euler in being able to: fully couple the translational and rotational dynamics (i.e. *floating-platform*) in a straightforward and rigorous manner; to include other momentum storage devices such as reaction wheels, dampers and gears; and to directly apply environmental forces/moments at arbitrary locations along the serial chain.

First, for context, a table of notational differences between the present multi-body equations and those of Hughes[8] is presented. Next, this paper presents the recursive equations of motion suitable for real-time dynamic simulation, provides a low-order (non-recursive) example of the form of the equations, and outlines the solution algorithm. Clarification on a method for including disturbances is then discussed followed by an example application in the field of space-based robotic control. Lastly, consideration is given to computational efficiency, numerical precision, and future extensibility.

NOTATION

Since a derivation of the dynamics is not included in this paper, rather the final form of the equations, it is important to understand that the underlying development is an extension to the work of P. C. Hughes[8]. In general, an effort was made to comply with the notation and style of this reference. Scalars are expressed in a *medium italic* typeface, and 3×1 column 3×3 square matrices are expressed in **boldface**. A superscript cross ($^\times$) indicates a left-unary operation on a column vector that forms a skew-symmetric matrix, and a superscript ($^\intercal$) indicates a transpose of a matrix. For *matrices*, the over-dot notation, ($\dot{\ }$), is used to represent the time derivative of a variable with *respect to a non-inertial body-fixed frame.* In contrast, the over-dot on the *Gibbsian vectors* of Eqs.(1) and Eq.(2), are time derivatives with respect to an inertial frame.

Table 1: Description and Cross-Referencing of Symbols

Symbol	Ref.[8]	Description
$\mathcal{A}_m^n$	$\mathbf{C}_{mn}$	attitude (direction cosine) matrix that transforms via left multiplication a column vector from the n^{th} to the m^{th} frame
$\hat{\mathbf{a}}_n$	$\mathbf{a}$	axis of rotation of the n^{th} joint expressed in the n^{th} body frame
$\mathbf{b}_{m,n}$	$\mathbf{b}$	vector expressed in the m^{th} body frame pointing from the m^{th} body origin to the n^{th} body origin
$\mathbf{C}_n, \mathbf{c}_n$	$\mathbf{c}$	1^{st} mass moment $(m_n \cdot (r_{cm})_n)$ of an individual body ($\mathbf{c}_n$) or the composite ($\mathbf{C}_n$) of the n^{th} body and all bodies outboard, expressed in the n^{th} body-fixed frame
$\mathbf{f}_n^{\text{ext}}, \mathbf{F}_n^{\text{ext}}$	$\mathbf{f}$	external (composite) force acting on the n^{th} body/composite expressed in the n^{th} body-fixed frame
$\mathbf{g}_n^{\text{ext}}, \mathbf{G}_n^{\text{ext}}$	$\mathbf{g}$	external (composite) moment acting about the n^{th} body's origin expressed in the n^{th} body-fixed frame
$\mathbf{g}_n$	$\mathbf{g}_p$	inter-body moments exerted on the n^{th} body by the $(n-1)^{th}$ body, about the n^{th} body origin, expressed in the n^{th} body frame
m_n, M_n	m_n	mass of n^{th} body / composite mass of n^{th} body plus all outboard bodies
$\mathbf{h}_n, \mathbf{H}_n, H_n^a$	$\mathbf{h}_n, \mathbf{h}$	absolute angular momentum of n^{th} body/composite of n^{th} body plus outboard bodies, expressed in the n^{th} body frame (or projected along $\hat{\mathbf{a}}_n$)
$\mathbf{j}_n, \mathbf{J}_n$	$\mathbf{J}_n$	2^{nd} mass moment of inertia of the n^{th} body and the composite of n^{th} body plus all outboard bodies, expressed in the n^{th} body frame
$\mathbf{J}_{m,n}$	$\mathbf{J}_{12}$	"mixed" 2^{nd} mass moment of inertia of the n^{th} and all outboard bodies, expressed both in the m^{th} and in the n^{th} body frames
$\mathbf{P}_n$	$\mathbf{p}$	composite linear momentum of the n^{th} and outboard bodies expressed in the n^{th} body frame
$\mathbf{v}_n$	$\mathbf{v}$	velocity of the n^{th} body with respect to the inertial frame, expressed in the n^{th} body frame
ω_n, ω_n^a	ω_p	inertial angular velocity of n^{th} body or, relative scalar component of n^{th} body wrt $(n-1)^{th}$ body along the joint axis, expressed in n^{th} frame

In some instances it is expedient to depart from Hughes' notation. One example, *vectrices*—an invention of Hughes for book-keeping the projection of vectors into coordinate frames (and a source of some confusion to the unfamiliar)—are used in the derivation but are not needed in this paper's context. Instead (except for Eqs(1,2)), the equations are expressed in *matrix form* only, with the frame explicitly indicated. Additionally, other small changes are made where it is felt it benefits clarity, such as the use of uppercase to imply a *composite* property, or one belonging to a collection of bodies—usually the n^{th} body and outboard to the N^{th}. A cross-reference between the present N-link momentum state dynamics and Hughes (two-body) notation is given in Table 1.

DYNAMICAL EQUATIONS OF MOTION

In Hughes[8], the familiar Newton-Euler equations of motion for a *rigid body* are derived from first principles and expressed in terms of Gibbsian vectors:

$$\dot{\underrightarrow{\mathbf{p}}} = \underrightarrow{\mathbf{f}} \tag{1}$$

$$\dot{\underrightarrow{\mathbf{h}}} + \underrightarrow{\mathbf{v}}_o \times \underrightarrow{\mathbf{p}} = \underrightarrow{\mathbf{g}}_o \tag{2}$$

where the linear momentum of the body is $\underrightarrow{\mathbf{p}} = m\underrightarrow{\mathbf{v}}_o + \underrightarrow{\mathbf{c}} \times \underrightarrow{\omega}$, and the angular momentum about the body-fixed reference origin is $\underrightarrow{\mathbf{h}}_o = \underrightarrow{\mathbf{c}} \times \underrightarrow{\mathbf{v}}_o + \underleftrightarrow{\mathbf{J}} \circ \underrightarrow{\omega}$. The only unfamiliar term might be the first mass moment vector ($\underrightarrow{\mathbf{c}}$), defined by an integral of the mass density function, $\rho(\underrightarrow{\mathbf{r}})$, over the rigid body, $\mathcal{R}$, as $\underrightarrow{\mathbf{c}} = \int_{\mathcal{R}} \underrightarrow{\mathbf{r}}\rho(\underrightarrow{\mathbf{r}})dV = m \cdot \underrightarrow{\mathbf{r}}_{cm}$. With the simplification that the *origin* of the rigid-body coordinate system is *at the center of mass* (i.e. $\underrightarrow{\mathbf{c}} = \underrightarrow{\mathbf{0}}$), and the notational adjustment that the second mass moment of inertia dyadic ($\underleftrightarrow{\mathbf{J}}$) is expressed as $\underleftrightarrow{\mathbf{I}}$ when it is about the center of mass, the equations of motion take a more familiar form:

$$\dot{\mathbf{p}} = -\boldsymbol{\omega}^{\times} m\mathbf{v}_{cm} + \mathbf{f} \tag{3}$$

$$\dot{\mathbf{h}}_{cm} = -\boldsymbol{\omega}^{\times} \mathbf{I}\boldsymbol{\omega} + \mathbf{g}_{cm} \tag{4}$$

where Eqs.(3) and (4) are now expressed in *matrix form*, with the vectors realized in the non-inertial body-fixed frame. Note that these are first order ODEs with *momentum variables* on the left-hand side and *velocity variables* on the right-hand-side. Hughes[8] also introduces symbols for the *generalized momentum* ($\mathcal{P}$), *generalized mass* ($\mathcal{M}$), and *generalized velocity* ($\mathcal{V}$) matrices. For this simplified example, the equation that relates the *generalized momenta* to the *generalized velocities*—again in matrix form expressed in the body-fixed frame—is as follows:

$$\mathcal{P} = \mathcal{M}\mathcal{V} \tag{5}$$

$$\begin{bmatrix} \mathbf{p} \\ \mathbf{h}_{cm} \end{bmatrix} = \begin{bmatrix} m \cdot \mathbb{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{cm} \\ \boldsymbol{\omega} \end{bmatrix} \tag{6}$$

The intent is not to over-complicate the classically simple equations for rigid body spacecraft position and attitude motion (which are de-coupled by the choice of the center of mass as the origin) but instead to make an explicit notational bridge between the method to be presented and the familiar territory of practicing engineers—acknowledging that notation is often the largest barrier in the acceptance of a new methodology.

