1wp

~~N71-14589~~

71-00172

3

MCR-70-457 (Vol I)

FINAL REPORT

SPIN VECTOR CONTROL FOR A SPINNING SPACE STATION

VOLUME I: USER'S MANUAL

Get DRF

By:

T. Hendricks
W. Guderian
G. Johnson
G. Haynes

November 1970

Prepared under Contract No. NAS8-25247 by

MARTIN MARIETTA CORPORATION
Denver Division
P. O. Box 179
Denver, Colorado 80201

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

FOREWORD

This document represents Volume II of the final report on NASA Huntsville Contract entitled "Spin Vector Control for a Spinning Space Station". The report is prepared in two volumes:

Volume I - User's Manual

Volume II - Analytical Manual

MCR-70-457 (Vol I)

FINAL REPORT

SPIN VECTOR CONTROL FOR A SPINNING SPACE STATION

VOLUME I: USER'S MANUAL

By:

T. Hendricks
W. Guderian
G. Johnson
G. Haynes

November 1970

Prepared under Contract No. NAS8-25247 by

MARTIN MARIETTA CORPORATION
Denver Division
P. O. Box 179
Denver, Colorado 80201

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

TABLE OF CONTENTS

FINAL REPORT

SPIN VECTOR CONTROL OF A ROTATING SPACE STATION

VOLUME I: USER'S MANUAL

By: T. Hendricks, Walter Guderian, George Haynes, Gary Johnson

SUMMARY

This document presents the formulation, computational logic, input/output options, subroutine description and other pertinent information that should aid the user of the SPIN VECTOR CONTROL COMPUTER PROGRAM (MD246).

I. INTRODUCTION

This document is concerned with the design use and implementation of a digital computer program to facilitate the study of the dynamic behaviour and control of dual spin space vehicles. This volume is a companion to Volume II (analytical manual) of the final report under NASA-Huntsville Contract NAS8-25247.

The Spin Vector Control Program (MD246) is a Fortran Program that was written and checked out using the CDC 6400/6500 digital computer. To minimize possible system incompatabilities care has been exercised to assume that only the basic features of the system are used. Thus the program should be operable on most digital machines with a FORTRAN 4 compiler.

The program is capable of solving the rotational dynamics of dual spin earth orbiting spacecraft. Several control options as well as spacecraft configurations are possible. Among the available control actuators are CMGs, reaction wheels, reaction jets and torque motors. This program is intended for but not restricted to attitude control studies of a rotating space station. The generalized spacecraft configuration along with geometrical definitions is shown in Figure 1. Figure 2 is a specific spacecraft configuration.

The remaining contents of this document discusses in varying degree of detail how to use the program. The first chapter Input Deck Construction describes those cards which are necessary when exercising the various program options. Chapter 2 Data Deck User's Guide presents a complete sequence and format description of all the data input cards. For a description and definition of the input variables refer to Appendix A.

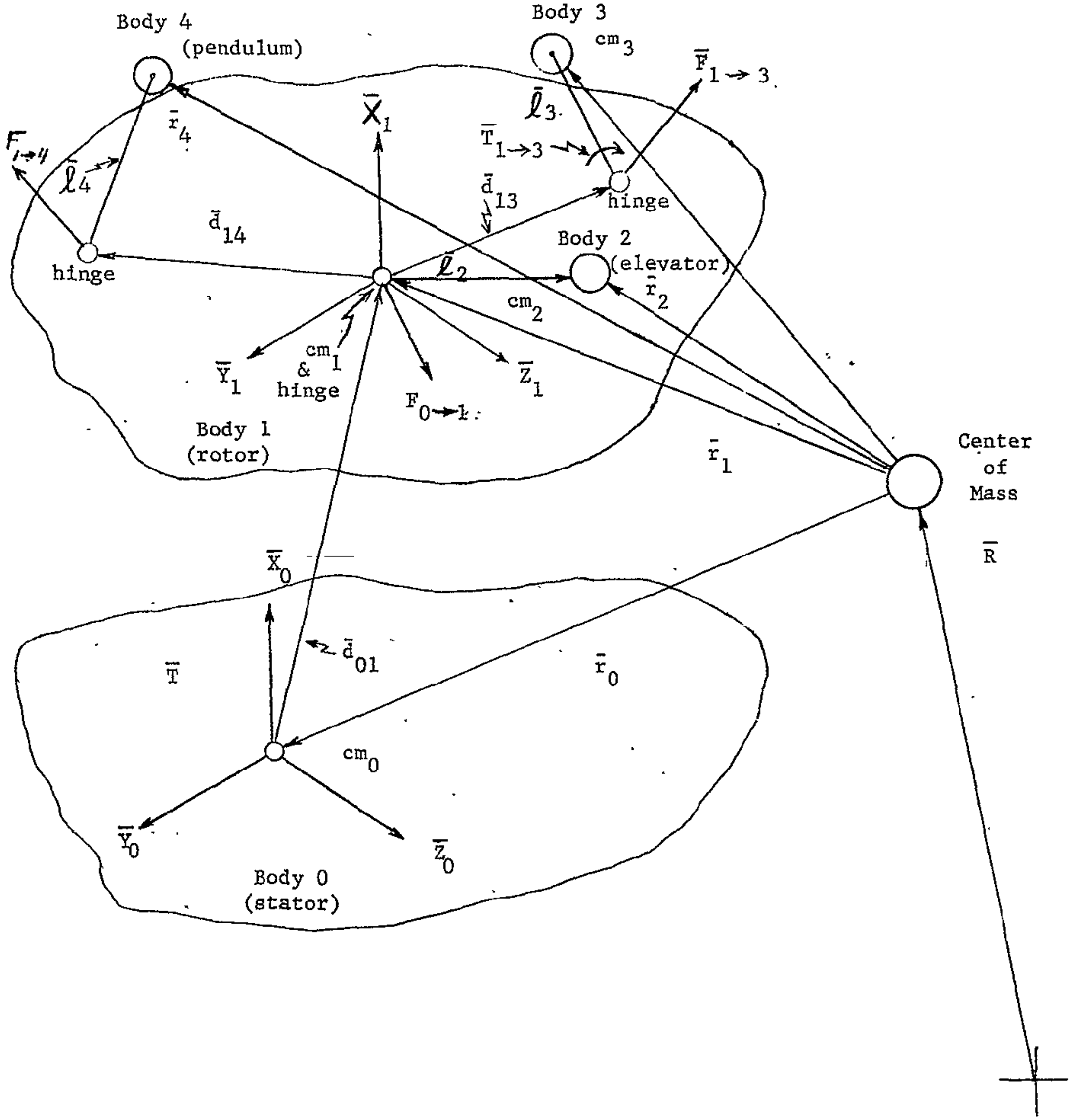


Figure 1 General Body Configuration

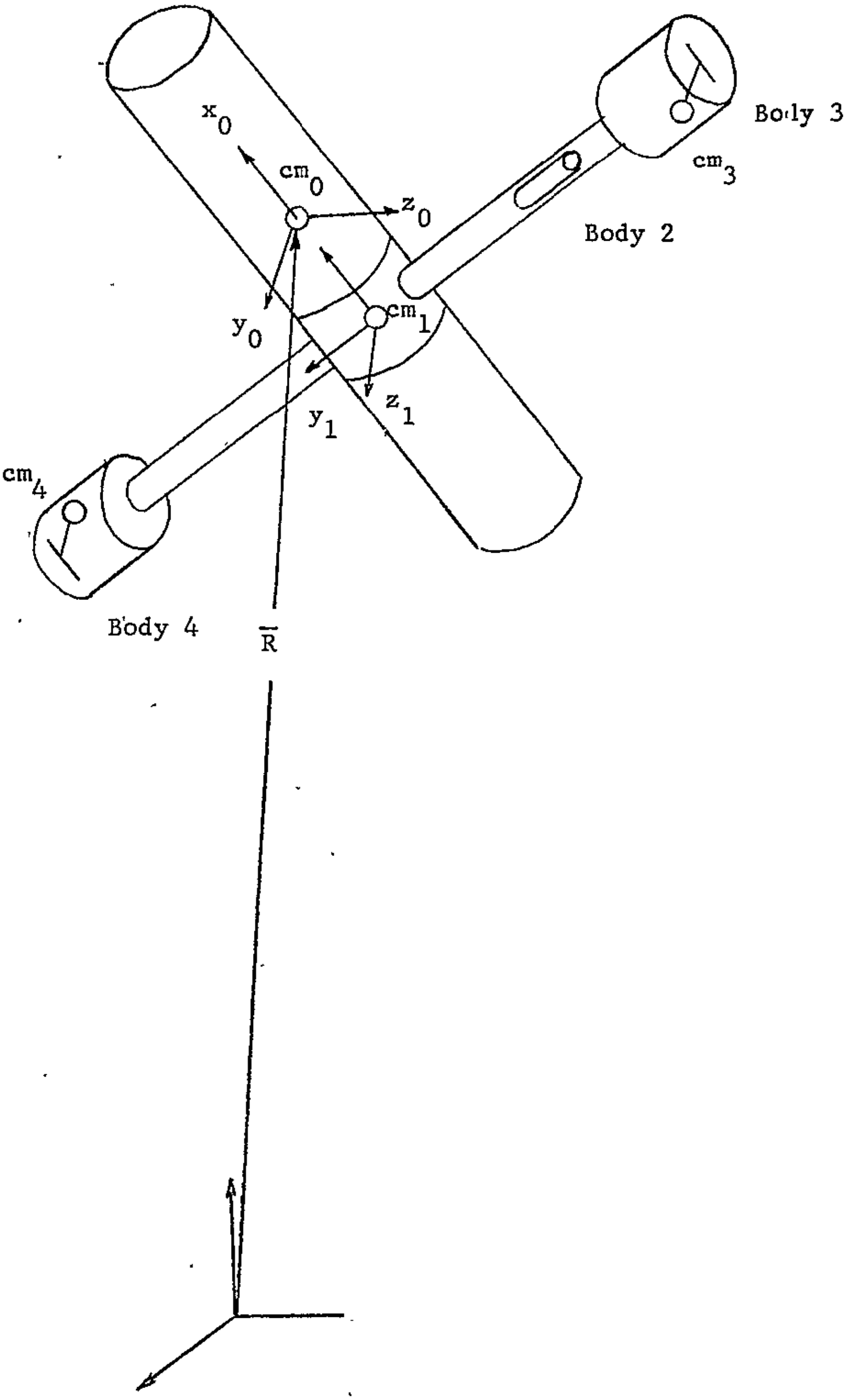


Figure 2   Spacecraft Body Configuration

INPUT DECK CONSTRUCTION

The basic data deck for operating the program with essentially no designated options is as follows.

Card #	Variable	
1	NDECK = __	
2	IPNDLM = 0	First card of run data
3	IPRINT = __	
4	TSTART, TSTOP, DELTAT	
5	Alt.	
6-14	TIBOI(1,1) - TIBOI(3,3)	
15	WO(1), WO(2), WO(3)	
16	BOMASS	
17-25	BODYOI(1,1) - BODYOI(3,3)	
26	NUMCMG = 0	
225	IPROPF = 0	
242	B1MASS	
243-251	BODY1I(1,1) - BODY1I(3,3)	
252	THETA1	
253	OMEGA1	
254	DO1(1), DO1(2), DO1(3)	
255	IB2F = 0	
282	SP	
283	NGAIN = 0	
294	IGRAVF = 0	
295	IDOCK = 0	

Note that the above is also a list of the cards that must always be present.

Some examples of data deck arrangements for various options will be given. All data decks consist of the basic data deck with changes indicated.

1. Pendulum

   Card #2 IPNDLM = 1

   Before card #282 add cards #268 through #281, the pendulum parameter and initial value data.

2. Two single DOF CMGs

   Card #26 NUMCMG = 2

   After card #26 add -

   Card #27 IDOF(1) = 1
   #28 - HW(1)
   #29 - #37 AOCJ(1,1,1) - AOCJ(1,3,3)
   #38 - #42 AII(1,1,1) - AII(1,3,3)
   #47 - THATA(1)
   #48 - THATAD(1)
   #60 - IDOF(2) = 1
   #61 - HW(2)
   #62 - #70 AOCJ(2,1,1) - AOCJ(2,3,3)
   #71 - #79 AII(2,1,1) - AII(2,3,3)
   #80 - THATA(2)
   #81 - THATAD(2)

Card #283 NGAIN = 5

After card #283 add cards #284 - #288

(Cards #287 is control gain for CMG #1 and card #288 is control gain for CMG #2 in the present subroutines)

3. Propulsion on Body 0 (no attitude control)

Card #225 IPROPF = 1

After card #225 add -

Card #226 IATTIF = 0

Card #230 AOJ(1) = (non-zero) CGAINO(1)

Card #231 AOJ(2) = (non-zero) CGAINO(2)

Card #232 AOJ(3) = (non-zero) CGAINO(3)

After card #259, if present, otherwise after card #255 add -

Card #260 A1J(1) = (non-zero) CGAIN1(1) = 0

Card #261 A1J(2) = (non-zero) CGAIN(2) = 0

4. Movable Mass

Card #255 IB2F = 1

After card #255 add cards #256 - #259 pertaining to movable mass

5. Attitude Control with Propulsion on Body 1

Card #225 IPROPF = 1

After card #225 add -

Card #226 IATT1F = 1

Card #227 CA(1), CA(2), CA(3)

Card #230 AOJ(1) = (non-zero) CGAIN(1) = 0

Card #231 AOJ(2) = (non-zero) CGAIN(2) = 0

Card #232 AOJ(3) = (non-zero) CGAIN(3) = 0

After card #259 if present, otherwise after card #255 add -

Card #260 A1J(1) = (non-zero) CGAIN1(1)

Card #261 A1J(2) = (non-zero) CGAIN1(2)

6. Gravity Gradient

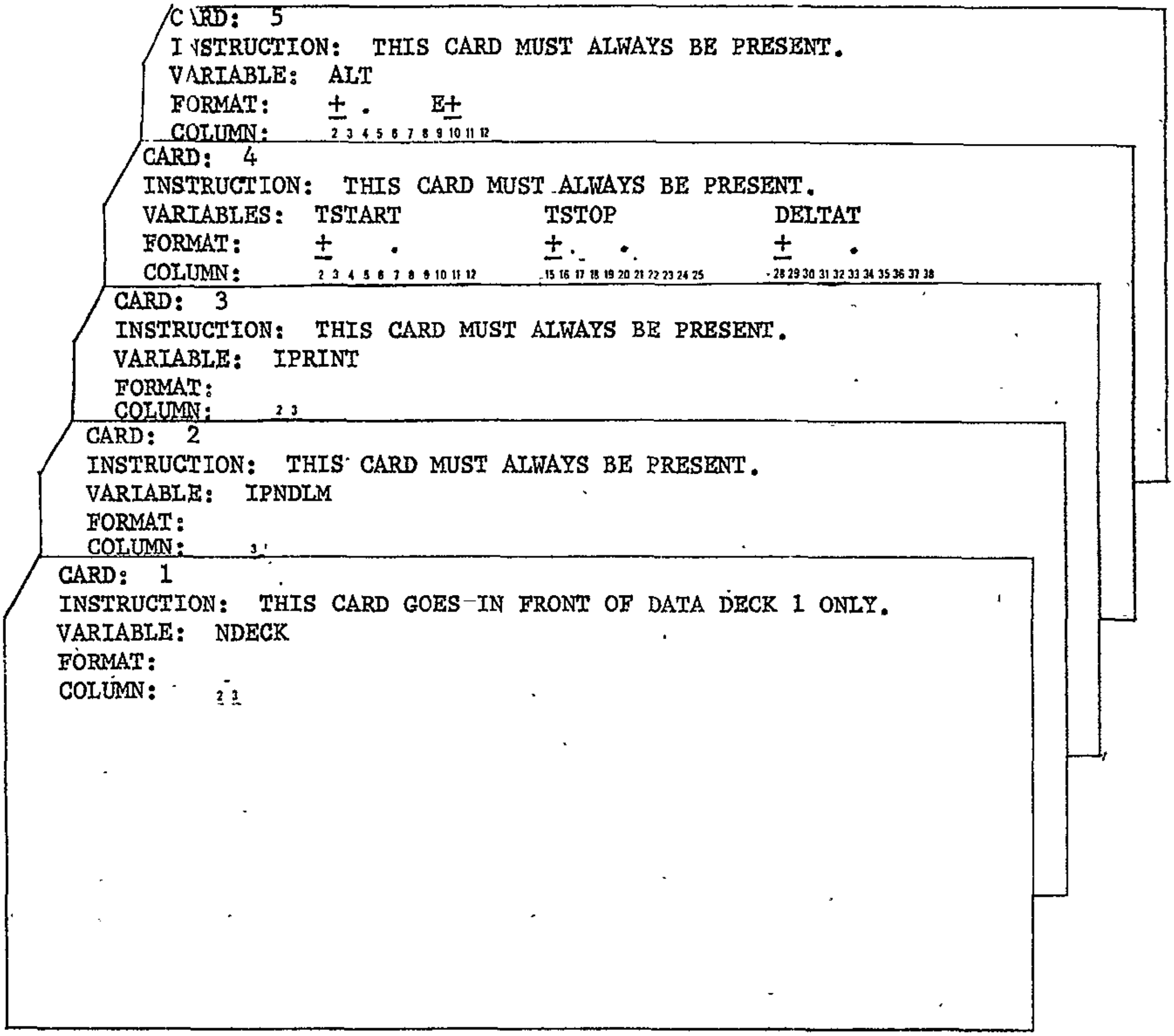
Card #294 IGRAVF = 1

7. Docking

Card #295 IDOCK = 1

After card #295 add cards #296 - #307 docking quantities.

DATA DECK USER'S GUIDE



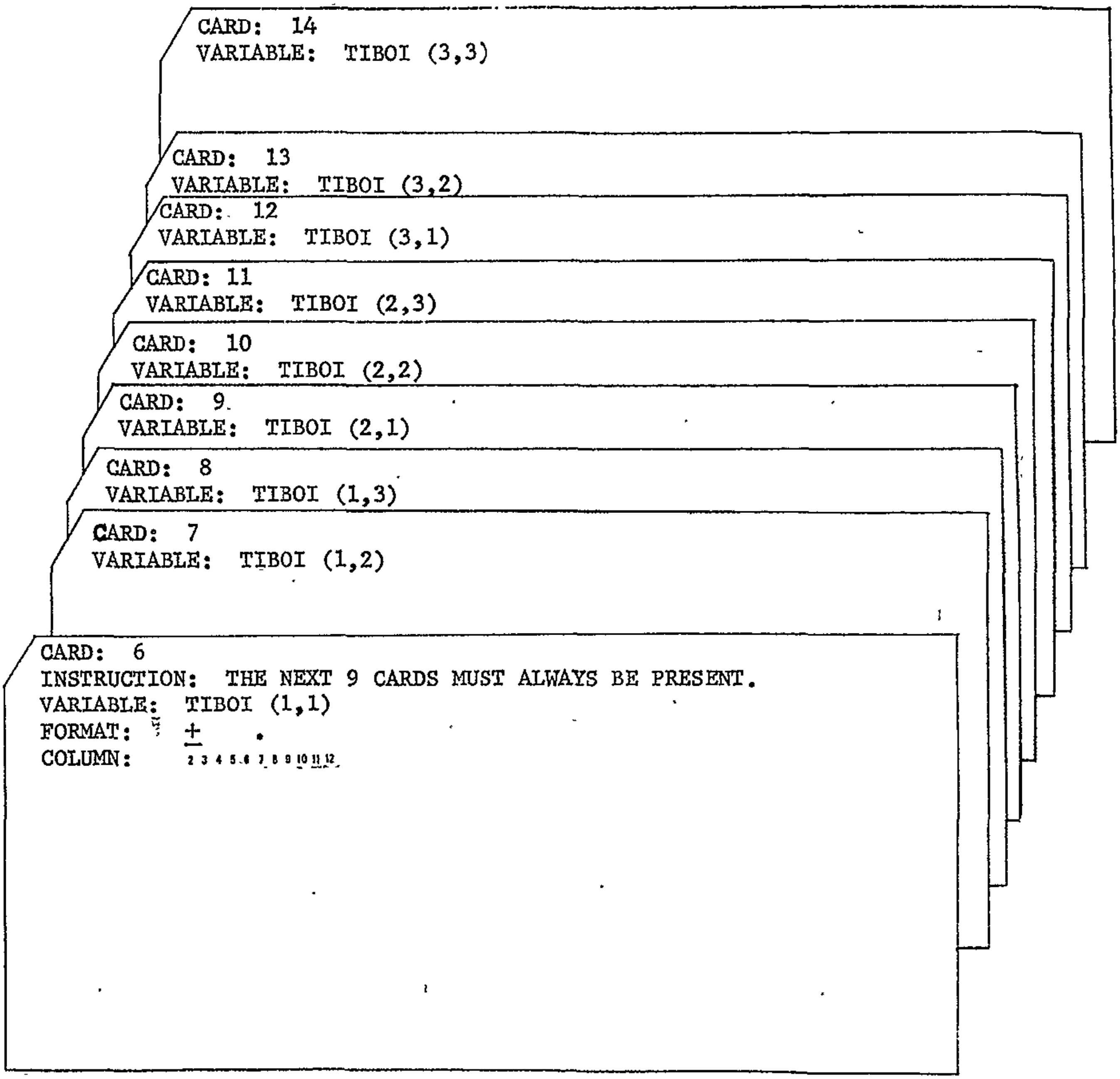
CARD: 5
INSTRUCTION: THIS CARD MUST ALWAYS BE PRESENT.
VARIABLE: ALT
FORMAT: ± . E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 4
INSTRUCTION: THIS CARD MUST ALWAYS BE PRESENT.
VARIABLES: TSTART TSTOP DELTAT
FORMAT: ± . ± . ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12 15 16 17 18 19 20 21 22 23 24 25 28 29 30 31 32 33 34 35 36 37 38

CARD: 3
INSTRUCTION: THIS CARD MUST ALWAYS BE PRESENT.
VARIABLE: IPRINT
FORMAT:
COLUMN: 2 3

CARD: 2
INSTRUCTION: THIS CARD MUST ALWAYS BE PRESENT.
VARIABLE: IPNDLM
FORMAT:
COLUMN: 3

CARD: 1
INSTRUCTION: THIS CARD GOES IN FRONT OF DATA DECK 1 ONLY.
VARIABLE: NDECK
FORMAT:
COLUMN: 2 3



CARD: 14
VARIABLE: TIBOI (3,3)

CARD: 13
VARIABLE: TIBOI (3,2)

CARD: 12
VARIABLE: TIBOI (3,1)

CARD: 11
VARIABLE: TIBOI (2,3)

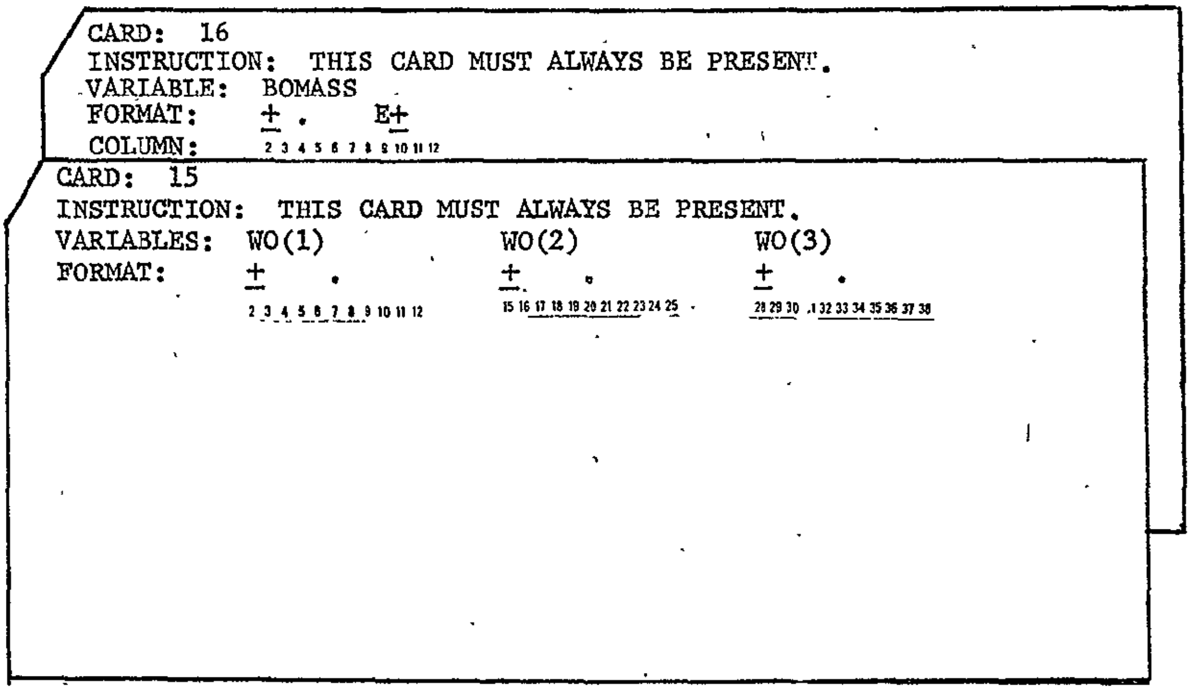
CARD: 10
VARIABLE: TIBOI (2,2)

CARD: 9
VARIABLE: TIBOI (2,1)

CARD: 8
VARIABLE: TIBOI (1,3)

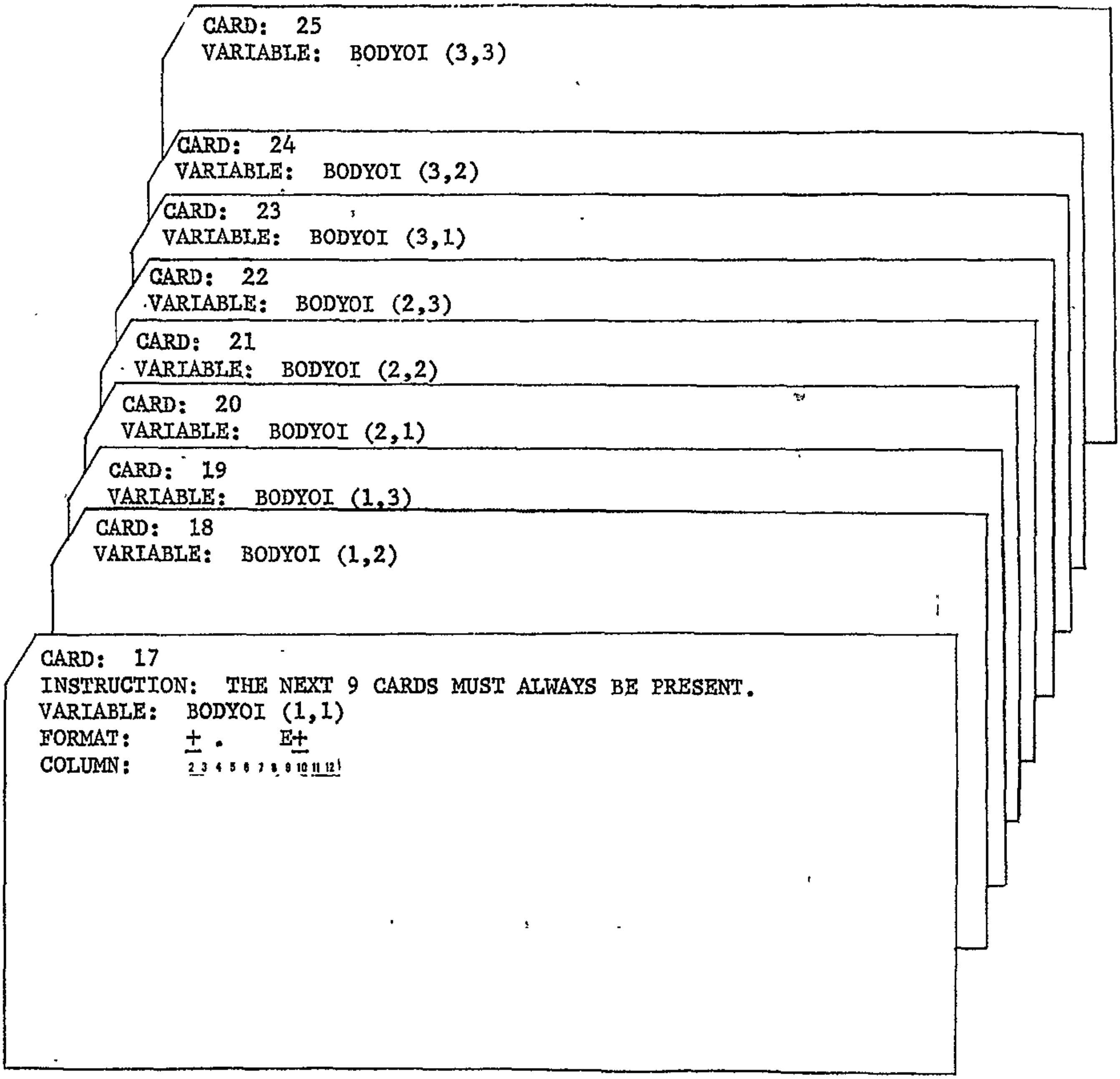
CARD: 7
VARIABLE: TIBOI (1,2)

CARD: 6
INSTRUCTION: THE NEXT 9 CARDS MUST ALWAYS BE PRESENT.
VARIABLE: TIBOI (1,1)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12



CARD: 16
INSTRUCTION: THIS CARD MUST ALWAYS BE PRESENT.
VARIABLE: BOMASS
FORMAT: ± . E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 15
INSTRUCTION: THIS CARD MUST ALWAYS BE PRESENT.
VARIABLES: WO(1) WO(2) WO(3)
FORMAT: ± . ± . ± .
2 3 4 5 6 7 8 9 10 11 12 15 16 17 18 19 20 21 22 23 24 25 28 29 30 31 32 33 34 35 36 37 38



CARD: 25
VARIABLE: BODYOI (3,3)

CARD: 24
VARIABLE: BODYOI (3,2)

CARD: 23
VARIABLE: BODYOI (3,1)

CARD: 22
VARIABLE: BODYOI (2,3)

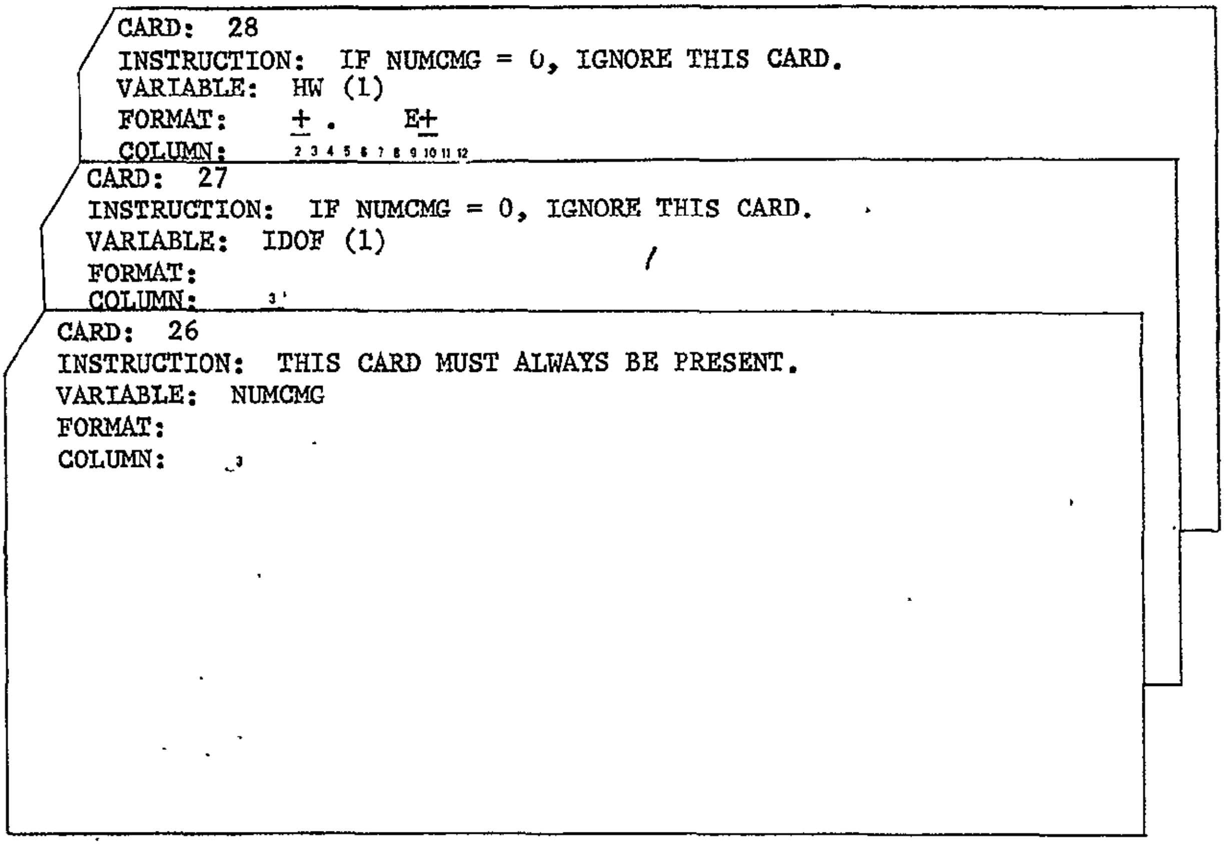
CARD: 21
VARIABLE: BODYOI (2,2)

CARD: 20
VARIABLE: BODYOI (2,1)

CARD: 19
VARIABLE: BODYOI (1,3)

CARD: 18
VARIABLE: BODYOI (1,2)

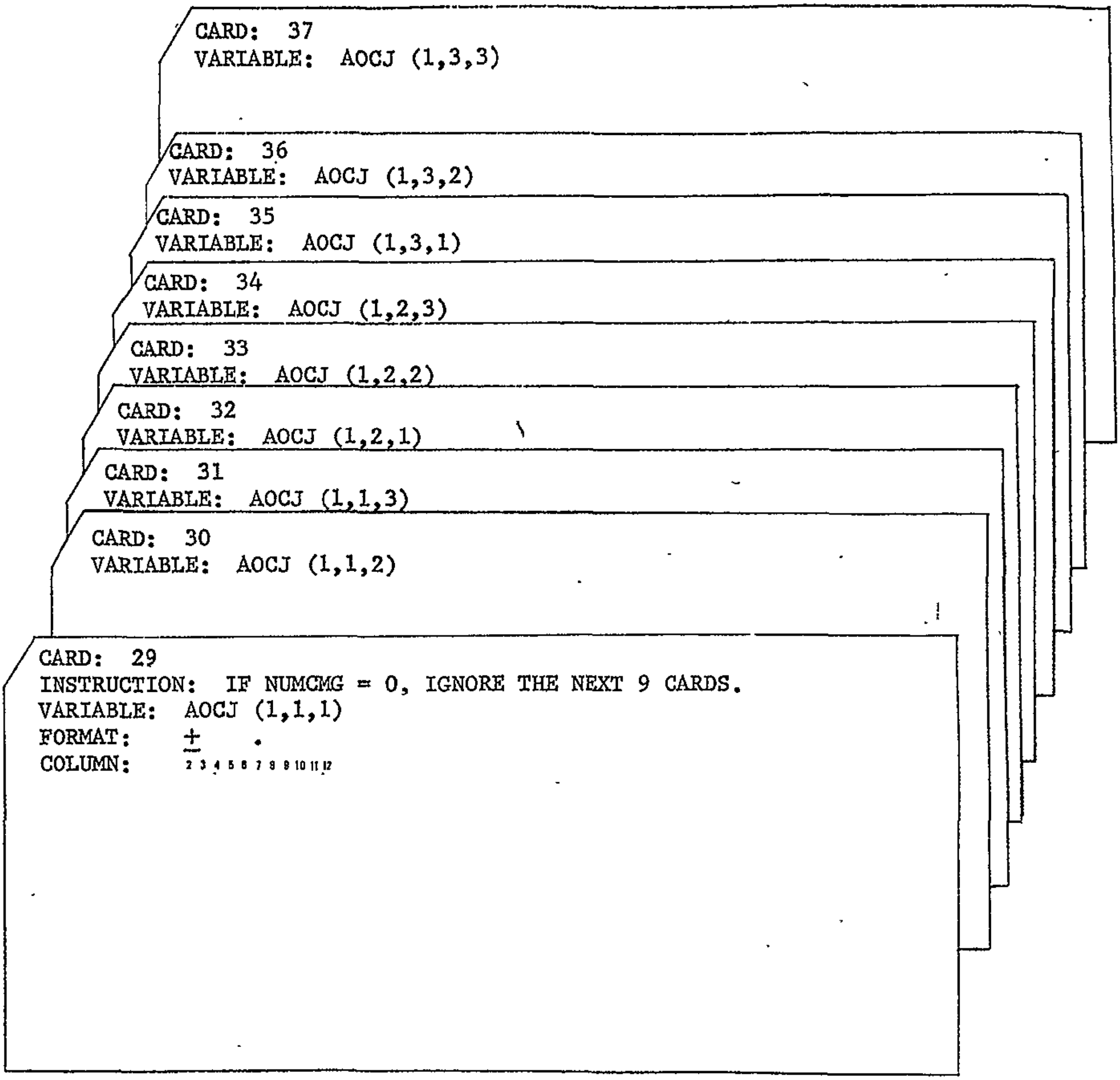
CARD: 17
INSTRUCTION: THE NEXT 9 CARDS MUST ALWAYS BE PRESENT.
VARIABLE: BODYOI (1,1)
FORMAT: ± .   E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12



CARD: 28
INSTRUCTION: IF NUMCMG = 0, IGNORE THIS CARD.
VARIABLE: HW (1)
FORMAT: ± . E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 27
INSTRUCTION: IF NUMCMG = 0, IGNORE THIS CARD.
VARIABLE: IDOF (1)
FORMAT:
COLUMN: 3

CARD: 26
INSTRUCTION: THIS CARD MUST ALWAYS BE PRESENT.
VARIABLE: NUMCMG
FORMAT:
COLUMN: 3



CARD: 37
VARIABLE: AOCJ (1,3,3)

CARD: 36
VARIABLE: AOCJ (1,3,2)

CARD: 35
VARIABLE: AOCJ (1,3,1)

CARD: 34
VARIABLE: AOCJ (1,2,3)

CARD: 33
VARIABLE: AOCJ (1,2,2)

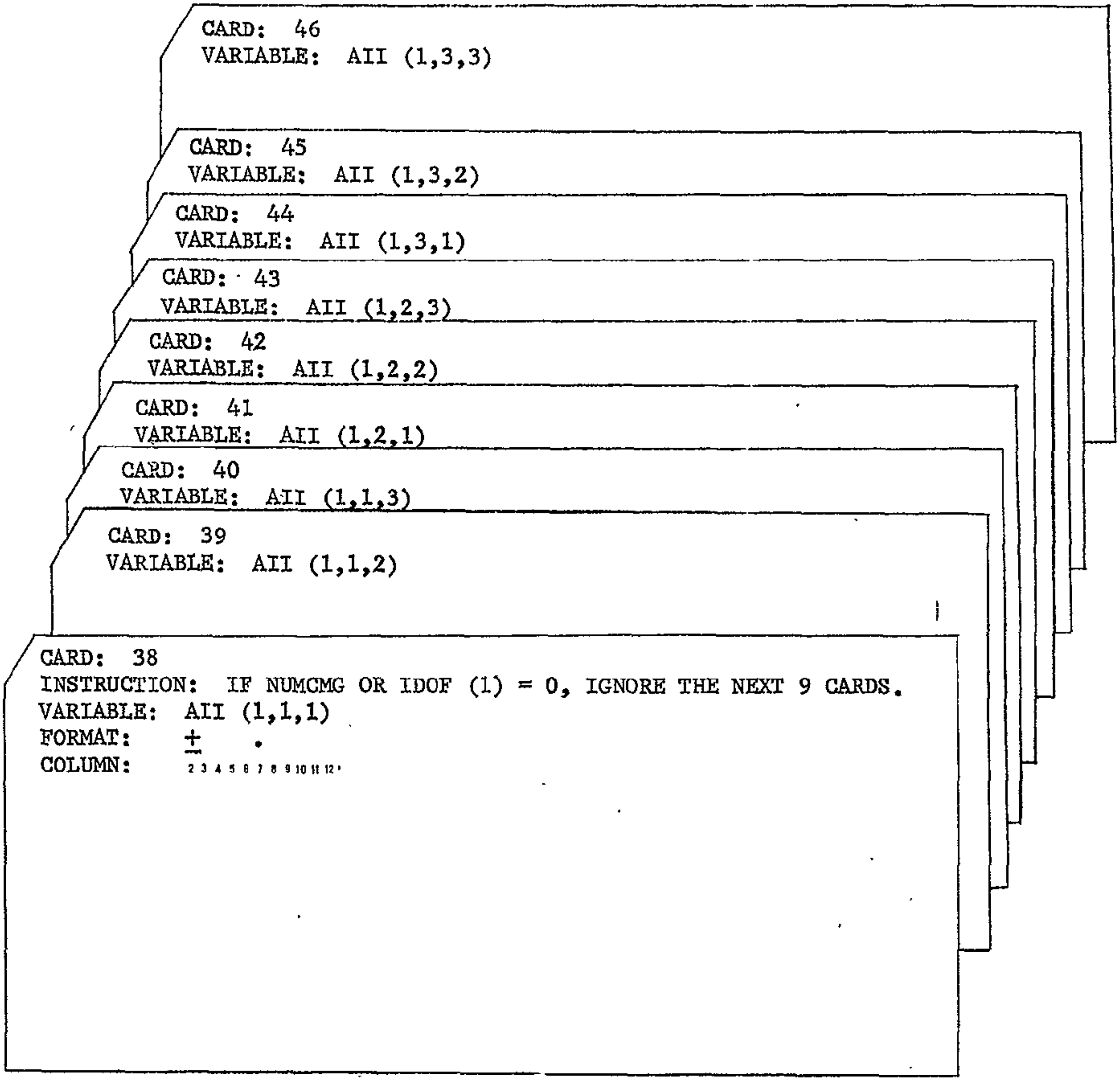
CARD: 32
VARIABLE: AOCJ (1,2,1)

CARD: 31
VARIABLE: AOCJ (1,1,3)

CARD: 30
VARIABLE: AOCJ (1,1,2)

CARD: 29
INSTRUCTION: IF NUMCMG = 0, IGNORE THE NEXT 9 CARDS.
VARIABLE: AOCJ (1,1,1)

```
FORMAT:     ±         .
COLUMN:     2 3 4 5 6 7 8 9 10 11 12
```



CARD: 46
VARIABLE: AII (1,3,3)

CARD: 45
VARIABLE: AII (1,3,2)

CARD: 44
VARIABLE: AII (1,3,1)

CARD: 43
VARIABLE: AII (1,2,3)

CARD: 42
VARIABLE: AII (1,2,2)

CARD: 41
VARIABLE: AII (1,2,1)

CARD: 40
VARIABLE: AII (1,1,3)

CARD: 39
VARIABLE: AII (1,1,2)

CARD: 38
INSTRUCTION: IF NUMCMG OR IDOF (1) = 0, IGNORE THE NEXT 9 CARDS.
VARIABLE: AII (1,1,1)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 48
VARIABLE: THATAD (1)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 47
INSTRUCTION: IF NUMCMG OR IDOF (1) = 0, IGNORE THE NEXT 2 CARDS.
VARIABLE: THATA (1)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 57
VARIABLE: AIO (1,3,3)

CARD: 56
VARIABLE: AIO (1,3,2)

CARD: 55
VARIABLE: AIO (1,3,1)

CARD: 54
VARIABLE: AIO (1,2,3)

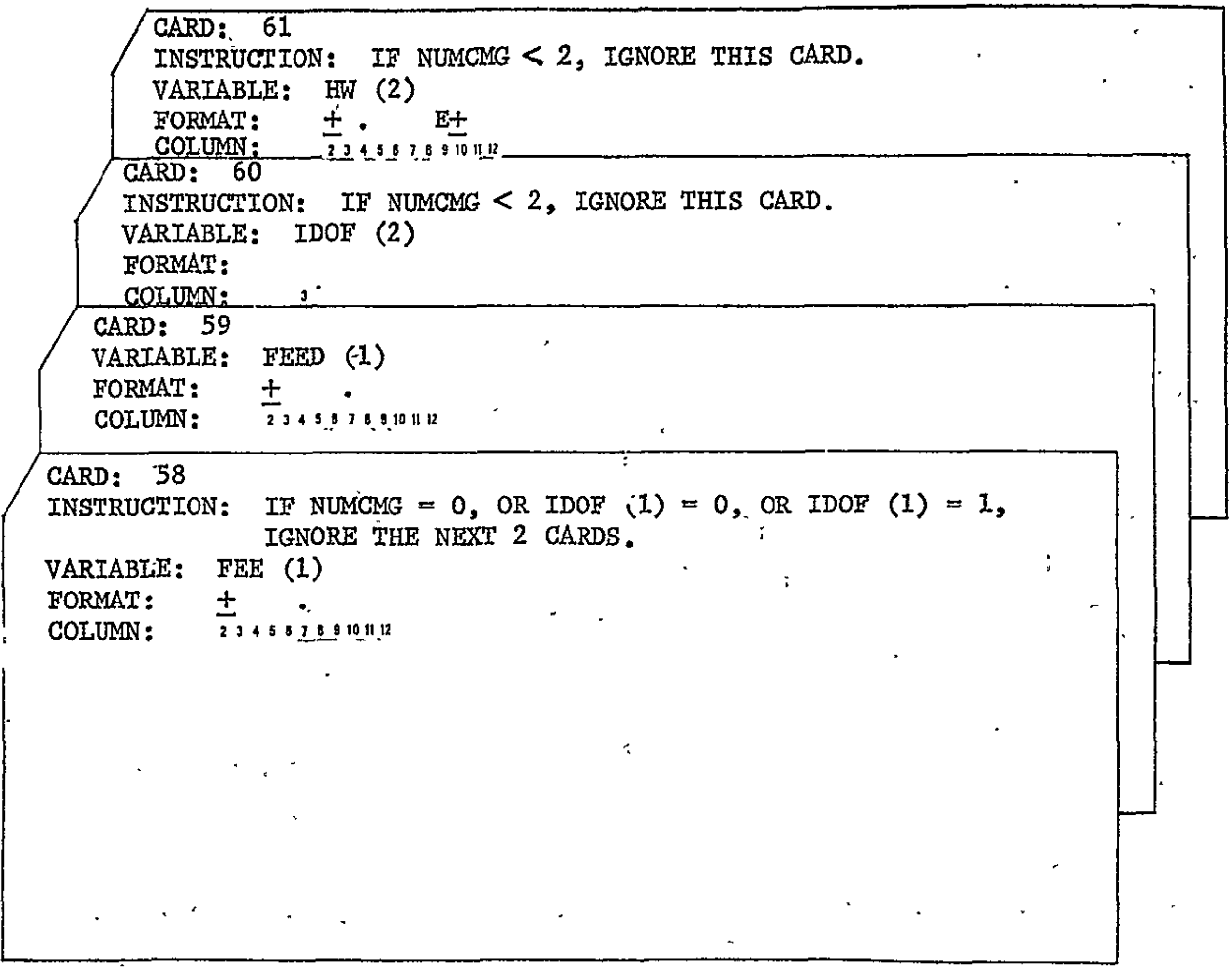
CARD: 53
VARIABLE: AIO (1,2,2)

CARD: 52
VARIABLE: AIO (1,2,1)

CARD: 51
VARIABLE: AIO (1,1,3)

CARD: 50
VARIABLE: AIO (1,1,2)

```
CARD:  49
INSTRUCTION:  IF NUMCMG = 0, OR IDOF (1) = 0, OR IDOF (1) = 1,
              IGNORE THE NEXT 9 CARDS.
VARIABLE:  AIO (1,1,1)
FORMAT:    ±    .
COLUMN:    2 3 4 5 6 7 8 9 10 11 12
```

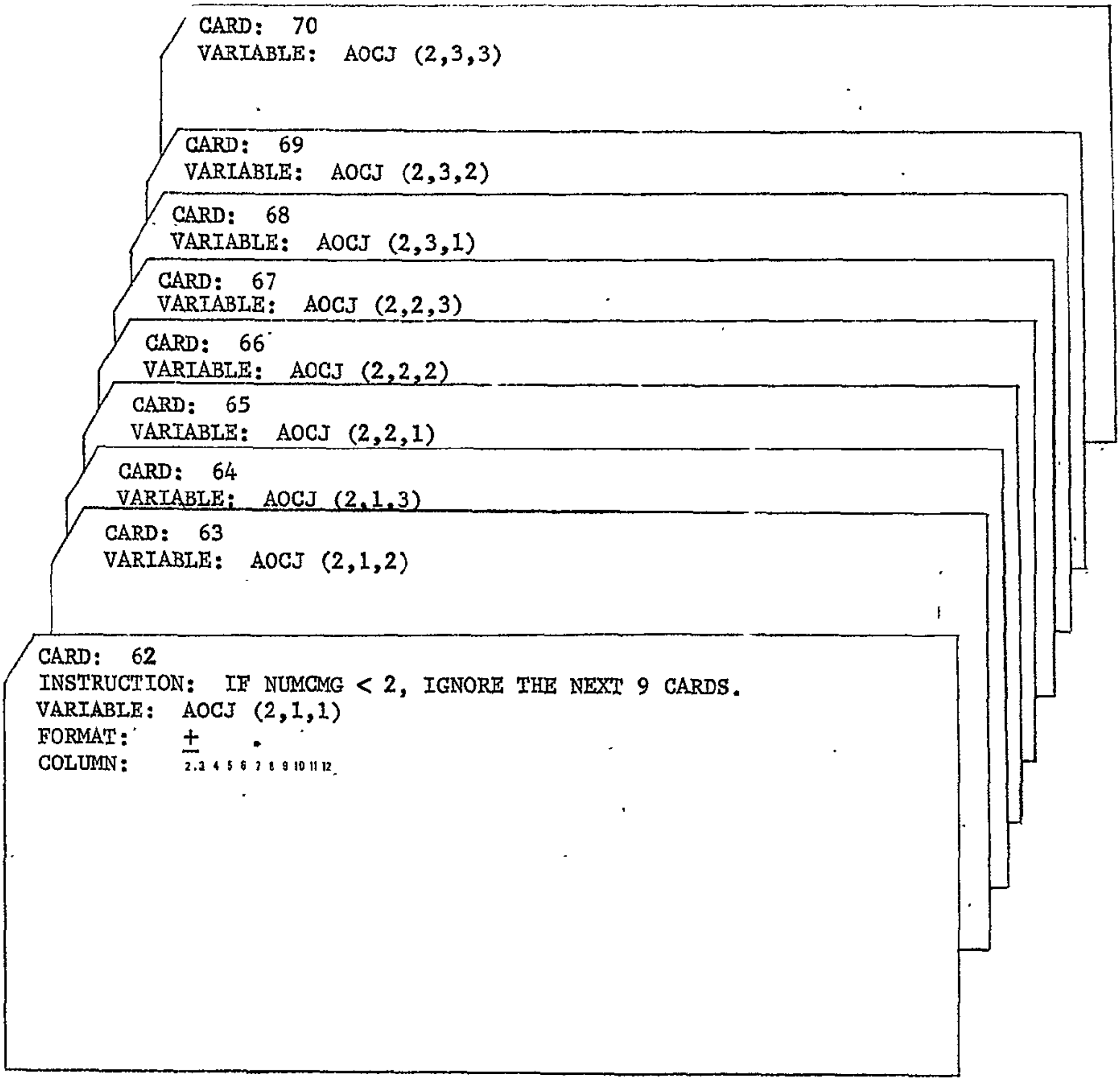


CARD: 61
INSTRUCTION: IF NUMCMG < 2, IGNORE THIS CARD.
VARIABLE: HW (2)
FORMAT: ± . E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 60
INSTRUCTION: IF NUMCMG < 2, IGNORE THIS CARD.
VARIABLE: IDOF (2)
FORMAT:
COLUMN: 3

CARD: 59
VARIABLE: FEED (-1)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 58
INSTRUCTION: IF NUMCMG = 0, OR IDOF (1) = 0, OR IDOF (1) = 1,
IGNORE THE NEXT 2 CARDS.
VARIABLE: FEE (1)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12



CARD: 70
VARIABLE: AOCJ (2,3,3)

CARD: 69
VARIABLE: AOCJ (2,3,2)

CARD: 68
VARIABLE: AOCJ (2,3,1)

CARD: 67
VARIABLE: AOCJ (2,2,3)

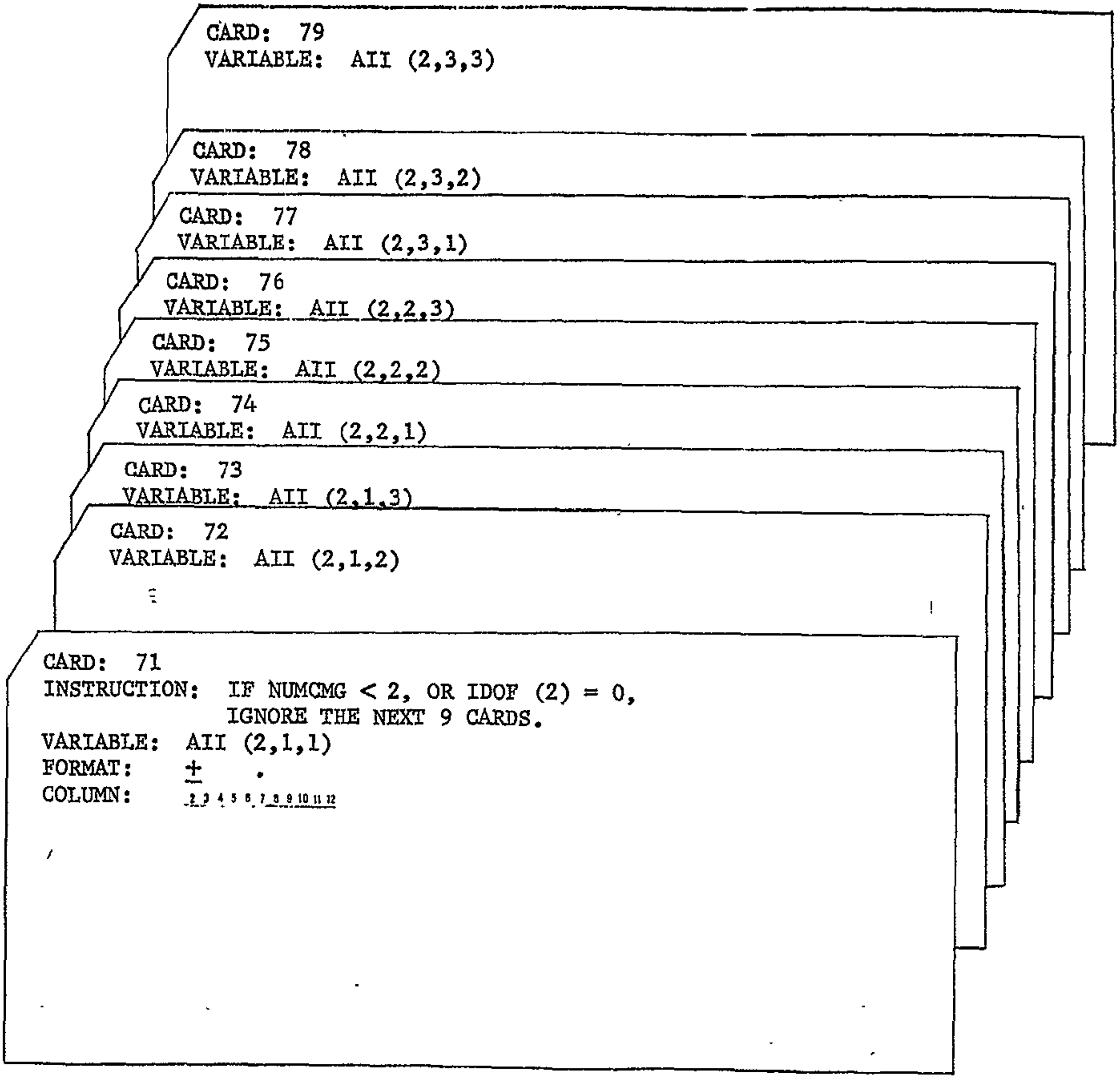
CARD: 66
VARIABLE: AOCJ (2,2,2)

CARD: 65
VARIABLE: AOCJ (2,2,1)

CARD: 64
VARIABLE: AOCJ (2,1,3)

CARD: 63
VARIABLE: AOCJ (2,1,2)

CARD: 62
INSTRUCTION: IF NUMCMG < 2, IGNORE THE NEXT 9 CARDS.
VARIABLE: AOCJ (2,1,1)
FORMAT: ± .
COLUMN: 2.3 4 5 6 7 8 9 10 11 12



CARD: 79
VARIABLE: AII (2,3,3)

CARD: 78
VARIABLE: AII (2,3,2)

CARD: 77
VARIABLE: AII (2,3,1)

CARD: 76
VARIABLE: AII (2,2,3)

CARD: 75
VARIABLE: AII (2,2,2)

CARD: 74
VARIABLE: AII (2,2,1)

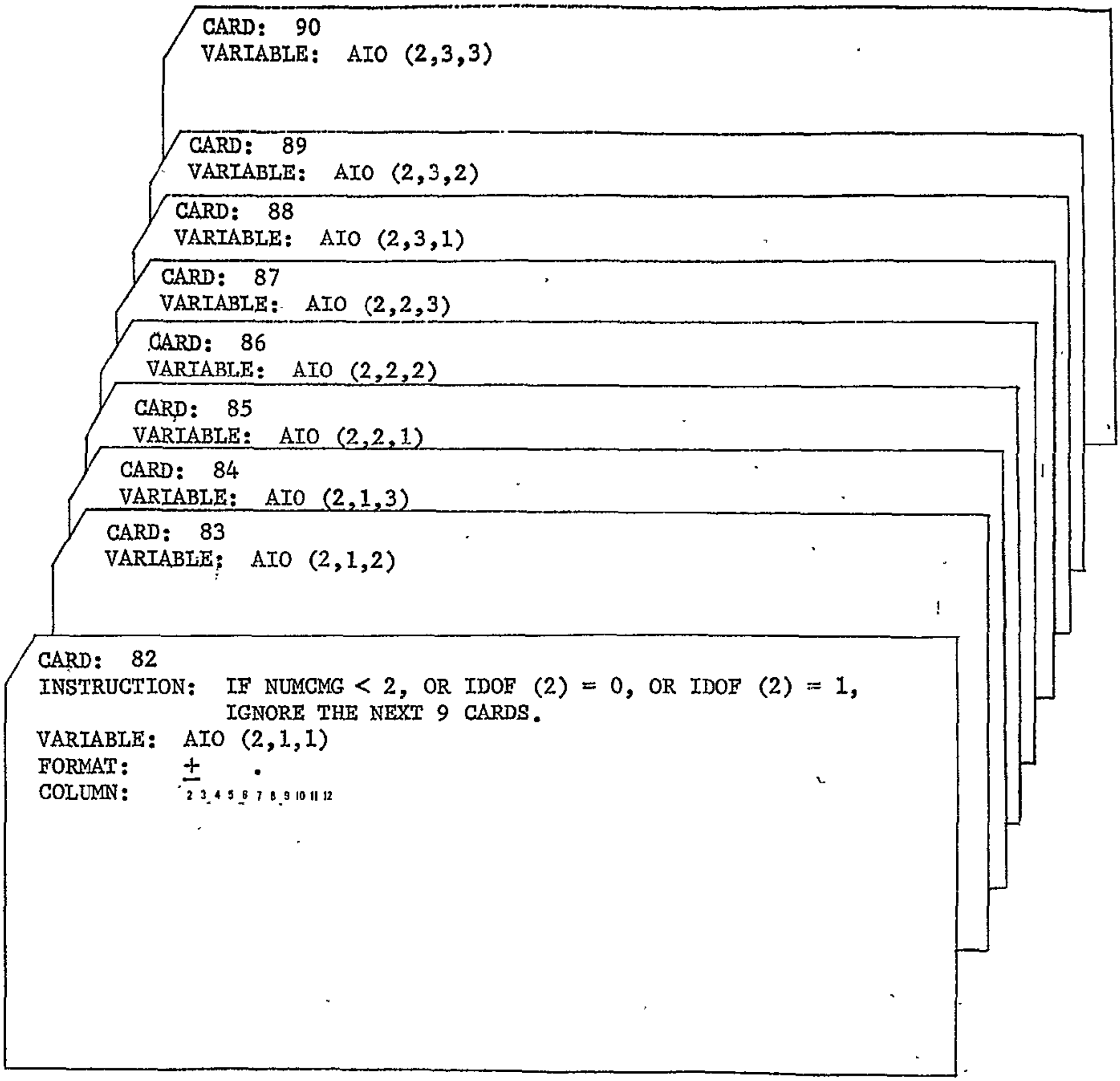
CARD: 73
VARIABLE: AII (2,1,3)

