Technical Report 32-1592

Attitude Dynamics Simulation Subroutines for Systems of Hinge-Connected Rigid Bodies

G. E. Fleischer
P. W. Likins

May 1, 1974
Technical Report 32-1592

Attitude Dynamics Simulation Subroutines for Systems of Hinge-Connected Rigid Bodies

G. E. Fleischer
P. W. Likins

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

May 1, 1974
Preface

The work described in this report was performed by the Guidance and Control Division of the Jet Propulsion Laboratory.
Errata for Technical Report 32-1565

Since the users of this report will certainly be referring to JPL TR 32-1565 (Ref. 9) for a much more detailed treatment of hinge-connected rigid body system equations and their derivation, it seems appropriate here to add a few technical remarks that may contribute to that report's utility and to correct several significant errors in the text which have been brought to the author's attention.

(1) In the last paragraph on page 30, the hinge torques $\tau^i$, $\tau^i$, and $\tau^i$ are referred to. The symbol $\tau^i$ means $\tau^i/g^i$.

(2) On page 35, when considering the significance of Eq. (34), note that $C^TC = (\hat{C} + \hat{C})^T (\hat{C} + \hat{C}) = (U - \hat{C})(U + \hat{C}) \simeq U + \cdots$ (2nd-degree terms), so orthogonality is preserved in the linear approximation. Still, for planar motion (all $\tilde{\xi}$s the same), it would be impossible to associate an angle of rotation with $C$, since the cosine = 1 and the angle is large. Thus orthogonality is preserved, but physical interpretation is jeopardized.

(3) Note the text corrections that follow.
<table>
<thead>
<tr>
<th>Page</th>
<th>Line</th>
<th>Equation</th>
<th>Corrected Text</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Last</td>
<td>$\epsilon_{\text{seg}} V_s = \cdots$</td>
<td>Change subscript</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Last</td>
<td>$\text{the set } \mathcal{B} \setminus {k} \text{ is } \cdots$</td>
<td>Replace boldface k</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>17</td>
<td>$a_{\text{hk}} = \sum_{\mathcal{R}<em>k \in \mathcal{B}} \epsilon</em>{\text{hk}} C_{\text{r}\Gamma_r} \Gamma_r \Gamma_r^h \Delta C_{\text{r}\Gamma_r} \Gamma_r^h$</td>
<td>Change subscript</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>18</td>
<td>$- \sum_{\mathcal{R}<em>k \in \mathcal{B}} \sum</em>{j \in \mathcal{B} \setminus k} \epsilon_{\text{hk}} (C_{\text{r}\Gamma_r} \Gamma_r^T \Gamma_r C_{\text{r}\Gamma_r} \Gamma_r - \cdots$</td>
<td>Change subscript and summation index</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>12</td>
<td>$+ \sum_{j \in \mathcal{B} \setminus k} (\tilde{D}<em>{\text{jk}} C</em>{\text{jk}} \Gamma_j \cdots$</td>
<td>Add tildes</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>4</td>
<td>$C = \begin{bmatrix} c_1 c_3 &amp; -c_3 s_3 &amp; s_2 \ s_3 s_2 c_3 + s_3 c_1 &amp; -s_3 s_3 s_3 + c_3 c_1 &amp; c_3 s_3 s_3 + c_3 s_3 \ -c_3 s_2 c_3 + s_3 c_1 &amp; -s_3 c_3 + c_3 c_2 &amp; c_3 c_2 \end{bmatrix}$</td>
<td>Note several changes</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>1</td>
<td>$+ \sum_{k \in \mathcal{B} \setminus \mathcal{R}} (\tilde{a}<em>{\text{hk}} + \tilde{a}</em>{\text{hk}}) \gamma_k$</td>
<td>Change summation index</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>4th from last</td>
<td>Any terminal appendage in \ldots</td>
<td>Drop s from appendages</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>26</td>
<td>$\sum_{k \in \mathcal{B} \setminus \mathcal{R}} (\tilde{a}_{\text{hk}} + \cdots$</td>
<td>Add subscript; change summation index</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>5</td>
<td>$- \sum_{k \in \mathcal{B} \setminus \mathcal{R}} \sum_{j \in \mathcal{B} \setminus k} \epsilon_{\text{jk}} (\cdots$</td>
<td>Change subscript and summation index</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>19</td>
<td>$- \sum_{k \in \mathcal{B} \setminus \mathcal{R}} \sum_{j \in \mathcal{B} \setminus k} \epsilon_{\text{jk}} (\cdots$</td>
<td>Change subscript</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>8</td>
<td>$\cdots + \sum_{k \in \mathcal{B} \setminus \mathcal{R}} \epsilon_{\text{jk}} \Gamma_j \cdots$</td>
<td>Drop superscript $j$</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>11</td>
<td>$\cdots$</td>
<td>Add overdot</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>25</td>
<td>\ldots coordinate truncation are feasible.</td>
<td>Replace &quot;transformation is&quot;</td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>9</td>
<td>$\sum_{k \in \mathcal{B}} \sum_{\mathcal{R}<em>k \in \mathcal{B}} (C</em>{\text{r}\Gamma_r} \Gamma_r^T \Gamma_r C_{\text{r}\Gamma_r} \Gamma_r - \cdots$</td>
<td>Note several changes</td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>25</td>
<td>$g^* (\sum_{r \in \mathcal{R}} \sum_{k \in \mathcal{B}} (C_{\text{r}\Gamma_r} \Gamma_r^T \Gamma_r C_{\text{r}\Gamma_r} \Gamma_r - \cdots$</td>
<td>Note several changes</td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>29</td>
<td>$g^* { (\sum_{r \in \mathcal{R}} \sum_{k \in \mathcal{B}} (C_{\text{r}\Gamma_r} \Gamma_r^T \Gamma_r C_{\text{r}\Gamma_r} \Gamma_r - \cdots$</td>
<td>Replace $+ \text{ by } \times$</td>
<td></td>
</tr>
</tbody>
</table>
\( [D^{01\tau}D^{10}U - D^{01\tau}D^{10\tau}] \)

101

4

- \( - \gamma_1 D^{01\tau}\bar{u}'D^{10}U - \gamma_1 D^{01\tau}\bar{\bar{u}}' + D^{12\tau}D^{12}U - D^{12\tau}D^{12\tau} \)

Interchange superscripts

101

5

- \( - (\gamma_1 - \gamma_0) D^{21\tau}\bar{u}'D^{12}U - \gamma_1 D^{21\tau}D^{12\tau} \)

Interchange superscripts

101

9

- \( u^{17} D^{01}D^{10} = u^{17} D^{21}D^{12} = \ldots \)

Interchange superscripts

101

9

- \( D^{01\tau}u^{1}D^{10} = D^{017\tau}u^{1}D^{12} = 0 \)

Interchange superscripts
# Contents

I. Introduction .................................................. 1

II. Unrestricted Systems ........................................ 3
   A. System Mathematical Model ................................. 3
   B. The Dynamical Equations .................................. 8
   C. Kinematical Equations ..................................... 10
   D. System Specification ....................................... 10
   E. Subroutine MBDY Usage ..................................... 11
   F. A Five-Body System Simulation ............................. 13

III. Totally Linearized Systems ................................. 36
   A. The Equations ............................................... 36
   B. Subroutines MBDYTL and MBDYL ........................... 64
   C. Application of MBDYTL and MBDYL to the Five-Body System ... 65

IV. Partially Linearized Systems ............................... 79
   A. The Equations ............................................... 79
   B. Subroutine MBDYPL Usage .................................... 80
   C. Application of MBDYPL to the Five-Body System .......... 81

V. Summary and Conclusions .................................... 82

Appendix A. Subroutine MBDY Description ..................... 99

Appendix B. Summary of MBDYTL, MBDYL User Requirements .... 108

Appendix C. Summary of MBDYPL User Requirements ............ 120

References ........................................................ 130

Figures

1. Eleven rigid bodies interconnected by ten hinges ........... 4
2. Alternate labeling schemes for a system of seven bodies .... 6
3. Augmented $\phi$ from Fig. 1 .................................. 8
4. A space vehicle idealized as a system of five hinge-connected rigid bodies .......................... 15
5. Single-axis attitude control loop ............................ 16
6. Five-body system simulation using MBDY (Case 1) .......... 18
7. Results of Case 1 simulation using MBDY ................... 20
Contents (contd)

Figures (contd)

8. Results of Case 1 simulation using MBDY (components of \( \omega^c \); first 10 s) ........................................... 30
9. Case 1 simulation program using MBDY, with prescribed platform motion ........................................... 34
10. \( \gamma \), prescribed function of time (\( \gamma_t = -\gamma \)) ................................................................. 36
11. Results of Case 1 simulation with prescribed platform motion ................................................................. 37
12. Results of Case 1 simulation without control torques ................................................................. 47
13. Case 2 simulation program using MBDY ................................................................. 50
14. Results of Case 2 simulation ................................................................. 52
15. Case 2 simulation using MBDYTL ................................................................. 67
16. Results of Case 2 simulation using MBDYTL ................................................................. 69
17. Case 1 simulation program using MBDYPL ................................................................. 84
18. Results of Case 1 simulation using MBDYPL ................................................................. 86
19. Results of Case 1 simulation using MBDYPL (components of \( \omega^c \); first 10 s) ................................................................. 96
Abstract

This report describes several computer subroutines designed to provide the solution to minimum-dimension sets of discrete-coordinate equations of motion for systems consisting of an arbitrary number of hinge-connected rigid bodies assembled in a tree topology. In particular, these routines may be applied to (1) the case of completely unrestricted hinge rotations, (2) the totally linearized case (i.e., all system rotations are small), and (3) the mixed, or partially linearized, case. The use of the programs in each case is demonstrated using a five-body spacecraft and attitude control system configuration. The ability of the subroutines to accommodate prescribed motions of system bodies is also demonstrated. Complete listings and user instructions are included for these routines (written in Fortran V) which are intended as multi- and general-purpose tools in the simulation of spacecraft and other complex electromechanical systems.
Attitude Dynamics Simulation Subroutines for Systems of Hinge-Connected Rigid Bodies

I. Introduction

Since Hooker and Margulies (Ref. 1) published their derivation of the vector-dyadic equations of motion for an arbitrary number of point-connected rigid bodies in a topological tree, a number of computer programs (Refs. 2–7) based on these equations and similar generic formulations have come into wide use in the aerospace industry. Meanwhile, Hooker (Ref. 8) has devised a procedure for reducing the original Hooker–Margulies equations to a minimum-dimension set (i.e., equal to the number of degrees of rotational freedom) by summing selected subsets of the equations (paralleling Russell’s approach—Ref. 2) and dot-multiplying by unit vectors associated with the connecting joint geometry.

In Reference (9) a variation of Hooker’s procedure was used to derive a minimum-dimension set of equations of motion for a hinge-connected system of rigid bodies. (It is always possible to substitute a hinge-connected system for a point-connected system by introducing auxiliary bodies and without changing the dimension of the problem.) The objective was, having derived this set of discrete-coordinate equations in a highly systematic and detailed way, to (1) isolate certain subsets of the system’s rigid bodies that may be considered collectively as a flexible substructure, (2) linearize in the variables describing relative motions of contiguous bodies within the substructure, and (3) transform from substructure discrete variables to distributed (modal) coordinates which may then be truncated for computational efficiency. The result is, in general, a new kind of hybrid-coordinate (discrete + distributed) formulation which can accommodate large gross deformations of an appendage even though its local deformations are small.
However, while computer subroutines which solve the hybrid-coordinate equations of Ref. 9 will be discussed in a subsequent report, this report will confine itself to descriptions of subroutines which solve only the discrete-coordinate equations of motion for the hinge-connected system. Specifically, it will describe and apply these programs as dynamic simulation tools for three possible situations: (1) the case in which all the system bodies are permitted large or unrestricted rotations (Section II), (2) the case in which all system rotations (and their derivatives) are small, allowing total linearization of the equations (Section III), and (3) the mixed case of large and small rotations for which partial linearization is appropriate (Section IV).

The programs, since they are based on the discrete-coordinate equations of Ref. 9, will differ from those developed in Ref. 4 in several significant respects. First, they will be of minimum dimension. Also, interbody coordinate transformations are accomplished internal to the programs, the user being required only to arbitrarily specify some nominal system configuration. Input has been simplified. Pains have been taken to minimize interbody coordinate transformations and other computations to improve speed. And, finally, the user may, if he desires, prescribe the motion of some system bodies rather than allow the subroutine to calculate it.

The subroutine approach to solving attitude dynamics equations is a logical consequence of the approach taken historically at JPL to develop computer programs (analog or digital) specialized to each particular control mode or portion of a control mode under study, e.g., the thrust vector control mode during propulsive trajectory corrections. In contrast, however, to the narrow specialization of each simulation program to specific control problems has been the extensive use of general-purpose simulation languages to facilitate the creation of such programs. Over the past decade, such digital simulation languages as MIDAS, MIMIC, DSL/90, and most recently, CSSL III (Continuous System Simulation Language) have proved extremely valuable since they relieve the programmer of the worry over numerical integration routines, and since they can properly sort the many, at times highly interrelated, computational statements which occur and thereby help to ensure a correct solution.

Many of these "simulators" also provide a powerful array of useful special functions, in addition to the standard Fortran library of mathematical and intrinsic functions, such as hysteresis loops, noise generators, time delays, pulse generators, logic gates, etc. In fact, provision is also made for inclusion into the language of any user-created functions or subroutines. An example of an extremely useful subroutine that has been made a part of the CSSL III processor is HCK (Hamilton-Cayley-Klein), a rigid-body attitude dynamics package, written by E. H. Kopf. HCK calculates the angular acceleration components of a single rigid body, given its angular velocities, applied torques, and inertial properties. The HCK parameters (i.e., Euler parameters) are used in the necessary kinematic calculations and provide body-to-inertial (or vice versa) coordinate transformations as desired. Euler angles may also be requested to indicate the angular position of the body with respect to the inertial reference frame.

Just like HCK, the subroutines to be described here calculate angular accelerations, except that they do so for every body in a system of hinge-connected rigid bodies. All integration operations are confined to the main CSSL III simulation program (which calls its centralized numerical integration routines as required). Thus,
performing no numerical integration and required only to compute angular acceleration components (given the necessary system states, forcing functions, and physical properties), the routines are intended simply as very specialized and yet powerful simulation tools for application to a wide variety of current space vehicle attitude control problems.

II. Unrestricted Systems

A. System Mathematical Model

Figure 1 illustrates a set of eleven hinge-connected rigid bodies, labeled according to a convention designed to facilitate the processing of the equations of motion. This example will be useful in illustrating and interpreting the general mathematical model and the labeling conventions adopted here for its description. (In reading the list of conventions and symbol definitions, frequent reference should be made to Fig. 1.) Only after entertaining the complete cast of symbols used in what follows will consideration be given to the much smaller list of symbols to which values must be assigned by the user of the computer programs.

The mathematical model consists of a set of \( n + 1 \) rigid bodies interconnected by \( n \) hinges; the indicated numbers of bodies and hinges imply a tree topology. Any interbody connection forces in addition to those at the \( n \) hinges must be treated as forces external to the entire system. (If there were a linear spring connecting the mid-point of body \( \mathcal{A}_6 \) in Fig. 1 to some point on body \( \mathcal{A}_{10} \), for example, it would be necessary to replace the spring with a pair of equal and opposite forces applied to the spring attachment points on \( \mathcal{A}_6 \) and \( \mathcal{A}_{10} \); the magnitude and direction of these forces would have to be expressed in terms of the unknown kinematical variables of the system.)

1. Definitions of fundamental symbols

Def. 1. Let \( n \) be the number of hinges interconnecting a set of \( n + 1 \) bodies.

Def. 2. Define the integer sets \( \mathcal{B} \triangleq \{0, 1, \cdots, n\} \) and \( \mathcal{D} \triangleq \{1, \cdots, n\} \).

Def. 3. Let \( \mathcal{A}_0 \) be a label assigned to one body chosen arbitrarily as a reference body, and let \( \mathcal{A}_0, \cdots, \mathcal{A}_n \) be labels assigned to the remaining bodies in such a way that if \( \mathcal{A}_j \) is located between \( \mathcal{A}_\alpha \) and \( \mathcal{A}_\beta \), then \( 0 < j < k \).

Def. 4. Define the dextral orthogonal sets of unit vectors \( b_1, b_2, b_3 \) so as to be imbedded in \( \mathcal{A}_k \) for \( k \in \mathcal{B} \), and such that in some arbitrarily selected nominal configuration of the system \( b^\alpha_k = b^\alpha_j \) for \( \alpha = 1, 2, 3 \) and \( k, j \in \mathcal{B} \).

Def. 5. Define \( \{b^\alpha\} \) as the column array of unit vectors\(^2\)

\(^1\)The word "hinge" as used here implies a connection which maintains a line common to both bodies; such a connection is sometimes called a line hinge.

\(^2\)Note that braces \( \{} \) are used in this report both to identify column arrays of vectors and to enclose sets of integers; the distinction is apparent as soon as the objects within the braces are identified as vectors (boldface) or scalars.
Def. 6. Let $\omega^0 = \{b^0\}^T \omega^0$ be the inertial angular velocity vector of $\mathcal{A}_0$, so that $\omega^0$ is the corresponding 3-by-1 matrix in vector basis $\{b^0\}$.

Def. 7. Let $c_k$ be the mass center of $\mathcal{A}_k$ for $k \in \mathcal{B}$ (see Fig. 1).

Def. 8. Let $p_k$ be a point on the hinge axis common to $\mathcal{A}_k$ and $\mathcal{A}_j$ for $j < k$ and $k \in \mathcal{B}$ (see Fig. 1).

Def. 9. Define the $k$th neighbor set $\mathcal{B}_k$ for $k \in \mathcal{B}$ such that $j \in \mathcal{B}_k$ if $\mathcal{A}_j$ is attached to $\mathcal{A}_k$.

Def. 10. For $j \in \mathcal{B}_k$, let $p^h_j = \{b^h\}^T p^h_j$ be the position vector with respect to $c_h$ of the labeled point on the hinge axis (either $p_i$ or $p_h$) which is common to $\mathcal{A}_h$ and $\mathcal{A}_j$, so that $p^h_j$ is the corresponding 3-by-1 matrix in vector basis $\{b^h\}$ (see Fig. 1, and note the special cases $p^{21} = p^{22} = p^{32} = p^{34} = p^{42} = 0$.)

Fig. 1. Eleven rigid bodies interconnected by ten hinges
Def. 11. Let \( g^k \overset{\Delta}{=} (b^k)^T g^k \) be a unit vector parallel to the hinge axis through \( \mathcal{A}_k \), so that \( g^k \) is the corresponding 3-by-1 matrix in vector basis \( \{b^k\} \).

Def. 12. For \( k \in \mathcal{D} \), let \( \gamma_k \) be the angle of a \( g^k \)-rotation of \( \mathcal{A}_k \) with respect to the body attached at \( \mathcal{A}_k \), that is, a rotation during which a right-handed screw fixed in \( \mathcal{A}_k \) advances in the direction of \( g^k \). Assign the value zero to \( \gamma_k \) for \( k \in \mathcal{D} \) when the system is in its arbitrarily chosen nominal configuration, for which \( b^k_0 = b^i_\alpha \), \( \alpha = 1, 2, 3; k, i \in \mathcal{B} \).

Def. 13. Let \( m_k \) be the mass of \( \mathcal{A}_k \) for \( k \in \mathcal{D} \).

Def. 14. Let \( I^k \overset{\Delta}{=} \{b^k\}^T I_k \{b^k\} \) be the inertia dyadic of \( \mathcal{A}_k \) for \( c_k \), so that \( I^k \) is the corresponding inertia matrix in vector basis \( \{b^k\} \), i.e.,
\[
I^k = \begin{bmatrix}
P_{11} & P_{12} & P_{13} \\
P_{12} & P_{22} & P_{23} \\
P_{13} & P_{23} & P_{33}
\end{bmatrix}
\]

Def. 15. Let \( F^k \overset{\Delta}{=} (b^k)^T F_k \) be the resultant vector of all forces applied to \( \mathcal{A}_k \) except for those due to interbody forces transmitted at hinge connections, so that \( F^k \) is the corresponding 3-by-1 matrix in vector basis \( \{b^k\} \). (Thus, any interbody forces due to spring connections, etc., which are separate from the \( n \) hinge connections would contribute to \( F^k \), as would all forces from sources external to the system.)

Def. 16. Let \( T^k \overset{\Delta}{=} (b^k)^T T_k \) be the resultant moment vector with respect to \( c_k \) of all forces applied to \( \mathcal{A}_k \) except for those due to interbody forces transmitted at hinge connections, so that \( T^k \) is the corresponding 3-by-1 matrix in vector basis \( \{b^k\} \).

Def. 17. Let \( \tau_k \) be the scalar magnitude of the torque component applied to \( \mathcal{A}_k \) in the direction of \( g^k \) by the body connected at \( \mathcal{A}_n \).

Note that Def. 3 allows a certain freedom in labeling the system bodies after the reference body \( \mathcal{A}_0 \) has been chosen, the requirement always being that label values increase as one progresses outward from \( \mathcal{A}_0 \) along any chain of bodies. Figure 2 shows, for example, two equally acceptable ways of labeling a seven-body system.