Returning to the more general case, in which the base coordinate system does not coincide with the system center of mass and the translational and rotational equations of momentum are coupled, Hughes[8] shows how this straight-forward framework can be extended to several simple, but powerful, multi-body applications: a point mass (sliding) damper, a rotor, and an appended rigid body connected to the "base" body by a three degree-of-freedom spherical joint. Additionally, Ford and Hall[11] also applied the Hughes framework to the problem of modeling gimbaled momentum wheels (Control Moment Gyros). To this tool-chest of spacecraft component models, the present work adds an efficient formulation for an appendage consisting of a chain of N rigid linkages connected in series by single degree-of-freedom rotational joints. Structures of

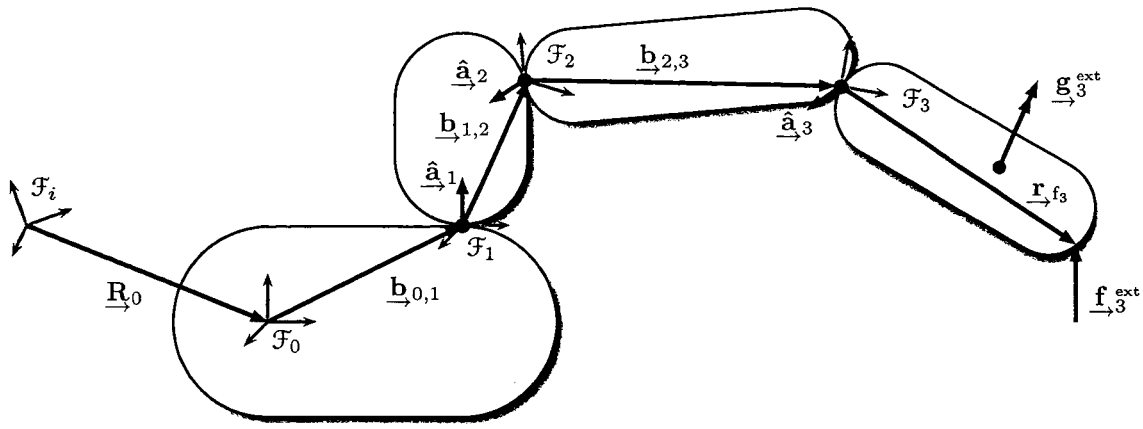

Figure 1: Four Body (Three Linkage) System Configuration

this nature are often used as *robotic manipulators*, which is in fact the motivation for this development effort.

For the purpose of exposition, a four-body ("base" body plus three linkages) example will be presented explicitly. Once the form of the equations is made clear, the more general N-body recursive formula will be given. The four-body system configuration is shown in Figure 1. Note that $\mathcal{F}_i$ is an inertial reference frame, and $\mathcal{F}_0$ is the "base" body-fixed frame that is nominally *not* located at the system center of mass or at the base body center of mass. Each linkage has an embedded frame, $\mathcal{F}_n$, offset from the frame of the previous link (or base) by an outboard translation ($\mathbf{b}_{n,n+1}$) and associated outboard rotation parameters (quaternion), $\mathbf{q}^n_{n+1}$. The two static configuration parameters, $\mathbf{b}_n$ and $\mathbf{q}^n_{n+1}$, produce an equivalent homogeneous transformation matrix between bodies similar to Denavit-Hartenberg[12] parameters common to robotics literature. When the rotational degree-of-freedom is aligned with an axis of the link frame (typically the z-axis), the simplification can be used to improve computational efficiency. The expression for the generalized momentum of the four-body system can be written:

$$\mathcal{P} = \mathcal{M}\mathcal{V} \tag{7}$$

$$\begin{bmatrix} \mathbf{P}_0 \\ \mathbf{H}_0 \\ H_1^a \\ H_2^a \\ H_3^a \end{bmatrix} = \begin{bmatrix} M_0 \cdot \mathbb{I} & -\mathbf{C}_0^\times & -\mathcal{A}_0^1\mathbf{C}_1^\times\hat{\mathbf{a}}_1 & -\mathcal{A}_0^2\mathbf{C}_2^\times\hat{\mathbf{a}}_2 & -\mathcal{A}_0^3\mathbf{C}_3^\times\hat{\mathbf{a}}_3 \\ \mathbf{C}_0^\times & \mathbf{J}_0 & \mathbf{J}_{0,1}\hat{\mathbf{a}}_1 & \mathbf{J}_{0,2}\hat{\mathbf{a}}_2 & \mathbf{J}_{0,3}\hat{\mathbf{a}}_3 \\ \hat{\mathbf{a}}_1^\intercal\mathbf{C}_1^\times\mathcal{A}_1^0 & \hat{\mathbf{a}}_1^\intercal\mathbf{J}_{1,0} & \hat{\mathbf{a}}_1^\intercal\mathbf{J}_1\hat{\mathbf{a}}_1 & \hat{\mathbf{a}}_1^\intercal\mathbf{J}_{1,2}\hat{\mathbf{a}}_2 & \hat{\mathbf{a}}_1^\intercal\mathbf{J}_{1,3}\hat{\mathbf{a}}_3 \\ \hat{\mathbf{a}}_2^\intercal\mathbf{C}_2^\times\mathcal{A}_2^0 & \hat{\mathbf{a}}_2^\intercal\mathbf{J}_{2,0} & \hat{\mathbf{a}}_2^\intercal\mathbf{J}_{2,1}\hat{\mathbf{a}}_1 & \hat{\mathbf{a}}_2^\intercal\mathbf{J}_2\hat{\mathbf{a}}_2 & \hat{\mathbf{a}}_2^\intercal\mathbf{J}_{2,3}\hat{\mathbf{a}}_3 \\ \hat{\mathbf{a}}_3^\intercal\mathbf{C}_3^\times\mathcal{A}_3^0 & \hat{\mathbf{a}}_3^\intercal\mathbf{J}_{3,0} & \hat{\mathbf{a}}_3^\intercal\mathbf{J}_{3,1}\hat{\mathbf{a}}_1 & \hat{\mathbf{a}}_3^\intercal\mathbf{J}_{3,2}\hat{\mathbf{a}}_2 & \hat{\mathbf{a}}_3^\intercal\mathbf{J}_3\hat{\mathbf{a}}_3 \end{bmatrix} \begin{bmatrix} \mathbf{v}_0 \\ \boldsymbol{\omega}_0 \\ \omega_1^a \\ \omega_2^a \\ \omega_3^a \end{bmatrix} \tag{8}$$

