DEVELOPMENT AND EVALUATION OF AN ORDER-N FORMULATION FOR
MULTI-FLEXIBLE BODY SPACE SYSTEMS

Tushar K. Ghosh
L-3 Communications, STRATIS
1002, Gemini Avenue, Suite 200
Houston, TX 77058, USA
E-mail: tushar.ghosh@l-3com.com

Leslie J. Quiocho
Mail Code ER7
NASA Lyndon B. Johnson Space Center
2101 NASA Parkway
Houston, TX, 77058, USA
E-mail: leslie.j.quiocho@nasa.gov

KEYWORDS
Multibody dynamics, Rigid and flexible bodies, Tree topology, Constraints, Closed loops, Order-N.

ABSTRACT
This paper presents development of a generic recursive Order-N algorithm for systems with rigid and flexible bodies, in tree or closed-loop topology, with N being the number of bodies of the system. Simulation results are presented for several test cases to verify and evaluate the performance of the code compared to an existing efficient dense mass matrix-based code. The comparison brought out situations where Order-N or mass matrix-based algorithms could be useful.

INTRODUCTION
The Software, Robotics and Simulation Division (SRSD) of NASA Lyndon B. Johnson Space Center provides math modeling and simulation in support of engineering analyses and crew training activities for the center. The division currently has an efficient generic multibody dynamics code based on a dense mass-matrix formulation, which is used for simulating systems involving on-orbit robotic manipulators such as the Canadian Space Agency-built Space Station Manipulator System (SSRMS). It is generally known that Order-N (O(N)) algorithms, which involve arithmetic operation counts of the order N, where N is the number of bodies, perform more efficiently for systems with large degrees of freedom, compared to mass matrix-based with operations of order N^3. It was therefore decided to develop an O(N) simulation for SRSD to investigate applications where they may perform better. This development was performed in-house to allow maximum flexibility and control in different simulations.

ALGORITHM DEVELOPMENT
There are several methods in the literature that may be used for developing an O(N) algorithm. References may be found in (Banerjee, 2003). The formulation presented here is based on algebraically putting together the following steps: (1) kinematic equations relating motions between consecutive joints, (2) equations of motion of a single body, rigid or flexible, (3) equations relating the total spatial force and active forces and moments at the joint and (4) constraint conditions.

Derivation of the Order-N algorithm is presented in the Appendix. The effect of motion-induced stiffness (Banerjee, 1993) in flexible bodies of the system has not been incorporated yet. Inter-body forces may cause this effect to be important in some cases even for slow motions typical of space systems.

CODE VERIFICATION
The Order-N code was verified against the existing mass matrix-based code for several test cases, which itself was verified against other simulations in the industry, including TREETOPS (Singh et al.,1985). The mass-matrix algorithm has been in use for many facilities at the Johnson Space Center for many years. Results from the two implementations matched with high accuracy (i.e., within 1.0e-10 or better).

SIMULATION TEST CASES
The test cases are based on variations of the system shown in Figure 1. The base plate B₀ is a rigid circular plate floating in inertial space.

![Figure 1: Test Model Description](https://ntrs.nasa.gov/search.jsp?R=2013009745)

Three identical articulating legs and two identical manipulator arms are rigidly attached to the top plate. The other ends of the legs are rigidly connected to the base plate and the other ends of the manipulators hold a test object rigidly. All links are modeled as cylindrical rods. All joints of the legs and manipulators are single axis rotational joints. Two configurations of the legs and manipulators are considered, one with six joints the other with seven. The axes of the seven jointed manipulators and legs are in the order roll, yaw, pitch, pitch, pitch, yaw.
and roll. The axes for ones with six joints are in the order
roll, yaw, pitch, pitch, yaw and roll. In several
configurations the boom elements of the legs have been
split into two parts of equal length and joined rigidly, for
adding additional bodies and flex degrees of freedom to
the system. Only the boom elements are modeled as
flexible. Each flexible rod has four bending modes (two in
each of the bending planes). The system was driven by
forces and moments on the top plate and the test object,
and moments on joints of the first leg. The forces and
moments were held constant for every 20 seconds and
then switched in sign. Table 1 shows the configurations.
Open loop cases are obtained by freeing the joints
between legs 2, 3 and top plate and between manipulator
2 and the test object. For closed loop simulations
constraint forces were determined at these same points.