CARD: 72
VARIABLE: AII (2,1,2)

CARD: 71
INSTRUCTION: IF NUMCMG < 2, OR IDOF (2) = 0,
IGNORE THE NEXT 9 CARDS.
VARIABLE: AII (2,1,1)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 81
INSTRUCTION: IF NUMCMG < 2, OR IDOF (2) = 0, IGNORE THIS CARD.
VARIABLE: THATAD (2)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 80
INSTRUCTION: IF NUMCMG < 2, OR IDOF (2) = 0, IGNORE THIS CARD.
VARIABLE: THATA (2)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12



CARD: 90
VARIABLE: AIO (2,3,3)

CARD: 89
VARIABLE: AIO (2,3,2)

CARD: 88
VARIABLE: AIO (2,3,1)

CARD: 87
VARIABLE: AIO (2,2,3)

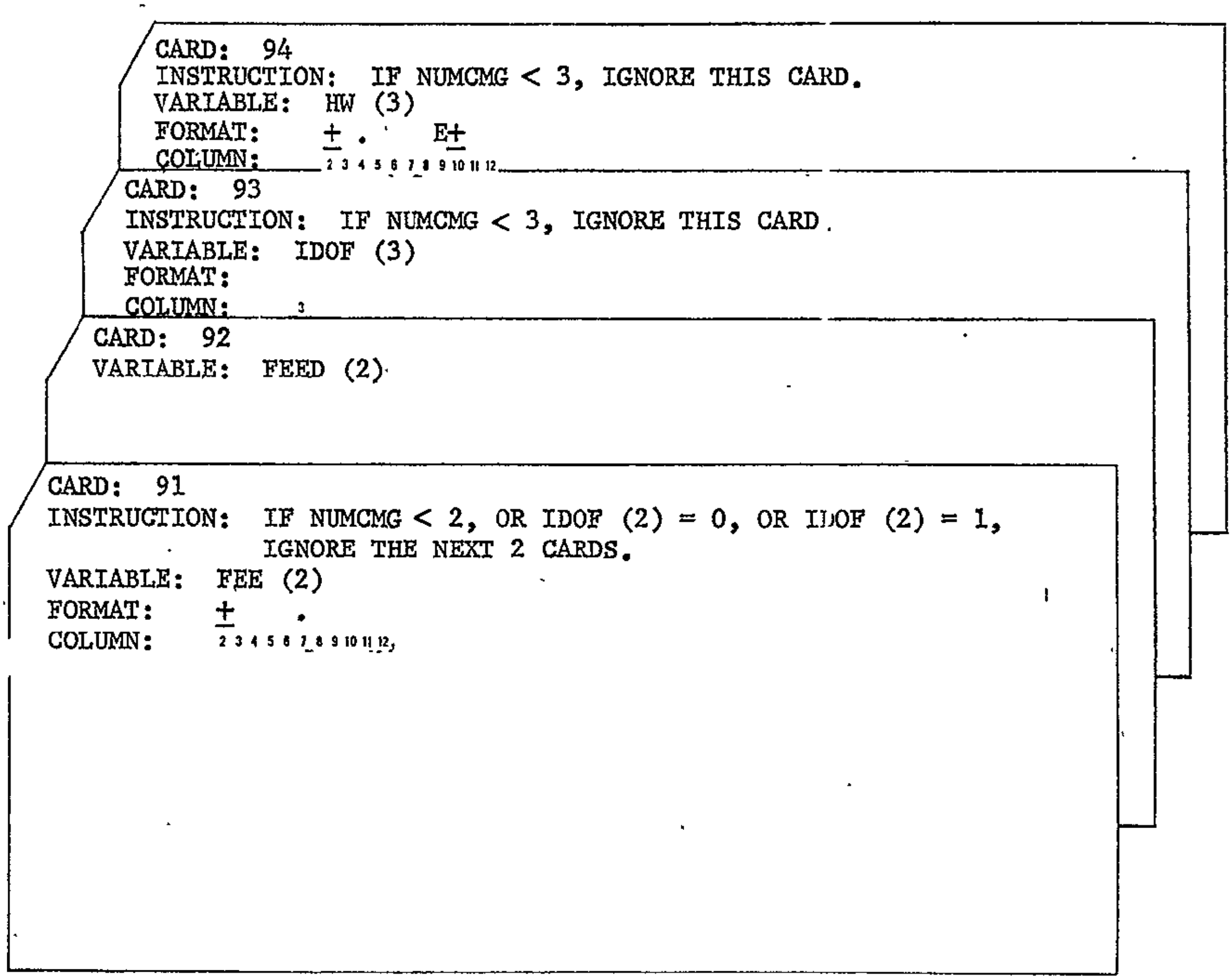
CARD: 86
VARIABLE: AIO (2,2,2)

CARD: 85
VARIABLE: AIO (2,2,1)

CARD: 84
VARIABLE: AIO (2,1,3)

CARD: 83
VARIABLE: AIO (2,1,2)

CARD: 82
INSTRUCTION: IF NUMCMG < 2, OR IDOF (2) = 0, OR IDOF (2) = 1,
IGNORE THE NEXT 9 CARDS.
VARIABLE: AIO (2,1,1)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

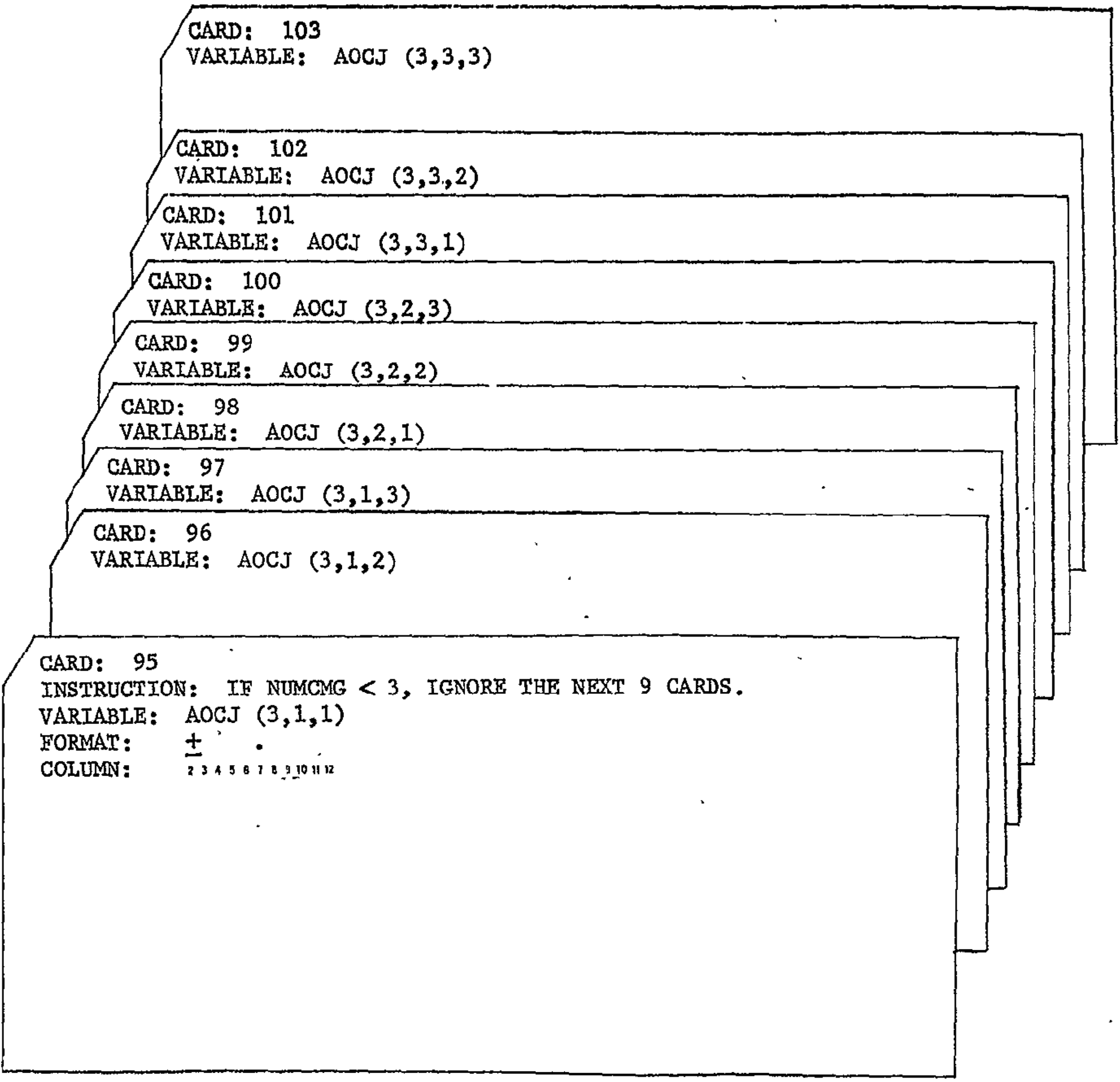


CARD: 94
INSTRUCTION: IF NUMCMG < 3, IGNORE THIS CARD.
VARIABLE: HW (3)
FORMAT: ± . E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 93
INSTRUCTION: IF NUMCMG < 3, IGNORE THIS CARD.
VARIABLE: IDOF (3)
FORMAT:
COLUMN: 3

CARD: 92
VARIABLE: FEED (2)

CARD: 91
INSTRUCTION: IF NUMCMG < 2, OR IDOF (2) = 0, OR IDOF (2) = 1,
IGNORE THE NEXT 2 CARDS.
VARIABLE: FEE (2)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12



CARD: 103
VARIABLE: AOCJ (3,3,3)

CARD: 102
VARIABLE: AOCJ (3,3,2)

CARD: 101
VARIABLE: AOCJ (3,3,1)

CARD: 100
VARIABLE: AOCJ (3,2,3)

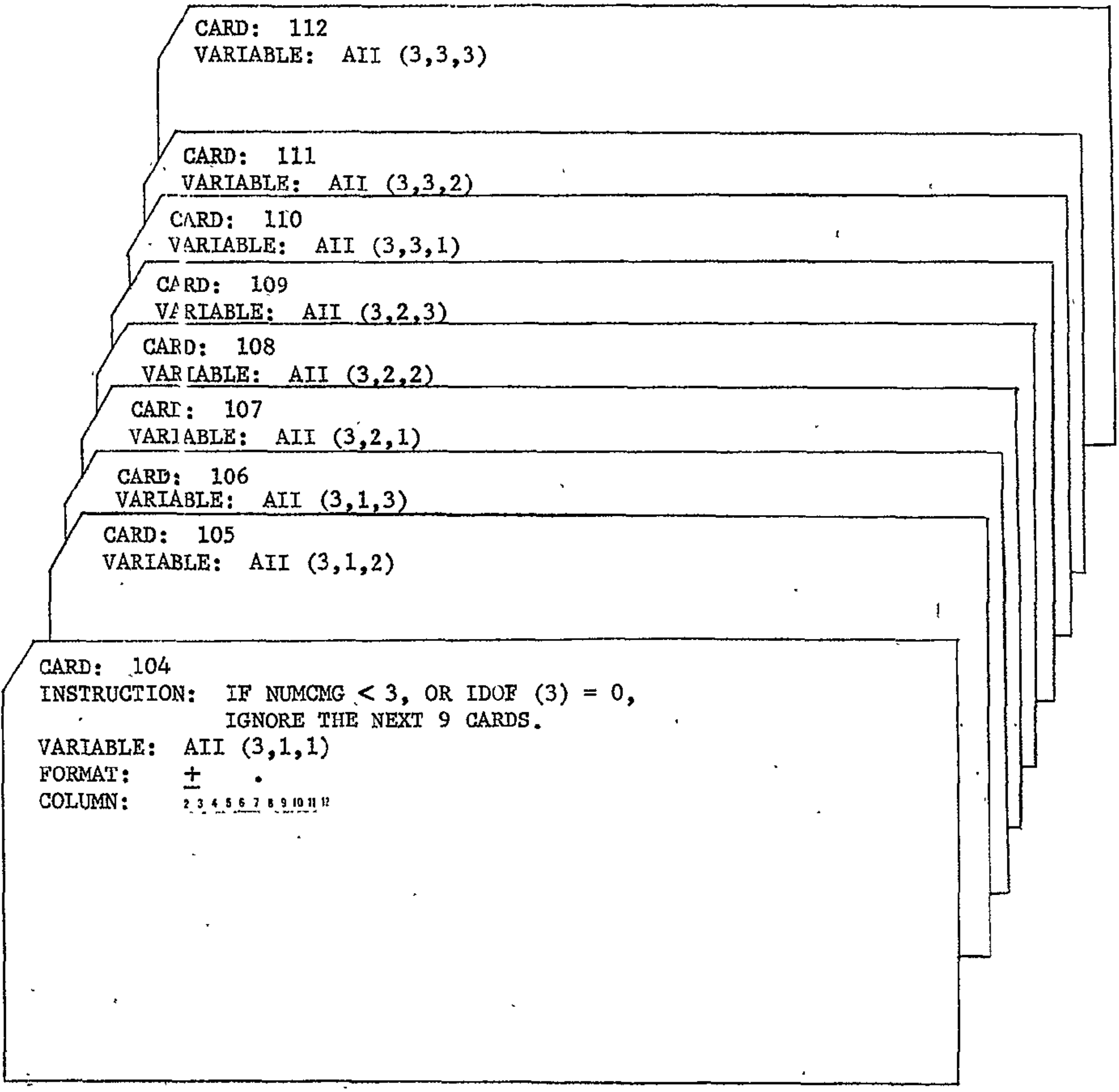
CARD: 99
VARIABLE: AOCJ (3,2,2)

CARD: 98
VARIABLE: AOCJ (3,2,1)

CARD: 97
VARIABLE: AOCJ (3,1,3)

CARD: 96
VARIABLE: AOCJ (3,1,2)

CARD: 95
INSTRUCTION: IF NUMCMG < 3, IGNORE THE NEXT 9 CARDS.
VARIABLE: AOCJ (3,1,1)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12



CARD: 112
VARIABLE: AII (3,3,3)

CARD: 111
VARIABLE: AII (3,3,2)

CARD: 110
VARIABLE: AII (3,3,1)

CARD: 109
VARIABLE: AII (3,2,3)

CARD: 108
VARIABLE: AII (3,2,2)

CARD: 107
VARIABLE: AII (3,2,1)

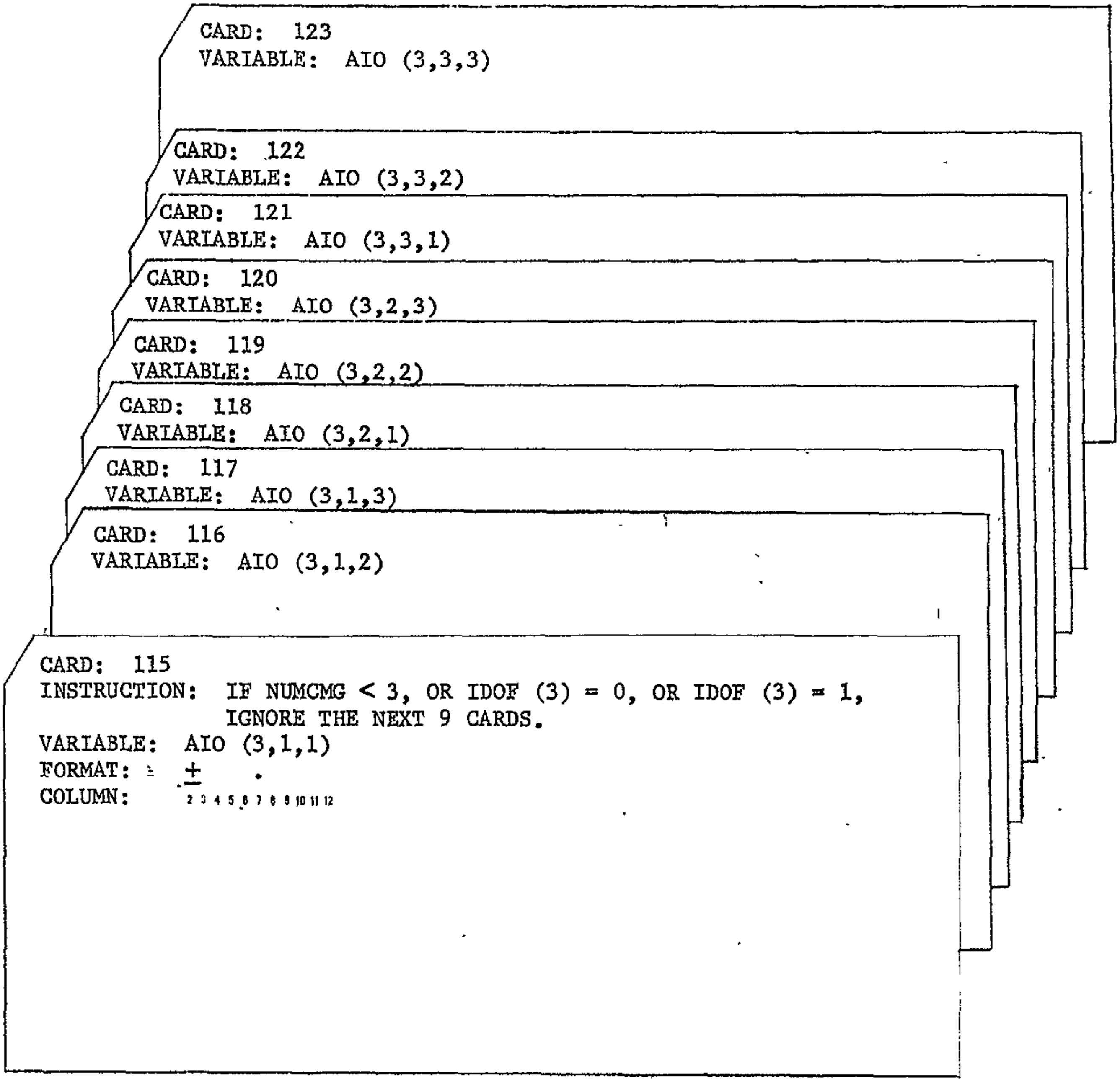
CARD: 106
VARIABLE: AII (3,1,3)

CARD: 105
VARIABLE: AII (3,1,2)

CARD: 104
INSTRUCTION: IF NUMCMG < 3, OR IDOF (3) = 0,
IGNORE THE NEXT 9 CARDS.
VARIABLE: AII (3,1,1)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 114
INSTRUCTION: IF NUMCMG < 3, OR IDOF (3) = 0, IGNORE THIS CARD.
VARIABLE: THATAD (3)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 113
INSTRUCTION: IF NUMCMG < 3, OR IDOF (3) = 0, IGNORE THIS CARD.
VARIABLE: THATA (3)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12



CARD: 123
VARIABLE: AIO (3,3,3)

CARD: 122
VARIABLE: AIO (3,3,2)

CARD: 121
VARIABLE: AIO (3,3,1)

CARD: 120
VARIABLE: AIO (3,2,3)

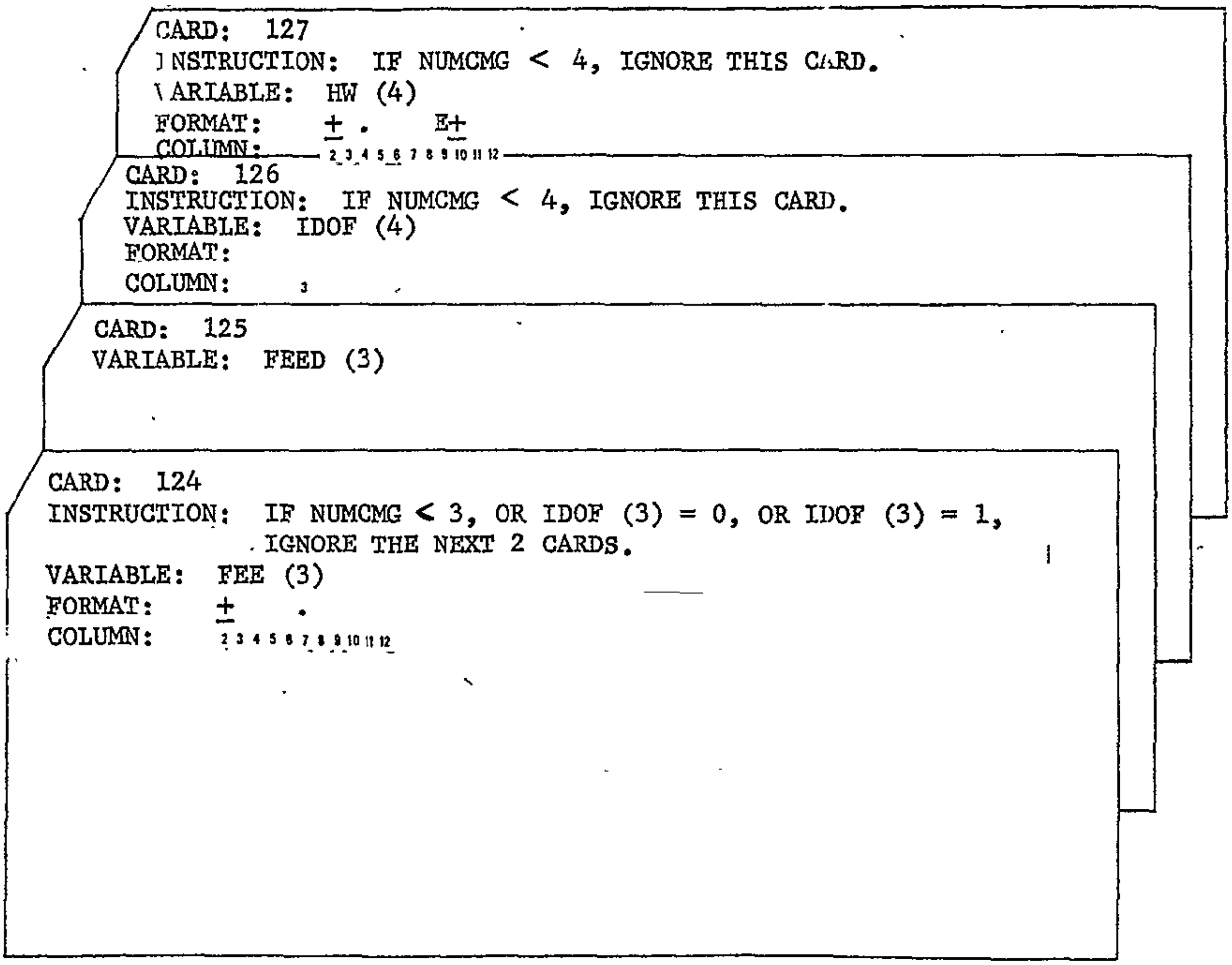
CARD: 119
VARIABLE: AIO (3,2,2)

CARD: 118
VARIABLE: AIO (3,2,1)

CARD: 117
VARIABLE: AIO (3,1,3)

CARD: 116
VARIABLE: AIO (3,1,2)

CARD: 115
INSTRUCTION: IF NUMCMG < 3, OR IDOF (3) = 0, OR IDOF (3) = 1,
IGNORE THE NEXT 9 CARDS.
VARIABLE: AIO (3,1,1)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12



CARD: 127
INSTRUCTION: IF NUMCMG < 4, IGNORE THIS CARD.
VARIABLE: HW (4)
FORMAT: ± . E+
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 126
INSTRUCTION: IF NUMCMG < 4, IGNORE THIS CARD.
VARIABLE: IDOF (4)
FORMAT:
COLUMN: 3

CARD: 125
VARIABLE: FEED (3)

CARD: 124
INSTRUCTION: IF NUMCMG < 3, OR IDOF (3) = 0, OR IDOF (3) = 1,
IGNORE THE NEXT 2 CARDS.
VARIABLE: FEE (3)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 136
VARIABLE: AOCJ (4,3,3)

CARD: 135
VARIABLE: AOCJ (4,3,2)

CARD: 134
VARIABLE: AOCJ (4,3,1)

CARD: 133
VARIABLE: AOCJ (4,2,3)

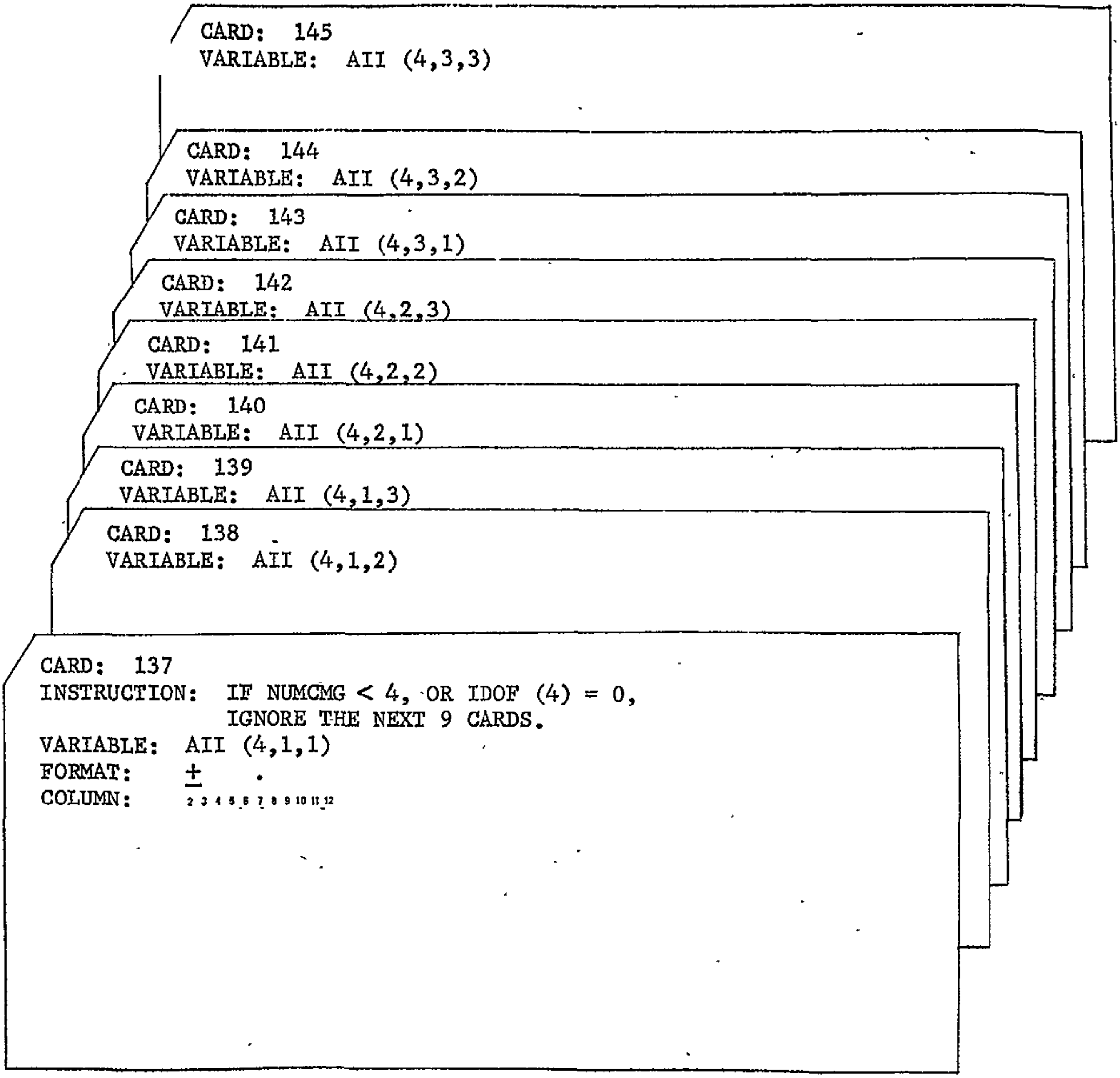
CARD: 132
VARIABLE: AOCJ (4,2,2)

CARD: 131
VARIABLE: AOCJ (4,2,1)

CARD: 130
VARIABLE: AOCJ (4,1,3)

CARD: 129
VARIABLE: AOCJ (4,1,2)

```
CARD:  128
INSTRUCTION:  IF NUMCMG < 4, IGNORE THE NEXT 9 CARDS.
VARIABLE:  AOCJ (4,1,1)
FORMAT:       ±       .
COLUMN:       2 3 4 5 6 7 8 9 10 11 12
```



CARD: 145
VARIABLE: AII (4,3,3)

CARD: 144
VARIABLE: AII (4,3,2)

CARD: 143
VARIABLE: AII (4,3,1)

CARD: 142
VARIABLE: AII (4,2,3)

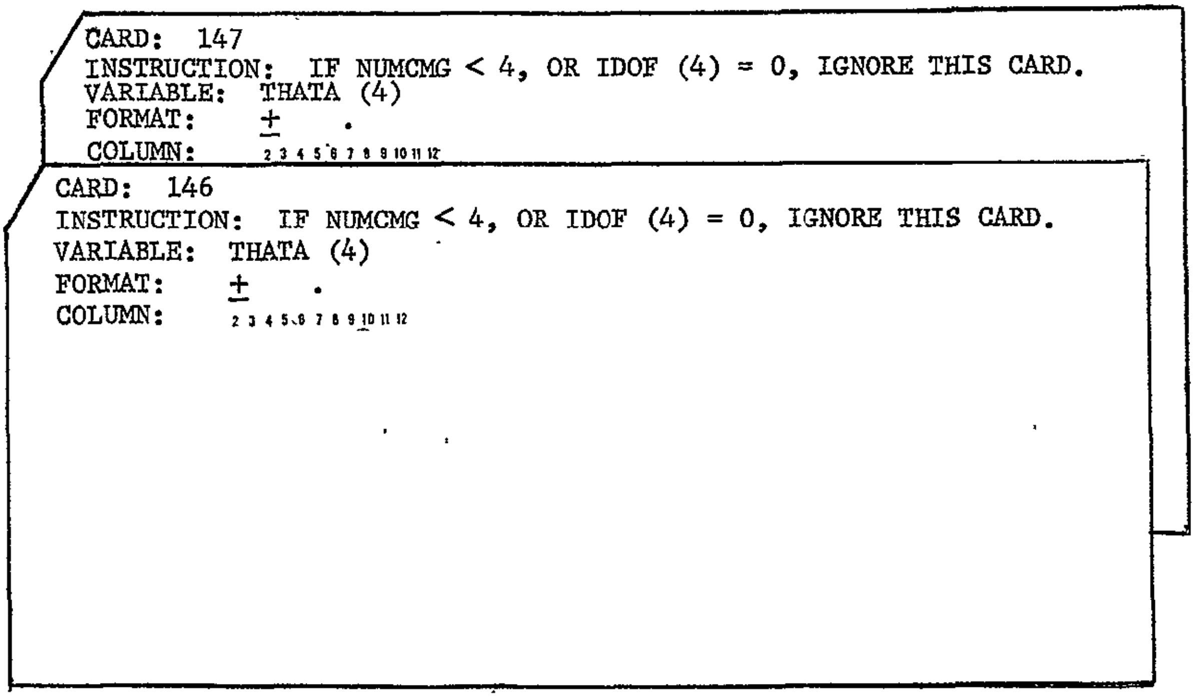
CARD: 141
VARIABLE: AII (4,2,2)

CARD: 140
VARIABLE: AII (4,2,1)

CARD: 139
VARIABLE: AII (4,1,3)

CARD: 138
VARIABLE: AII (4,1,2)

CARD: 137
INSTRUCTION: IF NUMCMG < 4, OR IDOF (4) = 0,
IGNORE THE NEXT 9 CARDS.
VARIABLE: AII (4,1,1)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12



CARD: 147

INSTRUCTION: IF NUMCMG < 4, OR IDOF (4) = 0, IGNORE THIS CARD.

VARIABLE: THATA (4)

```
FORMAT:     ±   .
COLUMN:     2 3 4 5 6 7 8 9 10 11 12
```

CARD: 146

INSTRUCTION: IF NUMCMG < 4, OR IDOF (4) = 0, IGNORE THIS CARD.

VARIABLE: THATA (4)

```
FORMAT:     ±   .
COLUMN:     2 3 4 5 6 7 8 9 10 11 12
```

```
CARD:  148
INSTRUCTION:  IF NUMCMG < 4, OR IDOF (4) = 0, OR IDOF (4) = 1,
              IGNORE THE NEXT 9 CARDS.
VARIABLE:  AIO (4,1,1)
FORMAT:    ±     .
COLUMN:    2 3 4 5 6 7 8 9 10 11 12
```

```
CARD:  149
VARIABLE:  AIO (4,1,2)
```

```
CARD:  150
VARIABLE:  AIO (4,1,3)
```

```
CARD:  151
VARIABLE:  AIO (4,2,1)
```

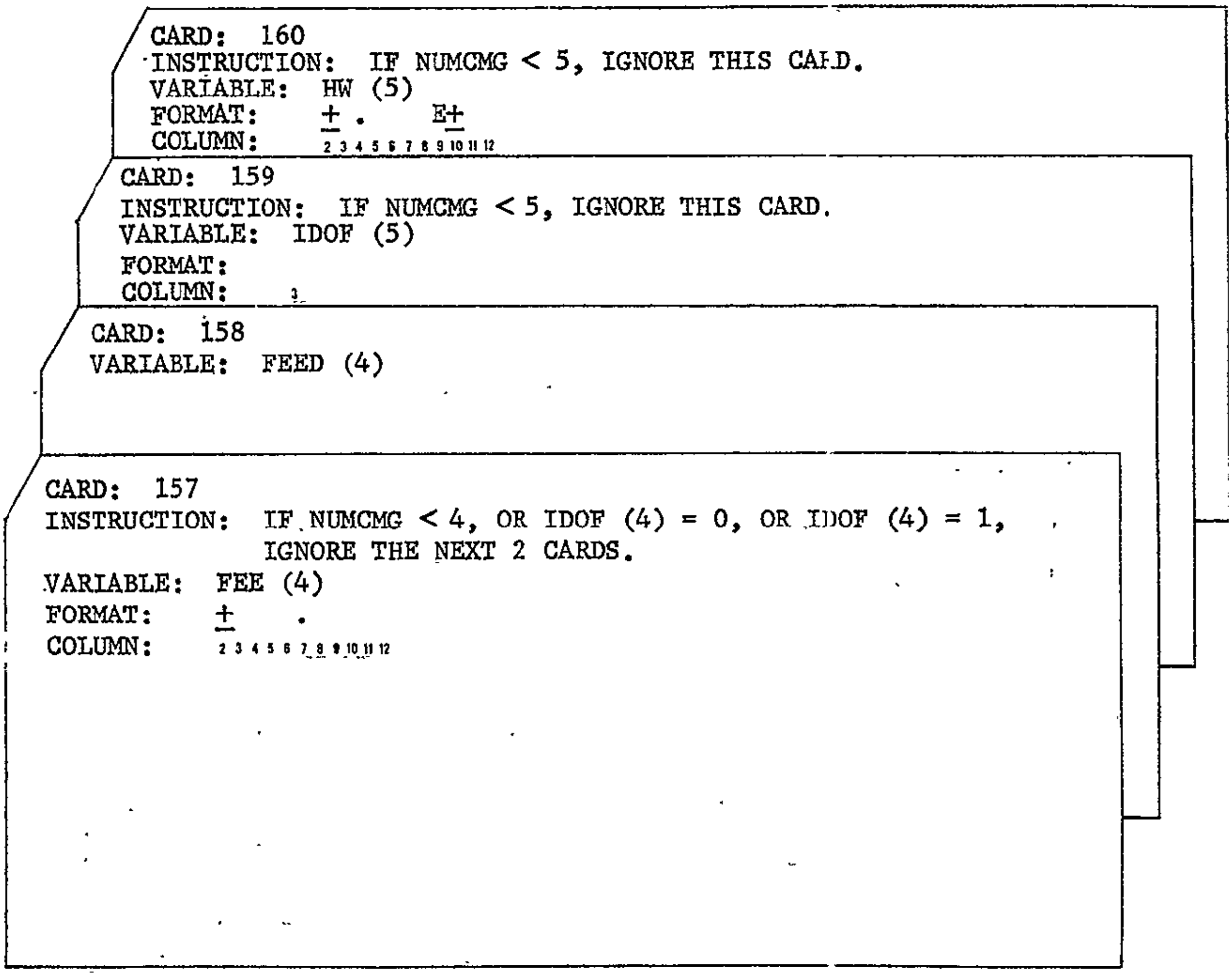
```
CARD:  152
VARIABLE:  AIO (4,2,2)
```

```
CARD:  153
VARIABLE:  AIO (4,2,3)
```

```
CARD:  154
VARIABLE:  AIO (4,3,1)
```

```
CARD:  155
VARIABLE:  AIO (4,3,2)
```

```
CARD:  156
VARIABLE:  AIO (4,3,3)
```

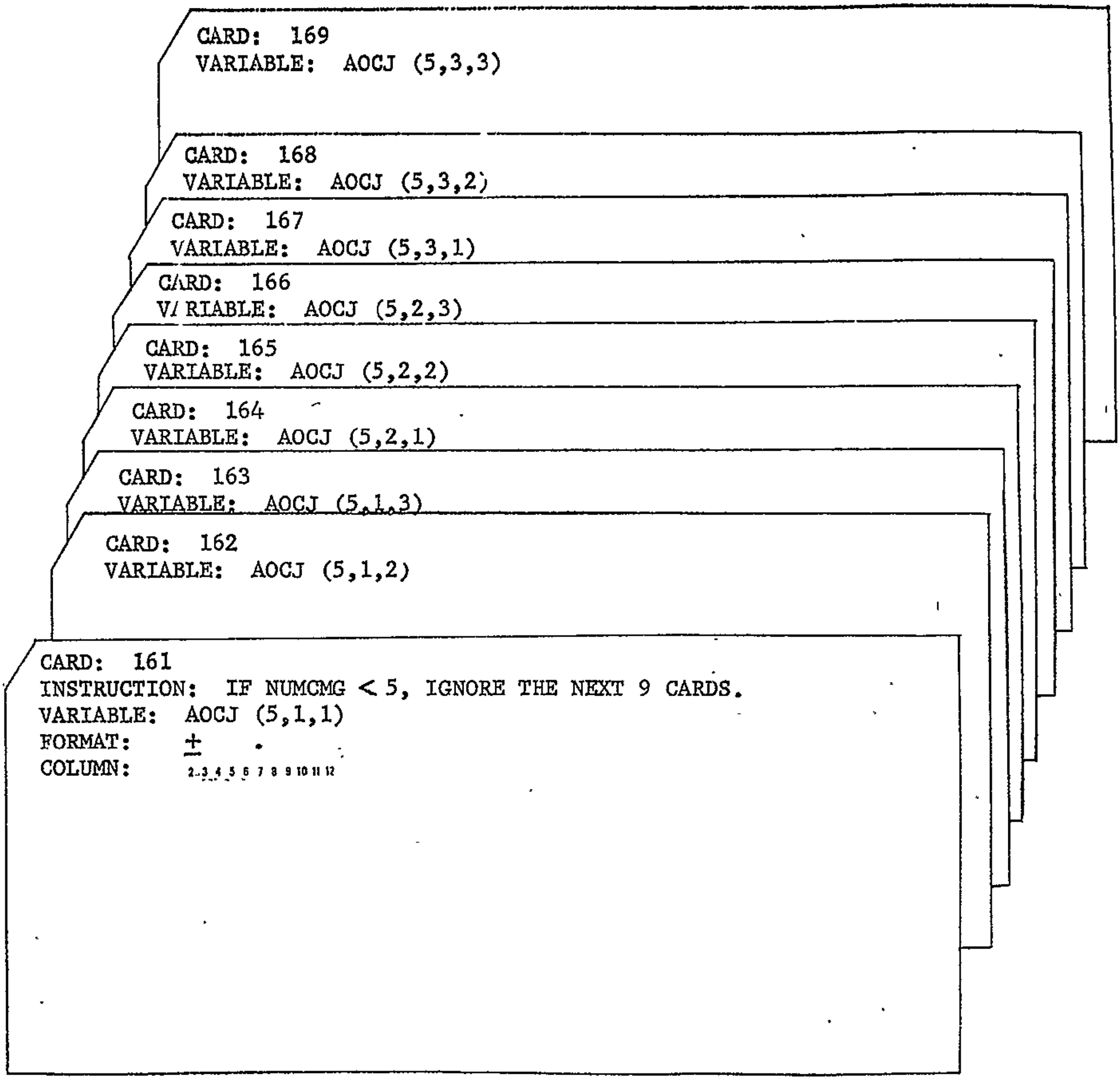


CARD: 160
INSTRUCTION: IF NUMCMG < 5, IGNORE THIS CAFD.
VARIABLE: HW (5)
FORMAT: ± . E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 159
INSTRUCTION: IF NUMCMG < 5, IGNORE THIS CARD.
VARIABLE: IDOF (5)
FORMAT:
COLUMN: 3

CARD: 158
VARIABLE: FEED (4)

CARD: 157
INSTRUCTION: IF NUMCMG < 4, OR IDOF (4) = 0, OR IDOF (4) = 1,
IGNORE THE NEXT 2 CARDS.
VARIABLE: FEE (4)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12



CARD: 169
VARIABLE: AOCJ (5,3,3)

CARD: 168
VARIABLE: AOCJ (5,3,2)

CARD: 167
VARIABLE: AOCJ (5,3,1)

CARD: 166
VARIABLE: AOCJ (5,2,3)

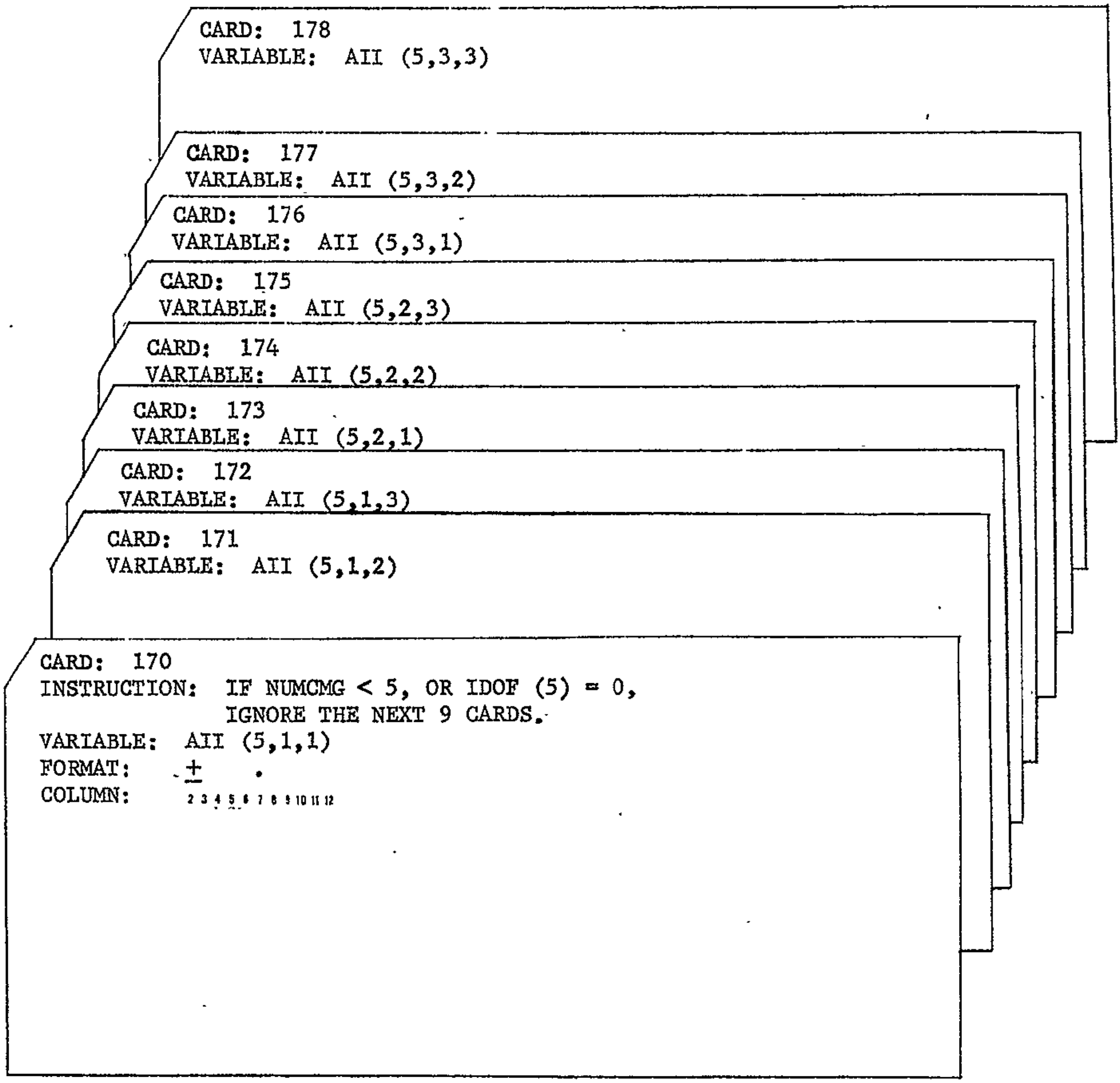
CARD: 165
VARIABLE: AOCJ (5,2,2)

CARD: 164
VARIABLE: AOCJ (5,2,1)

CARD: 163
VARIABLE: AOCJ (5,1,3)

CARD: 162
VARIABLE: AOCJ (5,1,2)

CARD: 161
INSTRUCTION: IF NUMCMG < 5, IGNORE THE NEXT 9 CARDS.
VARIABLE: AOCJ (5,1,1)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12



CARD: 178
VARIABLE: AII (5,3,3)

CARD: 177
VARIABLE: AII (5,3,2)

CARD: 176
VARIABLE: AII (5,3,1)

CARD: 175
VARIABLE: AII (5,2,3)

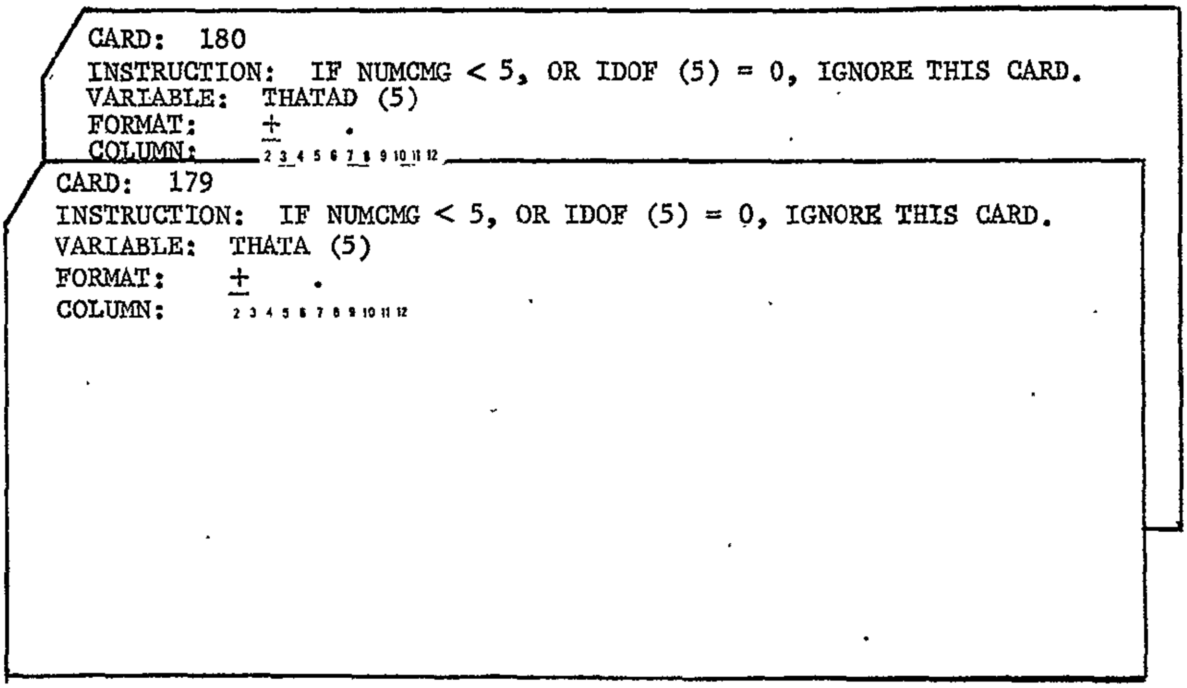
CARD: 174
VARIABLE: AII (5,2,2)

CARD: 173
VARIABLE: AII (5,2,1)

CARD: 172
VARIABLE: AII (5,1,3)

CARD: 171
VARIABLE: AII (5,1,2)

CARD: 170
INSTRUCTION: IF NUMCMG < 5, OR IDOF (5) = 0,
IGNORE THE NEXT 9 CARDS.
VARIABLE: AII (5,1,1)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12



CARD: 180
INSTRUCTION: IF NUMCMG < 5, OR IDOF (5) = 0, IGNORE THIS CARD.
VARIABLE: THATAD (5)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 179
INSTRUCTION: IF NUMCMG < 5, OR IDOF (5) = 0, IGNORE THIS CARD.
VARIABLE: THATA (5)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 189
VARIABLE: AIO (5,3,3)

CARD: 188
VARIABLE: AIO (5,3,2)

CARD: 187
VARIABLE: AIO (5,3,1)

CARD: 186
VARIABLE: AIO (5,2,3)

CARD: 185
VARIABLE: AIO (5,2,2)

CARD: 184
VARIABLE: AIO (5,2,1)

CARD: 183
VARIABLE: AIO (5,1,3)

CARD: 182
VARIABLE: AIO (5,1,2)

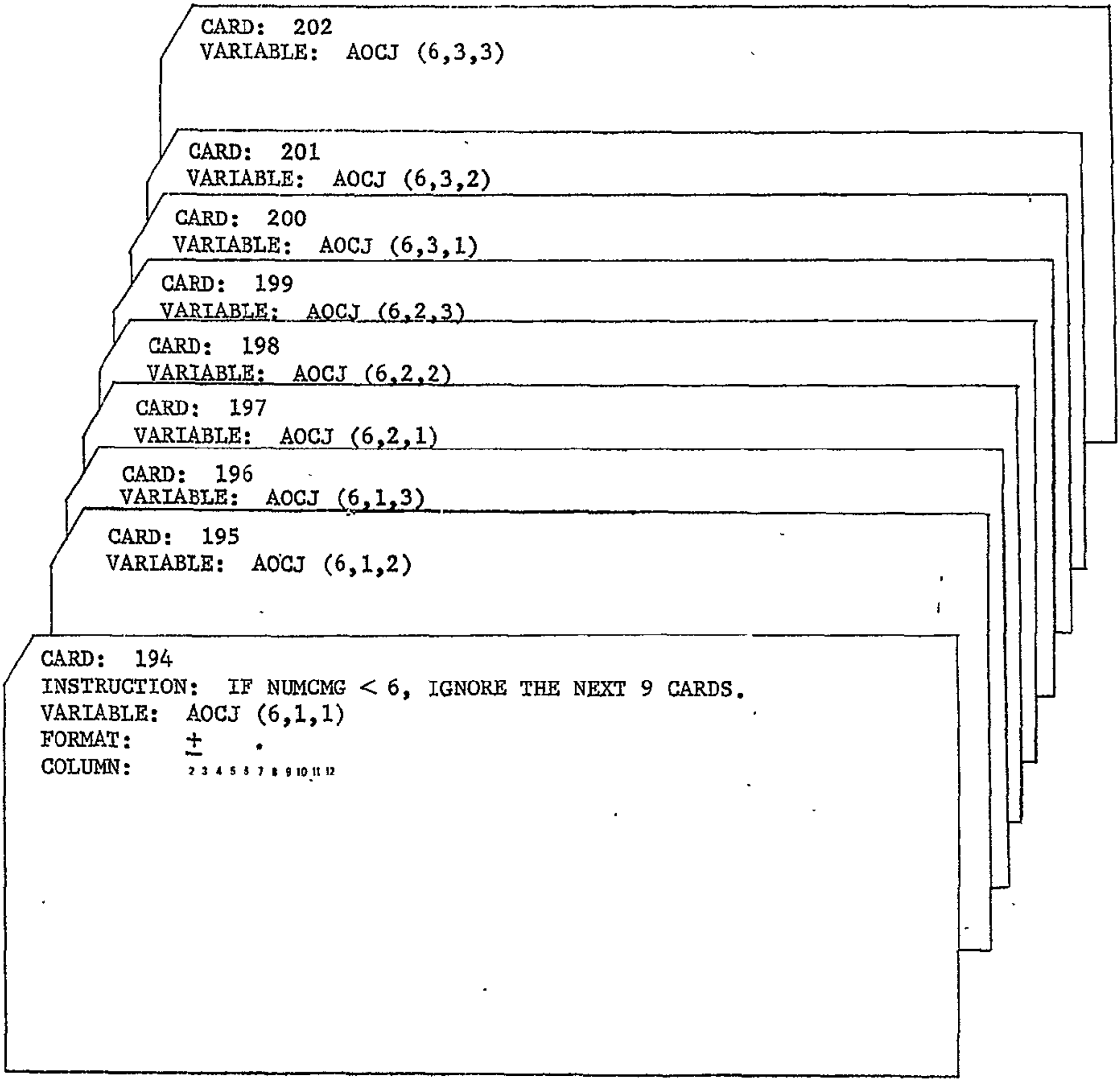
```
CARD:  181
INSTRUCTION:  IF NUMCMG < 5, OR IDOF (5) = 0, OR IDOF (5) = 1,
              IGNORE THE NEXT 9 CARDS.
VARIABLE:  AIO (5,1,1)
FORMAT:    ±    .
COLUMN:    2 3 4 5 6 7 8 9 10 11 12
```

```
CARD:  193
INSTRUCTION:  IF NUMCMG < 6, IGNORE THIS CARD.
VARIABLE:  HW (6)
FORMAT:    ± .       E±
COLUMN:    2 3 4 5 6 7 8 9 10 11 12
```

```
CARD:  192
INSTRUCTION:  IF NUMCMG < 6, IGNORE THIS CARD.
VARIABLE:  IDOF (6)
FORMAT:
COLUMN:    3
```

```
CARD:  191
VARIABLE:  FEED (5)
```

```
CARD:  190
INSTRUCTION:  IF NUMCMG < 5, OR IDOF (5) = 0, OR IDOF (5) = 1
              IGNORE THE NEXT 2 CARDS.
VARIABLE:  FEE (5)
FORMAT:    ±     .
COLUMN:    2 3 4 5 6 7 8 9 10 11 12
```



CARD: 202
VARIABLE: AOCJ (6,3,3)

CARD: 201
VARIABLE: AOCJ (6,3,2)

CARD: 200
VARIABLE: AOCJ (6,3,1)

CARD: 199
VARIABLE: AOCJ (6,2,3)

CARD: 198
VARIABLE: AOCJ (6,2,2)

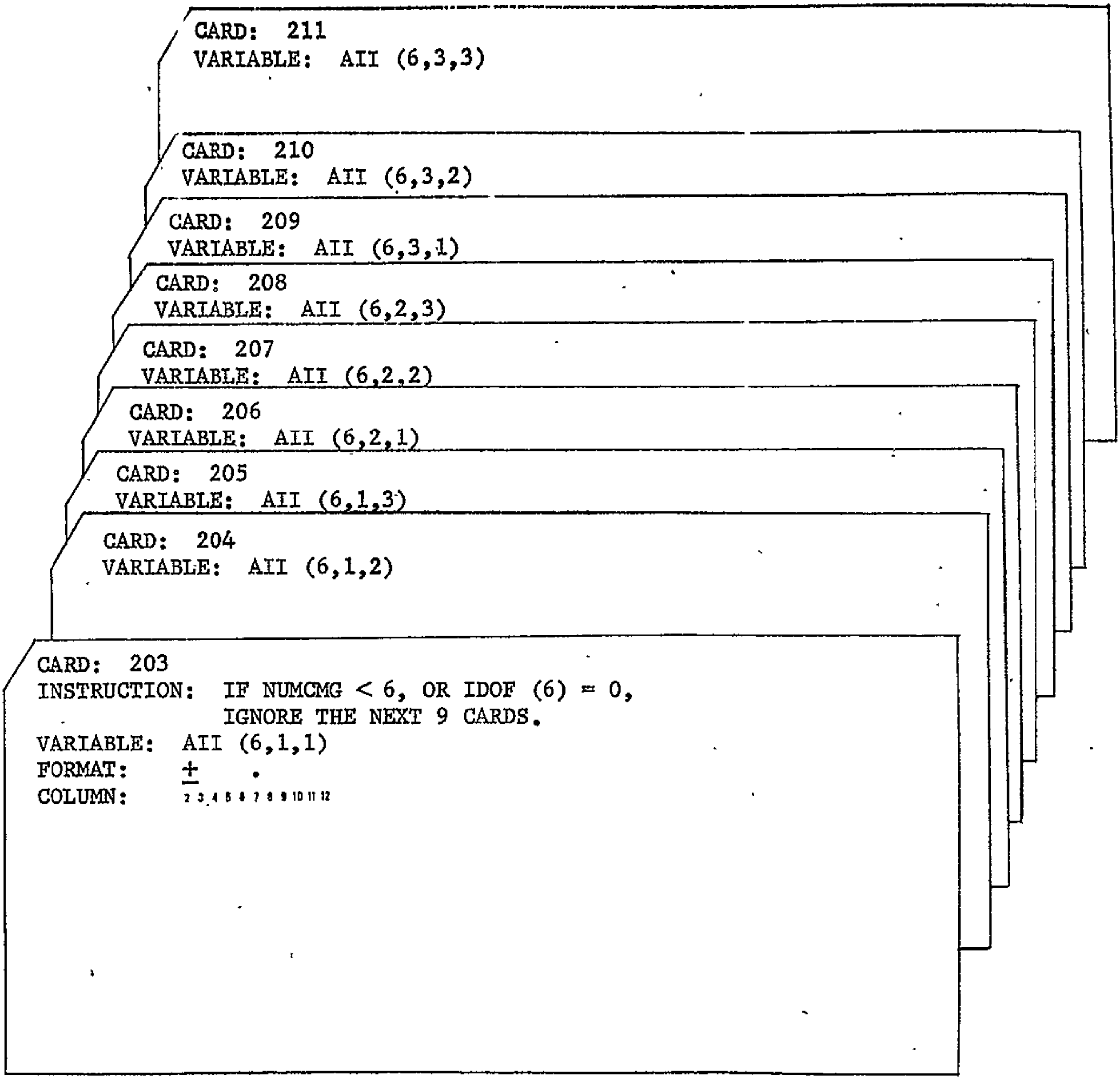
CARD: 197
VARIABLE: AOCJ (6,2,1)

CARD: 196
VARIABLE: AOCJ (6,1,3)

CARD: 195
VARIABLE: AOCJ (6,1,2)

CARD: 194
INSTRUCTION: IF NUMCMG < 6, IGNORE THE NEXT 9 CARDS.
VARIABLE: AOCJ (6,1,1)

```
FORMAT:    ±        .
COLUMN:    2 3 4 5 6 7 8 9 10 11 12
```



CARD: 211
VARIABLE: AII (6,3,3)

CARD: 210
VARIABLE: AII (6,3,2)

CARD: 209
VARIABLE: AII (6,3,1)

CARD: 208
VARIABLE: AII (6,2,3)

CARD: 207
VARIABLE: AII (6,2,2)

CARD: 206
VARIABLE: AII (6,2,1)

CARD: 205
VARIABLE: AII (6,1,3)

CARD: 204
VARIABLE: AII (6,1,2)

CARD: 203
INSTRUCTION: IF NUMCMG < 6, OR IDOF (6) = 0,
IGNORE THE NEXT 9 CARDS.
VARIABLE: AII (6,1,1)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 213

INSTRUCTION: IF NUMCMG < 6, OR IDOF (6) = 0, IGNORE THIS CARD.

