To:	Distribution
From:	Eric Stoneking
Date:	Oct 2021
Subject:	Implementation of Order-N Multibody Dynamics

1 Nomenclature

j, i, o, c	Indices used for bodies
k, N, d	Indices used for joints
U	Identity matrix
N	The inertial reference frame
B_j	The jth body
G_k	The kth joint
G_N	The "inertial" joint, between the inertial frame and B_1
ω	Angular velocity (3×1)
v	Translational velocity (3×1) (overloaded notation)
r	Relative position vector
v	Spatial velocity (6×1) (overloaded notation)
u_k	Generalized speeds associated with joint G_k $(N_u \times 1)$
$P_{\omega k}$	Partial angular velocities for G_k
P_{vk}	Partial velocities for G_k
σ_k	Rotational generalized speeds for G_k
s_k	Translational generalized speeds for G_k
θ_k	Rotational kinematic states for G_k
x_k	Translational kinematic states for G_k
α_r	Remainder angular acceleration
a_r	Remainder acceleration
a_u	Acceleration due to $P\dot{u}$
I_j	Central inertia matrix for B_j
m_j	Mass matrix for B_j
M_j	Mass matrix of B_j
R_j	Spatial Force
F_j	Total (Spatial) Force
F^*_{rj}	Remainder Inertia Force
$\overline{M}_k$	Articulated body mass about G_k
$\overline{F}_k$	Articulated body force about G_k

P_k	Joint partial matrix for joint G_k $(6 \times N_{uk})$
D_k	Dynamics matrix for joint G_k
A_k	Absorption matrix for joint G_k
T_k	Transmission matrix for joint G_k
${}^{B}C^{A}$	Direction cosine matrix from A to B
${}^{B}\overline{C}^{A}$	"Spatial" direction cosine matrix from A to B
$S_v(r)$	Velocity shift operator
$S_F(r)$	Force shift operator

2 Introduction

Multibody dynamics leads naturally to a system of differential equations to be solved, linear in the variables to be integrated. Accuracy and efficiency are two metrics we can use to compare approaches for solving these systems. Kane's method strives for both by solving a minimum-dimension system of equations. When solving by standard matrix methods (Gaussian elimination and back substitution), the computational effort grows as the cube of the number of variables (written $O(N^3)$), so minimizing the dimension of the problem can result in significant time savings. But there is another technique that can go even farther. *Order-N* dynamics, as the name implies, is set up so the computational effort grows linearly with the number of bodies. For problems with more than a few bodies, the time savings over the $O(N^3)$ solution is dramatic. Kane's method is still the basis for it, so the accuracy and minimum-dimension features are still obtained. The Order-N technique avoids assembling the whole system of equations, though. Instead, it solves the problem piecemeal, one joint at a time. (Those who remember solving trusses by the method of joints may see a comparison there.)

There are papers presenting Order-N dynamics dating back to the 1980's, but this author found them unreadable. I only got traction on the problem when I found Jain's book [1]. Readers new to the topic may find Featherstone [2] a more approachable introduction. This memo documents my adaptation of Jain's methodology to my own habitual style.

My emphasis in this memo is on implementation. I'll only introduce enough theory to support that end. I do recommend the interested reader spend some time with the references.

3 Foundations

3.1 Basis-Free and Basis-Loaded

In implementation, we will need to properly account for expressing vector quantities in reference frames when we combine them. For instance, to find the resultant torque of a set of forces, we would need to account for what point we are taking moments about, and would need to express the forces in a common frame to add them. For the first pass through the derivation, we will assume that accounting is taken care of, so that we can write expressions like $F_1 + F_2$ without worrying about what reference points or frames are involved. We'll then work through the required transformation operators as we get into the basis-loaded implementation.

3.2 Spatial Vectors

Spatial vectors combine rotational and translational quantities, accounting for all the potential DOF of a rigid body in a unified formalism. The advantage is compact notation. The disadvantage is more complicated addition, multiplication, and transformation operators.