Def. 18. Let the tilde symbol \( (~) \) signify in application to a 3-by-1 matrix \( V \) with elements \( V_\theta \) \( (\theta = 1, 2, 3) \) transformation to a skew-symmetric 3-by-3 matrix \( \tilde{V} \). Thus, the matrix \( \tilde{V} \) has the expanded form
\[
\tilde{V} \overset{\Delta}{=} \begin{bmatrix}
0 & -V_3 & V_2 \\
V_3 & 0 & -V_1 \\
-V_2 & V_1 & 0
\end{bmatrix}
\]
Fig. 2. Alternate labeling schemes for a system of seven bodies

2. Definitions of derived symbols

**Def. 19.** Define the scalar \( \varepsilon_{sk} \) such that for \( k \in \mathcal{B} \) and \( s \in \mathcal{P} \),

\[
\varepsilon_{sk} = \begin{cases} 
1 & \text{if } s_k \text{ lies between } \mathcal{B}_k \text{ and } \mathcal{B}_l \\
0 & \text{otherwise}
\end{cases}
\]

The term *path elements* will be applied to the scalars \( \varepsilon_{sk} \), which are \( n(n+1) \) in number.

**Def. 20.** Let \( C'^j \) be the direction cosine matrix defined by

\[
\{b^r\} \overset{\Delta}{=} C'^j \{b^j\}
\]

for \( r, j \in \mathcal{B} \). In the nominal state, \( \gamma_k = 0 \) for \( k \in \mathcal{P} \), and all unit vectors of corresponding subscript are aligned, so that \( C'^j \) then is \( U \), the 3-by-3 unit matrix.

**Def. 21.** Define \( \mathcal{M} \overset{\Delta}{=} \sum_{k \in \mathcal{B}} m_k \), so that \( \mathcal{M} \) is the total system mass.
Def. 22. Define \( \mathcal{M}_{ij} \) the total mass on the branch of the system attached to \( \mathcal{E}_k \) and commencing with \( \mathcal{E}_j \).

Def. 23. For \( j \in \{\mathcal{E} \setminus k\} \), define \( L_k \) as the vector from \( c_k \) to the hinge point attached to \( \mathcal{E}_k \) and on the path leading from \( \mathcal{E}_k \) toward \( \mathcal{E}_j \). (In Fig. 1, for example, \( L_5 = L_6 = L_7 = 56 \) and \( L_3 = 0 \) for \( j \in \mathcal{E}_3 \).)

Def. 24. Define \( \{b_k\}^T \hat{D}^{bk} = \hat{D}^{bk} = \sum_{j \in \mathcal{E}} L_k m_j / \mathcal{M} \) for \( k \in \mathcal{E} \).

Def. 25. Let \( b_k \) be a point called the barycenter of \( \mathcal{E}_k \) and fixed relative to \( \mathcal{E}_k \) such that \( \hat{D}^{bk} \) is the position vector of \( c_k \) with respect to \( b_k \).

Def. 26. Define \( \{b_k\}^T \hat{D}^{bj} = \hat{D}^{bj} = \hat{D}^{bk} + L_{kj} \) for \( k, j \in \mathcal{E} \). Thus, \( \hat{D}^{bj} \) is the vector from \( b_k \) to the hinge point fixed in \( \mathcal{E}_k \) and on the path leading to \( \mathcal{E}_j \).

Def. 27. Define the dyadic
\[
\Phi^{bk} = \mathbf{I} + m_k (\hat{D}^{bk} \cdot \hat{D}^{bk} \mathbf{I} - \hat{D}^{bk} \hat{D}^{bk}) + \sum_{j \in \mathcal{E}} \mathcal{M}_{bij} (\hat{D}^{bj} \cdot \hat{D}^{bj} \mathbf{I} - \hat{D}^{bj} \hat{D}^{bj})
\]
where \( \mathbf{I} \) is the unit dyadic.

Def. 28. For \( j \in \{\mathcal{E} \setminus k\} \), define the dyadic
\[
\Phi^{bj} = -\mathcal{M}_{kj} (\hat{D}^{jk} \cdot \hat{D}^{jk} \mathbf{I} - \hat{D}^{jk} \hat{D}^{jk})
\]

3. Augmented bodies and barycenters. The barycenter \( b_k \) of body \( \mathcal{E}_k \) is defined in Def. 25 in mathematical terms which admit an interesting interpretation as a physical property of an abstraction called the augmented body. If one imagines the body \( \mathcal{E}_k \) to be augmented by the addition of a particle of mass \( \mathcal{M}_{ij} \) at the connection point of \( \mathcal{E}_k \) with \( \mathcal{E}_j \) (for \( j \in \mathcal{E} \)), then the result is called the \( k \)-th augmented body, here referred to as \( \mathcal{E}_k \). Since \( \mathcal{M}_{ij} \) is the total mass of the branch of bodies attached to \( \mathcal{E}_k \) commencing with \( \mathcal{E}_j \), the mass of each augmented body is \( \mathcal{M} \), the total system mass.

Definition 24 provides
\[
-\mathcal{M} \hat{D}^{bk} = \sum_{j \in \mathcal{E}} m_j L^{kj}
\]
which reveals the fact that \( -\hat{D}^{bk} \) is the position vector of the mass center of \( \mathcal{E}_k \) with respect to the mass center \( c_k \) of \( b_k \). Thus, the barycenter \( b_k \) is the mass center of the \( k \)-th augmented body \( \mathcal{E}_k \). With this interpretation comes the relationship
\[
\sum_{j \in \mathcal{E}} m_j \hat{D}^{kj} = m_k \hat{D}^{bk} + \sum_{j \in \mathcal{E} \setminus k} m_j \hat{D}^{kj} = 0
\]
Since \( \hat{D}^{bk} \) is the position vector from the barycenter \( b_k \) to the mass center \( c_k \) of \( \mathcal{E}_k \) and (from Def. 26) \( \hat{D}^{kj} \) is the position vector from \( b_k \) to a particle of mass \( \mathcal{M}_{ij} \) located at the connection point of \( \mathcal{E}_k \) leading to body \( \mathcal{E}_j \), for \( j \in \mathcal{E} \), one can interpret \( \Phi^{bk} \) from Def. 27 as the inertia dyadic of the augmented body \( \mathcal{E}_k \) with respect to the barycenter \( b_k \). Such physical interpretation is not possible for the dyadic \( \Phi^{bj} \) defined in Def. 28, since this dyadic involves vector \( \hat{D}^{bk} \) fixed in \( \mathcal{E}_i \) and vector \( \hat{D}^{bj} \) fixed in \( \mathcal{E}_k \).
In illustration of the concepts just described, Fig. 3 shows the augmented body $\mathcal{A}_a$ associated with body $\mathcal{A}_o$ in the system of Fig. 1.

B. The Dynamical Equations

Using the symbols defined in the preceding section, the minimum-dimension set of equations for the rotational motion of the hinge-connected system (i.e., with $n$ line hinges) can be written in vector-dyadic form as

$$\sum_{k \in \mathcal{B}} [T_k + \sum_{j \in \mathcal{B}} D^{kj} \times F^j - \Phi^{kk} \cdot \dot{\omega}^k - \omega^k \times \Phi^{kk} \cdot \omega^k + \sum_{k \in \mathcal{B} - k} D^{kj} \times \ddot{D}^{jk}] = 0$$

(1a)

$$\tau_r + \mathbf{g}^r \cdot \sum_{k \in \mathcal{B}} \epsilon_{rh} [T_k + \sum_{j \in \mathcal{B}} D^{kj} \times F^j - \Phi^{kk} \cdot \dot{\omega}^k - \omega^k \times \Phi^{kk} \cdot \omega^k + \sum_{k \in \mathcal{B} - k} D^{kj} \times \ddot{D}^{jk}] = 0 \quad r \in \mathcal{B}$$

(1b)

This result is, of course, the same as that derived by Hooker (Ref. 8) and proved in Ref. 9. Equation (1b) provides $n$ scalar equations, and the vector equation (1a) completes the minimum-dimension set of $n + 3$ scalar equations. By substituting the following kinematic expansions,

$$\ddot{D}^{jk} = \dot{\omega}^j \times D^{jk} + \omega^j \times (\omega^i \times D^{ik})$$

(2)

and

$$\dot{\omega}^i = \dot{\omega}^i + \sum_{s \in \mathcal{B}} \epsilon_{sij} (g^{ir} \gamma_s + \omega^s \times g^{sr} \gamma_s)$$

(3)

the vector-dyadic equations can be written in the form:

$$\mathbf{a}^{oo} \cdot \dot{\omega}^o + \sum_{s \in \mathcal{B}} a_{os} \gamma_s = \sum_{k \in \mathcal{B}} A^k$$

(4a)

and (for $r \in \mathcal{B}$)

$$\mathbf{a}^{oo} \cdot \dot{\omega}^o + \sum_{s \in \mathcal{B}} a_{rs} \gamma_s = g^{r} \cdot \sum_{k \in \mathcal{B}} \epsilon_{kr} A^k + \tau_r$$

(4b)
where

\[ E = \sum_{k \in \mathcal{S}} \sum_{j \in \mathcal{S}} (D^{ij} \cdot U - D^{kj} D^{ij}) \]

Def. 30. \( a^{ij} \equiv \sum_{k \in \mathcal{S}} \left( \varepsilon_{ijk} D^{ij} - \eta_{ijk} \sum_{j \in \mathcal{S}} (D^{ij} \cdot D^{kj} u - D^{ik} D^{ij}) \right) \cdot g^s \)

Def. 31. \( a^{ij} \equiv g^s \sum_{k \in \mathcal{S}} \varepsilon_{ijk} \cdot g^s \)

Def. 32. \( a_{rs} \equiv g^s \sum_{k \in \mathcal{S}} \varepsilon_{rsk} \cdot g^s \)

Def. 33. \( A^k \equiv T^k + \sum_{i \in \mathcal{S}} D^{ki} \times F^i - \omega^k \times \Phi^{ik} \cdot \omega^k \)

A matrix equivalent to Eqs. (4a) and (4b) may be obtained by defining the following symbols:

Def. 34. Define the 3-by-3 matrix in vector basis \( b^k \) as \( a_{00} \equiv \{ b^0 \} \cdot a^{00} \cdot \{ b^0 \}^T \).

Def. 35. For \( s \in \mathcal{S} \), define the 3-by-1 matrices in basis \( b^s \) as \( a_{0s} \equiv \{ b^s \} \cdot a^{00} \), and let \( a_{0s} \equiv a_{0s}^\omega \).

Def. 36. For \( k \in \mathcal{S} \), define the 3-by-1 matrices in basis \( b^k \) as \( E_k \equiv \{ b^k \} \cdot A^k \).

In terms of these symbols, the system matrix equation can be written as

\[
\begin{bmatrix}
a_{00} & a_{01} & a_{02} & \cdots & a_{0n} \\
a_{10} & a_{11} & a_{12} & \cdots & a_{1n} \\
a_{20} & a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
a_{n0} & a_{n1} & a_{n2} & \cdots & a_{nn}
\end{bmatrix}
\begin{bmatrix}
\gamma^0 \\
\gamma^1 \\
\gamma^2 \\
\gamma^3 \\
\gamma^n
\end{bmatrix} =
\begin{bmatrix}
\sum_{k \in \mathcal{S}} E_k \\
g^s \sum_{k \in \mathcal{S}} \varepsilon_{1k} E_k + \tau_1 \\
g^s \sum_{k \in \mathcal{S}} \varepsilon_{2k} E_k + \tau_2 \\
g^s \sum_{k \in \mathcal{S}} \varepsilon_{nk} E_k + \tau_n
\end{bmatrix}
\]

(5)

The coefficient matrix on the left side of (5) is a real, nonsingular, symmetric matrix which varies with time as the kinematical variables \( \gamma_1, \cdots, \gamma_n \) change in...
value. In general, then, the solution to this set of differential equations requires (e.g., when using numerical integration) repeated solutions of the matrix equation, (i.e., at every integration step).

If one wishes to replace any variable $y_j$ ($j \in \mathcal{P}$) by a known function of time, then one can remove $\ddot{y}_j$ from the angular acceleration matrix on the left side of Eq. (5), removing also the corresponding row and column of its coefficient matrix, and then rewrite on the right side those terms just deleted by the removal of the column on the left. This will be performed automatically by the subroutines if the user desires.

Equation (5) constitutes a complete set of rotational dynamics equations, but they are not fully descriptive of the system motion until they are augmented by control equations specifying $\tau_1, \cdots, \tau_n$ and certain external forces and torques, and augmented also by kinematical equations as provided in the next section.

C. Kinematical Equations

The kinematical variables adopted in the preceding sections are as follows: $y_k$ for $k \in \mathcal{P}$ (Def. 12); $C^{ri}$ for $r, j \in \mathcal{B}$ (Def. 20); and $\omega = (b^0) \cdot \omega$ (Def. 6). Although the equations of motion have been expressed in terms of these quantities, the latter are not all independent. Relationships among kinematical variables developed in this section must therefore either be considered in conjunction with the dynamical equations or be substituted into them to remove redundant variables whenever a solution is sought.

The direction cosine matrix $C^{ri}$ (Def. 20) relates sets of orthogonal unit vectors fixed in $\mathcal{B}_i$ and $\mathcal{B}_j$, and hence depends upon those angles $\gamma_{ij}$ for which $\mathcal{B}_i$ lies between $\mathcal{B}_i$ and $\mathcal{B}_j$, and also upon the corresponding unit vectors $g^0$ defining the intervening hinge axes. For the special case in which $\mathcal{B}_i$ and $\mathcal{B}_j$ are contiguous and $j < r$, it is always possible (see Ref. 9) to express $C^{ri}$ (and $C^{jr}$) in terms of the single angle $\gamma_r$ and the single matrix $g^0$, as follows:

$$C^{ri} = U \cos \gamma_r - \tilde{g}^0 \sin \gamma_r + \tilde{g}^0 g^0 (1 - \cos \gamma_r)$$

and

$$C^{jr} = U \cos \gamma_r + \tilde{g}^0 \sin \gamma_r + \tilde{g}^0 g^0 (1 - \cos \gamma_r) = (C^{ri})^T$$

Definitions 34, 35, and 36 make clear that all $n + 3$ scalar equations in (5) are described in the vector basis $\{b^0\}$ of the reference body. This is for the purpose of minimizing the number of coordinate transformation computations that must be performed by the subroutines. Thus, it becomes unnecessary to calculate $C^{ri}$ for all $r \in \mathcal{B}$, $j \in \mathcal{B}$. It is only required that $C^{ri}$ be determined where $\mathcal{B}_i$ and $\mathcal{B}_j$ are contiguous and, since $C^{0r} = C^{0i}C^{ir}$, to then derive matrices $C^{0r}$ for $r \in \mathcal{P}$. An algorithm for accomplishing this task is described in Appendix A.

D. System Specification

Once the appropriate equations from the three preceding sections are embodied in a computer program, the user need not deal directly with all of the symbols and concepts introduced in the formulation of these equations. He can provide the computer with the very limited body of input required to specify his system, and then concentrate on interpreting the numerical integration output.
It is assumed that the engineer in his wisdom and experience has devised for his physical system a mathematical model consisting of a system of \( n + 1 \) bodies interconnected by \( n \) hinges (such as that illustrated in Fig. 1). He has labeled the bodies as indicated by Def. 3 (Section IIA), and fixed in his mind a nominal configuration (for which \( y_3 = 0 \) for \( k \in \mathcal{B} \) and \( b_i^k = b_i^j \) for \( i, k \in \mathcal{B} \) and \( a = 1, 2, 3 \)). One final aid, however, will be needed to communicate the system's interconnection pattern to the computer subroutine.

Def. 37. For \( k \in \mathcal{B} \), let \( h_k \) denote the index of the body attached to \( \phi_k \) and on the path leading to \( \phi_0 \). The scalars \( h_k \) will be termed "connection elements.” (In Fig. 1, \( h_1 = 0, h_2 = 1, h_3 = 2, h_4 = 3, h_5 = 0, h_6 = 5, h_7 = 6, h_8 = 7, h_9 = 8, \) and \( h_{10} = 6 \).)

With the definition of \( h_k \), it is now possible to specify a complete set of input information for the subroutine:

**Subroutine input (required)**

1. The integer \( n \) (see Def. 1).
2. The \( n \) connection elements \( h_k \) ( \( k \in \mathcal{B} \)) (see Def. 37).
3. For \( k \in \mathcal{B} \) and \( j \in \mathcal{B}_k \), the 3-by-1 matrices \( p^{kj} \) (see Def. 10).
4. The \( n \) 3-by-1 matrices \( g_k \) for \( k \in \mathcal{B} \) (see Def. 11).
5. The \( n + 1 \) masses \( m_k \) for \( k \in \mathcal{B} \) (see Def. 13).
6. The \( n + 1 \) 3-by-3 inertia matrices \( I_k \) for \( k \in \mathcal{B} \) (see Def. 14).
7. For \( k \in \mathcal{B} \), the \( 2(n + 1) \) 3-by-1 matrices \( F^k \) and \( T^k \) (seeDefs. 15 and 16).
8. For \( k \in \mathcal{B} \), the \( n \) scalar values \( \tau_k \) (see Def. 17).
9. Values for the \( 2n \) scalars \( y_k \) and \( \dot{y}_k \) for \( k \in \mathcal{B} \) (see Def. 12) and the three elements of the angular velocity matrix \( \omega \) (see Def. 6).

**Subroutine input (optional)**

Any kinematically prescribed variables, such as \( \gamma_j(t) \) for some \( j \in \mathcal{B} \). As noted in Section IIB, it is possible to modify Eq. (5) so as to accommodate the substitution of an explicit function of time for a variable, such as \( \gamma_j(t) \), which in Eq. (5) is treated as an unknown. If this option is elected, the computer program input must include the specification of variables to be prescribed and appropriate values for \( \gamma_j, \dot{\gamma}_j, \) and \( \ddot{\gamma}_j \).

No additional input is required. The subroutine is programmed to construct numerically all intermediate concepts (such as the path elements \( \varepsilon_{ik} \) for \( k \in \mathcal{B}, s \in \mathcal{B} \), the barycentric position matrices \( D^{ik} \) and \( D^{ij} \) for \( k, i \in \mathcal{B} \), the augmented body inertia matrices \( \Phi^{ik} \), etc. It can then evaluate numerically the functions displayed in Eq. (5) and provide the solution for \( \phi^0 \) and \( \dot{\gamma}_i, \dot{\gamma}_j, \ldots, \dot{\gamma}_n \). The subroutine and explicit details of its use are described in the following section.

**E. Subroutine MBDY Usage**

The computer subroutine, called MBDY, which provides the solution to Eq. (5), i.e., the solution to the unrestricted rotations of an arbitrary, hinge-connected, rigid-body tree system, may be exercised by means of either of two call statements.
The first of these initializes the program, supplying it with data which will remain constant throughout the dynamic simulation run.

**Initializing Call Statement**

```
CALL MBDY(NH,H,MB,MA,PB,PA,G,PI)
```

where

- \( NH \) = the integer \( n \) = number of system hinges.
- \( H(k) \) = array containing the connection elements \( h_k, k \in \mathcal{E} \)
  \( H(1) = h_1, H(2) = h_2, \cdots, H(n) = h_n \)
- \( MB(j) \) = array of reference body \( (\mathcal{E}_b) \) inertial constants; \( j = 1, \cdots, 7 \)
  \( \text{Specifically: } MB(1) = I_{b1}, MB(2) = I_{b2}, MB(3) = I_{b3}, \)
  \( MB(4) = -I_{b4}, MB(5) = -I_{b5}, MB(6) = -I_{b6}, \) and \( MB(7) = m_b \)
- \( MA(i,j) \) = array of appendage body \( (\mathcal{E}_a, \cdots, \mathcal{E}_a) \) inertial constants; \( i \in \mathcal{E}, j = 1, 2, 3 \)
  \( \text{Thus, } MA(i,1) = I_{a1}, MA(i,2) = I_{a2}, \cdots, MA(i,7) = m_a \)
- \( PB(i,j) \) = array containing elements of \( p^0; i \in \mathcal{B}_0, j = 1, 2, 3 \)
- \( PA(i,j,k) \) = array containing elements of \( p^i; i \in \mathcal{E}, j \in \mathcal{B}_1, k = 1, 2, 3 \)
  \( \text{Exception! If } j < i, \text{ set } PA(i,j,k) = p^{ij}. \text{ Example: } PA(3,3,1) = p_{33}^1. \)
  \( \text{All } PA(i,j,k), \text{ where } j < i, \text{ will be ignored.} \)
- \( G(i,j) \) = array containing elements of \( g^i; i \in \mathcal{E}, j = 1, 2, 3 \)
- \( PI(i) \) = array of prescribed variable indicators; \( i \in \mathcal{E} \)
  \( \text{If } y_i \text{ is a prescribed variable, } PI(i) = 1. \text{ Otherwise, } PI(i) = 0. \)

The statement CALL MBDY(NH,H,...) need only be executed once prior to a simulation run, but subsequently the subroutine must be entered (at every integration step) to compute the angular accelerations \( \dot{\phi}, \dot{y}_1, \cdots, \dot{y}_n \), i.e., the system's dynamic response to its forcing functions. This is accomplished by executing the "dynamic" call statement.

**Dynamic Call Statement**

```
CALL MRATE(NH,TH,TB,TA,FB,FA,GM,GMD,GMD,WQ,WDOT)
```

where

- \( NH \) = the integer \( n \) = number of system hinges
- \( TH(i) \) = array containing the hinge torques, \( \tau_i; i \in \mathcal{E} \)
- \( TB(j) \) = array containing the elements of \( T^0; j = 1, 2, 3 \)
- \( TA(i,j) \) = array containing the elements of \( T^i; i \in \mathcal{E}, j = 1, 2, 3 \)
- \( FB(j) \) = array containing the elements of \( F^0; j = 1, 2, 3 \)
- \( FA(i,j) \) = array containing the elements of \( F^i; i \in \mathcal{E}, j = 1, 2, 3 \)
- \( GM(i) \) = array of the angles, \( y_i; i \in \mathcal{E} \)
- \( GMD(i) \) = array of the angular velocities, \( \dot{y}_i; i \in \mathcal{E} \)
- \( GMDD(i) \) = array of the prescribed angular accelerations, \( \ddot{y}_i; i \in \mathcal{E} \)
\( WO(j) = \) array containing the components of \( \omega^o; j = 1, 2, 3 \)

\( \text{WDOT}(j) = \) solution vector containing the elements of \( \dot{\omega}^o \) and \( \dot{\gamma}_i; \)
\( i = 1, \ldots, n + 3 \) (WDOT(1) = \( \dot{\omega}^o \), WDOT(2) = \( \dot{\omega}_p \), WDOT(3) = \( \dot{\omega}_r \), WDOT(4) = \( \dot{\gamma}_1 \), WDOT(5) = \( \dot{\gamma}_2 \), \ldots, WDOT(n + 3) = \( \dot{\gamma}_n \)  

The solution vector array, WDOT, is returned to the main calling program by the subroutine and contains the current computed value for the reference body's angular acceleration relative to the inertial frame \( \omega^o \) and the relative angular accelerations \( \gamma_i \) of the remaining appendage bodies.