VARIABLE: THATAD (6)

FORMAT: ± .

COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 212

INSTRUCTION: IF NUMCMG < 6, OR IDOF (6) = 0, IGNORE THIS CARD.

VARIABLE: THATA (6)

FORMAT: ± .

COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 222
VARIABLE: AIO (6,3,3)

CARD: 221
VARIABLE: AIO (6,3,2)

CARD: 220
VARIABLE: AIO (6,3,1)

CARD: 219
VARIABLE: AIO (6,2,3)

CARD: 218
VARIABLE: AIO (6,2,2)

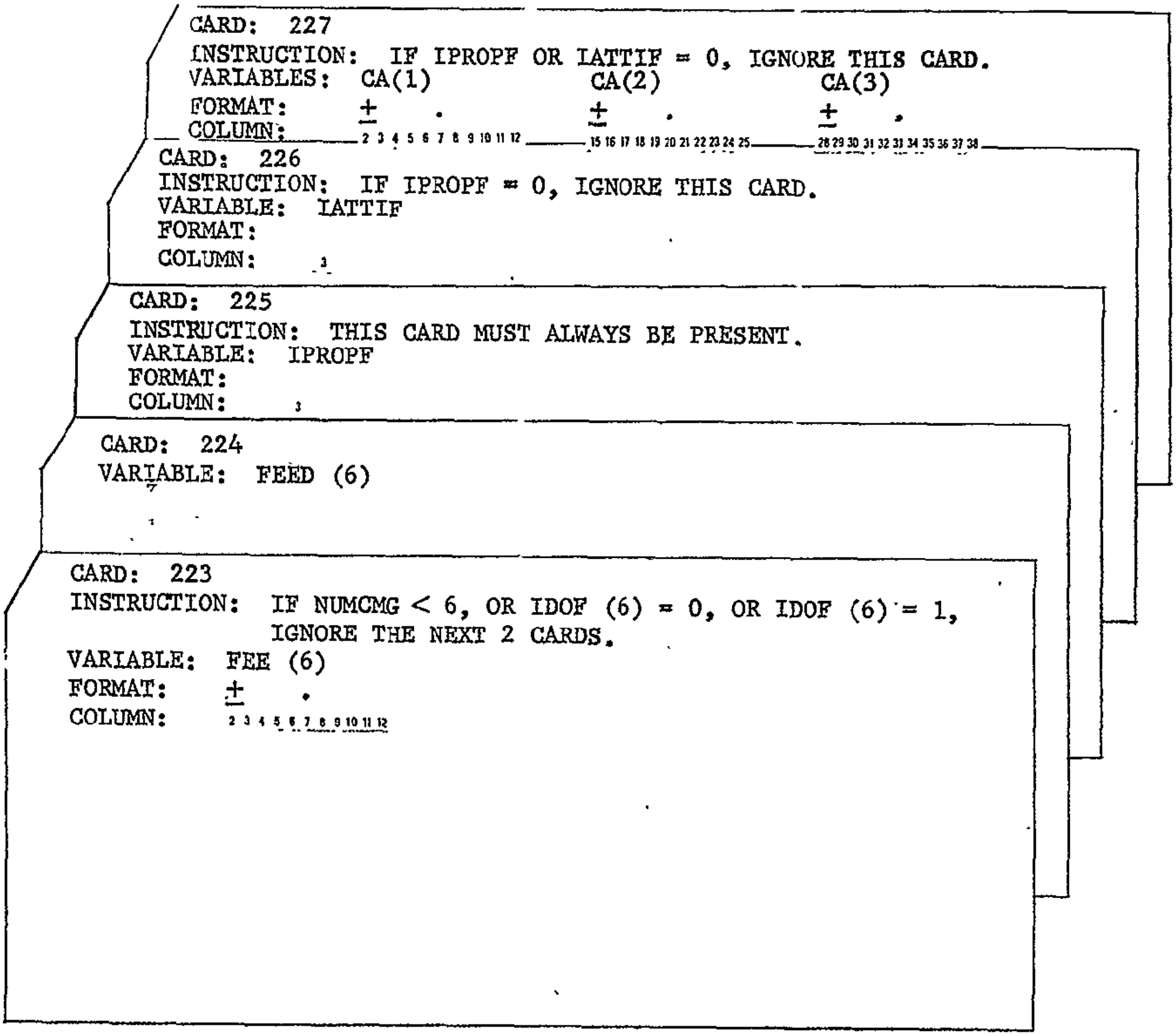
CARD: 217
VARIABLE: AIO (6,2,1)

CARD: 216
VARIABLE: AIO (6,1,3)

CARD: 215
VARIABLE: AIO (6,1,2)

```
CARD:  214
INSTRUCTION:  IF NUMCMG < 6, OR IDOF (6) = 0, OR IDOF (6) = 1,
              IGNORE THE NEXT 9 CARDS.
VARIABLE:  AIO (6,1,1)
FORMAT:    ±         .
COLUMN:    2 3 4 5 6 7 8 9 10 11 12
```

DUE TO MODIFICATIONS, THE NEXT DATA CARD HAS THE NUMBER 230.



CARD: 227
INSTRUCTION: IF IPROPF OR IATTIF = 0, IGNORE THIS CARD.
VARIABLES: CA(1) CA(2) CA(3)
FORMAT: ± . ± . ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12 15 16 17 18 19 20 21 22 23 24 25 28 29 30 31 32 33 34 35 36 37 38

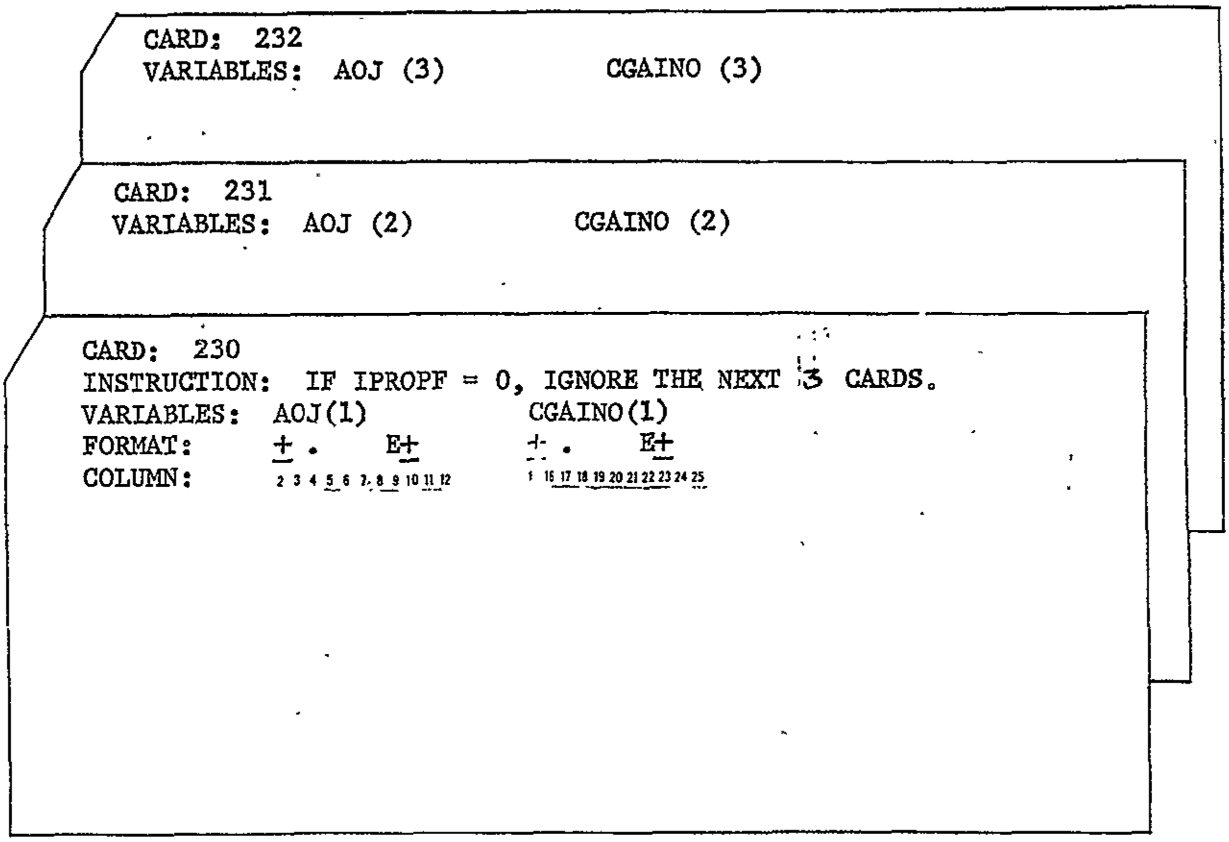
CARD: 226
INSTRUCTION: IF IPROPF = 0, IGNORE THIS CARD.
VARIABLE: IATTIF
FORMAT:
COLUMN: 3

CARD: 225
INSTRUCTION: THIS CARD MUST ALWAYS BE PRESENT.
VARIABLE: IPROPF
FORMAT:
COLUMN: 3

CARD: 224
VARIABLE: FEED (6)

CARD: 223
INSTRUCTION: IF NUMCMG < 6, OR IDOF (6) = 0, OR IDOF (6) = 1, IGNORE THE NEXT 2 CARDS.
VARIABLE: FEE (6)
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

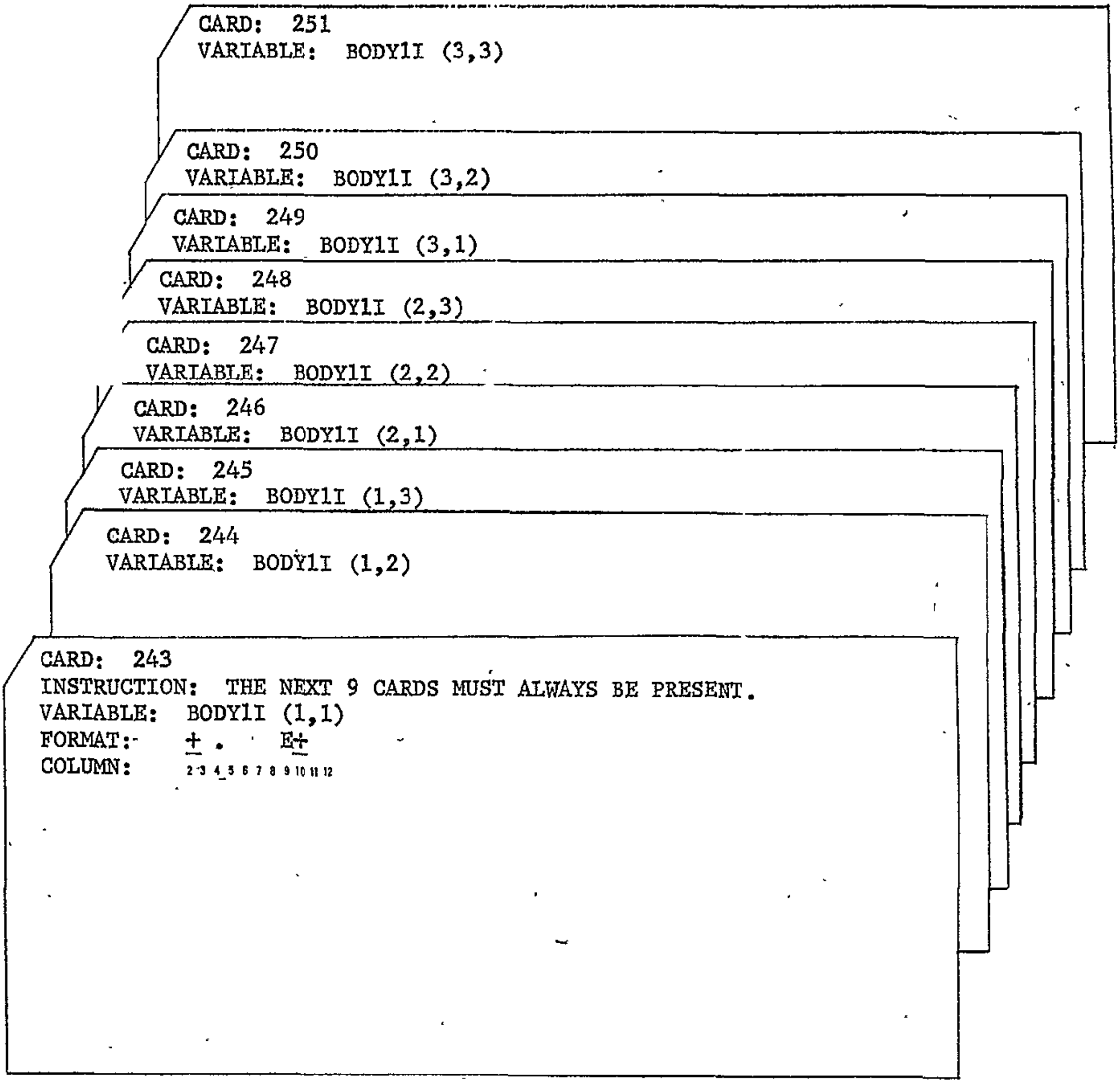
DUE TO MODIFICATIONS, THE NEXT DATA CARD HAS THE NUMBER 242.



CARD: 232
VARIABLES: AOJ (3) CGAINO (3)

CARD: 231
VARIABLES: AOJ (2) CGAINO (2)

CARD: 230
INSTRUCTION: IF IPROPF = 0, IGNORE THE NEXT 3 CARDS.
VARIABLES: AOJ(1) CGAINO(1)
FORMAT: ± . E± ± . E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12 1 16 17 18 19 20 21 22 23 24 25



CARD: 251
VARIABLE: BODY1I (3,3)

CARD: 250
VARIABLE: BODY1I (3,2)

CARD: 249
VARIABLE: BODY1I (3,1)

CARD: 248
VARIABLE: BODY1I (2,3)

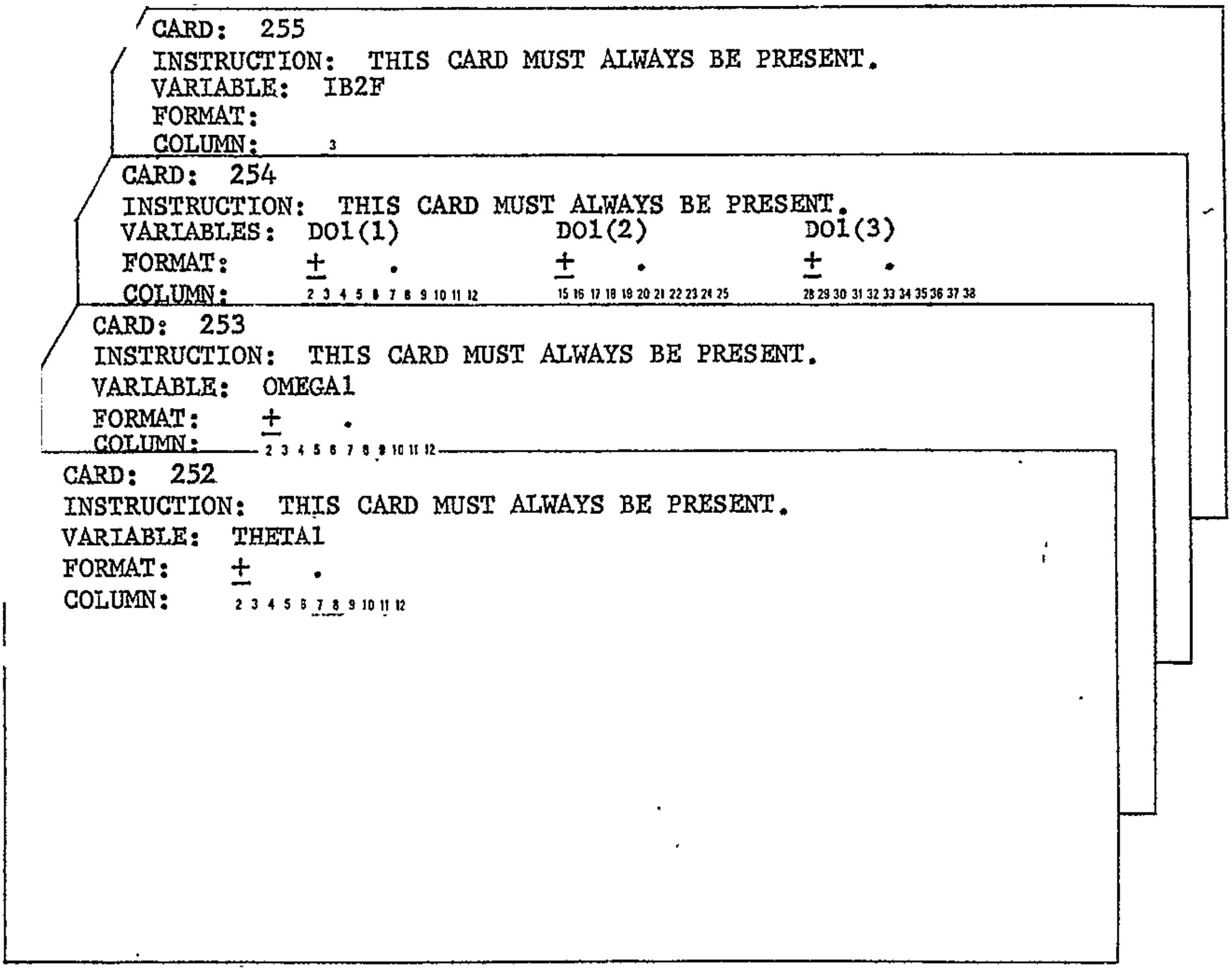
CARD: 247
VARIABLE: BODY1I (2,2)

CARD: 246
VARIABLE: BODY1I (2,1)

CARD: 245
VARIABLE: BODY1I (1,3)

CARD: 244
VARIABLE: BODY1I (1,2)

CARD: 243
INSTRUCTION: THE NEXT 9 CARDS MUST ALWAYS BE PRESENT.
VARIABLE: BODY1I (1,1)
FORMAT: ± . E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12



```
CARD: 255
INSTRUCTION:  THIS CARD MUST ALWAYS BE PRESENT.
VARIABLE:  IB2F
FORMAT:
COLUMN:       3

CARD: 254
INSTRUCTION:  THIS CARD MUST ALWAYS BE PRESENT.
VARIABLES:  D01(1)                     D01(2)                          D01(3)
FORMAT:     ±     .                    ±     .                         ±     .
COLUMN:     2 3 4 5 6 7 8 9 10 11 12   15 16 17 18 19 20 21 22 23 24 25  28 29 30 31 32 33 34 35 36 37 38

CARD: 253
INSTRUCTION:  THIS CARD MUST ALWAYS BE PRESENT.
VARIABLE:  OMEGA1
FORMAT:    ±     .
COLUMN:    2 3 4 5 6 7 8 9 10 11 12

CARD: 252
INSTRUCTION:  THIS CARD MUST ALWAYS BE PRESENT.
VARIABLE:  THETA1
FORMAT:    ±     .
COLUMN:    2 3 4 5 6 7 8 9 10 11 12
```

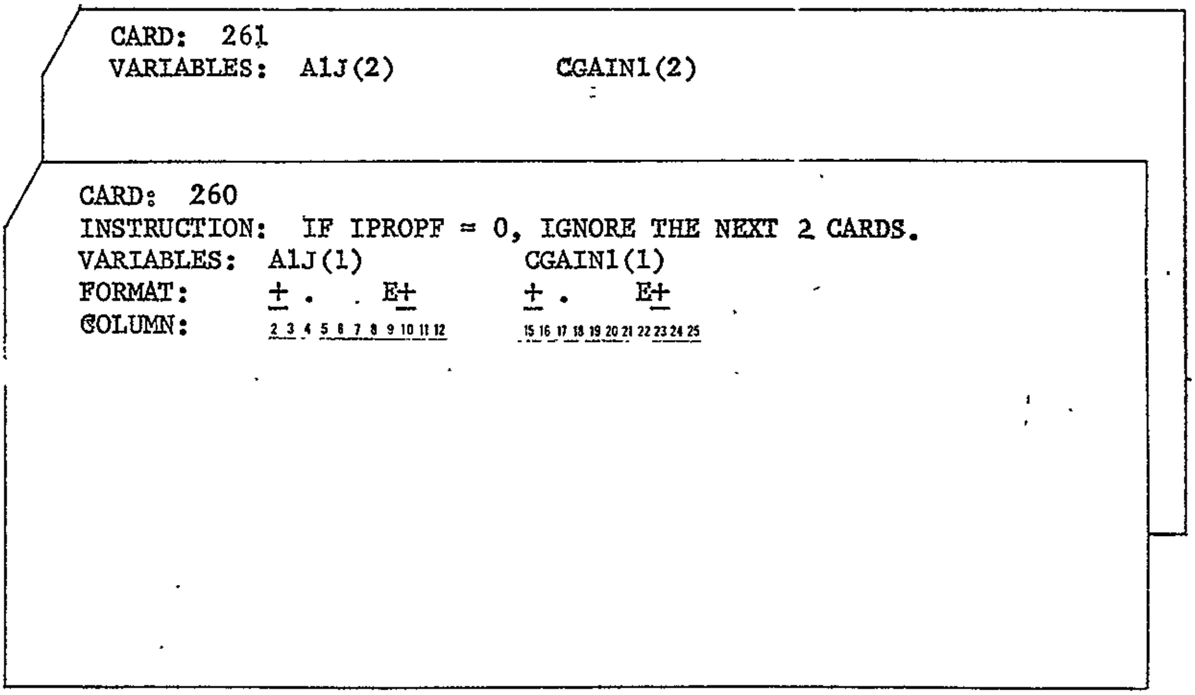
```
CARD:  259
INSTRUCTION:  IF IB2F = 0, IGNORE THIS CARD.
VARIABLES:  S                              SDOT
FORMAT:     ±      .                       ±      .
COLUMN:     2 3 4 5 6 7 8 9 10 11 12       15 16 17 18 19 20 21 22 23 24 25
```

```
CARD:  258
INSTRUCTION:  IF IB2F = 0, IGNORE THIS CARD.
VARIABLES:  S2(1)                          S2(2)                              S2(3)
FORMAT:     ±      .                       ±      .                           ±      .
COLUMN:     2 3 4 5 6 7 8 9 10 11 12       15 16 17 18 19 20 21 22 23 24 25   28 29 30 31 32 33 34 35 36 37 38
```

```
CARD:  257
INSTRUCTION:  IF IB2F = 0, IGNORE THIS CARD.
VARIABLES:  D12(1)                         D12(2)                             D12(3)
FORMAT:     ±    .                         ±      .                           ±      .
COLUMN:     2 3 4 5 6 7 8 9 10 11 12       15 16 17 18 19 20 21 22 23 24 25   28 29 30 31 32 33 34 35 36 37 38
```

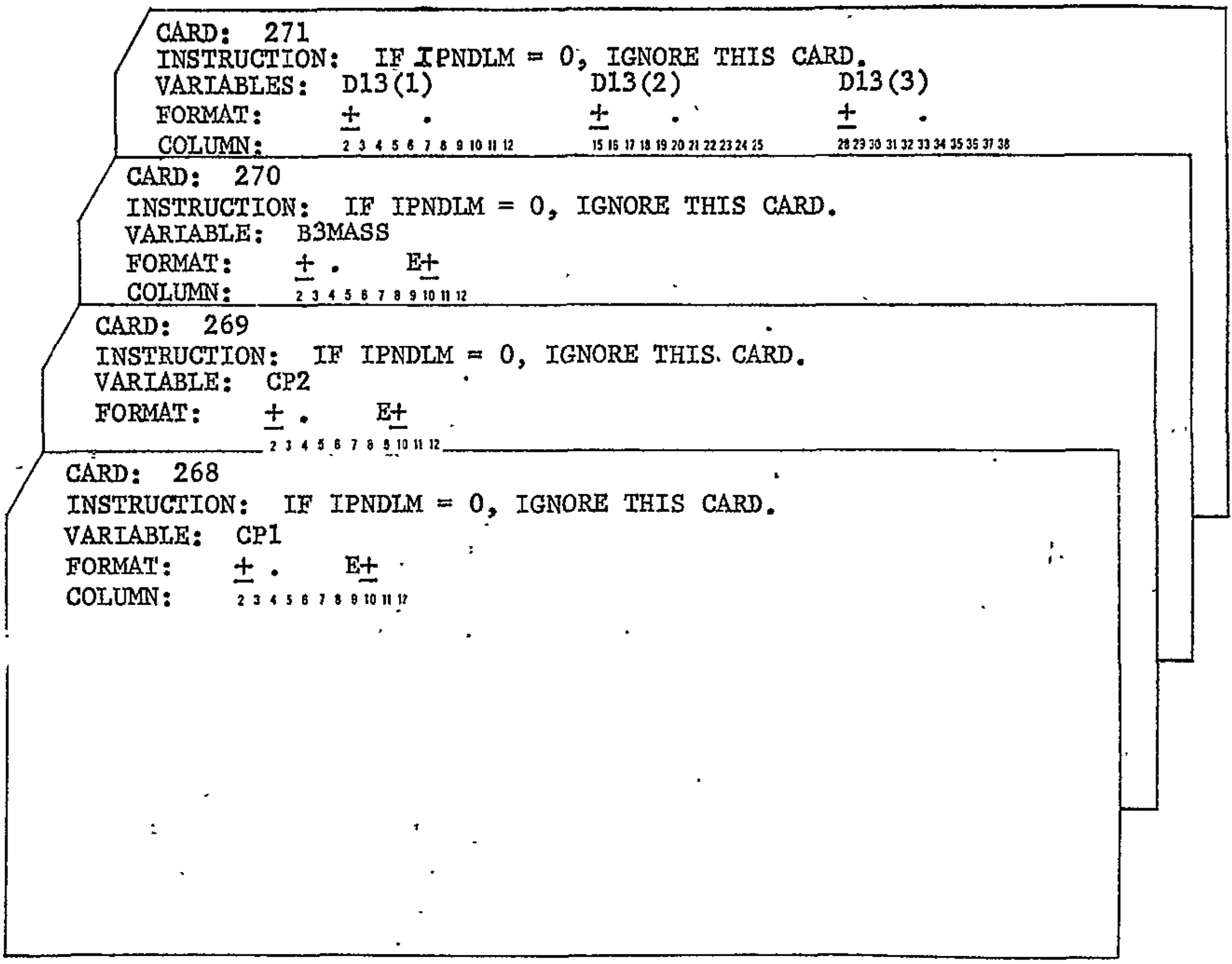
```
CARD:  256
INSTRUCTION:  IF IB2F = 0, IGNORE THIS CARD.
VARIABLE:  B2MASS
FORMAT:     ± .     E±
COLUMN:     2 3 4 5 6 7 8 9 10 11 12
```

DUE TO MODIFICATIONS, THE NEXT DATA CARD HAS THE NUMBER 268.



```
CARD:  261
VARIABLES:  A1J(2)          CGAIN1(2)
```

```
CARD:  260
INSTRUCTION:  IF IPROPF = 0, IGNORE THE NEXT 2 CARDS.
VARIABLES:  A1J(1)                 CGAIN1(1)
FORMAT:     ± .    . E±            ± .      E±
COLUMN:     2 3 4 5 6 7 8 9 10 11 12    15 16 17 18 19 20 21 22 23 24 25
```

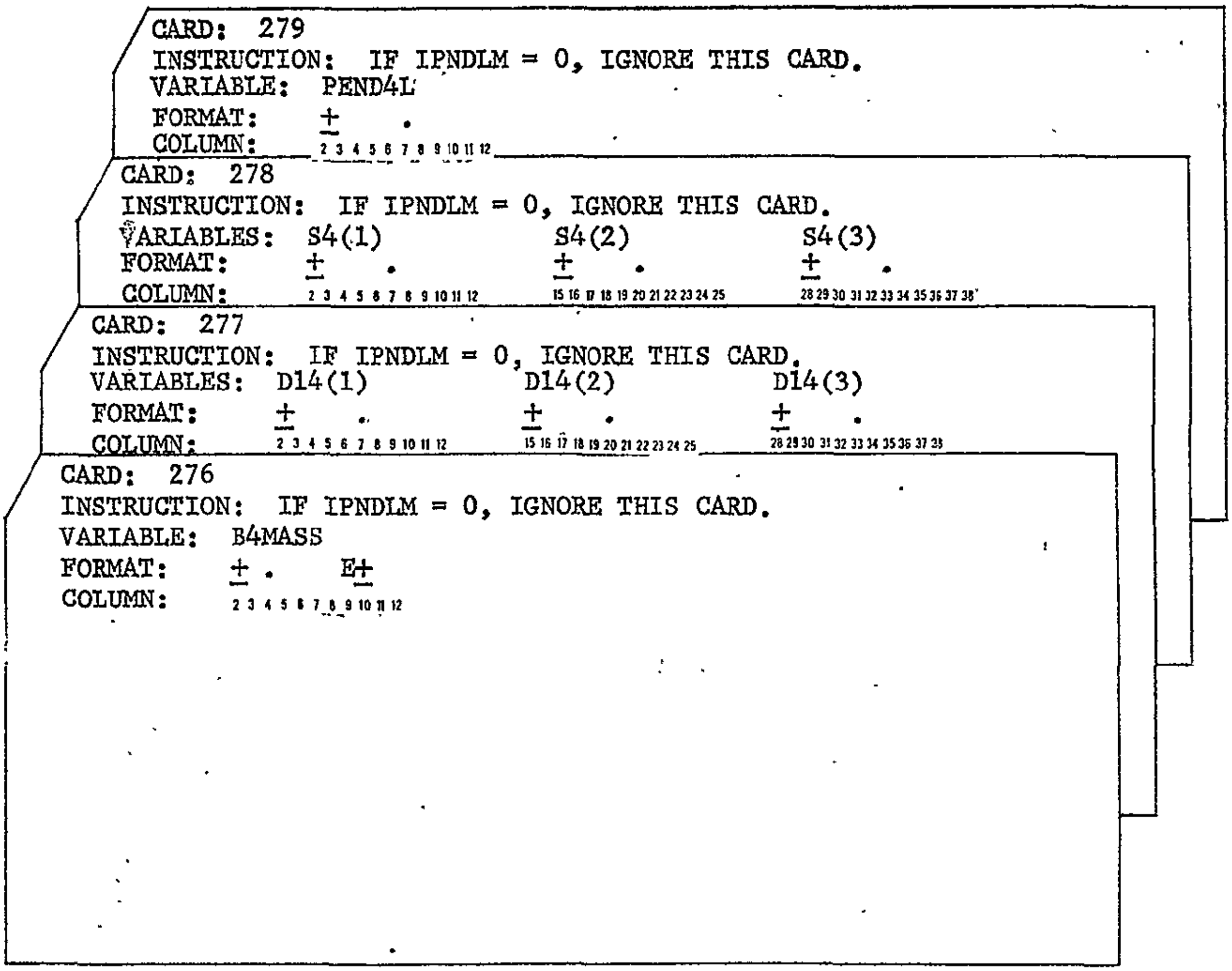


CARD: 271
INSTRUCTION: IF IPNDLM = 0, IGNORE THIS CARD.
VARIABLES: D13(1) D13(2) D13(3)
FORMAT: ± . ± . ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12 15 16 17 18 19 20 21 22 23 24 25 28 29 30 31 32 33 34 35 36 37 38

CARD: 270
INSTRUCTION: IF IPNDLM = 0, IGNORE THIS CARD.
VARIABLE: B3MASS
FORMAT: ± . E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 269
INSTRUCTION: IF IPNDLM = 0, IGNORE THIS CARD.
VARIABLE: CP2
FORMAT: ± . E±
2 3 4 5 6 7 8 9 10 11 12

CARD: 268
INSTRUCTION: IF IPNDLM = 0, IGNORE THIS CARD.
VARIABLE: CP1
FORMAT: ± . E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

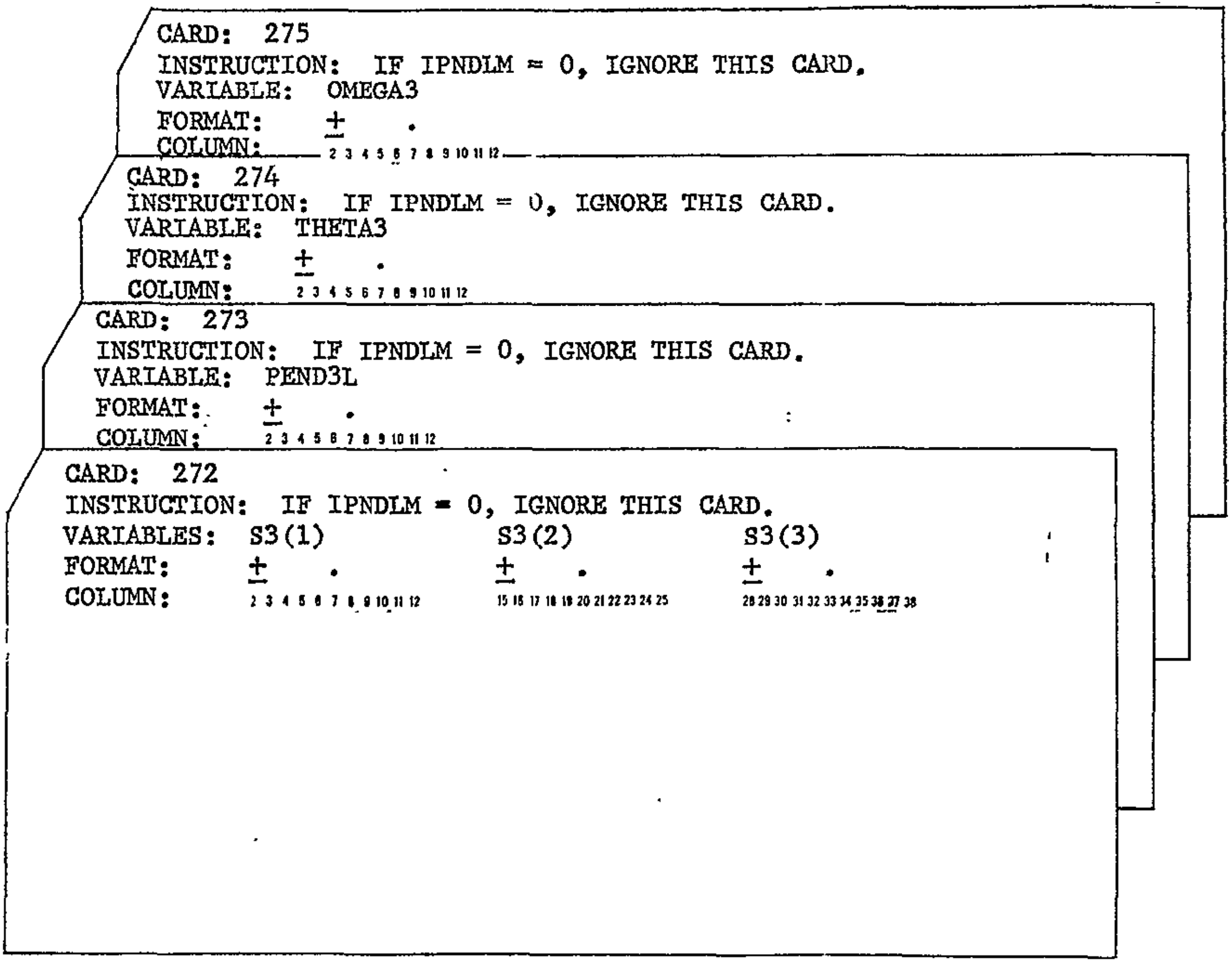


CARD: 279
INSTRUCTION: IF IPNDLM = 0, IGNORE THIS CARD.
VARIABLE: PEND4L
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 278
INSTRUCTION: IF IPNDLM = 0, IGNORE THIS CARD.
VARIABLES: S4(1) S4(2) S4(3)
FORMAT: ± . ± . ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12 15 16 17 18 19 20 21 22 23 24 25 28 29 30 31 32 33 34 35 36 37 38

CARD: 277
INSTRUCTION: IF IPNDLM = 0, IGNORE THIS CARD.
VARIABLES: D14(1) D14(2) D14(3)
FORMAT: ± . ± . ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12 15 16 17 18 19 20 21 22 23 24 25 28 29 30 31 32 33 34 35 36 37 38

CARD: 276
INSTRUCTION: IF IPNDLM = 0, IGNORE THIS CARD.
VARIABLE: B4MASS
FORMAT: ± . E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12



CARD: 275

INSTRUCTION: IF IPNDLM = 0, IGNORE THIS CARD.

VARIABLE: OMEGA3

FORMAT: ± .

COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 274

INSTRUCTION: IF IPNDLM = 0, IGNORE THIS CARD.

VARIABLE: THETA3

FORMAT: ± .

COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 273

INSTRUCTION: IF IPNDLM = 0, IGNORE THIS CARD.

VARIABLE: PEND3L

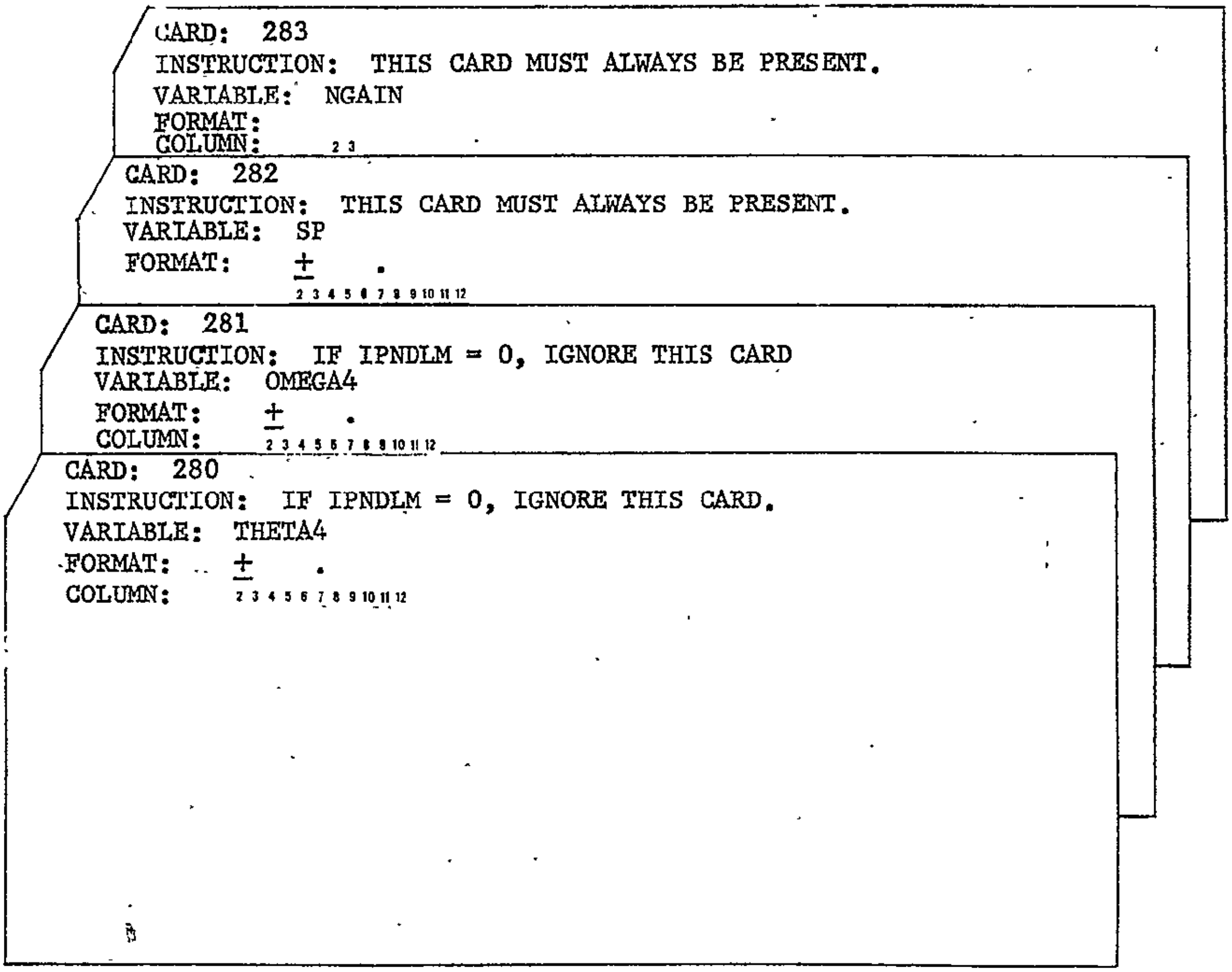
FORMAT: ± .

COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 272

INSTRUCTION: IF IPNDLM = 0, IGNORE THIS CARD.

VARIABLES:	S3(1)	S3(2)	S3(3)
FORMAT:	± .	± .	± .
COLUMN:	2 3 4 5 6 7 8 9 10 11 12	15 16 17 18 19 20 21 22 23 24 25	28 29 30 31 32 33 34 35 36 37 38

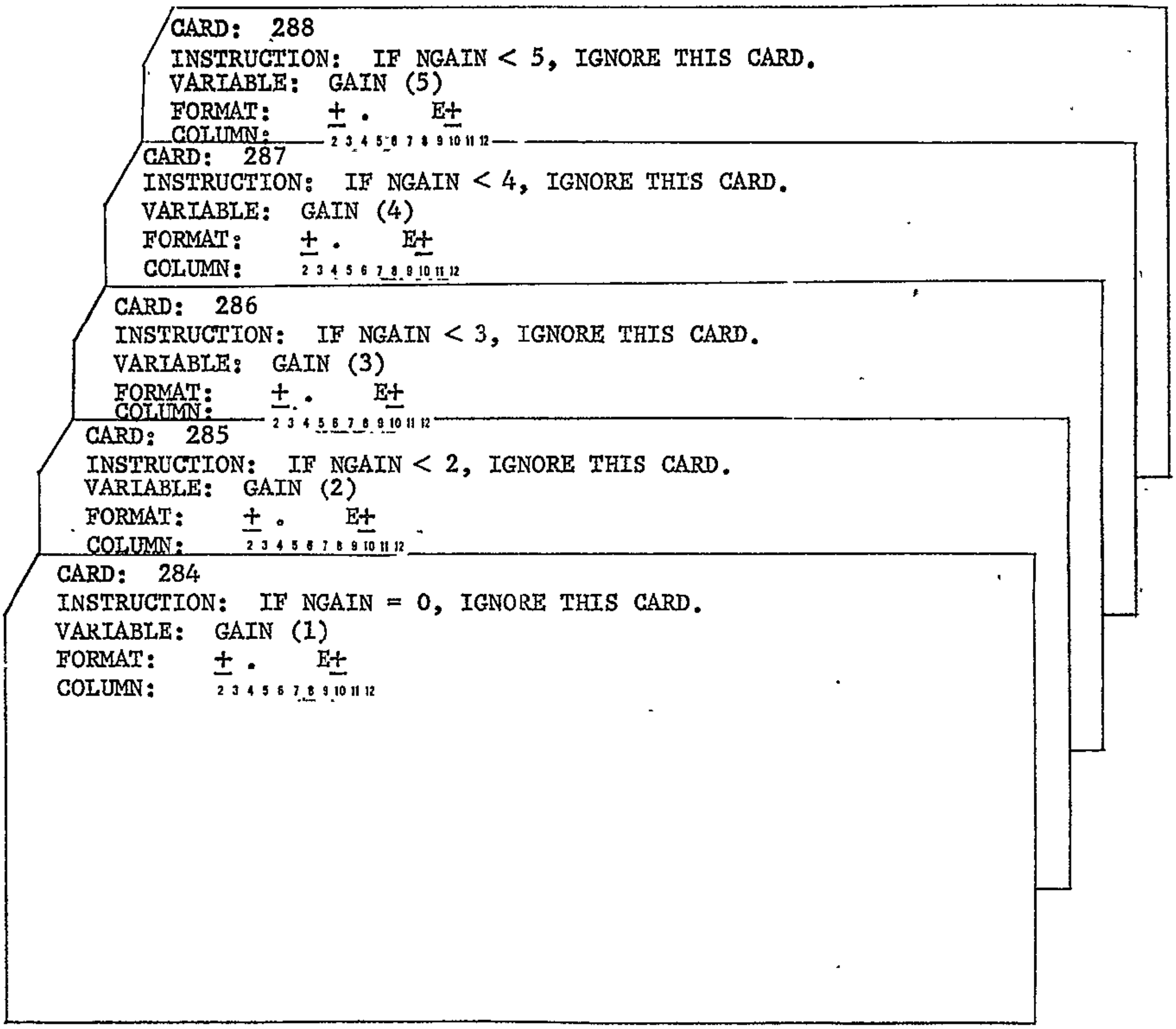


CARD: 283
INSTRUCTION: THIS CARD MUST ALWAYS BE PRESENT.
VARIABLE: NGAIN
FORMAT:
COLUMN: 2 3

CARD: 282
INSTRUCTION: THIS CARD MUST ALWAYS BE PRESENT.
VARIABLE: SP
FORMAT: ± .
2 3 4 5 6 7 8 9 10 11 12

CARD: 281
INSTRUCTION: IF IPNDLM = 0, IGNORE THIS CARD
VARIABLE: OMEGA4
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 280
INSTRUCTION: IF IPNDLM = 0, IGNORE THIS CARD.
VARIABLE: THETA4
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12



CARD: 288

INSTRUCTION: IF NGAIN < 5, IGNORE THIS CARD.

VARIABLE: GAIN (5)

FORMAT: ± . E±

COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 287

INSTRUCTION: IF NGAIN < 4, IGNORE THIS CARD.

VARIABLE: GAIN (4)

FORMAT: ± . E±

COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 286

INSTRUCTION: IF NGAIN < 3, IGNORE THIS CARD.

VARIABLE: GAIN (3)

FORMAT: ± . E±

COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 285

INSTRUCTION: IF NGAIN < 2, IGNORE THIS CARD.

VARIABLE: GAIN (2)

FORMAT: ± . E±

COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 284

INSTRUCTION: IF NGAIN = 0, IGNORE THIS CARD.

VARIABLE: GAIN (1)

FORMAT: ± . E±

COLUMN: 2 3 4 5 6 7 8 9 10 11 12

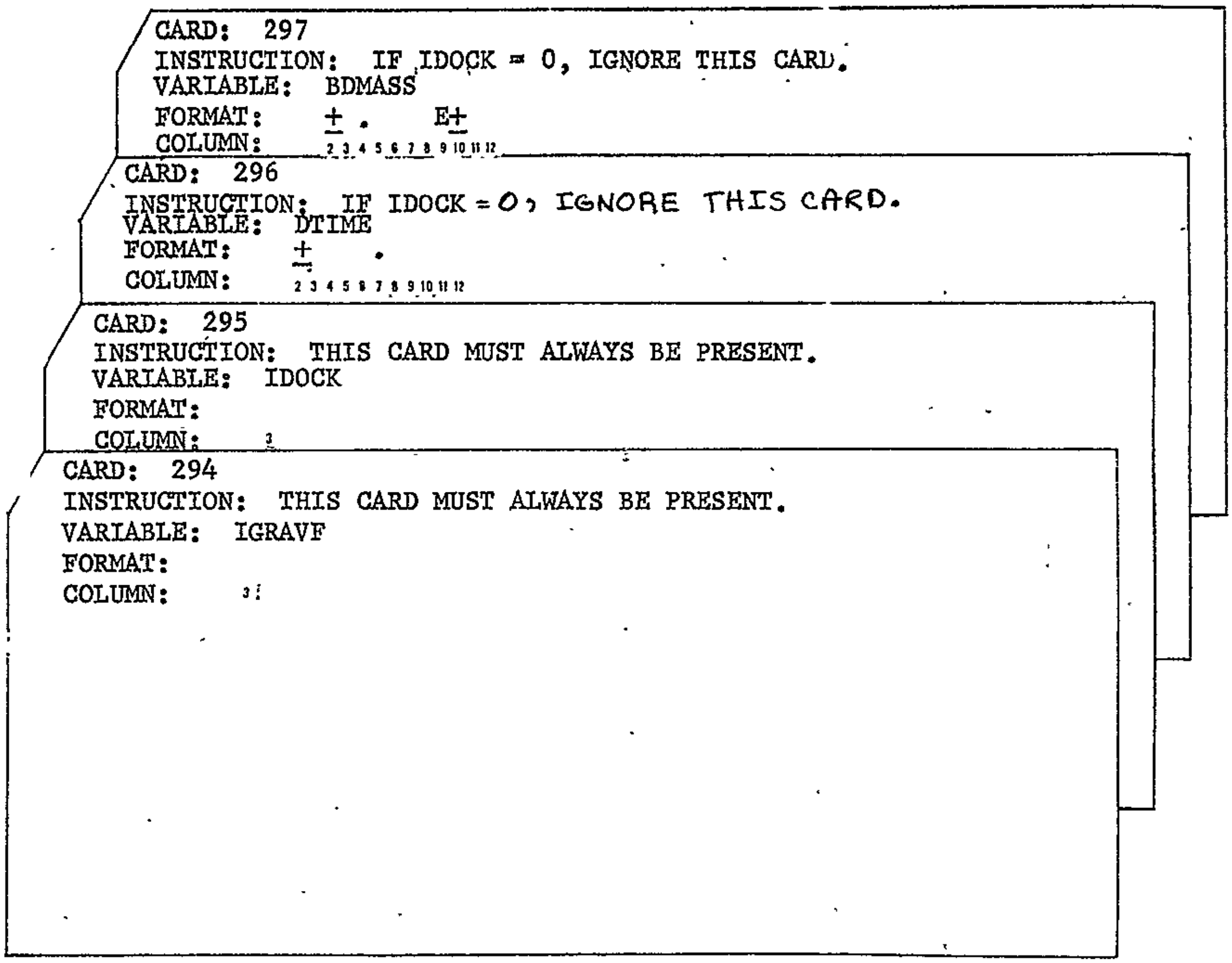
CARD: 293
INSTRUCTION: IF NGAIN < 10, IGNORE THIS CARD.
VARIABLE: GAIN (10)
FORMAT: ± . E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 292
INSTRUCTION: IF NGAIN < 9, IGNORE THIS CARD.
VARIABLE: GAIN (9)
FORMAT: ± . E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 291
INSTRUCTION: IF NGAIN < 8, IGNORE THIS CARD.
VARIABLE: GAIN (8)
FORMAT: ± . E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 290
INSTRUCTION: IF NGAIN < 7, IGNORE THIS CARD.
VARIABLE: GAIN (7)
FORMAT: ± . E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 289
INSTRUCTION: IF NGAIN < 6, IGNORE THIS CARD.
VARIABLE: GAIN (6)
FORMAT: ± . E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12



CARD: 297
INSTRUCTION: IF IDOCK = 0, IGNORE THIS CARD.
VARIABLE: BDMASS
FORMAT: ± . E±
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 296
INSTRUCTION: IF IDOCK = 0, IGNORE THIS CARD.
VARIABLE: DTIME
FORMAT: ± .
COLUMN: 2 3 4 5 6 7 8 9 10 11 12

CARD: 295
INSTRUCTION: THIS CARD MUST ALWAYS BE PRESENT.
VARIABLE: IDOCK
FORMAT:
COLUMN: 3

CARD: 294
INSTRUCTION: THIS CARD MUST ALWAYS BE PRESENT.
VARIABLE: IGRAVF
FORMAT:
COLUMN: 3

CARD: 306
VARIABLE: BODYDI (3,3)

CARD: 305
VARIABLE: BODYDI (3,2)

CARD: 304
VARIABLE: BODYDI (3,1)

CARD: 303
VARIABLE: BODYDI (2,3)

CARD: 302
VARIABLE: BODYDI (2,2)

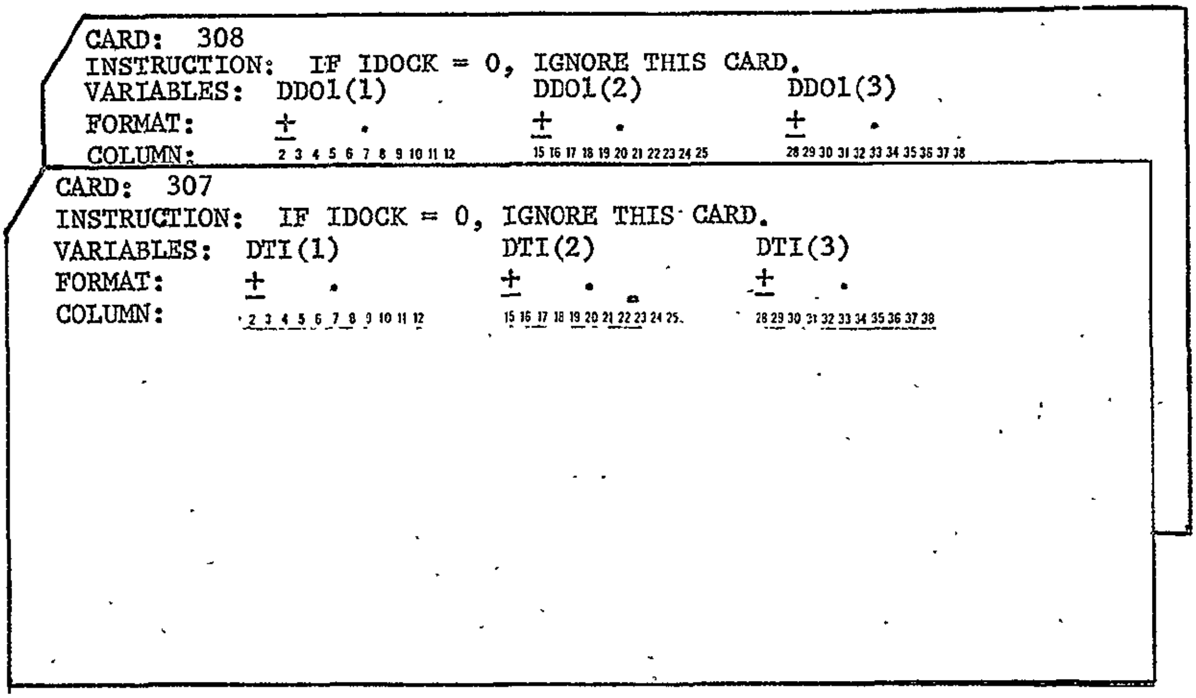
CARD: 301
VARIABLE: BODYDI (2,1)

CARD: 300
VARIABLE: BODYDI (1,3)

CARD: 299
VARIABLE: BODYDI (1,2)

CARD: 298
INSTRUCTION: IF IDOCK = 0, IGNORE THE NEXT 9 CARDS.
VARIABLE: BODYDI (1,1)

```
FORMAT:    ± .      E±
COLUMN:    2 3 4 5 6 7 8 9 10 11 12
```



```
CARD:  308
INSTRUCTION:  IF IDOCK = 0, IGNORE THIS CARD.
VARIABLES:  DD01(1)                 DD01(2)                  DD01(3)
FORMAT:       ±     .                 ±     .                  ±     .
COLUMN:     2 3 4 5 6 7 8 9 10 11 12  15 16 17 18 19 20 21 22 23 24 25  28 29 30 31 32 33 34 35 36 37 38
```

```
CARD:  307
INSTRUCTION:  IF IDOCK = 0, IGNORE THIS CARD.
VARIABLES:  DTI(1)                  DTI(2)                   DTI(3)
FORMAT:       ±     .                 ±     .                  ±     .
COLUMN:     2 3 4 5 6 7 8 9 10 11 12  15 16 17 18 19 20 21 22 23 24 25  28 29 30 31 32 33 34 35 36 37 38
```

APPENDIX A, DEFINITIONS AND REFERENCES FOR INPUT VARIABLES

This appendix is in two parts. The first part contains a list of all input variables in alphabetical order. The second part contains the input variables in categories and the input variables in each category are alphabetized.

[AII(J,M,N)]:

$$[AII(J,M,N)] = [II_J] = \begin{bmatrix} AII(J,1,1) & AII(J,1,2) & AII(J,1,3) \\ AII(J,2,1) & AII(J,2,2) & AII(J,2,3) \\ AII(J,3,1) & AII(J,3,2) & AII(J,3,3) \end{bmatrix}$$

This array is the inertia matrix for the inner gimbal of the Jth control moment gyro aboard the stator. A maximum of six CMGs may be used. (i.e. J = 1,6) All CMGs are also constrained to be located at the center of mass on the stator, body 0. If the Jth CMG has one or two degrees of freedom, it will have an inner gimbal. (Refer to the write up on CMGs for further discussion.)

UNITS: (slug-ft^2)

FORMAT: # 5006 = (1X, F11.5)

AIO(J,M,N):

$$[AIO(J,M,N)] = [IO_J] = \begin{bmatrix} AIO(J,1,1) & AIO(J,1,2) & AIO(J,1,3) \\ AIO(J,2,1) & AIO(J,2,2) & AIO(J,2,3) \\ AIO(J,3,1) & AIO(J,3,2) & AIO(J,3,3) \end{bmatrix}$$

This array is the inertia matrix for the outer gimbal of the Jth control moment gyro aboard the stator. A maximum of six CMGs may be used. (i.e. J = 1,6) All CMGs are constrained to be located at the center of mass on the stator, body 0. The

Jth CMG will have an outer gimbal only if it has two degrees of freedom. (Refer to the write up on CMGs for further discussion.)

UNITS: (slug-ft^2)

FORMAT: # 5006 = (1X, F11.5)

<u>ALT</u>:

ALT is altitude of the center of mass of the space station configuration measured from the surface of the earth. All orbits are constrained to be circular with no oblateness effects. Therefore, altitude is the only pertinent orbit parameter.

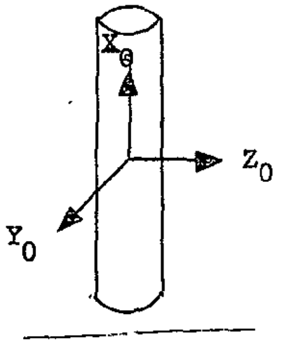
UNITS: (miles)

FORMAT: # 5004 = (1X, E11.4)

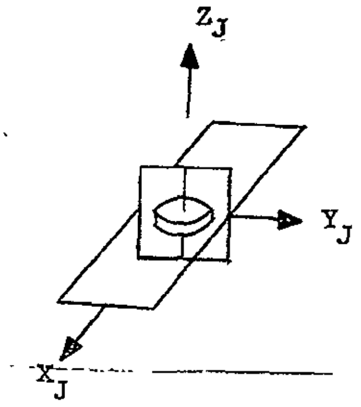
<u>AOCJ(J,M,N)</u>:

$$[\mathrm{AOCJ}(J,M,N)] = [0,C_J] = \begin{bmatrix} \mathrm{AOCJ}(J,1,1) & \mathrm{AOCJ}(J,1,2) & \mathrm{AOCJ}(J,1,3) \\ \mathrm{AOCJ}(J,2,1) & \mathrm{AOCJ}(J,2,2) & \mathrm{AOCJ}(J,2,3) \\ \mathrm{AOCJ}(J,3,1) & \mathrm{AOCJ}(J,3,2) & \mathrm{AOCJ}(J,3,3) \end{bmatrix}$$

This array is the coordinate transformation matrix <u>from</u> the CMG null gimbal coordinate frame <u>to</u> the coordinate frame of body 0, the stator. (Refer to the coordinate transformation appendix.) For example, consider the two coordinate systems shown below:



Body 0 Frame



Jth Null Gimbal Frame

For the above situation:

$$[AOCJ(J,M,N)] = \begin{bmatrix} 0. & 0. & 1. \\ 1. & 0. & 0. \\ 0. & 1. & 0. \end{bmatrix}$$

UNITS: (None)

FORMAT: # 5006 = (1X, F11.5)

AOJ(M):

AOJ(1) is the distance between the jets of the pure couple producing the torque around the X axis of body 0.