where $\mathbb{I}$ is the 3×3 identity matrix. The associated first order differential equations of motion for the momentum states are:

$$\dot{\mathbf{P}}_0 = -\boldsymbol{\omega}_0^\times\mathbf{P}_0 + \mathbf{F}_0^{\text{ext}} \tag{9}$$

$$\dot{\mathbf{H}}_0 = -\boldsymbol{\omega}_0^\times\mathbf{H}_0 - \mathbf{v}_0^\times\mathbf{P}_0 + \mathbf{G}_0^{\text{ext}} \tag{10}$$

$$\dot{H}_1^a = \boldsymbol{\omega}_1^\intercal\hat{\mathbf{a}}_1^\times\mathbf{H}_1 + \mathbf{v}_1^\intercal\hat{\mathbf{a}}_1^\times\mathbf{P}_1 + \hat{\mathbf{a}}_1^\intercal\mathbf{g}_1 + \hat{\mathbf{a}}_1^\intercal\mathbf{G}_1^{\text{ext}} \tag{11}$$

$$\dot{H}_2^a = \boldsymbol{\omega}_2^\intercal\hat{\mathbf{a}}_2^\times\mathbf{H}_2 + \mathbf{v}_2^\intercal\hat{\mathbf{a}}_2^\times\mathbf{P}_2 + \hat{\mathbf{a}}_2^\intercal\mathbf{g}_2 + \hat{\mathbf{a}}_2^\intercal\mathbf{G}_2^{\text{ext}} \tag{12}$$

$$\dot{H}_3^a = \boldsymbol{\omega}_3^\intercal\hat{\mathbf{a}}_3^\times\mathbf{H}_3 + \mathbf{v}_3^\intercal\hat{\mathbf{a}}_3^\times\mathbf{P}_3 + \hat{\mathbf{a}}_3^\intercal\mathbf{g}_3 + \hat{\mathbf{a}}_3^\intercal\mathbf{G}_3^{\text{ext}} \tag{13}$$

The most notable difference between Eqs.(8-13) and those of the rigid-body, Eqs.(3-6), are the three additional states (H_1^a, H_2^a, H_3^a) representing the composite absolute angular momentum of the n^{th} through N^{th} links, about the n^{th} link origin (i.e. joint), projected onto the n^{th} joint's axis of rotation ($\hat{\mathbf{a}}_n$). Additionally, there are three new associated (scalar) angular velocities (ω_n^a), representing the *relative* angular rate of the

n^{th} body with respect to the $(n-1)^{th}$ body about $\hat{\mathbf{a}}_n$. Also, the shift from lowercase to uppercase notation is a mnemonic for the fact that these momenta are for a collection of bodies (n^{th} plus all outboard) which are referred to here as *composite* properties. Finally, the generalized mass matrix is now fully populated—indicating that the translational and rotational motion is no longer de-coupled. The application of the internal and external forces and moments $(\mathbf{g}_n, \mathbf{F}_n^{\text{ext}}, \mathbf{G}_n^{\text{ext}})$ will be covered in a dedicated section of this paper.

The calculation of the time rate of change of the composite momenta is somewhat complicated by the fact that the derivatives depend on the instantaneous momenta of the body and those outboard, but only the total system momenta $(\mathbf{P}_0, \mathbf{H}_0)$ and the scalar components of the momentum about the individual joint axes (H_n^a) are propagated as states. The (vectorial) momenta are reconstructed using the following relationships, as shown for the four-body case:

$$\begin{aligned}
\boldsymbol{\omega}_1 &= \mathcal{A}_1^0\boldsymbol{\omega}_0 + \omega_1^a\hat{\mathbf{a}}_1 && (14) \qquad & \mathbf{v}_1 &= \mathcal{A}_1^0\left(\mathbf{v}_0 + \boldsymbol{\omega}_0^\times\mathbf{b}_{0,1}\right) && (17)\\
\boldsymbol{\omega}_2 &= \mathcal{A}_2^1\boldsymbol{\omega}_1 + \omega_2^a\hat{\mathbf{a}}_2 && (15) & \mathbf{v}_2 &= \mathcal{A}_2^1\left(\mathbf{v}_1 + \boldsymbol{\omega}_1^\times\mathbf{b}_{1,2}\right) && (18)\\
\boldsymbol{\omega}_3 &= \mathcal{A}_3^2\boldsymbol{\omega}_2 + \omega_3^a\hat{\mathbf{a}}_3 && (16) & \mathbf{v}_3 &= \mathcal{A}_3^2\left(\mathbf{v}_2 + \boldsymbol{\omega}_2^\times\mathbf{b}_{2,3}\right) && (19)
\end{aligned}$$

$$\begin{aligned}
\mathbf{P}_3 &= m_3\mathbf{v}_3 + \boldsymbol{\omega}_3^\times\mathbf{c}_3 && (20)\\
\mathbf{P}_2 &= m_2\mathbf{v}_2 + \boldsymbol{\omega}_2^\times\mathbf{c}_2 + \mathcal{A}_2^3\mathbf{P}_3 && (21)\\
\mathbf{P}_1 &= m_1\mathbf{v}_1 + \boldsymbol{\omega}_1^\times\mathbf{c}_1 + \mathcal{A}_1^2\mathbf{P}_2 && (22)
\end{aligned}$$

$$\begin{aligned}
\mathbf{H}_1 &= \mathbf{C}_1^\times\mathcal{A}_1^0\mathbf{v}_0 + \mathbf{J}_{1,0}\boldsymbol{\omega}_0 + \mathbf{J}_1\hat{\mathbf{a}}_1\omega_1^a + \mathbf{J}_{1,2}\hat{\mathbf{a}}_2\omega_2^a + \mathbf{J}_{1,3}\hat{\mathbf{a}}_3\omega_3^a && (23)\\
\mathbf{H}_2 &= \mathbf{C}_2^\times\mathcal{A}_2^0\mathbf{v}_0 + \mathbf{J}_{2,0}\boldsymbol{\omega}_0 + \mathbf{J}_{2,1}\hat{\mathbf{a}}_1\omega_1^a + \mathbf{J}_2\hat{\mathbf{a}}_2\omega_2^a + \mathbf{J}_{2,3}\hat{\mathbf{a}}_3\omega_3^a && (24)\\
\mathbf{H}_3 &= \mathbf{C}_3^\times\mathcal{A}_3^0\mathbf{v}_0 + \mathbf{J}_{3,0}\boldsymbol{\omega}_0 + \mathbf{J}_{3,1}\hat{\mathbf{a}}_1\omega_1^a + \mathbf{J}_{3,2}\hat{\mathbf{a}}_2\omega_2^a + \mathbf{J}_3\hat{\mathbf{a}}_3\omega_3^a && (25)
\end{aligned}$$