Spatial velocity combines angular and translational velocity:

$$v = \left\{ \begin{array}{c} \omega \\ v \end{array} \right\} \tag{1}$$

Partial velocity:

$$P = \left\{ \begin{array}{c} \Omega \\ V \end{array} \right\} \tag{2}$$

Mass matrix combines moments of inertia and mass in a 6×6 matrix:

$$M = \left[\begin{array}{cc} I & 0 \\ 0 & mU \end{array} \right] \tag{3}$$

Spatial force combines torque and force:

$$R = \left\{ \begin{array}{c} T \\ F \end{array} \right\} \tag{4}$$

We use R for spatial force because we want to reserve F for *total force*, i.e. the sum of applied forces and remainder inertia forces.

3.3 Kane's Equations

For a multibody system, we could write the equations of motion in terms of Newton's Second Law:

$$[M]\{\dot{v}\} = \{F\} \tag{5}$$

For a system of N_b bodies, there are $6N_b$ equations, including coupled equations of motion and equations of constraint. Kane's method uses the *partial angular velocities* Ω and *partial velocities* V to project the system of equations into the subspace spanned by the independent *generalized speeds*, u. This projection discards the equations of constraint, yielding a smaller system of equations to solve.

Using separate Ω and V, we write Kane's equations in the form

$$(\Omega^T I \Omega + V^T m V)\dot{u} = \Omega^T (T - I\alpha_r - \omega \times H) + V^T (F - ma_r) \tag{6}$$

Using spatial vectors, we can write it more compactly as

$$P^T M P \dot{u} = P^T F \tag{7}$$

where

$$F = \left\{ \begin{array}{c} T \\ F \end{array} \right\} + \left\{ \begin{array}{c} -I\alpha_r - \omega \times H \\ -ma_r \end{array} \right\} \tag{8}$$

3.4 Motion Space and Constraint Space

For a system of N_b rigid bodies, the $6N_b$ space of equations may be partitioned into two mutually orthogonal subspaces: the *motion space* and the *constraint space*. All motion occurs in the motion space. No motion occurs in the constraint space. Kane's method uses the partial velocities to project the $6N_b$ equations of motion into the motion space spanned by the generalized speeds (u). The benefit is a minimum-dimension set of equations to solve, and avoiding the numerical complexities of solving for constraints.

Writing $P^T M P \dot{u} = P^T F$ gives an N_u system of equations in the form $A\dot{u} = b$ to solve for $\dot{u}$. This can be solved using standard Gaussian elimination and back substitution. As the system grows in size, however, the computational time grows as the cube of N_u. The Order-N procedure presented here attacks the problem piecemeal, avoiding assembling the whole N_u system.

3.5 The Tree

We consider a system composed of multiple rigid bodies connected by joints in a tree topology (no loops). We designate one body as the *root*, *base*, or *main* body (we'll use these terms interchangeably). From the base, we can traverse the tree through the connecting joints until we reach the *tip* or *terminal* bodies. Each joint connects two bodies. The body closer to the root is that joint's *inner* body and the one farther from the root is that joint's *outer* body. We number the bodies $1, 2, ..., N_b$, with the root being B_1. We require that the inner body of each joint have a lower index than its outer body. We number the joints $1, 2, ..., N_g$, with $N_g = N_b - 1$ for a tree. We also have a "zeroth joint" representing the six DOF between inertial space and the root body. In some respects, this is a joint like any other, but in other ways it is special. We will denote it as G_N when we group it with the other joints.

P_k is the $6 \times N_{uk}$ matrix of joint partials for the joint G_k. In principle, P_k could be a full matrix. In practice, however, we do not expect to encounter any joints where a single DOF has both rotational and translational effects (e.g. helical screws). So we will partition $P_k u_k$ into

$$P_k u_k = \begin{bmatrix} P_{\omega k} & 0 \\ 0 & P_{vk} \end{bmatrix} \left\{ \begin{array}{c} u_{\omega k} \\ u_{vk} \end{array} \right\} = \left\{ \begin{array}{c} P_{\omega k}\sigma_k \\ P_{vk} s_k \end{array} \right\} \tag{9}$$

Referring back to notation I've used in earlier formulations, $P_{\omega k} = \Gamma_k$ and $P_{vk} = \Delta_k$.