AOJ(2) is the distance between the jets of the pure couple producing the torque around the Y axis of body 0.

AOJ(3) is the distance between the jets of the pure couple producing the torque around the Z axis of body 0.

Note: The variables CGAIN0(1), CGAIN0(2), and CGAIN0(3) are also read on the same data cards as AOJ(1), AOJ(2), and AOJ(3). Punch the values of AOJ(1), AOJ(2), and AOJ(3) in columns 2 through 12 of the data cards. Non-zero values should be read in for AOJ(M). Control on an axis can be disabled by reading zero for the CGAIN0.

UNITS: (feet)

FORMAT: # 5008 = (1X, E11.4, 2X, E11.4)

A1J(M):

A1J(1) is the distance between the jets of the pure couple producing the torque around the X axis of body 1.

A1J(2) is the distance between the jets of the pure couple producing the torque around the Y axis of body 1.

Note: The variables CGAIN1(1) and CGAIN1(2) are also read on the same data cards as A1J(1) and A1J(2). Punch the values of A1J(1) and A1J(2) in columns 2 through 12 of the data cards. Non-zero values should be read in for A1J(M). Control on an axis can be disabled by reading zero for the CGAIN1.

UNITS: (feet)

FORMAT: # 5008 = (1X, E11.4, 2X, E11.4)

BDMASS:

BDMASS is the sum of the mass of body 0, the stator, and the mass of the docking vehicle. The docking vehicle is constrained to dock on body 0, the stator. BDMASS = m_D

UNITS: (slugs)

FORMAT: # 5004 = (1X, E11.4)

BODYDI(M,N):

$$[\text{BODYDI}(M,N)] = [I_D] = \begin{bmatrix} \text{BODYDI}(1,1) & \text{BODYDI}(1,2) & \text{BODYDI}(1,3) \\ \text{BODYDI}(2,1) & \text{BODYDI}(2,2) & \text{BODYDI}(2,3) \\ \text{BODYDI}(3,1) & \text{BODYDI}(3,2) & \text{BODYDI}(3,3) \end{bmatrix}$$

This array is the inertia matrix of the docked body which consists of the docking vehicle connected to the stator.

UNITS: (slug-ft^2)

FORMAT: # 5004 = (1X, E11.4)

BODYOI(M,N):

$$[\text{BODYOI}(M,N)] = [I_0] = \begin{bmatrix} \text{BODYOI}(1,1) & \text{BODYOI}(1,2) & \text{BODYOI}(1,3) \\ \text{BODYOI}(2,1) & \text{BODYOI}(2,2) & \text{BODYOI}(2,3) \\ \text{BODYOI}(3,1) & \text{BODYOI}(3,2) & \text{BODYOI}(3,3) \end{bmatrix}$$

This array is the inertia matrix of body 0, the stator.

UNITS: (slug-ft^2)

FORMAT: # 5004 = (1X, E11.4)

BODY1I(M,N):

$$[\text{BODY1I(M,N)}] = [I_1] = \begin{bmatrix} \text{BODY1I(1,1)} & \text{BODY1I(1,2)} & \text{BODY1I(1,3)} \\ \text{BODY1I(2,1)} & \text{BODY1I(2,2)} & \text{BODY1I(2,3)} \\ \text{BODY1I(3,1)} & \text{BODY1I(3,2)} & \text{BODY1I(3,3)} \end{bmatrix}$$

This array is the inertia matrix for body 1, the rotor.

UNITS: (slug-ft^2)

FORMAT: # 5004 = (1X, E11.4)

BOMASS:

BOMASS is the mass of body 0, the stator. BOMASS = m_0

UNITS: (slugs)

FORMAT: # 5004 = (1X, E11.4)

B1MASS:

B1MASS is the mass of body 1, the rotor. B1MASS = m_1

UNITS: (slugs)

FORMAT: # 5004 = (1X, E11.4)

B2MASS:

B2MASS is the mass of body 2, the elevator. B2MASS = m_2

UNITS: (slugs)

FORMAT: # 5004 = (1X, E11.4)

B3MASS:

B3MASS is the mass of body 3, a pendulum. B3MASS = m_3

UNITS: (slugs)

FORMAT: # 5004 = (1X, E11.4)

B4MASS:

B4MASS is the mass of body 4, a pendulum. B4MASS = m_4

UNITS: (slugs)

FORMAT: # 5004 = (1X, E11.4)

CA(M):

CA(1), CA(2), and CA(3) are the three direction cosines of the desired attitude reference direction in the inertial frame.

UNITS: (None)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

CGAINO(M):

CGAINO(1) equals the control gain of all four reaction jets in the pure force couples producing torque around the X axis of body 0.

CGAINO(2) equals the control gain of all four reaction jets in the pure force couples producing torque around the Y axis of body 0.

CGAINO(3) equals the control gain of all four reaction jets in the pure force couples producing torque around the Z axis of body 0.

Note: The variables AOJ(1), AOJ(2), and AOJ(3) are also read in on the same data cards as CGAINO(1), CGAINO(2), and CGAINO(3). Punch the values of CGAINO(1), CGAINO(2), and CGAINO(3) in columns 15 through 25 of the data cards.

UNITS: (lb/radians/second)

FORMAT: # 5008 = (1X, E11.4, 2X, E11.4)

CGAIN1(M):

CGAIN1(1) equals the control gain of all four reaction jets in the pure force couples producing torque around the X axis of body 1.

CGAIN1(2) equals the control gain of all four reaction jets in the pure force couples producing torque around the Y axis of body 1.

Note: The variables A1J(1) and A1J(2) are also read in on the same date cards as CGAIN1(1) and CGAIN1(2). Punch the values of CGAIN1(1) and CGAIN1(2) in columns 15 through 25 of the data cards.

UNITS: (lb/radians/second)

FORMAT: # 5008 = (1X, E11.4, 2X, E11.4)

CP1 and CP2:

CP1 and CP2 are parameters used in determining the stiffness and the resonant frequency of the pendulums. For example, the control laws which are governed by the parameters are:

T13 = - CP1 * OMEGA3 - CP2 * THETA3

T14 = - CP1 * OMEGA4 - CP2 * (THETA4 - π)

Note: These variables are read on separate data cards.

UNITS: (none)

FORMAT: # 5004 = (1X, E11.4)

DDO1(M):

DDO1(1), DDO1(2), and DDO1(3) are the X, Y, and Z components of the vector from the center of mass of the docked body to the hinge line of body 1, the rotor. The vector is expressed in body 0 coordinates. The docked body shall be defined as the configuration of the docking vehicle attached to body 0.

UNITS: (feet)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

DELTAT:

DELTAT = Δt

DELTAT is the time increment used in the integration algorithm which solves the rotational equations of motion. When "TIME" is updated we have:

TIME = TIME + DELTAT

Note: Punch the value of DELTAT in columns 28 through 38

UNITS: (seconds)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

DO1(M):

DO1(1), DO1(2), and DO1(3) are the X, Y, and Z components of the vector from the center of mass of body 0 to the hinge line of body 1, the rotor. The vector is expressed in body 0 coordinates. DO1(1) should in most cases be zero.

UNITS: (feet)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

DTI(M):

DTI(1), DTI(2) and DTI(3) are the X, Y, and Z components of the docking torque impulse.

UNITS: lb-ft-sec

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

DTIME:

DT1ME is the time when docking occurs. (i.e. "Docking Time") The restrictions placed on DTIME are as follows:

a) $T_{start} <$ DTIME $< T_{stop}$

b) DTIME must be an integer multiple of Δt.

UNITS: (seconds)

FORMAT: # 5006 = (1X, F11.5)

D12(M):

D12(1), D12(2) and D12(3) are the X, Y and Z components of a fixed vector locating the starting position of the movable mass, the elevator. The vector equation that describes the motion of the movable mass is $\bar{\ell}_2 = \bar{d}_{12} + \bar{s}_2 s$. Where $\bar{s}_2$ is a unit vector which defines the direction in which the movable mass travels and s is a scalar prespecified function of time. For most cases $s(t_o = 0) = 0$ so that $\bar{d}_{12}$ specifies the initial starting position of the movable mass. The elevator is known alternately as the movable mass or body 2. D12 is expressed in body 1 coordinates. For example, if the elevator were constrained to travel along the X axis, then $\bar{d}_{12}$ could have the following values: D12(1) = 1., D12(2) = 0., D12(3) = 0.

UNITS: (feet)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

D13(M):

D13(1), D13(2), and D13(3) are the X, Y, and Z components of the vector from the center of mass of body 1 to the hinge line of body 3. $\overline{D13}$ is expressed in body 1 coordinates.

UNITS: (feet)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

D14(M):

D14(1), D14(2), and D14(3) are the X, Y, and Z components of the vector from the center of mass of body 1 to the hinge line of body 4.

D14 is expressed in body 1 coordinates.

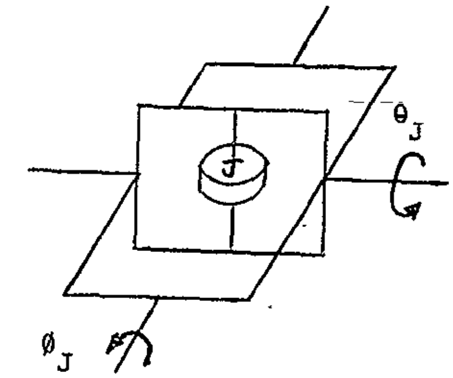
UNITS: (feet)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

FEE(J):

FEE(J) = $\emptyset_J$

FEE(J) is the outer gimbal angle of a two degree of freedom control moment gyro as shown pictorially below:



The subscript J refers to the number assigned to the CMG. (Refer to the write up on CMGs for further discussion.)

UNITS: (radians)

FORMAT: # 5006 = (1X, F11.5)

INITIAL VALUE AT TIME = T_{start}

FEED(J):

FEED(J) = $\dot{\phi}_J$

FEED(J) is the outer gimbal rate of a two degree of freedom control moment gyro. The subscript J refers to the number assigned to the CMG. (Refer to the write up on CMGs for further discussion.)

UNITS: (radians/second)

FORMAT: # 5006 = (1X, F11.5)

GAIN(M):

GAIN(M) is an array of numbers the dimension of which is determined by another input variable NGAIN. These numbers, once read in, are stored in common and can be used for a variety of purposes. In many CMG control laws it is necessary to have control gains. GAIN(M) can be used for this purpose among others.

UNITS: (None)

FORMAT: # 5004 = (1X, E11.4)

HW(J):

HW(J) is the angular momentum of the wheel associated with the Jth momentum device. (Refer to the write up on CMGs for further discussion.)

UNITS: (slug-ft^2/second)

FORMAT: # 5004 = (1X, E11.4)

IATTIF:

IATTIF is the attitude flag.

IATTIF = 1 implies an attempt to change attitude will be made.

IATCIF = 0 implies an attempt to change attitude will not be made.

SPECIAL INSTRUCTIONS: Punch a 0 or a 1 in column 3 of the data card.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

IB2F:

IB2F is the body 2 flag. (i.e. the elevator flag)

IB2F = 1 implies body 2 will be present.

IB2F = 0 implies body 2 will not be present.

SPECIAL INSTRUCTIONS: Punch a 0 or a 1 in column 3 of the data card.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

IDOCK:

IDOCK is the docking flag.

IDOCK = 1 implies docking will occur.

IDOCK = 0 implies docking will not occur.

SPECIAL INSTRUCTIONS: Punch a 0 or a 1 in column 3 of the data card.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

IDOF(J):

IDOF(J) = 0 implies the Jth controller aboard body 0 is a reaction wheel.

IDOF(J) = 1 implies the Jth controller aboard body 0 is a one degree of freedom control moment gyro.

IDOF(J) = 2 implies the Jth controller aboard body 0 is a two degree of freedom control moment gyro.

SPECIAL INSTRUCTIONS: Punch a 0, 1, or a 2 in column 3 of the data card.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

IGRAVF:

IGRAVF is the gravity gradient flag.

IGRAVF = 1 implies gravity gradient torques will be present.

IGRAVF = 0 implies gravity gradient torques will not be present.

SPECIAL INSTRUCTIONS: Punch a 0 or a 1 in column 3 of the data card.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

IPNDLM:

IPNDLM is the pendulum flag.

IPNDLM = 1 implies body 3 and body 4, the pendulums, will be present.

IPNDLM = 0 implies body 3 and body 4, the pendulums, will not be present.

SPECIAL INSTRUCTIONS: Punch a 0 or a 1 in column 3 of the data card.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

IPRINT:

IPRINT is an integer variable used to determine how often output data is printed. Data is printed in time increments of IPRINT * DELTAT. For example, if IPRINT = 50 and DELTAT = .2, then data will be printed when time = 10, 20, 30, 40, 50, ...

SPECIAL INSTRUCTIONS: If IPRINT has a value less than 10, punch the integer in column 3 of the data card. If IPRINT has a value greater than 9, punch the integer in columns 2 and 3.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

IPROPF:

IPROPF is the propulsion flag.

IPROPF = 1 implies propulsion forces will be considered.

IPROPF = 0 implies propulsion forces will not be considered.

SPECIAL INSTRUCTIONS: Punch a 0 or a 1 in column 3 of the data card.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

NDECK:

NDECK is an integer variable equal to the number of data decks present at run time.

SPECIAL INSTRUCTIONS: This variable goes in front of the first data deck only. If NDECK has a value less than 10, punch the integer in column 3 of the data card. If NDECK has a value greater than 9, punch the integer in columns 2 and 3.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

NGAIN:

NGAIN is an integer variable and refers to the number of arbitrary gains which are input. If NGAIN = 3 then values for GAIN(1), GAIN(2) and GAIN(3) will be input. The maximum value of NGAIN is 10.

SPECIAL INSTRUCTIONS: If NGAIN has a value less than 10, punch the integer in column 3 of the data card. Otherwise, use columns 2 and 3.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

NUMCMG:

NUMCMG is the number of controllers aboard body 0. A maximum of 6 controllers may be used.

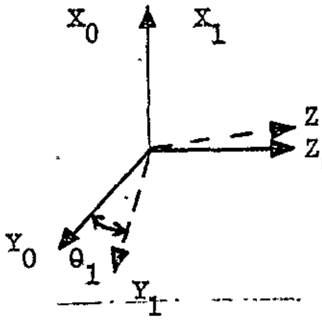
SPECIAL INSTRUCTIONS: Punch the integer in column 3 of the data card.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

OMEGA1:

OMEGA1 is the initial relative angular velocity measure about the spin axis between bodies 0 and 1. Another definition of OMEGA1 can be visualized by referring to the sketch shown below showing the orthogonal coordinate systems located on bodies 0 and 1.



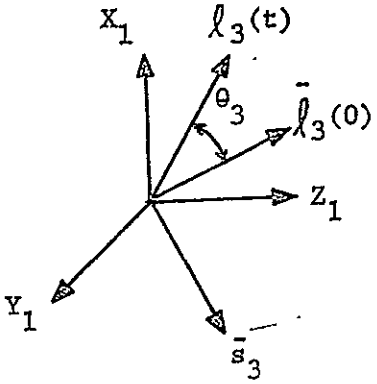
Implicit in this sketch is the hinge line between the stator and roto: is aligned parallel to both the $\overline{X}_0$ and $\overline{X}_1$ axes. Therefore, the orientation of the stator and the rotor differ only in a rotation θ_1 = THETA1 and OMEGA1 = $\dot{\theta}_1$.

UNITS: (radians per second)

FORMAT: # 5006 = (1X, F11.5)

OMEGA3:

OMEGA3 is the angular velocity of body 3 about the hinge line $\bar{s}_3$. OMEGA3 is also the time derivative of THETA3 when the datum for the angle THETA3 will be the $\overline{Y}_1$, $\overline{Z}_1$ plane as illustrated below:



In other words, $\bar{s}_3$ determines the positive direction of rotation by the right hand rule and the positive Y_1 - Z_1 plane determines to starting position.

UNITS: (radians per second)

FORMAT: # 5006 = (1X, F11.5)

INITIAL VALUE AT TIME = T_{start}

OMEGA4:

OMEGA4 is the angular velocity of body 4 about the hinge line $\bar{s}_4$. OMEGA4 is also the time derivative of THETA4 where the datum for the angle THETA4 is the same as THETA3. (See OMEGA3)

UNITS: (radians per second)

FORMAT: # 5006 = (1X, F11.5)

INITIAL VALUE AT TIME = T_{start}

PENDЗL:

PEND3L is the scalar distance from the hinge line, $\bar{s}_3$, to the center of mass of body 3. (i.e. the length of pendulum 3)

UNITS: (feet)

FORMAT: # 5006 = (1X, F11.5)

PEND4L:

PEND4L is the scalar distance from the hinge line, $\bar{s}_4$, to the center of mass of body 4. (i.e. the length of pendulum 4)

UNITS: (feet)

FORMAT: # 5006 = (1X, F11.5)

S:

S is a scalar parameter used in defining the position of body 2. S defines the magnitude of displacement of body 2 from the zero position.

Note: Punch the value of S in columns 2 through 12.

UNITS: (feet)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

INITIAL VALUE AT TIME = T_{start}

SDOT:

SDOT defines the magnitude of the velocity vector of body 2.

Note: Punch the value of SDOT in columns 15 through 25.

UNITS: (feet per second)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

INITIAL VALUE AT TIME = T_{start}

SP:

SP is the desired spin magnitude of body 1 relative to body 0.

UNITS: (radians per second)

FORMAT: # 5006 = (1X, F11.5)

S2(M):

S2(1), S2(2), and S2(3) are the X, Y, and Z components of a unit vector defining the direction of travel of body 2. $\bar{s}_2$ is expressed in body 1 coordinates.

UNITS: (none)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

S3(M):

S3(1), S3(2), and S3(3) are the X, Y, and Z components of a unit vector which defines the hinge line of body 3. $\bar{s}_3$ is expressed in body 1 coordinates. S3(1) must always be zero.

UNITS: (none)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

S4(M):

S4(1), S4(2), and S4(3) are the X, Y, and Z components of a unit vector which defines the hinge line of body 4. $\bar{s}_4$ is expressed in body 1 coordinates. S4(1) must always be zero.

UNITS: (none)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

THATA(J):

THAT(J) is the gimbal angle of the Jth controller aboard body 0 assuming this controller is either a one or two degree of freedom control moment gyro. If it is a two degree of freedom CMG, then this variable refers to the inner gimbal angle. J may have a maximum value of 6.

UNITS: (radians)

FORMAT: # 5006 = (1X, F11.5)

INITIAL VALUE AT TIME = T_{start}

THATAD(J):

THATAD(J) is the gimbal rate associated with THATA(J).

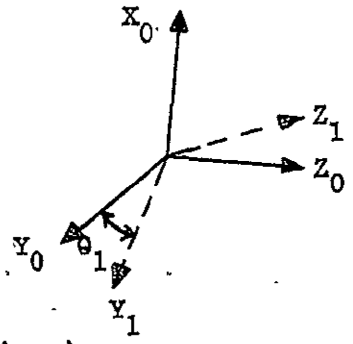
UNITS: (radians per second)

FORMAT: # 5006 = (1X, F11.5)

INITIAL VALUE AT TIME = T_{start}

THETA1:

THETA1 is the relative angular displacement measured about the spin axis, between bodies 0 and 1.



UNITS: (radians)

FORMAT: # 5006 = (1X, F11.5)

INITIAL VALUE AT TIME = T_{start}

THETA3:

THETA3 is the angle between pendulum 3 and the Y_-Z_- plane. The axis about which THETA3 rotates is the hinge line s_3. For an explanation of the THETA3 datum refer to the discussion of OMEGA3.

UNITS: (radians)

FORMAT: # 5006 = (1X, F11.5)

INITIAL VALUE AT TIME = T_{start}

THETA4:

THETA4 is the angle between pendulum 4 and the Y_-Z_- plane. The axis about which THETA4 rotates is the hinge line s_4. The datum for THETA4 is the same as for THETA3.

UNITS: (radians)

FORMAT: # 5006 = (1X, F11.5)

INITIAL VALUE AT TIME = T_{start}

TIBOI(M,N):

$$[\text{TIBOI}(M,N)] = \begin{bmatrix} \text{TIBOI}(1,1) & \text{TIBOI}(1,2) & \text{TIBOI}(1,3) \\ \text{TIBOI}(2,1) & \text{TIBOI}(2,2) & \text{TIBOI}(2,3) \\ \text{TIBOI}(3,1) & \text{TIBOI}(3,2) & \text{TIBOI}(3,3) \end{bmatrix}$$

This array is the initial transformation matrix of the body 0 coordinate system to the inertial coordinate system. If initially body 0 is aligned with the inertial system, TIBOI(M,N) would be the identity matrix. Refer to the coordinate transformation appendix for further details.

UNITS: (none)

FORMAT: # 5006 = (1X, F11.5)

INITIAL VALUES AT TIME = T_{start}

TSTART:

TSTART is the time at which you wish the program to start calculating the equations of motion. Except for restarting, TSTART is usually set to zero.

SPECIAL INSTRUCTIONS: Punch the value of TSTART in columns 2 through 12.

UNITS: (seconds)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

TSTOP:

TSTOP is the time at which you wish the program to stop calculating.

SPECIAL INSTRUCTIONS: Punch the value of TSTOP in columns 15 through 25.

UNITS: (seconds)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

WO(M):

WO(1), WO(2), and WO(3) are the X, Y, and Z components of the angular velocity vector of body 0.

UNITS: (radians per second)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

INITIAL VALUES AT TIME = T_{start}

ALT:

ALT is altitude of the center of mass of the space station configuration measured from the surface of the earth. All orbits are constrained to be circular with no oblateness effects. Therefore, altitude is the only pertinent orbit parameter.

UNITS: (miles)

FORMAT: # 5004 = (1X, E11.4)

BDMASS:

BDMASS is the sum of the mass of body 0, the stator, and the mass of the docking vehicle. The docking vehicle is constrained to dock on body 0, the stator. BDMASS = m_D

UNITS: (slugs)

FORMAT: # 5004 = (1X, E11.4)

BODYDI(M,N):

$$[\text{BODYDI}(M,N)] = [I_D] = \begin{bmatrix} \text{BODYDI}(1,1) & \text{BODYDI}(1,2) & \text{BODYDI}(1,3) \\ \text{BODYDI}(2,1) & \text{BODYDI}(2,2) & \text{BODYDI}(2,3) \\ \text{BODYDI}(3,1) & \text{BODYDI}(3,2) & \text{BODYDI}(3,3) \end{bmatrix}$$

This array is the inertia matrix of the docked body which consists of the docking vehicle connected to the stator.

UNITS: (slug-ft^2)

FORMAT: # 5004 = (1X, E11.4)

CA(M):

CA(1), CA(2), and CA(3) are the three direction cosines of the desired attitude reference direction in the inertial frame.

UNITS: (None)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

CP1 and CP2:

CP1 and CP2 are parameters used in determining the stiffness and the resonant frequency of the pendulums. For example, the control laws which are governed by the parameters are:

T13 = - CP1 * OMEGA3 - CP2 * THETA3

T14 = - CP1 * OMEGA4 - CP2 * (THETA4 - π)

Note: These variables are read on separate data cards.

UNITS: (none)

FORMAT: # 5004 = (1X, E11.4)

DD01(M):

DD01(1), DD01(2), and DD01(3) are the X, Y, ani Z components of the vector from the center of mass of the docked body to the hinge line of body 1, the rotor. The vector is expressed in body 0 coordinates. The docked body shall be defined as the configuration of the docking vehicle attached to body 0.

UNITS: (feet)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

DELTAT:

DELTAT = Δt

DELTAT is the time increment used in the integration algorithm which solves the rotational equations of motion. When "TIME" is updated we have:

TIME = TIME + DELTAT

Note: Punch the value of DELTAT in columns 28 through 38.

UNITS: (seconds)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

DTI(M):

DTI(1), DTI(2) and DTI(3) are the X, Y, and Z components of the docking torque impulse.

UNITS: lb-ft-sec

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

DTIME:

DTIME is the time when docking occurs. (i.e. "Docking Time"). The restrictions placed on DTIME are as follows:

a) $T_{start} <$ DTIME $< T_{stop}$

b) DTIME must be an integer multiple of Δt.

UNITS: (seconds)

FORMAT: # 5006 = (1X, F11.5)

GAIN(M):

GAIN(M) is an array of numbers the dimension of which is determined by another input variable NGAIN. These numbers, once read in, are stored in common and can be used for a variety of purposes. In many CMG control laws it is necessary to have control gains. GAIN(M) can be used for this purpose among others.

UNITS: (None)

FORMAT: # 5004 = (1X, E11.4)

IATTIF:

IATTIF is the attitude flag.

IATTIF = 1 implies an attempt to change attitude will be made.

IATTIF = 0 implies an attempt to change attitude will not be made.

SPECIAL INSTRUCTIONS: Punch a 0 or a 1 in column 3 of the data card.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

IB2F:

IB2F is the body 2 flag. (i.e. the elevator flag)

IB2F = 1 implies body 2 will be present.

IB2F = 0 implies body 2 will not be present.

SPECIAL INSTRUCTIONS: Punch a 0 or a 1 in column 3 of the data card.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

IDOCK:

IDOCK is the docking flag.

IDOCK = 1 implies docking will occur.

IDOCK = 0 implies docking will not occur.

SPECIAL INSTRUCTIONS: Punch a 0 or a 1 in column 3 of the data card.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

IGRAVF:

IGRAVF is the gravity gradient flag.

IGRAVF = 1 implies gravity gradient torques will be present.

IGRAVF = 0 implies gravity gradient torques will not be present.

SPECIAL INSTRUCTIONS: Punch a 0 or a 1 in column 3 of the data card.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

IPNDLM:

IPNDLM is the pendulum flag.

IPNDLM = 1 implies body 3 and body 4, the pendulums, will be present.

IPNDLM = 0 implies body 3 and body 4, the pendulums, will not be present.

SPECIAL INSTRUCTIONS: Punch a 0 or a 1 in column 3 of the data card.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

IPRINT:

IPRINT is an integer variable used to determine how often output data is printed. Data is printed in time increments of IPRINT * DELTAT. For example, if IPRINT = 50 and DELTAT = .2, then data will be printed when time = 10, 20, 30, 40, 50, ...

SPECIAL INSTRUCTIONS: If IPRINT has a value less than 10, punch the integer in column 3 of the data card. If IPRINT has a value greater than 9, punch the integer in columns 2 and 3.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

IPROPF:

IPROPF is the propulsion flag.

IPROPF = 1 implies propulsion forces will be considered.

IPROPF = 0 implies propulsion forces will not be considered.

SPECIAL INSTRUCTIONS: Punch a 0 or a 1 in column 3 of the data card.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

NDECK:

NDECK is an integer variable equal to the number of data decks present at run time.

SPECIAL INSTRUCTIONS: This variable goes in front of the first data deck only. If NDECK has a value less than 10, punch the integer in column 3 of the data card. If NDECK has a value greater than 9, punch the integer in columns 2 and 3.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

NGAIN:

NGAIN is an integer variable and refers to the number of arbitrary gains which are input. If NGAIN = 3 then values for GAIN(1), GAIN(2) and GAIN(3) will be input. The maximum value of NGAIN is 10.

SPECIAL INSTRUCTIONS: If NGAIN has a value less than 10, punch the integer in column 3 of the data card. Otherwise, use columns 2 and 3.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

TSTART:

TSTART is the time at which you wish the program to start calculating the equations of motion. Except for restarting, TSTART is usually set to zero.

SPECIAL INSTRUCTIONS: Punch the value of TSTART in columns 2 through 12.

UNITS: (seconds)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

TSTOP:

TSTOP is the time at which you wish the program to stop calculating.

SPECIAL INSTRUCTIONS: Punch the value of TSTOP in columns 15 through 25.

UNITS: (seconds)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

BODY 0 INPUT VARIABLES

[AII(J,M,N)]:

$$[AII(J,M,N)] = [II_J] = \begin{bmatrix} AII(J,1,1) & AII(J,1,2) & AII(J,1,3) \\ AII(J,2,1) & AII(J,2,2) & AII(J,2,3) \\ AII(J,3,1) & AII(J,3,2) & AII(J,3,3) \end{bmatrix}$$

This array is the inertia matrix for the inner gimbal of the Jth control moment gyro aboard the stator. A maximum of six CMGs may be used. (i.e. J = 1,6) All CMGs are also constrained to be located at the center of mass on the stator, body 0. If the Jth CMG has one or two degrees of freedom, it will have an inner gimbal. (Refer to the write up on CMGs for further discussion.)

UNITS: (slug-ft^2)

FORMAT: # 5006 = (1X, F11.5)

AIO(J,M,N):

$$[AIO(J,M,N)] = [IO_J] = \begin{bmatrix} AIO(J,1,1) & AIO(J,1,2) & AIO(J,1,3) \\ AIO(J,2,1) & AIO(J,2,2) & AIO(J,2,3) \\ AIO(J,3,1) & AIO(J,3,2) & AIO(J,3,3) \end{bmatrix}$$

This array is the inertia matrix for the outer gimbal of the Jth control moment gyro aboard the stator. A maximum of six CMGs may be used. (i.e. J = 1,6) All CMGs are constrained to be located at the center of mass on the stator, body 0. The Jth CMG will have an outer gimbal only if it has two degrees of freedom. (Refer to the write up on CMGs for further discussion.)

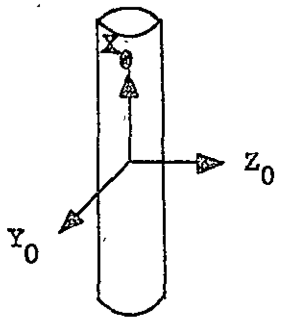
UNITS: (slug-ft^2)

FORMAT: # 5006 = (1X, F11.5)

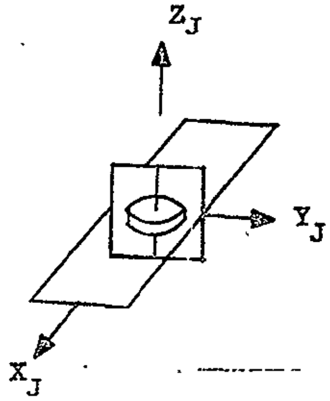
AOCJ(J,M,N):

$$[AOCJ(J,M,N)] = [0,C_J] = \begin{bmatrix} AOCJ(J,1,1) & AOCJ(J,1,2) & AOCJ(J,1,3) \\ AOCJ(J,2,1) & AOCJ(J,2,2) & AOCJ(J,2,3) \\ AOCJ(J,3,1) & AOCJ(J,3,2) & AOCJ(J,3,3) \end{bmatrix}$$

This array is the coordinate transformation matrix from the CMG null gimbal coordinate frame to the coordinate frame of body 0, the stator. (Refer to the coordinate transformation appendix.) For example, consider the two coordinate systems shown below:



Body 0 Frame



Jth Null Gimbal Frame

For the above situation:

$$[AOCJ(J,M,N)] = \begin{bmatrix} 0. & 0. & 1. \\ 1. & 0. & 0. \\ 0. & 1. & 0. \end{bmatrix}$$

UNITS: (None)

FORMAT: # 5006 = (1X, F11.5)

AOJ(M):

AOJ(1) is the distance between the jets of the pure couple producing the torque around the X axis of body 0.

AOJ(2) is the distance between the jets of the pure couple producing the torque around the Y axis of body 0.

AOJ(3) is the distance between the jets of the pure couple producing the torque around the Z axis of body 0.

Note: The variables CGAINO(1), CGAINO(2), and CGAINO(3) are also read on the same data cards as AOJ(1), AOJ(2), and AOJ(3). Punch the values of AOJ(1), AOJ(2), and AOJ(3) in columns 2 through 12 of the data cards.

UNITS: (feet)

FORMAT: # 5008 = (1X, E11.4, 2X, E11.4)

BODYOI(M,N):

$$[\text{BODYOI}(M,N)] = [I_0] = \begin{bmatrix} \text{BODYOI}(1,1) & \text{BODYOI}(1,2) & \text{BODYOI}(1,3) \\ \text{BODYOI}(2,1) & \text{BODYOI}(2,2) & \text{BODYOI}(2,3) \\ \text{BODYOI}(3,1) & \text{BODYOI}(3,2) & \text{BODYOI}(3,3) \end{bmatrix}$$

This array is the inertia matrix of body 0, the stator.

UNITS: (slug-ft^2)

FORMAT: # 5004 = (1X, E11.4)

BOMASS:

BOMASS is the mass of body 0, the stator. BOMASS = m_0

UNITS: (slugs)

FORMAT: # 5004 = (1X, E11.4)

CGAINO(M):

CGAINO(1) equals the control gain of all four reaction jets in the pure force couples producing torque around the X axis of body 0.

CGAINO(2) equals the control gain of all four reaction jets in the pure force couples producing torque around the Y axis of body 0.

CGAINO(3) equals the control gain of all four reaction jets in the pure force couples producing torque around the Z axis of body 0.

Note: The variables AOJ(1), AOJ(2), and AOJ(3) are also read in on the same data cards as CGAINO(1), CGAINO(2), and CGAINO(3). Punch the values of CGAINO(1), CGAINO(2), and CGAINO(3) in columns 15 through 25 of the data cards.

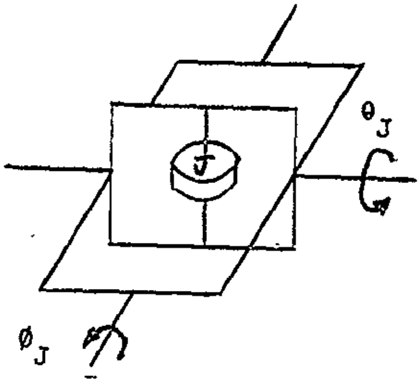
UNITS: (lb/radians/second)

FORMAT: # 5008 = (1X, E11.4, 2X, E11.4)

FEE(J):

FEE(J) = $\emptyset_J$

FEE(J) is the outer gimbal angle of a two degree of freedom control moment gyro as shown pictorially below:



The subscript J refers to the number assigned to the CMG. (Refer to the write up on CMGs for further discussion.)

UNITS: (radians)

FORMAT: # 5006 = (1X, F11.5)

INITIAL VALUE AT TIME = T_{start}

FEED(J):

FEED(J) = $\dot{\emptyset}_J$

FEED(J) is the outer gimbal rate of a two degree of freedom control moment gyro. The subscript J refers to the number assigned to the CMG. (Refer to the write up on CMGs for further discussion.)

UNITS: (radians/second)

FORMAT: # 5006 = (1X, F11.5)

HW(J):

HW(J) is the angular momentum of the wheel associated with the Jth momentum device. (Refer to the write up on CMGs for further discussion.)

UNITS: (slug-ft^2/second)

FORMAT: # 5004 = (1X, E11.4)

IDOF(J):

IDOF(J) = 0 implies the Jth controller aboard body 0 is a reaction wheel.

IDOF(J) = 1 implies the Jth controller aboard body 0 is a one degree of freedom control moment gyro.

IDOF(J) = 2 implies the Jth controller aboard body 0 is a two degree of freedom control moment gyro.

SPECIAL INSTRUCTIONS: Punch a 0, 1, or a 2 in column 3 of the data card.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

NUMCMG:

NUMCMG is the number of controllers aboard body 0. A maximum of 6 controllers may be used.

SPECIAL INSTRUCTIONS: Punch the integer in column 3 of the data card.

UNITS: (none)

FORMAT: # 5000 = (1X, I2)

THATA(J):

THAT(J) is the gimbal angle of the Jth controller aboard body 0 assuming this controller is either a one or two degree of freedom control moment gyro. If it is a two degree of freedom CMG, then this variable refers to the inner gimbal angle. J may have a maximum value of 6.

UNITS: (radians)

FORMAT: # 5006 = (1X, F11.5)

INITIAL VALUE AT TIME = T_{start}

THATAD(J):

THATAD(J) is the gimbal rate associated with THATA(J).

UNITS: (radians per second)

FORMAT: # 5006 = (1X, F11.5)

INITIAL VALUE AT TIME = T_{start}

TIBOI(M,N):

$$\left[\text{TIBOI}(M,N)\right] = \begin{bmatrix} \text{TIBOI}(1,1) & \text{TIBOI}(1,2) & \text{TIBOI}(1,3) \\ \text{TIBOI}(2,1) & \text{TIBOI}(2,2) & \text{TIBOI}(2,3) \\ \text{TIBOI}(3,1) & \text{TIBOI}(3,2) & \text{TIBOI}(3,3) \end{bmatrix}$$

This array is the initial transformation matrix of the body 0 coordinate system to the inertial coordinate system. If initially body 0 is aligned with the inertial system, TIBOI(M,N) would be the identity matrix. Refer to the coordinate transformation appendix for further details.

UNITS: (none)

FORMAT: # 5006 = (1X, F11.5)

INITIAL VALUES AT TIME = T_{start}

WO(M):

WO(1), WO(2), and WO(3) are the X, Y, and Z components of the angular velocity vector of body 0.

UNITS: (radians per second)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

INITIAL VALUES AT TIME = T_{start}

BODY 1 INPUT VARIABLES

A1J(M):

A1J(1) is the distance between the jets of the pure couple producing the torque around the X axis of body 1.

A1J(2) is the distance between the jets of the pure couple producing the torque around the Y axis of body 1.

Note: The variables CGAIN1(1) and CGAIN1(2) are also read on the same data cards as A1J(1) and A1J(2). Punch the values of A1J(1) and A1J(2) in columns 2 through 12 of the data cards.

UNITS: (feet)

FORMAT: # 5008 = (1X, E11.4, 2X, E11.4)

BODY1I(M,N):

$$[\mathrm{BODY1I}(M,N)] = [I_1] = \begin{bmatrix} \mathrm{BODY1I}(1,1) & \mathrm{BODY1I}(1,2) & \mathrm{BODY1I}(1,3) \\ \mathrm{BODY1I}(2,1) & \mathrm{BODY1I}(2,2) & \mathrm{BODY1I}(2,3) \\ \mathrm{BODY1I}(3,1) & \mathrm{BODY1I}(3,2) & \mathrm{BODY1I}(3,3) \end{bmatrix}$$

This array is the inertia matrix for body 1, the rotor.

UNITS: (slug-ft^2)

FORMAT: # 5004 = (1X, E11.4)

B1MASS:

B1MASS is the mass of body 1, the rotor. B1MASS = m_1

UNITS: (slugs)

FORMAT: # 5004 = (1X, E11.4)

CGAIN1(M):

CGAIN1(1) equals the control gain of all four reaction jets in the pure force couples producing torque around the X axis of body 1.

CGAIN1(2) equals the control gain of all four reaction jets in the pure force couples producing torque around the Y axis of body 1.

Note: The variables A1J(1) and A1J(2) are also read in on the same date cards as CGAIN1(1) and CGAIN1(2). Punch the values of CGAIN1(1) and CGAIN1(2) in columns 15 through 25 of the data cards.

UNITS: (lb/radians/second)

FORMAT: # 5008 = (1X, E11.4, 2X, E11.4)

D01(M):

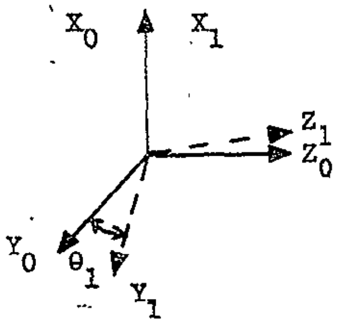
D01(1), D01(2), and D01(3) are the X, Y, and Z components of the vector from the center of mass of body 0 to the hinge line of body 1, the rotor. The vector is expressed in body 0 coordinates. D01(1) should in most cases be zero.

UNITS: (feet)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

OMEGA1:

OMEGA1 is the initial relative angular velocity measure about the spin axis between bodies 0 and 1. Another definition of OMEGA1 can be visualized by referring to the sketch shown below showing the orthogonal coordinate systems located on bodies 0 and 1.



Imp icit in this sketch is the hinge line between the stator and roto· is aligned parallel to both the $\overline{X}_0$ and $\overline{X}_1$ axes. Therefore, th·: orientation of the stator and the rotor differ only in a rot ıtion θ_1 = THETA1 and OMEGA1 = $\dot{\theta}_1$.

UNI :S: (radians per second)

FOR·IAT: # 5006 = (1X, F11.5)

SP:

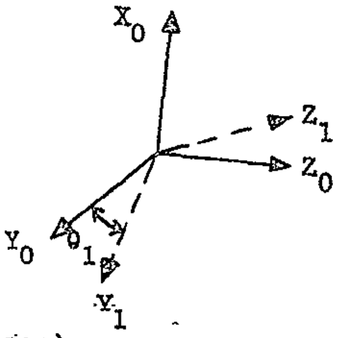
SP is the desired spin magnitude of body 1 relative to body 0.

UNI ΓS: (radians per second)

FORMAT: # 5006 = (1X, F11.5)

THETA1:

THETA1 is the relative angular displacement measured about the spin axis, between bodies 0 and 1.



UNITS: (radians)

FORMAT: # 5006= (1X, F11.5)

INITIAL VALUE AT TIME = T_{start}

BODY 2 INPUT VARIABLES

B2MASS:

B2MASS is the mass of body 2, the elevator. B2MASS = m_2

UNITS: (slugs)

FORMAT: # 5004 = (1X, E11.4)

D12(M):

D12(1), D12(2) and D12(3) are the X, Y and Z components of a fixed vector locating the starting position of the movable mass, the elevator. The vector equation that describes the motion of the movable mass is $\bar{\ell}_2 = \bar{d}_{12} + \bar{s}_2 s$. Where $\bar{s}_2$ is a unit vector which defines the direction in which the movable mass travels and s is a scalar prespecified function of time. For most cases $s(t_o = 0) = 0$ so that $\bar{d}_{12}$ specifies the initial starting position of the movable mass. The elevator is known alternately as the movable mass or body 2. D12 is expressed in body 1 coordinates. For example, if the elevator were constrained to travel along the X axis, then $\bar{d}_{12}$ could have the following values: D12(1) = 1., D12(2) = 0., D12(3) = 0.

UNITS: (feet)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

S:

S is a scalar parameter used in defining the position of body 2. S defines the magnitude of displacement of body 2 from the zero position.

Note: Punch the value of S in columns 2 through 12.

UNITS: (feet)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

INITIAL VALUE AT TIME = T_{start}

SDOT:

SDOT defines the magnitude of the velocity vector of body 2.

Note: Punch the value of SDOT in columns 15 through 25.

UNITS: (feet per second)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

INITIAL VALUE AT TIME = T_{start}

S2(M):

S2(1), S2(2), and S2(3) are the X, Y, and Z components of a unit vector defining the direction of travel of body 2. $\bar{s}_2$ is expressed in body 1 coordinates.

UNITS: (none)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

BODY 3 INPUT VARIABLES

B3MASS:

B3MASS is the mass of body 3, a pendulum. B3MASS = m_3

UNITS: (slugs)

FORMAT: # 5004 = (1X, E11.4)

D13(M):

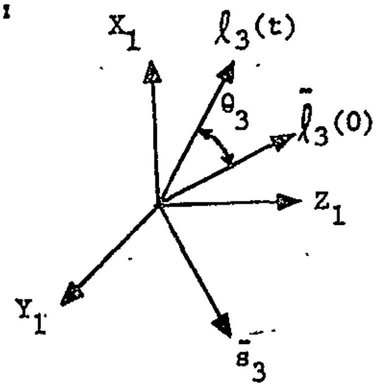
D13(1), D13(2), and D13(3) are the X, Y, and Z components of the vector from the center of mass of body 1 to the hinge line of body 3. $\overline{D13}$ is expressed in body 1 coordinates.

UNITS: (feet)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

OMEGA3:

OMEGA3 is the angular velocity of body 3 about the hinge line $\bar{s}_3$. OMEGA3 is also the time derivative of THETA3 when the datum for the angle THETA3 will be the $\bar{Y}_1$, $\bar{Z}_1$ plane as illustrated below:



In other words, $\bar{s}_3$ determines the positive direction of rotation by the right hand rule and the positive Y_1 - Z_1 plane determines to starting position.

UNITS: (radians per second)

FORMAT: # 5006 = (1X, F11.5)

INITIAL VALUE AT TIME = T_{start}

PEND3L:

PEND3L is the scalar distance from the hinge line, $\bar{s}_3$, to the center of mass of body 3. (i.e. the length of pendulum 3)

UNITS: (feet)

FORMAT: # 5006 = (1X, F11.5)

S3(M):

S3(1), S3(2), and S3(3) are the X, Y, and Z components of a unit vector which defines the hinge line of body 3. $\bar{s}_3$ is expressed in body 1 coordinates. S3(1) must always be zero.

UNITS: (none)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

THETA3:

THETA3 is the angle between pendulum 3 and the Y_1 Z_1 plane. The axis about which THETA3 rotates is the hinge line $\bar{s}_3$. For an explanation of the THETA3 datum refer to the discussion of OMEGA3.

UNITS: (radians)

FORMAT: # 5006 = (1X, F11.5)

INITIAL VALUE AT TIME = T_{start}

BODY 4 INPUT VARIABLES

B4MASS:

B4MASS is the mass of body 4, a pendulum. B4MASS = m_4

UNITS: (slugs)

FORMAT: # 5004 = (1X, E11.4)

D14(M):

D14(1), D14(2), and D14(3) are the X, Y, and Z components of the vector from the center of mass of body 1 to the hinge line of body 4.

D14 is expressed in body 1 coordinates.

UNITS: (feet)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

OMEGA4:

OMEGA4 is the angular velocity of body 4 about the hinge line $\bar{s}_4$. OMEGA4 is also the time derivative of THETA4 where the datum for the angle THETA4 is the same as THETA3. (See OMEGA3)

UNITS: (radians per second)

FORMAT: # 5006 = (1X, F11.5)

INITIAL VALUE AT TIME = T_{start}

PEND4L:

PEND4L is the scalar distance from the hinge line, $\bar{s}_4$, to the center of mass of body 4. (i.e. the length of pendulum 4)

UNITS: (feet)

FORMAT: # 5006 = (1X, F11.5)

S4(M):

S4(1), S4(2), and S4(3) are the X, Y, and Z components of a unit vector which defines the hinge line of body 4. $\bar{s}_4$ is expressed in body 1 coordinates. S4(1) must always be zero.

UNITS: (none)

FORMAT: # 5002 = (1X, 3(F11.5, 2X))

THETA4:

THETA4 is the angle between pendulum 4 and the Y, Z, plane. The axis about which THETA4 rotates is the hinge line $\bar{s}_4$. The datum for THETA4 is the same as for THETA3.

UNITS: (radians)

FORMAT: # 5006 = (1X, F11.5)

INITIAL VALUE AT TIME = T_{start}

APPENDIX B, COORDINATE TRANSFORMATIONS

The following pages contain a pictorial guide to aid the user in computing the initial transformation matrix from one right hand orthogonal coordinate system to another. The transformation matrix from coordinate system B to coordinate A shall be denoted [A,B].

Mathematically:
$$\begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix} = [A,B] \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix}$$

i.e.
$$X_A = AB(1,1) * X_B + AB(1,2) * Y_B + AB(1,3) * Z_B$$
$$Y_A = AB(2,1) * X_B + AB(2,2) * Y_B + AB(2,3) * Z_B$$
$$Z_A = AB(3,1) * X_B + AB(3,2) * Y_B + AB(3,3) * Z_B$$

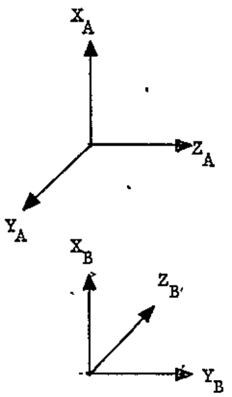
For simplicity, the coordinate systems in the following illustrations are orthogonal to each other in one way or another. i.e. There are not small offsetting rotations. Hence, the components of A,B may assume only certain values. The components may be ± 1., 0., ± sin∅, ± cos∅, ± sinθ, ± cosθ, ± sinψ, or ± cosψ where ∅, θ, and ψ are angles of rotation about the X, Y, Z axes respectively.

FLOW CHART & BLOCK DIAGRAM

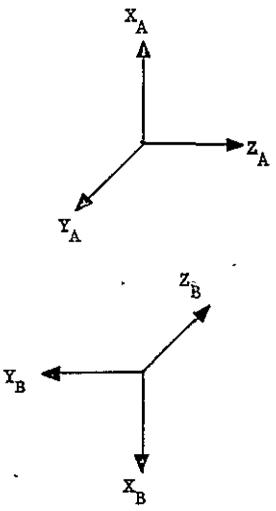
FORM DEN 1103-01 (4-64)

Application ______________________ Date ____________ Page ____ of ____

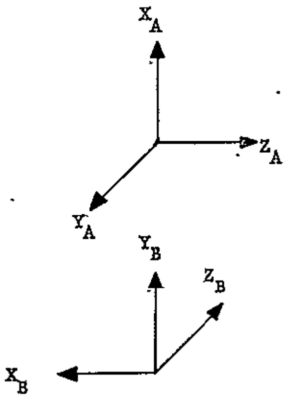
Procedure ______________________ Drawn By ____________



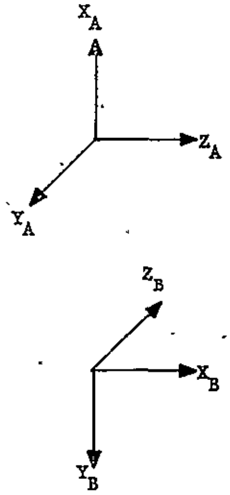
$$\left[A, B\right] = \begin{bmatrix}1. & 0. & 0.\\ 0. & 0. & -1.\\ 0. & 1. & 0.\end{bmatrix}$$



$$\left[A, B\right] = \begin{bmatrix}-1. & 0. & 0.\\ 0. & 0. & -1.\\ 0. & -1. & 0.\end{bmatrix}$$



$$\left[A, B\right] = \begin{bmatrix}0. & 1. & 0.\\ 0. & 0. & -1.\\ -1. & 0. & 0.\end{bmatrix}$$

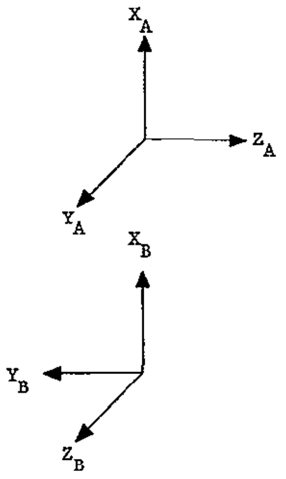


$$\left[A, B\right] = \begin{bmatrix}0. & -1. & 0.\\ 0. & 0. & -1.\\ 1. & 0. & 0.\end{bmatrix}$$

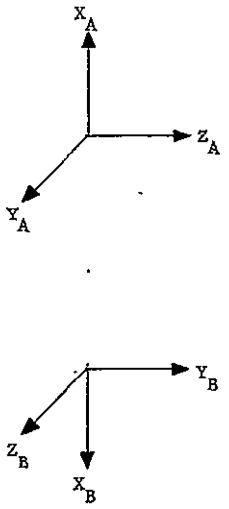
FLOW CHART & BLOCK DIAGRAM

Application ______________________ Date ______________ Page ____ of ____

Procedure ______________________ Drawn By ______________________

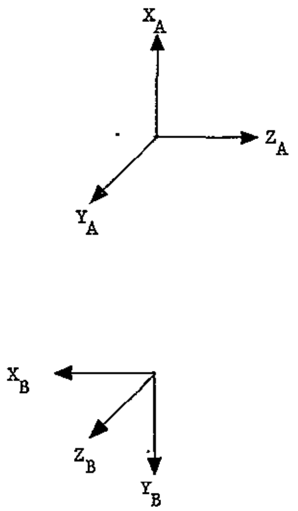


$$\left[A, B\right] = \begin{bmatrix} 1. & 0. & 0. \\ 0. & 0. & 1. \\ 0. & -1. & 0. \end{bmatrix}$$

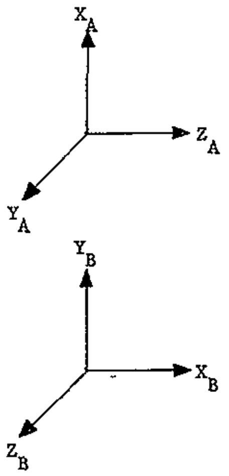


$$\left[A, B\right] = \begin{bmatrix} -1. & 0. & 0. \\ 0. & 0. & 1. \\ 0. & 1. & 0. \end{bmatrix}$$

Fold



$$\left[A, B\right] = \begin{bmatrix} 0. & -1. & 0. \\ 0. & 0. & 1. \\ -1. & 0. & 0. \end{bmatrix}$$



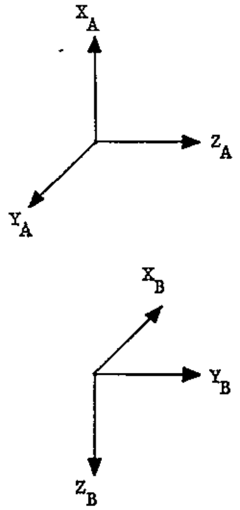
$$\left[A, B\right] = \begin{bmatrix} 0. & 1. & 0. \\ 0. & 0. & 1. \\ 1. & 0. & 0. \end{bmatrix}$$

FLOW CHART & BLOCK DIAGRAM

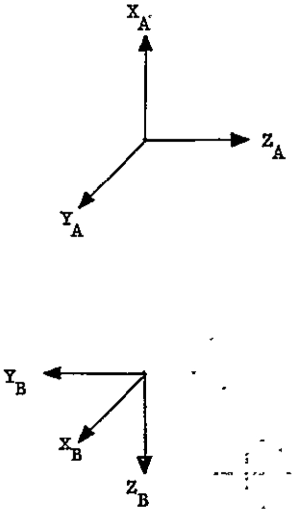
Form DEN 1103-01 (4-6-)

Application ______________________ Date ______________ Page ____ of ____

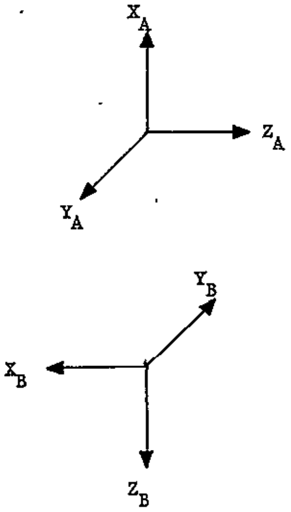
Procedure ______________________ Drawn By ______________



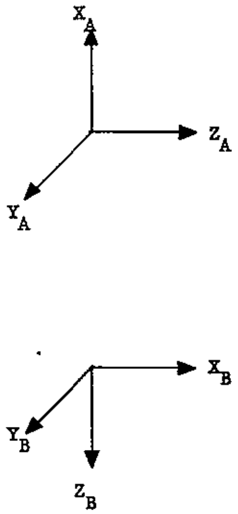
$$\begin{bmatrix} A, & B \end{bmatrix} = \begin{bmatrix} 0. & 0. & -1. \\ -1. & 0. & 0. \\ 0. & 1. & 0. \end{bmatrix}$$



$$\begin{bmatrix} A, & B \end{bmatrix} = \begin{bmatrix} 0. & 0. & -1. \\ 1. & 0. & 0. \\ 0. & -1. & 0. \end{bmatrix}$$



$$\begin{bmatrix} A, & B \end{bmatrix} = \begin{bmatrix} 0. & 0. & -1. \\ 0. & -1. & 0. \\ -1. & 0. & 0. \end{bmatrix}$$



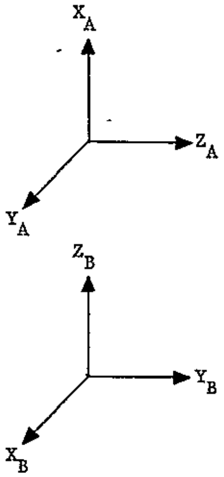
$$\begin{bmatrix} A, & B \end{bmatrix} = \begin{bmatrix} 0. & 0. & -1. \\ 0. & 1. & 0. \\ 1. & 0. & 0. \end{bmatrix}$$

← Fold to dotted line

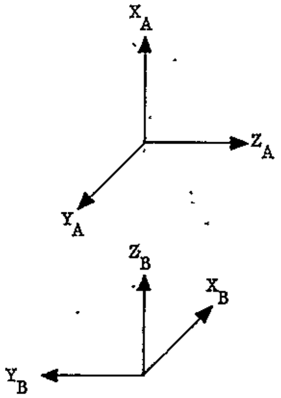
FLOW CHART & BLOCK DIAGRAM

Application ______________________ Date ____________ Page ______ of ______

Procedure ______________________ Drawn By ______________________

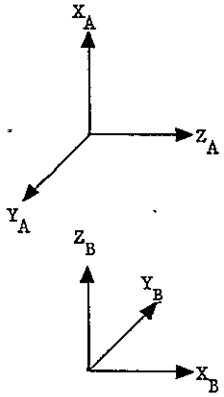


$$[A, B] = \begin{bmatrix} 0. & 0. & 1. \\ 1. & 0. & 0. \\ 0. & 1. & 0. \end{bmatrix}$$

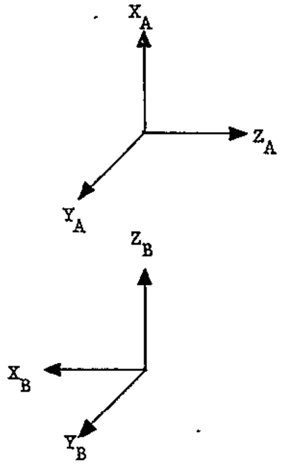


$$[A, B] = \begin{bmatrix} 0. & 0. & 1. \\ -1. & 0. & 0. \\ 0. & -1. & 0. \end{bmatrix}$$

Fold



$$[A, B] = \begin{bmatrix} 0. & 0. & 1. \\ 0. & -1. & 0. \\ 1. & 0. & 0. \end{bmatrix}$$



$$[A, B] = \begin{bmatrix} 0. & 0. & 1. \\ 0. & 1. & 0. \\ -1. & 0. & 0. \end{bmatrix}$$

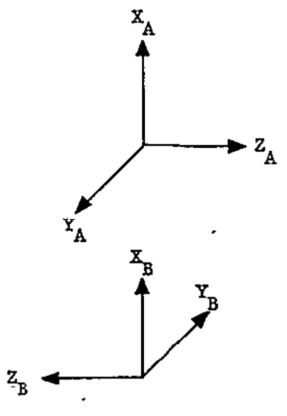
Fold to dotted line

FLOW CHART & BLOCK DIAGRAM

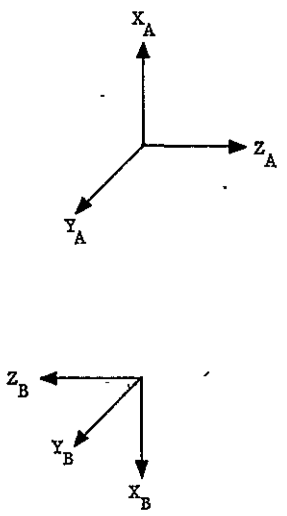
FORM DEN 1103-01 (4-64)

Application ______________________ Date ______________ Page _____ of _____

Procedure ______________________ Drawn By ______________________

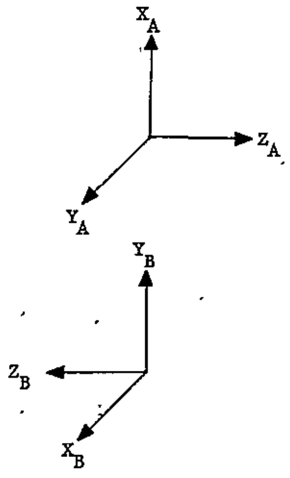


$$[A, B] = \begin{bmatrix} 1. & 0. & 0. \\ 0. & -1. & 0. \\ 0. & 0. & -1. \end{bmatrix}$$