To summarize, the call to MRATE supplies the subroutine with current instantaneous values for hinge torques and externally applied torques and forces on the system. Explicit expressions for computing the \( T^i, F^i, \) and \( \tau_i, \) often as functions of the \( \gamma_i \) and \( \dot{\gamma}_i \) and other system or control variables, are located in the main simulation program (see sample problem which follows). Current values of \( \omega^o, \gamma_i, \) and \( \dot{\gamma}_i \) are continuously produced by the main program's numerical integration operators (operating on the solution vector components, WDOT(\( j \))) and are therefore always available for input to the subroutine. The \( \dot{\gamma}_i \) inputs to the subroutine are used only when the motion of certain system bodies is prescribed.

When the MBDY subroutine is used, the main calling program must contain Fortran V (or IV) statements which specify "type" and allocate storage for the variables and arrays being used and assign values to these variables and arrays. Some examples of this will be shown in the next section. However, the mandatory specification statements are listed here.

**Required Specification Statements**

\[
\begin{align*}
\text{INTEGER} & \quad NH, H(n), P(n) \\
\text{REAL} & \quad MB(7), MA(n,7), PB(n,3), PA(n,n,3), G(n,3), TH(n), \\
& \quad TB(3), TA(n,3), FB(3), FA(n,3), GM(n), GMD(n), \\
& \quad GMDD(n), WO(3) \\
\text{DOUBLE PRECISION} \quad \text{WDOT}(n + 3)
\end{align*}
\]

Also, in order that storage allocation for arrays internal to the subroutine MBDY be minimized (i.e., made consistent with \( n \) hinges and no more), the following Fortran V (or IV) statement must appear in the subroutine:

\[
\text{PARAMETER} \quad Q = n
\]

The proper placement of this statement in MBDY is shown in the subroutine listing (Appendix A).

**F. A Five-Body System Simulation**

To illustrate the application of the MBDY subroutine for the unrestricted motion case, the space vehicle configuration shown in Fig. 4 will be simulated. The vehicle
is idealized to consist of a rigid central body, which will be designated as the reference body \( \mathcal{A}_0 \), including the bus, solar panels, gas jets, sensors, communications equipment, etc.

Hinged to the reference body is a rotating hub, labeled \( \mathcal{A}_1 \), which in turn is hinged to platform \( \mathcal{A}_2 \), carrying a television camera. The platform hinge axes located by \( g^1 \) and \( g^2 \) are orthogonal, \( g^1 = b_0^1 \) (+Z-direction), and the rotation angles \( \gamma_1 \) and \( \gamma_2 \) are arbitrarily defined to be zero when the TV boresight unit vector \( u \) is in the Y-direction. Also, \( u \) is defined perpendicular to \( g^2 \). Thus, we may write

\[
u = -\sin \gamma_1 \cos \gamma_2 b_0^1 + \cos \gamma_1 \cos \gamma_2 b_2^2 - \sin \gamma_1 b_2^2 \]

\[
g^1 = b_0^1
\]

\[
g^2 = -b_1^1
\]

A large instrumented boom, \( \mathcal{A}_3 \), is also hinged to the reference body by the mutually orthogonal and coplanar axes defined by \( g^3 \) and \( g^4 \). (In this case, the intermediate rigid body, \( \mathcal{A}_3 \), will be assigned zero mass.) Therefore,

\[
g^3 = b_3^0
\]

\[
g^4 = b_4^0
\]

The following list of system parameters will be used to define the system and initialize the MBDY subroutine:

1. \( n = 4 \)
2. \( h_1 = 0, h_2 = 1, h_3 = 0, h_4 = 3 \)
3. \( p^{10} = [0 \ 0 \ -1.5]^T, p^{03} = [0 \ -1.20 \ 0]^T \) meters
   \( p^{10} = [0 \ 0 \ -0.75]^T, p^{12} = [0 \ -0.10 \ -0.75]^T \) meters
   \( p^{21} = [0 \ -0.22 \ 0.20]^T \) meters
   \( p^{10} = [0 \ 0 \ 0]^T, p^{14} = [0 \ 0 \ 0]^T \) meters
   \( p^{14} = [0 \ 3.3 \ 0]^T \) meters
4. \( g^1 = [0 \ 0 \ 1]^T \) meters
   \( g^3 = [-1 \ 0 \ 0]^T \) meters
   \( g^4 = [1 \ 0 \ 0]^T \) meters
   \( g^4 = [0 \ 0 \ 1]^T \) meters
5. \( m_0 = 410 \text{ kg} \)
   \( m_1 = 6.8 \text{ kg} \)
   \( m_2 = 57.5 \text{ kg} \)
   \( m_3 = 0 \)
   \( m_4 = 10.7 \text{ kg} \)
This completes the list of constants that are needed to characterize the system. It remains then to develop expressions for the dynamic forcing functions $T^k$, $F^k$, and $r^k$.

Control of the vehicle shown in Fig. 4 is to be maintained by firing pairs of gas jets located on the reference body to produce, ideally, a pure torque couple on that body. It will be assumed in the simulations to follow that this is the case and that no net force is applied to any of the system bodies; i.e., $F^k = 0$ for $k \in \mathcal{B}$. The control torque $T^0$ can be expressed by

$$T^0 = [L_1 f_1 \quad L_2 f_2 \quad L_3 f_3]^T$$

where the control functions $f_1$, $f_2$, and $f_3$ each may take on the values $\pm 1$ or $0$, depending on the single-axis control law defined in Fig. 5. (All other $T^k$ for $k \in \mathcal{B}$ are assumed to be zero.) The control law is based on position feedback signals derived from celestial sensors (i.e., sun sensors, star trackers) as well as rate feedback derived from gyro measurements. If the sum of these feedback errors is

![Fig. 4. A space vehicle idealized as a system of five hinge-connected rigid bodies](image-url)
greater than some arbitrary value \((\pm \theta_{DB} \text{ in Fig. 5})\), a switching amplifier fires the gas jets to apply corrective torques. A list of values assigned to the reference body attitude control parameters follows:

\[
\theta_{DB} = 0.005 \text{ rad} \quad L_1 = 0.23 \text{ n-m} \\
\tau_B = 2.0 \text{ s} \quad L_2 = 0.21 \text{ n-m} \\
\tau_R = 2.0 \text{ s} \quad L_3 = 0.31 \text{ n-m} \\
TM = 0.020 \text{ s}
\]

Finally, it is necessary to characterize the hinge torques \(\tau_k (k \epsilon \mathcal{O})\). Since the boom will not be articulated, the boom hinge torques can be described by

\[
\tau_3 = -K_3 \gamma_3 - B_3 \dot{\gamma}_3 \\
\tau_4 = -K_4 \gamma_4 - B_4 \dot{\gamma}_4
\]

where

\[
K_3 = 2000 \text{ n-m/rad}, \quad B_3 = 10 \text{ n-m-s/rad} \\
K_4 = 2000 \text{ n-m/rad}, \quad B_4 = 10 \text{ n-m-s/rad}
\]

However, since the TV platform will be articulated relative to its nominal pointing direction, it is convenient to introduce the variables \(\gamma_{1c}\) and \(\gamma_{2c}\) to represent the commanded position of the platform. The hinge torques can then be written as

\[
\tau_1 = -K_1 (\gamma_1 - \gamma_{1c}) - B_1 \dot{\gamma}_1 \\
\tau_2 = -K_2 (\gamma_2 - \gamma_{2c}) - B_2 \dot{\gamma}_2
\]

where

\[
K_1 = 3500 \text{ n-m/rad}, \quad B_1 = 20 \text{ n-m-s/rad} \\
K_2 = 3500 \text{ n-m/rad}, \quad B_2 = 20 \text{ n-m-s/rad}
\]

1. Case 1—large platform maneuver. The first simulation of the five-body system of Fig. 4 and its controller will show its dynamic response to a large platform maneuver.
maneuver. Initially, the platform is positioned at $\gamma_1 = 218$ deg and $\gamma_2 = -30$ deg. All other system variables, i.e., angular positions and rates, are assumed initially at zero. At $t = 0$, the command is issued to rotate the platform (about both axes simultaneously) at the constant rate of 1 deg/s to the position $\gamma_1 = 168$ deg, $\gamma_2 = 20$ deg. The program used to simulate this maneuver is shown in Fig. 6.

Note that the required MBDY specification statement REAL MB(7), ... has been replaced by its equivalent in the CSSL III language, i.e., ARRAY MB(7), ...

Also, note that the HCK routine mentioned in Section I is used to establish the position of the reference body in the inertially fixed frame, and it thereby provides a means for calculating $\theta_1$, $\theta_2$, and $\theta_3$, the attitude angles measured by celestial sensors fixed to the reference body.

Three other CSSL III built-in functions are also used in the simulation. The function “SWAMP” simulates the switching amplifier shown in Fig. 5. “STEP” provides the unit step function when the independent variable, TIME, is greater than the specified constant. And finally, “INTEG($a_1$, $a_2$)” signifies the integration operation, where $a_1$ is the integrand and $a_2$ is the initial condition.

All arithmetic statements are in Fortran, although CSSL III allows several statements on a single line if separated by a “$.” Variables to be plotted at every communication interval CI are listed in the PREPAR statement. Printed variables are listed in the OUTPUT statement.

The statement “CALL MBDY(NH,...)” is located in the INITIAL section and is therefore executed only once, i.e., prior to the dynamic calculations. However, “CALL MRATE(NH,...)” is in the DERIVATIVE section and is thus executed at every integration step, as required.

Results of the simulation described in Fig. 6 are shown in the computer plots of Fig. 7. All variables plotted are in terms of radians, radians per second, and seconds.

Notice that all three components ($W_1$, $W_2$, $W_3$) of the reference body angular velocity $\omega$, show evidence of platform hinge vibrations immediately following platform acceleration or deceleration (i.e., at $t = 0$ and $t = 50$ s). Only $W_1$ and $W_3$ show any significant evidence of the lower-frequency boom vibrations, since the boom is nominally aligned with the reference body’s “2” axis. Attitude angles $\theta_1$, $\theta_2$, and $\theta_3$ (THET1, THET2, THET3) have obviously been effectively confined to the \pm 0.005-rad deadband by the control system. Platform and boom angular rate and position responses clearly show the transients at $t = 0$ and $t = 50$ s. Some platform vibration is discernible in the boom responses.

Figure 8 repeats the transient solutions for $W_1$, $W_2$, and $W_3$ over just the first 10 s in order to display the response in better detail. The $W_2$ response indicates that control jets began pulsing at about $t = 4.6$ s in that axis, while pulsing in the other axes was not required until much later.

2. Case 1 with prescribed motion. In order to demonstrate the utility of the MBDY subroutine’s option for prescribed motion, Case 1 was rerun with the platform rotations $\gamma_1$ and $\gamma_2$ prescribed. As indicated in Section IIE, the motion of any
PROGRAM 5-BODY SYSTEM SIMULATION USING MBDY -- CASE 1

INTEGER NH,FII4LsH$)
ARRAY MB)j,MA)R,1) ,pacq*j) ,PA(q,3)
G('1,3) ,TbI3)
ARRAY FB(),FA(4,3),THi(') ,GM()JGMUqi ,.GMUD('4) ,A(3)
DOUdLE PRECISUO ADOOT

DATA H/OsIO,3/PI/OU#0,O/
DATA /I.31.,4OI'.I9,R6'I/
DATA MA(2,6)/-.Sq/MA)2o7)/57.5/
DATA MA(4, 1)/2-/.2/MA ,*2)/.2/MAIR,3)/27.2/t4AII
DATA Pt3(I,3)/-I.5/Pk3(3s2)/-I.2/
DATA PA)),) ,3)/-.75/PA) ) 2,2j/-. IC/PA(1I 213)/-.7/
DATA PA(Z2o2)/-.22/'PA)Z,2.3)/.2O/
DATA 6 ,)I/(,)-.G3I/.c(43,.CONSTANT Tm'0u 2
0 DB~po0b .TN=2.o
CONSTANT TFINALS6O.tCLKfIM=IOO.
CONSTANT LI.*2J,L2-.2I,L3=*31
CONSTANT SLRATS.01745329
CONSTANT PHIZ=O. ,THETZ=De PSIZ=Q.
CONSTANT lI,'2c.3U.
CONSTANT c1I 3
OC.KZ35D~pI3E2oCO.,KN=2O0.
CONSTANT 3I120$*B2u2
9
,0,
9
=B3=I0.pB5qI
9.
CONSTANT Icl=2
1
18., IC23-30.
NH=9
RD=57.2957795 $ GM1=1.4D0 3 GM2=C1=1C1/RD $ GM2C1=1C2/RD
GM1=1C1/RD $ GM2=1C2/RD
CALL MBDY(NH,NH,MB,MA,PB,PA,G,PF)
END

DYNAMIC
IF(TIME*GL*TFINAL) GO TO SI
DERIVATIVE BODYS
VARIABLE TIME=Q.
CINTERVAL C1=06
XE"OR R W1=1E-6,2=1E-6,3=1E-6,GM3=1E-4
STPCLK CLKTIM
OUTPUT SU,NID,A20,N1,WZ,THET1,THET2,THET3,.. IN1,IN2,IN3,GM1,GM2,GM3,GM4,PL1,PL2
PREPAR W1,A3,THET1,THET2,THET3,GM1,GM2,GM3,GM4,GM5,GM6
GM1=GM1
GM2=GM2
GM3=GM3
GM4=GM4
GM5=GM5
GM6=GM6
GMD(1)=GMD(1)+GM(1)*GMD(2)=GMD(2)+GM(2)*GMD(3)=GMD(3)+GM(3)*GMD(4)=GMD(4)+GM(4)*GM(4)

NOSORT
NO(1)=N 5 NO(2)=N 2 $ NO(3)=N 3
PL2=R6=GM1 $ P2=MD=GM2
GM(1)=GM1 $ GM(2)=GM2 $ GM(3)=GM3 $ GM(4)=GM4
GMD(1)=GMD(1)+GM(1)*GMD(2)=GMD(2)+GM(2)*GMD(3)=GMD(3)+GM(3)*GMD(4)=GMD(4)+GM(4)*GM(4)

COMMENT
COMMENT CELESTIAL-SENSOR-MEASURED ANGLES
COMMENT
DI2=CHCK(MATRIX,PI2,PX,PA,PB,PC)
NXO,NYO,NZMDHCK(1TOB,0=0,1=1,0=1,0=0)
LXO,LYO,LO=HCK(1TOB,1=1,0=0,1=0,1=0)
THET1=ATAN2(NY0,NZ0) $ THET2=ATAN2(-NX0,NZ0)
THET3=ATAN2(-LYO,LO)

Fig. 6. Five-body system simulation using MBDY (Case 1)
Fig. 6 (contd)
Fig. 7. Results of Case 1 simulation using MBDY
Fig. 7 (contd)
Fig. 7 (contd)
Fig. 7 (contd)
Fig. 7 (contd)
Fig. 7 (contd)
Fig. 7 (contd)
Fig. 8. Results of Case 1 simulation using MBDY (components of $\omega_0$; first 10 s)
body $\delta_i$, ($i \in \mathcal{G}$) may be prescribed relative to an adjoining body $\delta_j$, where $j = h$, (Def. 37), by setting $P(i) = 1$ and supplying the appropriate values or function for $\gamma_i$ (GMDD($i$)). This was done for the platform bodies $\delta_1$ and $\delta_2$, resulting in the simulation program of Fig. 9. Notice that the platform command angles have been replaced by time functions for GMDD(1) and GMDD(2) (using the CSSL III "RAMP" function). The GMDD(1) function is plotted in Fig. 10.

The expressions for the hinge torques $\tau_1$ and $\tau_2$ ($TH(1), TH(2)$) have also been removed from the original program (although it was not necessary) to emphasize the fact that they are not used when $\gamma_1$ and $\gamma_2$ are prescribed.

Figure 11 shows the simulation results. It is clear that the overall system response is virtually the same as that obtained from the first simulation (Fig. 7), except for the absence of the characteristic platform hinge vibrations. Since the simulation with prescribed platform motion took only $4\frac{1}{2}$ min of computation time, compared with $7\frac{1}{2}$ min for the first run, the usefulness of the prescribed motion option is well demonstrated. In this case, not only was the order of the system equation reduced, but the absence of the high-frequency platform vibrations helped to speed up the numerical integration process.

Of course, some caution must also be exercised when employing prescribed variables to make sure that the dynamics being eliminated are not important. In the sample problem considered here, the platform hinge vibrations could well have triggered gas jet firings before damping out if $\tau_x$ had been larger and/or the initial conditions had been different. The absence of these vibrations might then have resulted in an entirely different time history of the reference body attitude angles.

One additional simulation of the Case 1 platform maneuver was performed without the application of control torques. In the absence of external torques, the system angular velocities should return to their initial values (i.e., zero) after the platform maneuver is completed (Conservation of Angular Momentum). Of particular interest are the reference body angular rates, which are given in Fig. 12. Although vibrations are still in evidence at $t = 60$ s, it is clear that the final average rates are zero, as they should be. (The small high-frequency disturbances which seem to occur randomly, particularly in the W2 and W3 responses, are due to numerical integration errors.)

3. Case 2—small platform maneuver. The subroutine MBDY, since it solves the complete and exact set of equations of motion, must certainly be used when relatively large angles and angular rates are expected at each hinge. However, if the hinge rotations or both the rotations and rotational rates are "small" at any hinge, some linearization of the equations of motion may be effected, often to good advantage in simplifying and speeding the computational task.

In subsequent sections of this report, modified versions of MBDY, employing some form of linearization, will be described. One version, MBDYTL, employs total linearization; i.e., all angular position and velocity responses in the system are assumed "small" in some sense. A sample simulation, using the five-body configuration of Fig. 4, will be performed later using MBDYTL, and it is therefore of interest to have for comparison purposes an "exact" solution to the same problem using MBDY.
Fig. 9. Case 1 simulation program using MBDY, with prescribed platform motion
COMMENT PSEUDO-RATE FEEDBACK SIGNAL AND SWITCHING AMP. DYNAMICS
COMMENT
IN1=THET1-IR*W1 $ IN2=THET2-IR*W2 $ IN3=THET3-IR*W3
AMP1=SWAMP(TIME*TM+DB*IN1)
AMP2=SWAMP(TIME*TM+DB*IN2)
AMP3=SWAMP(TIME*TM+DB*IN3)
COMMENT
COMMENT PRESCRIBED PLATFORM ACCELERATIONS
COMMENT
GMDD(1)=6.98136*(1-RAMPIO.*TIME1+2.*RAMPIO.*TIME1-...
RAMP(1)*TIME1+RAMP(50.*TIME1)-2.*RAMP(50.*TIME1)+RAMP(50.*TIME1+RAMP(50.*TIME1))
GMDD(2)=-GMDD(1)
COMMENT
COMMENT REFERENCE BODY EXTERNAL TORQUE
COMMENT HINGE TORQUES
COMMENT
TB1(1)=L1*AMP1 $ TB1(2)=L2*AMP2 $ TB1(3)=L3*AMP3
TH(1)=-K3*GM3-0.3*GM1D
TH(4)=-K4*GM4-0.4*GM4D
COMMENT
COMMENT SYSTEM ANGULAR ACCELERATIONS
COMMENT
CALL MRATE(NH,TH,THA,F6+F8A*GM+GMDD*W0+WD01)
W1D=WD01(11) $ W2D=WD01(12) $ W3D=WD01(13)
COMMENT
COMMENT SYSTEM ANGULAR RATES AND POSITIONS
COMMENT
W1=INTEG(WDOT(11)+W11)
W2=INTEG(WDOT(12)+W21)
W3=INTEG(WDOT(13)+W31)
GM1D=INTEG(WDOTF(1)+0.) $ GM2D=INTEG(WDOTF(2)+0.)
GM3D=INTEG(WDOTF(3)+0.) $ GM4D=INTEG(WDOTF(4)+0.)
GM1=INTEG(GM1D+GM11) $ GM2=INTEG(GM2D+GM21)
GM3=INTEG(GM3D+0.) $ GM4=INTEG(GM4D+0.)
COMMENT
COMMENT HCK PARAPEIER RATES AND POSITIONS
COMMENT
P20D=PADO+PBD0+PCOD=HCK1*HCK1*HCK1*HCK1*HCK1*HCK1*HCK1*HCK1*HCK1*HCK1*HCK1*HCK1
P3D=INTEG(P20D+P21)
P40=INTEG(PADO+P41)
P90=INTEG(PBDD+P91)
PC0=INTEG(PCOD+PC1)
END
END
END
TERMINAL
SI. CONTINUE
END
END

Fig. 9 (contd)
In Case 1, we have already seen that the reference body and boom responses are relatively small even though the platform angular excursions are substantial. If the platform maneuvers were also restricted to small values, this would seem the ideal case for which to apply MBDYTL and to obtain a check solution using MBDY. Results of the latter will be presented here.