Table 1: System Test Case Configurations

<table>
<thead>
<tr>
<th>System Configuration</th>
<th>Legs</th>
<th>Manipulators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Bodies</td>
<td>Number of Joints</td>
<td>Number of Bodies</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>7</td>
</tr>
</tbody>
</table>

Plots of representative data are shown in Figures 2 and 3.
Figure 2 shows a co-plot of displacements of the center
of the top plate, and the joint between manipulator 1 and
the test object obtained from O(N) and mass matrix
simulations for Case 16. The two results match to within
1.0e-10. Figure 3 is a co-plot of angular acceleration of
the bottom pitch joint of leg 2 for Cases 12 and 16 (rigid
and flex respectively) for O(N). Distances are shown in
meters, while angles are shown in radians.

Figures 3: Leg 2 Bottom Pitch Joint Angular Acceleration

Performance of the O(N) code is measured by the CPU
time for a 100 second simulation and comparing it with
the existing code. The results of the comparison are listed
below for rigid and flexible models separately for both
open-loop and closed-loop scenarios. Rigid cases were
run with 0.001 second and flexible cases were run with
0.0001 second integration time step. An Euler-Cromer
integration scheme was used.

Table 2: Timing Results

<table>
<thead>
<tr>
<th>Case Number</th>
<th>Rigid (R) or Flex (F) Simulation</th>
<th>Configuration (Table 1)</th>
<th>Degrees of Freedom</th>
<th>CPU Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R</td>
<td>I</td>
<td>24</td>
<td>10.1</td>
</tr>
<tr>
<td>2</td>
<td>R</td>
<td>2</td>
<td>24</td>
<td>12.4</td>
</tr>
<tr>
<td>3</td>
<td>R</td>
<td>3</td>
<td>27</td>
<td>15.0</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>4</td>
<td>42</td>
<td>22.3</td>
</tr>
</tbody>
</table>
Cases 9, 10, 11, 13, 14 and 15 have two loops, cases 12 and 16 have three loops.

**DISCUSSION AND CONCLUSIONS**

The timing results confirm that the dense mass matrix formulation is faster than the O(N) formulation for smaller degrees of freedom (DOF’s) and O(N) is faster otherwise. In our cases for all rigid systems mass-matrix take less CPU time than O(N) for both open-loop and closed-loop systems because they have fewer DOFs. Flexible body systems have more DOF and O(N) run faster, except for the case 13 involving two loops and fewer DOF than the other flex cases. Loops added large amount of computation and more so for O(N). For systems with loops the advantage of O(N) is reduced.

With more loops the reduction is more because it requires solution of coupled linear constraint equations which can have a large number of unknown constraint forces and moments, and are solved in the usual manner that requires \( n_c^3 \) arithmetic operations, where \( n_c \) is the number of constrained degrees of freedom.

The high degree of match between the mass-matrix and O(N) results is expected because the two methods solve the same set of equations using many common codes.

**ACKNOWLEDGMENT**

The authors acknowledge the support provided by An Huynh and Thomas Brain of METECS Corporation, Houston Texas, USA in coding the simulation, and to Arun Banerjee, formerly of Lockheed Martin Palo Alto Research Laboratory, California, USA for review of the manuscript.

**REFERENCES**


**APPENDIX: DERIVATION OF O(N) EQUATIONS**

**System Description and Definitions**

**Figures A1: Multibody System Definitions**

**Figures A2: Constrained Motion Definitions**

Consider the set of rigid and/or flexible bodies, in Figure A1, with the associated labeling of forces and points. \( \mathbf{a}_n \) and \( \mathbf{\dot{a}}_k \) are frames attached to body \( n (B_n) \) at its inboard and outboard joints \( O_n \) and \( O_{k_i} \) respectively.

Figure A2 shows the labeling for constrained systems. Throughout this derivation it is assumed that all the vectors and inertia matrices are either generated in or are converted to a single reference frame, which is the structural reference frame \( \mathbf{a}_0 \) of the base body of the multibody system. The conversion is performed every time the system states are updated.

**Kinematics of Motion between \( O_p \) and \( O_n \)**

Let \( \mathbf{\dot{v}}_n, \mathbf{\ddot{a}}_n \) represent the inertial velocity and acceleration respectively, of \( O_n \), \( \mathbf{\dot{w}}_n \) represent the inertial angular
velocity of $\delta_n$ and $\eta_n$ represent the flexible variables of the body $B_p$. Kinematic equations relating position, velocity and acceleration of $O_n$ and $\delta_n$, and angular velocity and acceleration of frames $\delta_n$ and $\eta_n$ are given by:

$$\ddot{\eta}_n = \ddot{\delta}_n + \dot{\omega}_p \times \eta_n + \ddot{\omega}_p \times \dot{\eta}_n + 2 \dot{\omega}_p \times \dot{\omega}_p \times \eta_n$$

where $\ddot{\delta}_n$ is the undeformed value of $\dot{p}_O$, $r_j(\mathbf{1.3})$, the vector from $p_O$ to point $j$ on body $n$ at $n_O$. They act in the directions $\hat{n}S$, $\hat{n}S$, $\hat{n}S$

$$\rho$$

be in the inertial frame of freedom of joint $n$ and $n,i$ the rotations $n,i$ of the degrees of freedom of the joint.