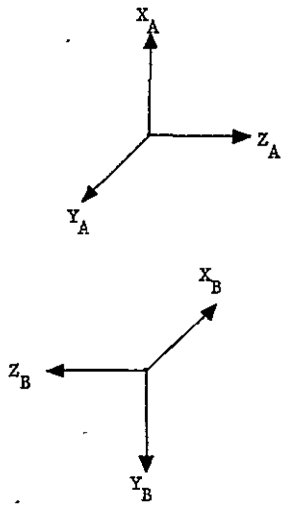


$$[A, B] = \begin{bmatrix} -1. & 0. & 0. \\ 0. & 1. & 0. \\ 0. & 0. & -1. \end{bmatrix}$$

Fold



$$[A, B] = \begin{bmatrix} 0. & 1. & 0. \\ 1. & 0. & 0. \\ 0. & 0. & -1. \end{bmatrix}$$



$$[A, B] = \begin{bmatrix} 0. & -1. & 0. \\ -1. & 0. & 0. \\ 0. & 0. & -1. \end{bmatrix}$$

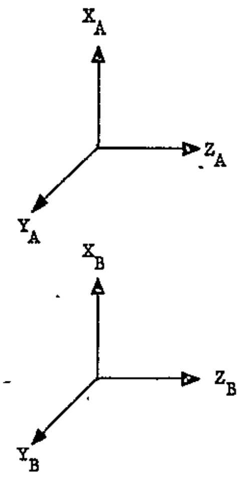
← Fold to dotted line

FLOW CHART & BLOCK DIAGRAM

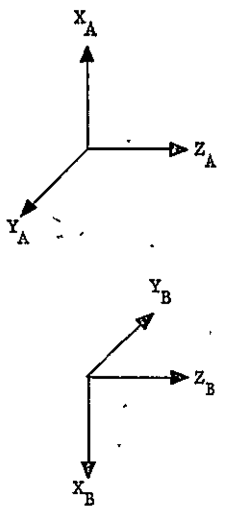
FORM DEN 1103-01 (4-64)

Application ______________________ Date ____________ Page ____ of ____

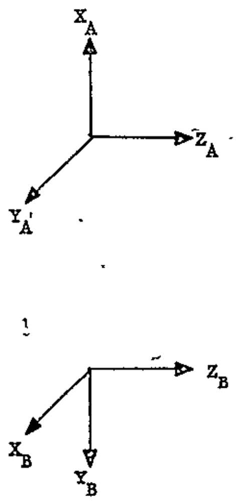
Procedure ______________________ Drawn By ____________



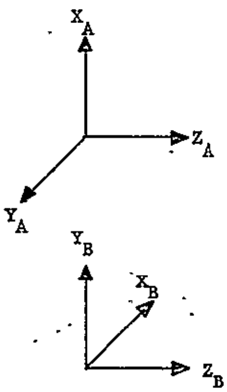
$$[A, B] = \begin{bmatrix} 1. & 0. & 0. \\ 0. & 1. & 0. \\ 0. & 0. & 1. \end{bmatrix}$$



$$[A, B] = \begin{bmatrix} -1. & 0. & 0. \\ 0. & -1. & 0. \\ 0. & 0. & 1. \end{bmatrix}$$



$$[A, B] = \begin{bmatrix} 0. & -1. & 0. \\ 1. & 0. & 0. \\ 0. & 0. & 1. \end{bmatrix}$$