In this case, it will be assumed that the platform again starts from the position $\gamma_1 = 218$ deg and $\gamma_2 = -30$ deg and that at $t = 0$ the command is issued to rotate to $\gamma_1 = 220$ deg at a commanded rate of 1 deg/s. Then, at $t = 5$ s, the command is issued to rotate about the other axis to the position $\gamma_2 = -25$ deg, at the same rate. A CSSL III program to simulate this maneuver sequence is listed in Fig. 13.

The simulation results are shown in Fig. 14. Obviously no gas jets were fired. These solutions will be compared in following sections to those obtained from linearized versions of MBDY.

III. Totally Linearized Systems

A. The Equations

When any justification can be found for the linearization of any of the kinematical variables appearing in Eq. (5), this step should be taken, since the mathematical and computational benefits of linearization are quite substantial. The primary immediate benefit is the elimination of variable terms from the coefficient matrix on the left side of Eq. (5). If $\omega$ and all angles $\gamma_1, \gamma_2, \ldots, \gamma_n$ and their time derivatives can be assumed to remain arbitrarily small, then the indicated coefficient matrix is formally constant, and much computer simulation time can be gained by computing its inverse only once, in advance of numerical integration. When the angles are expected to be small in an engineering sense, less rigorous arguments can be marshalled to gain the same computational advantage. However, more than sufficient reason for employing linearization can very often be found by experimental demonstration.
Fig. 11. Results of Case 1 simulation with prescribed platform motion
Fig. 11 (contd)
Fig. 11 (contd)
Fig. 11 (contd)
Fig. 11 (contd)
Fig. 11 (contd)
Fig. 11 (contd)
Fig. 12. Results of Case 1 simulation without control torques
Fig. 12 (contd)
PROGRAM 5-BODY SYSIEM SIMULATION USING MBDY --- CASE 2

INTEGER NH,NPI,MBY,PA
REAL PA,MA,MA1,MA2,PA1,PA2,PA3,MA11,MA21,PA11,PA21,PA31
REAL A1,A2,MA111,MA211,PA111,PA211
REAL TN,THETA,THET1,THET2,THET3,GM1,GM2,GM3
REAL W1,W2,W3,THETAN1,THETAN2,THETAN3

DATA NH=4,RO=57.2957795,GM1CI=IC1/RD $ GM2CI=IC2/RD $ GM1=IC1/RD $ GM2=IC2/RD
CALL MBDY(NH,MA,MB,PA,PI)

INITIAL
NH=4
RO;57.2957795 $ DR=1./RO & GM1CI=IC1/RD & GM2CI=IC2/RD
GM1CI=IC1/RD & GM2CI=IC2/RD
CALL MBDY(NH,MA,MB,PA,PI)

NOSORI
W1(1)=W1 $ W1(2)=W2 $ W1(3)=W3
PL1=RD $GM1 $ PL2=RD $GM2
GM1=GM1 & GM2CI=IC2/RD & GM1=GM1 & GM2=GM2

COMMENT
COMMENT CELESTIAL-SENSOR-MEASURED ANGLES

Fig. 13. Case 2 simulation program using MBDY
COMMENT RATE FEEDBACK SIGNAL AND SWITCHING AMP. DYNAMICS
COMMENT
IN1=-THE1-TR*W1 $ IN2=-THE2-TR*W2 $ IN3=-THE3-TR*W3
AMP1=SWAMP(TIME*TH0+IN1)
AMP2=SWAMP(TIME*TH0+IN2)
AMP3=SWAMP(TIME*TH0+IN3)
COMMENT PLATFORM POSITION COMMANDS
COMMENT
GM1CD=SLRAT1*STEP12.*TIME1 $ GM1C=INTEGR(GM1CD+GM1CI)
GM2CD=SLRAT1*STEP15.*TIME1 $ GM2C=INTEGR(GM2CD+GM2CI)
COMMENT
COMMENT REFERENCE Eddy External Torque
COMMENT Hinge Torques
COMMENT
TH11=-K1*GM1-G1*111+B1+GM1D
TH12=-K2*GM2-GM2C-B2+GM2D
TH13=-K3*GM3-E3+GM3D...
TH14=-K4*GM4-B4+GM4D
COMMENT
COMMENT SYSTEM ANGULAR ACCELERATIONS
COMMENT
CALL MRATE(NH1+TH1+TA+FB+FA+GM+GMDO+GW0+WDOT1
W10=WDOT11 $ W20=WDOT21 $ W30=WDOT31
COMMENT
COMMENT SYSTEM ANGULAR RATES AND POSITIONS
COMMENT
W1=INTEGR(WDOTI11)-W11
W2=INTEGR(WDOT21-1W21
W3=INTEGR(WDOT31-1W31)
GM1D=INTEGR(WDOT141)*0.1 $ GM2D=INTEGR(WDOT151)*0.1
GM3D=INTEGR(WDOT161)*0.1 $ GM4D=INTEGR(WDOT171)*0.1
GM1=INTEGR(GM1D+GM1I) $ GM2=INTEGR(GM2D+GM2I)
GM3=INTEGR(GM3D+GM3I) $ GM4=INTEGR(GM4D+GM4I)
COMMENT
COMMENT HCK PARAMETER RATES AND POSITIONS
COMMENT
PZCD=PADO+PBO+PCD=HCK+PZC+PBO+PAC+FBG+PAG+PZI
PZQ=INTEGR(PZQ+PZI)
PAC=INTEGR(PAC+PAC)
PCD=INTEGR(PCD+PCD)
PCQ=INTEGR(PCQ+PCQ)
END
END
END
TERMINAL
SI.. CONTINUE
END
END

Fig. 13 (contd)
Fig. 14. Results of Case 2 simulation
Fig. 14 (contd)
Fig. 14 (contd)
Fig. 14 (contd)
Fig. 14 (contd)
Fig. 14 (contd)
Fig. 14 (contd)
In any case, Eqs. (5) can be rewritten in the form

$$a_{\text{o}t} + \sum_{k \in \mathcal{K}} a_{\text{o}k} \gamma_k = \sum_{k \in \mathcal{K}} E_k$$  \hspace{1cm} (8a)

and

$$a_{\text{o}t} + \sum_{k \in \mathcal{K}} a_{ik} \gamma_k = g^{i}f \sum_{k \in \mathcal{K}} e_{ik} E_k + \tau_i$$ \hspace{1cm} (8b)

The next step is the linearization of Eq. (8) in the variables $\omega^o, \gamma_k$ and their time derivatives. To this end, we expand each symbol in Eq. (8) which may involve such variables into three parts, the first being free of these variables (indicated by overbar), the second being linear in the variables (indicated by overcaret), and the third containing terms above the first degree in the variables (indicated by three dots). In particular, we write (for any $i, k$)

$$C^{io} = \bar{C}^{io} + \hat{C}^{io} + \cdots$$  \hspace{1cm} (9a)

$$\tau = \bar{\tau} + \hat{\tau} + \cdots$$  \hspace{1cm} (9b)

$$a_{ik} = \bar{a}_{ik} + \hat{a}_{ik} + \cdots$$  \hspace{1cm} (9c)

$$E_k = \bar{E}_k + \hat{E}_k + \cdots$$  \hspace{1cm} (9d)

and then determine explicit expressions for the new barred and careted symbols from the definitions of $a_{ik}$ (Defs. 32, 34, 35), $E_k$ (Def. 36), and the expansion for $C^{io}$ (Appendix A). In terms of the symbols introduced in Eq. (9), the linearized form of Eq. (8) becomes

$$\bar{a}_{\text{o}t} + \sum_{k \in \mathcal{K}} \bar{a}_{\text{ok}} \gamma_k = \sum_{k \in \mathcal{K}} (\bar{E}_k + \hat{E}_k)$$ \hspace{1cm} (10a)

$$\bar{a}_{\text{o}t} + \sum_{k \in \mathcal{K}} \bar{a}_{ik} \gamma_k = g^{i}f \sum_{k \in \mathcal{K}} e_{ik} [\bar{C}^{io}(\bar{E}_k + \hat{E}_k) + \hat{C}^{io}\bar{E}_k] + \bar{\tau}_i + \hat{\tau}_i$$ \hspace{1cm} (10b)

In deriving explicit expressions for $\bar{C}^{io}$ and $\hat{C}^{io}$, we may begin by recognizing that for any $C^{rj}$, where $r$ and $j$ are labels of adjacent bodies,

$$\bar{C}^{rj} = \bar{C}^{jr} = U = 3 \times 3$$

and

$$\hat{C}^{rj} = -\gamma_j \bar{C}^{rj} \hspace{1cm} (r > j)$$

$$\hat{C}^{jr} = \gamma_j \bar{C}^{jr} = (\hat{C}^{rj})^T$$

It is clear that $\hat{C}^{io}$ must then be given by

$$\hat{C}^{io} = -\sum_{k \in \mathcal{K}} e_{ik} \bar{C}^{io} \gamma_k$$

and

$$\hat{C}^{io} = U$$  \hspace{1cm} (11a)

and

$$\hat{C}^{rj} = \sum_{s \in \mathcal{K}} (e_{sj} - e_{sr}) \bar{C}^{rj} \gamma_s$$  \hspace{1cm} (11b)
From Def. 36, $\bar{E}_k$ and $\hat{P}_h$ are given by

$$\bar{E}_k = \bar{C}^{ok} [\bar{T}^k + \sum_{j \in \partial} \bar{D}^{kj} \bar{C}^{kj} \bar{F}^j] = \bar{T}^k + \sum_{j \in \partial} \bar{D}^{kj} \bar{F}^j \quad (12a)$$

$$\hat{P}_h = \hat{C}^{0h} [\hat{T}^h + \sum_{i \in \partial} \hat{D}^{hi} \hat{C}^{hi} \hat{F}^i] + \hat{C}^{0h} [\hat{T}^h + \sum_{j \in \partial} \hat{D}^{hj} \hat{C}^{hj} \hat{F}^j]$$

$$= \hat{T}^h + \sum_{i \in \partial} \hat{D}^{hi} \hat{C}^{hi} \hat{F}^i + \sum_{j \in \partial} \hat{D}^{hj} \hat{F}^i + \hat{C}^{0h} [\hat{T}^h + \sum_{j \in \partial} \hat{D}^{hj} \hat{F}^j] \quad (12b)$$

Of course, the most important result of the complete linearization is the absence of any time-varying elements in $\bar{a}_{0o}$, $\bar{a}_{0k}$, $\bar{a}_{1o}$, and $\bar{a}_{1k}$. These constant coefficients now allow the square matrix on the left side of (5) to be inverted once and for all, in advance of numerical integration.

It would remain then to determine $\bar{T}^o$, $\bar{T}^k$, $\bar{F}^o$, $\bar{F}^k$, $\bar{T}_i$, and $\bar{F}_i$ for the specific system under study and, together with the relations in (11) and (12), to carry out the computations implied by (10). However, in constructing a subroutine to perform these computations, it was found to be much easier to directly manipulate the combined form:

$$\bar{a}_{0o} \bar{a}^0 + \sum_{k \in \partial} \bar{a}_{0k} \bar{T}_k = \sum_{k \in \partial} \bar{E}_k \quad (13a)$$

$$\bar{a}_{1o} \bar{a}^0 + \sum_{k \in \partial} \bar{a}_{1k} \bar{T}_k = \bar{E}_k \quad (13b)$$

where

$$\bar{E}_k = \bar{C}^{ok} [\bar{T}^k + \sum_{j \in \partial} \bar{D}^{kj} \bar{C}^{kj} \bar{F}^j]$$

Equations (13) obviate the computation of any body-to-body coordinate transformation matrices, $\bar{C}^{ij}$, at all.
Subroutines which provide solutions to both Eqs. (13) and (14) will be described in the next section.

**B. Subroutines MBDYTL and MBDYL**

The subroutines MBDYTL and MBDYL provide the solutions to Eqs. (13) and (14) of the preceding section, respectively. As in the case of MBDY, they each may be exercised by either of two call statements, the first of which initializes the program with the system constants.

**Initializing Call Statements**

```
CALL MBDYTL(NH,H,MB,MA,PB,PA,G,PI)
```

or

```
CALL MBDYL(NH,H,MB,MA,PB,PA,G,PI)
```

Except for the call statement name, the statements are exactly the same as that for MBDY. All the arguments are defined exactly as given in Section IIE.

To compute the angular accelerations \( \dot{\omega}, \dot{\gamma}, \cdots, \dot{\gamma} \) the subroutine MBDYTL (or MBDYL) must be entered at every integration interval. This is accomplished by executing the dynamic call statement.

**Dynamic Call Statement**

```
CALL MRATE(NH,TH,TB,TA,FB,FA,GM,GMD,GMDD,WO,WDOT)
```

Again, the dynamic call statement arguments are the same as those defined for MBDY in Section IIE, with the following exceptions:

- \( TH(i) = \text{array of hinge torques, } \dot{\gamma}_i; i \in \mathcal{D} \)
- \( TB(j) = \text{array of elements of } \dot{T}_o^i; j = 1, 2, 3 \)
- \( TA(i,j) = \text{array of elements of } \dot{T}_i; i \in \mathcal{D}, j = 1, 2, 3 \)
- \( FB(j) = \text{array of elements of } \dot{F}_o; j = 1, 2, 3 \)
- \( FA(i,j) = \text{array of elements of } \dot{F}_i; i \in \mathcal{D}, j = 1, 2, 3 \)

Even though the arguments GMD and WO are not used by either MBDYTL or MBDYL, and GM is not used by MBDYL, these “dummy” arguments are retained in the call statements for the sake of uniformity and to allow easy interchange of subroutines if desired.

The Fortran “type” and storage allocation statements which must appear in the main program that calls MBDYTL (or MBDYL) are identical to those required for MBDY.

**Required Specification Statements**

```
INTEGER NH, H(n), PI(n)
REAL MB(7), MA(n,7), PB(n,3), PA(n,n,3), G(n,3), TH(n),
     TB(3), TA(n,3), FB(3), FA(n,3), GM(n), GMD(n),
     GMDD(n), WO(3)
DOUBLE PRECISION WDOT(n + 3)
```
Also, the statement

\[
\text{PARAMETER } Q = n
\]

must appear \textit{within} the subroutine MBDYTL (or MBDYLY) to allocate the proper amount of internal storage. Complete listings of the MBDYTL and MBDYL programs are given in Appendix B.

\textbf{C. Application of MBDYTL and MBDYL to the Five-Body System}

\textbf{1. Small platform maneuver simulation.} We may now proceed to apply the linearized subroutines to the same vehicle dynamics/control configuration examined in Section IIIF. The vehicle is shown in Fig. 4 and the control system in Fig. 5.

Expressions previously given for \( T^i, \tau_1, \tau_2, \tau_3, \) and \( \tau_4 \) may now be re-interpreted as describing \( \hat{T}^i, \hat{\tau}_1, \hat{\tau}_2, \hat{\tau}_3, \) and \( \hat{\tau}_4, \) respectively, since they contain no second-order or higher terms in the variables \( \omega_k, \gamma_k \) and their derivatives. In fact, in order to again simulate Case 2, i.e., the small platform maneuver from \( \gamma_1 = 218 \text{ deg to } \gamma_1 = 220 \text{ deg (at } t = 0) \) and \( \gamma_2 = -30 \text{ deg to } \gamma_2 = -25 \text{ deg (at } t = 5 \text{ s}), \) it is necessary to replace the statement "CALL MBDY(NH, ...)" with the statement "CALL MBDYTL(NH, ... )" and to modify the arrays MA(2,1), PA(1,2,j), PA(2,2,j), and G(2,j).

The reason for the required modifications to these arrays stems from Defs. 4 and 12, which require all local, body-fixed coordinate frames to coincide (i.e., \( b_\alpha = b_\beta \) for \( \alpha = 1, 2, 3 \) and \( \beta = \{b\} \)) when the system is in its nominal state, i.e., when \( \gamma_k = 0 \) for all \( k \in \mathcal{K} \). When the routines MBDYTL and MBDYL are used, the variables \( \gamma_k \) are, by definition, required to remain very close to the nominal state \( \gamma_k = 0 \). Otherwise, the linearization process is meaningless. It is clear, then, that to simulate Case 2 with the linearized programs, the new linearized variables \( \hat{\gamma}_k (k = 1, 4) \) must be redefined in terms of the old \( \gamma_k \) such that, for example,

\[
\begin{align*}
\hat{\gamma}_1 &= \gamma_1 - 219 \text{ deg} \\
\hat{\gamma}_2 &= \gamma_2 + 27.5 \text{ deg} \\
\hat{\gamma}_3 &= \gamma_3 \\
\hat{\gamma}_4 &= \gamma_4
\end{align*}
\]

And now it is also clear that bodies \( d_1 \) and \( d_2 \) have different orientations in their own body-fixed frames, implying different inertia matrices \( I^1, I^2 \) and different components for the vectors \( p^{1*}, p^{2*}, p^{1*}, \) and \( g^1, g^2 \) in their respective frames, \( \{b^1\} \) and \( \{b^2\} \).

In this particular vehicle configuration, the platform hub, \( d_3 \), has symmetrical mass properties about its axis of rotation, which in turn is aligned to one of the body-fixed coordinate axes. Thus, \( I^1 \), and therefore MA(1,j), \( j = 1 - 7 \) remains unchanged, as does \( p^{1*} \) (PA(1,1,j), \( j = 1 - 3 \)) and \( g^1 \) (G(1,j), \( j = 1 - 3 \)). However, it is apparent that \( p^{1*}, p^{2*} \), as well as \( g^2 \), and \( I^2 \) must change, since they were previously defined for \( \gamma_1 = 0, \gamma_2 = 0 \) and now must reflect the configuration defined by \( \hat{\gamma}_1 = 0, \hat{\gamma}_2 = 0, \hat{\gamma}_3 = 0, \hat{\gamma}_4 = 219 \text{ deg, } \gamma_2 = -27.5 \text{ deg} \).
The components of a vector described in the old frame fixed in $\mathcal{I}_2$ can be related to its components in the new frame by the sequence of rotations $\gamma_1$ and $\gamma_2$, as follows:

$$
\begin{bmatrix}
\mathbf{v}'^0_x \\
\mathbf{v}'^0_y \\
\mathbf{v}'^0_z
\end{bmatrix} = \begin{bmatrix}
\mathbf{c}_{\gamma_1} & -\mathbf{s}_{\gamma_1} \mathbf{c}_{\gamma_2} & -\mathbf{s}_{\gamma_1} \mathbf{s}_{\gamma_2} \\
\mathbf{s}_{\gamma_1} \mathbf{c}_{\gamma_2} & \mathbf{c}_{\gamma_1} \mathbf{c}_{\gamma_2} & \mathbf{c}_{\gamma_1} \mathbf{s}_{\gamma_2} \\
0 & -\mathbf{s}_{\gamma_2} & \mathbf{c}_{\gamma_2}
\end{bmatrix}
\begin{bmatrix}
\mathbf{v}^0_x \\
\mathbf{v}^0_y \\
\mathbf{v}^0_z
\end{bmatrix} = C^{\mathcal{I}_2}_{\mathcal{I}_0}
\begin{bmatrix}
\mathbf{v}^0_x \\
\mathbf{v}^0_y \\
\mathbf{v}^0_z
\end{bmatrix}
$$

($$\mathbf{s}_{\gamma} = \sin \gamma, \mathbf{c}_{\gamma} = \cos \gamma$$, etc.)

Therefore, the components of $g^2$ in a simulation linearized about $\gamma_1 = 219$, $\gamma_2 = -27.5$ deg become:

$$
\begin{bmatrix}
\mathbf{g}^2_x \\
\mathbf{g}^2_y \\
\mathbf{g}^2_z
\end{bmatrix} = C^{\mathcal{I}_2}_{\mathcal{I}_0}
\begin{bmatrix}
-0.77715 \\
0.62932 \\
0
\end{bmatrix}
$$

Similarly, the components of $p^{21}$ become

$$
\begin{bmatrix}
\mathbf{p}^{21}_x \\
\mathbf{p}^{21}_y \\
\mathbf{p}^{21}_z
\end{bmatrix} = C^{\mathcal{I}_2}_{\mathcal{I}_0}
\begin{bmatrix}
-0.22 \\
0.20 \\
0
\end{bmatrix}
$$

The vector $p^{12}$ only changes due to the $\gamma_1$ rotation, i.e.,

$$
\begin{bmatrix}
\mathbf{p}^{12}_x \\
\mathbf{p}^{12}_y \\
\mathbf{p}^{12}_z
\end{bmatrix} = C^{\mathcal{I}_2}_{\mathcal{I}_0}
\begin{bmatrix}
\mathbf{c}_{\gamma_1} & -\mathbf{s}_{\gamma_1} & 0 \\
\mathbf{s}_{\gamma_1} & \mathbf{c}_{\gamma_1} & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
0 \\
-0.10 \\
-0.75
\end{bmatrix} = \begin{bmatrix}
\mathbf{0.1} s_{\gamma_1} \\
\mathbf{-0.1} c_{\gamma_1} \\
\mathbf{-0.75}
\end{bmatrix}
$$

Finally, the modified value of $I^2$ is given by

$$
I^2 = C^{\mathcal{I}_2}_{\mathcal{I}_0}
\begin{bmatrix}
4.85 & 0.41 & -0.07 \\
0.41 & 2.20 & 0.54 \\
-0.07 & 0.54 & 5.50
\end{bmatrix}
(C^{\mathcal{I}_2}_{\mathcal{I}_0})^T = \begin{bmatrix}
3.52 & 1.25 & -0.75 \\
1.25 & 3.79 & 0.73 \\
-0.75 & 0.73 & 5.24
\end{bmatrix}
$$

Having determined these changes in the system configuration description, the appropriate values of $M_{A(i,j)}$, $i = 1 - 6$, $P_{A(1,2,j)}$, $P_{A(2,2,j)}$, $G(2,j)$ may be entered to produce the simulation program of Fig. 15. Notice in Fig. 15 that it was also necessary to change $IC1$ and $IC2$ to reflect proper initial values of the new variables $\gamma_1$ and $\gamma_2$ (i.e., $\gamma_1(0) = -1$ deg, $\gamma_2(0) = -2.5$ deg; these variables correspond to $\gamma_1(0) = 218$ deg, $\gamma_2(0) = -30$ deg). Using this program and the MBDYTL subroutine, Case 2 was simulated again, and the results are shown in Fig. 16. They may be directly compared to the dynamic responses of Fig. 14. While there are some very slight differences visible in the angular rate responses, the overall result is virtually identical with that obtained using MBDY, and is obtained in one third the time used by MBDY (MBDY used 140 s, MBDYTL 48 s on the Univac 1108). Further, the substitution of MBDYL for MBDYTL gives exactly the same results as those shown in Fig. 16, since no external forces or torques were applied to the system, and calculates them in only 42 s.
Obviously, the gain in computational speed when the linearized routines are used is quite significant, even for this relatively simple example. In Ref. 7, a 17-degree-of-freedom system is simulated using MBDYTL exclusively, since it gave a speed advantage over MBDY of about 25:1 with more than adequate solution accuracy.