We will traverse the tree three times. The first time will be from base to tip, recursively building quantities such as velocities. The second traversal will be from tip to base, recursively agglomerating masses and forces. The final traversal will be from base to tip, finding each $\dot{u}$ in turn. Speaking in general terms, we may say that a base-to-tip traversal is *scattering* some quantity, and a tip-to-base traversal is *gathering* something.

During the mass-gathering (tip-to-base) traversal, we will examine each joint in turn. A key concept for this procedure is that a joint splits the gathered mass and force terms into their motion-space and constraint-space components, based on the degrees of freedom of that joint. Any mass or force terms that lie in the motion space of that joint are said to be *absorbed* by the joint; they are active in that joint's equation of motion. Any mass or force terms that lie in the constraint space of that joint are said to be *transmitted* through the joint, to be dealt with down the line. Part of the procedure will be to find the absorption and transmission operator matrices for each joint.

3.6 Joint Relations

For conciseness, we introduce $\sigma_k = u_{\omega k},\ s_k = u_{vk}$ as the subparts of the generalized speeds associated with joint G_k. Also, the translational kinematic variables appear: x_k. The rotational kinematic variables, θ_k, don't appear directly in the equations of motion, but the joint partials do depend on them. (For a spherical joint, $P_\omega = U$.) The velocity relations are:

$$\begin{aligned} \omega_o &= \omega_i + P_{\omega k}\sigma_k && (10)\\ v_o &= v_i + P_{vk}s_k + \omega_i \times (r_{ik} + P_{vk}x_k) - \omega_o \times r_{ok} && (11) \end{aligned}$$

We have chosen to put the translational DOF of the joint on the "inboard" side of the rotational DOF. This is a simplification from the perfectly general case; it makes P_{vk} constant, and conveniently expressed in B_i. We'll see the grouping $(r_{ik} + P_{vk}x_k)$ show up frequently as a consequence.

$P_{\omega k}$ is naturally expressed in B_o, so when we add bases, we'll see some direction cosine matrices appear embedded in P_k.

Note that ω_o appears in the right hand side of the second equation, which would be undesirable if we wanted to express these relations in spatial-vector form. We aren't motivated to express them as such, though, since the three-vector form is clear and economical in implementation. We calculate ω_o first, so using it in the v_o equation presents no difficulties.

Differentiating, we obtain the acceleration relations:

$$\begin{aligned} \alpha_o &= \alpha_i + P_{\omega k}\dot{\sigma}_k + \dot{P}_{\omega k}\sigma_k + \omega_o \times P_{\omega k}\sigma_k && (12)\\ a_o &= a_i + P_{vk}\dot{s}_k + 2\omega_i \times P_{vk}s_k \\ &\quad + \alpha_i \times (r_{ik} + P_{vk}x_k) + \omega_i \times \omega_i \times (r_{ik} + P_{vk}x_k) \\ &\quad - \alpha_o \times r_{ok} - \omega_o \times \omega_o \times r_{ok} && (13) \end{aligned}$$

Note that we assume joints in which P_v is constant, so $\dot{P}_v$ is always zero.

We will split these accelerations into two terms, a_u and a_r. The a_u term for a body B_j is the acceleration due to rates of change of the generalized speeds: $a_{uj} = P_j\dot{u}$. The a_r term is everything else, i.e. the remainder accelerations.

G_N is the joint between the inertial frame N and the root body B_1. The origin of N is taken to be the mass center of the spacecraft. (This, along with partitioning external forces properly, decouples the multibody motion considered here from the orbital motion of the mass center.) With $r_{ik} = r_{ok} = \omega_i = \alpha_i = 0$ and $P_{\omega N} = P_{vN} = U$, the joint relations for G_N may be simplified thus:

$$\begin{aligned} \omega_1 &= \sigma_N && (14)\\ v_1 &= s_N && (15)\\ \alpha_1 &= \dot{\sigma}_N && (16)\\ a_1 &= \dot{s}_N && (17) \end{aligned}$$

4 Basis-Free Description of the Procedure

We seek to solve for $\dot{u}$, the time derivatives of the generalized speeds. At the time of solution, all of the generalized speeds and coordinates are known, as well as the applied forces and torques. In general, our focus is on resolving the equations of motion at each joint G_k.