N-body Recursive Equations

Even from the simple four-body example, the pattern in the equations is evident. A generalized N-link set of equations can be generated from a compact and computationally efficient *recursive* collection of formulas,

$$\begin{aligned}
\mathbf{P}_0 &= M_0\cdot\mathbb{I}\mathbf{v}_0 - \mathbf{C}_0^\times\boldsymbol{\omega}_0 - \sum_{n=1}^{N}\mathcal{A}_0^n\mathbf{C}_n^\times\hat{\mathbf{a}}_n\omega_n^a && (26)\\
\mathbf{H}_0 &= \mathbf{C}_0^\times\mathbf{v}_0 + \mathbf{J}_0\boldsymbol{\omega}_0 + \sum_{n=1}^{N}\mathbf{J}_{0,n}\hat{\mathbf{a}}_n\omega_n^a && (27)\\
\mathbf{P}_n &= m_n\mathbf{v}_n + \boldsymbol{\omega}_n^\times\mathbf{c}_n + \mathcal{A}_n^{n+1}\mathbf{P}_{n+1} && (28)\\
\mathbf{H}_n &= \mathbf{C}_n^\times\mathcal{A}_n^0\mathbf{v}_0 + \mathbf{J}_{n,0}\boldsymbol{\omega}_0 + \sum_{m=1}^{N}\mathbf{J}_{n,m}\hat{\mathbf{a}}_m\omega_m^a \qquad \left(\text{where } \mathbf{J}_{n,n} \triangleq \mathbf{J}_n\right) && (29)
\end{aligned}$$

$$\begin{aligned}
\dot{\mathbf{P}} &= -\boldsymbol{\omega}_0^\times\mathbf{P} + \mathbf{F}_0^{\text{ext}} && (30)\\
\dot{\mathbf{H}}_0 &= -\boldsymbol{\omega}_0^\times\mathbf{H}_0 - \mathbf{v}_0^\times\mathbf{P} + \mathbf{G}_0^{\text{ext}} && (31)\\
\dot{H}_n^a &= \boldsymbol{\omega}_n^\intercal\hat{\mathbf{a}}_n^\times\mathbf{H}_n + \mathbf{v}_n^\intercal\hat{\mathbf{a}}_n^\times\mathbf{P}_n + \hat{\mathbf{a}}_n^\intercal\mathbf{g}_n + \hat{\mathbf{a}}_n^\intercal\mathbf{G}_n^{\text{ext}} && (32)
\end{aligned}$$

and,

$$\begin{aligned}
H_n^a &= \hat{\mathbf{a}}_n^\intercal\mathbf{H}_n && (33)\\
\boldsymbol{\omega}_n &= \mathcal{A}_n^{n-1}\boldsymbol{\omega}_{n-1} + \omega_n^a\hat{\mathbf{a}}_n && (34)\\
\mathbf{v}_n &= \mathcal{A}_n^{n-1}\left(\mathbf{v}_{n-1} + \boldsymbol{\omega}_{n-1}^\times\mathbf{b}_{n-1,n}\right) && (35)
\end{aligned}$$

where $n = 1, 2, \ldots, N-1, N$ (zeroth body is the "base" body/frame). All of the recursions begin at the base body and work outboard except for the composite linear momentum $(\mathbf{P}_n)$ which recurses inboard.

MASS PROPERTIES

Solution of the linear set of equations in Eq.(8) is needed in order to recover the rates used in calculating momentum state derivatives. This, in turn, requires computation of elements of the generalized mass matrix, $\mathcal{M}$, a non-trivial task. The following recursive equations provide a method for determining the quantities needed to populate the generalized mass matrix.

Composite Mass

The composite mass is a straight-forward summation of the individual link mass (m_n) with that of the outboard masses and which needs to be determined only once for fixed-mass systems.

$$M_n = \begin{cases} m_n & n = N \\ m_n + M_{n+1} & n = N-1, \ldots, 1, 0 \end{cases} \tag{36}$$

Composite First Mass Moment of Inertia

The first mass moment for a single body ($\mathbf{c}_n$) is simply the position vector to the n^{th} body's center of mass ($\mathbf{r}_{cm_n}$) scaled by the body's mass (m_n). The *composite* first mass moment is just the equivalent for the collection of bodies that includes the n^{th} and all outboard bodies. A recursive formula for generating the composite first mass moment—expressed in the n^{th} link frame—starts at the outer-most body as follows:

$$\mathbf{C}_n = \begin{cases} \mathbf{c}_n & n = N \\ \mathbf{c}_n + M_{n+1}\mathbf{b}_{n,n+1} + \mathcal{A}_n^{n+1}\mathbf{C}_{n+1} & n = N-1, \ldots, 1, 0 \end{cases} \tag{37}$$

Second and "Mixed" Mass Moments of Inertia

Computation of the composite second mass moments of the n^{th} and outboard bodies about the n^{th} body origin and expressed in the n^{th} frame from the mass properties of the individual bodies is facilitated by the following recursion relationship:

$$\mathbf{J}_n = \begin{cases} \mathbf{j}_n & n = N \\ \mathbf{j}_n + \mathcal{A}_n^{n+1}\mathbf{J}_{n+1}\mathcal{A}_{n+1}^n - (\mathbf{C}_n - \mathbf{c}_n)^\times \mathbf{b}_{n,n+1}^\times - \mathbf{b}_{n,n+1}^\times \mathcal{A}_n^{n+1}\mathbf{C}_{n+1}^\times \mathcal{A}_{n+1}^n & n = N-1, \ldots, 1, 0 \end{cases} \tag{38}$$

which was derived using the matrix identity $-\mathbf{u}^\times\mathbf{v}^\times = (\mathbf{u}^\intercal\mathbf{v})\mathbb{I} - \mathbf{v}\mathbf{u}^\intercal$. Using the same identity, one may write a compact expression of the *parallel axis theorem* for the second mass moment of inertia matrix of a rigid body about a point B given its inertia matrix about point A (in the same frame) expressed in matrix form as $\mathbf{j}_B = \mathbf{j}_A + m\cdot\mathbf{b}_{A,B}^\times\mathbf{b}_{B,A}^\times + \mathbf{b}_{A,B}^\times\mathbf{c}^\times - \mathbf{c}^\times\mathbf{b}_{B,A}^\times$, which may provide some insight regarding the form of Eq.(38).

The so called "mixed" moments of inertia ($\mathbf{J}_{m,n}$), as extended here, are derived from a construct of Hughes[8] that significantly simplifies the expressions of the composite momenta. Expressing succinctly the physical meaning of the mixed moments is not at first straightforward. However, it is clear that the mixed moments of inertia capture the coupling between the momentum of a given collection of bodies and the momentum of a subset of bodies outboard of a particular "base" body—a sort of *transformation* of the outboard system momentum into the "local" body-fixed frame. Similar to coordinate transformation matrices (e.g. $\mathcal{A}_m^n$), mixed mass moment matrices *span* multiple frames. Quantities that *right* multiply a mixed mass moment matrix, $\mathbf{J}_m^n$, need to be realized in the n-frame and vectors or matrices that *left* multiply need to be in the m-frame. One relationship for calculating the mixed mass moments is:

$$\mathbf{J}_{m,n} = \mathcal{A}_m^n\mathbf{J}_n - \mathbf{b}_{m,n}^\times\mathcal{A}_m^n\mathbf{C}_n^\times \tag{39}$$

The mixed mass moment is *not* symmetric and $\mathbf{J}_{m,n} = \mathbf{J}_{n,m}^\intercal$ can be shown to be true instead. Using Eq.(39), an alternate form of Eq.(38b) can be derived. The alternative and more computationally economical recursion formula for the second mass moment of inertia is

$$\mathbf{J}_n = \mathbf{j}_n + \mathbf{J}_{n,n+1}\mathcal{A}_{n+1}^n - (\mathbf{C}_n - \mathbf{c}_n)^\times\mathbf{b}_{n,n+1}^\times \qquad n = N-1, \ldots, 1, 0 \tag{40}$$

where it is evident that the mixed mass moment may also be thought of as a "portion" of the second mass moment.

FORCES AND MOMENTS

The dynamical model permits general application of forces or moments to the system. The treatment of the force or moment differs slightly if it is from an *internal* or *external* source. The appropriate application for the two cases is discussed in the subsections that follow.