PROGRAM S-BODY SYSTEM SIMULATION USING MBDYTL --- CASE 2
INTEGER NHFI(I1,II,IR)
DATA NH/II,17/PI/100,10/ DATA MAI II) .351 .10 MA3, 1,0)
DATA MAI 2) .41 .2 MA4, 1,0) DATA MAI 3) /5.2 MA5, 1,0)
DATA MAI 4) /1.9 MA6, 1,0) DATA MAI 5) /1.9 MA7, 1,0)
DATA MAI 6) /1.9 MA8, 1,0) DATA MAI 7) /1.9 MA9, 1,0)
DATA MAI 8) /1.9 MA10, 1,0) DATA MAI 9) /1.9 MA11, 1,0)
DATA MAI 10) /1.9 MA12, 1,0) DATA MAI 11) /1.9 MA13, 1,0)
CALL MBDYTL(NH,HI,HI,MA,PA,PZ,PI,PHI,ZT)
END

Fig. 15. Case 2 simulation using MBDYTL
Fig. 15 (contd)
Fig. 16. Results of Case 2 simulation using MBDYTL
Fig. 16 (contd)
Fig. 16 (contd)
Fig. 16 (contd)
Fig. 16 (contd)
IV. Partially Linearized Systems

A. The Equations

One further distinction that can certainly be made in characterizing systems of hinge-connected rigid bodies (somewhere between those systems where all hinges are permitted unrestricted rotation and those systems whose hinge and reference body rotations are all “small”) is the partially linearized system. In this report, the term “partially linearized” will refer to the case in which the reference body rotations are unrestricted and some, but not all, of the $\gamma_k (k \in \mathcal{C})$ and their derivatives are considered to be “small.”

The indices of those angles for which linearization is in some sense justified may be collected in the set $\mathcal{R}$, and, using the expansions of Eq. (9), the linearized form of Eq. (8) becomes

\begin{equation}
(a_{oo} + a_{oo}) \dot{\omega} + \sum_{k \in \mathcal{R}} \tilde{a}_{ok} \dot{\gamma}_k + \sum_{k \in \mathcal{C} - \mathcal{R}} (\tilde{a}_{ok} + \tilde{a}_{ok}) \dot{\gamma}_k = \sum_{k \in \mathcal{C}} (\tilde{E}_k + \tilde{E}_k) \tag{15a}
\end{equation}

\begin{equation}
(a_{io} + a_{io}) \dot{\omega} + \sum_{k \in \mathcal{R}} \tilde{a}_{ik} \dot{\gamma}_k + \sum_{k \in \mathcal{C} - \mathcal{R}} (\tilde{a}_{ik} + \tilde{a}_{ik}) \dot{\gamma}_k = g^{ij} \sum_{k \in \mathcal{C}} \epsilon_{ik} [\tilde{C}^{ij} (\tilde{E}_k + \tilde{E}_k) + \tilde{C}_{ik} \tilde{E}_k] + \tilde{\tau}_i + \tilde{\tau}_j \tag{15b}
\end{equation}

It should be noted that, for a wide class of systems, it can be recognized in advance that $\epsilon_{ik} = 0$ for $i \in \mathcal{R}$ and $k \in \mathcal{C} - \mathcal{R}$. This is the case whenever none of the hinge points whose indices comprise $\mathcal{R}$ lies on a path from $\mathcal{C}$, to any hinge point whose index is not in $\mathcal{R}$. In physical terms, this means that the small-angle rotations are confined to terminal appendages, each of which is attached to only one body whose index lies outside of $\mathcal{R}$. In fact, primary attention is focused on those scalar elements $a_{ik}$ in (15b) where $i, k \in \mathcal{R}$ and $i, k$ are members of the same terminal appendage set. It can be shown that such elements $a_{ik}$ are all constants.

**Def. 38.** For $v \in \mathcal{R}$, let $\mathcal{R}_v$ denote a “terminal appendage set.” Thus $\mathcal{R}_v$ is a subset of $\mathcal{R}_v$, containing the body indices of one or more terminal appendages which share a common base body, $v$.

By appropriate exchanges of rows and columns of the symmetric system matrix, the constant $a_{ik}$’s may be isolated in an $r \times r$ symmetric matrix, $A_v$, forming the lower right-hand corner of the original matrix. One can then employ a matrix partitioning scheme, where the constant matrix $A_v$ is inverted just once, and a time-varying matrix equation, now of order $(n + 3 - r)$, is solved at every integration step. Clearly, the size of $A_v$ should be maximized for the greatest computational advantage, and, since only one terminal appendage set (there may be several) may be used to form $A_v$, the largest terminal appendage set should be used.

With this approach, i.e., a single partitioning of the system matrix into constant and time-varying parts, another version of the MBDY subroutine, called MBDYPL, was written to solve the partially linearized case. However, in a move to further simplify and speed up the computational task, Eq. (15) was modified as follows:

\begin{equation}
(\tilde{a}_{oo} + \sum_{k \in \mathcal{C}} \tilde{a}_{ok}) \dot{\omega} = \sum_{k \in \mathcal{C}} \tilde{E}_k \tag{16a}
\end{equation}
\[
\hat{a}_{1i0} + \sum_{k \in \mathcal{D} - A} \hat{a}_{1k} \hat{y}_k + \sum_{k \in A} \hat{a}_{ik} \hat{y}_k = g^{17} \hat{C}_{10} \sum_{k \in \mathcal{D}} \epsilon_{ik} \hat{E}_k + \hat{r}_i \ 
\]

where
\[
\hat{a}_{10} = \bar{a}_{10} + \hat{a}_{10} + \cdots \\
\hat{a}_{ik} = \bar{a}_{ik} + \hat{a}_{ik} + \cdots \\
\hat{a}_{1i} = \bar{a}_{1i} + \hat{a}_{1i} + \cdots \\
\hat{a}_{ik} = \bar{a}_{ik} + \hat{a}_{ik} + \cdots 
\]

The intent of Eq. (16) is to avoid having to form the separate quantities \(\bar{a}_{10}, \hat{a}_{10}, \bar{a}_{ik}, \hat{a}_{ik}, \) etc., and instead, to compute only the combined forms \(\hat{a}_{10}, \hat{a}_{ik}, \) etc., using the combined form of the coordinate transformation, \(\hat{C}_{10}\). This approach saves overall computation time, although it allows the retention of some second-order terms in the linearized variables that would have disappeared otherwise. Thus, Eq. (16) is the matrix equation actually solved by MBDYPL, a somewhat more accurate representation than that of Eq. (15).

B. Subroutine MBDYPL Usage

Just as in the other subroutines, MBDYPL is entered by means of either of two call statements, one of which initializes the program with the system constants.

Initializing Call Statement

\[
\text{CALL MBDYPL(NH,H,MB,MA,PB,PA,G,PI,LI,TI)}
\]

The call statement arguments are exactly as defined previously, with exception of the two additions

\(\text{LI(i) = array of linearized variable indicators; } \text{i} \in \mathcal{D} \text{ (If } \gamma_i \text{ and its derivatives are assumed small and therefore are to be linearized, LI(i) = 1. Otherwise, LI(i) = 0.)}\)

\(\text{TI(i) = array of terminal appendage body indicators; } \text{i} \in \mathcal{D} \text{ (If TI(i) = 1, } \gamma_i \text{ is a member of the largest terminal appendage in the system. If TI(i) = 0, it is not, although } \gamma_i \text{ could still be a linearized variable.)}\)

It should be made clear that LI(i) is an array to be supplied \textit{by the user}; i.e., it is an \textit{input} array. On the other hand, TI(i) is calculated \textit{by the subroutine} for the user's information, and is an \textit{output} quantity. As mentioned previously, it is desirable to identify the \textit{largest} terminal appendage of the system and its member bodies. An algorithm to do just that is located in the initializing portion of the subroutine, and the results of its calculations are later used for optimum rearrangement and partitioning of the system matrix, as well as made available to the user in TI.

As before, MBDYPL is executed at every numerical integration step using the dynamic call statement.

Dynamic Call Statement

\[
\text{CALL MRATE(NH,TH,TB,TA,FB,FA,GM,GMMD,WO,WDOT)}
\]
All arguments of the dynamic call statement are exactly as defined for MBDY, with the understanding that \( \theta(i) \), \( T(i) \), and \( F(i) \) refer to \( \dot{\theta}_{i} \), \( \dot{T}_i \), and \( \dot{F}_i \), respectively, when \( i \in A \). In general, any torque or force expressions should be free of second-order terms in the linearized variables (for which \( Li(i) = 1 \)) or their derivatives.

The necessary specification statements for MBDYPL are the same as for MBDY, with the addition of the integers \( Li \) and \( Ti \).

**Required Specification Statements**

\[
\text{INTEGER NH, H(n), P1(n), Li(n), Ti(n)}
\]

\[
\text{REAL MB(7), MA(n,7), PB(n,3), PA(n,n,3), G(n,3), TH(n),}
\]

\[
\text{TB(3), TA(n,3), FB(3), FA(n,3), GM(n), GMD(n),}
\]

\[
\text{CMDD(n), WQ(3)}
\]

\[
\text{DOUBLE PRECISION WDOT(n + 3)}
\]

Again, the statement,

\[
\text{PARAMETER Q = n}
\]

must appear within the subroutine MBDYPL to properly allocate internal storage. A complete listing of MBDYPL is provided in Appendix C.

**C. Application of MBDYPL to the Five-Body System**

1. **Large platform maneuver.** Since MBDYPL is designed to accommodate both large and small rotations in a hinge-connected system, Case 1 would be the obvious dynamic maneuver with which to test MBDYPL.

Recall that the maneuver consists of rotating the platform from its initial position \((\gamma_1(0) = 218 \text{ deg}, \gamma_2 = -30 \text{ deg})\) to the position \(\gamma_1 = 168 \text{ deg}, \gamma_2 = 20 \text{ deg}\) by simultaneous slewing about both axes at the commanded rate of 1 deg/s (starting at \( t = 0 \)).

The main simulation program listed in Fig. 6 can be used virtually “as is” by MBDYPL, with only the addition of \( Li(4) \) and \( Ti(4) \) to the INTEGER statement and \( Li \) input elements to the DATA statement. In this case, linearization will be requested for \( \gamma_1 \) and \( \gamma_2 \), so that \( Li(1) = 0 \), \( Li(2) = 0 \), \( Li(3) = 1 \), and \( Li(4) = 1 \). Examination of \( Ti \) after executing “CALL MBDYPL (NH,...)" would show it to be identical to \( Li \); i.e., bodies \( \phi \) and \( \phi \), would be identified as members of the largest system terminal appendage. The modified simulation program using MBDYPL is listed in Fig. 17.

Results of the platform maneuver, Case 1, using the partially linearized approach are shown in Fig. 18. Comparison of these to the results produced by MBDY in Fig. 7 shows the differences to be indistinguishable. Figure 19 expands the plots of \( \omega^0 \) components for just the first 10 s for closer comparison (with Fig. 8).

While the accuracy of the MBDYPL run in this case was apparently excellent, the speed advantage over MBDY was only very slight, i.e., 430 versus 440 s of computation time for this four-hinge system. Since MBDYPL does perform a number of manipulations associated with partitioning and rearranging the system matrix as well as matrix multiplication to solve the partitioned system, some tradeoff between
the time used to accomplish these tasks and the time saved by recognizing certain constants and advanced matrix inversion should be expected. Just exactly where the "break-even" point between MBDY and MBDYPL speed is depends on both the number of hinges and the number of linearized hinges. However, in the case simulated here, we seem to be quite near the break-even point. In contrast, the system simulation in Ref. 7 with 14 hinges (10 linearized) was speeded up by a factor of two using MBDYPL.

It should also be noted that the subroutine MBDYPL does accommodate prescribed motions at any of its hinges, as do the other subroutines, whether the hinge is linearized or not.

V. Summary and Conclusions

This report confines itself to the consideration of mathematical models useful for the description of hinge-connected rigid body system dynamics and, further, the computer simulation of those idealizations using general-purpose subroutines. As the sequel to Ref. 9, it documents in explicit detail the application of each of these subroutines under the circumstances for which they were especially designed, i.e., MBDY for unrestricted systems, MBDYTL and MBDYL for totally linearized systems, and MBDYPL for partially linearized systems. In each case, the subroutines, which are written in Fortran V, serve as the powerful tool that is called upon by the main simulation program (written here in CSSL III) to calculate vehicle attitude dynamics.

In the process of drawing conclusions about the relative utility of these programs, one must certainly begin with the obvious conclusion that MBDY, since it solves the exact and unrestricted equations of motion for the system, serves as the ultimate standard of accuracy for the rest. However, MBDY need not be restricted in its use to just the role of a "check routine." For small systems, it will run faster than MBDYPL, and one can certainly envision system rotations large enough to make the linearized routines completely unsuitable for the simulation accuracies required. Further, the judicious use of the option available in all the routines for prescribed hinge rotations can speed the running time of a simulation using MBDY remarkably. Thus, MBDY should be used often and to good advantage.

Of course, the utility of the completely linearized dynamic simulation subroutines, MBDYTL and MBDYL, was well exposed in Section III. For large or small systems of bodies in which all the hinge and reference body rotations are deemed "small," either due to fortunate circumstance or deliberately controlled constraint, the use of MBDYTL or MBDYL can offer tremendous speed advantages over MBDY and still maintain acceptable accuracies of the dynamic response. This totally linearized approach has proved to be of the greatest value in analyzing control problems at JPL, since most systems fall, or can be forced into, this category. In fact, the accuracy of MBDYL has been more than adequate for a large variety of space vehicle control problems studied at JPL. Of course, if the vehicle control dynamics can also be linearized, the inverted attitude dynamics matrix used by MBDYTL or MBDYL can be used directly in control stability analyses, i.e., to compute overall system root loci and/or eigenvalues.

Finally, the utility of MBDYPL is perhaps the least of the four routines, although it still offers some advantages, particularly for large systems. Where there are a
few, very large hinge rotations to be simulated but the large majority of system hinges exhibit only small excursions, MBDYPL may offer the only chance to achieve the desired accuracies in a reasonable simulation time. In similar circumstances, it may serve as a more economical reference or check solution source than MBDY. All in all, the partitioning scheme employed in MBDYPL, while not offering the tremendous speed advantages of MBDYL, at least provides some gain in efficiency in certain specialized instances.

As for the general-purpose character of all four subroutines, it is felt that their presentation in Fortran will make them widely useful, both to those who are comfortable with special-purpose simulation languages and to those who prefer to write their own simulation programs and integration subroutines. In order to make the routines more easily accessible, the entire package has been submitted to COSMIC (Computer Software Management and Information Center), University of Georgia, Athens, GA 30601, for evaluation and dissemination to interested governmental agencies as well as industrial, educational, and research institutions. While many may wish to further specialize one or more of these programs to their own particular needs, the four programs, as they are described here, can serve as a reliable and valuable addition to any simulation subroutine library.
PROGRAM 5-BODY SYSTEM SIMULATION USING MBDYPL --- CASE 1

INTEGER NH, HI, NA, MA(1:47), PB(1:3), PA(1:4, 1:4), G(1:4, 1:4), GM(1:4), GM0(1:4), W(1:4)

DOUBLE PRECISION WOOT(1)

DATA W/0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85/ W(1:4)/D0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85/

DATA MA(1, 1), MA(1, 2), MA(1, 3), MA(1, 4), MA(2, 1), MA(2, 2), MA(2, 3), MA(2, 4)/

DATA MA(3, 1), MA(3, 2), MA(3, 3), MA(3, 4), MA(4, 1), MA(4, 2), MA(4, 3), MA(4, 4)/

DATA FA(1, 1), FA(1, 2), FA(1, 3), FA(1, 4), FA(2, 1), FA(2, 2), FA(2, 3), FA(2, 4)/

DATA FA(3, 1), FA(3, 2), FA(3, 3), FA(3, 4), FA(4, 1), FA(4, 2), FA(4, 3), FA(4, 4)/

DATA PA(1, 1), PA(1, 2), PA(1, 3), PA(1, 4), PA(2, 1), PA(2, 2), PA(2, 3), PA(2, 4)/

CONSTANT TM = 0.20, D0 = 0.05, TR = 2.0

CONSTANT TFINAL = 600, CLKTIM = 1100

CONSTANT T1 = 0.239, T2 = 0.21, T3 = 0.31

CONSTANT SLRAT = 0.17, 0.32

CONSTANT PHI = 0, THETA = 0, PSI = 0

CONSTANT W1 = 0, W2 = 1, W3 = W4 = 0

CONSTANT K1 = 3500, K2 = 3500, K3 = 2000, K4 = 2000

CONSTANT B1 = 20, B2 = 20, B3 = 10, B4 = 10

CONSTANT IC1 = 210, IC2 = 30

INITIAL

NH = 1

CALL MBDYPL NH, HI, PB, PA, G, GM0, W, TM, TFINAL, CLKTIM, B1, B2, B3, B4

DYNAMIC

IF TIME < TFINAL GO TO 51

DERIVATIVE BODY5

VARIABLE TIME = 0

CINTERVAL CT = 0.06

XPROR W1 = E, W2 = E, W3 = E, W4 = E, G3 = E, G4 = E

STCKT CLKTIM

OUTPUT 50, W1, W2, W3, W4, THETA1 = ATAN2(NY0, NZ0), THETA2 = ATAN2(-NX0, NZ0), THETA3 = ATAN2(-LY0, LX0)

PREPAR W1 = W2 = W3 = W4 = GM0(1) = GM0(2) = GM0(3) = GM0(4) = GM0

ENDSOFT

W(1:4) = 0

PL1 = RD = GM1 = GM2 = GM3 = GM4

COMMENT

COMMENT CELESTIAL-SENSOR-MEASURED ANGLES

COMMENT

D1 = 000001

NO

NX0 = NY0 = NZ0 = 0

LY0 = LX0 = 0

THETA1 = ATAN2(NY0, NZ0)

THETA2 = ATAN2(-NX0, NZ0)

THETA3 = ATAN2(-LY0, LX0)

Fig. 17. Case 1 simulation program using MBDYPL
COMMENT RATE FEEDBACK SIGNAL AND SWITCHING AMP. DYNAMICS
COMMENT
IN1=-\theta_{1} - TR * W1 $ IN2=-\theta_{2} - TR * W2 $ IN3=-\theta_{3} - TR * W3
AMP1=SWAMP\left(\text{TIME}+\text{TR}+\text{IN1}\right)
AMP2=SWAMP\left(\text{TIME}+\text{TR}+\text{IN2}\right)
AMP3=SWAMP\left(\text{TIME}+\text{TR}+\text{IN3}\right)
COMMENT PLATFORM POSITION COMMANDS
COMMENT
GM1C=SLRAT\left(-1.4*\text{STEP}(50,\text{TIME})\right) $ GM1C=\text{INTEG}(\text{GM1C}+\text{GM1C})$
GM2C=SLRAT\left(1.0*\text{STEP}(50,\text{TIME})\right) $ GM2C=\text{INTEG}(\text{GM2C}+\text{GM2C})$
COMMENT REFERENCE BODY EXTERNAL TORSUS
COMMENT HINGE TORQUES
COMMENT
TB(1)=L1*AMP1 $ TB(2)=L2*AMP2 $ TB(3)=L3*AMP3
TH(1)=-k1*(GM1-GM1C)-b1*GM1D
TH(2)=-k2*(GM2-GM2C)-b2*GM2D
TH(3)=-k3*(GM3-GM3C)-b3*GM3D
TH(4)=-k4*(GM4-GM4C)-b4*GM4D
COMMENT SYSTEM ANGULAR ACCELERATIONS
COMMENT
CALL RATE\left(\text{IN}\theta, \text{TH}, \text{TB}, \text{TA}, \text{FB}, \text{FA} , \text{GM}, \text{MD}, \text{W0}, \text{WDOT}\right)
W1=\text{INTEG}(\text{WDOT}(1)+\text{W1})$ W2=\text{INTEG}(\text{WDOT}(2)+\text{W2})$ W3=\text{INTEG}(\text{WDOT}(3)+\text{W3})$
GM1D=\text{INTEG}(\text{WDOT}(4)+\text{W4})$ GM2D=\text{INTEG}(\text{WDOT}(5)+\text{W5})$
GM3D=\text{INTEG}(\text{WDOT}(6)+\text{W6})$ GM4D=\text{INTEG}(\text{WDOT}(7)+\text{W7})$
GM1=\text{INTEG}(\text{GM1D}+\text{GM1})$ GM2=\text{INTEG}(\text{GM2D}+\text{GM2})$
GM3=\text{INTEG}(\text{GM3D}+\text{GM3})$ GM4=\text{INTEG}(\text{GM4D}+\text{GM4})$
COMMENT HCK PARAMETER RATES AND POSITIONS
COMMENT
PZ0+PA0+PB0+PC0=HCK\left(\text{HCK},\text{PZ0},\text{PA0},\text{PB0},\text{PC0},\text{W1},\text{W2},\text{W3}\right)$
PZ0=\text{INTEG}(\text{PZ0}+\text{PZI})$ PA0=\text{INTEG}(\text{PA0}+\text{PAI})$ PB0=\text{INTEG}(\text{PB0}+\text{PBI})$
PC0=\text{INTEG}(\text{PC0}+\text{PCI})$
END
END
TERMINAL
S1= CONTINUE
END
END

Fig. 17 (contd)
Fig. 18. Results of Case 1 simulation using MBDYPL
Fig. 18 (contd)
Fig. 18 (contd)
Fig. 18 (condt)
Fig. 18 (contd)
Fig. 18 (contd)
Fig. 18 (contd)
Fig. 18 (contd)
Fig. 18 (contd)
Fig. 19. Results of Case 1 simulation using MBDYPL (components of $\omega_i^0$ first 10 s)
Fig. 19 (contd)
Fig. 19 (contd)
Appendix A
Subroutine MBDY Description

I. Algorithm for Calculating Direction Cosine Matrices, \( C^{or} (r \in \mathcal{O}) \)

As was pointed out in Section II.C, all the coordinate transformation, or direction cosine, matrices, \( C^{ij} (r, j \in \mathcal{O}) \) need not be calculated if all the system equations are expressed in the reference body \( \mathcal{O}_e \) coordinate frame. Thus, the objective is simply to obtain \( C^{or} \) for all \( r \in \mathcal{O} \). An algorithm for accomplishing this is incorporated in MBDY and in MBDYTL and MBDYPL as well (MBDYL uses no direction cosine transformations). The computation procedure is as follows:

Step 1. Let \( r = 1 \)

Step 2. Calculate elements of the direction cosine matrix \( C^{ir} \), where \( j = h_r \) (see Eq. 6)

Step 3. If \( j = 0 \), jump to Step 5

Step 4. Calculate elements of \( C^{or} \) from relation \( C^{or} = C^{oi}C^{ir} \)

Step 5. Let \( r = r + 1 \)

Step 6. If \( r < n + 1 \), go back to Step 1. Otherwise, stop.