$nL$ is a matrix whose columns are translation and rotation. The translation of $n_O$ with respect to $n_O$ is given by $nL_n$, and $nR$ is the number of rotational degrees of freedom of the joint $n$. Differentiating $\delta_n$ in the inertial frame with respect to $nO$ is given by $\dot{\delta}_n$.

$$\ddot{\delta}_n = \ddot{\delta}_n + \dot{\omega}_p \times \ddot{\delta}_n + \dot{\omega}_p \times \dot{\delta}_n + 2 \ddot{\omega}_p \times \dot{\omega}_p \times \delta_n$$

where $\dot{\delta}_n$ is the undetermined value of $\dot{p}_O$, the vector from $O_p$ to $O_n$. Combining the last two equations we can write

$$\dot{A}_n = \dot{s}_{i,n} A_p + S_n \dot{\eta}_n + A_{n,r}$$

where, $s_{i,n} = \left[ \begin{array}{cc} 1 & - \tilde{\eta}_n \\ 0 & 1 \end{array} \right]$, $S_n = \left[ \begin{array}{c} \tilde{\delta}_n \\ \ddot{\eta}_n \end{array} \right]$

and $A_{n,r}$, $A_{n,r}$, $\delta_{n,r}$, $\tilde{\delta}_{n,r}$ is the usual cross product operator on a vector.

The joint $n$ may have up to six DOFs allowing both translation and rotation. The translation of $O_n$ with respect to $O_n$ is given by $\tilde{y}_n = \sum_{i=1}^{NT_n} \tilde{\delta}_{i,n} \delta_{i,n} = G_n \delta_n$ where $\tilde{\delta}_{i,n}$ are linearly independent unit vectors fixed in $O_n$, $NT_n \leq 3$ is the number of translational degrees of freedom of joint $n$ and $\delta_{i,n}$ are scalar quantities, representing joint translations in the directions of the unit vectors. The inertial acceleration of $O_n$ is:

$$\ddot{\delta}_n = \ddot{\eta}_n - \ddot{\omega}_p \times \tilde{y}_n + \dot{\omega}_p \times \dot{\delta}_n + \ddot{\omega}_p \times \dot{\omega}_p \times \delta_n$$

where, $\ddot{\delta}_n = \omega_2 \omega_3 \dot{y}_n + 2 \omega_3 \omega_2 \delta_n$.

Inertial angular velocity of $\delta_n$ is given by

$$\ddot{\delta}_n = \ddot{\omega}_n + \sum_{i=1}^{NR_n} \dot{\delta}_{i,n} \dot{\theta}_{i,n} = \ddot{\omega}_n + L_n \dot{\theta}_n$$

where $\dot{L}_n$ is the unit vector in the direction of the rotation $\dot{\theta}_{i,n}$ and $NR_n$ is the number of rotational degrees of freedom of the joint. $L_n$ is a matrix whose columns are the unit vectors $\dot{L}_n$ and $\theta_n$ is a column matrix containing the rotations $\dot{\theta}_{i,n}$. Differentiating $\ddot{\delta}_n$ in the inertial frame we get

$$\ddot{\ddot{\delta}}_n = \ddot{\ddot{\omega}}_n + L_n \ddot{\theta}_n + \dddot{\omega}_n$$

Combining Equations (1.2) and (1.3),

$$A_n = \tilde{s}_{n,\delta} A_n + P_n \dot{\gamma}_n + A_{n,\theta}$$

$$A_n = \left[ \begin{array}{c} \dot{\delta}_n \\ \dot{\eta}_n \end{array} \right]$$

where

$$\dot{\gamma}_n = \left[ \begin{array}{c} \dot{\delta}_n \\ \dot{\eta}_n \end{array} \right]$$

$$A_{n,\theta} = \left[ \begin{array}{c} \alpha_r \\ \alpha_\theta \end{array} \right]$$

Finally, combining Equations (1.1) and (1.4) we get

$$A_n = \tilde{s}_{n,\delta} A_n + P_n \dot{\gamma}_n + A_{n,\theta}$$

where, using the relationship $\tilde{\gamma}_n = \tilde{\delta}_n + \tilde{\eta}_n$,

$$\tilde{s}_{n,\delta} = \tilde{s}_{n,\delta} \tilde{s}_{n,\eta} = \tilde{s}_{n,\delta} = \tilde{s}_{n,\delta} A_n + A_{n,\theta}$$