$$[A, B] = \begin{bmatrix} 0. & 1. & 0. \\ -1. & 0. & 0. \\ 0. & 0. & 1. \end{bmatrix}$$

```
CHARGE, 03 MD246  2070000603744524 8947 56665     P,HENDRICKS2284
MAP.
RUN24(P,,,,,,377777,,1)
LGO.
EXIT.
DMP.
DMP(0,150000)

      PROGRAM MD246(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,FILMPL)
C     *
C     *
C     *
C     ********************************************************************
      COMMON      A(3)          ,     AE(5)         ,     AED(5)        , COMMO
     *            AFOUR(2)      ,     AII(6,3,3)    ,                     COMMO
     *            AIO(6,3,3)    ,     AJ1(3)        ,     ALT           , COMMO
     *            AOCJ(6,3,3)   ,     AOJ(3)        ,     AONE(7)       , COMMO
     *            ATCPT2(3,3)   ,     ATHREE(5)     ,     ATWO(4)       , COMMO
     *            A1(3,3)       ,     A1J(2)                              COMMO
      COMMON      BDMASS        ,     BFOUR(2)      ,     BMOM          , COMMO
     *            BODYDI(3,3)   ,     BODYOI(3,3)   ,     BODY1I(3,3)   , COMMO
     *            BOMASS        ,     BONE(7)       ,     BTHREE(5)     , COMMO
     *            BTWO(4)       ,     B1MASS        ,     B2MASS        , COMMO
     *            B3MASS        ,     B4MASS                              COMMO
      COMMON      CA(3)         ,     CB(3)         ,     CGAIN0(3)     , COMMO
     *            CGAIN1(2)     ,                                         COMMO
     *            COSFEJ        ,     COSTTJ        ,     COSTT0        , COMMO
     *            COSTT1        ,     COSTT3        ,     COSTT4        , COMMO
     *            CO2T          ,     CP1           ,     CP2           , COMMO
     *            CST           ,     C1                                  COMMO
      COMMON      DB(3)         ,     DDO1(3)       ,                     COMMO
     *            DELTAT        ,     DO1(3)        ,     DO1DOT(3)     , COMMO
     *            DTI(3)        ,                                         COMMO
     *            DTIME         ,     D12(3)        ,     D13(3)        , COMMO
     *            D13DOT(3)     ,     D13YCS        ,     D13YSN        , COMMO
     *            D13ZCS        ,     D13ZSN        ,     D14(3)        , COMMO
     *            D14DOT(3)     ,     D14YCS        ,     D14YSN        , COMMO
     *            D14ZCS        ,     D14ZSN                              COMMO
      COMMON      EEE(3,3)      ,     EEJ(3,3)      ,     EL2(3)        , COMMO
     *            EL2DOT(3)     ,     EL2YCS        ,     EL2YSN        , COMMO
     *            EL2ZCS        ,     EL2ZSN        ,     EL3(3)        , COMMO
     *            EL3DOT(3)     ,     EL3YCS        ,     EL3YSN        , COMMO
     *            EL3ZCS        ,     EL3ZSN        ,     EL4(3)        , COMMO
     *            EL4DOT(3)     ,     EL4YCS        ,     EL4YSN        , COMMO
     *            EL4ZCS        ,     EL4ZSN        ,     EMI(6,6)        COMMO
      COMMON      FAT(8)        ,                                         COMMO
     *            FEE(6)        ,     FEED(6)       ,     FFF(3)        , COMMO
     *            FFJ(3)        ,     FLAG1         ,     FLAG2         , COMMO
     *            FLAG3         ,     FLAG4         ,     FN            , COMMO
     *            FO(3)         ,     FO1(3)        ,     FO2(3)        , COMMO
     *            FO3(3)        ,     F1(3)         ,     F11(3)        , COMMO
     *            FPT(5)        ,                                         COMMO
     *            F12(3)        ,     F13(3)                              COMMO
      COMMON      GAIN(10)      ,     G3            ,                     COMMO
     *            G3DOT         ,     G4            ,     G4DOT           COMMO
      COMMON      H(3)          ,     HCMG(3)       ,     HDOT(3)       , COMMO
     *            HI(3)         ,     HO(3)         ,     HW(6)         , COMMO
     *            H1(3)         ,     H1PDOT(3)     ,     H1PRIM(3)     , COMMO
     *          H3PRIM(3)     ,      H4PRIM(3)                                COMMC
      COMMON    IB2F          ,      ICFA          ,      ICFB          ,     COMMC
     *          ICFC          ,      ICFD          ,      IDOCK         ,     COMMC
     *          IDOF(6)       ,                                             COMMC
     *          IGRAVF        ,      IPNDLM        ,      IPNTCK        ,     COMMC
     *          IPRINT        ,      IPROPF                                 COMMC
      COMMON    NCASE         ,      NCHECK        ,      NDECK         ,     COMMC
     *          NGAIN         ,      NUMCMG                                 COMMC
      COMMON    OMEGA1        ,      OMEGA3        ,      OMEGA4              COMMC
      COMMON    PEND3L        ,      PEND4L                                 COMMC
      COMMON    Q(4,4)                                                      COMMC
      COMMON    R             ,      R0(3)         ,      R1(3)         ,     COMMC
     *          R1DOT(3)      ,      R1YCS         ,      R1YSN         ,     COMMC
     *          R1ZCS         ,      R1ZSN         ,      R2(3)         ,     COMMC
     *          R2DOT(3)      ,      R2YCS         ,      R2YSN         ,     COMMC
     *          R2ZCS         ,      R2ZSN         ,      R3(3)         ,     COMMC
     *          R3DOT(3)      ,      R3YCS         ,      R3YSN         ,     COMMC
     *          R3ZCS         ,      R3ZSN         ,      R4(3)         ,     COMMC
     *          R4DOT(3)      ,      R4YCS         ,      R4YSN         ,     COMMC
     *          R4ZCS         ,      R4ZSN                                  COMMC
      COMMON    S             ,      SDOT          ,      SINFEJ        ,     COMMC
     *          SINTTJ        ,      SINTT0        ,      SINTT1        ,     COMMC
     *          SINTT2        ,      SINTT3        ,      SINTT4        ,     COMMC
     *          SP            ,      SUM1          ,      SUM2          ,     COMMC
     *          SUM3          ,      S2(3)         ,      S3(3)         ,     COMMC
     *          S4(3)                                                       COMMC
      COMMON    T(3,3)        ,      TC(3,3)       ,      TEMP1(3)      ,     COMMC
     *          TEMP2(3)      ,                                             COMMC
     *          TEMP3(3)      ,      TEMP4(3)      ,      TEMP5(3,3)    ,     COMMC
     *          TEMP6(3,3)    ,      TEMP7(3,3)    ,      TEMP8(3,3)    ,     COMMC
     *          TEMP9(3,3)    ,      TEMP10(3,3)   ,      TEMP11(3,3)   ,     COMMC
     *          TEMP12(3,3)   ,      TEMP13(3,3)   ,      TEMP14(3,3)   ,     COMMC
     *          TEMP15(3,3)   ,      TERM1(3)      ,      TERM2(3)      ,     COMMC
     *          TFRICT        ,      THATA(6)      ,      THATAD(6)     ,     COMMC
     *          THETA1        ,      THETA3        ,      THETA4        ,     COMMC
     *          THET0         ,      TIB0(3,3)     ,      TIB0I(3,3)    ,     COMMC
     *          TIME          ,      TJ            ,      TJ1(10)       ,     COMMC
     *          TJ2(10)       ,      TJ3(10)       ,      TJ4(10)       ,     COMMC
     *          TMOTOR        ,                                             COMMC
     *          T0EF(3)       ,      TOTMAS        ,      T01           ,     COMMC
     *          TQ0G(3)       ,      TQ0P(3)       ,      TQ1G(3)       ,     COMMC
     *          TQ1P(3)       ,      TSTART        ,      TSTOP         ,     COMMC
     *          TT1DOT        ,      TT3DOT        ,      TT4DOT        ,     COMMC
     *          T1EF(3)       ,      T13           ,      T14                 COMMC
      COMMON    V(3)                                                        COMMC
      COMMON    W0(3)         ,      WS            ,      W1(3)         ,     COMMC
     *          W3(3)         ,      W4(3)                                  COMMC
      COMMON    X(6,7)        ,      XC            ,      XCDOT         ,     COMMC
     *          XMU                                                         COMMC
C                                                                           COMMC
C     ************************************************************************
C     *
C     *
C     *
C     ************************************************************************
C                                                                            RCON
C     THIS IS THE ENTRY POINT TO THE RUN CONTROL MODULE.                     RCON
C     THE FUNCTION OF THIS MODULE IS TO MAKE THE DECISION TO STOP ALL        RCON
C     CALCULATIONS AND EXIT THE PROGRAM OR CONTINUE TO THE INPUT MODULE      RCON
C     AND READ IN THE DATA FOR THE NEXT CASE.                                RCON
C                                                                            RCON
      NCHECK = 0                                                             RCON
```

```
C     READ IN THE NUMBER OF DATA DECKS PRESENT AT RUN TIME.                RCON
      READ (5,5000) NDECK                                                  RCON
C     PRINT NDECK ON A NEW SHEET OF PAPER.                                 RCON
      WRITE(6,6000) NDECK                                                  RCON
C     THE PROGRAM RETURNS TO THE FOLLOWING STATEMENT NUMBER AFTER EACH     RCON
C     DATA DECK HAS BEEN COMPLETELY PROCESSED.                             RCON
   10 NCHECK = NCHECK + 1                                                  RCON
      IF (NDECK .GE. NCHECK) GO TO 20                                      RCON
      STOP                                                                 RCON
   20 CONTINUE                                                             RCON
C                                                                          RCON
C     ************************************************************************
C     *
C     *
C     *
C     ************************************************************************
C                                                                          INPU
C     THIS IS THE ENTRY POINT TO THE INPUT MODULE.                         INPU
C     THE FUNCTION OF THIS MODULE IS TO READ IN ALL DATA PERTAINING TO     INPU
C     THE NEXT CASE.  AFTER EACH VARIABLE HAS BEEN READ IN, IT WILL BE     INPU
C     PRINTED OUT TO INSURE PROPER CONVERSION AND TO RETAIN A RECORD OF    INPU
C     THE INPUT DATA.                                                      INPU
C                                                                          INPU
      DO 25 M=1,10                                                         INPU
      GAIN(M) = 0.                                                         INPU
   25 CONTINUE                                                             INPU
      WRITE(6,6500) NCHECK                                                 INPU
C     READ THE PENDULUM FLAG.                                              INPU
      READ (5,5000) IPNDLM                                                 INPU
      WRITE(6,6082) IPNDLM                                                 INPU
C     READ THE PRINT FLAG.  (I.E.  PRINT EVERY IPRINT TIME POINT.)         INPU
      READ (5,5000) IPRINT                                                 INPU
      WRITE(6,6544) IPRINT                                                 INPU
C     READ THE STARTING TIME, STOPING TIME, AND DELTAT.                    INPU
      READ (5,5002) TSTART,TSTOP,DELTAT                                    INPU
      WRITE(6,6002) TSTART,TSTOP,DELTAT                                    INPU
C     READ THE ORBIT ALTITUDE.                                             INPU
      READ (5,5004) ALT                                                    INPU
      WRITE(6,6004) ALT                                                    INPU
C     READ THE TRANSFORMATION FROM THE BODY 0 FRAME TO THE I FRAME.        INPU
      DO 30 M=1,3                                                          INPU
      DO 30 N=1,3                                                          INPU
      READ (5,5006)     TIBOI(M,N)                                         INPU
   30 CONTINUE                                                             INPU
      DO 35 M=1,3                                                          INPU
      WRITE(6,6006) M,TIBOI(M,1),M,TIBOI(M,2),M,TIBOI(M,3)                 INPU
   35 CONTINUE                                                             INPU
C     READ BODY 0 ANGULAR RATES.                                           INPU
      READ (5,5002) WO(1),WO(2),WO(3)                                      INPU
      WRITE(6,6008) WO(1),WO(2),WO(3)                                      INPU
C     READ THE MASS OF BODY 0.                                             INPU
      READ (5,5004)    BOMASS                                              INPU
      J = 0                                                                INPU
      WRITE(6,6010) J,BOMASS                                               INPU
C     READ THE INERTIA MATRIX FOR BODY 0.                                  INPU
      DO 40 M=1,3                                                          INPU
      DO 40 N=1,3                                                          INPU
      READ (5,5004)     BODYOI(M,N)                                        INPU
   40 CONTINUE                                                             INPU
      DO 45 M=1,3                                                          INPU
      WRITE(6,6012) M,BODYOI(M,1),M,BODYOI(M,2),M,BODYOI(M,3)              INPU
   45 CONTINUE                                                             INPU
```

```
C     READ THE NUMBER OF CONTROL MOMENT GYROS ABOARD BODY 0.              INPUT
      READ (5,5000) NUMCMG                                                INPUT
      WRITE(6,6014) NUMCMG                                                INPUT
      IF (NUMCMG .EQ. 0) GO TO 120                                        INPUT
      DO 110 J=1,NUMCMG                                                   INPUT
C     READ THE DEGREE OF FREEDOM OF THE JTH CMG.                          INPUT
      READ (5,5000) IDOF(J)                                               INPUT
      IF (IDOF(J) .NE. 0) GO TO 50                                        INPUT
      WRITE(6,6016) J                                                     INPUT
      GO TO 70                                                            INPUT
   50 IF (IDOF(J) .NE. 1) GO TO 60                                        INPUT
      WRITE(6,6018) J                                                     INPUT
      GO TO 70                                                            INPUT
   60 WRITE(6,6020) J                                                     INPUT
   70 CONTINUE                                                            INPUT
C     READ THE ANGULAR MOMENTUM OF THE JTH CMG.                           INPUT
      READ (5,5004)   HW(J)                                               INPUT
      WRITE(6,6022) J,HW(J)                                               INPUT
C     READ THE TRANSFORMATION FROM THE JTH NULL GIMBAL FRAME TO BODY 0    INPUT
C     FRAME.                                                              INPUT
      DO 80 M=1,3                                                         INPUT
      DO 80 N=1,3                                                         INPUT
      READ (5,5006)       AOCJ(J,M,N)                                     INPUT
   80 CONTINUE                                                            INPUT
      DO 85 M=1,3                                                         INPUT
      WRITE(6,6024) J,M,AOCJ(J,M,1),J,M,AOCJ(J,M,2),J,M,AOCU(J,M,3)       INPUT
   85 CONTINUE                                                            INPUT
      IF (IDOF(J) .EQ. 0) GO TO 110                                       INPUT
C     READ THE INERTIA MATRIX FOR THE JTH INNER GIMBAL.                   INPUT
      DO 90 M=1,3                                                         INPUT
      DO 90 N=1,3                                                         INPUT
      READ (5,5006)       AII(J,M,N)                                      INPUT
   90 CONTINUE                                                            INPUT
      DO 95 M=1,3                                                         INPUT
      WRITE(6,6026) J,M,AII(J,M,1),J,M,AII(J,M,2),J,M,AII(J,M,3)          INPUT
   95 CONTINUE                                                            INPUT
C     READ THE INNER GIMBAL ANGLE AND RATE OF THE JTH CMG.                INPUT
      READ (5,5006) THATA(J)                                              INPUT
      READ (5,5006) THATAD(J)                                             INPUT
      WRITE(6,6028) J,THATA(J),J,THATAD(J)                                INPUT
      FEE(J) = 0.                                                         INPUT
      FEED(J) = 0.                                                        INPUT
      IF (IDOF(J) .EQ. 1) GO TO 110                                       INPUT
C     READ THE INERTIA MATRIX FOR THE JTH OUTER GIMBAL.                   INPUT
      DO 100 M=1,3                                                        INPUT
      DO 100 N=1,3                                                        INPUT
      READ (5,5006)       AIO(J,M,N)                                      INPUT
  100 CONTINUE                                                            INPUT
      DO 105 M=1,3                                                        INPUT
      WRITE(6,6030) J,M,AIO(J,M,1),J,M,AIO(J,M,2),J,M,AIO(J,M,3)          INPUT
  105 CONTINUE                                                            INPU
C     READ THE OUTER GIMBAL ANGLE AND RATE OF THE JTH CMG.                INPU
      READ (5,5006) FEE(J)                                                INPU
      READ (5,5006) FEED(J)                                               INPU
      WRITE(6,6032) J,FEE(J),J,FEED(J)                                    INPU
  110 CONTINUE                                                            INPU
  120 CONTINUE                                                            INPU
C     READ THE PROPULSION FLAG.                                           INPU
      READ (5,5000) IPROPF                                                INPU
      WRITE(6,6072) IPROPF                                                INPU
      IF (IPROPF .EQ. 0) GO TO 140                                        INPU
      READ (5,5000) IATTIF                                                INPU
      WRITE(6,6562) IATTIF                                              INPUT
      IF (IATTIF .EQ. 0) GO TO 125                                      INPUT
      READ (5,5002) CA(1),CA(2),CA(3)                                   INPUT
      WRITE(6,6564) CA(1),CA(2),CA(3)                                   INPUT
  125 CONTINUE                                                          INPUT
      DO 130 J=1,3                                                      INPUT
      READ (5,5008)    AOJ(J),  CGAINO(J)                               INPUT
      WRITE(6,6074) J,AOJ(J),J,CGAINO(J)                                INPUT
  130 CONTINUE                                                          INPUT
  140 CONTINUE                                                          INPUT
C     READ IN VARIABLES RELATED TO BODY 1.                              INPUT
C     READ THE MASS OF BODY 1.                                          INPUT
      READ (5,5004) B1MASS                                              INPUT
      J = 1                                                             INPUT
      WRITE(6,6010) J,B1MASS                                            INPUT
C     READ THE INERTIA MATRIX FOR BODY 1.                               INPUT
      DO 150 M=1,3                                                      INPUT
      DO 150 N=1,3                                                      INPUT
      READ (5,5004)       BODY1I(M,N)                                   INPUT
  150 CONTINUE                                                          INPUT
      DO 153 M=1,3                                                      INPUT
      WRITE(6,6040) M,BODY1I(M,1),M,BODY1I(M,2),M,BODY1I(M,3)           INPUT
  153 CONTINUE                                                          INPUT
C     READ THE PRIMARY GIMBAL ANGLE AND RATE OF BODY 1 W.R.T. BODY 0.   INPUT
      READ (5,5006) THETA1                                              INPUT
      READ (5,5006) OMEGA1                                              INPUT
      WRITE(6,6042) THETA1,OMEGA1                                       INPUT
C     READ THE VECTOR FROM THE CM OF BODY 0 TO THE HINGE POINT BETWEEN  INPUT
C     BODY 0 AND BODY 1.  (BODY 0 COORDINATES)                          INPUT
      READ (5,5002) D01(1),D01(2),D01(3)                                INPUT
      WRITE(6,6044) D01(1),D01(2),D01(3)                                INPUT
C     READ THE BODY 2 FLAG.  (I.E.  THE ELEVATOR FLAG.)                 INPUT
      READ (5,5000) IB2F                                                INPUT
      WRITE(6,6546) IB2F                                                INPUT
      IF (IB2F .EQ. 0) GO TO 155                                        INPUT
C     READ THE MASS OF THE ELEVATOR.                                    INPUT
      READ (5,5004)    B2MASS                                           INPUT
      J = 2                                                             INPUT
      WRITE(6,6010) J,B2MASS                                            INPUT
      READ (5,5002) D12(1),D12(2),D12(3)                                INPUT
      WRITE(6,6535) D12(1),D12(2),D12(3)                                INPUT
      READ (5,5002)    S2(1),  S2(2),  S2(3)                            INPUT
      J = 2                                                             INPUT
      WRITE(6,6052) J,S2(1),J,S2(2),J,S2(3)                             INPUT
C     READ THE POSITION AND VELOCITY OF THE ELEVATOR.                   INPUT
      READ (5,5002) S,SDOT                                              INPUT
      WRITE(6,6548) S,SDOT                                              INPUT
      GO TO 157                                                         INPUT
  155 B2MASS = 0.                                                       INPUT
      D12(1) = 0.                                                       INPUT
      D12(2) = 0.                                                       INPUT
      D12(2) = 0.                                                       INPUT
      S2(1) = 0.                                                        INPUT
      S2(2) = 0.                                                        INPUT
      S2(3) = 0.                                                        INPUT
      SDOT = 0.                                                         INPUT
      S = 0.                                                            INPUT
  157 CONTINUE                                                          INPUT
      IF (IPROPF .EQ. 0) GO TO 170                                      INPUT
      DO 160 J=1,2                                                      INPUT
      READ (5,5008)    A1J(J),  CGAIN1(J)                               INPUT
      WRITE(6,6076) J,A1J(J),J,CGAIN1(J)                                INPUT
  160 CONTINUE                                                          INPUT
  170 CONTINUE                                                          INPUT
      IF (IPNDLM .EQ. 0) GO TO 172                                      INPUT
      READ (5,5004) CP1                                                 INPUT
      WRITE(6,6582) CP1                                                 INPUT
      READ (5,5004) CP2                                                 INPUT
      WRITE(6,6584) CP2                                                 INPUT
C     READ THE VARIABLES ASSOCIATED WITH BODY 3.  (I.E. PENDULUM 3)     INPUT
C     READ THE MASS OF BODY 3.                                          INPUT
      READ (5,5004)   B3MASS                                            INPUT
      J = 3                                                             INPUT
      WRITE(6,6010) J,B3MASS                                            INPUT
C     READ THE VECTOR FROM THE CM OF BODY 1 TO THE HINGE POINT BETWEEN  INPUT
C     BODY 1 AND PENDULUM 3.  (BODY 1 COORDINATES)                      INPUT
      READ (5,5002) D13(1),D13(2),D13(3)                                INPUT
      WRITE(6,6048) D13(1),D13(2),D13(3)                                INPUT
      READ (5,5002)   S3(1),  S3(2),  S3(3)                             INPUT
      J = 3                                                             INPUT
      WRITE(6,6052) J,S3(1),J,S3(2),J,S3(3)                             INPUT
C     READ THE LENGTH OF BODY 3.  (I.E. PENDULUM 3)                     INPUT
      READ (5,5006) PEND3L                                              INPUT
      WRITE(6,6058) PEND3L                                              INPUT
C     READ THE PRIMARY GIMBAL ANGLE OF BODY 3 W.R.T. BODY 1.            INPUT
      READ (5,5006) THETA3                                              INPUT
      WRITE(6,6060) THETA3                                              INPUT
C     READ THE PRIMARY GIMBAL RATE OF BODY 3 W.R.T. BODY 1.             INPUT
      READ (5,5006) OMEGA3                                              INPUT
      WRITE(6,6062) OMEGA3                                              INPUT
C     READ THE VARIABLES ASSOCIATED WITH BODY 4.  (I.E. PENDULUM 4)     INPUT
C     READ THE MASS OF BODY 4.                                          INPUT
      READ (5,5004)   B4MASS                                            INPUT
      J = 4                                                             INPUT
      WRITE(6,6010) J,B4MASS                                            INPUT
C     READ THE VECTOR FROM THE CM OF BODY 1 TO THE HINGE POINT BETWEEN  INPUT
C     BODY 1 AND PENDULUM 4.  (BODY 1 COORDINATES)                      INPUT
      READ (5,5002) D14(1),D14(2),D14(3)                                INPUT
      WRITE(6,6050) D14(1),D14(2),D14(3)                                INPUT
      READ (5,5002)   S4(1),  S4(2),  S4(3)                             INPUT
      J = 4                                                             INPUT
      WRITE(6,6052) J,S4(1),J,S4(2),J,S4(3)                             INPUT
C     READ THE LENGTH OF BODY 4.  (I.E. PENDULUM 4)                     INPUT
      READ (5,5006) PEND4L                                              INPUT
      WRITE(6,6064) PEND4L                                              INPUT
C     READ THE PRIMARY GIMBAL ANGLE AND RATE OF BODY 4 W.R.T. BODY 1.   INPUT
      READ (5,5006) THETA4                                              INPUT
      READ (5,5006) OMEGA4                                              INPUT
      WRITE(6,6066) THETA4,OMEGA4                                       INPUT
      GO TO 174                                                         INPUT
  172 B3MASS = 0.                                                       INPUT
      CP1 = 0.                                                          INPUT
      CP2 = 0.                                                          INPUT
      D13(1) = 0.                                                       INPUT
      D13(2) = 0.                                                       INPUT
      D13(3) = 0.                                                       INPUT
      S3(1) = 0.                                                        INPUT
      S3(2) = 0.                                                        INPUT
      S3(3) = 0.                                                        INPUT
      PEND3L = 0.                                                       INPUT
      THETA3 = 0.                                                       INPUT
      OMEGA3 = 0.                                                       INPUT
      B4MASS = 0.                                                       INPUT
      D14(1) = 0.                                                       INPUT
      D14(2) = 0.                                                       INPUT
      D14(3) = 0.                                                       INPUT
      S4(1) = 0.                                                        INPUT
      S4(2) = 0.                                                        INPUT
      S4(3) = 0.                                                        INPUT
      PEND4L = 0.                                                       INPUT
      THETA4 = 0.                                                       INPUT
      OMEGA4 = 0.                                                       INPUT
  174 CONTINUE                                                          INPUT
      READ (5,5006) SP                                                  INPUT
      WRITE(6,6080) SP                                                  INPUT
      READ (5,5000) NGAIN                                               INPUT
      WRITE(6,6570) NGAIN                                               INPUT
      IF (NGAIN .EQ. 0) GO TO 176                                       INPUT
      DO 175 J=1,NGAIN                                                  INPUT
      READ (5,5004) GAIN(J)                                             INPUT
      WRITE(6,6572) J,GAIN(J)                                           INPUT
  175 CONTINUE                                                          INPUT
  176 CONTINUE                                                          INPUT
C     READ THE GRAVITY GRADIENT FLAG.                                   INPUT
      READ (5,5000) IGRAVF                                              INPUT
      WRITE(6,6078) IGRAVF                                              INPUT
C     READ THE DOCKING FLAG.  IDOCK = 1 IMPLIES A DOCKING WILL OCCUR.    INPUT
      READ (5,5000) IDOCK                                               INPUT
      WRITE(6,6068) IDOCK                                               INPUT
      IF (IDOCK .EQ. 0) GO TO 180                                       INPUT
C     READ THE TIME OF DOCKING.                                         INPUT
      READ (5,5006) DTIME                                               INPUT
      WRITE(6,6070) DTIME                                               INPUT
      DTMIN = DTIME - DELTAT/10.                                        INPUT
      DTMAX = DTIME + DELTAT/10.                                        INPUT
      DCHMIN = DTIME - 1.1*DELTAT                                       INPUT
      DCHMAX = DTIME - 0.9*DELTAT                                       INPUT
      READ (5,5004) BDMASS                                              INPUT
      WRITE(6,6574) BDMASS                                              INPUT
      DO 177 M=1,3                                                      INPUT
      DO 177 N=1,3                                                      INPUT
      READ (5,5004) BODYDI(M,N)                                         INPUT
  177 CONTINUE                                                          INPUT
      DO 178 M=1,3                                                      INPUT
      WRITE(6,6576) M,BODYDI(M,1),M,BODYDI(M,2),M,BODYDI(M,3)           INPUT
  178 CONTINUE                                                          INPUT
      READ (5,5002) DTI(1),DTI(2),DTI(3)                                INPUT
      WRITE(6,6578) DTI(1),DTI(2),DTI(3)                                INPUT
      READ (5,5002) DD01(1),DD01(2),DD01(3)                             INPUT
      WRITE(6,6586) DD01(1),DD01(2),DD01(3)                             INPUT
  180 CONTINUE                                                          INPUT
C                                                                       INPUT
C     *******************************************************************
C     *
C     *
C     *
C     *******************************************************************
C                                                                       INIT
C     THIS IS THE ENTRY POINT TO THE INITIALIZATION BLOCK.              INIT
C     ALL INITIAL CALCULATIONS ARE PERFORMED ONE TIME ONLY FOR EACH CASEINIT
C                                                                       INIT
C     CALCULATE THE TOTAL MASS OF THE CONFIGURATION.                    INIT
      TOTMAS = B0MASS + B1MASS + B2MASS + B3MASS + B4MASS               INIT
      ICFA = 0                                                          INIT
      ICFB = 0                                                          INIT
      ICFC = 0                                                          INIT
      ICFD = 0                                                          INIT
C     INITIALIZE THE FLAGS NEEDED TO START THE INTEGRATIONS.            INIT
      FLAG1 = 0.                                                        INIT
      FLAG2 = 0.                                                        INIT
      FLAG3 = 0.                                                        INIT
      FLAG4 = 0.                                                        INIT
C     INITIALIZE THE QUATERNION COMPONENTS.                             INIT
      AE(2) = 1.                                                        INIT
      AE(3) = 0.                                                        INIT
      AE(4) = 0.                                                        INIT
      AE(5) = 0.                                                        INIT
      AED(2) = 1.                                                       INIT
      AED(3) = 0.                                                       INIT
      AED(4) = 0.                                                       INIT
      AED(5) = 0.                                                       INIT
      ATWO(2) = THETA1                                                  INIT
      ATWO(3) = THETA3                                                  INIT
      ATWO(4) = THETA4                                                  INIT
      ATHREE(2) = 1.0                                                   INIT
      ATHREE(3) = 0.0                                                   INIT
      ATHREE(4) = 0.0                                                   INIT
      ATHREE(5) = 0.0                                                   INIT
C     INITIALIZE THE PRINT CHECK FLAG.                                  INIT
      IPNTCK = 1                                                        INIT
      TAP = 0.                                                          INIT
      TBP = 0.                                                          INIT
C     INITIALIZE TIME TO THE TSTART VALUE.                              INIT
      TIME = TSTART                                                     INIT
      XMU = 1.408E+16                                                   INIT
      R = ALT + 3960.                                                   INIT
      R = 5280.*R                                                       INIT
      C1 = XMU/R**3                                                     INIT
      WS = SQRT(C1)                                                     INIT
      EEE(1,1) = 0.                                                     INIT
      EEE(1,2) = 0.                                                     INIT
      EEE(1,3) = 0.                                                     INIT
      EEE(2,1) = 0.                                                     INIT
      EEE(2,2) = 0.                                                     INIT
      EEE(2,3) = 0.                                                     INIT
      EEE(3,1) = 0.                                                     INIT
      EEE(3,2) = 0.                                                     INIT
      EEE(3,3) = 0.                                                     INIT
      FFF(1) = 0.                                                       INIT
      FFF(2) = 0.                                                       INIT
      FFF(3) = 0.                                                       INIT
      F01(1) = 0.                                                       INIT
      F01(2) = 0.                                                       INIT
      F01(3) = 0.                                                       INIT
      F02(1)  = 0.                                                      INIT
      F02(2)  = 0.                                                      INIT
      F02(3)  = 0.                                                      INIT
      F0(1)  = 0.                                                       INIT
      F0(2)  = 0.                                                       INIT
      F0(3)  = 0.                                                       INIT
      F11(1) = 0.                                                       INIT
      F11(2) = 0.                                                       INIT
      F11(3) = 0.                                                       INIT
      F12(1)  = 0.                                                      INIT
      F12(2)  = 0.                                                      INIT
      F12(3)  = 0.                                                      INIT
      F1(1)  = 0.                                                       INIT
      F1(2)  = 0.                                                       INIT
      F1(3)  = 0.                                                       INIT
      TWOG(1) = 0.                                                      INIT
      TWOG(2) = 0.                                                      INIT
      TWOG(3) = 0.                                                      INIT
      TWOP(1) = 0.                                                      INIT
      TWOP(2) = 0.                                                      INIT
      TWOP(3) = 0.                                                      INIT
      TW1G(1) = 0.                                                      INIT
      TW1G(2) = 0.                                                      INIT
      TW1G(3) = 0.                                                      INIT
      TW1P(1) = 0.                                                      INIT
      TW1P(2) = 0.                                                      INIT
      TW1P(3) = 0.                                                      INIT
      IF (NUMCMG .EQ. 0) GO TO 184                                      INIT
      CALL CMG                                                          INIT
      DO 183 J=1,NUMCMG                                                 INIT
      THATA(J) = THATA(J) - THATAD(J)*DELTAT                            INIT
      FEE(J) = FEE(J) - FEED(J)*DELTAT                                  INIT
  183 CONTINUE                                                          INIT
  184 CONTINUE                                                          INIT
C     SUBROUTINE XDOT CALCULATES THE FOLLOWING VARIABLES,               INIT
C            A.   HDOT(1)                                               INIT
C            B.   HDOT(2)                                               INIT
C            C.   HDOT(3)                                               INIT
C            D.   H1PDOT(1)                                             INIT
C            E.   G3DOT                                                 INIT
C            F.   G4DOT                                                 INIT
      CALL XDOT                                                         INIT
C                                                                       INIT
C     *******************************************************************
C     *
C     *
C     *
C     *******************************************************************
C
  190 CONTINUE
C
C     *******************************************************************
C     *
C     *
C     *
C     *******************************************************************
C                                                                       INTEG
C     THIS IS THE ENTRY POINT TO INTEGRATION BLOCK ONE.                 INTEG
C     THE PURPOSE OF THE BLOCK IS TO CALL AN INTEGRATION SUBROUTINE TO  INTEG
C     INTEGRATE THE VARIABLES CALCULATED BY SUBROUTINE XDOT.            INTEG
C     THE ARRAYS USED BY THE INTEGRATION SUBROUTINE MUST BE CALCULATED  INTEG
C     EACH TIME BEFORE CALLING THE INTEGRATION SUBROUTINE.              INTEG
C                                                                       INTEG
      IF (TIME .NE. TSTART) GO TO 192                                   INTEG
      CALL EMCALC                                                       INTEG
      G3   = EM(5,1)*WO(1)   + EM(5,2)*WO(2)   + EM(5,3)*WO(3)          INTEG
     *       +EM(5,4)*OMEGA1 + EM(5,5)*OMEGA3 + EM(5,6)*OMEGA4          INTEG
      G4   = EM(6,1)*WO(1)   + EM(6,2)*WO(2)   + EM(6,3)*WO(3)          INTEG
     *       +EM(6,4)*OMEGA1 + EM(6,5)*OMEGA3 + EM(6,6)*OMEGA4          INTEG
  192 CONTINUE                                                          INTEG
      AONE(1) = TIME                                                    INTEG
      AONE(2) = H(1)                                                    INTEG
      AONE(3) = H(2)                                                    INTEG
      AONE(4) = H(3)                                                    INTEG
      AONE(5) = H1PRIM(1)                                               INTEG
      AONE(6) = G3                                                      INTEG
      AONE(7) = G4                                                      INTEG
      BONE(1) = DELTAT                                                  INTEG
      BONE(2) = HDOT(1)                                                 INTEG
      BONE(3) = HDOT(2)                                                 INTEG
      BONE(4) = HDOT(3)                                                 INTEG
      BONE(5) = H1PDOT(1)                                               INTEG
      BONE(6) = G3DOT                                                   INTEG
      BONE(7) = G4DOT                                                   INTEG
      CALL FOMS(AONE,BONE,7,FLAG1,TJ1)                                  INTEG
      H(1)       = AONE(2)                                              INTEG
      H(2)       = AONE(3)                                              INTEG
      H(3)       = AONE(4)                                              INTEG
      IF (IDOCK .EQ. 0) GO TO 193                                       INTEG
      IF ((TIME .LT. DCHMIN) .OR. (TIME .GT. DCHMAX)) GO TO 193         INTEG
      H(1) = H(1) + DTI(1)                                              INTEG
      H(2) = H(2) + DTI(2)                                              INTEG
      H(3) = H(3) + DTI(3)                                              INTEG
  193 CONTINUE                                                          INTEG
      H1PRIM(1) = AONE(5)                                               INTEG
      G3         = AONE(6)                                              INTEG
      G4         = AONE(7)                                              INTEG
C                                                                       INTEG
C     ******************************************************************
C     *
C     *
C     *
C     ******************************************************************
C
C     THIS SEGMENT SIMPLY CALCULATES THE GIMBAL ANGLE RATES OF BODIES
C     ONE, THREE, AND FOUR.  THAT IS TO SAY THE SUBSTITUTION NEEDED BY
C     INTEGRATION BLOCK TWO IS DONE AT THIS POINT.
C
      TT1DOT = OMEGA1
      TT3DOT = OMEGA3
      TT4DOT = OMEGA4
C
C     ******************************************************************
C     *
C     *
C     *
C     ******************************************************************
C                                                                       INTEG
C     THIS IS THE ENTRY POINT TO INTEGRATION BLOCK TWO.                 INTEG
C     THE PURPOSE OF THE BLOCK IS TO CALL AN INTEGRATION SUBROUTINE TO  INTEG
C     INTEGRATE THE GIMBAL ANGLE RATES OF BODIES ONE, THREE, AND FOUR   INTEG
C     TO PRODUCE THE CORRESPONDING GIMBAL ANGLE POSITIONS.  AS BEFORE,  INTEG
C     THE ARRAYS USED BY THE INTEGRATION SUBROUTINE MUST BE CALCULATED  INTEG
C     EACH TIME BEFORE CALLING THE INTEGRATION SUBROUTINE.              INTEG
C                                                                       INTEG
      ATWO(1) = TIME                                                    INTEG
      BTWO(1) = DELTAT                                                  INTEG
      BTWO(2) = TT1DOT                                                  INTEG
      BTWO(3) = TT3DOT                                                  INTEG
      BTWO(4) = TT4DOT                                                  INTEG
      CALL FOMS(ATWO,BTWO,4,FLAG2,TJ2)                                  INTEG
      THETA1 = ATWO(2)                                                  INTEG
      THETA3 = ATWO(3)                                                  INTEG
      THETA4 = ATWO(4)                                                  INTEG
C                                                                       INTEG
C     ******************************************************************
C     *
C     *
```

```
C      *
C      ************************************************************************
C                                                                       QUAT
C      THIS IS THE ENTRY POINT TO THE QUATERNION BLOCK.                 QUAT
C                                                                       QUAT
       Q(1,1) =  0.                                                     QUAT
       Q(1,2) = -0.5*WO(1)                                              QUAT
       Q(1,3) = -0.5*WO(2)                                              QUAT
       Q(1,4) = -0.5*WO(3)                                              QUAT
       Q(2,1) = -Q(1,2)                                                 QUAT
       Q(2,2) =  0.                                                     QUAT
       Q(2,3) = -Q(1,4)                                                 QUAT
       Q(2,4) =  Q(1,3)                                                 QUAT
       Q(3,1) = -Q(1,3)                                                 QUAT
       Q(3,2) =  Q(1,4)                                                 QUAT
       Q(3,3) =  0.                                                     QUAT
       Q(3,4) = -Q(1,2)                                                 QUAT
       Q(4,1) = -Q(1,4)                                                 QUAT
       Q(4,2) = -Q(1,3)                                                 QUAT
       Q(4,3) =  Q(1,2)                                                 QUAT
       Q(4,4) =  0.                                                     QUAT
       AED(2) = Q(1,1)*AE(2) + Q(1,2)*AE(3) + Q(1,3)*AE(4) + Q(1,4)*AE(5)QUAT
       AED(3) = Q(2,1)*AE(2) + Q(2,2)*AE(3) + Q(2,3)*AE(4) + Q(2,4)*AE(5)QUAT
       AED(4) = Q(3,1)*AE(2) + Q(3,2)*AE(3) + Q(3,3)*AE(4) + Q(3,4)*AE(5)QUAT
       AED(5) = Q(4,1)*AE(2) + Q(4,2)*AE(3) + Q(4,3)*AE(4) + Q(4,4)*AE(5)QUAT
C      SET UP MATRICES USED BY THE INTEGRATION SUBROUTINE               QUAT
       ATHREE(1) = TIME                                                 QUAT
       BTHREE(1) = DELTAT                                               QUAT
       BTHREE(2) = AED(2)                                               QUAT
       BTHREE(3) = AED(3)                                               QUAT
       BTHREE(4) = AED(4)                                               QUAT
       BTHREE(5) = AED(5)                                               QUAT
       CALL FOMS(ATHREE,BTHREE,5,FLAG3,TJ3)                             QUAT
       AE(2) = ATHREE(2)                                                QUAT
       AE(3) = ATHREE(3)                                                QUAT
       AE(4) = ATHREE(4)                                                QUAT
       AE(5) = ATHREE(5)                                                QUAT
C      CALCULATE THE NORMALIZING FACTOR.                                QUAT
       FN = SQRT( AE(2)**2 + AE(3)**2 + AE(4)**2 + AE(5)**2 )           QUAT
       AE(2) = AE(2)/FN                                                 QUAT
       AE(3) = AE(3)/FN                                                 QUAT
       AE(4) = AE(4)/FN                                                 QUAT
       AE(5) = AE(5)/FN                                                 QUAT
       T(1,1) = AE(2)**2 + AE(3)**2 - AE(4)**2 - AE(5)**2               QUAT
       T(1,2) = 2.*(AE(3)*AE(4) - AE(2)*AE(5))                          QUAT
       T(1,3) = 2.*(AE(3)*AE(5) + AE(2)*AE(4))                          QUAT
       T(2,1) = 2.*(AE(3)*AE(4) + AE(2)*AE(5))                          QUAT
       T(2,2) = AE(2)**2 - AE(3)**2 + AE(4)**2 - AE(5)**2               QUAT
       T(2,3) = 2.*(AE(4)*AE(5) - AE(2)*AE(3))                          QUAT
       T(3,1) = 2.*(AE(3)*AE(5) - AE(2)*AE(4))                          QUAT
       T(3,2) = 2.*(AE(4)*AE(5) + AE(2)*AE(3))                          QUAT
       T(3,3) = AE(2)**2 - AE(3)**2 - AE(4)**2 + AE(5)**2               QUAT
       CALL MULT(DUM,DUM,DUM,TIBO,TIBOI,T,2)                            QUAT
       CALL MULT(HI,TIBO,H,DUM,DUM,DUM,1)                               QUAT
C                                                                       QUAT
C      ************************************************************************
C      *
C      *
C      *
C      ************************************************************************
C                                                                       CONTM
C      THIS IS THE ENTRY POINT TO CONTM                                 CONTM
```

```
C                                                                       CONTM
      IF (NUMCMG .EQ. 0) GO TO 195                                      CONTM
      CALL HCON                                                         CONTM
  195 CONTINUE                                                          CONTM
      FPT(1) = 0.                                                       CONTM
      FPT(2) = 0.                                                       CONTM
      FPT(3) = 0.                                                       CONTM
      FPT(4) = 0.                                                       CONTM
      FPT(5) = 0.                                                       CONTM
      FAT(1) = 0.                                                       CONTM
      FAT(2) = 0.                                                       CONTM
      FAT(3) = 0.                                                       CONTM
      FAT(4) = 0.                                                       CONTM
      FAT(5) = 0.                                                       CONTM
      FAT(6) = 0.                                                       CONTM
      FAT(7) = 0.                                                       CONTM
      FAT(8) = 0.                                                       CONTM
      DO 196 M=1,3                                                      CONTM
      TQOP(M) = 0.                                                      CONTM
      TQIP(M) = 0.                                                      CONTM
  196 CONTINUE                                                          CONTM
      IF (IPROPF .EQ. 0) GO TO 199                                      CONTM
      IF (IATTIF .EQ. 0) GO TO 197                                      CONTM
      CALL ATT                                                          CONTM
      IF (ICFA .EQ. 1) GO TO 199                                        CONTM
  197 CONTINUE                                                          CONTM
      CALL PCON                                                         CONTM
  199 CONTINUE                                                          CONTM
      TMP = FAT(1) + FAT(2) + FPT(2) + FPT(3) + FPT(4)                  CONTM
      TAP = TAP + TMP*DELTAT                                            CONTM
      TNP = FPT(1) + FPT(5)                                             CONTM
      TBP = TBP + TNP*DELTAT                                            CONTM
C                                                                       CONTM
C     ******************************************************************
C     *
C     *
C     *
C     ******************************************************************
C                                                                       INVER
C     THIS IS THE ENTRY POINT TO THE INVERSION BLOCK                    INVER
C                                                                       INVER
      CALL RECALC                                                       INVER
      CALL EMCALC                                                       INVER
C                                                                       INVER
C     SET UP THE X MATRIX                                               INVER
      DO 200 M=1,6                                                      INVER
      DO 200 N=1,6                                                      INVER
      X(M,N) = EM(M,N)                                                  INVER
  200 CONTINUE                                                          INVER
      STEMP1 = B2MASS*SDOT*(-R2(3)*S2(2) + R2(2)*S2(3))                 INVER
      STEMP2 = B2MASS*SDOT*((R2(3)*S2(1) - R2(1)*S2(3))*COSTT1 -        INVER
     *                     (-R2(2)*S2(1) + R2(1)*S2(2))*SINTT1)         INVER
      STEMP3 = B2MASS*SDOT*((R2(3)*S2(1) - R2(1)*S2(3))*SINTT1 +        INVER
     *                     (-R2(2)*S2(1) + R2(1)*S2(2))*COSTT1)         INVER
      STEMP4 = SDOT*(S2(2)*(-B2MASS*R2(3) + (B0MASS*B2MASS/TOTMAS)*     INVER
     * (D01(2)*SINTT1 - D01(3)*COSTT1)) + S2(3)*(B2MASS*R2(2) -         INVER
     * (B0MASS*B2MASS/TOTMAS)*(D01(2)*COSTT1 + D01(3)*SINTT1)))         INVER
      STEMP5 = (B2MASS*B3MASS/TOTMAS)*SDOT*((EL3(3)*S2(1) -             INVER
     * EL3(1)*S2(3))*S3(2) + (-EL3(2)*S2(1) + EL3(1)*S2(2))*S3(3))      INVER
      STEMP6 = (B2MASS*B4MASS/TOTMAS)*SDOT*((EL4(3)*S2(1) -             INVER
     * EL4(1)*S2(3))*S4(2) + (-EL4(2)*S2(1) + EL4(1)*S2(2))*S4(3))      INVER
      H(1) = H(1) - STEMP1                                              INVER
      H(2) = H(2) - STEMP2                                              INVER
      H(3) = H(3) - STEMP3                                              INVER
      H1PRIM(1) = H1PRIM(1) - STEMP4                                    INVER
      G3 = G3 + STEMP5                                                  INVER
      G4 = G4 + STEMP6                                                  INVER
      IF (IPNDLM .EQ. 0) GO TO 202                                      INVER
      DO 201 M=1,3                                                      INVER
      V(M) = H(M) - FFF(M)                                              INVER
  201 CONTINUE                                                          INVER
      X(1,7) = V(1)                                                     INVER
      X(2,7) = V(2)                                                     INVER
      X(3,7) = V(3)                                                     INVER
      X(4,7) = H1PRIM(1)                                                INVER
      X(5,7) = G3                                                       INVER
      X(6,7) = G4                                                       INVER
      CALL SYEQNS(X,6,6,7,FLAG)                                         INVER
      WO(1) =  X(1,7)                                                   INVER
      WO(2) =  X(2,7)                                                   INVER
      WO(3) =  X(3,7)                                                   INVER
      OMEGA1 = X(4,7)                                                   INVER
      OMEGA3 = X(5,7)                                                   INVER
      OMEGA4 = X(6,7)                                                   INVER
      GO TO 204                                                         INVER
  202 CONTINUE                                                          INVER
      DO 203 M=1,3                                                      INVER
      V(M) = H(M) - FFF(M)                                              INVER
  203 CONTINUE                                                          INVER
      X(1,5) = V(1)                                                     INVER
      X(2,5) = V(2)                                                     INVER
      X(3,5) = V(3)                                                     INVER
      X(4,5) = H1PRIM(1)                                                INVER
      CALL SYEQNS(X,4,6,7,FLAG)                                         INVER
      WO(1) = X(1,5)                                                    INVER
      WO(2) = X(2,5)                                                    INVER
      WO(3) = X(3,5)                                                    INVER
      OMEGA1 = X(4,5)                                                   INVER
      OMEGA3 = 0.                                                       INVER
      OMEGA4 = 0.                                                       INVER
  204 CONTINUE                                                          INVER
C                                                                       INVER
C     **********************************************************************
C     *
C     *
C     *
C     **********************************************************************
C
C     UPDATE TIME
C
      TIME = TIME + DELTAT
      CALL XDOT
C
C     **********************************************************************
C     *
C     *
C     *
C     **********************************************************************
C                                                                       OUTPU
C     THIS IS THE ENTRY POINT TO THE OUTPUT BLOCK.                      OUTPU
C                                                                       OUTPU
      IF (IPNTCK .NE. IPRINT) GO TO 215                                 OUTPU
C     SKIP TO A NEW PAGE AND PRINT A LINE OF ASTERICKS.                 OUTPU
      WRITE(6,6502)                                                     OUTPU
      WRITE(6,6504) TIME                                                OUTPL
      IF (IDOCK .EQ. 0) GO TO 208                                       OUTPL
      IF((TIME.LT.DTMIN).OR.(TIME.GT.DTMAX)) GO TO 208                  OUTPU
      WRITE(6,6580)                                                     OUTPU
      IDOCK = 0                                                         OUTPL
  208 CONTINUE                                                          OUTPU
      WRITE(6,6550)                                                     OUTPU
      WRITE(6,6510) H(1),H(2),H(3)                                      OUTPU
      WRITE(6,6514) HDOT(1),HDOT(2),HDOT(3)                             OUTPU
      WRITE(6,6512) HI(1),HI(2),HI(3)                                   OUTPU
      DO 207 M=1,3                                                      OUTPU
      WRITE(6,6538) M,TIBO(M,1),M,TIBO(M,2),M,TIBO(M,3)                 OUTPU
  207 CONTINUE                                                          OUTPU
      WRITE(6,6536) THETO,TOTMAS                                        OUTPU
      J = 0                                                             OUTPU
      WRITE(6,6552) J                                                   OUTPU
      WRITE(6,6008) WO(1),WO(2),WO(3)                                   OUTPU
      J = 0                                                             OUTPU
      WRITE(6,6526) J,RO(1),J,RO(2),J,RO(3)                             OUTPU
      WRITE(6,6554)                                                     OUTPU
      J = 1                                                             OUTPU
      WRITE(6,6552) J                                                   OUTPU
      WRITE(6,6506) J,W1(1),J,W1(2),J,W1(3)                             OUTPU
      WRITE(6,6526) J,R1(1),J,R1(2),J,R1(3)                             OUTPU
      WRITE(6,6528) J,R1DOT(1),J,R1DOT(2),J,R1DOT(3)                    OUTPU
      WRITE(6,6042) THETA1,OMEGA1                                       OUTPL
      WRITE(6,6554)                                                     OUTPL
      IF (IB2F .EQ. 0) GO TO 209                                        OUTPL
      J = 2                                                             OUTPL
      WRITE(6,6552) J                                                   OUTPL
      WRITE(6,6556) S,SDOT                                              OUTPU
      WRITE(6,6522) J,EL2(1),J,EL2(2),J,EL2(3)                          OUTPL
      WRITE(6,6524) J,EL2DOT(1),J,EL2DOT(2),J,EL2DOT(3)                 OUTPL
      WRITE(6,6526) J,R2(1),J,R2(2),J,R2(3)                             OUTPU
      WRITE(6,6528) J,R2DOT(1),J,R2DOT(2),J,R2DOT(3)                    OUTPU
  209 CONTINUE                                                          OUTPL
      WRITE(6,6554)                                                     OUTPU
      IF (IPNDLM .EQ. 0) GO TO 211                                      OUTPL
      J = 3                                                             OUTPL
      WRITE(6,6552) J                                                   OUTPL
      WRITE(6,6506) J,W3(1),J,W3(2),J,W3(3)                             OUTPU
      WRITE(6,6516) J,H3PRIM(1),J,H3PRIM(2),J,H3PRIM(3)                 OUTPL
      WRITE(6,6522) J,EL3(1),J,EL3(2),J,EL3(3)                          OUTPL
      WRITE(6,6524) J,EL3DOT(1),J,EL3DOT(2),J,EL3DOT(3)                 OUTPU
      WRITE(6,6526) J,R3(1),J,R3(2),J,R3(3)                             OUTPL
      WRITE(6,6528) J,R3DOT(1),J,R3DOT(2),J,R3DOT(3)                    OUTPL
      WRITE(6,6558) J,THETA3,J,OMEGA3                                   OUTPL
      WRITE(6,6560) J,G3,J,G3DOT                                        OUTPL
      WRITE(6,6554)                                                     OUTPL
      J = 4                                                             OUTPL
      WRITE(6,6552) J                                                   OUTPL
      WRITE(6,6506) J,W4(1),J,W4(2),J,W4(3)                             OUTPL
      WRITE(6,6516) J,H4PRIM(1),J,H4PRIM(2),J,H4PRIM(3)                 OUTPL
      WRITE(6,6522) J,EL4(1),J,EL4(2),J,EL4(3)                          OUTPL
      WRITE(6,6524) J,EL4DOT(1),J,EL4DOT(2),J,EL4DOT(3)                 OUTPL
      WRITE(6,6526) J,R4(1),J,R4(2),J,R4(3)                             OUTPU
      WRITE(6,6528) J,R4DOT(1),J,R4DOT(2),J,R4DOT(3)                    OUTPL
      WRITE(6,6558) J,THETA4,J,OMEGA4                                   OUTPL
      WRITE(6,6560) J,G4,J,G4DOT                                        OUTPL
  211 CONTINUE                                                          OUTPL
      IF (IPROPF .EQ. 0) GO TO 212                                      OUTPL
      WRITE(6,6588) TAP                                                 OUTPL
      WRITE(6,6590) TBP                                                 OUTPL
  212 CONTINUE                                                          OUTPL
      IF (NUMCMG .EQ. 0) GO TO 214                                      OUTPL
      DO 213 J=1,NUMCMG                                                 OUTPL
      WRITE(6,6028) J,THATA(J),J,THATAD(J)                              OUTPL
      IF (IDOF(J) .NE. 2) GO TO 213                                     OUTPL
      WRITE(6,6032) J,FEE(J),J,FEED(J)                                  OUTPL
  213 CONTINUE                                                          OUTPL
  214 CONTINUE                                                          OUTPL
      IPNTCK = 0                                                        OUTPL
  215 IPNTCK = IPNTCK + 1                                               OUTPL
C                                                                       OUTPL
C     **************************************************************************
C     *
C     *
C     *
C     **************************************************************************
      IF (IDOCK .NE. 1) GO TO 220
      IF((TIME.LT.DCHMIN).OR.(TIME.GT.DCHMAX)) GO TO 220
      DO 218 M=1,3
      DO 218 N=1,3
      BODYOI(M,N) = BODYDI(M,N)
  218 CONTINUE
      BOMASS = BDMASS
      DO1(1) = DDO1(1)
      DO1(2) = DDO1(2)
      DO1(3) = DDO1(3)
      TOTMAS = BOMASS + B1MASS + B2MASS + B3MASS + B4MASS
      GO TO 190
  220 IF (TIME .GT. TSTOP) GO TO 10
      GO TO 190
C     **************************************************************************
C     *
C     *
C     *
C     **************************************************************************
C                                                                       I/O
C     THIS SECTION CONTAINS ALL OF THE INPUT AND OUTPUT FORMATS.        I/O
C                                                                       I/O
 5000 FORMAT(1X,I2)                                                     I/O
 5002 FORMAT(1X,3(F11.5,2X))                                            I/O
 5004 FORMAT(1X,E11.4)                                                  I/O
 5006 FORMAT(1X,F11.5)                                                  I/O
 5008 FORMAT(1X,E11.4,2X,E11.4)                                         I/O
 6000 FORMAT(1H1,1X,16H THERE ARE (IS) ,I2,22H DATA DECK(S) PRESENT.)   I/O
 6002 FORMAT(1H1,1X,14HTSTART       = ,E13.6,23X,14HTSTOP        = ,E13.6,I/O
     *              23X,14HDELTAT       = ,E13.6)                       I/O
 6004 FORMAT(1X,14HALTITUDE     = ,E13.6)                               I/O
 6006 FORMAT(1X,6HTIBOI(,I1,7H,1)    = ,F11.5,25X,6HTIBOI(,I1,7H,2)   = , I/O
     * F11.5,25X,6HTIBOI(,I1,7H,3)    = ,F11.5)                         I/O
 6008 FORMAT(1X,14HWO(1)        = ,E13.6,23X,14HWO(2)        = ,E13.6,23X,I/O
     *              14HWO(3)        = ,E13.6)                           I/O
 6010 FORMAT(1X,1HB,I1,12HMASS         = ,E13.6)                        I/O
 6012 FORMAT(1X,7HBODYOI(,I1,6H,1) = ,E13.6,23X,7HBODYOI(,I1,6H,2) = ,  I/O
     *E13.6,23X,7HBODYOI(,I1,6H,3) = ,E13.6)                            I/O
 6014 FORMAT(1X,14HNUMCMG       = ,I2)                                  I/O
 6016 FORMAT(1X,11HCMG NUMBER ,I1,21H IS A REACTION WHEEL.)             I/O
 6018 FORMAT(1X,11HCMG NUMBER ,I1,26H HAS ONE DEGREE OF FREEDOM)        I/O
 6020 FORMAT(1X,11HCMG NUMBER ,I1,27H HAS TWO DEGREES OF FREEDOM)       I/O
 6022 FORMAT(1X,35HTHE ANGULAR MOMENTUM OF CMG NUMBER ,I1,3H = ,E13.6)  I/O
 6024 FORMAT(1X,5HAOCJ(,I1,1H,,I1,6H,1) = ,F11.5,25X,5HAOCJ(,I1,1H,,I1, I/O
     * 6H,2) = ,F11.5,25X,5HAOCJ(,I1,1H,,I1,6H,3) = ,F11.5)             I/O
```

```
6026 FORMAT(1X,4HAII(,I1,1H,,I1,7H,1)  = ,E13.6,23X,4HAII(,I1,1H,,I1,   I/O
    * 7H,2)  = ,E13.6,23X,4HAII(,I1,1H,,I1,7H,3)  = ,E13.6)              I/O
6028 FORMAT(1X,6HTHATA(,I1,7H)     = ,E13.6,23X,7HTHATAD(,I1,6H)    = ,  I/O
    * E13.6)                                                              I/O
6030 FORMAT(1X,4HAIO(,I1,1H,,I1,7H,1)  = ,E13.6,23X,4HAIO(,I1,1H,,I1,   I/O
    * 7H,2)  = ,E13.6,23X,4HAIO(,I1,1H,,I1,7H,3)  = ,E13.6)              I/O
6032 FORMAT(1X,4HFEE(,I1,9H)       = ,E13.6,23X,5HFEED(,I1,8H)      = ,  I/O
    * E13.6)                                                              I/O
6040 FORMAT(1X,7HBODY1I(,I1,6H,1) = ,E13.6,23X,7HBODY1I(,I1,6H,2) = ,   I/O
    * E13.6,23X,7HBODY1I(,I1,6H,3) = ,E13.6)                              I/O
6042 FORMAT(1X,14HTHETA1          = ,E13.6,23X,14HOMEGA1         = ,E13.6)  I/O
6044 FORMAT(1X,14HD01(1)          = ,E13.6,23X,14HD01(2)         = ,E13.6,  I/O
    *        23X,14HD01(3)        = ,E13.6)                               I/O
6048 FORMAT(1X,14HD13(1)          = ,E13.6,23X,14HD13(2)         = ,E13.6,  I/O
    *        23X,14HD13(3)        = ,E13.6)                               I/O
6050 FORMAT(1X,14HD14(1)          = ,E13.6,23X,14HD14(2)         = ,E13.6,  I/O
    *        23X,14HD14(3)        = ,E13.6)                               I/O
6052 FORMAT(1X,1HS,I1,12H(1)          = ,E13.6,23X,1HS,I1,12H(2)        = ,I/O
    *E13.6,23X,1HS,I1,12H(3)          = ,E13.6)                           I/O
6058 FORMAT(1X,14HPEND3L          = ,E13.6)                               I/O
6060 FORMAT(1X,14HTHETA3          = ,E13.6)                               I/O
6062 FORMAT(1X,14HOMEGA3          = ,E13.6)                               I/O
6064 FORMAT(1X,14HPEND4L          = ,E13.6)                               I/O
6066 FORMAT(1X,14HTHETA4          = ,E13.6,23X,14HOMEGA4         = ,E13.6)  I/O
6068 FORMAT(1X,14HIDOCK           = ,I2)                                  I/O
6070 FORMAT(1X,14HDTIME           = ,E13.6)                               I/O
6072 FORMAT(1X,14HIPROPF          = ,I2)                                  I/O
6074 FORMAT(1X,4HA0J(,I2,8H)          = ,E13.6,23X,7HCGAIN0(,I2,5H)   = ,  I/O
    * E13.6)                                                              I/O
6076 FORMAT(1X,4HA1J(,I1,9H)          = ,E13.6,23X,7HCGAIN1(,I1,6H)    = ,  I/O
    * E13.6)                                                              I/O
6078 FORMAT(1X,14HIGRAVF          = ,I2)                                  I/O
6080 FORMAT(1X,14HSP              = ,E13.6)                               I/O
6082 FORMAT(1H1,1X,9HIPNDLM = ,I2,/)                                      I/O
6500 FORMAT(1H1,1X,46H THE FOLLOWING INPUT CORRESPONDS TO DATA DECK ,I2I/O
    *)                                                                    I/O
6502 FORMAT(1H1,1X,127H*****************************************************I/O
    ****************************************************************I/O
    **************,/)                                                     I/O
6504 FORMAT(1X,14HTIME            = ,F11.5,/)                             I/O
6506 FORMAT(1X,1HW,I1,12H(1)          = ,E13.6,23X,1HW,I1,12H(2)        = ,I/O
    *E13.6,23X,1HW,I1,12H(3)          = ,E13.6)                           I/O
6510 FORMAT(1X,14HH(1)            = ,E13.6,23X,14HH(2)           = ,E13.6,  I/O
    *        23X,14HH(3)          = ,E13.6)                               I/O
6512 FORMAT(1X,14HHI(1)           = ,E13.6,23X,14HHI(2)          = ,E13.6,  I/O
    *        23X,14HHI(3)         = ,E13.6)                               I/O
6514 FORMAT(1X,14HHDOT(1)         = ,E13.6,23X,14HHDOT(2)        = ,E13.6,  I/O
    *        23X,14HHDOT(3)       = ,E13.6)                               I/O
6516 FORMAT(1X,1HH,I1,12HPRIM(1)      = ,E13.6,23X,1HH,I1,12HPRIM(2)    = ,I/O
    *E13.6,23X,1HH,I1,12HPRIM(3)      = ,E13.6)                           I/O
6522 FORMAT(1X,2HEL,I1,11H(1)         = ,E13.6,23X,2HEL,I1,11H(2)       = ,I/O
    *E13.6,23X,2HEL,I1,11H(3)         = ,E13.6)                           I/O
6524 FORMAT(1X,2HEL,I1,11HDOT(1)      = ,E13.6,23X,2HEL,I1,11HDOT(2)    = ,I/O
    *E13.6,23X,2HEL,I1,11HDOT(3)      = ,E13.6)                           I/O
6526 FORMAT(1X,1HR,I1,12H(1)          = ,E13.6,23X,1HR,I1,12H(2)        = ,I/O
    *E13.6,23X,1HR,I1,12H(3)          = ,E13.6)                           I/O
6528 FORMAT(1X,1HR,I1,12HDOT(1)       = ,E13.6,23X,1HR,I1,12HDOT(2)     = ,I/O
    *E13.6,23X,1HR,I1,12HDOT(3)       = ,E13.6)                           I/O
6535 FORMAT(1X,14HD12(1)          = ,E13.6,23X,14HD12(2)         = ,E13.6,  I/O
    *        23X,14HD12(3)        = ,E13.6)                               I/O
6536 FORMAT(1X,14HTHETO           = ,F11.5,25X,14HTOTMAS         = ,E13.6,/)  I/O
6538 FORMAT(1X,5HTIBO(,I1,8H,1)       = ,F11.5,25X,5HTIBO(,I1,8H,2)     = ,  I/O
     *F11.5,25X,5HTIBO(,I1,8H,3)    = ,F11.5)                            I/O
 6544 FORMAT(1X,14HIPRINT        = ,I2)                                   I/O
 6546 FORMAT(1X,14HIB2F          = ,I2)                                   I/O
 6548 FORMAT(1X,14HS             = ,E13.6,23X,14HSDOT          = ,E13.6)  I/O
 6550 FORMAT(1X,19HGENERAL INFORMATION,/)                                 I/O
 6552 FORMAT(1X,5HBODY ,I1,12H INFORMATION,/)                             I/O
 6554 FORMAT(/)                                                           I/O
 6556 FORMAT(1X,14HS             = ,E13.6,23X,14HSDOT          = ,E13.6)  I/O
 6558 FORMAT(1X,5HTHETA,I1,8H        = ,E13.6,23X,5HOMEGA,I1,8H        = , I/O
     * E13.6)                                                             I/O
 6560 FORMAT(1X,1HG,I1,12H           = ,E13.6,23X,1HG,I1,12HDOT        = ,I/O
     * E13.6)                                                             I/O
 6562 FORMAT(1X,14HIATTIF        = ,I1)                                   I/O
 6564 FORMAT(1X,14HCA(1)         = ,E13.6,23X,14HCA(2)         = ,E13.6,  I/O
     *          23X,14HCA(3)     = ,E13.6)                                I/O
 6566 FORMAT(1X,14HAA01          = ,E13.6,23X,14HAGAIN1        = ,E13.6)  I/O
 6568 FORMAT(1X,14HAA02          = ,E13.6,23X,14HAGAIN2        = ,E13.6)  I/O
 6570 FORMAT(1X,14HNGAIN         = ,I2)                                   I/O
 6572 FORMAT(1X,5HGAIN(,I1,8H)      = ,E13.6)                             I/O
 6574 FORMAT(1X,14HBDMASS        = ,E13.6)                                I/O
 6576 FORMAT(1X,7HBODYDI(,I1,6H,1) = ,E13.6,23X,7HBODYDI(,I1,6H,2) = ,    I/O
     *E13.6,23X,7HBODYDI(,I1,6H,3) = ,E13.6)                              I/O
 6578 FORMAT(1X,14HDTI(1)        = ,E13.6,23X,14HDTI(2)        = ,E13.6,  I/O
     *          23X,14HDTI(3)    = ,E13.6)                                I/O
 6580 FORMAT(1X,20HDOCKING HAS OCCURRED)                                  I/O
 6582 FORMAT(1X,14HCP1           = ,E13.6)                                I/O
 6584 FORMAT(1X,14HCP2           = ,E13.6)                                I/O
 6586 FORMAT(1X,14HDD01(1)       = ,E13.6,23X,14HDD01(2)       = ,E13.6,  I/O
     *          23X,14HDD01(3)   = ,E13.6)                                I/O
 6588 FORMAT(1X,54HTHE TOTAL PROPULSION IMPULSE ON THE TRANSVERSE AXIS =I/O
     * ,E13.6)                                                            I/O
 6590 FORMAT(1X,48HTHE TOTAL PROPULSION IMPULSE ON THE SPIN AXIS = ,      I/O
     *  E13.6)                                                            I/O
C                                                                         I/O
C     **************************************************************************
C     *
C     *
C     *
      END
      SUBROUTINE ATT
C     *
C     *
C     *
C     ***************************************************************************
      COMMON    A(3)          ,   AE(5)          ,   AED(5)        , COMMC
     *          AFOUR(2)      ,   AII(6,3,3)     ,                   COMMC
     *          AIO(6,3,3)    ,   AJ1(3)         ,   ALT           , COMMC
     *          AOCJ(6,3,3)   ,   AOJ(3)         ,   AONE(7)       , COMMC
     *          ATCPT2(3,3)   ,   ATHREE(5)      ,   ATWO(4)       , COMMC
     *          A1(3,3)       ,   A1J(2)                             COMMC
      COMMON    BDMASS        ,   BFOUR(2)       ,   BMOM          , COMMC
     *          BODYDI(3,3)   ,   BODYOI(3,3)    ,   BODY1I(3,3)   , COMMC
     *          BOMASS        ,   BONE(7)        ,   BTHREE(5)     , COMMC
     *          BTWO(4)       ,   B1MASS         ,   B2MASS        , COMMC
     *          B3MASS        ,   B4MASS                             COMMC
      COMMON    CA(3)         ,   CB(3)          ,   CGAINO(3)     , COMMC
     *          CGAIN1(2)     ,                                      COMMC
     *          COSFEJ        ,   COSTTJ         ,   COSTTO        , COMMC
     *          COSTT1        ,   COSTT3         ,   COSTT4        , COMMC
     *          CO2T          ,   CP1            ,   CP2           , COMMC
     *          CST           ,   C1                                 COMMC
      COMMON    DB(3)         ,   DDO1(3)        ,                   COMMC
     *          DELTAT        ,   DO1(3)         ,   DO1DOT(3)     , COMMC
     *          DTI(3)        ,                                      COMMC
     *          DTIME         ,   D12(3)         ,   D13(3)        , COMMC
     *          D13DOT(3)     ,   D13YCS         ,   D13YSN        , COMMC
     *          D13ZCS        ,   D13ZSN         ,   D14(3)        , COMMC
     *          D14DOT(3)     ,   D14YCS         ,   D14YSN        , COMMC
     *          D14ZCS        ,   D14ZSN                             COMMC
      COMMON    EEE(3,3)      ,   EEJ(3,3)       ,   EL2(3)        , COMMC
     *          EL2DOT(3)     ,   EL2YCS         ,   EL2YSN        , COMMC
     *          EL2ZCS        ,   EL2ZSN         ,   EL3(3)        , COMMC
     *          EL3DOT(3)     ,   EL3YCS         ,   EL3YSN        , COMMC
     *          EL3ZCS        ,   EL3ZSN         ,   EL4(3)        , COMMC
     *          EL4DOT(3)     ,   EL4YCS         ,   EL4YSN        , COMMC
     *          EL4ZCS        ,   EL4ZSN         ,   EM(6,6)         COMMC
      COMMON    FAT(8)        ,                                      COMMC
     *          FEE(6)        ,   FEED(6)        ,   FFF(3)        , COMMC
     *          FFJ(3)        ,   FLAG1          ,   FLAG2         , COMMC
     *          FLAG3         ,   FLAG4          ,   FNI           , COMMC
     *          FO(3)         ,   FO1(3)         ,   FO2(3)        , COMM(
     *          FO3(3)        ,   F1(3)          ,   F11(3)        , COMM(
     *          FPT(5)        ,                                      COMM(
     *          F12(3)        ,   F13(3)                             COMM(
      COMMON    GAIN(10)      ,   G3             ,                   COMM(
     *          G3DOT         ,   G4             ,   G4DOT           COMM(
      COMMON    H(3)          ,   HCMG(3)        ,   HDOT(3)       , COMM(
     *          HI(3)         ,   HO(3)          ,   HW(6)         , COMM(
     *          H1(3)         ,   H1PDOT(3)      ,   H1PRIM(3)     , COMM(
     *          H3PRIM(3)     ,   H4PRIM(3)                          COMM(
      COMMON    IB2F          ,   ICFA           ,   ICFB          , COMM(
     *          ICFC          ,   ICFD           ,   IDOCK         , COMM(
     *          IDOF(6)       ,                                      COMM(
     *          IGRAVF        ,   IPNDLM         ,   IPNTCK        , COMM(
     *          IPRINT        ,   IPROPF                             COMM(
      COMMON    NCASE         ,   NCHECK         ,   NDECK         , COMM(
     *          NGAIN         ,   NUMCMG                             COMM(
      COMMON    OMEGA1        ,   OMEGA3         ,   OMEGA4          COMM(
      COMMON    PEND3L        ,   PEND4L                             COMM(
      COMMON    Q(4,4)                                               COMM(
      COMMON    R               ,    RO(3)         ,    R1(3)          , COMMC
     *          R1DOT(3)        ,    R1YCS         ,    R1YSN          , COMMC
     *          R1ZCS           ,    R1ZSN         ,    R2(3)          , COMMC
     *          R2DOT(3)        ,    R2YCS         ,    R2YSN          , COMMC
     *          R2ZCS           ,    R2ZSN         ,    R3(3)          , COMMC
     *          R3DOT(3)        ,    R3YCS         ,    R3YSN          , COMMC
     *          R3ZCS           ,    R3ZSN         ,    R4(3)          , COMMC
     *          R4DOT(3)        ,    R4YCS         ,    R4YSN          , COMMC
     *          R4ZCS           ,    R4ZSN                               COMMC
      COMMON    S               ,    SDOT          ,    SINPEJ         , COMMC
     *          SINTTJ          ,    SINTTO        ,    SINTT1         , COMMC
     *          SINTT2          ,    SINTT3        ,    SINTT4         , COMMC
     *          SP              ,    SUM1          ,    SUM2           , COMMC
     *          SUM3            ,    S2(3)         ,    S3(3)          , COMMC
     *          S4(3)                                                    COMMC
      COMMON    T(3,3)          ,    TC(3,3)       ,    TEMP1(3)       , COMMC
     *          TEMP2(3)        ,                                        COMMC
     *          TEMP3(3)        ,    TEMP4(3)      ,    TEMP5(3,3)     , COMMC
     *          TEMP6(3,3)      ,    TEMP7(3,3)    ,    TEMP8(3,3)     , COMMC
     *          TEMP9(3,3)      ,    TEMP10(3,3)   ,    TEMP11(3,3)    , COMMC
     *          TEMP12(3,3)     ,    TEMP13(3,3)   ,    TEMP14(3,3)    , COMMC
     *          TEMP15(3,3)     ,    TERM1(3)      ,    TERM2(3)       , COMMC
     *          TFRICT          ,    THATA(6)      ,    THATAD(6)      , COMMC
     *          THETA1          ,    THETA3        ,    THETA4         , COMMC
     *          THETO           ,    TIBO(3,3)     ,    TIBOI(3,3)     , COMMC
     *          TIME            ,    TJ            ,    TJ1(10)        , COMMC
     *          TJ2(10)         ,    TJ3(10)       ,    TJ4(10)        , COMMC
     *          TMOTOR          ,                                        COMMC
     *          TOEF(3)         ,    TOTMAS        ,    TO1            , COMMC
     *          TQOG(3)         ,    TQOP(3)       ,    TQ1G(3)        , COMMC
     *          TQ1P(3)         ,    TSTART        ,    TSTOP          , COMMC
     *          TT1DOT          ,    TT3DOT        ,    TT4DOT         , COMMC
     *          T1EF(3)         ,    T13           ,    T14              COMMC
      COMMON    V(3)                                                     COMMC
      COMMON    W0(3)           ,    WS            ,    W1(3)          , COMMC
     *          W3(3)           ,    W4(3)                               COMMC
      COMMON    X(6,7)          ,    XC            ,    XCDOT          , COMMC
     *          XMU                                                      COMMC
C                                                                        COMMC
C     ***********************************************************M***********
C     *
C     *
C     *
C     ************************************************************************
C                                                                        ATT
      ICFD = 0                                                           ATT
      IF (ICFA .EQ. 1) GO TO 11                                          ATT
      IF (ABS(W0(2)) .GT. 0.0002) GO TO 60                               ATT
      IF (ABS(W0(3)) .GT. 0.0002) GO TO 60                               ATT
      DO 10 M=1,3                                                        ATT
      CB(M) = TIBO(1,M)*CA(1) + TIBO(2,M)*CA(2) + TIBO(3,M)*CA(3)        ATT
   10 CONTINUE                                                           ATT
      IF (CB(1) .LT. 0.9994) GO TO 11                                    ATT
      ICFA = 0                                                           ATT
      ICFB = 0                                                           ATT
      ICFC = 0                                                           ATT
      GO TO 60                                                           ATT
   11 CONTINUE                                                           ATT
      ICFA = 1                                                           ATT
      IF (ICFB .EQ. 1) GO TO 30                                          ATT
      TMA = TIME                                                         ATT
      ZTCL = 0.                                                          ATT
      DO 13 M=1,3                                                       ATT
      DO 13 N=1,3                                                       ATT
      TC(M,N) = TIBO(M,N)                                               ATT
   13 CONTINUE                                                          ATT
      CN = CB(3)**2 + CB(2)**2                                          ATT
      CN = SQRT(CN)                                                     ATT
      CA2 = CB(2)/CN                                                    ATT
      CA3 = CB(3)/CN                                                    ATT
      AL = 0.5*ACOS(CB(1))                                              ATT
      AK = BODY1I(1,1)*OMEGA1*SIN(AL)/(5.5 + DELTAT)                    ATT
      ICFB = 1                                                          ATT
   30 CONTINUE                                                          ATT
      TMACHK = TMA + 5.0                                                ATT
      IF (TIME .GT. TMACHK) GO TO 40                                    ATT
      ICFD = 1                                                          ATT
      GO TO 60                                                          ATT
   40 CONTINUE                                                          ATT
      IF (ICFC .EQ. 1) GO TO 50                                         ATT
      ZTC = -CA3*(TC(1,2)*TIBO(1,1)+TC(2,2)*TIBO(2,1)+TC(3,2)*TIBO(3,1))ATT
     *       +CA2*(TC(1,3)*TIBO(1,1)+TC(2,3)*TIBO(2,1)+TC(3,3)*TIBO(3,1))ATT
      IF (ZTC .GT. ZTCL) GO TO 71                                       ATT
      ZTCL = ZTC                                                        ATT
      GO TO 60                                                          ATT
   71 CONTINUE                                                          ATT
      TMB = TIME                                                        ATT
      TMC = TMB - TMA - 6.0                                             ATT
      TMC = TMB + TMC                                                   ATT
      TMB = TMC + 5.0                                                   ATT
      ICFC = 1                                                          ATT
      GO TO 60                                                          ATT
   50 CONTINUE                                                          ATT
      IF (TIME .GT. TMB) GO TO 80                                       ATT
      IF (TIME .LT. TMC) GO TO 60                                       ATT
      ICFD = 1                                                          ATT
      GO TO 60                                                          ATT
   80 CONTINUE                                                          ATT
      ICFA = 0                                                          ATT
      ICFB = 0                                                          ATT
      ICFC = 0                                                          ATT
   60 CONTINUE                                                          ATT
      IF (ICFD .EQ. 0) GO TO 90                                         ATT
      TQOP(2) = AK*CA2                                                  ATT
      TQOP(3) = AK*CA3                                                  ATT
      FAT(1) = 2.*ABS(TQOP(2)/AOJ(2))                                   ATT
      FAT(2) = 2.*ABS(TQOP(3)/AOJ(3))                                   ATT
   90 CONTINUE                                                          ATT
   65 CONTINUE                                                          ATT
      RETURN                                                            ATT
C                                                                       ATT
C     ***************************************************************************
C     *
C     *
C     *
      END
      SUBROUTINE CMG
C     *
C     *
C     *
C     ************************************************************************
      COMMON      A(3)           ,    AE(5)          ,    AED(5)         , COMMC
     *            AFOUR(2)       ,    AII(6,3,3)     ,                     COMMC
     *            AIO(6,3,3)     ,    AJ1(3)         ,    ALT            , COMMC
     *            AOCJ(6,3,3)    ,    AOJ(3)         ,    AONE(7)        , COMMC
     *            ATCPT2(3,3)    ,    ATHREE(5)      ,    ATWO(4)        , COMMC
     *            A1(3,3)        ,    A1J(2)                               COMMC
      COMMON      BDMASS         ,    BFOUR(2)       ,    BMOM           , COMMC
     *            BODYDI(3,3)    ,    BODYOI(3,3)    ,    BODY1I(3,3)    , COMMC
     *            BOMASS         ,    BONE(7)        ,    BTHREE(5)      , COMMC
     *            BTWO(4)        ,    B1MASS         ,    B2MASS         , COMMC
     *            B3MASS         ,    B4MASS                               COMMC
      COMMON      CA(3)          ,    CB(3)          ,    CGAIN0(3)      , COMMC
     *            CGAIN1(2)      ,                                         COMMC
     *            COSFEJ         ,    COSTTJ         ,    COSTT0         , COMMC
     *            COSTT1         ,    COSTT3         ,    COSTT4         , COMMC
     *            CO2T           ,    CP1            ,    CP2            , COMMC
     *            CST            ,    C1                                   COMMC
      COMMON      DB(3)          ,    DD01(3)        ,                     COMMC
     *            DELTAT         ,    D01(3)         ,    D01DOT(3)      , COMMC
     *            DTI(3)         ,                                         COMMC
     *            DTIME          ,    D12(3)         ,    D13(3)         , COMMC
     *            D13DOT(3)      ,    D13YCS         ,    D13YSN         , COMMC
     *            D13ZCS         ,    D13ZSN         ,    D14(3)         , COMMC
     *            D14DOT(3)      ,    D14YCS         ,    D14YSN         , COMMC
     *            D14ZCS         ,    D14ZSN                               COMMC
      COMMON      EEE(3,3)       ,    EEJ(3,3)       ,    EL2(3)         , COMMC
     *            EL2DOT(3)      ,    EL2YCS         ,    EL2YSN         , COMMC
     *            EL2ZCS         ,    EL2ZSN         ,    EL3(3)         , COMMC
     *            EL3DOT(3)      ,    EL3YCS         ,    EL3YSN         , COMMC
     *            EL3ZCS         ,    EL3ZSN         ,    EL4(3)         , COMMC
     *            EL4DOT(3)      ,    EL4YCS         ,    EL4YSN         , COMMC
     *            EL4ZCS         ,    EL4ZSN         ,    EM(6,6)          COMMC
      COMMON      FAT(8)         ,                                         COMMC
     *            FEE(6)         ,    FEED(6)        ,    FFF(3)         , COMMC
     *            FFJ(3)         ,    FLAG1          ,    FLAG2          , COMMC
     *            FLAG3          ,    FLAG4          ,    FN             , COMMC
     *            F0(3)          ,    F01(3)         ,    F02(3)         , COMMC
     *            F03(3)         ,    F1(3)          ,    F11(3)         , COMMC
     *            FPT(5)         ,                                         COMMC
     *            F12(3)         ,    F13(3)                               COMMC
      COMMON      GAIN(10)       ,    G3             ,                     COMMC
     *            G3DOT          ,    G4             ,    G4DOT            COMMC
      COMMON      H(3)           ,    HCMG(3)        ,    HDOT(3)        , COMMC
     *            HI(3)          ,    H0(3)          ,    HW(6)          , COMMC
     *            H1(3)          ,    H1PDOT(3)      ,    H1PRIM(3)      , COMMC
     *            H3PRIM(3)      ,    H4PRIM(3)                            COMMC
      COMMON      IB2F           ,    ICFA           ,    ICFB           , COMMC
     *            ICFC           ,    ICFD           ,    IDOCK          , COMMC
     *            IDOF(6)        ,                                         COMMC
     *            IGRAVF         ,    IPNDLM         ,    IPNTCK         , COMMC
     *            IPRINT         ,    IPROPF                               COMMC
      COMMON      NCASE          ,    NCHECK         ,    NDECK          , COMMC
     *            NGAIN          ,    NUMCMG                               COMMC
      COMMON      OMEGA1         ,    OMEGA3         ,    OMEGA4           COMMC
      COMMON      PEND3L         ,    PEND4L                               COMMC
      COMMON      Q(4,4)                                                   COMMC
      COMMON      R              ,     R0(3)          ,     R1(3)          , COMMC
     *            R1DOT(3)       ,     R1YCS          ,     R1YSN          , COMMC
     *            R1ZCS          ,     R1ZSN          ,     R2(3)          , COMMC
     *            R2DOT(3)       ,     R2YCS          ,     R2YSN          , COMMC
     *            R2ZCS          ,     R2ZSN          ,     R3(3)          , COMMC
     *            R3DOT(3)       ,     R3YCS          ,     R3YSN          , COMMC
     *            R3ZCS          ,     R3ZSN          ,     R4(3)          , COMMC
     *            R4DOT(3)       ,     R4YCS          ,     R4YSN          , COMMC
     *            R4ZCS          ,     R4ZSN                                 COMMC
      COMMON      S              ,     SDOT           ,     SINFEJ         , COMMC
     *            SINTTJ         ,     SINTT0         ,     SINTT1         , COMMC
     *            SINTT2         ,     SINTT3         ,     SINTT4         , COMMC
     *            SP             ,     SUM1           ,     SUM2           , COMMC
     *            SUM3           ,     S2(3)          ,     S3(3)          , COMMC
     *            S4(3)                                                      COMMC
      COMMON      T(3,3)         ,     TC(3,3)        ,     TEMP1(3)       , COMMC
     *            TEMP2(3)       ,                                           COMMC
     *            TEMP3(3)       ,     TEMP4(3)       ,     TEMP5(3,3)     , COMMC
     *            TEMP6(3,3)     ,     TEMP7(3,3)     ,     TEMP8(3,3)     , COMMC
     *            TEMP9(3,3)     ,     TEMP10(3,3)    ,     TEMP11(3,3)    , COMMC
     *            TEMP12(3,3)    ,     TEMP13(3,3)    ,     TEMP14(3,3)    , COMMC
     *            TEMP15(3,3)    ,     TERM1(3)       ,     TERM2(3)       , COMMC
     *            TFRICT         ,     THATA(6)       ,     THATAD(6)      , COMMC
     *            THETA1         ,     THETA3         ,     THETA4         , COMMC
     *            THETO          ,     TIBO(3,3)      ,     TIBOI(3,3)     , COMMC
     *            TIME           ,     TJ             ,     TJ1(10)        , COMMC
     *            TJ2(10)        ,     TJ3(10)        ,     TJ4(10)        , COMMC
     *            TMOTOR         ,                                           COMMC
     *            TOEF(3)        ,     TOTMAS         ,     TO1            , COMMC
     *            TQ0G(3)        ,     TQ0P(3)        ,     TQ1G(3)        , COMMC
     *            TQ1P(3)        ,     TSTART         ,     TSTOP          , COMMC
     *            TT1DOT         ,     TT3DOT         ,     TT4DOT         , COMMC
     *            T1EF(3)        ,     T13            ,     T14              COMMC
      COMMON      V(3)                                                       COMMC
      COMMON      W0(3)          ,     WS             ,     W1(3)          , COMMC
     *            W3(3)          ,     W4(3)                                 COMMC
      COMMON      X(6,7)         ,     XC             ,     XCDOT          , COMMC
     *            XMU                                                        COMMC
C                                                                            COMMC
C     ************************************************************************
C     *
C     *
C     *
C     *
C     ************************************************************************
C                                                                            CMG
      FFF(1) = 0.                                                            CMG
      FFF(2) = 0.                                                            CMG
      FFF(3) = 0.                                                            CMG
      DO 5 M=1,3                                                             CMG
      DO 5 N=1,3                                                             CMG
      EEE(M,N) = 0.                                                          CMG
    5 CONTINUE                                                               CMG
      J = 0                                                                  CMG
   10 J = J + 1                                                              CMG
      IF(J .GT. NUMCMG) RETURN                                               CMG
      IF(IDOF(J) .NE. 0) GO TO 20                                            CMG
      FFJ(1) = A0CJ(J,1,3)*HW(J)                                             CMG
      FFJ(2) = A0CJ(J,2,3)*HW(J)                                             CMG
      FFJ(3) = A0CJ(J,3,3)*HW(J)                                             CMG
      DO 15 M=1,3                                                            CMG
      DO 15 N=1,3                                                            CMG
      EEJ(M,N) = 0.                                                          CMG
   15 CONTINUE                                                           CMG
      GO TO 65                                                           CMG
   20 IF(IDOF(J) .NE. 1) GO TO 35                                        CMG
      SINTTJ = SIN(THATA(J))                                             CMG
      COSTTJ = COS(THATA(J))                                             CMG
      SINFEJ = SIN(FEE(J))                                               CMG
      COSFEJ = COS(FEE(J))                                               CMG
      TEMP12(1) = COSTTJ*AII(J,1,2)*THATAD(J)                            CMG
     *            +SINTTJ*AII(J,3,2)*THATAD(J) + HW(J)*SINTTJ            CMG
      TEMP12(2) = SINFEJ*SINTTJ*AII(J,1,2)*THATAD(J)                     CMG
     *            +COSFEJ*AII(J,2,2)*THATAD(J)                           CMG
     *            -SINFEJ*COSTTJ*AII(J,3,2)*THATAD(J)                    CMG
      TEMP12(3) =-SINTTJ*COSFEJ*AII(J,1,2)*THATAD(J)                     CMG
     *            +SINFEJ*AII(J,2,2)*THATAD(J)                           CMG
     *            +COSFEJ*COSTTJ*AII(J,3,2)*THATAD(J) + HW(J)*COSTTJ     CMG
      DO 22 M=1,3                                                        CMG
      FFJ(M) = AOCJ(J,M,1)*TEMP12(1) + AOCJ(J,M,2)*TEMP12(2)             CMG
     *         +AOCJ(J,M,3)*TEMP12(3)                                    CMG
   22 CONTINUE                                                           CMG
      TEMP13(1,1) = COSTTJ*AOCJ(J,1,1) + SINFEJ*SINTTJ*AOCJ(J,1,2)       CMG
     *             -SINTTJ*COSFEJ*AOCJ(J,1,3)                            CMG
      TEMP13(1,2) = COSTTJ*AOCJ(J,2,1) + SINFEJ*SINTTJ*AOCJ(J,2,2)       CMG
     *             -SINTTJ*COSFEJ*AOCJ(J,2,3)                            CMG
      TEMP13(1,3) = COSTTJ*AOCJ(J,3,1) + SINFEJ*SINTTJ*AOCJ(J,3,2)       CMG
     *             -SINTTJ*COSFEJ*AOCJ(J,3,3)                            CMG
      TEMP13(2,1) = COSFEJ*AOCJ(J,1,2) + SINFEJ*AOCJ(J,1,3)              CMG
      TEMP13(2,2) = COSFEJ*AOCJ(J,2,2) + SINFEJ*AOCJ(J,2,3)              CMG
      TEMP13(2,3) = COSFEJ*AOCJ(J,3,2) + SINFEJ*AOCJ(J,3,3)              CMG
      TEMP13(3,1) = SINTTJ*AOCJ(J,1,1) - SINFEJ*COSTTJ*AOCJ(J,1,2)       CMG
     *             +COSFEJ*COSTTJ*AOCJ(J,1,3)                            CMG
      TEMP13(3,2) = SINTTJ*AOCJ(J,2,1) - SINFEJ*COSTTJ*AOCJ(J,2,2)       CMG
     *             +COSFEJ*COSTTJ*AOCJ(J,2,3)                            CMG
      TEMP13(3,3) = SINTTJ*AOCJ(J,3,1) - SINFEJ*COSTTJ*AOCJ(J,3,2)       CMG
     *             +COSFEJ*COSTTJ*AOCJ(J,3,3)                            CMG
      DO 25 M=1,3                                                        CMG
      DO 25 N=1,3                                                        CMG
      TEMP14(M,N) = AII(J,M,1)*TEMP13(1,N) + AII(J,M,2)*TEMP13(2,N)      CMG
     *             +AII(J,M,3)*TEMP13(3,N)                               CMG
   25 CONTINUE                                                           CMG
      TEMP15(1,1) =  COSTTJ*TEMP14(1,1) + SINTTJ*TEMP14(3,1)             CMG
      TEMP15(1,2) =  COSTTJ*TEMP14(1,2) + SINTTJ*TEMP14(3,2)             CMG
      TEMP15(1,3) =  COSTTJ*TEMP14(1,3) + SINTTJ*TEMP14(3,3)             CMG
      TEMP15(2,1) =  SINFEJ*SINTTJ*TEMP14(1,1) + COSFEJ*TEMP14(2,1)      CMG
     *              -SINFEJ*COSTTJ*TEMP14(3,1)                           CMG
      TEMP15(2,2) =  SINFEJ*SINTTJ*TEMP14(1,2) + COSFEJ*TEMP14(2,2)      CMG
     *              -SINFEJ*COSTTJ*TEMP14(3,2)                           CMG
      TEMP15(2,3) =  SINFEJ*SINTTJ*TEMP14(1,3) + COSFEJ*TEMP14(2,3)      CMG
     *              -SINFEJ*COSTTJ*TEMP14(3,3)                           CMG
      TEMP15(3,1) = -SINTTJ*COSFEJ*TEMP14(1,1) + SINFEJ*TEMP14(2,1)      CMG
     *              +COSFEJ*COSTTJ*TEMP14(3,1)                           CMG
      TEMP15(3,2) = -SINTTJ*COSFEJ*TEMP14(1,2) + SINFEJ*TEMP14(2,2)      CMG
     *              +COSFEJ*COSTTJ*TEMP14(3,2)                           CMG
      TEMP15(3,3) = -SINTTJ*COSFEJ*TEMP14(1,3) + SINFEJ*TEMP14(2,3)      CMG
     *              +COSFEJ*COSTTJ*TEMP14(3,3)                           CMG
      DO 30 M=1,3                                                        CMG
      DO 30 N=1,3                                                        CMG
      EEJ(M,N) = AOCJ(J,M,1)*TEMP15(1,N) + AOCJ(J,M,2)*TEMP15(2,N)       CMG
     *          +AOCJ(J,M,3)*TEMP15(3,N)                                 CMG
   30 CONTINUE                                                           CMG
      GO TO 60                                                           CMG
   35 CONTINUE                                                           CMG
      SINTTJ = SIN(THATA(J))                                             CMG
      COSTTJ = COS(THATA(J))                                           CMG
      SINFEJ = SIN(FEE(J))                                             CMG
      COSFEJ = COS(FEE(J))                                             CMG
      TEMP1(1) = AIO(J,1,1)*FEED(J)                                    CMG
      TEMP1(2) = COSFEJ*AIO(J,2,1)*FEED(J) - SINFEJ*AIO(J,3,1)*FEED(J) CMG
      TEMP1(3) = SINFEJ*AIO(J,2,1)*FEED(J) + COSFEJ*AIO(J,3,1)*FEED(J) CMG
      TEMP2(1)=COSTTJ*(AII(J,1,1)*COSTTJ*FEED(J)+AII(J,1,2)*THATAD(J)+ CMG
     * AII(J,1,3)*SINTTJ*FEED(J))+SINTTJ*(AII(J,3,1)*COSTTJ*FEED(J)+   CMG
     * AII(J,3,2)*THATAD(J)+AII(J,3,3)*SINTTJ*FEED(J))                 CMG
      TEMP2(2)=SINFEJ*SINTTJ*(AII(J,1,1)*COSTTJ*FEED(J)+               CMG
     * AII(J,1,2)*THATAD(J)+AII(J,1,3)*SINTTJ*FEED(J))+                CMG
     * COSFEJ*(AII(J,2,1)*COSTTJ*FEED(J)+AII(J,2,2)*THATAD(J)+         CMG
     * AII(J,2,3)*SINTTJ*FEED(J))-                                     CMG
     * SINFEJ*COSTTJ*(AII(J,3,1)*COSTTJ*FEED(J)+AII(J,3,2)*THATAD(J)+  CMG
     * AII(J,3,3)*SINTTJ*FEED(J))                                      CMG
      TEMP2(3)=-SINTTJ*COSFEJ*(AII(J,1,1)*COSTTJ*FEED(J)+              CMG
     * AII(J,1,2)*THATAD(J)+AII(J,1,3)*SINTTJ*FEED(J))+                CMG
     * SINFEJ*(AII(J,2,1)*COSTTJ*FEED(J)+AII(J,2,2)*THATAD(J)+         CMG
     * AII(J,2,3)*SINTTJ*FEED(J))+                                     CMG
     * COSFEJ*COSTTJ*(AII(J,3,1)*COSTTJ*FEED(J)+AII(J,3,2)*THATAD(J)+  CMG
     * AII(J,3,3)*SINTTJ*FEED(J))                                      CMG
      TEMP3(1) =  HW(J)*SINTTJ                                         CMG
      TEMP3(2) = -HW(J)*COSTTJ*SINFEJ                                  CMG
      TEMP3(3) =  HW(J)*COSTTJ*COSFEJ                                  CMG
      TEMP4(1) = TEMP1(1) + TEMP2(1) + TEMP3(1)                        CMG
      TEMP4(2) = TEMP1(2) + TEMP2(2) + TEMP3(2)                        CMG
      TEMP4(3) = TEMP1(3) + TEMP2(3) + TEMP3(3)                        CMG
      FFJ(1) = AOCJ(J,1,1)*TEMP4(1)                                    CMG
     *        +AOCJ(J,1,2)*TEMP4(2) + AOCJ(J,1,3)*TEMP4(3)             CMG
      FFJ(2) = AOCJ(J,2,1)*TEMP4(1)                                    CMG
     *        +AOCJ(J,2,2)*TEMP4(2) + AOCJ(J,2,3)*TEMP4(3)             CMG
      FFJ(3) = AOCJ(J,3,1)*TEMP4(1)                                    CMG
     *        +AOCJ(J,3,2)*TEMP4(2) + AOCJ(J,3,3)*TEMP4(3)             CMG
      TEMP5(1,1) =  AOCJ(J,1,1)                                        CMG
      TEMP5(1,2) =  AOCJ(J,2,1)                                        CMG
      TEMP5(1,3) =  AOCJ(J,3,1)                                        CMG
      TEMP5(2,1) =  COSFEJ*AOCJ(J,1,2) + SINFEJ*AOCJ(J,1,3)            CMG
      TEMP5(2,2) =  COSFEJ*AOCJ(J,2,2) + SINFEJ*AOCJ(J,2,3)            CMG
      TEMP5(2,3) =  COSFEJ*AOCJ(J,3,2) + SINFEJ*AOCJ(J,3,3)            CMG
      TEMP5(3,1) = -SINFEJ*AOCJ(J,1,2) + COSFEJ*AOCJ(J,1,3)            CMG
      TEMP5(3,2) = -SINFEJ*AOCJ(J,2,2) + COSFEJ*AOCJ(J,2,3)            CMG
      TEMP5(3,3) = -SINFEJ*AOCJ(J,3,2) + COSFEJ*AOCJ(J,3,3)            CMG
      DO 40 M=1,3                                                      CMG
      DO 40 N=1,3                                                      CMG
      TEMP6(M,N) = AIO(J,M,1)*TEMP5(1,N) + AIO(J,M,2)*TEMP5(2,N)       CMG
     *            +AIO(J,M,3)*TEMP5(3,N)                               CMG
   40 CONTINUE                                                         CMG
      TEMP7(1,1) =  TEMP6(1,1)                                         CMG
      TEMP7(1,2) =  TEMP6(1,2)                                         CMG
      TEMP7(1,3) =  TEMP6(1,3)                                         CMG
      TEMP7(2,1) =  COSFEJ*TEMP6(2,1) - SINFEJ*TEMP6(3,1)              CMG
      TEMP7(2,2) =  COSFEJ*TEMP6(2,2) - SINFEJ*TEMP6(3,2)              CMG
      TEMP7(2,3) =  COSFEJ*TEMP6(2,3) - SINFEJ*TEMP6(3,3)              CMG
      TEMP7(3,1) =  SINFEJ*TEMP6(2,1) + COSFEJ*TEMP6(3,1)              CMG
      TEMP7(3,2) =  SINFEJ*TEMP6(2,2) + COSFEJ*TEMP6(3,2)              CMG
      TEMP7(3,3) =  SINFEJ*TEMP6(2,3) + COSFEJ*TEMP6(3,3)              CMG
      TEMP8(1,1) = COSTTJ*AOCJ(J,1,1) + SINFEJ*SINTTJ*AOCJ(J,1,2)      CMG
     *            -SINTTJ*COSFEJ*AOCJ(J,1,3)                           CMG
      TEMP8(1,2) = COSTTJ*AOCJ(J,2,1) + SINFEJ*SINTTJ*AOCJ(J,2,2)      CMG
     *            -SINTTJ*COSFEJ*AOCJ(J,2,3)                           CMG
      TEMP8(1,3) = COSTTJ*AOCJ(J,3,1) + SINFEJ*SINTTJ*AOCJ(J,3,2)      CMG
     *            -SINTTJ*COSFEJ*AOCJ(J,3,3)                           CMG
```

```
      TEMP8(2,1) = COSFEJ*AOCJ(J,1,2) + SINFEJ*AOCJ(J,1,3)                CMG
      TEMP8(2,2) = COSFEJ*AOCJ(J,2,2) + SINFEJ*AOCJ(J,2,3)                CMG
      TEMP8(2,3) = COSFEJ*AOCJ(J,3,2) + SINFEJ*AOCJ(J,3,3)                CMG
      TEMP8(3,1) = SINTTJ*AOCJ(J,1,1) - SINFEJ*COSTTJ*AOCJ(J,1,2)         CMG
     *             +COSFEJ*COSTTJ*AOCJ(J,1,3)                             CMG
      TEMP8(3,2) = SINTTJ*AOCJ(J,2,1) - SINFEJ*COSTTJ*AOCJ(J,2,2)         CMG
     *             +COSFEJ*COSTTJ*AOCJ(J,2,3)                             CMG
      TEMP8(3,3) = SINTTJ*AOCJ(J,3,1) - SINFEJ*COSTTJ*AOCJ(J,3,2)         CMG
     *             +COSFEJ*COSTTJ*AOCJ(J,3,3)                             CMG
      DO 45 M=1,3                                                         CMG
      DO 45 N=1,3                                                         CMG
      TEMP9(M,N) = AII(J,M,1)*TEMP8(1,N) + AII(J,M,2)*TEMP8(2,N)          CMG
     *             +AII(J,M,3)*TEMP8(3,N)                                 CMG
   45 CONTINUE                                                            CMG
      TEMP10(1,1) = COSTTJ*TEMP9(1,1) + SINTTJ*TEMP9(3,1)                 CMG
      TEMP10(1,2) = COSTTJ*TEMP9(1,2) + SINTTJ*TEMP9(3,2)                 CMG
      TEMP10(1,3) = COSTTJ*TEMP9(1,3) + SINTTJ*TEMP9(3,3)                 CMG
      TEMP10(2,1) = SINFEJ*SINTTJ*TEMP9(1,1) + COSFEJ*TEMP9(2,1)          CMG
     *              -SINFEJ*COSTTJ*TEMP9(3,1)                             CMG
      TEMP10(2,2) = SINFEJ*SINTTJ*TEMP9(1,2) + COSFEJ*TEMP9(2,2)          CMG
     *              -SINFEJ*COSTTJ*TEMP9(3,2)                             CMG
      TEMP10(2,3) = SINFEJ*SINTTJ*TEMP9(1,3) + COSFEJ*TEMP9(2,3)          CMG
     *              -SINFEJ*COSTTJ*TEMP9(3,3)                             CMG
      TEMP10(3,1) =-SINTTJ*COSFEJ*TEMP9(1,1) + SINFEJ*TEMP9(2,1)          CMG
     *              +COSFEJ*COSTTJ*TEMP9(3,1)                             CMG
      TEMP10(3,2) =-SINTTJ*COSFEJ*TEMP9(1,2) + SINFEJ*TEMP9(2,2)          CMG
     *              +COSFEJ*COSTTJ*TEMP9(3,2)                             CMG
      TEMP10(3,3) =-SINTTJ*COSFEJ*TEMP9(1,3) + SINFEJ*TEMP9(2,3)          CMG
     *              +COSFEJ*COSTTJ*TEMP9(3,3)                             CMG
      DO 50 M=1,3                                                         CMG
      DO 50 N=1,3                                                         CMG
      TEMP11(M,N) = TEMP7(M,N) + TEMP10(M,N)                              CMG
   50 CONTINUE                                                            CMG
      DO 55 M=1,3                                                         CMG
      DO 55 N=1,3                                                         CMG
      EEJ(M,N) = AOCJ(J,M,1)*TEMP11(1,N) + AOCJ(J,M,2)*TEMP11(2,N)        CMG
     *           +AOCJ(J,M,3)*TEMP11(3,N)                                 CMG
   55 CONTINUE                                                            CMG
   60 THATA(J) = THATA(J) + THATAD(J)*DELTAT                              CMG
      FEE(J) = FEE(J) + FEED(J)*DELTAT                                    CMG
   65 CONTINUE                                                            CMG
      DO 70 M=1,3                                                         CMG
      FFF(M) = FFF(M) + FFJ(M)                                            CMG
   70 CONTINUE                                                            CMG
      DO 75 M=1,3                                                         CMG
      DO 75 N=1,3                                                         CMG
      EEE(M,N) = EEE(M,N) + EEJ(M,N)                                      CMG
   75 CONTINUE                                                            CMG
      GO TO 10                                                            CMG
C                                                                         CMG
C     ********************************************************************
C     *
C     *
C     *
      END
      SUBROUTINE EMCALC
C     *
C     *
C     *
C     ***********************************************************************
      COMMON    A(3)          ,     AE(5)         ,     AED(5)        ,  COMMC
     *          AFOUR(2)      ,     AII(6,3,3)    ,                      COMMC
     *          AIO(6,3,3)    ,     AJ1(3)        ,     ALT           ,  COMMC
     *          AOCJ(6,3,3)   ,     AOJ(3)        ,     AONE(7)       ,  COMMC
     *          ATCPT2(3,3)   ,     ATHREE(5)     ,     ATWO(4)       ,  COMMC
     *          A1(3,3)       ,     A1J(2)                               COMMC
      COMMON    BDMASS        ,     BFOUR(2)      ,     BMOM          ,  COMMC
     *          BODYDI(3,3)   ,     BODYOI(3,3)   ,     BODY1I(3,3)   ,  COMMC
     *          BOMASS        ,     BONE(7)       ,     BTHREE(5)     ,  COMMC
     *          BTWO(4)       ,     B1MASS        ,     B2MASS        ,  COMMC
     *          B3MASS        ,     B4MASS                               COMMC
      COMMON    CA(3)         ,     CB(3)         ,     CGAINO(3)     ,  COMMC
     *          CGAIN1(2)     ,                                          COMMC
     *          COSFEJ        ,     COSTTJ        ,     COSTTO        ,  COMMC
     *          COSTT1        ,     COSTT3        ,     COSTT4        ,  COMMC
     *          CO2T          ,     CP1           ,     CP2           ,  COMMC
     *          CST           ,     C1                                   COMMC
      COMMON    DB(3)         ,     DD01(3)       ,                        COMMC
     *          DELTAT        ,     D01(3)        ,     D01DOT(3)     ,  COMMC
     *          DTI(3)        ,                                          COMMC
     *          DTIME         ,     D12(3)        ,     D13(3)        ,  COMMC
     *          D13DOT(3)     ,     D13YCS        ,     D13YSN        ,  COMMC
     *          D13ZCS        ,     D13ZSN        ,     D14(3)        ,  COMMC
     *          D14DOT(3)     ,     D14YCS        ,     D14YSN        ,  COMMC
     *          D14ZCS        ,     D14ZSN                               COMMC
      COMMON    EEE(3,3)      ,     EEJ(3,3)      ,     EL2(3)        ,  COMMC
     *          EL2DOT(3)     ,     EL2YCS        ,     EL2YSN        ,  COMMC
     *          EL2ZCS        ,     EL2ZSN        ,     EL3(3)        ,  COMMC
     *          EL3DOT(3)     ,     EL3YCS        ,     EL3YSN        ,  COMMC
     *          EL3ZCS        ,     EL3ZSN        ,     EL4(3)        ,  COMMC
     *          EL4DOT(3)     ,     EL4YCS        ,     EL4YSN        ,  COMMC
     *          EL4ZCS        ,     EL4ZSN        ,     EM(6,6)          COMMC
      COMMON    FAT(8)        ,                                          COMMC
     *          FEE(6)        ,     FEED(6)       ,     FFF(3)        ,  COMMC
     *          FFJ(3)        ,     FLAG1         ,     FLAG2         ,  COMMC
     *          FLAG3         ,     FLAG4         ,     FN            ,  COMMC
     *          FO(3)         ,     FO1(3)        ,     FO2(3)        ,  COMMC
     *          FO3(3)        ,     F1(3)         ,     F11(3)        ,  COMMC
     *          FPT(5)        ,                                          COMMC
     *          F12(3)        ,     F13(3)                               COMMC
      COMMON    GAIN(10)      ,     G3            ,                        COMMC
     *          G3DOT         ,     G4            ,     G4DOT            COMMC
      COMMON    H(3)          ,     HCMG(3)       ,     HDOT(3)       ,  COMMC
     *          HI(3)         ,     HO(3)         ,     HW(6)         ,  COMMC
     *          H1(3)         ,     H1PDOT(3)     ,     H1PRIM(3)     ,  COMMC
     *          H3PRIM(3)     ,     H4PRIM(3)                            COMMC
      COMMON    IB2F          ,     ICFA          ,     ICFB          ,  COMMC
     *          ICFC          ,     ICFD          ,     IDOCK         ,  COMMC
     *          IDOF(6)       ,                                          COMMC
     *          IGRAVF        ,     IPNDLM        ,     IPNTCK        ,  COMMC
     *          IPRINT        ,     IPROPF                               COMMC
      COMMON    NCASE         ,     NCHECK        ,     NDECK         ,  COMMC
     *          NGAIN         ,     NUMCMG                               COMMC
      COMMON    OMEGA1        ,     OMEGA3        ,     OMEGA4           COMMC
      COMMON    PEND3L        ,     PEND4L                               COMMC
      COMMON    Q(4,4)                                                   COMMC
      COMMON    R              ,    R0(3)          ,    R1(3)          , COMMC
     *          R1DOT(3)       ,    R1YCS          ,    R1YSN          , COMMC
     *          R1ZCS          ,    R1ZSN          ,    R2(3)          , COMMC
     *          R2DOT(3)       ,    R2YCS          ,    R2YSN          , COMMC
     *          R2ZCS          ,    R2ZSN          ,    R3(3)          , COMMC
     *          R3DOT(3)       ,    R3YCS          ,    R3YSN          , COMMC
     *          R3ZCS          ,    R3ZSN          ,    R4(3)          , COMMC
     *          R4DOT(3)       ,    R4YCS          ,    R4YSN          , COMMC
     *          R4ZCS          ,    R4ZSN                                COMMC
      COMMON    S              ,    SDOT           ,    SINFEJ         , COMMC
     *          SINTTJ         ,    SINTT0         ,    SINTT1         , COMMC