The calculations involved in Step 2, of course, may include the use of \( \sin \gamma \) and \( \cos \gamma \), or they may use approximations thereof (i.e., \( \gamma_r = \sin \gamma_r, \cos \gamma_r = 1 \)), depending upon whether linearization is being employed or not. This sort of bootstrapping calculation of \( C^{or} \), by way of Step 4, is made possible by the fact that \( j \) is always less than \( r \), and therefore \( C^{oi} \) and \( C^{ir} \) must necessarily have been computed before coming to \( C^{or} \).

II. Summary of MBDY User Requirements

Subroutine Entry Statements

```
CALL MBDY(NH,H,MB,MA,PA,G,PI)
CALL MRATE(NH,TH,TB,TA,FA,GM,GMD,GMDD,WO,WDOT)
```

Input/Output Variable Type and Storage Specifications—Fortran V

```
INTEGER NH, H(n), PI(n)
REAL MB(7), MA(n,7), PB(n,3), PA(n,n,3), G(n,3), TH(n),
       TB(3), TA(n,3), FB(3), FA(n,3), GM(n), GMD(n),
       GMDD(n), WO(3)
DOUBLF PRECISION WDOT(n + 3)(where \( n = \) number of system hinges)
```

External Subroutines Called

SOD—double precision matrix equation solving subroutine; user-supplied (see line 437 in MBDY listing)
This subroutine solves the matrix equation $Ax = EC$, where $A$ is of order $IV \times IV$ and $IV \geq 3$. The solution is returned in $EC$.

**Subroutine Setup**

Insert the Fortran V statement (see listing)

```
PARAMETER  Q = n
```

where $n =$ number of system hinges.

**Data Restrictions**

$n \geq 1$

**Core Storage Required**

- Code: 2000 words
- Data: 500 words (minimum; increases with $n$)

**Listing**
SUBROUTINE MBDY(KH,KB,MA,PB,PAPA,QPI)

INTEGER HIPI(11)
REAL MHI(11),KA(11*7),PB(INH*31),PA(IH*31),KH*31,GK*31)
PARAMETER G=4
PARAMETER SA*1,SV=S*3,SV=S*3,SV=S,SV=S*3

ADJUSTABLE DIMENSIONS

INTEGER HIPI(11)
REAL ME(11*MAINHI,PBINH*3TPAINk33GIN3)

PARAMETER Q=4
PARAMETER S=Q*1 V:Q

ADDITIONAL DIMENSIONED VARIABLES

DOUBLE PRECISION ALV,VRK(4),BMASSI(11)
DIMENSION PHIS(31*31),IX(31*31),IZ(31*31),DX(31*31),DY(31*31),DZ(31*31)

EQUIVALENCE ALV,VRK(4),BMASSI(11)

IF ITIALIZE EPSK +1 WITH ZEROS AND DEFINE KK(IK)

DC 86 K=1
KK(IK) = K+1
DC 85 J=1
EPSK+J=0

DEFINE EPSK +J) USING HIJ)

DO 86 K=1,NH
DO 85 J=1,NB
IF (K.EQ.J+1) EPSK+J+1=1
IF (K.LT.J-1) GO TO 87
GO TO 86

CONTINUE

J=K+1
J=J-1
DO 89 L=J+1,J
EPSK +L=EPSK +L-1+1
CONTINUE

I)MBI(L)=MBI(L)
IXZ(L)=MBI(L)
IXZ(L)=MBI(L)
IXZ(L)=MBI(L)
IXZ(L)=MBI(L)

IF=BMASSI(1)
DO 85 J=2,NB
MAJ+1=MAJ+1+1
MAJ+1=MAJ+1+1
MAJ+1=MAJ+1+1
MAJ+1=MAJ+1+1

TH=TH*BMASSI(1)
DC 149 I=1

IT" 32-1992
IF(I.EC.J) GO 10

IF(I.EC.J) GO TO 70

IF(EPSII#J).EQ.1) GO TO 400

LYJ1=PA(N1+II*+1)

IF(I.EC.J) GO 10

LX1=PA(I1+II*+1)

CONTINUE

DO 13 N=1,NB

DO 30 J=1,NB

RETURN

DIMENSION T(31,7,18),TA(3),FB(3),FA(3),GM(3),GM(18)

DOUBLE PRECISION DELI

C CALCULATION OF AUGMENTED INERTIA D Y A D I C S FOR EACH BODY

C BODY-TO-BODY COORDINATE TRANSFORMATION MATRICES

DO 30 J=1,NB
SGM = SIN(GM(J+1))

CGM = CGM(J+1)

Q1 = CGM1 * GI(J+1)

Q2 = CGM1 * GI(J+2)

Q3 = CGM1 * GI(J+3)

SG1 = SG1 * GI(J+1)

SG2 = SG1 * GI(J+2)

SG3 = SG1 * GI(J+3)

AE1 = AE1(J+1) = CGM * G15

AE2 = AE2(J+1) = SG1 * G12

AE3 = AE3(J+1) = SG1 * G13

AE4 = AE4(J+1) = SG1 * G14

IF(IG.EQ.0) GO TO 3350

DO 334 L = 1,3

DC 334 M = 1,3

IF(J(J+1,L) .EQ. 0) GO TO 3350

DO 335 K = 1,3

DC 335 L = 1,3

DC 335 M = 1,3

CONTINUE

C COORD. TRANSFORMATION OF G VECTORS TO REF. BODY FRAME

DO 362 I = 1, NH

DC 362 J = 1,3

GO TO 36

CONTINUE

DC 363 J = 1, NB

CONTINUE

C AFX. VELOCITY COMPONENTS OF EACH BODY IN REF. BODY FRAME

CONTINUE

C IF AFXFORM AUGMENTED BODY INERTIA DYNAMICS TO REF. BODY FRAME

DC 363 I = 2, NB

CONTINUE

DC 364 J = 1,3

CONTINUE

DC 364 K = 1,3

CONTINUE

DC 364 L = 1,3

CONTINUE

DC 364 M = 1,3

CONTINUE

DC 365 J = 1,3

CONTINUE

DC 365 K = 1,3

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINU
ACO MATRIX ELEMENT COMPUTATION (3x3)

DC 3001 I=1,N
DC 3001 J=1,N
A001(I,J)=A001(I,J)
CONTINUE

ACO VECTOR ELEMENT COMPUTATION (3x1)

DC 3101 K=1,NH
AV(K,I)=0.
CONTINUE
IIF(EPS(K+J),EQ.0) GO TO 42
KB=KK(K)
C(N+1)*=GM(K)*G0(K+K)+G0(K+K)+G0(K+K)+G0(K+K)+G0(K+K)
C(N+1)*=GM(K)*G0(K+K)+G0(K+K)+G0(K+K)+G0(K+K)+G0(K+K)
22 CONTINUE
42 DC 45 K=1+3
43 GO TO 61
44 C(IP)=EAI(M)+PS(I+J+M)*C(L)
45 CONTINUE
46 C(K+3)=4(I-1)
47 E(K+3)=E(K+1)+E+11-EAI11)
48 E(K+2)=E(K+2)+E(K+1)+EAI12)
49 E(K+3)=E(K+3)+E(K+3)+EAI13)
50 CONTINUE
51 DO 55 M=1+3
52 K=J*(J-1)+M
53 DC 55 M=1+3
54 CONTINUE
55 GO TO 61
56 1:O
57 CONTINUE
58 DC 60 M=1+3
59 IF(PI(M).EQ.0) GO TO 60
60 CONTINUE
61 1:O
62 CONTINUE
63 DC 65 M=1+3
64 DC 65 M=1+3
65 J=1+(J-1)+M
68 CONTINUE
69 DC 610 I=1+3
70 DC 610 J=1+3
71 IF(PI(I+J).EQ.0) GO TO 610
72 E(I)=E(I)+E(J)+E(J)+E(J)
73 CONTINUE
74 K=0
75 DO 612 I=1+3
76 IF(PI(I).EQ.0) GO TO 612
77 K=K+1
78 IV=IV*3
79 DO 611 J=1+3
80 IF(PI(I+J).EQ.0) GO TO 611
81 IF(I+J+I+J).EQ.0) GO TO 51
82 E(I+J).EQ.0) GO TO 51
83 E(I+J).EQ.0) GO TO 51
84 E(I+J).EQ.0) GO TO 51
85 1:O
86 continue
87 1:O
88 continue
89 continue
90 continue
91 continue
92 continue
93 continue
94 continue
95 continue
96 continue
97 continue
98 continue
99 continue
100 continue
101 continue
102 continue
103 continue
104 continue
105 continue
106 continue
107 continue
108 continue
109 continue
110 continue
111 continue
112 continue
113 continue
114 continue
115 continue
116 continue
117 continue
118 continue
119 continue
120 continue
121 continue
122 continue
123 continue
124 continue
125 continue
126 continue
127 continue
128 continue
129 continue
130 continue
131 continue
132 continue
133 continue
134 continue
135 continue
136 continue
137 continue
138 continue
139 continue
140 continue
141 continue
142 continue
143 continue
144 continue
145 continue
146 continue
147 continue
148 continue
149 continue
150 continue
151 continue
152 continue
153 continue
154 continue
155 continue
156 continue
157 continue
158 continue
159 continue
160 continue
161 continue
162 continue
163 continue
164 continue
165 continue
166 continue
167 continue
168 continue
169 continue
170 continue
171 continue
172 continue
173 continue
174 continue
175 continue
176 continue
177 continue
178 continue
179 continue
180 continue
181 continue
182 continue
183 continue
184 continue
185 continue
186 continue
187 continue
188 continue
189 continue
190 continue
191 continue
192 continue
193 continue
194 continue
195 continue
196 continue
197 continue
198 continue
199 continue
200 continue
201 continue
202 continue
203 continue
204 continue
205 continue
206 continue
207 continue
208 continue
209 continue
210 continue
211 continue
212 continue
213 continue
214 continue
215 continue
216 continue
217 continue
218 continue
219 continue
220 continue
221 continue
222 continue
223 continue
224 continue
225 continue
226 continue
227 continue
228 continue
229 continue
230 continue
231 continue
232 continue
233 continue
234 continue
235 continue
236 continue
237 continue
238 continue
239 continue
240 continue
241 continue
242 continue
243 continue
244 continue
245 continue
246 continue
247 continue
248 continue
249 continue
250 continue
251 continue
252 continue
253 continue
254 continue
255 continue
256 continue
257 continue
258 continue
259 continue
260 continue
261 continue
262 continue
263 continue
264 continue
265 continue
266 continue
267 continue
268 continue
269 continue
270 continue
271 continue
272 continue
273 continue
274 continue
275 continue
276 continue
277 continue
278 continue
279 continue
280 continue
281 continue
282 continue
283 continue
284 continue
285 continue
286 continue
287 continue
288 continue
289 continue
290 continue
291 continue
292 continue
293 continue
294 continue
295 continue
296 continue
297 continue
298 continue
299 continue
300 continue
301 continue
302 continue
303 continue
304 continue
305 continue
306 continue
307 continue
308 continue
309 continue
310 continue
311 continue
312 continue
313 continue
314 continue
315 continue
316 continue
317 continue
318 continue
319 continue
320 continue
321 continue
322 continue
323 continue
324 continue
325 continue
326 continue
327 continue
328 continue
329 continue
330 continue
331 continue
332 continue
333 continue
334 continue
335 continue
336 continue
337 continue
338 continue
339 continue
340 continue
341 continue
342 continue
343 continue
344 continue
345 continue
346 continue
347 continue
348 continue
349 continue
350 continue
351 continue
352 continue
353 continue
354 continue
355 continue
356 continue
357 continue
358 continue
359 continue
360 continue
361 continue
362 continue
363 continue
364 continue
365 continue
366 continue
367 continue
368 continue
369 continue
370 continue
371 continue
372 continue
373 continue
374 continue
375 continue
376 continue
377 continue
378 continue
379 continue
380 continue
381 continue
382 continue
383 continue
384 continue
385 continue
386 continue
387 continue
388 continue
389 continue
390 continue
391 continue
392 continue
393 continue
394 continue
395 continue
396 continue
397 continue
398 continue
399 continue
400 continue
401 continue
402 continue
403 continue
404 continue
405 continue
406 continue
407 continue
408 continue
409 continue
410 continue
411 continue
412 continue
413 continue
414 continue
415 continue
416 continue
417 continue
418 continue
419 continue
420 continue
421 continue
K=K+1
L=0
DC 25 J=J+1
IF(IP=IFJ1) .NE. 0 GO TO 25
L=L+1
IF(KGT.L) GO TO 26
A(K+3*L+3)=A(S(J+J))
GO TO 25
A(K+3*L+3)=A(L+3*K+3)
CONTINUE
250 CONTINUE
CALL SOD(A+V,IV+EC+V+1+92+WRK)
DC 510 J=J+1
K=J-3
IF(ISK1) .NE. 0 GO TO 911
IV=IV-1
GO TO 910
911 ECWJ=GM001(K)
910 CONTINUE
92 CONTINUE
11L0
END
Appendix B
Summary of MBDYTL, MBDYL User Requirements

Subroutine Entry Statements

CALL MBDYTL(NH,H,MB,MA,PB,PA,G,PI)
or CALL MBDYL(NH,H,MB,MA,FB,PA,G,PI)

CALL MRATE(NH,TH,TB,TA,FB,FA,GM,GMD,GMDD,WO,WDOT)

Input/Output Variable Type and Storage Specifications—Fortran V

INTEGER NH, H(n), PI(n)

REAL MB(7), MA(n,7), PB(n,3), PA(n,n,3), G(n,3), TH(n),
TB(3), TA(n,3), FB(3), FA(n,3), CM(n), GMD(n),
GMDD(n), WO(3)

DOUBLE PRECISION WDOT(n + 3)(where n = number of system hinges)

External Subroutines Called

AINVD — double precision matrix inversion subroutine; user-supplied (see line
224 in MBDYTL listing, line 223 in MBDYL listing)

This subroutine inverts the matrix A, where A is of order IV X IV,
IV ≥ 3. The result is returned in A.

Subroutine Setup

Insert the Fortran V statement (see listings)

PARAMETER Q = n

where n = number of system hinges

Data Restrictions

n ≥ 1

Core Storage Required

Code: 1500 words

Data: 500 words (minimum; increases with n)

Listings
SUBROUTINE MODYILINH, MB, MA, P9, PAR, G, PI

C ADJUSTABLE DIMENSIONS

INTEGER HI(3), PI(3)
REAL MB(3), MA(3), P9(3), PAR(3), G
PARAMETER G = 4

C ADDITIONAL DIMENSIONED VARIABLES

C DOUBLe PRECISION A(100,400), B(100)

PARAMETER S = 4

PARAMEIER S, V(3), S = 4*V(3)

C ADDITIONAL DIMENSIONED VARIABLES

DOUBLE PRECISION A(V(3), 100) B(100)

DIMENSION H(3) PS(3), X1(3), X2(3), X3(3), L(3)

DO 40 I = 1, V(3)

FEX(3) = FE0(S) = IY1(S) = IY2(S) = IY3(S) = AB(3, I)

I = (3) = (3) = PS(3) = J(3) = AV(3) = AGG(3)

INTEGER K(I), EPSI(3)

REAL LX, LY, LZ = IXX(1), IY1(1), IY2(1), IY3(1)

EQUIVALENCE (IPS, A(I), L, 1, X, Y, Z)

NS: NS = I

C INITIALIZE EPSI(K, J) WITH ZEROS AND DEFINE KK(1)

DO 85 K = 1, NS

EPSI(K, J) = 0

DO 86 J = 2, NS

IF(IK.EQ.J-1) EPSI(K, J) = 1

IF(KL.EQ.L-1) GO TO 87

GO TO 86

CONTINUE

CONTINUE

DO 89 K = 1, NS

DO 90 J = 2, NS

I = (1) (EPI, S, EQ, 1) AND (H(J-1, EQ, L) = 1)

EPSI(K, J) = 1

CONTINUE

CONTINUE

INITIAL CALCULATION OF BARYCENTER VECTORS W.R.T. BODY C.G.S AND HINGE POINTS

IXX(1) = MB(11)

IY1(1) = MB(12)

IY2(1) = MB(13)

IY3(1) = MB(14)

IX1(1) = MB(15)

IY1(1) = MB(16)

BM00(11) = MB(17)

TM = BM00(11)

DO 35 J = 2, NS

IXX(J) = MA(J-1, 1)

IY(J) = MA(J-1, 2)

IY2(J) = MA(J-1, 3)

IY3(J) = MA(J-1, 4)

IX2(J) = MA(J-1, 5)

IY2(J) = MA(J-1, 6)

BM00(J) = MA(J-1, 7)

TM = TM + BM00(J)