Inter-Body and Joint Actuator Forces

Let $F_n$ be the spatial forces exerted by $B_p$ on $B_n$ at $O_n$ and $\tilde{F}_n$ be the spatial force exerted by $B_n$ on $B_p$ at $O_n$. Then from equilibrium considerations,

$$\tilde{F}_n = -\tilde{s}_{n,\delta}^T F_n$$

where $\tilde{s}_{n,\delta}$ is given by Equation (1.5). Let $\mu_n$ and $\nu_{j,n}$ be respectively, the $i$-th actuator force ($i \leq 3$) and $j$-th actuator torque ($j \leq 3$) on body $n$ at $O_n$. They act in the directions of the degrees of freedom $\dot{\delta}_{i,n}$ and $\dot{\theta}_{j,n}$ of the joint.

Defining $\sigma_n = \left[ \begin{array}{c} \mu_n \\ \nu_{j,n} \end{array} \right]$ as the array of actuator forces and moments at joint $n$, it is straightforward to show that

$$\sigma_n = P_n^T F_n$$

where $P_n$ is given by Equation (1.5).

Equations of Motion of Body $n$

A minor modification of equations of a flexible body in Quiocho, et. al (Quiocho, 2010), produces the equations of a body $B_n$ in rigid and flexible coordinates as

$$M_{er,n} A_n + M_{re,n} \tilde{\delta}_n = \sum_{i=1}^{NR_n} \tilde{S}_{i,n} F_{e,i,n} + B_{e,n}$$

We shall write the equations of motion of the body $n$ after separating the external spatial forces $F_{e,i,n}$ on $B_n$ as (i) from the previous body acting at the inboard joint ($F_{i,n}$), (ii) from child bodies $B_{ki}$ acting at joint, (iii) forces at constrained points ($F_{c,i,j}$) when there are such points on the body, and (iv) external forces ($F_{ext,i,n}$) acting on the body. Because we chose the reference frame of the body to be at the inboard joint, the shift function at the inboard joint is an identity matrix and the shape/slope function at the inboard joint is a null matrix. Equations for body $n$ are:
\[M_{n,n}A_n + M_{nn,n}\tilde{q}_n = F_n + \sum_{j \in Z_c(n)} S_{n,j}^T \tilde{F}_{c,j} + \sum_{i \in Z(n)} \tilde{s}_{i,n}T F_{k_i} + B_{r,n}\]

\[M_{er,n}A_n + M_{ee,n}\tilde{q}_n + K_{ee,n}\tilde{q}_n + D_{ee,n}\tilde{q}_n = \sum_{j \in Z(n)} S_{n,j}^T \tilde{F}_{c,j} + \sum_{i \in Z(n)} S_{i,n}^T F_{k_i} + B_{e,n}\]

In the above equations, \(F_{c,j}\) is the spatial force at the \(j\)-th constrained point from \(n_0\) and \(S_{c,j}\) is the shape/slope function of body \(n\) at the constrained point. 

Defining 
\[G_{r,n} = \sum_{i} S_{i,n}^T F_{k_i} + B_{r,n}\]
\[G_{e,n} = \sum_{i} S_{i,n}^T F_{k_i} + B_{e,n} - K_{ee,n}\tilde{q}_n - D_{ee,n}\tilde{q}_n\]

Using Equations (2.1) for \(\tilde{F}_{k_i}\) and then Equations (3.1) and (3.2) we get the equations of motion of the body \(n\) as 

\[M_{ir,n}A_n + M_{re,n}\tilde{q}_n = F_n - \sum_{j \in Z_c(n)} S_{n,j}^T F_{k_i} + G_{r,n}\]  

\[M_{ir,n}A_n + M_{re,n}\tilde{q}_n = -\sum_{j \in Z_c(n)} S_{n,j}^T F_{k_i} + G_{e,n}\]

Here, \(Z_c(n)\) is the set of indices for constrained points on body \(n\).

**Recursive Solution of Equations of Motion**

The dynamical equations of motion of the system are solved recursively in steps as follows.

**Step 1.** Forward Pass Kinematics: In this step all position and velocity states are generated starting with the base body using equations derived in Kinematics section.

**Step 2.** Inward Pass Dynamics Equations:

Observing the equations of motion it is reasonable to expect that for any \(n\) \(F_n\) can be expressed as linear functions of \(A_p\), \(\dot{q}_p\) and constraint forces on itself and on bodies on outer branches:

\[F_n = D_{A,n}A_p + D_{q,n}\dot{q}_p + d_n + \sum_{j \in Y_c(n)} D_{c,n,j}F_{c,j}\]

\(Y_c(n)\) is the set of indices of all constrained points on body \(n\) and on all bodies of its outer branches.