     *          SINTT2         ,    SINTT3         ,    SINTT4         , COMMC
     *          SP             ,    SUM1           ,    SUM2           , COMMC
     *          SUM3           ,    S2(3)          ,    S3(3)          , COMMC
     *          S4(3)                                                    COMMC
      COMMON    T(3,3)         ,    TC(3,3)        ,    TEMP1(3)       , COMMC
     *          TEMP2(3)       ,                                         COMMC
     *          TEMP3(3)       ,    TEMP4(3)       ,    TEMP5(3,3)     , COMMC
     *          TEMP6(3,3)     ,    TEMP7(3,3)     ,    TEMP8(3,3)     , COMMC
     *          TEMP9(3,3)     ,    TEMP10(3,3)    ,    TEMP11(3,3)    , COMMC
     *          TEMP12(3,3)    ,    TEMP13(3,3)    ,    TEMP14(3,3)    , COMMC
     *          TEMP15(3,3)    ,    TERM1(3)       ,    TERM2(3)       , COMMC
     *          TFRICT         ,    THATA(6)       ,    THATAD(6)      , COMMC
     *          THETA1         ,    THETA3         ,    THETA4         , COMMC
     *          THETO          ,    TIBO(3,3)      ,    TIBOI(3,3)     , COMMC
     *          TIME           ,    TJ             ,    TJ1(10)        , COMMC
     *          TJ2(10)        ,    TJ3(10)        ,    TJ4(10)        , COMMC
     *          TMOTOR         ,                                         COMMC
     *          TOEF(3)        ,    TOTMAS         ,    TO1            , COMMC
     *          TQOG(3)        ,    TQOP(3)        ,    TQ1G(3)        , COMMC
     *          TQ1P(3)        ,    TSTART         ,    TSTOP          , COMMC
     *          TT1DOT         ,    TT3DOT         ,    TT4DOT         , COMMC
     *          T1EF(3)        ,    T13            ,    T14              COMMC
      COMMON    V(3)                                                     COMMC
      COMMON    W0(3)          ,    WS             ,    W1(3)          , COMMC
     *          W3(3)          ,    W4(3)                                COMMC
      COMMON    X(6,7)         ,    XC             ,    XCDOT          , COMMC
     *          XMU                                                      COMMC
C                                                                        COMMC
C     ********************************************************************
C     *
C     *
C     *
C     ********************************************************************
C                                                                        EMCAL
C     THE FOLLOWING CALCULATIONS WILL BE USED REPEATEDLY TO CALCULATE M  EMCAL
C                                                                        EMCAL
      SINTT1 = SIN(THETA1)                                               EMCAL
      COSTT1 = COS(THETA1)                                               EMCAL
      CO2T = COSTT1**2                                                   EMCAL
      CST = COSTT1*SINTT1                                                EMCAL
      SI2T = SINTT1**2                                                   EMCAL
      R1YCS = R1(2)*COSTT1                                               EMCAL
      R2YCS = R2(2)*COSTT1                                               EMCAL
      R3YCS = R3(2)*COSTT1                                               EMCAL
      R4YCS = R4(2)*COSTT1                                               EMCAL
      R1ZCS = R1(3)*COSTT1                                               EMCAL
      R2ZCS = R2(3)*COSTT1                                               EMCAL
      R3ZCS = R3(3)*COSTT1                                               EMCAL
      R4ZCS = R4(3)*COSTT1                                               EMCAL
      R1YSN = R1(2)*SINTT1                                               EMCAL
      R2YSN = R2(2)*SINTT1                                               EMCAL
      R3YSN = R3(2)*SINTT1                                              EMCAL
      R4YSN = R4(2)*SINTT1                                              EMCAL
      R1ZSN = R1(3)*SINTT1                                              EMCAL
      R2ZSN = R2(3)*SINTT1                                              EMCAL
      R3ZSN = R3(3)*SINTT1                                              EMCAL
      R4ZSN = R4(3)*SINTT1                                              EMCAL
      EL2YCS = EL2(2)*COSTT1                                            EMCAL
      EL3YCS = EL3(2)*COSTT1                                            EMCAL
      EL4YCS = EL4(2)*COSTT1                                            EMCAL
      EL2ZCS = EL2(3)*COSTT1                                            EMCAL
      EL3ZCS = EL3(3)*COSTT1                                            EMCAL
      EL4ZCS = EL4(3)*COSTT1                                            EMCAL
      EL2YSN = EL2(2)*SINTT1                                            EMCAL
      EL3YSN = EL3(2)*SINTT1                                            EMCAL
      EL4YSN = EL4(2)*SINTT1                                            EMCAL
      EL2ZSN = EL2(3)*SINTT1                                            EMCAL
      EL3ZSN = EL3(3)*SINTT1                                            EMCAL
      EL4ZSN = EL4(3)*SINTT1                                            EMCAL
      D13YCS = D13(2)*COSTT1                                            EMCAL
      D14YCS = D14(2)*COSTT1                                            EMCAL
      D13ZCS = D13(3)*COSTT1                                            EMCAL
      D14ZCS = D14(3)*COSTT1                                            EMCAL
      D13YSN = D13(2)*SINTT1                                            EMCAL
      D14YSN = D14(2)*SINTT1                                            EMCAL
      D13ZSN = D13(3)*SINTT1                                            EMCAL
      D14ZSN = D14(3)*SINTT1                                            EMCAL
      BMOM=BOMASS/TOTMAS                                                EMCAL
      EM(1,1)=BODYOI(1,1)+BODY1I(1,1)+B1MASS*D01(2)*(R1YCS-R1ZSN)+      EMCAL
     * B1MASS*D01(3)*(R1YSN+R1ZCS)+B2MASS*(R2YCS-R2ZSN)*(D01(2)+EL2YCS- EMCAL
     * EL2ZSN)+B2MASS*(R2YSN+R2ZCS)*(D01(3)+EL2YSN+EL2ZCS)+B3MASS*(R3YCSEMCAL
     * -R3ZSN)*(D01(2)+D13YCS-D13ZSN+EL3YCS-EL3ZSN)+B3MASS*(R3YSN+R3ZCS)EMCAL
     * *(D01(3)+D13YSN+D13ZCS+EL3YSN+EL3ZCS)+B4MASS*(R4YCS-R4ZSN)*      EMCAL
     * (D01(2)+D14YCS-D14ZSN+EL4YCS-EL4ZSN)+B4MASS*(R4YSN+R4ZCS)*       EMCAL
     * (D01(3)+D14YSN+D14ZCS+EL4YSN+EL4ZCS)                             EMCAL
C                                                                       EMCAL
      EM(1,2)=BODYOI(1,2)+BODY1I(1,2)*COSTT1-BODY1I(1,3)*SINTT1         EMCAL
     *        -B1MASS*R1(1)*D01(2)-                                     EMCAL
     * B2MASS*R2(1)*(D01(2)+EL2YCS-EL2ZSN)-                             EMCAL
     * B3MASS*R3(1)*(D01(2)+D13YCS-D13ZSN+EL3YCS-EL3ZSN)-               EMCAL
     * B4MASS*R4(1)*(D01(2)+D14YCS-D14ZSN+EL4YCS-EL4ZSN)                EMCAL
C                                                                       EMCAL
      EM(1,3)=BODYOI(1,3)+BODY1I(1,2)*SINTT1+BODY1I(1,3)*COSTT1         EMCAL
     *        -B1MASS*R1(1)*D01(3)-                                     EMCAL
     * B2MASS*R2(1)*(D01(3)+EL2YSN+EL2ZCS)-                             EMCAL
     * B3MASS*R3(1)*(D01(3)+D13YSN+D13ZCS+EL3YSN+EL3ZCS)-               EMCAL
     * B4MASS*R4(1)*(D01(3)+D14YSN+D14ZCS+EL4YSN+EL4ZCS)                EMCAL
C                                                                       EMCAL
      EM(1,4)=BODY1I(1,1)-B1MASS*D01(3)*(BMOM*D01(3)-R1YSN-R1ZCS)+      EMCAL
     * B1MASS*D01(2)*(R1YCS-R1ZSN-BMOM*D01(2))-                         EMCAL
     * B2MASS*(D01(3)+EL2YSN+EL2ZCS)*(BMOM*D01(3)-R2YSN-R2ZCS)+         EMCAL
     * B2MASS*(D01(2)+EL2YCS-EL2ZSN)*(R2YCS-R2ZSN-BMOM*D01(2))-         EMCAL
     * B3MASS*(D01(3)+(D13(2)+EL3(2))*SINTT1+(D13(3)+EL3(3))*COSTT1)*   EMCAL
     * (BMOM*D01(3)-R3YSN-R3ZCS)+                                       EMCAL
     * B3MASS*(D01(2)+(D13(2)+EL3(2))*COSTT1-(D13(3)+EL3(3))*SINTT1)*   EMCAL
     * (R3YCS-R3ZSN-BMOM*D01(2))-                                       EMCAL
     * B4MASS*(D01(3)+(D14(2)+EL4(2))*SINTT1+(D14(3)+EL4(3))*COSTT1)*   EMCAL
     * (BMOM*D01(3)-R4YSN-R4ZCS)+                                       EMCAL
     * B4MASS*(D01(2)+(D14(2)+EL4(2))*COSTT1-(D14(3)+EL4(3))*SINTT1)*   EMCAL
     * (R4YCS-R4ZSN-BMOM*D01(2))                                        EMCAL
C                                                                       EMCAL
      EM(1,5)=-B3MASS*R3(3)*(S3(3)*EL3(1)-S3(1)*EL3(3))+                EMCAL
     * B3MASS*R3(2)*(-S3(2)*EL3(1)+S3(1)*EL3(2))                        EMCAL
```

```
C
      EM(1,6)=-B4MASS*R4(3)*(S4(3)*EL4(1)-S4(1)*EL4(3))+
     * B4MASS*R4(2)*(-S4(2)*EL4(1)+S4(1)*EL4(2))
C
      EM(2,1)=BODY0I(2,1)+BODY1I(2,1)*COSTT1-BODY1I(3,1)*SINTT1-
     * B1MASS*D01(1)*(R1YCS-R1ZSN)-
     * B2MASS*(R2YCS-R2ZSN)*(D01(1)+EL2(1))-
     * B3MASS*(R3YCS-R3ZSN)*(D01(1)+D13(1)+EL3(1))-
     * B4MASS*(R4YCS-R4ZSN)*(D01(1)+D14(1)+EL4(1))
C
      EM(2,2)=BODY0I(2,2)+BODY1I(2,2)*CO2T -BODY1I(2,3)*CST
     *  -BODY1I(3,2)*CST  + BODY1I(3,3)*SI2T +
     * B1MASS*R1(1)*D01(1)+B1MASS*D01(3)*(R1YSN+R1ZCS)+
     * B2MASS*R2(1)*(D01(1)+EL2(1))+
     * B2MASS*(R2YSN+R2ZCS)*(D01(3)+EL2YSN+EL2ZCS)+
     * B3MASS*R3(1)*(D01(1)+D13(1)+EL3(1))+
     * B3MASS*(R3YSN+R3ZCS)*(D01(3)+D13YSN+D13ZCS+EL3YSN+EL3ZCS)+
     * B4MASS*R4(2)*(D01(1)+D14(1)+EL4(1))+
     * B4MASS*(R4YSN+R4ZCS)*(D01(3)+D14YSN+D14ZCS+EL4YSN+EL4ZCS)
C
      EM(2,3)=BODY0I(2,3)+BODY1I(2,2)*CST  +BODY1I(2,3)*CO2T
     *  -BODY1I(3,2)*SI2T - BODY1I(3,3)*CST  -
     * B1MASS*D01(3)*(R1YCS-R1ZSN)-
     * B2MASS*(R2YCS-R2ZSN)*(D01(3)+EL2YSN+EL2ZCS)-
     * B3MASS*(R3YCS-R3ZSN)*(D01(3)+D13YSN+D13ZCS+EL3YSN+EL3ZCS)-
     * B4MASS*(R4YCS-R4ZSN)*(D01(3)+D14YSN+D14ZCS+EL4YSN+EL4ZCS)
C
      EM(2,4)=BODY1I(1,2)*COSTT1-BODY1I(1,3)*SINTT1-B1MASS*D01(1)*R1YCS+
     * B1MASS*D01(1)*R1ZSN+BMOM*B1MASS*D01(1)*D01(2)-
     * B2MASS*(D01(1)+EL2(1))*(R2YCS-R2ZSN-BMOM*D01(2))-
     * B3MASS*(D01(1)+D13(1)+EL3(1))*(R3YCS-R3ZSN-BMOM*D01(2))-
     * B4MASS*(D01(1)+D14(1)+EL4(1))*(R4YCS-R4ZSN-BMOM*D01(2))
C
      EM(2,5)=(B3MASS*R3(3)*(-S3(3)*EL3(2)+S3(2)*EL3(3))-
     * B3MASS*R3(1)*(-S3(2)*EL3(1)+S3(1)*EL3(2)))*COSTT1-
     * (-B3MASS*R3(2)*(-S3(3)*EL3(2)+S3(2)*EL3(3))+
     * B3MASS*R3(1)*(S3(3)*EL3(1)-S3(1)*EL3(3)))*SINTT1
C
      EM(2,6)=(B4MASS*R4(3)*(-S4(3)*EL4(2)+S4(2)*EL4(3))-
     * B4MASS*R4(1)*(-S4(2)*EL4(1)+S4(1)*EL4(2)))*COSTT1-
     * (-B4MASS*R4(2)*(-S4(3)*EL4(2)+S4(2)*EL4(3))+
     * B4MASS*R4(1)*(S4(3)*EL4(1)-S4(1)*EL4(3)))*SINTT1
C
      EM(3,1)=BODY0I(3,1)+BODY1I(2,1)*SINTT1+BODY1I(3,1)*COSTT1-
     * B1MASS*D01(1)*(R1YSN+R1ZCS)-
     * B2MASS*(D01(1)+EL2(1))*(R2YSN+R2ZCS)-
     * B3MASS*(D01(1)+D13(1)+EL3(1))*(R3YSN+R3ZCS)-
     * B4MASS*(D01(1)+D14(1)+EL4(1))*(R4YSN+R4ZCS)
C
      EM(3,2)=BODY0I(3,2)+BODY1I(2,2)*CST  -BODY1I(2,3)*SI2T
     *  +BODY1I(3,2)*CO2T-BODY1I(3,3)*CST-
     * B1MASS*D01(2)*(R1YSN+R1ZCS)-
     * B2MASS*(R2YSN+R2ZCS)*(D01(2)+EL2YCS-EL2ZSN)-
     * B3MASS*(R3YSN+R3ZCS)*(D01(2)+D13YCS-D13ZSN+EL3YCS-EL3ZSN)-
     * B4MASS*(R4YSN+R4ZCS)*(D01(2)+D14YCS-D14ZSN+EL4YCS-EL4ZSN)
C
      EM(3,3)=BODY0I(3,3)+BODY1I(2,2)*SI2T +BODY1I(2,3)*CST
     *  +BODY1I(3,2)*CST  + BODY1I(3,3)*CO2T +
     * B1MASS*R1(1)*D01(1)+B1MASS*(R1YCS-R1ZSN)*D01(2)+
     * B2MASS*R2(1)*(D01(1)+EL2(1))+
     * B2MASS*(R2YCS-R2ZSN)*(D01(2)+EL2YCS-EL2ZSN)+
     * B3MASS*R3(1)*(D01(1)+D13(1)+EL3(1))+
     * B3MASS*(R3YCS-R3ZSN)*(DO1(2)+D13YCS-D13ZSN+EL3YCS-EL3ZSN)+           EMCAL
     * B4MASS*R4(1)*(DO1(1)+D14(1)+EL4(1))+                                 EMCAL
     * B4MASS*(R4YCS-R4ZSN)*(DO1(2)+D14YCS-D14ZSN+EL4YCS-EL4ZSN)            EMCAL
C                                                                           EMCAL
      EM(3,4)=BODY1I(1,2)*SINTT1+BODY1I(1,3)*COSTT1+                        EMCAL
     * B1MASS*DO1(1)*(BMOM*DO1(3)-R1YSN-R1ZCS)+                             EMCAL
     * B2MASS*(DO1(1)+EL2(1))*(BMOM*DO1(3)-R2YSN-R2ZCS)+                    EMCAL
     * B3MASS*(DO1(1)+D13(1)+EL3(1))*(BMOM*DO1(3)-R3YSN-R3ZCS)+             EMCAL
     * B4MASS*(DO1(1)+D14(1)+EL4(1))*(BMOM*DO1(3)-R4YSN-R4ZCS)              EMCAL
C                                                                           EMCAL
      EM(3,5)=(B3MASS*R3(3)*(-S3(3)*EL3(2)+S3(2)*EL3(3))-                   EMCAL
     * B3MASS*R3(1)*(-S3(2)*EL3(1)+S3(1)*EL3(2)))*SINTT1+                   EMCAL
     * (-B3MASS*R3(2)*(-S3(3)*EL3(2)+S3(2)*EL3(3))+                         EMCAL
     * B3MASS*R3(1)*(S3(3)*EL3(1)-S3(1)*EL3(3)))*COSTT1                     EMCAL
C                                                                           EMCAL
      EM(3,6)=(B4MASS*R4(3)*(-S4(3)*EL4(2)+S4(2)*EL4(3))-                   EMCAL
     * B4MASS*R4(1)*(-S4(2)*EL4(1)+S4(1)*EL4(2)))*SINTT1+                   EMCAL
     * (-B4MASS*R4(2)*(-S4(3)*EL4(2)+S4(2)*EL4(3))+                         EMCAL
     * B4MASS*R4(1)*(S4(3)*EL4(1)-S4(1)*EL4(3)))*COSTT1                     EMCAL
C                                                                           EMCAL
C     NOW SET UP THE LOWER HALF OF THE M MATRIX.                            EMCAL
C                                                                           EMCAL
C     DEFINE SOME REOCCURRING TERMS                                         EMCAL
      SR3 = B3MASS*(D13(2)  + EL3(2))                                       EMCAL
      SR4 = B3MASS*(D13(3)  + EL3(3))                                       EMCAL
      SR5 = B4MASS*(D14(2)  + EL4(2))                                       EMCAL
      SR6 = B4MASS*(D14(3)  + EL4(3))                                       EMCAL
      SR1=BODY1I(1,2)-B2MASS*EL2(2)*R2(1)-SR3*R3(1)-SR5*R4(1)               EMCAL
      SR2=BODY1I(1,3)-B2MASS*EL2(3)*R2(1)-SR4*R3(1)-SR6*R4(1)               EMCAL
C                                                                           EMCAL
      EM(4,1)=BODY1I(1,1)+B2MASS*(EL2(3)*R2(3)+EL2(2)*R2(2))+               EMCAL
     * SR4*R3(3)+SR3*R3(2)+SR6*R4(3)+SR5*R4(2)                              EMCAL
C                                                                           EMCAL
      EM(4,2)=SR1*COSTT1-SR2*SINTT1                                         EMCAL
C                                                                           EMCAL
      EM(4,3)=SR1*SINTT1+SR2*COSTT1                                         EMCAL
C                                                                           EMCAL
C     REDEFINE SR1 AND SR2                                                  EMCAL
      SR1=BMOM*(DO1(2)*COSTT1+DO1(3)*SINTT1)                                EMCAL
      SR2=BMOM*(-DO1(2)*SINTT1+DO1(3)*COSTT1)                               EMCAL
C                                                                           EMCAL
      EM(4,4) = BODY1I(1,1)                                                 EMCAL
     *         +B2MASS*(EL2(3)*(R2(3)-SR2)  + EL2(2)*(R2(2) - SR1)) +       EMCAL
     * SR4*(R3(3)-SR2)+SR3*(R3(2)-SR1)+SR6*(R4(3)-SR2)+SR5*(R4(2)-SR1)      EMCAL
C                                                                           EMCAL
      EM(4,5)=B3MASS*(((R3(3)-SR2)*EL3(3)+(R3(2)-SR1)*EL3(2))*S3(1)-        EMCAL
     * (R3(2)-SR1)*EL3(1)*S3(2)-(R3(3)-SR2)*EL3(1)*S3(3))                   EMCAL
C                                                                           EMCAL
      EM(4,6)=B4MASS*(((R4(3)-SR2)*EL4(3)+(R4(2)-SR1)*EL4(2))*S4(1)-        EMCAL
     * (R4(2)-SR1)*EL4(1)*S4(2)-(R4(3)-SR2)*EL4(1)*S4(3))                   EMCAL
C                                                                           EMCAL
      EM(5,1)=-B3MASS*EL3(1)*(R3(2)*S3(2)+R3(3)*S3(3))                      EMCAL
C                                                                           EMCAL
      EM(5,2)=B3MASS*(((EL3(3)*R3(3)+EL3(1)*R3(1))*COSTT1+                  EMCAL
     * EL3(3)*R3YSN)*S3(2)+(-EL3(2)*R3ZCS-(EL3(2)*R3(2)+EL3(1)*R3(1))*      EMCAL
     * SINTT1)*S3(3))                                                       EMCAL
C                                                                           EMCAL
      EM(5,3)=B3MASS*(((EL3(3)*R3(3)+EL3(1)*R3(1))*SINTT1-                  EMCAL
     * EL3(3)*R3YCS)*S3(2)+(-EL3(2)*R3ZSN+(EL3(2)*R3(2)+EL3(1)*R3(1))*      EMCAL
     * COSTT1)*S3(3))                                                       EMCAL
C                                                                           EMCAL
      EM(5,4)=B3MASS*EL3(1)*((-R3(2)+SR1)*S3(2)+(-R3(3)+SR2)*S3(3))         EMCAL
```

```
C                                                                         EMCAL
      EM(5,5)=B3MASS*(1.-B3MASS/TOTMAS)*(((EL3(3)**2+EL3(1)**2)*S3(2)-   EMCAL
     * EL3(3)*EL3(2)*S3(3))*S3(2)+((EL3(2)**2+EL3(1)**2)*S3(3)-          EMCAL
     * EL3(2)*EL3(3)*S3(2))*S3(3))                                       EMCAL
C                                                                         EMCAL
      EM(5,6)=(B3MASS*B4MASS/TOTMAS)*((-(EL3(3)*EL4(3)+EL3(1)*EL4(1))*   EMCAL
     * S4(2)+EL3(3)*EL4(2)*S4(3))*S3(2)+(-(EL3(2)*EL4(2)+EL3(1)*EL4(1))*EMCAL
     * S4(3)+EL3(2)*EL4(3)*S4(2))*S3(3))                                 EMCAL
C                                                                         EMCAL
      EM(6,1)=-B4MASS*EL4(1)*(R4(2)*S4(2)+R4(3)*S4(3))                   EMCAL
C                                                                         EMCAL
      EM(6,2)=B4MASS*(((EL4(3)*R4(3)+EL4(1)*R4(1))*COSTT1+EL4(3)*R4YSN)*EMCAL
     * S4(2)+(-EL4(2)*R4ZCS-(EL4(2)*R4(2)+EL4(1)*R4(1))*SINTT1)*S4(3))  EMCAL
C                                                                         EMCAL
      EM(6,3)=B4MASS*(((EL4(3)*R4(3)+EL4(1)*R4(1))*SINTT1-EL4(3)*R4YCS)*EMCAL
     * S4(2)+(-EL4(2)*R4ZSN+(EL4(2)*R4(2)+EL4(1)*R4(1))*COSTT1)*S4(3))  EMCAL
C                                                                         EMCAL
      EM(6,4)=B4MASS*EL4(1)*((-R4(2)+SR1)*S4(2)+(-R4(3)+SR2)*S4(3))      EMCAL
C                                                                         EMCAL
      EM(6,5)=(B3MASS*B4MASS/TOTMAS)*((-(EL4(3)*EL3(3)+EL4(1)*EL3(1))*   EMCAL
     * S3(2)+EL4(3)*EL3(2)*S3(3))*S4(2)+(-(EL4(2)*EL3(2)+EL4(1)*EL3(1))*EMCAL
     * S3(3)+EL4(2)*EL3(3)*S3(2))*S4(3))                                 EMCAL
C                                                                         EMCAL
      EM(6,6)=B4MASS*(1.-B4MASS/TOTMAS)*(((EL4(3)**2+EL4(1)**2)*S4(2)-   EMCAL
     * EL4(3)*EL4(2)*S4(3))*S4(2)+((EL4(2)**2+EL4(1)**2)*S4(3)-          EMCAL
     * EL4(2)*EL4(3)*S4(2))*S4(3))                                       EMCAL
      DO 10 M=1,3                                                         EMCAL
      DO 10 N=1,3                                                         EMCAL
      EM(M,N)=EM(M,N) + EEE(M,N)                                          EMCAL
   10 CONTINUE                                                            EMCAL
      RETURN                                                              EMCAL
C                                                                         EMCAL
C     ******************************************************************
C     *
C     *
C     *
      END
      SUBROUTINE FOMS(A,B,N,E,TJ)
      DIMENSION  A(1),B(1),TJ(1)
C     ******************************************************************
C                                                                       FOMS
C     A(1) CONTAINS THE CURRENT TIME.  I.E.  A(1) = TIME.               FOMS
C     A(2) THROUGH A(N) CONTAIN THE INTEGRALS WHERE N EQUALS THE        FOMS
C     NUMBER OF INTEGRALS PLUS 1.                                       FOMS
C     TJ IS A SCRATCH ARRAY.  TJ(1) CONTAINS THE INITIAL DELT AND TJ(2) FOMS
C     THROUGH TJ(N) CONTAIN THE BACK VALUES OF THE DERIVATIVES.         FOMS
C     B(1) CONTAINS THE CURRENT DELT AND B(2) THROUGH B(N) CONTAIN THE  FOMS
C     CURRENT DERIVATIVES.                                              FOMS
C     IF E = 0., REINITIALIZE THE DERIVATIVES.                          FOMS
C     IF E = 1., CONTINUE THE INTEGRATION.                              FOMS
C                                                                       FOMS
C     ******************************************************************
C     *
C     *
C     *
C     ******************************************************************
C     *                                                                 FOMS
      IF(N .LE. 1) RETURN                                               FOMS
      IF(E .NE. 0.) GO TO 20                                            FOMS
      E = 1.                                                            FOMS
      DO 10 I=1,N                                                       FOMS
      TJ(I) = B(I)                                                      FOMS
   10 CONTINUE                                                          FOMS
   20 HO2 = B(1)*0.5                                                    FOMS
      DO 30 I=2,N                                                       FOMS
      A(I) = A(I) + HO2*(3.*B(I)-TJ(I))                                 FOMS
      TJ(I) = B(I)                                                      FOMS
   30 CONTINUE                                                          FOMS
      A(1) = A(1) + B(1)                                                FOMS
      RETURN                                                            FOMS
C                                                                       FOMS
C     ******************************************************************
C     *
C     *
C     *
      END
      SUBROUTINE GGRAD
C     *
C     *
C     *
C     ****************************************************************************
      COMMON    A(3)          ,  AE(5)         ,  AED(5)         , COMMC
     *          AFOUR(2)      ,  AII(6,3,3)    ,                   COMMC
     *          AIO(6,3,3)    ,  AJ1(3)        ,  ALT            , COMMC
     *          AOCJ(6,3,3)   ,  AOJ(3)        ,  AONE(7)        , COMMC
     *          ATCPT2(3,3)   ,  ATHREE(5)     ,  ATWO(4)        , COMMC
     *          A1(3,3)       ,  A1J(2)                            COMMC
      COMMON    BDMASS        ,  BFOUR(2)      ,  BMOM           , COMMC
     *          BODYDI(3,3)   ,  BODYOI(3,3)   ,  BODY1I(3,3)    , COMMC
     *          BOMASS        ,  BONE(7)       ,  BTHREE(5)      , COMMC
     *          BTWO(4)       ,  B1MASS        ,  B2MASS         , COMMC
     *          B3MASS        ,  B4MASS                            COMMC
      COMMON    CA(3)         ,  CB(3)         ,  CGAINO(3)      , COMMC
     *          CGAIN1(2)     ,                                    COMMC
     *          COSFEJ        ,  COSTTJ        ,  COSTTO         , COMMC
     *          COSTT1        ,  COSTT3        ,  COSTT4         , COMMC
     *          CO2T          ,  CP1           ,  CP2            , COMMC
     *          CST           ,  C1                                COMMC
      COMMON    DB(3)         ,  DDO1(3)       ,                   COMMC
     *          DELTAT        ,  DO1(3)        ,  DO1DOT(3)      , COMMC
     *          DTI(3)        ,                                    COMMC
     *          DTIME         ,  D12(3)        ,  D13(3)         , COMMC
     *          D13DOT(3)     ,  D13YCS        ,  D13YSN         , COMMC
     *          D13ZCS        ,  D13ZSN        ,  D14(3)         , COMMC
     *          D14DOT(3)     ,  D14YCS        ,  D14YSN         , COMMC
     *          D14ZCS        ,  D14ZSN                            COMMC
      COMMON    EEE(3,3)      ,  EEJ(3,3)      ,  EL2(3)         , COMMC
     *          EL2DOT(3)     ,  EL2YCS        ,  EL2YSN         , COMMC
     *          EL2ZCS        ,  EL2ZSN        ,  EL3(3)         , COMMC
     *          EL3DOT(3)     ,  EL3YCS        ,  EL3YSN         , COMMC
     *          EL3ZCS        ,  EL3ZSN        ,  EL4(3)         , COMMC
     *          EL4DOT(3)     ,  EL4YCS        ,  EL4YSN         , COMMC
     *          EL4ZCS        ,  EL4ZSN        ,  EM(6,6)          COMMC
      COMMON    FAT(8)        ,                                    COMMC
     *          FEE(6)        ,  FEED(6)       ,  FFF(3)         , COMMC
     *          FFJ(3)        ,  FLAG1         ,  FLAG2          , COMMC
     *          FLAG3         ,  FLAG4         ,  FN             , COMMC
     *          FO(3)         ,  FO1(3)        ,  FO2(3)         , COMMC
     *          FO3(3)        ,  F1(3)         ,  F11(3)         , COMMC
     *          FPT(5)        ,                                    COMMC
     *          F12(3)        ,  F13(3)                            COMMC
      COMMON    GAIN(10)      ,  G3            ,                   COMMC
     *          G3DOT         ,  G4            ,  G4DOT            COMMC
      COMMON    H(3)          ,  HCMG(3)       ,  HDOT(3)        , COMMC
     *          HI(3)         ,  HO(3)         ,  HW(6)          , COMMC
     *          H1(3)         ,  H1PDOT(3)     ,  H1PRIM(3)      , COMMC
     *          H3PRIM(3)     ,  H4PRIM(3)                         COMMC
      COMMON    IB2F          ,  ICFA          ,  ICFB           , COMMC
     *          ICFC          ,  ICFD          ,  IDOCK          , COMMC
     *          IDOF(6)       ,                                    COMMC
     *          IGRAVF        ,  IPNDLM        ,  IPNTCK         , COMMC
     *          IPRINT        ,  IPROPF                            COMMC
      COMMON    NCASE         ,  NCHECK        ,  NDECK          , COMMC
     *          NGAIN         ,  NUMCMG                            COMMC
      COMMON    OMEGA1        ,  OMEGA3        ,  OMEGA4           COMMC
      COMMON    PEND3L        ,  PEND4L                            COMMC
      COMMON    Q(4,4)                                             COMMC
      COMMON    R           ,    RO(3)        ,    R1(3)        , COMMC
     *          R1DOT(3)    ,    R1YCS        ,    R1YSN        , COMMC
     *          R1ZCS       ,    R1ZSN        ,    R2(3)        , COMMC
     *          R2DOT(3)    ,    R2YCS        ,    R2YSN        , COMMC
     *          R2ZCS       ,    R2ZSN        ,    R3(3)        , COMMC
     *          R3DOT(3)    ,    R3YCS        ,    R3YSN        , COMMC
     *          R3ZCS       ,    R3ZSN        ,    R4(3)        , COMMC
     *          R4DOT(3)    ,    R4YCS        ,    R4YSN        , COMMC
     *          R4ZCS       ,    R4ZSN                            COMMC
      COMMON    S           ,    SDOT         ,    SINFEJ       , COMMC
     *          SINTTJ      ,    SINTTO       ,    SINTT1       , COMMC
     *          SINTT2      ,    SINTT3       ,    SINTT4       , COMMC
     *          SP          ,    SUM1         ,    SUM2         , COMMC
     *          SUM3        ,    S2(3)        ,    S3(3)        , COMMC
     *          S4(3)                                             COMMC
      COMMON    T(3,3)      ,    TC(3,3)      ,    TEMP1(3)     , COMMC
     *          TEMP2(3)    ,                                     COMMC
     *          TEMP3(3)    ,    TEMP4(3)     ,    TEMP5(3,3)   , COMMC
     *          TEMP6(3,3)  ,    TEMP7(3,3)   ,    TEMP8(3,3)   , COMMC
     *          TEMP9(3,3)  ,    TEMP10(3,3)  ,    TEMP11(3,3)  , COMMC
     *          TEMP12(3,3) ,    TEMP13(3,3)  ,    TEMP14(3,3)  , COMMC
     *          TEMP15(3,3) ,    TERM1(3)     ,    TERM2(3)     , COMMC
     *          TFRICT      ,    THATA(6)     ,    THATAD(6)    , COMMC
     *          THETA1      ,    THETA3       ,    THETA4       , COMMC
     *          THETO       ,    TIBO(3,3)    ,    TIBO1(3,3)   , COMMC
     *          TIME        ,    TJ           ,    TJ1(10)      , COMMC
     *          TJ2(10)     ,    TJ3(10)      ,    TJ4(10)      , COMMC
     *          TMOTOR      ,                                     COMMC
     *          TOEF(3)     ,    TOTMAS       ,    TO1          , COMMC
     *          TQOG(3)     ,    TQOP(3)      ,    TQ1G(3)      , COMMC
     *          TQ1P(3)     ,    TSTART       ,    TSTOP        , COMMC
     *          TT1DOT      ,    TT3DOT       ,    TT4DOT       , COMMC
     *          T1EF(3)     ,    T13          ,    T14            COMMC
      COMMON    V(3)                                              COMMC
      COMMON    W0(3)       ,    WS           ,    W1(3)        , COMMC
     *          W3(3)       ,    W4(3)                            COMMC
      COMMON    X(6,7)      ,    XC           ,    XCDOT        , COMMC
     *          XMU                                               COMMC
C                                                                 COMMC
C     ****************************************************************
C     *
C     *
C     *
C     ****************************************************************
C                                                                 GGRAI
      SINTTO = SIN(THETO)                                         GGRAI
      COSTTO = COS(THETO)                                         GGRAI
      DB(1) = -SINTTO*TIBO(1,1) + COSTTO*TIBO(2,1)                GGRAI
      DB(2) = -SINTTO*TIBO(1,2) + COSTTO*TIBO(2,2)                GGRAI
      DB(3) = -SINTTO*TIBO(1,3) + COSTTO*TIBO(2,3)                GGRAI
      REMP = 0.                                                   GGRAI
      DO 10 M=1,3                                                 GGRAI
      REMP = REMP + DB(M)*RO(M)                                   GGRAI
   10 CONTINUE                                                    GGRAI
      DO 20 L=1,3                                                 GGRAI
      FO1(L) = C1*BOMASS*(3.*REMP*DB(L) - RO(L))                  GGRAI
      A(L) = 0.                                                   GGRAI
      DO 20 M=1,3                                                 GGRAI
      A(L) = A(L) + BODYOI(L,M)*DB(M)                             GGRAI
   20 CONTINUE                                                    GGRAI
      TQOG(1) = 3.*C1*(DB(2)*A(3) - DB(3)*A(2))                   GGRAI
      TQOG(2) = 3.*C1*(DB(3)*A(1) - DB(1)*A(3))                   GGRAI
      TQOG(3) = 3.*C1*(DB(1)*A(2) - DB(2)*A(1))                          GGRAD
      DB(1) = -SINTTO*TIBO(1,1) + COSTTO*TIBO(2,1)                       GGRAD
      DB(2) = COSTT1*(-SINTTO*TIBO(1,2) + COSTTO*TIBO(2,2)) +            GGRAD
     *        SINTT1*(-SINTTO*TIBO(1,3) + COSTTO*TIBO(2,3))              GGRAD
      DB(3) =-SINTT1*(-SINTTO*TIBO(1,2) + COSTTO*TIBO(2,2)) +            GGRAD
     *        COSTT1*(-SINTTO*TIBO(1,3) + COSTTO*TIBO(2,3))              GGRAD
      REMP = 0.                                                          GGRAD
      DO 30 M=1,3                                                        GGRAD
      REMP = REMP + DB(M)*R1(M)                                          GGRAD
   30 CONTINUE                                                           GGRAD
      DO 40 L=1,3                                                        GGRAD
      F11(L) = C1*B1MASS*(3.*REMP*DB(L) - R1(L))                         GGRAD
      A(L) = 0.                                                          GGRAD
      DO 40 M=1,3                                                        GGRAD
      A(L) = A(L) + BODY1I(L,M)*DB(M)                                    GGRAD
   40 CONTINUE                                                           GGRAD
      TQ1G(1) = 3.*C1*(DB(2)*A(3) - DB(3)*A(2))                          GGRAD
      TQ1G(2) = 3.*C1*(DB(3)*A(1) - DB(1)*A(3))                          GGRAD
      TQ1G(3) = 3.*C1*(DB(1)*A(2) - DB(2)*A(1))                          GGRAD
      RETURN                                                             GGRAD
C                                                                        GGRAD
C     ***********************************************************************
C     *
C     *
C     *
      END
      SUBROUTINE HCON
C     *
C     *
C     *
C     ************************************************************************
      COMMON    A(3)           ,     AE(5)          ,     AED(5)         , COMM
     *          AFOUR(2)       ,     AII(6,3,3)     ,                      COMM
     *          AIO(6,3,3)     ,     AJ1(3)         ,     ALT            , COMM
     *          AOCJ(6,3,3)    ,     AOJ(3)         ,     AONE(7)        , COMM
     *          ATCPT2(3,3)    ,     ATHREE(5)      ,     ATWO(4)        , COMM
     *          A1(3,3)        ,     A1J(2)                                COMM
      COMMON    BDMASS         ,     BFOUR(2)       ,     BMOM           , COMM
     *          BODYDI(3,3)    ,     BODY0I(3,3)    ,     BODY1I(3,3)    , COMM
     *          BOMASS         ,     BONE(7)        ,     BTHREE(5)      , COMM
     *          BTWO(4)        ,     B1MASS         ,     B2MASS         , COMM
     *          B3MASS         ,     B4MASS                                COMM
      COMMON    CA(3)          ,     CB(3)          ,     CGAIN0(3)      , COMM
     *          CGAIN1(2)      ,                                           COMM
     *          COSFEJ         ,     COSTTJ         ,     COSTTO         , COMM
     *          COSTTI         ,     COSTT3         ,     COSTT4         , COMM
     *          CO2T           ,     CP1            ,     CP2            , COMM
     *          CST            ,     C1                                    COMM
      COMMON    DB(3)          ,     DDO1(3)        ,                      COMM
     *          DELTAT         ,     DO1(3)         ,     DO1DOT(3)      , COMM
     *          DTI(3)         ,                                           COMM
     *          DTIME          ,     D12(3)         ,     D13(3)         , COMM
     *          D13DOT(3)      ,     D13YCS         ,     D13YSN         , COMM
     *          D13ZCS         ,     D13ZSN         ,     D14(3)         , COMM
     *          D14DOT(3)      ,     D14YCS         ,     D14YSN         , COMM
     *          D14ZCS         ,     D14ZSN                                COMM
      COMMON    EEE(3,3)       ,     EEJ(3,3)       ,     EL2(3)         , COMM
     *          EL2DOT(3)      ,     EL2YCS         ,     EL2YSN         , COMM
     *          EL2ZCS         ,     EL2ZSN         ,     EL3(3)         , COMM
     *          EL3DOT(3)      ,     EL3YCS         ,     EL3YSN         , COMM
     *          EL3ZCS         ,     EL3ZSN         ,     EL4(3)         , COMM
     *          EL4DOT(3)      ,     EL4YCS         ,     EL4YSN         , COMM
     *          EL4ZCS         ,     EL4ZSN         ,     EMI(6,6)         COMM
      COMMON    FAT(8)         ,                                           COMM
     *          FEE(6)         ,     FEED(6)        ,     FFF(3)         , COMM
     *          FFJ(3)         ,     FLAG1          ,     FLAG2          , COMM
     *          FLAG3          ,     FLAG4          ,     FN             , COMM
     *          FO(3)          ,     FO1(3)         ,     FO2(3)         , COMM
     *          FO3(3)         ,     F1(3)          ,     F11(3)         , COMM
     *          FPT(5)         ,                                           COMM
     *          F12(3)         ,     F13(3)                                COMM
      COMMON    GAIN(10)       ,     G3             ,                      COMM
     *          G3DOT          ,     G4             ,     G4DOT            COMM
      COMMON    H(3)           ,     HCMG(3)        ,     HDOT(3)        , COMM
     *          HI(3)          ,     HO(3)          ,     HW(6)          , COMM
     *          H1(3)          ,     H1PDOT(3)      ,     H1PRIM(3)      , COMM
     *          H3PRIM(3)      ,     H4PRIM(3)                             COMM
      COMMON    IB2F           ,     ICFA           ,     ICFB           , COMM
     *          ICFC           ,     ICFD           ,     IDOCK          , COMM
     *          IDOF(6)        ,                                           COMM
     *          IGRAVF         ,     IPNDLM         ,     IPNTCK         , COMM
     *          IPRINT         ,     IPROPF                                COMM
      COMMON    NCASE          ,     NCHECK         ,     NDECK          , COMM
     *          NGAIN          ,     NUMCMG                                COMM
```

```
      COMMON   OMEGA1      ,    OMEGA3      ,    OMEGA4           COMMO
      COMMON   PEND3L      ,    PEND4L                            COMMO
      COMMON   Q(4,4)                                             COMMO
      COMMON   R           ,    R0(3)       ,    R1(3)       ,    COMMO
     *         R1DOT(3)    ,    R1YCS       ,    R1YSN       ,    COMMO
     *         R1ZCS       ,    R1ZSN       ,    R2(3)       ,    COMMO
     *         R2DOT(3)    ,    R2YCS       ,    R2YSN       ,    COMMO
     *         R2ZCS       ,    R2ZSN       ,    R3(3)       ,    COMMO
     *         R3DOT(3)    ,    R3YCS       ,    R3YSN       ,    COMMO
     *         R3ZCS       ,    R3ZSN       ,    R4(3)       ,    COMMO
     *         R4DOT(3)    ,    R4YCS       ,    R4YSN       ,    COMMO
     *         R4ZCS       ,    R4ZSN                             COMMO
      COMMON   S           ,    SDOT        ,    SINFEJ      ,    COMMO
     *         SINTTJ      ,    SINTT0      ,    SINTT1      ,    COMMO
     *         SINTT2      ,    SINTT3      ,    SINTT4      ,    COMMO
     *         SP          ,    SUM1        ,    SUM2        ,    COMMO
     *         SUM3        ,    S2(3)       ,    S3(3)       ,    COMMO
     *         S4(3)                                              COMMO
      COMMON   T(3,3)      ,    TC(3,3)     ,    TEMP1(3)    ,    COMMO
     *         TEMP2(3)    ,                                      COMMO
     *         TEMP3(3)    ,    TEMP4(3)    ,    TEMP5(3,3)  ,    COMMO
     *         TEMP6(3,3)  ,    TEMP7(3,3)  ,    TEMP8(3,3)  ,    COMMO
     *         TEMP9(3,3)  ,    TEMP10(3,3) ,    TEMP11(3,3) ,    COMMO
     *         TEMP12(3,3) ,    TEMP13(3,3) ,    TEMP14(3,3) ,    COMMO
     *         TEMP15(3,3) ,    TERM1(3)    ,    TERM2(3)    ,    COMMO
     *         TFRICT      ,    THATA(6)    ,    THATAD(6)   ,    COMMO
     *         THETA1      ,    THETA3      ,    THETA4      ,    COMMO
     *         THETO       ,    TIBO(3,3)   ,    TIBOI(3,3)  ,    COMMO
     *         TIME        ,    TJ          ,    TJ1(10)     ,    COMMO
     *         TJ2(10)     ,    TJ3(10)     ,    TJ4(10)     ,    COMMO
     *         TMOTOR      ,                                      COMMO
     *         TOEF(3)     ,    TOTMAS      ,    T01         ,    COMMO
     *         TQ0G(3)     ,    TQ0P(3)     ,    TQ1G(3)     ,    COMMO
     *         TQ1P(3)     ,    TSTART      ,    TSTOP       ,    COMMO
     *         TT1DOT      ,    TT3DOT      ,    TT4DOT      ,    COMMO
     *         T1EF(3)     ,    T13         ,    T14              COMMO
      COMMON   V(3)                                               COMMO
      COMMON   W0(3)       ,    WS          ,    W1(3)       ,    COMMO
     *         W3(3)       ,    W4(3)                             COMMO
      COMMON   X(6,7)      ,    XC          ,    XCDOT       ,    COMMO
     *         XMU                                                COMMO
C                                                                 COMMO
C     ****************************************************************
C     *
C     *
C     *
C     ****************************************************************
C                                                                 HCON
      THATAD(1) = +GAIN(4)*TIBO(3,2) +GAIN(5)*W0(1)               HCON
      THATAD(2) =  +THATAD(1)                                     HCON
      CALL CMG                                                    HCON
      RETURN                                                      HCON
C                                                                 HCON
C     ****************************************************************
C     *
C     *
C     *
      END
      SUBROUTINE MULT(C,A,B,F,D,E,MTYPE)
C     *
C     *
C     *
C     ***************************************************************************
C                                                                             MULT
      DIMENSION  A(3,3)        ,      B(3)          ,      C(3)          ,    MULT
     *           D(3,3)        ,      E(3,3)        ,      F(3,3)             MULT
C                                                                             MULT
      IF (MTYPE .NE. 1) GO TO 100                                             MULT
      C(1) = A(1,1)*B(1) + A(1,2)*B(2) + A(1,3)*B(3)                          MULT
      C(2) = A(2,1)*B(1) + A(2,2)*B(2) + A(2,3)*B(3)                          MULT
      C(3) = A(3,1)*B(1) + A(3,2)*B(2) + A(3,3)*B(3)                          MULT
      RETURN                                                                  MULT
  100 CONTINUE                                                                MULT
      F(1,1) = D(1,1)*E(1,1) + D(1,2)*E(2,1) + D(1,3)*E(3,1)                  MULT
      F(1,2) = D(1,1)*E(1,2) + D(1,2)*E(2,2) + D(1,3)*E(3,2)                  MULT
      F(1,3) = D(1,1)*E(1,3) + D(1,2)*E(2,3) + D(1,3)*E(3,3)                  MULT
      F(2,1) = D(2,1)*E(1,1) + D(2,2)*E(2,1) + D(2,3)*E(3,1)                  MULT
      F(2,2) = D(2,1)*E(1,2) + D(2,2)*E(2,2) + D(2,3)*E(3,2)                  MULT
      F(2,3) = D(2,1)*E(1,3) + D(2,2)*E(2,3) + D(2,3)*E(3,3)                  MULT
      F(3,1) = D(3,1)*E(1,1) + D(3,2)*E(2,1) + D(3,3)*E(3,1)                  MULT
      F(3,2) = D(3,1)*E(1,2) + D(3,2)*E(2,2) + D(3,3)*E(3,2)                  MULT
      F(3,3) = D(3,1)*E(1,3) + D(3,2)*E(2,3) + D(3,3)*E(3,3)                  MULT
      RETURN                                                                  MULT
C                                                                             MULT
C     ***************************************************************************
C     *
C     *
C     *
      END
      SUBROUTINE PCON
C     *
C     *
C     *
C     ******************************************************************
      COMMON     A(3)          ,      AE(5)        ,      AED(5)       , COMMC
     *           AFOUR(2)      ,      AII(6,3,3)   ,                     COMMC
     *           AIO(6,3,3)    ,      AJ1(3)       ,      ALT          , COMMC
     *           AOCJ(6,3,3)   ,      AOJ(3)       ,      AONE(7)      , COMMC
     *           ATCPT2(3,3)   ,      ATHREE(5)    ,      ATWO(4)      , COMMC
     *           A1(3,3)       ,      A1J(2)                             COMMC
      COMMON     BDMASS        ,      BFOUR(2)     ,      BMOM         , COMMC
     *           BODYDI(3,3)   ,      BODYOI(3,3)  ,      BODY1I(3,3)  , COMMC
     *           BOMASS        ,      BONE(7)      ,      BTHREE(5)    , COMMC
     *           BTWO(4)       ,      B1MASS       ,      B2MASS       , COMMC
     *           B3MASS        ,      B4MASS                             COMMC
      COMMON     CA(3)         ,      CB(3)        ,      CGAINO(3)    , COMMC
     *           CGAIN1(2)     ,                                         COMMC
     *           COSFEJ        ,      COSTTJ       ,      COSTTO       , COMMC
     *           COSTT1        ,      COSTT3       ,      COSTT4       , COMMC
     *           CO2T          ,      CP1          ,      CP2          , COMMC
     *           CST           ,      C1                                 COMMC
      COMMON     DB(3)         ,      DD01(3)      ,                     COMMC
     *           DELTAT        ,      D01(3)       ,      D01DOT(3)    , COMMC
     *           DTI(3)        ,                                         COMMC
     *           DTIME         ,      D12(3)       ,      D13(3)       , COMMC
     *           D13DOT(3)     ,      D13YCS       ,      D13YSN       , COMMC
     *           D13ZCS        ,      D13ZSN       ,      D14(3)       , COMMC
     *           D14DOT(3)     ,      D14YCS       ,      D14YSN       , COMMC
     *           D14ZCS        ,      D14ZSN                             COMMC
      COMMON     EEE(3,3)      ,      EEJ(3,3)     ,      EL2(3)       , COMMC
     *           EL2DOT(3)     ,      EL2YCS       ,      EL2YSN       , COMMC
     *           EL2ZCS        ,      EL2ZSN       ,      EL3(3)       , COMMC
     *           EL3DOT(3)     ,      EL3YCS       ,      EL3YSN       , COMMC
     *           EL3ZCS        ,      EL3ZSN       ,      EL4(3)       , COMMC
     *           EL4DOT(3)     ,      EL4YCS       ,      EL4YSN       , COMMC
     *           EL4ZCS        ,      EL4ZSN       ,      EM(6,6)        COMMC
      COMMON     FAT(8)        ,                                         COMMC
     *           FEE(6)        ,      FEED(6)      ,      FFF(3)       , COMMC
     *           FFJ(3)        ,      FLAG1        ,      FLAG2        , COMMC
     *           FLAG3         ,      FLAG4        ,      FN           , COMMC
     *           F0(3)         ,      F01(3)       ,      F02(3)       , COMMC
     *           F03(3)        ,      F1(3)        ,      F11(3)       , COMMC
     *           FPT(5)        ,                                         COMMC
     *           F12(3)        ,      F13(3)                             COMMC
      COMMON     GAIN(10)      ,      G3           ,                     COMMC
     *           G3DOT         ,      G4           ,      G4DOT          COMMC
      COMMON     H(3)          ,      HCMG(3)      ,      HDOT(3)      , COMMC
     *           HI(3)         ,      HO(3)        ,      HW(6)        , COMMC
     *           H1(3)         ,      H1PDOT(3)    ,      H1PRIM(3)    , COMMC
     *           H3PRIM(3)     ,      H4PRIM(3)                          COMMC
      COMMON     IB2F          ,      ICFA         ,      ICFB         , COMMC
     *           ICFC          ,      ICFD         ,      IDOCK        , COMMC
     *           IDOF(6)       ,                                         COMMC
     *           IGRAVF        ,      IPNDLM       ,      IPNTCK       , COMMC
     *           IPRINT        ,      IPROPF                             COMMC
      COMMON     NCASE         ,      NCHECK       ,      NDECK        , COMMC
     *           NGAIN         ,      NUMCMG                             COMMC
      COMMON     OMEGA1        ,      OMEGA3       ,      OMEGA4         COMMC
      COMMON     PEND3L        ,      PEND4L                             COMMC
      COMMON     Q(4,4)                                                  COMMC
      COMMON     R             ,     RO(3)         ,     R1(3)          , COMMC
     *           R1DOT(3)      ,     R1YCS         ,     R1YSN          , COMMC
     *           R1ZCS         ,     R1ZSN         ,     R2(3)          , COMMC
     *           R2DOT(3)      ,     R2YCS         ,     R2YSN          , COMMC
     *           R2ZCS         ,     R2ZSN         ,     R3(3)          , COMMC
     *           R3DOT(3)      ,     R3YCS         ,     R3YSN          , COMMC
     *           R3ZCS         ,     R3ZSN         ,     R4(3)          , COMMC
     *           R4DOT(3)      ,     R4YCS         ,     R4YSN          , COMMC
     *           R4ZCS         ,     R4ZSN                                COMMC
      COMMON     S             ,     SDOT          ,     SINFEJ         , COMMC
     *           SINTTJ        ,     SINTTO        ,     SINTT1         , COMMC
     *           SINTT2        ,     SINTT3        ,     SINTT4         , COMMC
     *           SP            ,     SUM1          ,     SUM2           , COMMC
     *           SUM3          ,     S2(3)         ,     S3(3)          , COMMC
     *           S6(3)                                                    COMMC
      COMMON     T(3,3)        ,     TC(3,3)       ,     TEMP1(3)       , COMMC
     *           TEMP2(3)      ,                                          COMMC
     *           TEMP3(3)      ,     TEMP4(3)      ,     TEMP5(3,3)     , COMMC
     *           TEMP6(3,3)    ,     TEMP7(3,3)    ,     TEMP8(3,3)     , COMMC
     *           TEMP9(3,3)    ,     TEMP10(3,3)   ,     TEMP11(3,3)    , COMMC
     *           TEMP12(3,3)   ,     TEMP13(3,3)   ,     TEMP14(3,3)    , COMMC
     *           TEMP15(3,3)   ,     TERM1(3)      ,     TERM2(3)       , COMMC
     *           TFRICT        ,     THATA(6)      ,     THATAD(6)      , COMMC
     *           THETA1        ,     THETA3        ,     THETA4         , COMMC
     *           THETO         ,     TIBO(3,3)     ,     TIBOI(3,3)     , COMMC
     *           TIME          ,     TJ            ,     TJ1(10)        , COMMC
     *           TJ2(10)       ,     TJ3(10)       ,     TJ4(10)        , COMMC
     *           TMOTOR        ,                                          COMMC
     *           TOEF(3)       ,     TOTMAS        ,     TO1            , COMMC
     *           TQOG(3)       ,     TQOP(3)       ,     TQ1G(3)        , COMMC
     *           TQ1P(3)       ,     TSTART        ,     TSTOP          , COMMC
     *           TT1DOT        ,     TT3DOT        ,     TT4DOT         , COMMC
     *           T1EF(3)       ,     T13           ,     T14              COMMC
      COMMON     V(3)                                                     COMMC
      COMMON     WO(3)         ,     WS            ,     W1(3)          , COMMC
     *           W3(3)         ,     W4(3)                                COMMC
      COMMON     X(6,7)        ,     XC            ,     XCDOT          , COMMC
     *           XMU                                                      COMMC
C                                                                         COMMC
C     ***************************************************************************
C     *
C     *
C     *
C     ***************************************************************************
C                                                                         PCON
      SINTT1 = SIN(THETA1)                                                PCON
      COSTT1 = COS(THETA1)                                                PCON
      DO 30 M=1,3                                                         PCON
      FS = CGAINO(M)*WO(M)                                                PCON
      FPT(M) = 2.*ABS(FS)                                                 PCON
      TQOP(M) = FS*AOJ(M)                                                 PCON
   30 CONTINUE                                                            PCON
      FS = 0.                                                             PCON
      IF (COSTT1 .GT.  0.87)FS =  CGAIN1(2)*WO(2)                         PCON
      IF (COSTT1 .LT. -0.87)FS = -CGAIN1(2)*WO(2)                         PCON
      IF (SINTT1 .GT.  0.87)FS =  CGAIN1(2)*WO(3)                         PCON
      IF (SINTT1 .LT. -0.87)FS = -CGAIN1(2)*WO(3)                         PCON
      FPT(4) = 2.*ABS(FS)                                                 PCON
      TQ1P(2) = FS*A1J(2)                                                 PCON
      FS = CGAIN1(1)*(OMEGA1 - SP)                                        PCON
      FPT(5) = 2.*ABS(FS)                                                 PCON
      TQ1P(1) = FS*A1J(1)                                                 PCON
      RETURN                                                            PCON
C                                                                       PCON
C     ***********************************************************************
C     *
C     *
C     *
      END
      SUBROUTINE RECALC
C     *
C     *
C     *
C     ****************************************************************
      COMMON     A(3)          ,     AE(5)          ,     AED(5)         , COMMC
     *           AFOUR(2)      ,     AII(6,3,3)     ,                      COMMC
     *           AIO(6,3,3)    ,     AJ1(3)         ,     ALT            , COMMC
     *           AOCJ(6,3,3)   ,     AOJ(3)         ,     AONE(7)        , COMMC
     *           ATCPT2(3,3)   ,     ATHREE(5)      ,     ATWO(4)        , COMMC
     *           A1(3,3)       ,     A1J(2)                                COMMC
      COMMON     BDMASS        ,     BFOUR(2)       ,     BMOM           , COMMC
     *           BODYDI(3,3)   ,     BODYOI(3,3)    ,     BODY1I(3,3)    , COMMC
     *           BOMASS        ,     BONE(7)        ,     BTHREE(5)      , COMMC
     *           BTWO(4)       ,     B1MASS         ,     B2MASS         , COMMC
     *           B3MASS        ,     B4MASS                                COMMC
      COMMON     CA(3)         ,     CB(3)          ,     CGAINO(3)      , COMMC
     *           CGAIN1(2)     ,                                           COMMC
     *           COSFEJ        ,     COSTTJ         ,     COSTTO         , COMMC
     *           COSTT1        ,     COSTT3         ,     COSTT4         , COMMC
     *           CO2T          ,     CP1            ,     CP2            , COMMC
     *           CST           ,     C1                                    COMMC
      COMMON     DB(3)         ,     DDO1(3)        ,                      COMMC
     *           DELTAT        ,     DO1(3)         ,     DO1DOT(3)      , COMMC
     *           DTI(3)        ,                                           COMMC
     *           DTIME         ,     D12(3)         ,     D13(3)         , COMMC
     *           D13DOT(3)     ,     D13YCS         ,     D13YSN         , COMMC
     *           D13ZCS        ,     D13ZSN         ,     D14(3)         , COMMC
     *           D14DOT(3)     ,     D14YCS         ,     D14YSN         , COMMC
     *           D14ZCS        ,     D14ZSN                                COMMC
      COMMON     EEE(3,3)      ,     EEJ(3,3)       ,     EL2(3)         , COMMC
     *           EL2DOT(3)     ,     EL2YCS         ,     EL2YSN         , COMMC
     *           EL2ZCS        ,     EL2ZSN         ,     EL3(3)         , COMMC
     *           EL3DOT(3)     ,     EL3YCS         ,     EL3YSN         , COMMC
     *           EL3ZCS        ,     EL3ZSN         ,     EL4(3)         , COMMC
     *           EL4DOT(3)     ,     EL4YCS         ,     EL4YSN         , COMMC
     *           EL4ZCS        ,     EL4ZSN         ,     EM(6,6)          COMMC
      COMMON     FAT(8)        ,                                           COMMC
     *           FEE(6)        ,     FEED(6)        ,     FFF(3)         , COMMC
     *           FFJ(3)        ,     FLAG1          ,     FLAG2          , COMMC
     *           FLAG3         ,     FLAG4          ,     FN             , COMMC
     *           FO(3)         ,     FO1(3)         ,     FO2(3)         , COMMC
     *           FO3(3)        ,     F1(3)          ,     F11(3)         , COMMC
     *           FPT(5)        ,                                           COMMC
     *           F12(3)        ,     F13(3)                                COMMC
      COMMON     GAIN(10)      ,     G3             ,                      COMMC
     *           G3DOT         ,     G4             ,     G4DOT            COMMC
      COMMON     H(3)          ,     HCMG(3)        ,     HDOT(3)        , COMM(
     *           HI(3)         ,     HO(3)          ,     HW(6)          , COMM(
     *           H1(3)         ,     H1PDOT(3)      ,     H1PRIM(3)      , COMM(
     *           H3PRIM(3)     ,     H4PRIM(3)                             COMM(
      COMMON     IB2F          ,     ICFA           ,     ICFB           , COMM(
     *           ICFC          ,     ICFD           ,     IDOCK          , COMM(
     *           IDOF(6)       ,                                           COMM(
     *           IGRAVF        ,     IPNDLM         ,     IPNTCK         , COMM(
     *           IPRINT        ,     IPROPF                                COMM(
      COMMON     NCASE         ,     NCHECK         ,     NDECK          , COMM(
     *           NGAIN         ,     NUMCMG                                COMM(
      COMMON     OMEGA1        ,     OMEGA3         ,     OMEGA4           COMM(
      COMMON     PEND3L        ,     PEND4L                                COMM(
      COMMON     Q(4,6)                                                    COMM(
      COMMON   R             ,   R0(3)         ,   R1(3)         , COMMC
     *         R1DOT(3)      ,   R1YCS         ,   R1YSN         , COMMC
     *         R1ZCS         ,   R1ZSN         ,   R2(3)         , COMMC
     *         R2DOT(3)      ,   R2YCS         ,   R2YSN         , COMMC
     *         R2ZCS         ,   R2ZSN         ,   R3(3)         , COMMC
     *         R3DOT(3)      ,   R3YCS         ,   R3YSN         , COMMC
     *         R3ZCS         ,   R3ZSN         ,   R4(3)         , COMMC
     *         R4DOT(3)      ,   R4YCS         ,   R4YSN         , COMMC
     *         R4ZCS         ,   R4ZSN                           COMMC
      COMMON   S             ,   SDOT          ,   SINFEJ        , COMMC
     *         SINTTJ        ,   SINTTO        ,   SINTT1        , COMMC
     *         SINTT2        ,   SINTT3        ,   SINTT4        , COMMC
     *         SP            ,   SUM1          ,   SUM2          , COMMC
     *         SUM3          ,   S2(3)         ,   S3(3)         , COMMC
     *         S4(3)                                               COMMC
      COMMON   T(3,3)        ,   TC(3,3)       ,   TEMP1(3)      , COMMC
     *         TEMP2(3)      ,                                     COMMC
     *         TEMP3(3)      ,   TEMP4(3)      ,   TEMP5(3,3)    , COMMC
     *         TEMP6(3,3)    ,   TEMP7(3,3)    ,   TEMP8(3,3)    , COMMC
     *         TEMP9(3,3)    ,   TEMP10(3,3)   ,   TEMP11(3,3)   , COMMC
     *         TEMP12(3,3)   ,   TEMP13(3,3)   ,   TEMP14(3,3)   , COMMC
     *         TEMP15(3,3)   ,   TERM1(3)      ,   TERM2(3)      , COMMC
     *         TFRICT        ,   THATA(6)      ,   THATAD(6)     , COMMC
     *         THETA1        ,   THETA3        ,   THETA4        , COMMC
     *         THETO         ,   TIBO(3,3)     ,   TIBOI(3,3)    , COMMC
     *         TIME          ,   TJ            ,   TJ1(10)       , COMMC
     *         TJ2(10)       ,   TJ3(10)       ,   TJ4(10)       , COMMC
     *         TMOTOR        ,                                     COMMC
     *         TOEF(3)       ,   TOTMAS        ,   TO1           , COMMC
     *         TQOG(3)       ,   TQOP(3)       ,   TQ1G(3)       , COMMC
     *         TQ1P(3)       ,   TSTART        ,   TSTOP         , COMMC
     *         TT1DOT        ,   TT3DOT        ,   TT4DOT        , COMMC
     *         T1EF(3)       ,   T13           ,   T14             COMMC
      COMMON   V(3)                                                COMMC
      COMMON   W0(3)         ,   WS            ,   W1(3)         , COMMC
     *         W3(3)         ,   W4(3)                             COMMC
      COMMON   X(6,7)        ,   XC            ,   XCDOT         , COMMC
     *         XMU                                                 COMMC
C                                                                  COMMC
C     ******************************************************************
C     *
C     *
C     *
C     ******************************************************************
C                                                                  RECAL
      SINTT1 = SIN(THETA1)                                         RECAL
      COSTT1 = COS(THETA1)                                         RECAL
      CALL SCALC                                                   RECAL
      EL2(1) = D12(1) + S*S2(1)                                    RECAL
      EL2(2) = D12(2) + S*S2(2)                                    RECAL
      EL2(3) = D12(3) + S*S2(3)                                    RECAL
      SINTT3 = SIN(THETA3)                                         RECAL
      COSTT3 = COS(THETA3)                                         RECAL
      EL3(1) =  PEND3L*SINTT3                                      RECAL
      EL3(2) = -PEND3L*COSTT3*S3(3)                                RECAL
      EL3(3) =  PEND3L*COSTT3*S3(2)                                RECAL
      SINTT4 = SIN(THETA4)                                         RECAL
      COSTT4 = COS(THETA4)                                         RECAL
      EL4(1) =  PEND4L*SINTT4                                      RECAL
      EL4(2) = -PEND4L*COSTT4*S4(3)                                RECAL
      EL4(3) =  PEND4L*COSTT4*S4(2)                                RECAL
      R1(1) = (BOMASS/TOTMAS)*DO1(1)                               RECAL
     *          -(B2MASS/TOTMAS)*EL2(1) - (B3MASS/TOTMAS)*(D13(1) + EL3(1))RECAL
     *          -(B4MASS/TOTMAS)*(D14(1) + EL4(1))                         RECAL
      R1(2) = (B0MASS/TOTMAS)*(D01(2)*COSTT1 + D01(3)*SINTT1)              RECAL
     *          -(B2MASS/TOTMAS)*EL2(2)                                    RECAL
     *          -(B3MASS/TOTMAS)*(D13(2) + EL3(2))                         RECAL
     *          -(B4MASS/TOTMAS)*(D14(2) + EL4(2))                         RECAL
      R1(3) = (B0MASS/TOTMAS)*(-D01(2)*SINTT1 + D01(3)*COSTT1)             RECAL
     *          -(B2MASS/TOTMAS)*EL2(3)                                    RECAL
     *          -(B3MASS/TOTMAS)*(D13(3) + EL3(3))                         RECAL
     *          -(B4MASS/TOTMAS)*(D14(3) + EL4(3))                         RECAL
      R2(1) = R1(1) + EL2(1)                                               RECAL
      R2(2) = R1(2) + EL2(2)                                               RECAL
      R2(3) = R1(3) + EL2(3)                                               RECAL
      R3(1) = R1(1) + D13(1) + EL3(1)                                      RECAL
      R3(2) = R1(2) + D13(2) + EL3(2)                                      RECAL
      R3(3) = R1(3) + D13(3) + EL3(3)                                      RECAL
      R4(1) = R1(1) + D14(1) + EL4(1)                                      RECAL
      R4(2) = R1(2) + D14(2) + EL4(2)                                      RECAL
      R4(3) = R1(3) + D14(3) + EL4(3)                                      RECAL
      RETURN                                                               RECAL
C                                                                          RECAL
C     ******************************************************************
C     *
C     *
C     *
C     *
      END
      SUBROUTINE SCALC
C     *
C     *
C     *
C     ******************************************************************************
      COMMON     A(3)           ,   AE(5)          ,   AED(5)         , COMMC
     *           AFOUR(2)       ,   AII(6,3,3)     ,                    COMMC
     *           AIO(6,3,3)     ,   AJ1(3)         ,   ALT            , COMMC
     *           AOCJ(6,3,3)    ,   AOJ(3)         ,   AONE(7)        , COMMC
     *           ATCPT2(3,3)    ,   ATHREE(5)      ,   ATWO(4)        , COMMC
     *           A1(3,3)        ,   A1J(2)                              COMMC
      COMMON     BDMASS         ,   BFOUR(2)       ,   BMOM           , COMMC
     *           BODYDI(3,3)    ,   BODYOI(3,3)    ,   BODY1I(3,3)    , COMMC
     *           BOMASS         ,   BONE(7)        ,   BTHREE(5)      , COMMC
     *           BTWO(4)        ,   B1MASS         ,   B2MASS         , COMMC
     *           B3MASS         ,   B4MASS                              COMMC
      COMMON     CA(3)          ,   CB(3)          ,   CGAINO(3)      , COMMC
     *           CGAIN1(2)      ,                                       COMMC
     *           COSFEJ         ,   COSTTJ         ,   COSTTO         , COMMC
     *           COSTT1         ,   COSTT3         ,   COSTT4         , COMMC
     *           CO2T           ,   CP1            ,   CP2            , COMMC
     *           CST            ,   C1                                  COMMC
      COMMON     DB(3)          ,   DDO1(3)        ,                    COMMC
     *           DELTAT         ,   DO1(3)         ,   DO1DOT(3)      , COMMC
     *           DTI(3)         ,                                       COMMC
     *           DTIME          ,   D12(3)         ,   D13(3)         , COMMC
     *           D13DOT(3)      ,   D13YCS         ,   D13YSN         , COMMC
     *           D13ZCS         ,   D13ZSN         ,   D14(3)         , COMMC
     *           D14DOT(3)      ,   D14YCS         ,   D14YSN         , COMMC
     *           D14ZCS         ,   D14ZSN                              COMMC
      COMMON     EEE(3,3)       ,   EEJ(3,3)       ,   EL2(3)         , COMMC
     *           EL2DOT(3)      ,   EL2YCS         ,   EL2YSN         , COMMC
     *           EL2ZCS         ,   EL2ZSN         ,   EL3(3)         , COMMC
     *           EL3DOT(3)      ,   EL3YCS         ,   EL3YSN         , COMMC
     *           EL3ZCS         ,   EL3ZSN         ,   EL4(3)         , COMMC
     *           EL4DOT(3)      ,   EL4YCS         ,   EL4YSN         , COMMC
     *           EL4ZCS         ,   EL4ZSN         ,   EM(6,6)          COMMC
      COMMON     FAT(8)         ,                                       COMMC
     *           FEE(6)         ,   FEED(6)        ,   FFF(3)         , COMMC
     *           FFJ(3)         ,   FLAG1          ,   FLAG2          , COMMC
     *           FLAG3          ,   FLAG4          ,   FN             , COMMC
     *           FO(3)          ,   FO1(3)         ,   FO2(3)         , COMMC
     *           FO3(3)         ,   F1(3)          ,   F11(3)         , COMMC
     *           FPT(5)         ,                                       COMMC
     *           F12(3)         ,   F13(3)                              COMMC
      COMMON     GAIN(10)       ,   G3             ,                    COMMC
     *           G3DOT          ,   G4             ,   G4DOT            COMMC
      COMMON     H(3)           ,   HCMG(3)        ,   HDOT(3)        , COMMC
     *           HI(3)          ,   HO(3)          ,   HW(6)          , COMMC
     *           H1(3)          ,   H1PDOT(3)      ,   H1PRIM(3)      , COMMC
     *           H3PRIM(3)      ,   H4PRIM(3)                           COMMC
      COMMON     IB2F           ,   ICFA           ,   ICFB           , COMMC
     *           ICFC           ,   ICFD           ,   IDOCK          , COMMC
     *           IDOF(6)        ,                                       COMMC
     *           IGRAVF         ,   IPNDLM         ,   IPNTCK         , COMMC
     *           IPRINT         ,   IPROPF                              COMMC
      COMMON     NCASE          ,   NCHECK         ,   NDECK          , COMMC
     *           NGAIN          ,   NUMCMG                              COMMC
      COMMON     OMEGA1         ,   OMEGA3         ,   OMEGA4           COMMC
      COMMON     PEND3L         ,   PEND4L                              COMMC
      COMMON     Q(4,4)                                                 COMMC
      COMMON     R            ,     R0(3)        ,     R1(3)        , COMMC
     *           R1DOT(3)     ,     R1YCS        ,     R1YSN        , COMMC
     *           R1ZCS        ,     R1ZSN        ,     R2(3)        , COMMC
     *           R2DOT(3)     ,     R2YCS        ,     R2YSN        , COMMC
     *           R2ZCS        ,     R2ZSN        ,     R3(3)        , COMMC
     *           R3DOT(3)     ,     R3YCS        ,     R3YSN        , COMMC
     *           R3ZCS        ,     R3ZSN        ,     R4(3)        , COMMC
     *           R4DOT(3)     ,     R4YCS        ,     R4YSN        , COMMC
     *           R4ZCS        ,     R4ZSN                             COMMC
      COMMON     S            ,     SDOT         ,     SINFEJ       , COMMC
     *           SINTTJ       ,     SINTT0       ,     SINTT1       , COMMC
     *           SINTT2       ,     SINTT3       ,     SINTT4       , COMMC
     *           SP           ,     SUM1         ,     SUM2         , COMMC
     *           SUM3         ,     S2(3)        ,     S3(3)        , COMMC
     *           S4(3)                                                COMMC
      COMMON     T(3,3)       ,     TC(3,3)      ,     TEMP1(3)     , COMMC
     *           TEMP2(3)     ,                                       COMMC
     *           TEMP3(3)     ,     TEMP4(3)     ,     TEMP5(3,3)   , COMMC
     *           TEMP6(3,3)   ,     TEMP7(3,3)   ,     TEMP8(3,3)   , COMMC
     *           TEMP9(3,3)   ,     TEMP10(3,3)  ,     TEMP11(3,3)  , COMMC
     *           TEMP12(3,3)  ,     TEMP13(3,3)  ,     TEMP14(3,3)  , COMMC
     *           TEMP15(3,3)  ,     TERM1(3)     ,     TERM2(3)     , COMMC
     *           TFRICT       ,     THATA(6)     ,     THATAD(6)    , COMMC
     *           THETA1       ,     THETA3       ,     THETA4       , COMMC
     *           THETO        ,     TIBO(3,3)    ,     TIBOI(3,3)   , COMMC
     *           TIME         ,     TJ           ,     TJ1(10)      , COMMC
     *           TJ2(10)      ,     TJ3(10)      ,     TJ4(10)      , COMMC
     *           TMOTOR       ,                                       COMMC
     *           TOEF(3)      ,     TOTMAS       ,     TO1          , COMMC
     *           TQOG(3)      ,     TQOP(3)      ,     TQ1G(3)      , COMMC
     *           TQ1P(3)      ,     TSTART       ,     TSTOP        , COMMC
     *           TT1DOT       ,     TT3DOT       ,     TT4DOT       , COMMC
     *           T1EF(3)      ,     T13          ,     T14            COMMC
      COMMON     V(3)                                                 COMMC
      COMMON     W0(3)        ,     WS           ,     W1(3)        , COMMC
     *           W3(3)        ,     W4(3)                             COMMC
      COMMON     X(6,7)       ,     XC           ,     XCDOT        , COMMC
     *           XMU                                                  COMMC
C                                                                     COMMC
C     ****************************************************************
C     *
C     *
C     ****************************************************************
C                                                                     SCAL
C     THIS SUBROUTINE CALCULATES THE ELEVATOR POSITION.               SCAL
C                                                                     SCAL
      S = 0.                                                          SCAL
      RETURN                                                          SCAL
C                                                                     SCAL
C     ****************************************************************
C     *
C     *
      END
      SUBROUTINE SDCALC
C     *
C     *
C     *
C     ***********************************************************************
      COMMON    A(3)          ,    AE(5)          ,    AED(5)         , COMMC
     *          AFOUR(2)      ,    AII(6,3,3)     ,                     COMMC
     *          AIO(6,3,3)    ,    AJ1(3)         ,    ALT            , COMMC
     *          AOCJ(6,3,3)   ,    AOJ(3)         ,    AONE(7)        , COMMC
     *          ATCPT2(3,3)   ,    ATHREE(5)      ,    ATWO(4)        , COMMC
     *          A1(3,3)       ,    A1J(2)                               COMMC
      COMMON    BDMASS        ,    BFOUR(2)       ,    BMOM           , COMMC
     *          BODYDI(3,3)   ,    BODYOI(3,3)    ,    BODY1I(3,3)    , COMMC
     *          BOMASS        ,    BONE(7)        ,    BTHREE(5)      , COMMC
     *          BTWO(4)       ,    B1MASS         ,    B2MASS         , COMMC
     *          B3MASS        ,    B4MASS                               COMMC
      COMMON    CA(3)         ,    CB(3)          ,    CGAINO(3)      , COMMC
     *          CGAIN1(2)     ,                                         COMMC
     *          COSFEJ        ,    COSTTJ         ,    COSTTO         , COMMC
     *          COSTT1        ,    COSTT3         ,    COSTT4         , COMMC
     *          CO2T          ,    CP1            ,    CP2            , COMMC
     *          CST           ,    C1                                   COMM
      COMMON    DB(3)         ,    DD01(3)        ,                     COMM
     *          DELTAT        ,    D01(3)         ,    D01DOT(3)      , COMM
     *          DTI(3)        ,                                         COMM
     *          DTIME         ,    D12(3)         ,    D13(3)         , COMM
     *          D13DOT(3)     ,    D13YCS         ,    D13YSN         , COMM
     *          D13ZCS        ,    D13ZSN         ,    D14(3)         , COMM
     *          D14DOT(3)     ,    D14YCS         ,    D14YSN         , COMM
     *          D14ZCS        ,    D14ZSN                               COMM
      COMMON    EEE(3,3)      ,    EEJ(3,3)       ,    EL2(3)         , COMM
     *          EL2DOT(3)     ,    EL2YCS         ,    EL2YSN         , COMM
     *          EL2ZCS        ,    EL2ZSN         ,    EL3(3)         , COMM
     *          EL3DOT(3)     ,    EL3YCS         ,    EL3YSN         , COMM
     *          EL3ZCS        ,    EL3ZSN         ,    EL4(3)         , COMM
     *          EL4DOT(3)     ,    EL4YCS         ,    EL4YSN         , COMM
     *          EL4ZCS        ,    EL4ZSN         ,    EM(6,6)          COMM
      COMMON    FAT(8)        ,                                         COMM
     *          FEE(6)        ,    FEED(6)        ,    FFF(3)         , COMM
     *          FFJ(3)        ,    FLAG1          ,    FLAG2          , COMM
     *          FLAG3         ,    FLAG4          ,    FN             , COMM
     *          FO(3)         ,    FO1(3)         ,    FO2(3)         , COMM
     *          FO3(3)        ,    F1(3)          ,    F11(3)         , COMM
     *          FPT(5)        ,                                         COMM
     *          F12(3)        ,    F13(3)                               COMM
      COMMON    GAIN(10)      ,    G3             ,                     COMM
     *          G3DOT         ,    G4             ,    G4DOT            COMM
      COMMON    H(3)          ,    HCMG(3)        ,    HDOT(3)        , COMM
     *          HI(3)         ,    HO(3)          ,    HW(6)          , COMM
     *          H1(3)         ,    H1PDOT(3)      ,    H1PRIM(3)      , COMM
     *          H3PRIM(3)     ,    H4PRIM(3)                            COMM
      COMMON    IB2F          ,    ICFA           ,    ICFB           , COMM
     *          ICFC          ,    ICFD           ,    IDOCK          , COMM
     *          IDOF(6)       ,                                         COMM
     *          IGRAVF        ,    IPNDLM         ,    IPNTCK         , COMM
     *          IPRINT        ,    IPROPF                               COMM
      COMMON    NCASE         ,    NCHECK         ,    NDECK          , COMM
     *          NGAIN         ,    NUMCMG                               COMM
      COMMON    OMEGA1        ,    OMEGA3         ,    OMEGA4           COMM
      COMMON    PEND3L        ,    PEND4L                               COMM
      COMMON    Q(4,4)                                                  COMM
      COMMON    R            ,    R0(3)        ,    R1(3)        , COMMC
     *          R1DOT(3)     ,    R1YCS        ,    R1YSN        , COMMC
     *          R1ZCS        ,    R1ZSN        ,    R2(3)        , COMMC
     *          R2DOT(3)     ,    R2YCS        ,    R2YSN        , COMMC
     *          R2ZCS        ,    R2ZSN        ,    R3(3)        , COMMC
     *          R3DOT(3)     ,    R3YCS        ,    R3YSN        , COMMC
     *          R3ZCS        ,    R3ZSN        ,    R4(3)        , COMMC
     *          R4DOT(3)     ,    R4YCS        ,    R4YSN        , COMMC
     *          R4ZCS        ,    R4ZSN                            COMMC
      COMMON    S            ,    SDOT         ,    SINFEJ       , COMMC
     *          SINTTJ       ,    SINTT0       ,    SINTT1       , COMMC
     *          SINTT2       ,    SINTT3       ,    SINTT4       , COMMC
     *          SP           ,    SUM1         ,    SUM2         , COMMC
     *          SUM3         ,    S2(3)        ,    S3(3)        , COMMC
     *          S4(3)                                              COMMC
      COMMON    T(3,3)       ,    TC(3,3)      ,    TEMP1(3)     , COMMC
     *          TEMP2(3)     ,                                     COMMC
     *          TEMP3(3)     ,    TEMP4(3)     ,    TEMP5(3,3)   , COMMC
     *          TEMP6(3,3)   ,    TEMP7(3,3)   ,    TEMP8(3,3)   , COMMC
     *          TEMP9(3,3)   ,    TEMP10(3,3)  ,    TEMP11(3,3)  , COMMC
     *          TEMP12(3,3)  ,    TEMP13(3,3)  ,    TEMP14(3,3)  , COMMC
     *          TEMP15(3,3)  ,    TERM1(3)     ,    TERM2(3)     , COMMC
     *          TFRICT       ,    THATA(6)     ,    THATAD(6)    , COMMC
     *          THETA1       ,    THETA3       ,    THETA4       , COMMC
     *          THETO        ,    TIBO(3,3)    ,    TIBOI(3,3)   , COMMC
     *          TIME         ,    TJ           ,    TJ1(10)      , COMMC
     *          TJ2(10)      ,    TJ3(10)      ,    TJ4(10)      , COMMC
     *          TMOTOR       ,                                     COMMC
     *          TOEF(3)      ,    TOTMAS       ,    TO1          , COMMC
     *          TQOG(3)      ,    TQOP(3)      ,    TQ1G(3)      , COMMC
     *          TQ1P(3)      ,    TSTART       ,    TSTOP        , COMMC
     *          TT1DOT       ,    TT3DOT       ,    TT4DOT       , COMMC
     *          T1EF(3)      ,    T13          ,    T14            COMMC
      COMMON    V(3)                                               COMMC
      COMMON    W0(3)        ,    WS           ,    W1(3)        , COMMC
     *          W3(3)        ,    W4(3)                            COMMC
      COMMON    X(6,7)       ,    XC           ,    XCDOT        , COMMC
     *          XMU                                                COMMC
C                                                                  COMMC
C     ****************************************************************
C     *
C     *
C     ****************************************************************
C                                                                  SDCAL
C     THIS SUBROUTINE CALCULATES THE ELEVATOR VELOCITY.            SDCAL
C                                                                  SDCAL
      SDOT = 0.                                                    SDCAL
      RETURN                                                       SDCAL
C                                                                  SDCAL
C     ****************************************************************
C     *
C     *
      END
      SUBROUTINE SYEQNS(A,N,NR,NC,FLAG)
C     *
C     *
C     *
C     ************************************************************************
C     *                                                                  SYEQN
C              LINEAR SIMULTANEOUS EQUATIONS                             SYEQN
C         A   AUGMENTED  MATRIX  (AC)   WHERE     AX = C                 SYEQN
C         ORIGINAL  DATA DESTROYED                                       SYEQN
C         N   NUMBER  OF EQUATIONS                                       SYEQN
C         SOLUTION   X  WILL  BE   COLUMN   N+1  OF  MATRIX  A           SYEQN
C         FLAG = 0.     SOLUTION  EXIST     F= 1.0  NO  SOLUTION         SYEQN
      DIMENSION A(NR,NC)                                                 SYEQN
      N1 = N                                                             SYEQN
      N2 = N1+1                                                          SYEQN
      N0 = N1 -1                                                         SYEQN
      FLAG = 0.                                                          SYEQN
      DO  60  I =1,N1                                                    SYEQN
      M = I                                                              SYEQN
      MI = M                                                             SYEQN
      IF(MI -N1)2,12,2                                                   SYEQN
 2    DO 10   II =M,N0                                                   SYEQN
      IF( ABS(A(MI,M))- ABS(A(II+1,M))) 5,10,10                          SYEQN
C     FIND LARGEST ABSOLUTE VALUE COLUMN M..   CALL IT  BIG              SYEQN
 5    MI = II+ 1                                                         SYEQN
 10   CONTINUE                                                           SYEQN
 12   BIG = A(MI,M)                                                      SYEQN
      IF( BIG )15,100,15                                                 SYEQN
C     BIG = 0 IMPLIES THERE IS NO SOLUTION                               SYEQN
 15   IF(MI- M)18,25,18                                                  SYEQN
 18   DO 20 JJ =M,N2                                                     SYEQN
      TEMP =A(M,JJ)                                                      SYEQN
      A(M,JJ) = A(MI,JJ)/BIG                                             SYEQN
C     NORMALIZE ROW MI AND EXCHANGE WITH ROW M                           SYEQN
 20   A(MI,JJ)= TEMP                                                     SYEQN
      GO TO 35                                                           SYEQN
 25   DO 30 JJ=M,N2                                                      SYEQN
C     NORMALIZE ROW M                                                    SYEQN
 30   A(M,JJ) =A(M,JJ)/BIG                                               SYEQN
 35   DO 50 II = 1,N1                                                    SYEQN
C     DO ROW OPERATIONS TO ZERO ELEMENTS OF COLUMN M EXCEPT FOR          SYEQN
C     ELEMENT A(M,M)                                                     SYEQN
      TEMP =A(II,M)                                                      SYEQN
      IF(M-II)38,50,38                                                   SYEQN
 38   IF(TEMP) 39,50,39                                                  SYEQN
 39   DO 40  JJ= M,N2                                                    SYEQN
      A(II,JJ)=A(II,JJ)- TEMP *A(M,JJ)                                   SYEQN
 40   CONTINUE                                                           SYEQN
 50   CONTINUE                                                           SYEQN
 60   CONTINUE                                                           SYEQN
 70   RETURN                                                             SYEQN
C     READ SOLUTION AS N ELEMENTS FROM COLUMN  (N+1)                     SYEQN
 100  FLAG =1.0                                                          SYEQN
C        SORRY THE COEFFICIENT MATRIX IS SINGULAR                        SYEQN
      RETURN                                                             SYEQN
C     *                                                                  SYEQN
C     ************************************************************************
C     *
C     *
C     *
      END
      SUBROUTINE TORKO1
C     *
C     *
C     *
C     ****************************************************************************
      COMMON      A(3)           ,    AE(5)          ,    AED(5)         , COMMO
     *            AFOUR(2)       ,    AII(6,3,3)     ,                     COMMO
     *            AIO(6,3,3)     ,    AJ1(3)         ,    ALT            , COMMO
     *            AOCJ(6,3,3)    ,    AOJ(3)         ,    AONE(7)        , COMMO
     *            ATCPT2(3,3)    ,    ATHREE(5)      ,    ATWO(4)        , COMMO
     *            A1(3,3)        ,    A1J(2)                               COMMO
      COMMON      BDMASS         ,    BFOUR(2)       ,    BMOM           , COMMO
     *            BODYDI(3,3)    ,    BODYOI(3,3)    ,    BODY1I(3,3)    , COMMO
     *            BOMASS         ,    BONE(7)        ,    BTHREE(5)      , COMMO
     *            BTWO(4)        ,    B1MASS         ,    B2MASS         , COMMO
     *            B3MASS         ,    B4MASS                               COMMO
      COMMON      CA(3)          ,    CB(3)          ,    CGAINO(3)      , COMMO
     *            CGAIN1(2)      ,                                         COMMO
     *            COSFEJ         ,    COSTTJ         ,    COSTTO         , COMMO
     *            COSTT1         ,    COSTT3         ,    COSTT4         , COMMO
     *            CO2T           ,    CP1            ,    CP2            , COMMO
     *            CST            ,    C1                                   COMMO
      COMMON      DB(3)          ,    DD01(3)        ,                     COMMO
     *            DELTAT         ,    D01(3)         ,    D01DOT(3)      , COMMO
     *            DTI(3)         ,                                         COMMO
     *            DTIME          ,    D12(3)         ,    D13(3)         , COMM
     *            D13DOT(3)      ,    D13YCS         ,    D13YSN         , COMM
     *            D13ZCS         ,    D13ZSN         ,    D14(3)         , COMM
     *            D14DOT(3)      ,    D14YCS         ,    D14YSN         , COMM
     *            D14ZCS         ,    D14ZSN                               COMM
      COMMON      EEE(3,3)       ,    EEJ(3,3)       ,    EL2(3)         , COMM
     *            EL2DOT(3)      ,    EL2YCS         ,    EL2YSN         , COMM
     *            EL2ZCS         ,    EL2ZSN         ,    EL3(3)         , COMM
     *            EL3DOT(3)      ,    EL3YCS         ,    EL3YSN         , COMM
     *            EL3ZCS         ,    EL3ZSN         ,    EL4(3)         , COMM
     *            EL4DOT(3)      ,    EL4YCS         ,    EL4YSN         , COMM
     *            EL4ZCS         ,    EL4ZSN         ,    EM(6,6)          COMM
      COMMON      FAT(8)         ,                                         COMM
     *            FEE(6)         ,    FEED(6)        ,    FFF(3)         , COMM
     *            FFJ(3)         ,    FLAG1          ,    FLAG2          , COMM
     *            FLAG3          ,    FLAG4          ,    FN             , COMM
     *            FO(3)          ,    FO1(3)         ,    FO2(3)         , COMM
     *            FO3(3)         ,    F1(3)          ,    F11(3)         , COMM
     *            FPT(5)         ,                                         COMM
     *            F12(3)         ,    F13(3)                               COMM
      COMMON      GAIN(10)       ,    G3             ,                     COMM
     *            G3DOT          ,    G4             ,    G4DOT            COMM
      COMMON      H(3)           ,    HCMG(3)        ,    HDOT(3)        , COMM
     *            HI(3)          ,    HO(3)          ,    HW(6)          , COMM
     *            H1(3)          ,    H1PDOT(3)      ,    H1PRIM(3)      , COMM
     *            H3PRIM(3)      ,    H4PRIM(3)                            COMM
      COMMON      IB2F           ,    ICFA           ,    ICFB           , COMM
     *            ICFC           ,    ICFD           ,    IDOCK          , COMM
     *            IDOF(6)        ,                                         COMM
     *            IGRAVF         ,    IPNDLM         ,    IPNTCK         , COMM
     *            IPRINT         ,    IPROPF                               COMM
      COMMON      NCASE          ,    NCHECK         ,    NDECK          , COMM
     *            NGAIN          ,    NUMCMG                               COMM
      COMMON      OMEGA1         ,    OMEGA3         ,    OMEGA4           COMM
      COMMON      PEND3L         ,    PEND4L                               COMM
      COMMON      Q(4,4)                                                   COMM
      COMMON   R             ,   RO(3)          ,   R1(3)          , COMMC
     *         R1DOT(3)      ,   R1YCS          ,   R1YSN          , COMMC
     *         R1ZCS         ,   R1ZSN          ,   R2(3)          , COMMC
     *         R2DOT(3)      ,   R2YCS          ,   R2YSN          , COMMC
     *         R2ZCS         ,   R2ZSN          ,   R3(3)          , COMMC
     *         R3DOT(3)      ,   R3YCS          ,   R3YSN          , COMMC
     *         R3ZCS         ,   R3ZSN          ,   R4(3)          , COMMC
     *         R4DOT(3)      ,   R4YCS          ,   R4YSN          , COMMC
     *         R4ZCS         ,   R4ZSN                               COMMC
      COMMON   S             ,   SDOT           ,   SINFEJ         , COMMC
     *         SINTTJ        ,   SINTTO         ,   SINTT1         , COMMC
     *         SINTT2        ,   SINTT3         ,   SINTT4         , COMMC
     *         SP            ,   SUM1           ,   SUM2           , COMMC
     *         SUM3          ,   S2(3)          ,   S3(3)          , COMMC
     *         S4(3)                                                 COMMC
      COMMON   T(3,3)        ,   TC(3,3)        ,   TEMP1(3)       , COMMC
     *         TEMP2(3)      ,                                       COMMC
     *         TEMP3(3)      ,   TEMP4(3)       ,   TEMP5(3,3)     , COMMC
     *         TEMP6(3,3)    ,   TEMP7(3,3)     ,   TEMP8(3,3)     , COMMC
     *         TEMP9(3,3)    ,   TEMP10(3,3)    ,   TEMP11(3,3)    , COMMC
     *         TEMP12(3,3)   ,   TEMP13(3,3)    ,   TEMP14(3,3)    , COMMC
     *         TEMP15(3,3)   ,   TERM1(3)       ,   TERM2(3)       , COMMC
     *         TFRICT        ,   THATA(6)       ,   THATAD(6)      , COMMC
     *         THETA1        ,   THETA3         ,   THETA4         , COMMC
     *         THETO         ,   TIBO(3,3)      ,   TIBOI(3,3)     , COMMC
     *         TIME          ,   TJ             ,   TJ1(10)        , COMMC
     *         TJ2(10)       ,   TJ3(10)        ,   TJ4(10)        , COMMC
     *         TMOTOR        ,                                       COMMC
     *         TOEF(3)       ,   TOTMAS         ,   TO1            , COMMC
     *         TQOG(3)       ,   TQOP(3)        ,   TQ1G(3)        , COMMC
     *         TQ1P(3)       ,   TSTART         ,   TSTOP          , COMMC
     *         TT1DOT        ,   TT3DOT         ,   TT4DOT         , COMMC
     *         T1EF(3)       ,   T13            ,   T14              COMMC
      COMMON   V(3)                                                  COMMC
      COMMON   WO(3)         ,   WS             ,   W1(3)          , COMMC
     *         W3(3)         ,   W4(3)                               COMMC
      COMMON   X(6,7)        ,   XC             ,   XCDOT          , COMMC
     *         XMU                                                   COMMC
C                                                                    COMMC
C     ****************************************************************
C     *
C     *
C     *
C     ****************************************************************
C                                                                    TORKC
C     THIS SUBROUTINE COMPUTES THE TORQUE BETWEEN BODY 0 AND BODY 1. TORKC
C     THIS TORQUE CONSISTS OF THE CONTROL TORQUE AND FRICTION TORQUE. TORKC
C                                                                    TORKC
       IF (TIME .NE. TSTART) GO TO 10                                TORKC
       XC=-GAIN(2)*OMEGA1                                            TORKC
       AFOUR(2)=XC                                                   TORKC
   10   CONTINUE                                                     TORKC
       AFOUR(1)=TIME                                                 TORKC
       BFOUR(1)=DELTAT                                               TORKC
       XCDOT=GAIN(3)*(OMEGA1-SP)                                     TORKC
       BFOUR(2)=XCDOT                                                TORKC
      CALL FOMS(AFOUR,BFOUR,2,FLAG4,TJ4)                             TORKC
       XC=AFOUR(2)                                                   TORKC
       TMOTOR=GAIN(2)*OMEGA1+XC                                      TORKC
       TFRICT=GAIN(1)*SIN(OMEGA1)                                    TORKC
       TO1=TMOTOR+TFRICT                                             TORKC
      RETURN                                                         TORKC
```

```
C                                                                       TORK(
C     ***********************************************************************
C     *
C     *
      END
      SUBROUTINE TORK13(T13,CP1,CP2,THETA3,OMEGA3)
C     *
C     *
C     ***********************************************************************
C                                                                      TORK1
C     THIS SUBROUTINE COMPUTES THE TORQUE BETWEEN BODY 1 AND BODY 3.    TORK1
C     THIS TORQUE CONSISTS OF THE CONTROL TORQUE AND FRICTION TORQUE.   TORK1
C                                                                      TORK1
      T13 = - CP1*OMEGA3 - CP2*THETA3                                  TORK1
      RETURN                                                           TORK1
C                                                                      TORK1
C     ***********************************************************************
C     *
C     *
      END
      SUBROUTINE TORK14(T14,CP1,CP2,THETA4,OMEGA4)
C     *
C     *
C     ******************************************************************
C                                                                      TORK1
C     THIS SUBROUTINE COMPUTES THE TORQUE BETWEEN BODY 1 AND BODY 4.   TORK1
C     THIS TORQUE CONSISTS OF THE CONTROL TORQUE AND FRICTION TORQUE.  TORK1
C                                                                      TORK1
      T14 = - CP1*OMEGA4 - CP2*(THETA4 - 3.14159)                      TORK1
      RETURN                                                           TORK1
C                                                                      TORK1
C     ******************************************************************
C     *
C     *
      END
      SUBROUTINE XDOT
C      *
C      *
C      *
C      ******************************************************************************
      COMMON    A(3)          ,  AE(5)          ,  AED(5)         , COMMC
     *          AFOUR(2)      ,  AII(6,3,3)     ,                   COMMC
     *          AIO(6,3,3)    ,  AJ1(3)         ,  ALT            , COMMC
     *          AOCJ(6,3,3)   ,  AOJ(3)         ,  AONE(7)        , COMMC
     *          ATCPT2(3,3)   ,  ATHREE(5)      ,  ATWO(4)        , COMMC
     *          A1(3,3)       ,  A1J(2)                             COMMC
      COMMON    BDMASS        ,  BFOUR(2)       ,  BMOM           , COMMC
     *          BODYDI(3,3)   ,  BODYOI(3,3)    ,  BODY1I(3,3)    , COMMC
     *          BOMASS        ,  BONE(7)        ,  BTHREE(5)      , COMMC
     *          BTWO(4)       ,  B1MASS         ,  B2MASS         , COMMC
     *          B3MASS        ,  B4MASS                             COMMC
      COMMON    CA(3)         ,  CB(3)          ,  CGAINO(3)      , COMMC
     *          CGAIN1(2)     ,                                     COMMC
     *          COSFEJ        ,  COSTTJ         ,  COSTTO         , COMMC
     *          COSTT1        ,  COSTT3         ,  COSTT4         , COMMC
     *          CO2T          ,  CP1            ,  CP2            , COMMC
     *          CST           ,  C1                                 COMMC
      COMMON    DB(3)         ,  DD01(3)        ,                   COMMC
     *          DELTAT        ,  D01(3)         ,  D01DOT(3)      , COMMC
     *          DTI(3)        ,                                     COMMC
     *          DTIME         ,  D12(3)         ,  D13(3)         , COMMC
     *          D13DOT(3)     ,  D13YCS         ,  D13YSN         , COMMC
     *          D13ZCS        ,  D13ZSN         ,  D14(3)         , COMMC
     *          D14DOT(3)     ,  D14YCS         ,  D14YSN         , COMMC
     *          D14ZCS        ,  D14ZSN                             COMMC
      COMMON    EEE(3,3)      ,  EEJ(3,3)       ,  EL2(3)         , COMMC
     *          EL2DOT(3)     ,  EL2YCS         ,  EL2YSN         , COMMC
     *          EL2ZCS        ,  EL2ZSN         ,  EL3(3)         , COMMC
     *          EL3DOT(3)     ,  EL3YCS         ,  EL3YSN         , COMMC
     *          EL3ZCS        ,  EL3ZSN         ,  EL4(3)         , COMMC
     *          EL4DOT(3)     ,  EL4YCS         ,  EL4YSN         , COMMC
     *          EL4ZCS        ,  EL4ZSN         ,  EM(6,6)          COMMC
      COMMON    FAT(8)        ,                                     COMMC
     *          FEE(6)        ,  FEED(6)        ,  FFF(3)         , COMMC
     *          FFJ(3)        ,  FLAG1          ,  FLAG2          , COMMC
     *          FLAG3         ,  FLAG4          ,  FN             , COMMC
     *          F0(3)         ,  F01(3)         ,  F02(3)         , COMMC
     *          F03(3)        ,  F1(3)          ,  F11(3)         , COMMC
     *          FPT(5)        ,                                     COMMC
     *          F12(3)        ,  F13(3)                             COMMC
      COMMON    GAIN(10)      ,  G3             ,                   COMMC
     *          G3DOT         ,  G4             ,  G6DOT            COMMC
      COMMON    H(3)          ,  HCMG(3)        ,  HDOT(3)        , COMMC
     *          HI(3)         ,  HO(3)          ,  HW(6)          , COMMC
     *          H1(3)         ,  H1PDOT(3)      ,  H1PRIM(3)      , COMMC
     *          H3PRIM(3)     ,  H4PRIM(3)                          COMMC
      COMMON    IB2E          ,  ICFA           ,  ICFB           , COMMC
     *          ICFC          ,  ICFD           ,  IDOCK          , COMMC
     *          IDOF(6)       ,                                     COMMC
     *          IGRAVF        ,  IPNDLM         ,  IPNTCK         , COMMC
     *          IPRINT        ,  IPROPF                             COMMC
      COMMON    NCASE         ,  NCHECK         ,  NDECK          , COMMC
     *          NGAIN         ,  NUMCMG                             COMMC
      COMMON    OMEGA1        ,  OMEGA3         ,  OMEGA4           COMMC
      COMMON    PEND3L        ,  PEND4L                             COMMC
      COMMON    Q(4,4)                                              COMMC
      COMMON     R              ,    R0(3)          ,    R1(3)          , COMMC
     *           R1DOT(3)       ,    R1YCS          ,    R1YSN          , COMMC
     *           R1ZCS          ,    R1ZSN          ,    R2(3)          , COMMC
     *           R2DOT(3)       ,    R2YCS          ,    R2YSN          , COMMC
     *           R2ZCS          ,    R2ZSN          ,    R3(3)          , COMMC
     *           R3DOT(3)       ,    R3YCS          ,    R3YSN          , COMMC
     *           R3ZCS          ,    R3ZSN          ,    R4(3)          , COMMC
     *           R4DOT(3)       ,    R4YCS          ,    R4YSN          , COMMC
     *           R4ZCS          ,    R4ZSN                                COMMC
      COMMON     S              ,    SDOT           ,    SINFEJ         , COMMC
     *           SINTTJ         ,    SINTT0         ,    SINTT1         , COMMC
     *           SINTT2         ,    SINTT3         ,    SINTT4         , COMMC
     *           SP             ,    SUM1           ,    SUM2           , COMMC
     *           SUM3           ,    S2(3)          ,    S3(3)          , COMMC
     *           S4(3)                                                    COMMC
      COMMON     T(3,3)         ,    TC(3,3)        ,    TEMP1(3)       , COMMC
     *           TEMP2(3)       ,                                         COMMC
     *           TEMP3(3)       ,    TEMP4(3)       ,    TEMP5(3,3)     , COMMC
     *           TEMP6(3,3)     ,    TEMP7(3,3)     ,    TEMP8(3,3)     , COMMC
     *           TEMP9(3,3)     ,    TEMP10(3,3)    ,    TEMP11(3,3)    , COMMC
     *           TEMP12(3,3)    ,    TEMP13(3,3)    ,    TEMP14(3,3)    , COMMC
     *           TEMP15(3,3)    ,    TERM1(3)       ,    TERM2(3)       , COMMC
     *           TFRICT         ,    THATA(6)       ,    THATAD(6)      , COMMC
     *           THETA1         ,    THETA3         ,    THETA4         , COMMC
     *           THET0          ,    TIB0(3,3)      ,    TIB0I(3,3)     , COMMC
     *           TIME           ,    TJ             ,    TJ1(10)        , COMMC
     *           TJ2(10)        ,    TJ3(10)        ,    TJ4(10)        , COMMC
     *           TMOTOR         ,                                         COMMC
     *           TOEF(3)        ,    TOTMAS         ,    T01            , COMMC
     *           TQ0G(3)        ,    TQ0P(3)        ,    TQ1G(3)        , COMMC
     *           TQ1P(3)        ,    TSTART         ,    TSTOP          , COMMC
     *           TT1DOT         ,    TT3DOT         ,    TT4DOT         , COMMC
     *           T1EF(3)        ,    T13            ,    T14              COMMC
      COMMON     V(3)                                                     COMMC
      COMMON     W0(3)          ,    WS             ,    W1(3)          , COMMC
     *           W3(3)          ,    W4(3)                                COMMC
      COMMON     X(6,7)         ,    XC             ,    XCDOT          , COMMC
     *           XMU                                                      COMMC
C                                                                         COMMC
C     ********************************************************************
C     *
C     *
C     *
C     ********************************************************************
C                                                                          XDOT
      SINTT1 = SIN(THETA1)                                                 XDOT
      COSTT1 = COS(THETA1)                                                 XDOT
      W1(1) =  W0(1) + OMEGA1                                              XDOT
      W1(2) =  W0(2)*COSTT1 + W0(3)*SINTT1                                 XDOT
      W1(3) = -W0(2)*SINTT1 + W0(3)*COSTT1                                 XDOT
      IF (TIME .NE. TSTART) GO TO 2                                        XDOT
      CALL RECALC                                                          XDOT
    2 CONTINUE                                                             XDOT
      CALL SDCALC                                                          XDOT
      EL2DOT(1) = -W1(3)*EL2(2) + W1(2)*EL2(3) + S2(1)*SDOT                XDOT
      EL2DOT(2) =  W1(3)*EL2(1) - W1(1)*EL2(3) + S2(2)*SDOT                XDOT
      EL2DOT(3) = -W1(2)*EL2(1) + W1(1)*EL2(2) + S2(3)*SDOT                XDOT
      R0(1) = R1(1) - D01(1)                                               XDOT
      R0(2) = R1(2)*COSTT1 - R1(3)*SINTT1 - D01(2)                         XDOT
      R0(3) = R1(2)*SINTT1 + R1(3)*COSTT1 - D01(3)                         XDOT
      W3(1) = W1(1)                                                        XDOT
      W3(2) = OMEGA3*S3(2) + W1(2)                                         XDOT
      W3(3) = OMEGA3*S3(3) + W1(3)                                        XDOT
      EL3DOT(1) = -W3(3)*EL3(2) + W3(2)*EL3(3)                            XDOT
      EL3DOT(2) =  W3(3)*EL3(1) - W3(1)*EL3(3)                            XDOT
      EL3DOT(3) = -W3(2)*EL3(1) + W3(1)*EL3(2)                            XDOT
      W4(1) = W1(1)                                                       XDOT
      W4(2) = OMEGA4*S4(2) + W1(2)                                        XDOT
      W4(3) = OMEGA4*S4(3) + W1(3)                                        XDOT
      EL4DOT(1) = -W4(3)*EL4(2) + W4(2)*EL4(3)                            XDOT
      EL4DOT(2) =  W4(3)*EL4(1) - W4(1)*EL4(3)                            XDOT
      EL4DOT(3) = -W4(2)*EL4(1) + W4(1)*EL4(2)                            XDOT
      D01DOT(1) = -W0(3)*D01(2) + W0(2)*D01(3)                            XDOT
      D01DOT(2) =  W0(3)*D01(1) - W0(1)*D01(3)                            XDOT
      D01DOT(3) = -W0(2)*D01(1) + W0(1)*D01(2)                            XDOT
      D13DOT(1) = -W1(3)*D13(2) + W1(2)*D13(3)                            XDOT
      D13DOT(2) =  W1(3)*D13(1) - W1(1)*D13(3)                            XDOT
      D13DOT(3) = -W1(2)*D13(1) + W1(1)*D13(2)                            XDOT
      D14DOT(1) = -W1(3)*D14(2) + W1(2)*D14(3)                            XDOT
      D14DOT(2) =  W1(3)*D14(1) - W1(1)*D14(3)                            XDOT
      D14DOT(3) = -W1(2)*D14(1) + W1(1)*D14(2)                            XDOT
      R1DOT(1) = (B0MASS/TOTMAS)*D01DOT(1)                                XDOT
     *           -(B2MASS/TOTMAS)*EL2DOT(1)                               XDOT
     *           -(B3MASS/TOTMAS)*(D13DOT(1) + EL3DOT(1))                 XDOT
     *           -(B4MASS/TOTMAS)*(D14DOT(1) + EL4DOT(1))                 XDOT
      R1DOT(2) = (B0MASS/TOTMAS)*(D01DOT(2)*COSTT1 + D01DOT(3)*SINTT1)    XDOT
     *           - (B2MASS/TOTMAS)*EL2DOT(2)                              XDOT
     *           -(B3MASS/TOTMAS)*(D13DOT(2) + EL3DOT(2))                 XDOT
     *           -(B4MASS/TOTMAS)*(D14DOT(2) + EL4DOT(2))                 XDOT
      R1DOT(3) = (B0MASS/TOTMAS)*(-D01DOT(2)*SINTT1 + D01DOT(3)*COSTT1)   XDOT
     *           - (B2MASS/TOTMAS)*EL2DOT(3)                              XDOT
     *           -(B3MASS/TOTMAS)*(D13DOT(3) + EL3DOT(3))                 XDOT
     *           -(B4MASS/TOTMAS)*(D14DOT(3) + EL4DOT(3))                 XDOT
      R2DOT(1) = R1DOT(1) + EL2DOT(1)                                     XDOT
      R2DOT(2) = R1DOT(2) + EL2DOT(2)                                     XDOT
      R2DOT(3) = R1DOT(3) + EL2DOT(3)                                     XDOT
      R3DOT(1) = R1DOT(1) + D13DOT(1) + EL3DOT(1)                         XDOT
      R3DOT(2) = R1DOT(2) + D13DOT(2) + EL3DOT(2)                         XDOT
      R3DOT(3) = R1DOT(3) + D13DOT(3) + EL3DOT(3)                         XDOT
      R4DOT(1) = R1DOT(1) + D14DOT(1) + EL4DOT(1)                         XDOT
      R4DOT(2) = R1DOT(2) + D14DOT(2) + EL4DOT(2)                         XDOT
      R4DOT(3) = R1DOT(3) + D14DOT(3) + EL4DOT(3)                         XDOT
      CALL MULT(H0,BODY0I,W0,DUM,DUM,DUM,1)                               XDOT
      CALL MULT(H1,BODY1I,W1,DUM,DUM,DUM,1)                               XDOT
      H3PRIM(1) = B3MASS*(-EL3(3)*R3DOT(2) + EL3(2)*R3DOT(3))             XDOT
      H3PRIM(2) = B3MASS*( EL3(3)*R3DOT(1) - EL3(1)*R3DOT(3))             XDOT
      H3PRIM(3) = B3MASS*(-EL3(2)*R3DOT(1) + EL3(1)*R3DOT(2))             XDOT
      H4PRIM(1) = B4MASS*(-EL4(3)*R4DOT(2) + EL4(2)*R4DOT(3))             XDOT
      H4PRIM(2) = B4MASS*( EL4(3)*R4DOT(1) - EL4(1)*R4DOT(3))             XDOT
      H4PRIM(3) = B4MASS*(-EL4(2)*R4DOT(1) + EL4(1)*R4DOT(2))             XDOT
      H1PRIM(2) = H1(2) + H3PRIM(2) + H4PRIM(2)                           XDOT
     *  -B2MASS*( - EL2(3)*R2DOT(1) + EL2(1)*R2DOT(3))                    XDOT
     *  -B3MASS*( - D13(3)*R3DOT(1) + D13(1)*R3DOT(3))                    XDOT
     *  -B4MASS*( - D14(3)*R4DOT(1) + D14(1)*R4DOT(3))                    XDOT
      H1PRIM(3) = H1(3) + H3PRIM(3) + H4PRIM(3)                           XDOT
     *  -B2MASS*( EL2(2)*R2DOT(1) - EL2(1)*R2DOT(2))                      XDOT
     *  -B3MASS*( D13(2)*R3DOT(1) - D13(1)*R3DOT(2))                      XDOT
     *  -B4MASS*( D14(2)*R4DOT(1) - D14(1)*R4DOT(2))                      XDOT
      IF (TIME .NE. TSTART) GO TO 5                                       XDOT
      H1PRIM(1) = H1(1) + H3PRIM(1) + H4PRIM(1)                           XDOT
     *  -B2MASS*(EL2(3)*R2DOT(2) - EL2(2)*R2DOT(3))                       XDOT
     *  -B3MASS*(D13(3)*R3DOT(2) - D13(2)*R3DOT(3))                       XDOT
     *  -B4MASS*(D14(3)*R4DOT(2) - D14(2)*R4DOT(3))                       XDOT
C     LET US DEFINE SOME INTERMEDIATE VALUES NEEDED TO COMPUTE H.         XDOT
```

```
      SUM1 = B1MASS*R1DOT(1) + B2MASS*R2DOT(1)                          XDOT
     *       +B3MASS*R3DOT(1) + B4MASS*R4DOT(1)                         XDOT
      SUM2 = B1MASS*R1DOT(2) + B2MASS*R2DOT(2)                          XDOT
     *       +B3MASS*R3DOT(2) + B4MASS*R4DOT(2)                         XDOT
      SUM3 = B1MASS*R1DOT(3) + B2MASS*R2DOT(3)                          XDOT
     *       +B3MASS*R3DOT(3) + B4MASS*R4DOT(3)                         XDOT
C     CALCULATE H.                                                      XDOT
      DO 4 I=1,3                                                        XDOT
      HCMG(I) = FFF(I)                                                  XDOT
      DO 4 J=1,3                                                        XDOT
      HCMG(I) = EEE(I,J)*WO(J) + HCMG(I)                                XDOT
    4 CONTINUE                                                          XDOT
      H(1) = HO(1) + H1PRIM(1)                                          XDOT
     *       +(-D01(3)*COSTT1 + D01(2)*SINTT1)*SUM2                     XDOT
     *       +( D01(3)*SINTT1 + D01(2)*COSTT1)*SUM3                     XDOT
     *       + HCMG(1)                                                  XDOT
      H(2) = HO(2) + COSTT1*H1PRIM(2) - SINTT1*H1PRIM(3)                XDOT
     *       + D01(3)*SUM1 - D01(1)*SINTT1*SUM2 - D01(1)*COSTT1*SUM3    XDOT
     *       + HCMG(2)                                                  XDOT
      H(3) = HO(3) + SINTT1*H1PRIM(2) + COSTT1*H1PRIM(3)                XDOT
     *       - D01(2)*SUM1 + D01(1)*COSTT1*SUM2 - D01(1)*SINTT1*SUM3    XDOT
     *       + HCMG(3)                                                  XDOT
    5 CONTINUE                                                          XDOT
C     COMPUTE THE UNIT VECTOR J1.                                       XDOT
      AJ1(1) = B2MASS*EL2(1)                                            XDOT
     *         +B3MASS*D13(1) + B3MASS*EL3(1)                           XDOT
     *         +B4MASS*D14(1) + B4MASS*EL4(1)                           XDOT
      AJ1(2) = B2MASS*EL2(2)                                            XDOT
     *         +B3MASS*D13(2) + B3MASS*EL3(2)                           XDOT
     *         +B4MASS*D14(2) + B4MASS*EL4(2)                           XDOT
      AJ1(3) = B2MASS*EL2(3)                                            XDOT
     *         +B3MASS*D13(3) + B3MASS*EL3(3)                           XDOT
     *         +B4MASS*D14(3) + B4MASS*EL4(3)                           XDOT
C     UPDATE THE ORBIT ANGLE                                            XDOT
      THETO = TIME*WS                                                   XDOT
      IF (IGRAVF .EQ. 0) GO TO 10                                       XDOT
C     CALCULATE THE GRAVITY GRADIENT FORCES AND TORQUES.                XDOT
      CALL GGRAD                                                        XDOT
   10 CONTINUE                                                          XDOT
C     SUM THE FORCES ON BODY ZERO.                                      XDOT
      FO(1) = FO1(1)                                                    XDOT
      FO(2) = FO1(2)                                                    XDOT
      FO(3) = FO1(3)                                                    XDOT
C     SUM THE FORCES ON BODY ONE.                                       XDOT
      F1(1) = F11(1)                                                    XDOT
      F1(2) = F11(2)                                                    XDOT
      F1(3) = F11(3)                                                    XDOT
C     SUM THE TORQUES ON BODY ZERO.                                     XDOT
      TOEF(1) = TQOG(1) + TQOP(1)                                       XDOT
      TOEF(2) = TQOG(2) + TQOP(2)                                       XDOT
      TOEF(3) = TQOG(3) + TQOP(3)                                       XDOT
C     SUM THE TORQUES ON BODY ONE.                                      XDOT
      T1EF(1) = TQ1G(1) + TQ1P(1)                                       XDOT
      T1EF(2) = TQ1G(2) + TQ1P(2)                                       XDOT
      T1EF(3) = TQ1G(3) + TQ1P(3)                                       XDOT
C     LET US DEFINE SOME INTERMEDIATE TERMS USED TO CALCULATE HDOT.     XDOT
      TERM1(1) = (BOMASS - TOTMAS)*D01(1) - AJ1(1)                      XDOT
      TERM1(2) = (BOMASS - TOTMAS)*D01(2)                               XDOT
     *  - COSTT1*AJ1(2) + SINTT1*AJ1(3)                                 XDOT
      TERM1(3) = (BOMASS - TOTMAS)*D01(3)                               XDOT
     *  - SINTT1*AJ1(2) - COSTT1*AJ1(3)                                 XDOT
      TERM2(1) = BOMASS*D01(1) - AJ1(1)                                 XDOT
      TERM2(2) = BOMASS*( COSTT1*DO1(2) + SINTT1*DO1(3)) - AJ1(2)       XDOT
      TERM2(3) = BOMASS*(-SINTT1*DO1(2) + COSTT1*DO1(3)) - AJ1(3)       XDOT
      ATCPT2(1,1) =  0.                                                 XDOT
      ATCPT2(1,2) = -TERM2(3)                                           XDOT
      ATCPT2(1,3) =  TERM2(2)                                           XDOT
      ATCPT2(2,1) =  COSTT1*TERM2(3) + SINTT1*TERM2(2)                  XDOT
      ATCPT2(2,2) = -SINTT1*TERM2(1)                                    XDOT
      ATCPT2(2,3) = -COSTT1*TERM2(1)                                    XDOT
      ATCPT2(3,1) =  SINTT1*TERM2(3) - COSTT1*TERM2(2)                  XDOT
      ATCPT2(3,2) =  COSTT1*TERM2(1)                                    XDOT
      ATCPT2(3,3) = -SINTT1*TERM2(1)                                    XDOT
      HDOT(1) = WO(3)*H(2) - WO(2)*H(3)                                 XDOT
     *          +(-TERM1(3)*FO(2) + TERM1(2)*FO(3))/TOTMAS              XDOT
     *+(ATCPT2(1,1)*F1(1)+ATCPT2(1,2)*F1(2)+ATCPT2(1,3)*F1(3))/TOTMAS   XDOT
     *+ TOEF(1) + T1EF(1)                                               XDOT
      HDOT(2) = -WO(3)*H(1) + WO(1)*H(3)                                XDOT
     *          +( TERM1(3)*FO(1) - TERM1(1)*FO(3))/TOTMAS              XDOT
     *+(ATCPT2(2,1)*F1(1)+ATCPT2(2,2)*F1(2)+ATCPT2(2,3)*F1(3))/TOTMAS   XDOT
     *+ TOEF(2) + COSTT1*T1EF(2) - SINTT1*T1EF(3)                       XDOT
      HDOT(3) =  WO(2)*H(1) - WO(1)*H(2)                                XDOT
     *          + (-TERM1(2)*FO(1) + TERM1(1)*FO(2))/TOTMAS             XDOT
     *+(ATCPT2(3,1)*F1(1)+ATCPT2(3,2)*F1(2)+ATCPT2(3,3)*F1(3))/TOTMAS   XDOT
     *+ TOEF(3) + SINTT1*T1EF(2) + COSTT1*T1EF(3)                       XDOT
C     CALCULATE THE TORQUE BETWEEN BODY 0 AND BODY 1. (CONTROL,FRICTION)XDOT
      CALL TORK01                                                       XDOT
      H1PDOT(1) = -W1(2)*H1PRIM(3) + W1(3)*H1PRIM(2)                    XDOT
     * +R1DOT(2)*(-B2MASS*EL2DOT(3)-B3MASS*(D13DOT(3)+EL3DOT(3))-       XDOT
     * B4MASS*(D14DOT(3)+EL4DOT(3)))-R1DOT(3)*(-B2MASS*EL2DOT(2)-       XDOT
     * B3MASS*(D13DOT(2)+EL3DOT(2))-B4MASS*(D14DOT(2)+EL4DOT(2)))       XDOT
     *-AJ1(2)*(-FO(2)*SINTT1/TOTMAS+FO(3)*COSTT1/TOTMAS)                XDOT
     *+AJ1(3)*( FO(2)*COSTT1/TOTMAS+FO(3)*SINTT1/TOTMAS)                XDOT
     *+AJ1(3)*F1(2)/TOTMAS-AJ1(2)*F1(3)/TOTMAS+T1EF(1)+T01              XDOT
C     CALCULATE THE TORQUE BETWEEN BODY 1 AND BODY 3. (CONTROL,FRICTION)XDOT
      CALL TORK13(T13,CP1,CP2,THETA3,OMEGA3)                            XDOT
      G3DOT=-S3(2)*(W1(3)*H3PRIM(1)-W1(1)*H3PRIM(3))-S3(3)*(W1(1)*H3PRIMXDOT
     *(2)-W1(2)*H3PRIM(1))+B3MASS*S3(2)*(EL3DOT(3)*R3DOT(1)-EL3DOT(1)*R3XDOT
     *DOT(3))+B3MASS*S3(3)*(EL3DOT(1)*R3DOT(2)-EL3DOT(2)*R3DOT(1))-     XDOT
     *(B3MASS/TOTMAS)*S3(2)*(EL3(3)*(FO(1)+F1(1))-EL3(1)*(-FO(2)*SINTT1+XDOT
     *FO(3)*COSTT1+F1(3)))-(B3MASS/TOTMAS)*S3(3)*(EL3(1)*(FO(2)*COSTT1+ XDOT
     *FO(3)*SINTT1+F1(2))-EL3(2)*(FO(1)+F1(1)))+T13                     XDOT
C     CALCULATE THE TORQUE BETWEEN BODY 1 AND BODY 4. (CONTROL,FRICTION)XDOT
      CALL TORK14(T14,CP1,CP2,THETA4,OMEGA4)                            XDOT
      G4DOT=-S4(2)*(W1(3)*H4PRIM(1)-W1(1)*H4PRIM(3))-S4(3)*(W1(1)*H4PRIMXDOT
     *(2)-W1(2)*H4PRIM(1))+B4MASS*S4(2)*(EL4DOT(3)*R4DOT(1)-EL4DOT(1)*R4XDOT
     *DOT(3))+B4MASS*S4(3)*(EL4DOT(1)*R4DOT(2)-EL4DOT(2)*R4DOT(1))-     XDOT
     *(B4MASS/TOTMAS)*S4(2)*(EL4(3)*(FO(1)+F1(1))-EL4(1)*(-FO(2)*SINTT1+XDOT
     *FO(3)*COSTT1+F1(3)))-(B4MASS/TOTMAS)*S4(3)*(EL4(1)*(FO(2)*COSTT1+ XDOT
     *FO(3)*SINTT1+F1(2))-EL4(2)*(FO(1)+F1(1)))+T14                     XDOT
      RETURN                                                            XDOT
C     **********************************************************************
C     *
C     *
C     *
      END
```