4.1 Scattering States, Accelerations, Remainder Inertia Forces

We traverse the tree from root to tips. At each joint G_k, the velocities of the joint's outer body are related to those of its inner body by:

$$\omega_o = \omega_i + P_{\omega k}\sigma_k \tag{18}$$

$$v_o = v_i + P_{vk}s_k + \omega_i \times (r_{ik} + P_{vk}x_k) - \omega_o \times r_{ok} \tag{19}$$

The remainder accelerations of the outer body of G_k are related to those of the inner body by:

$$\alpha_{ro} = \alpha_{ri} + \dot{P}_{\omega k}\sigma_k + \omega_o \times P_{\omega k}\sigma_k \tag{20}$$

$$\begin{aligned} a_{ro} = \; & a_{ri} + 2\omega_i \times P_{vk}s_k \\ & + \alpha_{ri} \times (r_{ik} + P_{vk}x_k) + \omega_i \times \omega_i \times (r_{ik} + P_{vk}x_k) \\ & - \alpha_{ro} \times r_{ok} - \omega_o \times \omega_o \times r_{ok} \end{aligned} \tag{21}$$

The remainder inertia force is:

$$F^*_{rj} = \left\{ \begin{array}{c} -I\alpha_r - \omega \times H \\ -ma_r \end{array} \right\}_j \tag{22}$$

4.2 Gathering Mass and Force

Now we traverse the tree from tips to root. Consider a body B_j which is the outer body for joint G_k and is the inner body for distal joint(s) G_d. We gather the articulated force vector and articulated mass matrix for each G_k.

$$F_j = R_j + F^*_{rj} \tag{23}$$

$$\overline{F}_k = F_j + \sum_d T_d \overline{F}_d \tag{24}$$

$$\overline{M}_k = M_j + \sum_d T_d \overline{M}_d \tag{25}$$

These must be computed for all joints, including G_N. Tip bodies have no distal joints, so for them, $\overline{F}_k = F_j$, $\overline{M}_k = M_j$.

4.3 Computing Dynamics Matrices

As we traverse from tips to root, we compute the dynamics matrix, absorption matrix, and transmission matrix for each joint. The dynamics matrix D_k is a mass matrix, gathered at G_k and projected into the subspace spanned by the DOF of G_k. It may be thought of as the apparent mass for motion of the joint. The absorption matrix A_k projects whatever it multiplies into the subspace spanned by the DOF of the joint. Whatever isn't absorbed is transmitted through the joint to the next body inboard via the transmission matrix T_k.

$$
\begin{aligned}
D_k &= P_k^T \overline{M}_k P_k && (26)\\
A_k &= \overline{M}_k P_k D_k^{-1} P_k^T && (27)\\
T_k &= U - A_k && (28)
\end{aligned}
$$

These must be computed for joints G_1 to G_{N_g}, but are not needed for joint G_N. (G_N being the end of the line, $A_N = U$ and $T_N = 0$ always.)

4.4 Scattering State Derivatives

The third and final traversal is from root to tips. Starting from the "inertial joint" G_N, we solve for the variables of integration $\dot{u}_N$:

$$
\begin{aligned}
\dot{u}_N &= \overline{M}_N^{-1} \overline{F}_N && (29)\\
a_{u1} &= \dot{u}_N && (30)
\end{aligned}
$$

we then march outward through the other joints, solving for $\dot{u}_k$ and scattering the associated spatial acceleration for each body as we go:

$$
\begin{aligned}
\dot{u}_k &= D_k^{-1} P_k^T (\overline{F}_k - \overline{M}_k a_{ui}) && (31)\\
a_{uo} &= a_{ui} + P_k \dot{u}_k && (32)
\end{aligned}
$$

5 Basis-Loaded Procedure

Now that we understand the procedure in its uncluttered basis-free form, we repeat it, accounting for reference frames and reference points. We express all terms in a spatial vector (eg. $\overline{F}$) or spatial matrix (eg. $\overline{M}$) in a common frame. We use "spatial" direction cosine matrices to rotate spatial vectors and spatial matrices from one frame to another, and shift matrices to move them from point to point.