We shall determine the coefficients in the above equations recursively. In Equation (4.1) using \(k_i\) in place of \(n\) and \(n\) in place of \(p\), and using the result in Equation (4.1) we get

\[\ddot{q}_n = \dot{Q}_{A,n}A_n + \ddot{q}_n + \sum_{j \in Y_c(n)} \dot{Q}_{c,n,j}F_{c,j}\]

\[\dot{\dot{Q}}_{A,n} = M_{\dot{q},n}^{-1}\dot{M}_{\dot{q},n}, \quad \dot{M}_{\dot{q},n} = \dot{M}_{ee,n} + \sum_{i \in Z(n)} S_{i,n}D_{k_i}\]

\[\dot{\dot{Q}}_{c,n,j} = -M_{\dot{q},n}S_{c,n,j}D_{c,k_i,j}\quad \text{for } i \in Z(n), \quad j \in Y_c(k_i)\]

\[\dot{\dot{Q}}_{c,n,j} = -M_{ee,n}\ddot{S}_{c,n,j}\quad \text{for } j \in Z_c(n)\]

\(Z(n)\) is the set of indices of child bodies of \(n\). In the same manner, substituting for \(F_{k_i}\) in Eq. (3.3) and rearranging we get

\[\ddot{q}_{n} = \dot{Q}_{\dot{q},n}A_n + \ddot{q}_{n} + \sum_{j \in Y_c(n)} \dot{Q}_{c,n,j}F_{c,j}\]

\[\dot{\dot{Q}}_{\dot{q},n} = M_{\dot{q},n}^{-1}\dot{M}_{\dot{q},n}, \quad \dot{M}_{\dot{q},n} = \dot{M}_{ee,n} + \sum_{i \in Z(n)} S_{i,n}D_{k_i}\]

\[\dot{\dot{Q}}_{c,n,j} = -M_{ee,n}\ddot{S}_{c,n,j}\quad \text{for } i \in Z(n), \quad j \in Y_c(k_i)\]

\[\dot{\dot{Q}}_{c,n,j} = -M_{ee,n}\ddot{S}_{c,n,j}\quad \text{for } j \in Z_c(n)\]

Using Equation (4.2) for \(\ddot{q}_n\) in Equation (4.3) we get

\[\ddot{q}_{n} = \dot{Q}_{\dot{q},n}A_n + \ddot{q}_{n} + \sum_{j \in Y_c(n)} \dot{Q}_{c,n,j}F_{c,j}\]

Pre-multiplying this equation by \(P_{n}^T\), using

\[\sigma_n = P_{n}^TF_n\quad \text{from Equation (2.2)}\] and solving the resulting equation for \(\ddot{q}_n\) we get

\[\ddot{q}_n = B_{A,n}A_p + B_{q,n}\dot{q}_p + b_n + \sum_{j \in Y_c(n)} B_{c,n,j}F_{c,j}\]

(4.5)

\[B_{A,n} = -M_{\dot{q},n}^{-1}P_{n}^T\dot{M}_{\dot{q},n}S_{n,p}\]

\[B_{q,n} = -M_{\dot{q},n}^{-1}P_{n}^T\dot{M}_{\dot{q},n}S_{q,p}\]

\[b_n = -M_{\dot{q},n}^{-1}\left(P_{n}^T\dot{G}_{r,n} - \dot{M}_{\dot{q},n}A_nR\right)\]

\[B_{c,n,j} = -M_{\dot{q},n}^{-1}P_{n}^TD_{c,n,j}\quad \text{and } M_{\dot{q},n} = P_{n}^T\dot{M}_{\dot{q},n}P_n\]

Equation (4.5) for \(\ddot{q}_n\) in Equation (1.6) yields for \(A_n\)

\[A_n = W_{A,n}A_p + W_{q,n}\dot{q}_p + w_n + \sum_{j \in Y_c(n)} W_{c,n,j}F_{c,j}\]

(4.6)
\[ W_{A,n} = \delta_{a,p} + P_a B_{A,n}, \quad W_{q,n} = S_{a,p} + P_q B_{q,n}, \]
\[ w_n = P_n b_n + A_{n,R}, \quad \text{and} \quad W_{c,n,j} = P_n B_{c,n,j} \]

Equation (4.6) for \( A_n \) in Equation (4.2) gives
\[ \ddot{q}_n = Q_{A,n} A_p + Q_{q,n} \ddot{q}_p + q_n + \sum_{j \in E(n)} Q_{c,n,j} F_{c,j} \quad (4.7) \]
\[ Q_{A,n} = \ddot{Q}_{A,n} W_{A,n} \quad Q_{q,n} = \ddot{Q}_n W_{q,n} \quad \ddot{q}_n = \ddot{Q}_{A,n} w_n + \ddot{q}_n \]
and \( Q_{c,n,j} = \ddot{Q}_{A,n} W_{c,n,j} + \ddot{Q}_{c,n,j} \)