Internal Disturbances

At the total system level $(\mathbf{P}_0, \mathbf{H}_0)$, internal forces and moments are absent from the linear and angular momentum equations of motion, due to the action-reaction cancellation (Newton's Third Law). However, at the subsystem level $(\mathbf{P}_n, \mathbf{H}_n)$, internal forces and torques can appear as in Eqs.(11-13).

Specifically for a serial chain of revolutes, only the internal torques, $\mathbf{g}_n$, matter (and actually only the scalar projection along the joint's axis of revolution) since there are no translational and only one rotational degree-of-freedom for the n^{th} subsystem relative to the $(n-1)^{th}$ subsystem. This *inter-body* torque might represent joint friction or a motor torque. The salient feature is that it acts on the n^{th} body (through the n^{th} joint) and *must* have a corresponding equal-and-opposite torque acting on the $(n-1)^{th}$ body. Conceptually, the composite system of masses—that includes the n^{th} body and all outboard bodies—experiences the torque from the $(n-1)^{th}$ body as if it were an external torque, since its source (body $n-1$) is external to that particular *subsystem*. Explicitly, $\mathbf{g}_n$ is the torque on the n^{th} body, expressed in the n-frame, due to the $(n-1)^{th}$ body. Moving inboard to the $(n-1)^{th}$ subsystem calculations, the torque $\mathbf{g}_n$ now becomes a wholly internal action-reaction and so is absent from the state derivative equation for the composite angular momentum.

For a corresponding treatment of an internal *force*, refer to the example in the section "A System with Damping" in reference [8].

External Disturbances

External disturbances may be applied at any location and on any body. In Figure 1 an external force ($\underset{\rightarrow}{\mathbf{f}}_3^{\text{ext}}$) and external torque ($\underset{\rightarrow}{\mathbf{g}}_3^{\text{ext}}$) is shown being applied to the last link of the kinematic chain—such as might be experienced by an end-effector of a robotic manipulator as it tries to grapple a target—but, as in the case of gravity, could also be experienced independently by all of the bodies. To capture the cumulative effect simply, we have chosen to define *composite* forces ($\underset{\rightarrow}{\mathbf{F}}_n^{\text{ext}}$) and torques ($\underset{\rightarrow}{\mathbf{G}}_n^{\text{ext}}$) that propagate in-board via the recursion relations

$$\mathbf{F}_n^{\text{ext}} = \begin{cases} \mathbf{f}_n^{\text{ext}} & n = N \\ \mathbf{f}_n^{\text{ext}} + \mathcal{A}_n^{n+1}\mathbf{F}_{n+1}^{\text{ext}} & n = N-1, \ldots, 1, 0 \end{cases} \tag{41}$$

$$\mathbf{G}_n^{\text{ext}} = \begin{cases} \mathbf{g}_n^{\text{ext}} + \mathbf{r}_{f_n}^{\times}\mathbf{f}_n^{\text{ext}} & n = N \\ \mathbf{g}_n^{\text{ext}} + \mathbf{r}_{f_n}^{\times}\mathbf{f}_n^{\text{ext}} + \mathcal{A}_n^{n+1}\mathbf{G}_{n+1}^{\text{ext}} + \mathbf{b}_{n,n+1}^{\times}\mathcal{A}_n^{n+1}\mathbf{F}_{n+1}^{\text{ext}} & n = N-1, \ldots, 1, 0 \end{cases} \tag{42}$$

One important point worth mentioning is regarding the *point of resolution* of external disturbances. Specifically, $\underset{\rightarrow}{\mathbf{G}}_n^{\text{ext}}$ is defined to be the *total* moment acting on the n^{th} body, about the n^{th} frame origin, and expressed in the n^{th} frame. If the external force acting on the n^{th} body acts at some point other than the origin, as in Figure 1, the resulting moment due to the offset point of application ($\underset{\rightarrow}{\mathbf{r}}_f \times \underset{\rightarrow}{\mathbf{f}}_n$) must be added to any *pure* external moment couple ($\underset{\rightarrow}{\mathbf{g}}_n^{\text{ext}}$) also acting on the individual body, in order to compute the total external moment *about the n^{th} body origin*, $\underset{\rightarrow}{\mathbf{G}}_n^{\text{ext}}$.

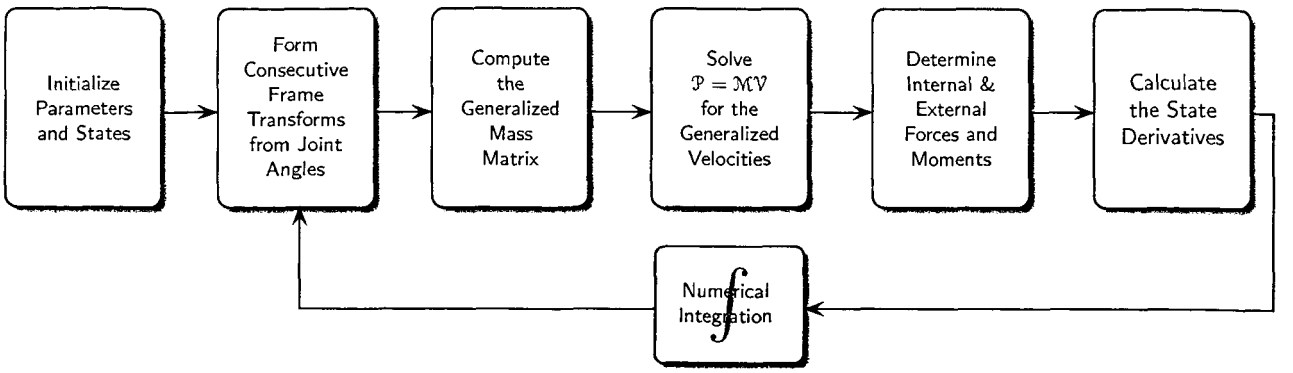

Figure 2: Dynamic Simulation Procedure

DYNAMIC SIMULATION

The general procedure for simulating a spacecraft using momentum state dynamics involves at least one unconventional step—the recovery of rates. Additionally, the requirements for a minimal spacecraft simulation are states representing the vehicle position ($\mathbf{R}_0$), attitude ($\mathbf{q}_0^i$) and joint angles ($\boldsymbol{\theta}$). Figure 2 depicts the procedure necessary to dynamically simulate the behavior of the multi-body system.

An additional computation that may aid analysis during initialization and post-processing is determining velocity with respect to an inertial frame of the composite center of mass for the entire system (expressed in the "base" body-fixed frame). It may be calculated using the following relationship:

$$\mathbf{v}_{cm_0} = \mathbf{v}_0 + \frac{1}{M_0}\left\{\boldsymbol{\omega}_0^{\times}\mathbf{C}_0 + \sum_{n=1}^{N}\omega_n^a \mathcal{A}_0^n \hat{\mathbf{a}}_n^{\times}\mathbf{c}_n\right\} \tag{43}$$

Tracking the total kinetic energy of the system is also straightforward, given as $KE = \frac{1}{2}\mathcal{V}^{\mathsf{T}}\mathcal{M}\mathcal{V}$.

The dynamic formulation exists as an ANSI C code module in GSFC's real-time *Freespace* simulation and analysis environment. As a validation and performance comparison, a version of the four-body model depicted in Figure 1 is tested against a similar model implemented in the commercial, general-purpose, multi-body simulation software package LMS/DADS© (v9.6). After one thousand seconds of simulated forced motion, the system rates (shown in Figure 3) differed by a maximum of 5×10^{-6}. The system parameters for the cross-validation test case are given in Table 2. Units are in MKS and degrees per second. Recall that $\mathbf{I}_n$ is the second mass moment of inertia about the body center of mass, as opposed to $\mathbf{j}_n$ that is about the body frame origin. All other configuration parameters not explicitly stated in Table 2 are zero.