We define a "spatial" direction cosine matrix as:

$$ {}^{B}\overline{C}^{A} = \begin{bmatrix} {}^{B}C^{A} & 0 \\ 0 & {}^{B}C^{A} \end{bmatrix} \tag{33} $$

so that rotating a spatial vector is accomplished as:

$$ {}^{B}z = \left\{ \begin{array}{c} {}^{B}z_{\omega} \\ {}^{B}z_{v} \end{array} \right\} = \begin{bmatrix} {}^{B}C^{A} & 0 \\ 0 & {}^{B}C^{A} \end{bmatrix} \left\{ \begin{array}{c} {}^{A}z_{\omega} \\ {}^{A}z_{v} \end{array} \right\} = \left\{ \begin{array}{c} {}^{B}C^{A\,A}z_{\omega} \\ {}^{B}C^{A\,A}z_{v} \end{array} \right\} \tag{34} $$

Rotating a spatial matrix is done like:

$$ \begin{aligned} {}^{B}Z = \begin{bmatrix} {}^{B}Z_{11} & {}^{B}Z_{12} \\ {}^{B}Z_{21} & {}^{B}Z_{22} \end{bmatrix} &= \begin{bmatrix} {}^{B}C^{A} & 0 \\ 0 & {}^{B}C^{A} \end{bmatrix} \begin{bmatrix} {}^{A}Z_{11} & {}^{A}Z_{12} \\ {}^{A}Z_{21} & {}^{A}Z_{22} \end{bmatrix} \begin{bmatrix} {}^{A}C^{B} & 0 \\ 0 & {}^{A}C^{B} \end{bmatrix} \\ &= \begin{bmatrix} {}^{B}C^{A\,A}Z_{11}\,{}^{A}C^{B} & {}^{B}C^{A\,A}Z_{12}\,{}^{A}C^{B} \\ {}^{B}C^{A\,A}Z_{21}\,{}^{A}C^{B} & {}^{B}C^{A\,A}Z_{22}\,{}^{A}C^{B} \end{bmatrix} \end{aligned} \tag{35} $$

We'll use two shift operator matrices: S_F is used to shift forces and masses, while S_v is used to shift velocities and accelerations. The force shift operator matrix is:

$$ S_F(r) = \begin{bmatrix} U & -r^{\times} \\ 0 & U \end{bmatrix} \tag{36} $$

with r being the displacement vector oriented from the origin of the shift to the destination, and

$$ r^{\times} = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix} \tag{37} $$

As an example, let's apply it to shifting a force spatial vector from the mass center of B_o to the joint G_k:

$$ \overline{F}_k = S_F(r_{ok})F_o = \begin{bmatrix} U & -r_{ok}^{\times} \\ 0 & U \end{bmatrix} \left\{ \begin{array}{c} T_o \\ F_o \end{array} \right\} = \left\{ \begin{array}{c} T_o - r_{ok} \times F_o \\ F_o \end{array} \right\} \tag{38} $$

All quantities involved (r_{ok}, T_o, F_o) must be expressed in a common frame. We will express them in B_o. As a consequence, $\overline{F}_k$ will also be expressed in B_o. Mass matrices are shifted like this:

$$ \begin{aligned} & S_F(r)MS_F^T(r) \\ &\quad = \begin{bmatrix} U & -r^{\times} \\ 0 & U \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} U & 0 \\ r^{\times} & U \end{bmatrix} \\ &\quad = \begin{bmatrix} M_{11} - r^{\times}M_{21} + M_{12}r^{\times} - r^{\times}M_{22}r^{\times} & M_{12} - r^{\times}M_{22} \\ M_{21} + M_{22}r^{\times} & M_{22} \end{bmatrix} \end{aligned} \tag{39} $$