Finally, using Equation (4.6) in Equation (4.4) we get
\[ F_n = D_{A,n} A_p + D_{q,n} \ddot{q}_p + d_n + \sum_{j \in E(n)} D_{c,n,j} F_{c,j} \quad (4.8) \]
\[ D_{A,n} = \ddot{M}_{rr,n} W_{A,n} \quad D_{q,n} = \ddot{M}_{rr,n} W_{q,n} \]
\[ d_n = \ddot{M}_{rr,n} w_n - \ddot{G}_{r,n}, \quad \text{and} \quad \]
\[ D_{c,n,j} = \ddot{M}_{rr,n} W_{c,n,j} - D_{c,n,j} \]

We can see that Equation (4.8) is in the same form as Equation (4.1) we started with, confirming that if the latter is true for child bodies, it would be true for the current body also. Following the same steps as above for any outermost body for which \( F_{k_i} = 0 \) it is easy to show that Equation (5.1) is true for such bodies. Using the induction logic it therefore follows that Equation (4.1) is true for all bodies of the system.

Computations for systems with multiple branches need to start from an outermost body, move inward till a body with multiple child bodies is reached. Inward computation from a body with multiple child bodies should continue only after computations for all of its child bodies are completed.

\( A_0 \) and \( F_0 \) for the Base Body \( B_0 \) in terms of the constraint forces are obtained from Equation (4.4). When \( B_0 \) is fixed in inertial frame \( A_0 = 0 \) and we then have
\[ F_0 = -\ddot{G}_{r,0} + \sum_{j \in Y_0(0)} D_{c,0,j} F_{c,j} \]
When \( B_0 \) is free in the inertial frame \( F_0 = 0 \) and
\[ A_0 = \ddot{M}_{rr,0} [\ddot{G}_{r,0} + \sum_{j \in Y_0(n)} D_{c,0,j} F_{c,j}] \quad (4.9) \]
The flexible coordinate acceleration \( \ddot{q}_0 \) for the base body is obtained from Equation (4.2)
\[ \ddot{q}_0 = \ddot{Q}_{A,0} A_0 + \ddot{q}_0 + \sum_{j \in Y_0(n)} \ddot{Q}_{c,n,j} F_{c,j} \quad (4.10) \]
For systems without closed loops the \( F_{c,j} \) term drops out and the equations obtained above are sufficient for determining \( A_0 \) and \( \ddot{q}_0 \), and then the system accelerations, by successive computation of \( \ddot{q}_n \), \( \ddot{q}_n \) and \( A_n \) in an outward sweep.

**Step 3. Accelerations of Constrained Points in terms of Forces at Constrained Points:**

For the determination of acceleration of constrained points in terms of forces at these points we seek to express the accelerations \( A_n \) and \( \ddot{q}_n \) for bodies in the path from base body to the constrained points in the form
\[ A_n = h_n^A + \sum_{j} H_{n,j} F_{c,j} \quad \text{and} \quad \ddot{q}_n = h_n^q + \sum_{j} H_{n,j} F_{c,j} \quad (4.11) \]

Here the summation index \( j \) covers all constrained points. It follows from Equations (4.9) and (4.11) that when the base is free in inertial frame
\[ h_0^A = \ddot{M}_{rr,0} \ddot{G}_{r,0}\quad H_{0,j} = \ddot{M}_{rr,0} \ddot{D}_{c,0,j}\quad \ddot{q}_0 = \ddot{Q}_{A,0} \ddot{q}_0 + \ddot{q}_0 \quad \text{and} \quad H_{0,j} = \ddot{Q}_{c,0,j}. \]

When the base is fixed, \( h_0^A = 0 \), \( H_{0,j} = 0 \), \( \ddot{q}_0 = \ddot{q}_0 \) and \( H_{0,j} = \ddot{Q}_{c,0,j}. \)

Using Equations (4.7) and (4.11) we get the recursive equations for \( h_n^A \) and \( H_n^A \):
\[ h_n^A = W_{A,n} h_{n-1}^A + W_{q,n} h_{n-1}^q + q_n \quad \text{and} \quad H_n^{A} = W_{A,n} H_{n-1}^{A} + W_{q,n} H_{n-1}^{q} + W_{c,n,j} \]

Using Equation (4.6) and (4.11) we get the recursive equations for \( h_n^q \) and \( H_n^q \):
\[ h_n^q = Q_{A,n} h_{n-1}^A + Q_{q,n} h_{n-1}^q + q_n \quad \text{and} \quad H_n^{q} = Q_{A,n} H_{n-1}^{A} + Q_{q,n} H_{n-1}^{q} + Q_{c,n,j} \]