Table 2: Parameters for Test Simulation

$\hat{\mathbf{a}}_{0,1,2,3}$	=	$[0\ \ 0\ \ 1]^{\mathsf{T}}$	$(\mathbf{I}_{xx})_0$	=	21261	$m_{1,2,3}$	=	200	
$\mathbf{b}_{0,1}$	=	$[-1.524\ \ 0\ \ 2.5]^{\mathsf{T}}$	$(\mathbf{I}_{xx})_{1,2,3}$	=	2.25	$\mathbf{q}_1^0$	=	$\left[0\ \ 0\ \ -\frac{\sqrt{2}}{2}\ \ \frac{\sqrt{2}}{2}\right]^{\mathsf{T}}$	
$\mathbf{b}_{1,2}$	=	$[5\ \ 0\ \ 0]^{\mathsf{T}}$	$(\mathbf{I}_{yy,zz})_0$	=	30730	$\mathbf{q}_{n+1}^n$	=	$[0\ \ 0\ \ 0\ \ 1]^{\mathsf{T}}$	$n = 1,2,3$
$\mathbf{b}_{2,3}$	=	$[5\ \ 0\ \ 0]^{\mathsf{T}}$	$(\mathbf{I}_{yy,zz})_{1,2,3}$	=	417.917	ω_1^a	=	-0.01	
$\mathbf{c}_{1,2}$	=	$[1250\ \ 0\ \ 0]^{\mathsf{T}}$	m_0	=	10158	$\mathbf{f}_3^{\text{ext}}$	=	$[0\ \ 0.1\ \ 0]^{\mathsf{T}}$	

In addition to executing more than four times quicker than the DADS© four-body model (which is to be expected since the present model is not a general purpose solver), there were also some *alternate* test

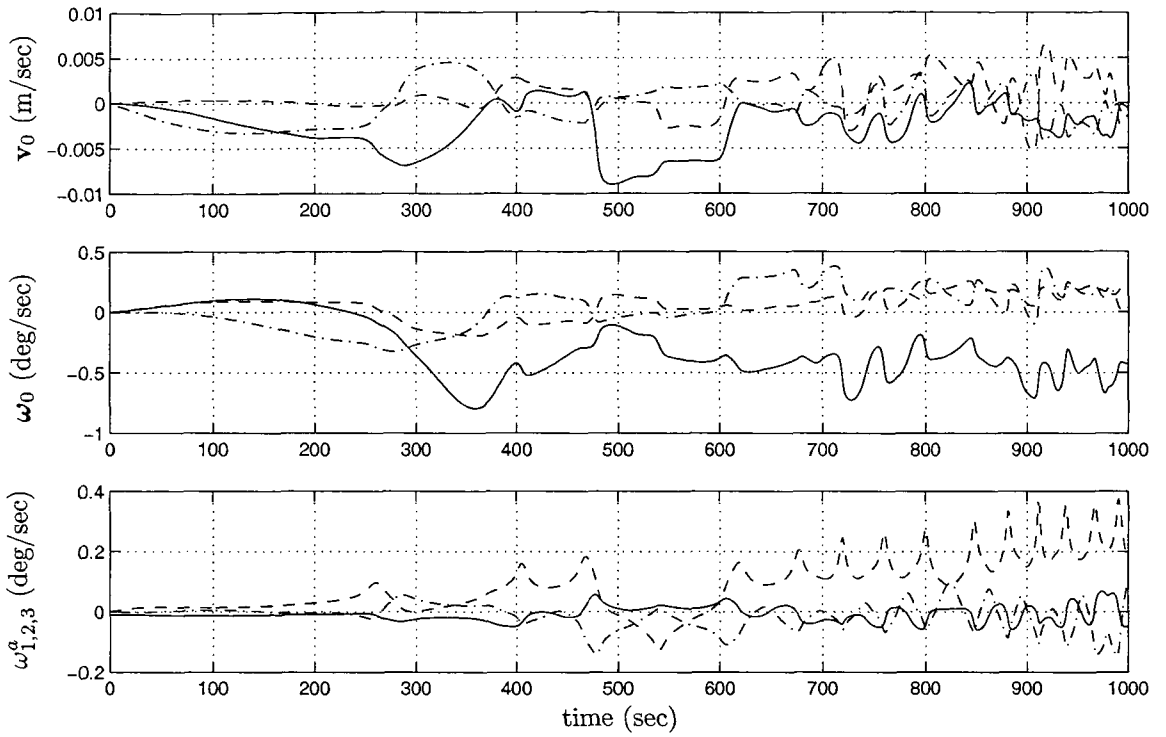

Figure 3: Validated Simulation Rates for Base Body with Three Links

cases with higher initial body rates for which the DADS© solvers reported constraint tolerance violations—after which the two solutions diverged. Recalling some of the discussions in the introduction, specifically regarding general purpose solvers and the added complexity of the over-determined solution methods (i.e. ODE + DAE), it is likely that the present method is also more stable for this class of models.

APPLICATION

The dynamics model described was developed for the Hubble Robotic Servicing and Deorbit Mission's (HRSDM) design, analysis and testing effort. The mission consisted of an autonomous grapple of the Hubble Space Telescope by a chaser vehicle using a six degrees-of-freedom robotic manipulator, similar to the RMS on the Space Shuttle (Figure 4). Additionally, post-grapple mission operations involved attaching a dexterous, two-armed robot (seven degrees-of-freedom per arm) to the end of the grapple arm and using the combined manipulators and tools to perform complex servicing operations. The design for the flight systems (robots and controller) was the responsibility of McDonald-Detwiler Associates, Canada. However, the present model was implemented independently in a real-time dynamic simulation for validation and planning of the grapple and servicing. The open kinematic chain consisted of up to thirteen linkages connected in series with a non-flight joint controller for directing the robotic maneuvers. Some details of the manner in which a controller can benefit from the form of the dynamics will now be discussed.

Joint Control

The control laws used in the simulation to control the robot arms are restricted to *joint-space control.* The calculated control is applied directly but with saturation levels. There are several control modes in the simulation and in each case, the calculated torque is scaled by the outboard inertia about the axis of each joint. An actuator motor control torque is added to the joint dynamics so that Eq.(32) becomes:

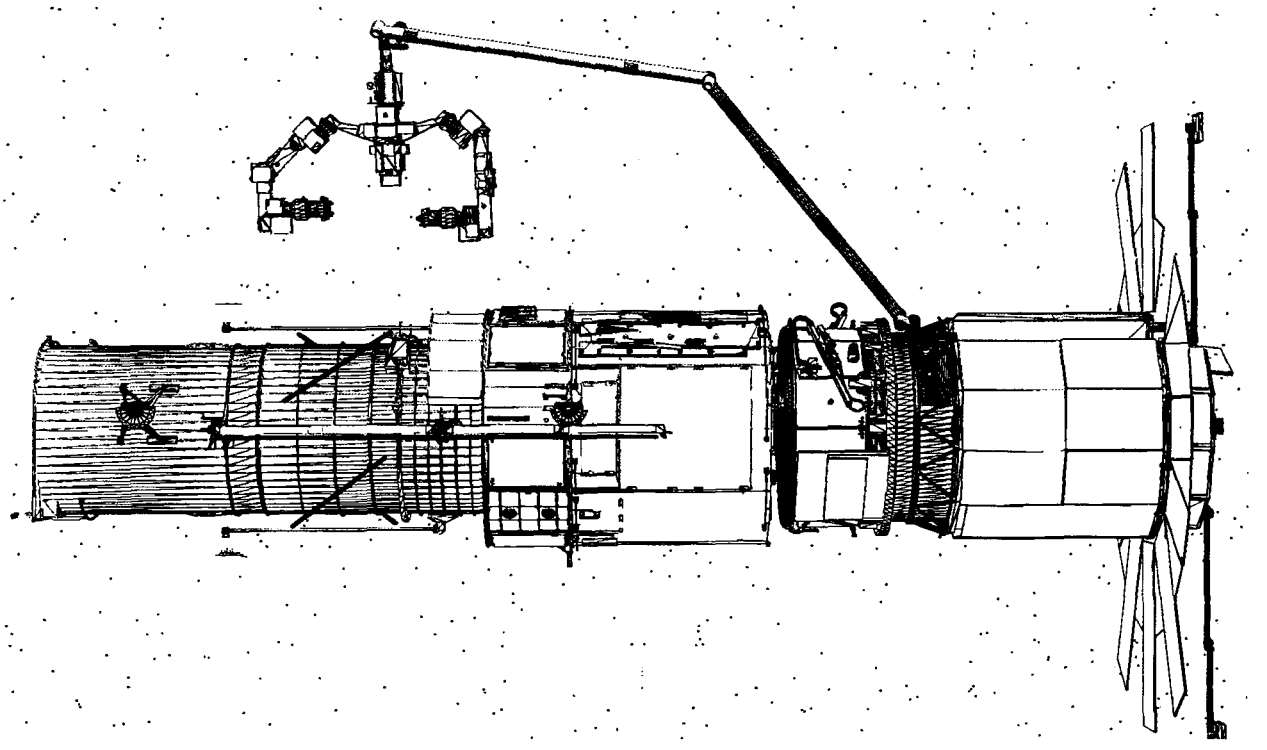

Figure 4: Hubble Robotic Servicing and Deorbit Mission with Grapple Arm and Dexterous Robot

$$\dot{H}_n^a = \omega_n^{\mathsf{T}} \hat{\mathbf{a}}_n^{\times} \mathbf{H}_n + \mathbf{v}_n^{\mathsf{T}} \hat{\mathbf{a}}_n^{\times} \mathbf{P}_n + \hat{\mathbf{a}}_n^{\mathsf{T}} \mathbf{g}_n + \hat{\mathbf{a}}_n^{\mathsf{T}} \mathbf{G}_n^{\text{ext}} + g_n^a \tag{44}$$

where g_n^a is governed by a proportional, derivative and feedforward control law expressible as,

$$g_n^a = \tilde{g}_n^a - (K_\theta)_n \cdot J_n^a \left(\tilde{\theta}_n - \theta_n\right) - (K_\omega)_n \cdot J_n^a \left(\tilde{\omega}_n^a - \omega_n^a\right) \tag{45}$$

Here, θ_n is the joint angle, $(K_\theta)_n$ and $(K_\omega)_n$ are the proportional and velocity control gains respectively, and $J_n^a = \hat{\mathbf{a}}_n^{\mathsf{T}} \mathbf{J}_n \hat{\mathbf{a}}_n$ refers to the composite second mass moment of inertia of the n^{th} body and all bodies outboard, about the n^{th} joint axis of rotation. This inertia term is clearly time-varying as the dynamics of the robot arm evolve. Note that for this controller, the gyroscopic coupling and external disturbances, such as gravity gradient, are not taken into account by the torque control law. At this point the control torque computed in Eq.(45) is sufficient since the motion of the grapple arm and/or dexterous arm is restricted by very low joint rate limits intended to keep base body disturbances small. The gyroscopic coupling terms in Eq.(44) are second order terms and are neglected. The gravity gradient terms are also very small and are effectively compensated for by the control torques.

The terms, $\tilde{g}_n^a$, $\tilde{\theta}_n$ and $\tilde{\omega}_n^a$, denote desired or commanded quantities, and their determination depends on the control mode chosen for a particular task. For the HRSDM effort, five major control modes were implemented. They include: *Static* (Regulator) Control, *Time-Optimal Path* Control, *Manual* (man-in-the-loop) Control, *Individual Joint* Control, and *Scripted Sequence* Control.

COMPUTATIONAL CONSIDERATIONS

Efficiency

In 1980, computational schemes for solving robot dynamics underwent a "quantum-leap" in efficiency with the introduction of a recursive formulation for the serial revolute manipulator by the MIT team of Luh, Walker and Paul[10]. Their new algorithm for inverse dynamics relied on computations that grow linearly

with the number of links, commonly referred to as Order-N or, more simply, $\mathcal{O}(N)$ algorithms. Two years later, Walker and Orin[3] constructed an efficient *forward* dynamics solution using the inverse dynamics method of Luh et al[10]. Walker and Orin's most efficient implementation (Method 3) exploited symmetry of the mass matrix and a recursive calculation of the mass properties to achieve solutions that were 100 times more efficient[5] than the established Euler-Lagrange methods for $N = 6$.

The Walker and Orin–Method 3 (WO-3) algorithm is, like the present momentum state dynamics (MSD) method, $\mathcal{O}(n^3)$ in complexity since both rely upon a numerical solution to a linear system of equations via Cholesky decomposition—which requires approximately $\frac{1}{6}N^3$ operations. Since the early 1980's, there have been several advances in $\mathcal{O}(N)$[7] and even $\mathcal{O}(\log N)$ techniques. However, as shown in Featherstone[7], the $\mathcal{O}(n^3)$ algorithms still tend to be faster for $N < 10$, the class of problem in which most aerospace applications fall. As a result, the WO-3 benchmark is still a valid measure of a new formulation's numerical efficiency. Table 3 provides a detailed comparison of the floating-point operations required for the two methods. Table 4 is a more detailed profile of the present algorithm's computational load.

Table 3: Comparison of Number of Computations to Solve Forward Dynamics

Method	Multiplications	Additions	Bodies	Links	Mult	Add
WO-3	$\frac{1}{6}N^3 + 13\frac{1}{2}N^2 + 192\frac{1}{3}N - 49$	$\frac{1}{6}N^3 + 8\frac{1}{2}N^2 + 165\frac{5}{6}N - 64$	6	6	1627	1255
MSD	$\frac{1}{6}N^3 + 14N^2 + 224N - 30$	$\frac{1}{6}N^3 + 11N^2 + 175N - 18$	6	5	1461	1153
			7	6	1854	1464

There is some ambiguity in comparison to the classic Walker and Orin–Method 3 algorithm. The present method provides a coupled translational and rotational motion solution. In contrast, the WO-3 method is specific to a fixed-base system. The former is more amenable to spacecraft simulations, whereas the latter is specific to terrestrial robotics. Depending on whether the number of *bodies* (base + links) or just the number of *links* is counted for comparison, for $N = 6$ the current method is either 10% faster or 15% slower than the WO-3 algorithm. Nevertheless, the momentum state dynamics method for serial manipulators is competitive with one of the best $\mathcal{O}(N^3)$ methods.