DO 145 I = 2, NS

CONTINUE
66* II=1-1
67* DO 149 J=1*N6
68* J1=J-1
69* IF(EPS(I1,J1,E0.1) GO TO 163
70* IF(I1.EQ.J1) GO TO 70
71* IF(I1.EQ.1) GO TO 80
72* IF(EPS(I1+J1,E0.1) GO TO 400
73* 70 LX(I1,J1)=FA(I1+I1+1)
74* LYT(J1)=PA(I1+I1+2)
75* LZ(I1+J1)=FA(I1+I1+3)
76* GO TO 149
77* 400 CONTINUE
78* DO 600 K=1,N9
79* IF(EPS(I1,K,E0.1) GO TO 500
80* 600 CONTINUE
81* GO TO 149
82* 500 LX(I1,K)=PA(I1+K,1)
83* LYT(K)=PA(I1+K,2)
84* LZ(I1+K)=PA(I1+K,3)
85* GO TO 149
86* 80 DO 90 L=1,J1
87* IF(EPS(L,J1,E0.1) GO TO 101
88* 90 CONTINUE
89* 101 LX(L,J1)=PB(L,1)
90* LYT(L,J1)=PB(L,2)
91* LZ(L,J1)=PB(L,3)
92* GO TO 149
93* 102 LX(L,J)=0.
94* LYT(L,J)=0.
95* LZ(L,J)=0.
96* 103 LX(L,J3)=P8(L,1,3)
97* 104 LX(L,J3)=0.
98* 105 LX(L,J3)=0.
99* 106 LX(L,J3)=0.
100* 107 CALCULATION OF AUGMENTED INERTIA DYADICS FOR EACH BODY
101* DO 31 N=1,N9
102* DO 110 J=1*N9
103* DO 109 N=1,N9
104* PH(N,1)=IXX(N)
105* PH(N,2)=IXY(N)
106* PH(N,3)=IZZ(N)
107* DO 31 J=1,N9
108* DO 110 N=1,N9
109* DO 109 J=1,N9
110* PH(N,1)=PH(N,1)+BMASS(N)*DZ(N)**2
111* PH(N,2)=PH(N,2)+BMASS(N)*DY(N)**2
112* PH(N,3)=PH(N,3)+BMASS(N)*DX(N)**2
113* PH(N,1,2)=PH(N,1,2)+DXX(N)**2
114* PH(N,1,3)=PH(N,1,3)+DXY(N)**2
115* PH(N,2,3)=PH(N,2,3)+DYZ(N)**2
116* PH(N)=PH(N)+BMASS(N)**2
117* DO 30 J=1,N9
118* PH(N,1)=PH(N,1)+BMASS(J)*DX(N)**2
119* PH(N,2)=PH(N,2)+BMASS(J)*DY(N)**2
120* PH(N,3)=PH(N,3)+BMASS(J)*DZ(N)**2
121* PH(N,1,2)=PH(N,1,2)+DXX(N,J)**2
122* PH(N,1,3)=PH(N,1,3)+DXY(N,J)**2
123* PH(N,2,3)=PH(N,2,3)+DYZ(N,J)**2
124* PH(N)=PH(N)+BMASS(J)**2
125* PH(N,1)=PH(N,1)+DXX(N,J)**2
126* PH(N,2)=PH(N,2)+DXY(N,J)**2
127* ADDITIONAL AUGMENTED INERTIA DYADICS (IN BODY I FRAME)
128* DO 37 T=1,N3
129* DO 75 J=1,N9
130* DO 75 I=1,N9
131* DO 75 J=1,N9
132* PH(J)*PH(N)*I=PH(J)*PH(N)*I
133* DO 37 T=1,N3
134* DO 75 J=1,N9
135* DO 75 I=1,N9
136* IF(I.EQ.J) GO TO 37
Continued...
209* CONTINUE
214* DO 250 I=1,NH
215* IF(P(I).EQ.NE.U) GO TO 250
218* K=K+1
220* DO 25 J=1,NH
222* IF(P(J).EQ.NE.U) GO TO 25
225* L=L+1
227* IF(K.GT.L) GO TO 26
250 CONTINUE
252* CONTINUE
254* CALL AINV(DA+IV*1099*WRK)
256* RETURN
258* ENTRY KFA(NH+TH*TB+TA+FB+FA+GM+GMD+GMDD+WE+EG)
260* DIMENSION TB(3),TA(NH+3),FB(3),FA(NH,3),GM(1),GMD1(1),GMDD1(1)
262* DOUBLE PRECISION EC(V),EQ(V)
264* C BODY-TO-BODY COORDINATE TRANSFORMATIONS
266* C
268* DO 335 J=1,NH
270* K=K(J)
272* AB1(I,J)=L
274* AB1(I,J)=AB1(I,J)*GM(J)*G3(J,J+3)
276* AB1(I,J)=GM(J)*G(J+2)
278* AB1(I,J)=AB1(I,J)*AB1(I,J)
280* AB1(I,J)*L
282* AB1(I,J)=AB1(I,J)*GM(J)*G(J+2)
284* AB1(I,J)=AB1(I,J)*L
286* IF(K.EQ.O) GO TO 335
288* DO 334 L=1,3
290* DO 334 M=1,3
292* T(J,J+1,L,M)=T(J,J+1,L,M)+AB1(I,J)*AB1(I,J)
294* CONTINUE
296* C
298* C COORD. TRANSFORMATION OF G VECTORS (TO REF. BODY FRAME)
300* C
302* DO 362 I=1,NH
304* DO 362 J=1,3
306* DO 362 K=I,3
308* DO 362 L=I,3
310* DO 362 M=I,3
312* CONTINUE
314* C
316* C EXTERNAL FORCES, EXTERNAL TORQUES
318* C
320* FEX0(I)=FB(I)
322* FEX0(I)=FB(I)
324* FEX0(I)=FB(I)
326* FEX0(I)=FB(I)
328* FEX0(I)=FB(I)
330* FEX0(I)=FB(I)
332* FEX0(I)=FB(I)
334* FEX0(I)=FB(I)
336* FEX0(I)=FB(I)
338* FEX0(I)=FB(I)
340* FEX0(I)=FB(I)
342* FEX0(I)=FB(I)
344* FEX0(I)=FB(I)
346* FEX0(I)=FB(I)
348* FEX0(I)=FB(I)
350* FEX0(I)=FB(I)
352* FEX0(I)=FB(I)
354* FEX0(I)=FB(I)
356* FEX0(I)=FB(I)
358* FEX0(I)=FB(I)
360* FEX0(I)=FB(I)
362* FEX0(I)=FB(I)
364* FEX0(I)=FB(I)
366* FEX0(I)=FB(I)
368* FEX0(I)=FB(I)
370* FEX0(I)=FB(I)
372* FEX0(I)=FB(I)
374* FEX0(I)=FB(I)
376* FEX0(I)=FB(I)
378* FEX0(I)=FB(I)
VECTOR CROSS PRODUCTS DESCRIBING SYSTEM ROTATIONAL COUPLING.

QUADRATIC TERMS INVOLVING THE CONNECTING BODY ANGULAR VELOCITIES AND THE MUTUAL BARYCENTER-MINGE VECTORS

DO 280 I=1,NB

DXO(I) = T(I,1,1)*DX(I+1)+T(I,1,2)*DX(I+2)+T(I,1,3)*DX(I+3)

DYO(I) = T(I,2,1)*DY(I+1)+T(I,2,2)*DY(I+2)+T(I,2,3)*DY(I+3)

DZO(I) = T(I,3,1)*DZ(I+1)+T(I,3,2)*DZ(I+2)+T(I,3,3)*DZ(I+3)

CONTINUE

C VECTOR CROSS PRODUCTS DESCRIBING SYSTEM ROTATIONAL COUPLING.

C QUADRATIC TERMS INVOLVING THE CONNECTING BODY ANGULAR VELOCITIES AND THE MUTUAL BARYCENTER-MINGE VECTORS

C

DO 230 N=1,NB

CPX=0.

CPY=0.

CPZ=0.

DO 230 L=1,NB

CPX=CPX+DY0(N,L)*FEZO(L)+DYO(N,L)*FEZ0(L)

CPY=CPY+DYO(N,L)*FEZO(L)+DX0(N,L)*FEZ0(L)

CPZ=CPZ+DX0(N,L)*FEZO(L)+DYO(N,L)*FEZ0(L)

CONTINUE

K = 3*(N-1)

E(K+1,1)=TX0(K+1)+CPX

E(K+2,1)=TY0(K+1)+CPY

E(K+3,1)=TZ0(K+1)+CPZ

CONTINUE

C COMPUTE CONNECTION ELEMENTS FOR (E) VECTOR

DO 55 MI=1,3

DO 55 J=2,NB

DO 55 M=1,3

K1=3*(J-1)+M

EC(MI)=EC(MI)+E(K1)

I=0

DO 60 K=1,K1

IF(PI(K).NE.0) GO TO 60

EC(I+3)=EC(I+3)+GO(KL).CE(KL)

EC(I+3)=EC(I+3)+TH(IK)

CONTINUE

DO 610 I=1,3

DO 610 J=1,NB

IF(PI(J).NE.0) GO TO 610

EC(I+3)=EC(I+3)+GO(KL).CE(KL)

EC(I+3)=EC(I+3)+TH(IK)

CONTINUE

DO 610 I=1,3

DO 610 J=1,NB

IF(PI(J).NE.0) GO TO 610

EC(I+3)=EC(I+3)+GO(KL).CE(KL)

EC(I+3)=EC(I+3)+TH(IK)

CONTINUE

DO 610 I=1,3

DO 610 J=1,NB

IF(PI(J).NE.0) GO TO 610

EC(I+3)=EC(I+3)+GO(KL).CE(KL)

EC(I+3)=EC(I+3)+TH(IK)

CONTINUE

DO 610 I=1,3

DO 610 J=1,NB

IF(PI(J).NE.0) GO TO 610

EC(I+3)=EC(I+3)+GO(KL).CE(KL)

EC(I+3)=EC(I+3)+TH(IK)

CONTINUE
350* IF(I.GT.J) AS(I,J)=AS(J,I)
351* EC(I+3)=EC(I+3-I,J)+AS(I,J)*GMDC(I)
352* CONTINUE
353* CONTINUE
354* C SOLVE SYSTEM MATRIX FOR REFERENCE BODY ANG. ACCELERATION AND FINGE
355* C (RELATIVE) ROTATIONAL ACCELERATIONS
356* C
357* DO 671 J=1,IV
358* F0(I,J)=0.
359* DO 671 J=1,IV
360* EQ(I,J)=EQ(I,J)+AS(I,J)*EC(J)
361* CONTINUE
362* DO 910 J=1,IV
363* K=J-3
364* IF(P(K).NE.0) GO TO 911
365* EQ(I,J)=EQ(I,J)
366* IV=IV+1
367* GO TO 910
368* 911 EQ(I,J)=GMDC(K)
369* 92 CONTINUE
370* 910 CONTINUE
371* RETURN
372* END
**SUBROUTINE** MBODYLH.value + MBH.value + PA.value + GS.value

**ADJUSTABLE DIMENSIONS**

**INTEGER** HI.value + PI.value

**REAL** MBH.value + MBH.value + PA.value + PA.value + S.value + G.value + S.value

**PARAMETER** S.value + V.value + S.value + S.value + S.value

**ADDITIONAL DIMENSIONED VARIABLES**

**DOUBLE PRECISION** A.value + W.value + R.value + MB.value

**DIMENSION** PHI.value + S.value + D.value + S.value + D.value + S.value + S.value + S.value

**REAL** LX.value + LY.value + LZ.value

**INTEGER** K.value + K.value + S.value + S.value

**REAL** LX.value + LY.value + LZ.value

**C** INITIALIZE EPS.value + J.value WITH ZEROS AND DEFINE K.value + K.value

**DO** 85 K = 1, HI.value

**DO** B5 J = 1, NB

**DEFINE EPS.value + J.value USING HI.value**

**GO** TO 86

85 CONTINUE

87 CONTINUE

88 CONTINUE

89 CONTINUE

8A CONTINUE

8B CONTINUE

**C** INITIAL CALCULATION OF BART CENTER VECTORS W.R.T. BODY C.G.S. AND HINGE POINTS

**GO** TO 35

**GO** TO 35

35 CONTINUE
11:1=I-1
65  DO 149 J=1,NB
66  JI=J-1
67  IRI.EQ.JI GO TO 169
68  IRI.GT.JI GO TO 70
69  IRI.EQ.JI GO TO 80
70  IF(EPSI.JI.EQ.JI) GO TO 460
71  LXII+JI=PA4111.1+1)
72  LYII+JI=PA4111.2)
73  LZII+JI=PA4111.3)
74  GO TO 149
75  CONTINUE
76  GO TO 143
77  DO 60 K=1,J1
78  IF(EPSI.JI.EQ.JI) GO TO 600
79  CONTINUE
80  GO TO 149
81  LXII+JI=PA4111+K+1)
82  LYII+JI=PA4111+K+2)
83  LZII+JI=PA4111+K+3)
84  GO TO 149
85  GO TO 149
86  DO 96 L=1,J1
87  IF(EPSI.L+J1.EQ.11) GO TO 600
88  CONTINUE
89  GO TO 149
90  LXII+J1=DIK1
91  LYII+J1=DIK2
92  LZII+J1=DIK3
93  GO TO 149
94  DO 149 I=1,NB
95  LXII+J1=DIK1
96  CONTINUE
97  DO 13 N=1,NB
98  DO 13 J=1,NB
99  DXX=J1,JLX(N+J)
100  DYY=JN,JLY(N+J)
101  DZZ=JN,JLZ(N+J)
102  IF(J=1) GO TO 149
103  DXX=JN,J=DXN+J1-18MASS(K)*LXK(M)
104  DYY=JN,J=DIY(N+J1-18MASS(K)*LYK(M)
105  DZZ=JN,J=DIZ(N+J1-18MASS(K)*LZI(M)
106  CONTINUE
107  DO 31 N=1,NB
108  DO 31 J=1,NB
109  P411=N+1,11=IXX(N)
110  P411=N+1,21=IXY(N)
111  P411=N+1,31=IXZ(N)
112  P411=N+2,11=IYY(N)
113  P411=N+2,21=IZY(N)
114  P411=N+2,31=IZZ(N)
115  DO 30 J=1,NB
116  P411=N+1,11=PHIN+1,11+BMASS(J1)*DYIN+J1+20ZIN+J1+2)
117  P411=N+1,21=PHIN+1,21-BMASS(J1)*DXX+N+J1+OYIN+J1
118  P411=N+1,31=PHIN+1,31-BMASS(J1)*DZI+N+J1+OXY(N+J1
119  P411=N+2,11=PHIN+2,11+BMASS(J1)*DXIN+J1+OYIN+J1
120  P411=N+2,21=PHIN+2,21-BMASS(J1)*DXIN+J1+OYIN+J1+20ZIN+J1+20ZIN+J1+2)
121  P411=N+2,31=PHIN+2,31-BMASS(J1)*DXIN+J1+OYIN+J1+20ZIN+J1+20ZIN+J1+20ZIN+J1+2)
122  30 P411=N+3,11=PHIN+3,11+BMASS(J1)*10XIN+J1+2+0YIN+J1+2+0YIN+J1+2+0YIN+J1+2+0YIN+J1+2+0YIN+J1+2+
123  P411=N+3,21=PHIN+3,21+BMASS(J1)*10XIN+J1+2+0YIN+J1+2+0YIN+J1+2+0YIN+J1+2+0YIN+J1+2+0YIN+J1+2+
124  P411=N+3,31=PHIN+3,31+BMASS(J1)*10XIN+J1+2+0YIN+J1+2+0YIN+J1+2+0YIN+J1+2+0YIN+J1+2+
125  31 P411=N+3,31=PHIN+3,31
126  CONTINUE
127  ADDITIONAL AUGMENTED INERTIA DYADICS (IN BODY I FRAME)
128  DO 751 J=1,NB
129  DO 751 M=1,N
130  DO 751 N=1,N
131  DO 751 M=1,N
132  PS1=J+M+1=PHN(J+M+N)
133  DO 37 I=1,NB
134  DO 37 J=1,NB
135  IF(EPSI.J+J GO TO 37
C     AOC MATRICE ELEMENT COMPUTATION (3X3)
C
DO 3 I=1,N3
DO 3 J=1,N3
AOC(I,J):AOCC(I,J)*PS(I,J,1)
AOC(J,I):AOCC(J,I)*PS(I,J,1)
AOC(I,J):AOCC(I,J)*PS(I,J,1)
AOC(J,I):AOCC(J,I)*PS(I,J,1)

CONTINUE
C
C     ACC VECTOR ELEMENT COMPUTATION (3X1)
C
DO 7 K=1,NH
    AV(K,1):=0.
    AV(K,2):=0.
    AV(K,3):=0.

DO 14 J=1,NB
    IF((EPS(J,1).GT.0).OR.(EPS(J,2).LT.0).OR.(EPS(J,3).LT.0)) GO TO 14
    AV(J,1):=0.
    AV(J,2):=0.
    AV(J,3):=0.

DO 14 K=1,NH
    IF(K.GT.M) GO TO 14
    AIS(K,M):=0.
    AIS(K,M):=0.
    AIS(K,M):=0.

DO 14 I=1,NB
    IF(EPS(J,1).LE.0) GO TO 14
    AIS(J,I):=0.
    AIS(J,I):=0.
    AIS(J,I):=0.

 CONTINUE
C
C     AKM SCALAR ELEMENT COMPUTATION
C
DO 18 K=1,NH
    DO 18 M=1,NH
        IF(K.GT.M) GO TO 18
        AIS(K,M):=0.
        AIS(K,M):=0.
        AIS(K,M):=0.

DO 18 I=1,NB
    IF(EPS(J,1).GT.0).OR.(EPS(J,2).LT.0).OR.(EPS(J,3).LT.0)) GO TO 18
    AIS(J,I):=0.
    AIS(J,I):=0.
    AIS(J,I):=0.

 CONTINUE
C
C     LOAD SYSTEM MATRIX (A) WITH AOC,AKM,AKM ELEMENTS
C
DO 23 I=1,N
    DO 23 J=1,N
        A(I,J):=AOC(I,J)*PS(I,J,1)

DO 23 J=1,N
    DO 23 I=1,N
        K=I

DO 24 J=1,N
    IF((EPS(J,1).LE.0).OR.(EPS(J,2).LE.0).OR.(EPS(J,3).LE.0)) GO TO 24
    K=I

DO 24 J=1,N
    IF(EPS(J,1).LE.0).OR.(EPS(J,2).LE.0).OR.(EPS(J,3).LE.0)) GO TO 24
    A(K,1):=0.
    A(K,2):=0.
    A(K,3):=0.\n
CONTINUE
C
C
CONIINUE
K=0
DO 25 J=1,NH
I=IPv(NJ,I)
L=I+1
IF1PI(L,NH).NE.0) GO TO 25
K=K+
DO 25 J=1,NH
I=IPv(NJ,31)
IF1PI(I,NH).NE.0) GO TO 25
L=L*I
DO 25 J=1,NH
AR(K+3,L,3)=ASK(IJ)
GO TO 25
A(K,31,L)=A(K,31,L)+AE(IJ)
DO 230 K=I+1,NH
ENTRY MRATEINH,BTHTPAlFBFAl GM.GMDwGMDD#WO,
EXTERNAL FORCES, EXTERNAL TORQUES, HINGE TORQUES
VECTOR CROSS PRODUCTS DESCRIBING SYSTEM ROTATIONAL COUPLING.
QUADRATIC TERMS INVOLVING THE CONNECTING BODY ANGULAR VELOCITIES
AND THE MUTUAL BARYCENTER-HINGE VECTORS
DO 230 N1,NB
CPX=0.
CPY=0.
CPZ=0.
DO 230 L=1,NB
CPX=CPX+DYN(L,J)*FEZ(IJ)-DZIN(L,J)*FEY(IJ)
CPY=CPY+DZIN(L,J)*FEY(IJ)-DYN(L,J)*FEZ(IJ)
CPZ=CPZ+DYN(L,J)*FEY(IJ)-DYN(L,J)*FEZ(IJ)
CONTINUE
K=3*N1+1
DO 264 M=1,NH
EKM+1=TNM1+CPX
EKM+2=TNM1+CPY
EKM+3=TNM1+CPZ
CONTINUE
DO 264 M=1,NH
DO 265 M=1,NH
DO 55 MI=1,NH
EcIMi=1
DO 55 J=1,NH
EcJm=1
DO 55 M=1,NH
Ki=J+1+1
DO 52 I=1,NH
Il=IPv(I)
DO 52 I=1,NH
Il=IPv(I)
DO 52 I=1,NH
Il=IPv(I)
JPL TECHNICAL REPORT 32-1592

SOLVE SYSTEM MATRIX FOR REFERENCE BODY ANG. ACCELERATION AND HINGE RELATIVE) ROTATIONAL ACCELERATIONS

DO 571 I=1,IV
EQ(I)=0.
DO 571 J=1,IV
EQ(I)=EQ(I)+AI(I,J)*EC(IJ)
CONTINUE
DO 910 J=1,N
EQ(I)=EQ(I)+AI(I,J)*EC(IJ)
CONTINUE
GO TO 910
EQ(J)=EQ(IJ)
IV=IV-1
GO TO 310
EQ(J)=OMOD(IK)
Appendix C
Summary of MBDYPL User Requirements

Subroutine Entry Statements

CALL MBDYPL(NH,H,MB,MA,PA,PB,G,PI,LI,LI,TI)
CALL MRATE(NH,TH,TB,TA,FB,FA,GM,GMD,GMD,GMDD,WO,WDOT)

Input/Output Variable Type and Storage Specifications—Fortran V

INTEGER NH, H(n), PI(n), LI(n), TI(n)
REAL MB(7), MA(n,7), PB(n,3), PA(n,n,3), G(n,3), TH(n),
   TB(3), TA(n,3), FB(3), FA(n,3), GM(n), GMD(n),
   GMDD(n), WO(3)
DOUBLE PRECISION WDOT(n + 3)(where n = number of system hinges)

External Subroutines Called

1. AINVd—double precision matrix inversion subroutine; user-supplied (see line 272 in MBDYPL listing)

This subroutine inverts the matrix AT, where AT is of order NMT X NMT, NMT ≥ 1. The result is returned in AT.

2. SOD—double precision matrix equation solving subroutine; user-supplied (see line 666 in MBDYPL listing)

This subroutine solves the matrix equation Ax = E, where A is of order NL X NL, NL ≥ 3.

Subroutine Setup

Insert the Fortran V statement (see listing)

PARAMETER Q = n

where n = number of system hinges.