Let \( E_i \) be the point in constraint \( i \) located on body \( n \) and \( E_i \) be the mating point of the same constraint, located on body \( \ell \). The spatial acceleration of \( E_i \) is
\[ A_{E_i} = \ddot{q}_{E_i} + \sum_{i} S_{E_i,n} \ddot{q}_n + \sum_{i} A_{E_i,r} \quad (4.12) \]
where \( \ddot{q}_{E_i} \) is the standard shift operator for the offset \( \ddot{q}_{E_i} \) of \( E_i \) with respect to \( O_n \)
\[ \ddot{q}_{E_i} \quad (4.12) \]
where \( \ddot{q}_{E_i} \) is the standard shift operator for the offset \( \ddot{q}_{E_i} \) of \( E_i \) with respect to \( O_n \)
\[ \ddot{q}_{E_i} \quad (4.12) \]
where \( \ddot{q}_{E_i} \) is the standard shift operator for the offset \( \ddot{q}_{E_i} \) of \( E_i \) with respect to \( O_n \)
\[ \ddot{q}_{E_i} \quad (4.12) \]
where \( \ddot{q}_{E_i} \) is the standard shift operator for the offset \( \ddot{q}_{E_i} \) of \( E_i \) with respect to \( O_n \)
\[ \ddot{q}_{E_i} \quad (4.12) \]
where \( \ddot{q}_{E_i} \) is the standard shift operator for the offset \( \ddot{q}_{E_i} \) of \( E_i \) with respect to \( O_n \)
\[ \ddot{q}_{E_i} \quad (4.12) \]
where \( \ddot{q}_{E_i} \) is the standard shift operator for the offset \( \ddot{q}_{E_i} \) of \( E_i \) with respect to \( O_n \)
\[ \ddot{q}_{E_i} \quad (4.12) \]
where \( \ddot{q}_{E_i} \) is the standard shift operator for the offset \( \ddot{q}_{E_i} \) of \( E_i \) with respect to \( O_n \)
\[ \ddot{q}_{E_i} \quad (4.12) \]
where \( \ddot{q}_{E_i} \) is the standard shift operator for the offset \( \ddot{q}_{E_i} \) of \( E_i \) with respect to \( O_n \)
\[ \ddot{q}_{E_i} \quad (4.12) \]
where \( \ddot{q}_{E_i} \) is the standard shift operator for the offset \( \ddot{q}_{E_i} \) of \( E_i \) with respect to \( O_n \)
**Step 4. Determination of Constraint Forces:**

Using equations (4.13) and (4.14) the difference in the accelerations at the constraint \( i \) may be written as

\[
\Delta A_{E_i} = A_{E_i} - A_{E_i} = \Delta h_{E_i} + \sum_k \Delta H_{E_i,k} F_{c,k} \quad (4.15)
\]

\[
\Delta h_{E_i} = \hat{s}_{E_i,j} h_{\hat{j}} + S_{E_i,j} h_{\hat{q}} + A_{E_i,r}
\]

\[
\Delta H_{E_i,k} = \hat{s}_{E_i,j} (H^{\Delta}_{E_i,k} - h_{\hat{j}}) - \hat{S}_{E_i,j} (H^\Delta_{E_i,k} - h_{\hat{q}}) + S_{E_i,j} (H^\Delta_{E_i,k} - h_{\hat{q}}) - S_{E_i,j} (H^\Delta_{E_i,k} - h_{\hat{q}})
\]

The summation range for \( k \) is all the constraints.

The linear and angular constraints at \( i \) are

\[
\hat{g}_{c,i,j} \cdot (\hat{v}_{E_i} - \hat{v}_{E_i}) = 0 \quad 0 < j \leq n_{t,i}
\]

\[
\hat{c}_{c,i,j} \cdot (\hat{\omega}_{E_i} - \hat{\omega}_{E_i}) = 0 \quad 0 < j \leq n_{r,i}
\]

where \( \hat{v} \) and \( \hat{\omega} \) represent the inertial velocity and angular velocity of points corresponding to the subscripts, \( \hat{g}_{c,i,j} \) and \( \hat{c}_{c,i,j} \) are unit vectors in the direction of translational and rotational constraints respectively, and \( n_{t,i} \) (\( \leq 3 \)) and \( n_{r,i} \) (\( \leq 3 \)) are the number of these constraints, respectively. \( \hat{v}_{E_i}, \hat{v}_{E_i}, \hat{\omega}_{E_i} \) and \( \hat{\omega}_{E_i} \) are determined in the manner used for the point \( \hat{O}_d \) in the Kinematics section. Defining