As an example, let's shift the mass of B_o to G_k:

$$\begin{aligned}
\overline{M}_k &= S_F(r_{ok}) M_o S_F^T(r_{ok}) \\
&= \left[\begin{array}{cc} U & -r_{ok}^{\times} \\ 0 & U \end{array}\right] \left[\begin{array}{cc} I_o & 0 \\ 0 & m_o U \end{array}\right] \left[\begin{array}{cc} U & 0 \\ r_{ok}^{\times} & U \end{array}\right] \\
&= \left[\begin{array}{cc} I_o - m_o r_{ok}^{\times} r_{ok}^{\times} & -m_o r_{ok}^{\times} \\ m_o r_{ok}^{\times} & m_o U \end{array}\right]
\end{aligned} \tag{40}$$

To check the sign on the $mr^{\times}r^{\times}$ term, we expand it out:

$$-mr^{\times}r^{\times} = m \left[\begin{array}{ccc} r_2^2 + r_3^2 & -r_1 r_2 & -r_1 r_3 \\ -r_2 r_1 & r_3^2 + r_1^2 & -r_2 r_3 \\ -r_3 r_1 & -r_3 r_2 & r_1^2 + r_2^2 \end{array}\right] \tag{41}$$

so we recognize $I - mr^{\times}r^{\times}$ as the familiar parallel-axis theorem.

The velocity shift matrix is

$$S_v(r) = \left[\begin{array}{cc} U & 0 \\ -r^{\times} & U \end{array}\right] \tag{42}$$

For example, let's shift a spatial velocity from the mass center of B_i to the joint G_k (neglecting $P_{vk} x_k$ for this illustration):

$$\overline{v}_k = S_v(r_{ik}) v_i = \left[\begin{array}{cc} U & 0 \\ -r_{ik}^{\times} & U \end{array}\right] \left\{\begin{array}{c} \omega_i \\ v_i \end{array}\right\} = \left\{\begin{array}{c} \omega_i \\ v_i + \omega_i \times r_{ik} \end{array}\right\} \tag{43}$$

Note that $S_v(r) = S_F(-r)^T$. One might choose to have only one shift operator, and keep track of where the sign and transpose are needed. We prefer to keep the two annotated versions for clarity.

This formalism introduces some zero submatrices. In the implementation, we write the shift and rotation functions so that we don't waste cycles multiplying things by zero.

5.1 Scattering States, Accelerations, Remainder Inertia Forces

All quantities associated with a body are expressed in that body. Rotational joint partials are expressed in the joint's outer body, while the translational joint partials are expressed in the joint's inner body. The rates and remainder accelerations are:

$$\begin{aligned}
\omega_o &= {}^oC^i\omega_i + P_{\omega k}\sigma_k && (44)\\
v_o &= {}^oC^i[v_i + P_{vk}s_k + \omega_i \times (r_{ik} + P_{vk}x_k)] - \omega_o \times r_{ok} && (45)\\
\alpha_{ro} &= {}^oC^i\alpha_{ri} + \dot{P}_{\omega k}\sigma_k + \omega_o \times P_{\omega k}\sigma_k && (46)\\
a_{ro} &= {}^oC^i[a_{ri} + 2\omega_i \times P_{vk}s_k \\
&\quad + \alpha_{ri} \times (r_{ik} + P_{vk}x_k) + \ \omega_i \times \omega_i \times (r_{ik} + P_{vk}x_k)] \\
&\quad - \alpha_{ro} \times r_{ok} - \omega_o \times \omega_o \times r_{ok} && (47)
\end{aligned}$$

For G_N, these simplify to:

$$\begin{aligned}
\omega_1 &= \sigma_N && (48)\\
v_1 &= {}^1C^N s_N && (49)\\
\alpha_{r1} &= 0 && (50)\\
a_{r1} &= 0 && (51)
\end{aligned}$$

Having the velocities and remainder accelerations, we can construct the remainder inertia forces:

$$F^*_{rj} = \left\{ \begin{array}{c} -I\alpha_r - \omega \times H \\ -ma_r \end{array} \right\}_j \tag{52}$$

where

$$H_j = I_j\omega_j + h_j \tag{53}$$

and h_j is the momentum contribution of embedded wheels or other spinning parts. The remainder acceleration terms contain the centrifugal, Coriolis, and Euler terms. The $\omega \times H$ term is sometimes known as the gyroscopic torque. All terms are expressed in B_j.