From the algorithm profile in Table 4, it is clear that—apart from linear system solution—calculating the generalized mass matrix is the most computationally expensive operation. In fact, in order to improve the efficiency of the algorithm as N grows, additional intermediate quantities are introduced and a portion of the $\mathcal{O}(N^2)$ operations were reformulated to have only $\mathcal{O}(N)$ dependencies. Specifically when calculating the "mixed" mass moments of inertia, both the $N-1$ off-diagonal terms of the generalized mass matrix ($\mathbf{J}_{0,1}, \mathbf{J}_{1,2}, \ldots \mathbf{J}_{n,n+1}$) and those of rows 4-6 ($\mathbf{J}_{0,1}, \mathbf{J}_{0,2} \ldots \mathbf{J}_{0,n}$), are calculated explicitly using the second mass moment as given in Eq.(39). However, for the remaining set of "mixed" moments of inertia, whose members grow as $\frac{1}{2}(N-2)(N-1)$, two new recursion relationships are required as follows:

Calculation (13 links)	**millisecs**	%
Mass Matrix Assembly	0.042	24%
Linear System Solution (Cholesky Decomposition)	0.111	65%
State Derivative Dynamics	0.018	11%
Total	0.171	100%

Table 4: Algorithm Component Run-times (MIPS R16000, 1 GHz Processor)

$$\mathbf{C}^{\times a}_{m,n} = \begin{cases} \mathcal{A}^n_m \mathbf{C}^{\times}_n \hat{\mathbf{a}}_n & m = n-1 \\ \mathcal{A}^{m+1}_m \mathbf{C}^{\times a}_{m+1,n} & m \neq n-1 \end{cases} \tag{46}$$

$$\mathbf{J}^a_{m,n} = \mathcal{A}^{m+1}_m \mathbf{J}^a_{m+1,n} - \mathbf{b}^{\times}_{m,m+1} \mathbf{C}^{\times a}_{m,n} \tag{47}$$

where $\mathbf{J}^a_{m,n} = \mathbf{J}_{m,n}\hat{\mathbf{a}}_n$. Efficiency is gained in the $\mathcal{O}(N^2)$ operations by propagating 3×1 column matrices in the place of 3×3's and also by eliminating many intermediate frame transformation constructions (e.g. $\mathcal{A}^4_1, \mathcal{A}^3_1, \mathcal{A}^4_2, \ldots$). Another notable feature of Eqs.(46-47) is that the generalized mass matrix $\mathcal{O}(N^2)$ operations are column-wise independent and are therefore candidates for parallel processing.

Precision

Two issues were discovered during the course of implementing the numerical simulation of the spacecraft and appendages. The first involves the propagation of the vehicle position and the second, round-off errors.

Numerical integrators, such as an explicit Runge-Kutta method, propagate by perturbing the system states by relative amounts. If some states describe vehicle attitude (e.g. a quaternion that transforms from inertial to body coordinates) and they are perturbed, all quantities in body coordinates are also perturbed due to cross-coupling. While the present form of the equations requires that the dynamics be realized in body-fixed coordinates, experience has shown it is not beneficial to propagate the base-frame position states ($\mathbf{R}_0$) in body coordinates as well. Instead, an inertial frame such as J2000 should be used. This prevents unnecessary step rejection due to the variations in attitude states coupling into perturbations of the orbit. By de-coupling the position and using double precision calculations, both an accurate attitude/appendage dynamics solution and a stable orbit can be achieved for low earth orbiting (LEO) spacecraft without resorting to quadruple precision, non-dimensionalized variables[13] or Clohessy-Wiltshire approximations.

The final numerical issue pertains to round-off precision. If one considers the equation for a single rigid-body spacecraft in LEO, the equations that propagate the angular momentum can be written as:

$$\dot{\mathbf{H}}_0 = -\boldsymbol{\omega}_0^\times \mathbf{H}_0 - \mathbf{v}_0^\times \mathbf{P}_0 + \mathbf{G}_0^{\text{ext}} \tag{48}$$

However, if the spacecraft mass on the order of 10^4 and the orbital velocity magnitude is of order 10^4 then a simple numerical experiment that calculates $\mathbf{v}_0^\times M_0 \mathbf{v}_0$ using 16 significant figures of precision can yield round-off residuals with magnitudes of $\sim 10^{-5}$, which may not be acceptable for a particular application. An alternate form of angular momentum propagation that explicitly cancels this term is:

$$\dot{\mathbf{H}}_0 = -\boldsymbol{\omega}_0^\times \mathbf{H}_0 + \mathbf{v}_0^\times \mathbf{C}_0^\times \boldsymbol{\omega}_0 + \mathbf{G}_0^{\text{ext}} \tag{49}$$

which experience has shown to be well behaved for high-precision solution tolerances.

CONCLUSIONS

A new formulation is given for calculating the system dynamics of a free-floating "base" body with an attached appendage consisting of rigid linkages interconnected, chain-like, by a series of single degree-of-freedom rotational joints. Presented as an extension to the momentum state dynamics framework of P. C. Hughes[8], the present work adds robotic manipulators to the existing suite of multi-body models well suited for spacecraft applications such as reaction wheels, nutation dampers, control moment gyros, and fuel slosh. A compact and efficient recursive relationship allows for flexible simulation implementation of an arbitrary number of links in the chain and is suitable for real-time applications.

The computational complexity of the solution method is $\mathcal{O}(N^3)$, and is shown to be competitive with the best algorithms of this class. However, opportunity exists for additional performance optimization such as using parallel computation methods for both the assembly of the generalized mass matrix and the solution of the linear set of momentum equations. Additional research might also suggest a method of avoiding the matrix decomposition altogether, yielding a lower order dependency on the number of bodies in the system.

Finally, topics of future publications will most likely include extensions of the method to other system topologies (branches, closed-loops, geared joints, and link flexibility) as well as explication on the method that can be used to recover joint constraint forces and moments.

References

[1] Uicker, J. J., "On the Dynamic Analysis of Spacial Linkages Using 4 by 4 Matrices", Ph.D. Thesis, Department of Mechanical and Astronautical Sciences, Northwestern University, (1965).

[2] Kane, T. R. and Levinson, D. A., "The Use of Kane's Dynamical Equations in Robotics", *The Int. Journal of Robotics Res. 2*, No. 3, pp. 3-21 (1983).

[3] Walker, M. W., Orin, D. E., "Efficient Dynamic Computer Simulation of Robotic Mechanisms", *Journal of Dynamic Systems, Measurements, and Control*, ASME, Vol. 104, pp. 205-211 (1982).

[4] Silver, W. M., "On the Equivalence of Lagrangian and Newton-Euler Dynamics for Manipulators", *The Int. Journal of Robotics Res. 1*, No. 2, pp. 118-128 (1982).

[5] Craig, J. J., *Introduction to Robotics: Mechanics and Control*, 2^{nd} ed., Addison-Wesley, (1989).

[6] Shabana, A. A., *Computational Dynamics*, 2^{nd} ed., Wiley & Sons, (2001).

[7] Featherstone, R., *Robot Dynamics Algorithms*, Kluwer, (1987).

[8] Hughes, Peter C., *Spacecraft Attitude Dynamics*, Dover, (2004).

[9] Ho, J. Y. L., "Direct Path Method for Flexible Multibody Spacecraft Dynamics", *Journal of Spacecraft and Rockets*, Vol. 14, pp. 102-110 (1977).

[10] Luh, J. Y. S., Walker, M. W., and Paul, R. P. C., "On Line Computational Scheme for Mechanical Manipulators", *Journal of Dynamic Systems, Measurements, and Control*, ASME, Vol. 102, pp. 69-76 (1980).

[11] Ford, K. A., and Hall, C. D., "Flexible Spacecraft Reorientations Using Gimbaled Momentum Wheels", *The Journal of the Astronautical Sciences*, ASME, Vol. 49, No. 3, pp. 421-441 (2001).

[12] Denavit, J. and Hartenberg, R. S., "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices", *Journal of Applied Mechanics*, June, pp. 215-221 (1955).

[13] Santini, P. and Gasbarri, P., "Dynamics of Multibody System in Space Environment: Lagrangian vs. Eulerian Approach", *Acta Astronautica*, No. 54, pp. 1-24 (2003).