\[
\Delta v_{E_i} = \begin{bmatrix} \hat{v}_{E_i} - \hat{v}_{E_i} \\ \hat{\omega}_{E_i} - \hat{\omega}_{E_i} \end{bmatrix}
\]

\[
P_{c,i} = \begin{bmatrix} G_{c,i} & 0 \\ 0 & L_{c,i} \end{bmatrix}
\]

where \( G_{c,i} \) is a \( 3 \times n_{t,i} \) matrix whose columns are the unit vectors \( \hat{g}_{c,i,j} \) and \( L_{c,i} \) is a \( 3 \times n_{r,i} \) matrix whose columns are the unit vectors \( \hat{c}_{c,i,j} \), the difference in the spatial velocities of the constrained points at constraint \( i \) may be written as \( P_{c,i}^T \Delta v_{E_i} \). The difference in the spatial acceleration in the constraint directions at the constraint point should be zero, giving

\[
P_{c,i}^T \Delta A_{E_i} + P_{c,i}^T \Delta v_{E_i} = 0
\]

Using Baumgarte’s stabilization scheme to limit constraint violation caused by numerical errors, this equation is modified to

\[
P_{c,i}^T \Delta A_{E_i} + P_{c,i}^T \Delta v_{E_i} + K_i P_{c,i}^T \Delta X_{E_i} + C_i P_{c,i}^T \Delta V_{E_i} = 0
\]

\( K_i \) and \( C_i \) are positive constants to provide constraint stabilization. Using Equation (4.15) for \( \Delta A_{E_i} \) we have

\[
P_{c,i}^T \sum_k \Delta H_{E_i,k} F_{c,k} = -P_{c,i}^T \Delta h_{E_i} - P_{c,i}^T \Delta V_{E_i}
\]

\[
- K_i P_{c,i}^T \Delta X_{E_i} - C_i P_{c,i}^T \Delta V_{E_i}
\]

Let us define \( f_{c,k,j} \) to be the constraint force in the direction \( \hat{g}_{c,k,j} \) and \( \tau_{c,k,j} \) the constraint torque in the direction \( \hat{c}_{c,k,j} \) and \( \hat{F}_{c,k} = \begin{bmatrix} f_{c,k} \\ \tau_{c,k} \end{bmatrix} \). The force \( F_{c,k} \) at the constraint point \( E_k \) due to forces and moments in the constraint directions is then given by \( P_{c,k}^T \hat{F}_{c,k} \). Let

\[
P_{c,k} = \begin{bmatrix} G_{c,k} & 0 \\ 0 & L_{c,k} \end{bmatrix}
\]

where \( G_{c,k} \) and \( L_{c,k} \) are made of the unit vectors normal to the constraint directions for translation and rotation and \( \hat{F}_{c,k} \) be the array of forces and moments in these directions. Net spatial force at \( E_k \) is

\[
F_{c,k} = P_{c,k} \hat{F}_{c,k} + P_{c,k} \hat{F}_{c,k}
\]

Restricting to cases where \( \hat{F}_{c,k} \) is fully known, we use \( F_{c,k} = P_{c,k} \hat{F}_{c,k} + P_{c,k} \hat{F}_{c,k} \) in Equation (4.17) to get

\[
P_{c,k}^T \sum_k \Delta H_{E_i,k} P_{c,k} \hat{F}_{c,k} = Z_{c,i}
\]

\[
Z_{c,i} = -P_{c,i}^T \left[ \sum_k \Delta H_{E_i,k} P_{c,k} \hat{F}_{c,k} + \Delta h_{E_i} \right]
\]

\[
- P_{c,i}^T \Delta V_{E_i} - K_i P_{c,i}^T \Delta X_{E_i} - C_i P_{c,i}^T \Delta V_{E_i}
\]

Stacking Equation (4.18) for all constraints we get

\[
M_{c} \hat{F}_{c} = Z_{c}
\]

where the \([i,k]\) submatrix of matrix \( M_c \) is given by

\[
M_{c,i,k} = P_{c,i}^T \Delta H_{E_i,k} P_{c,k}
\]

and \( Z_c \) is made of arrays \( Z_{c,i} \) given by Equation (4.19). Equation (4.20) is solved for \( \hat{F}_c \) and the spatial forces \( F_{c,k} \) at constraint points are found using Equation (4.17).

**Step 5. Computation of System Accelerations:**

After determination of forces at the constrained points, \( \hat{\eta}_0 \) and \( \hat{\eta}_0 \) are determined using Equations (4.9) and (4.10). \( \dot{\hat{V}}_n \), \( \dot{\hat{q}}_n \), and \( \dot{\hat{A}}_n \) are determined recursively in a forward pass using Equations (4.5), (4.7) and (1.6) respectively.