5.2 Gathering Mass and Force

Consider a body B_o which is the outer body for joint G_k and is the inner body for distal joint(s) G_d. For clarity, let's denote the body depending from G_d as a child body, B_c.

$$\begin{aligned}
F_o &= R_o + F^*_{ro} && (54)\\
\overline{F}_k &= S_F(r_{ok})F_o + \sum_d S_F(r_{dk})\,{}^o\overline{C}^c T_d \overline{F}_d && (55)\\
\overline{M}_k &= S_F(r_{ok})M_o S_F^T(r_{ok}) + \sum_d S_F(r_{dk})\,{}^o\overline{C}^c T_d \overline{M}_d\,{}^c\overline{C}^o [S_F(r_{dk})]^T && (56)
\end{aligned}$$

where $r_{dk} = r_{ok} - (r_{od} + P_{vd} x_d)$.

These quantities are computed for all joints, including G_N.

5.3 Computing Dynamics Matrices

With bases introduced, P_k is

$$P_k = \begin{bmatrix} P_{\omega k} & 0 \\ 0 & {}^oC^i P_{vk} \end{bmatrix} \tag{57}$$

so that all of P_k is expressed in B_o. For G_N,

$$P_N = \begin{bmatrix} U & 0 \\ 0 & {}^1C^N \end{bmatrix} \tag{58}$$

We then compute the dynamics matrix, absorption matrix, and transmission matrix:

$$\begin{aligned} D_k &= P_k^T (\overline{M}_k P_k) \quad (59) \\ A_k &= (\overline{M}_k P_k)(D_k^{-1} P_k^T) \quad (60) \\ T_k &= U - A_k \quad (61) \end{aligned}$$

All of these are expressed in B_o. The gratuitous-seeming parentheses highlight some groupings that show up more than once, enabling some efficiencies in implementation. A and T do not need to be computed for G_N ($A_N = U,\ T_N = 0$).

5.4 Scattering State Derivatives

$$\begin{aligned} \dot{u}_N &= (D_N^{-1} P_N^T) \overline{F}_N \quad (62) \\ a_{u1} &= P_N \dot{u}_N \quad (63) \\ \dot{u}_k &= (D_k^{-1} P_k^T)[\overline{F}_k - \overline{M}_k {}^o\overline{C}^i S_v(r_{ik} + P_{vk} x_k) a_{ui}] \quad (64) \\ a_{uo} &= S_v(r_{ko})[{}^o\overline{C}^i S_v(r_{ik} + P_{vk} x_k) a_{ui} + P_k \dot{u}_k] \quad (65) \end{aligned}$$

6 Conclusion

The Order-N procedure presented here has been implemented in 42. Numerical agreement with the existing Gaussian-elimination method is excellent. Solution speed is comparable for two-body systems. As the number of bodies increases, the Order-N method is seen to be dramatically faster than the Gaussian-elimination method, as expected.

As of this writing, some features supported by the Gaussian elimination method are not yet supported by the Order-N method: flexible bodies, locking some joint DOFs, and computing constraint forces. We intend to add those features at some point in time.

References

[1] Jain, Abhinandan, *Robot and Multibody Dynamics: Analysis and Algorithms*, Springer, 2011.

[2] Featherstone, Roy. *Rigid Body Dynamics Algorithms*, Springer, 2008.

[3] Kane, Thomas R. and Levinson, David A., *Dynamics: Theory and Application*, McGraw-Hill, 1985.

[4] Kane, Thomas R., Likins, Peter W., and Levinson, David A., *Spacecraft Dynamics*, McGraw-Hill, 1983.

[5] Hughes, Peter C., *Spacecraft Attitude Dynamics*, John Wiley and Sons, 1986.

[6] Amirouche, Farid, *Fundamentals of Multibody Dynamics: Theory and Applications*, Birkhäuser, 2006.