Data Restrictions

n ≥ 1

Core Storage Required

Code: 3000 words
Data: 1000 words (minimum; increases with n)

Listing
SUBROUTINE MODPLINH,N,M,B,A,P,G,PI,LH

ADJUSTABLE DIMENSIONS

INTEGER H(J),L(I),LH(I),H(I)
REAL M(J),MAINH(I),PS(INH),PI(A,NH),PI(NH),HI(NH)
PARAMETER 3=4
PARAMETER U=0.1,Y=0.3,N=4.5,V=5.3=U

ADDITIONAL DIMENSIONED VARIABLES

DOUBLE PRECISION A1,V,WS,X,JAS
DOUBLE PRECISION A1(V),AC(V),CE(V),AV
DIMENSION PS(U),U,DX(U),DZ(U),P(U),OZ(U),U
S
LX(U),LY(U),LZ(U),PH(U),J31
DIMENSION A1(U),V(1),A1(U),B1(U),C1(U)
PARAMETER a=4
PARAMETER U=0.4,V=0.4
ADDITIONAL PRECISION AIVUP1,WRKIV41BMASS(U)
ADDITIONAL PRECISION AT10(U)
EQUIVALENCE (A1,PS),LX(IDX),B1(III),LY(III),LZ(III),PH(III)
DIMENSION PS(U),U,DX(U),DZ(U)
DIMENSION A1(U),V(1),A1(U),B1(U),C1(U)
S
U=0.5
N=NH+1
IU=1
DO 85 K=1,NH
LISTM=LISHM+LISH
INITIALIZE EPS(K,J) WITH ZEROS AND DEFINE KK1K
DO 85 K=1,NH
KK(K)=K
DO 85 J=1,NH
EPS(K,J)=0
DO 86 J=2,NH
EPS(K,J)=1
IF(I.EQ.J) GO 10 808
GO TO 10 809
LL=LL+1
CONTINUE
GO TO 10 809
CONTINUE
LOCATE POTENTIAL TERMINAL APPENDAGE BODIES

M(NH)=0
M=0
LL=0
DO 805 CONTINUE
DO 808 J=1,NH
IF(H,J,EQ,1) GO 10 808
DO 809 LL=LL+1
TAMM=L
CONTINUE
DO 809 LL=LL+1
TAMM=L
K=TAMM
JPL TECHNICAL REPORT 32-1592 121
IF(TAML1.EQ.0) GO TO 810
NPI=0
67.* IF(I+EPSKJ.EQ.1) NPI=NEP+1
68.* IF(I+EPSKJ.EQ.1) AND.(I+EPSKJ.EQ.1) NPI=NPI+1
69.* 807 CONTINUE
70.* IF(NPI.EQ.1) GO TO 810
71.* GO TO 805
72.* CONTINUE
73.* IF(NPI.EQ.NEP) GO TO 810
74.* M=TAML1
75.* IF(NPI.EQ.NEP) G0 10
76.* TAML1=0
77.* 820 CONTINUE
78.* 820 CONTINUE
79.* 820 CONTINUE
80.* 820 CONTINUE
81.* GO TO 827
82.* 829 CONTINUE
83.* 829 CONTINUE
84.* 829 CONTINUE
85.* 829 CONTINUE
86.* C LOCATE TERM. APPENDAGE BASE BODIES AND NO. OF T.A. BODIES ON EACH
87.* C
88.* C
89.* GO 836 L=1+N
90.* NUMFIL=0
91.* GO 835 L=1+N
92.* GO 834 K=1+N
93.* IF(TAML1.EQ.K) GO TO 834
94.* N=N+1
95.* GO TO 833
96.* IF(I+EPSKJ.EQ.1) GO TO 833
97.* NUMFIN=NUMFIN+1
98.* 833 CONTINUE
99.* 833 CONTINUE
100.* 833 CONTINUE
101.* C FIND BASE BODY WITH THE MOST TERMINAL APPENDAGE BODIES ON IT
102.* C
103.* C
104.* NMX=0
105.* GO 837 L=1+N
106.* IF(NUMFIL.LE.NMX) GO TO 838
107.* GO TO 837
108.* 838 TAB=L-1
109.* NMX=NUMFIL
110.* 837 CONTINUE
111.* C IDENTIFY MEMBERS OF THE TERMINAL APPENDAGE SET
112.* C
113.* C
114.* GO 898 L=1+N
115.* TILL=0
116.* IF(TAB.EQ.NH) GO TO 896
117.* GO 899 J=1+N
118.* K=TAML1(J)
119.* IF(K.EQ.TAB) GO TO 899
120.* GO TO 899
121.* 899 CONTINUE
122.* 899 CONTINUE
123.* 899 CONTINUE
124.* 899 CONTINUE
125.* 899 CONTINUE
126.* 899 CONTINUE
127.* C INITIAL CALCULATION OF BARYCENTER VECTORS W.R.T. BODY C.G.'S
128.* C AND HINGE POINTS
129.* C
130.* C
131.* IXX(I1)=MB(I1)
132.* IYY(I1)=MB(I1)
133.* IZZ(I1)=MB(I1)
134.* IXY(I1)=MB(I1)
135.* IXZ(I1)=MB(I1)
136 IYZ(J)=XMB(1)
137 BMAS(I):=N(J)(1)
138 TM:=BMAS(I)
139 DO 35 J=2,NB
140 IXX(J)=MA(J-1,1)
141 IYY(J)=M(J-1,2)
142 IZZ(J)=MA(J-1,3)
143 IXY(J)=MA(J-1,4)
144 IXZ(J)=MA(J-1,5)
145 IYZ(J)=MA(J-1,6)
146 BMAS(I)=MA(I,1)
147 35 TM:=BMAS(I)
148 DO 149 I=1,NB
149 IX(J)=I
150 DO 149 J=1,NB
151 J=J+1
152 IF(I.EQ.J) GO TO 163
153 IF(I.GT.J) GO TO 70
154 IF(EPS(J).GT.0) GO TO 400
155 DO 70 IX(J)=PA(I)
156 DO 70 J=1,NB
157 IX(J)=PA(J)
158 GO TO 149
159 70 LX(J)=PA(J)
160 400 CONTINUE
161 DO 500 K=1,NB
162 IF(EPS(I).GT.0) GO TO 500
163 500 CONTINUE
164 GO TO 149
165 DO 100 K=1,NB
166 LY(J)=PA(I)
167 LZ(J)=PA(J)
168 GO TO 149
169 100 DO 90 L=1,NB
170 IF(EPS(J).GT.0) GO TO 103
171 90 CONTINUE
172 GO TO 149
173 LX(I,J)=PBL(J)
174 LY(I,J)=PBL(J)
175 LZ(I,J)=PBL(J)
176 GO TO 149
177 103 LX(I,J)=G
178 LY(I,J)=G
179 LZ(I,J)=G
180 149 CONTINUE
181 DO 13 N=1,NB
182 DO 13 J=1,NB
183 DX(N,J)=LX(N,J)
184 DY(N,J)=LY(N,J)
185 DZ(N,J)=LZ(N,J)
186 GO TO 149
187 DX(N,J)=DX(N,J)-BMAS(K)/TM*LX(N,K)
188 DY(N,J)=DY(N,J)-BMAS(K)/TM*LY(N,K)
189 DZ(N,J)=DZ(N,J)-BMAS(K)/TM*LZ(N,K)
190 C
191 C CALCULATION OF AUGMENTED INERTIA DIADICS FOR EACH BODY
192 C
193 DO 31 N=1,NB
194 PH(N)=IX(N)
195 PH(N)=IX(N)
196 PH(N)=IX(N)
197 PH(N)=IX(N)
198 PH(N)=IX(N)
199 PH(N)=IX(N)
200 DO 30 J=1,NB
201 PX(N,J)=PH(N,J)+BMAS(J)*DY(N,J)+2*DZ(N,J)
202 PH(N,J)=PX(N,J)
203 PH(N,J)=PX(N,J)
204 PH(N,J)=PX(N,J)
205 PH(N,J)=PX(N,J)
206 30 PH(N,J)=PX(N,J)
207* P(N+2*13)=PH(N,1*2)
208* PH(N+3*13)=PH(N,1*3)
209* P(N+3*2)=PH(N,2*3)
210* DO 571 R=1,NB
211* DO 571 J=1,R
212* DO 571 K=1,3
213* 571 PSI(R+J*K)=PH(R+J*K)
214* DO 763 J=-1,NH
215* GO TO 763
216* IF(NH).NE.1) GO TO 10
217* IF(NH).NE.1) GO TO 10
218* DO 3700 R=1,NB
219* DO 3700 S=1,NB
220* 3700 CONTINUE
221* IF(R.EQ.S) GO TO 10
222* IF(R.EQ.S) GO TO 10
223* IF(R.EQ.S) GO TO 10
224* IF(R.EQ.S) GO TO 10
225* IF(R.EQ.S) GO TO 10
226* IF(R.EQ.S) GO TO 10
227* IF(R.EQ.S) GO TO 10
228* IF(R.EQ.S) GO TO 10
229* IF(R.EQ.S) GO TO 10
230* IF(R.EQ.S) GO TO 10
231* IF(R.EQ.S) GO TO 10
232* IF(R.EQ.S) GO TO 10
233* IF(R.EQ.S) GO TO 10
234* IF(R.EQ.S) GO TO 10
235* IF(R.EQ.S) GO TO 10
236* IF(R.EQ.S) GO TO 10
237* 3700 CONTINUE
238* I=0
239* DO 1401 I=1,NH
240* IF(I).EQ.1) GO TO 1401
241* I=I+1
242* IF(I).EQ.1) GO TO 1401
243* K=0
244* DO 1400 K=1,NH
245* IF(I).EQ.1) GO TO 1400
246* K=K+1
247* IF(I).EQ.1) GO TO 1400
248* DO 1400 J=1,NH
249* DO 1400 S=2,NB
250* 1400 CONTINUE
251* DO 1802 J=1,3
252* DO 1802 L=1,3
253* CW(I,J)=0
254* DO 1802 J=1,3
255* CE(I,J)=0
256* DO 1802 L=1,3
257* CE(I,J)=CE(I,J)+PS(R+J,L)*G1*K+L
258* DO 1802 J=1,3
259* CW(I,J)=CW(I,J)+CE(I,J)
260* 1800 CONTINUE
261* ASB(I+1)= G(I,J) - CW(I,J) + G(I,J) - CW(I,J) + G(I,J) - CW(I,J) + G(I,J) - CW(I,J)
262* IF(I).EQ.1) OR IF(I).EQ.1) GO TO 1400
263* AT(I,J)=ASB(I,J)
264* AT(I,J)=AT(I,J)
265* 1400 CONTINUE
266* 1401 CONTINUE
267* 1402 CONTINUE
268* ENTRY MRATEINH.
269* DIMENSION I(NMT.EQ.0) G0 10 1422
270* MRATEINH.
271* IF(NM*EQ.0) GO TO 1422
272* CALL AINK(1,GMH1,*1422,WRK)
273* 1422 CONTINUE
274* RETURN
275* ENTRY MRATEINH.
276* DIMENSION I(NH+1) NH+1,FB,FA,GM,HMD,GMDD,WG,EO
277* RETURN
DOUBLE PRECISION ECIV)

BODY-TO-BODY COORDINATE TRANSFORMATION MATRICES

DO 335 J=1,NH
K=0
IF(I(J,J).EQ.0) GO TO 3021
SGM=SIGMA(J,J)

CGM=SGRT1-J-SGM*21-

IF(ISC(SGM(J,J)).LT.1.1*570796327) GO TO 5020
IF(ISC(SGM(J,J)).GT.1.4*712358981) GO TO 5019

CGM=SGM

DO 5019 I1-

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

COORD. TRANSFORMATION OF 6 VECTORS TO REF. BODY FRAME

DO 362 I=1,NH
DO 362 J=1,NH
DO 362 K=1,NH

ANG. VELOCITY COMPONENTS OF EACH BODY (IN REF. BODY FRAME)

DO 361 J=1,NH
DO 361 K=1,NH
355 361 CONTINUE
356 C TRANSFORM AUGMENTED BODY INERTIA DYADICS TO REF. BODY FRAME
357 C
358 DO 363 J=2,NB
359 DO 364 J=1,J-1
360 DO 364 K=1,J-1
361 A(I,J,K)=0.0
362 DO 364 L=1,J
363 A(I,J,K)=A(I,J,K)+PH(I,J,L)*T(I,L,K)
364 CONTINUE
365 364 CONTINUE
366 DO 366 J=1,N
367 DO 365 K=1,J
368 PSI(X,J,K)=0.0
369 DO 365 L=1,J
370 PSI(X,J,K)=PSI(X,J,K)+PH(I,J,L)*T(I,L,J)*A(I,J,K)
371 365 CONTINUE
372 363 CONTINUE
373 DO 373 J=1,N
374 DO 471 K=1,J
375 471 PSI(1,J,K)=PSI(1,J,K)
376 FEXC(J)=F8(1)
377 FEXC(J)=F8(2)
378 FEXC(J)=F8(B3)
379 T0(I,J)=T0(I,J)
380 T0(I,J)=T0(B3)
381 T0(I,J)=T0(B3)
382 DO 17 J=2,NB
383 K=1-J
384 FEXC(J)=T1(I,J,1,1)*FA(K,1)+T1(I,J,2,1)*FA(K,2)+T1(I,J,3,1)*FA(K,3)
385 FEXC(J)=T1(I,J,1,2)*FA(K,1)+T1(I,J,2,2)*FA(K,2)+T1(I,J,3,2)*FA(K,3)
386 FEXC(J)=T1(I,J,1,3)*FA(K,1)+T1(I,J,2,3)*FA(K,2)+T1(I,J,3,3)*FA(K,3)
387 T0(I,J)=T0(I,J)*TA(I,1,2,1)*TA(I,2,2,1)*TA(I,3,2,1)*TA(I,1,1,2)*TA(I,2,1,2)*TA(I,3,1,2)*TA(I,1,1,3)*TA(I,2,1,3)*TA(I,3,1,3)
388 T0(I,J)=T0(I,J)*TA(I,1,2,1)*TA(I,2,2,1)*TA(I,3,2,1)*TA(I,1,1,2)*TA(I,2,1,2)*TA(I,3,1,2)*TA(I,1,1,3)*TA(I,2,1,3)*TA(I,3,1,3)
389 17 CONTINUE
390 DO 390 J=1,NB
391 DO 400 I=1,NB
392 DY(I,J)=DY(I,J)
393 400 CONTINUE
394 C ADDITIONAL AUGMENTED INERTIA DYADICS (IN REF. BODY FRAME)
395 DO 402 J=1,NB
396 DO 403 J=1,NB
397 C VECTOR CROSS PRODUCTS DESCRIBING SYSTEM ROTATIONAL COUPLING.
398 C SQUARE TERMS INVOLVING THE CONNECTING BODY ANGULAR
399 C VELOCITIES AND THE MUTUAL GARYCENTER-HINGE VECTORS
400 DO 400 J=1,NB
401 C
402 IF(I.EQ.J) GO TO 37
403 PS1(I,J,J)=(-1.0+D0(I,J,J))*D2(I,J,J)*D2(I,J,J)
404 PS1(I,J,J)=(-1.0+D0(I,J,J))*D2(I,J,J)
405 PS1(I,J,J)=(-1.0+D0(I,J,J))*D2(I,J,J)
406 PS1(I,J,J)=(-1.0+D0(I,J,J))*D2(I,J,J)
407 PS1(I,J,J)=(-1.0+D0(I,J,J))*D2(I,J,J)
408 PS1(I,J,J)=(-1.0+D0(I,J,J))*D2(I,J,J)
409 PS1(I,J,J)=(-1.0+D0(I,J,J))*D2(I,J,J)
410 PS1(I,J,J)=(-1.0+D0(I,J,J))*D2(I,J,J)
411 PS1(I,J,J)=(-1.0+D0(I,J,J))*D2(I,J,J)
412 PS1(I,J,J)=(-1.0+D0(I,J,J))*D2(I,J,J)
413 PS1(I,J,J)=(-1.0+D0(I,J,J))*D2(I,J,J)
414 37 CONTINUE
415 C
416 C
```
        CPX=G
        CPY=G
        CPZ=G
        DO 2301 L=1,NB
        IF(N.EQ.L) GO TO 2301
        WDX=WYO(L) + DZ0(L) + WZ0(L) + DYO(L) + N
        WDX=WYO(L) + DO(L) + W0(L) + DYO(L) + N
        WDX=IM+WYO(L) + DYO(L) + WDX + WDX + WDX + WDX + WDX + WDX
        WDX=IM+WYO(L) + WDX + WDX + WDX + WDX + WDX + WDX + WDX + WDX
        WDX=IM+WYO(L) + WDX + WDX + WDX + WDX + WDX + WDX + WDX + WDX
        WDX=IM+WYO(L) + WDX + WDX + WDX + WDX + WDX + WDX + WDX + WDX + WDX
        CPX=CPX+WDX
        CPY=CPY+WDX
        CPZ=CPZ+WDX
        CPX=CPX+WDX
        CPY=CPY+WDX
        CPZ=CPZ+WDX
        DO 3C01 J=103
        DO 3 I=1,NB
        A00(I,3) = A00(I,3) + 0.
        2301 CONTINUE
        DO 3001 J=3,NB
        DO 3000 K=1,NH
        A00(K,J) = A00(K,J) + 0.
        3000 CONTINUE
        3001 CONTINUE
```
DO 15 J=-1, NB
DO 15 I=1, N9

IFI(EPSI.JI.EQ.0).OR.IEPSIM).EQ.0)
GO TO 10

DO 18 N=1, 3
DO 18 L=-1, 3
AIS(N)=AIS(N)*PS(K+1)*I=N+L)*G01(K+1)

CONTINUE

DO 41 M=1, 3
DO 42 K=I, H
IF(EPSI.K).NE.0)
GO TO 41
K9=KKIK)+I
CW(1)=CW(1)*GM01(K)+I-W01(K)+I*G01(K+3)
CW 2=1:CW 2)*G01(K+1)-W01(K)*G01(K+1)
CONTINUE

DO 45 M=1, 3
L=1, 3
EAIMI:EAIMI*PS
CONTINUE

K:3=I-2)
E(Kl+2)=E(K1-4)
E(K1+1)=E(K1+1)
CONTINUE

DO 55 M=1, 3
L=1, 3
CE(MI=ECIM)EI(J1)
CONTINUE

DO 60 K=1, N9
IF(PIIJK).NE.0)
GO TO 60
I=I+1

ECI 1=0.
DO 601 M=1, 3
DO 601 I=1, 3
IC=IC+1
ECI 1=ECI 1+I
CONTINUE

DO 610 I=1, 3
DO 610 J=1, NB
EC(I+3)=EC(I+3)*G01(K+L)*CE(I)
EC(I+3)=EC(I+3)*TH(K)
CONTINUE

K=0
IV=3
DO 612 I=1, 3
IF(PIIJK).NE.0)
GO TO 612
K=K+1
IV=IV+1
DO 613 J=1, 3
564* IF (PI(II).EQ.0) GO TO 611
565* I(I+Q1,J) ASI+J = AS(I+J)
566* EC(I+3)+ = EC(I+3)+ ASI+J+GMD(J)
567* 611 CONTINUE
568* 612 CONTINUE
569* C
570* C LOAD SYSTEM MATRIX (A) WITH AG0*AGK*AGM ELEMENTS
571* C
572* DO 23 I=1,3
573* DO 23 J=1,3
574* 23 A(I+J)=AG(I+J)
575* DO 24 I=1,3
576* K=0
577* DO 24 J=1,NH
578* IF (PI(I)+NE+G) GO TO 60 10 24
579* K=K+1
580* A(K+3)+ = AV(I+J)
581* A(I+K+3)+ = AV(I+J)
582* 24 CONTINUE
583* K=0
584* DO 250 I=1,NH
585* IF (PI(I)+NE+G) GO TO 60 10 250
586* K=K+1
587* L=0
588* DO 25 J=1,NH
589* IF (PI(I)+NE+G) GO TO 60 250
590* L=L+1
591* IF (K/G0L) GO TO 626
592* A(K+3)+ = AS(I+J)
593* 26 A(K+3)+ = AS(I+J)
594* 25 CONTINUE
595* 250 CONTINUE
596
597* C
598* C SOLVE SYSTEM MATRIX FOR REFERENCE BODY ANG. ACCELERATION AND HINGE
599* C RELATIVE ROTATIONAL ACCELERATIONS
600* C
601* C EXCHANGE ROWS IN A-MATRIX AND RH. SIDE VECTOR
602* C
603* DO 619 K=1,3
604* E1(K) = EC1(K)
605* DO 619 I=1,NIV
606* 619 E1(K)+ = A(K+I)
607* L=0
608* DO 620 K=1,NH
609* IF (T1(K)+EQ.+1) GO TO 620
610* L=L+1
611* E1(K)+ = EC1(K)
612* DO 621 I=1,NIV
613* 621 E1(K)+ = A(K+I)
614* 620 CONTINUE
615* DO 630 K=1,NH
616* IF (T1(K)+EQ.+1) GO TO 630
617* L=L+1
618* E1(K)+ = EC1(K)
619* DO 631 I=1,NIV
620* 631 E1(K)+ = A(K+I)
621* 630 CONTINUE
622* C
623* C EXCHANGE COLUMNS IN A-MATRIX
624* C
625* DO 639 K=1,3
626* DO 639 I=1,NIV
627* 639 A(I)+K = AE1(I)+K
628* L=0
629* DO 640 K=1,NH
630* IF (T1(K)+EQ.+1) GO TO 640
631* L=L+1
632* DO 640 I=1,NIV
633* 641 A(I)+K = AE1(I)+K
634* 640 CONTINUE
635* DO 650 K=1,NH

JPL TECHNICAL REPORT 32-1592 129
IF(IIKH*EQ.0) GO TO 650
L=L+1
DO 651 I=1,IV
651 A(I,I+3)=AEII*K+3)
650 CONTINUE
C
C SOLVE FOR HINGE-ANGLE ACCELERATIONS (NON-TERM, APPENDAGE)
C
NL=IV-NMT
IF(NMT.EQ.0) GO TO 651
DO 652 I=1,NL
DO 552 K=1,NMT
AL(I*K)=0.
DO 552 L=1,NMT
LL=NL*L
AL(I*K)=AL(I*K)*A(I*L)+AI*L*K)
CONTINUE
C
SOLVE FOR HINGE-ANGLE ACCELERATIONS (TERM, APPENDAGE)
C
IF(NMT.EQ.0) GO TO 6581
DO 657 I=1,NMT
II=NL+I
ECII=EEII
DO 657 L=1,NL
II=NL+I
ECII=ECII-AII*L*EEII)
DO 658 I=1,NMT
II=NL+I
EEII=GO.
DO 658 L=1,NMT
EEII=EEII+AI*I*L*ECII
CONTINUE
C
RECONSTRUCT SOLUTION VECTOR IN ORIGINAL ORDER
C
ECII=EEII)
EC(2)=EEII)
EC(3)=EEII)
L=3
DO 559 K=1,NH
IF(IIKH*K.EQ.11) OR.(PIK*K.EQ.1)) GO TO 659
ECII=EEII)
CONTINUE
L=NL
DO 661 K=1,NH
IF(IIIKH*K.EQ.11) OR.(PIK*K.EQ.1)) GO TO 661
L=NL
ECII=EEII)
CONTINUE
DO 662 K=1,NH
IF(PIKH*K.EQ.1) GO TO 662
ECII=ECII+GMDOIKK
CONTINUE
92 CONTINUE
RETURN
END
References