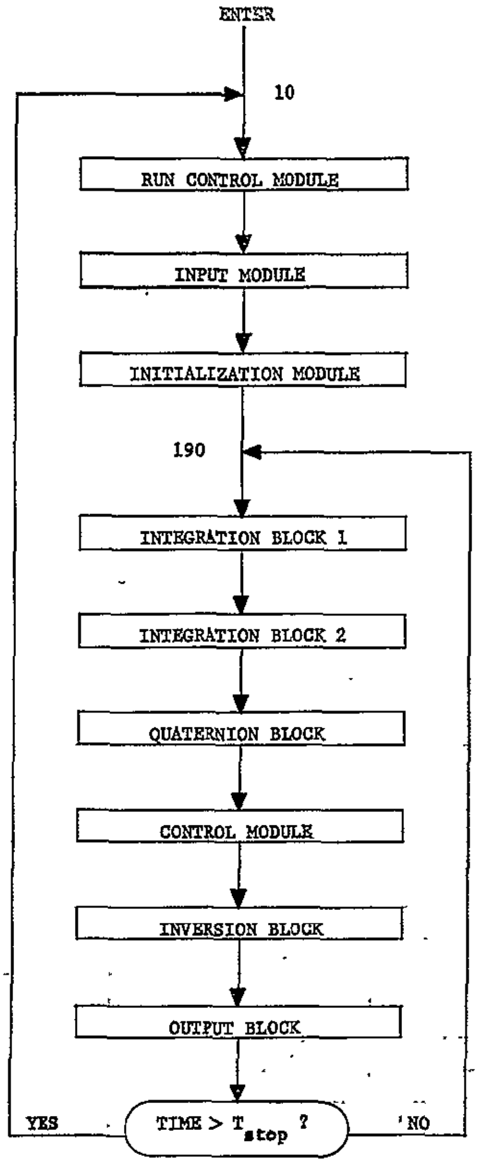
APPENDIX D, PROGRAM FLOW CHARTS

FLOW CHART & BLOCK DIAGRAM

FORM DEN 1103-01 (4-64)

Application MAIN PROGRAM FLOW CHART Date OCTOBER 1970 Page 1 of 1

Procedure Drawn By GARY JOHNSON

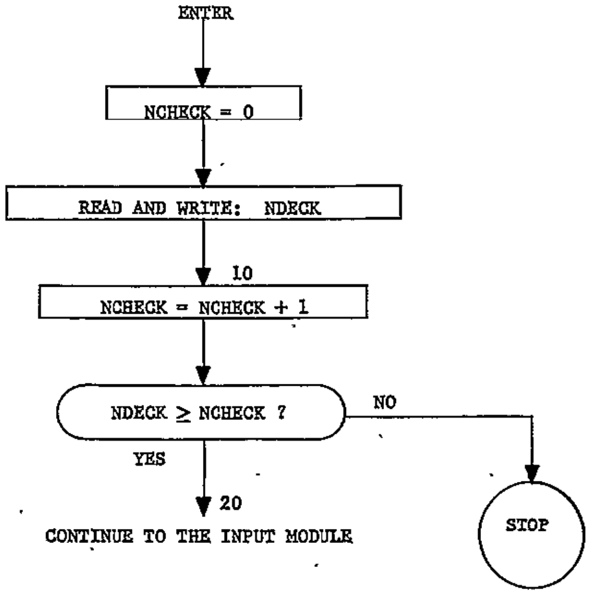


FLOW CHART & BLOCK DIAGRAM

FORM DES 1103-01 (4-64)

Application RUN CONTROL MODULE Date OCTOBER 1970 Page 1 of 1

Procedure Drawn By GARY JOHNSON

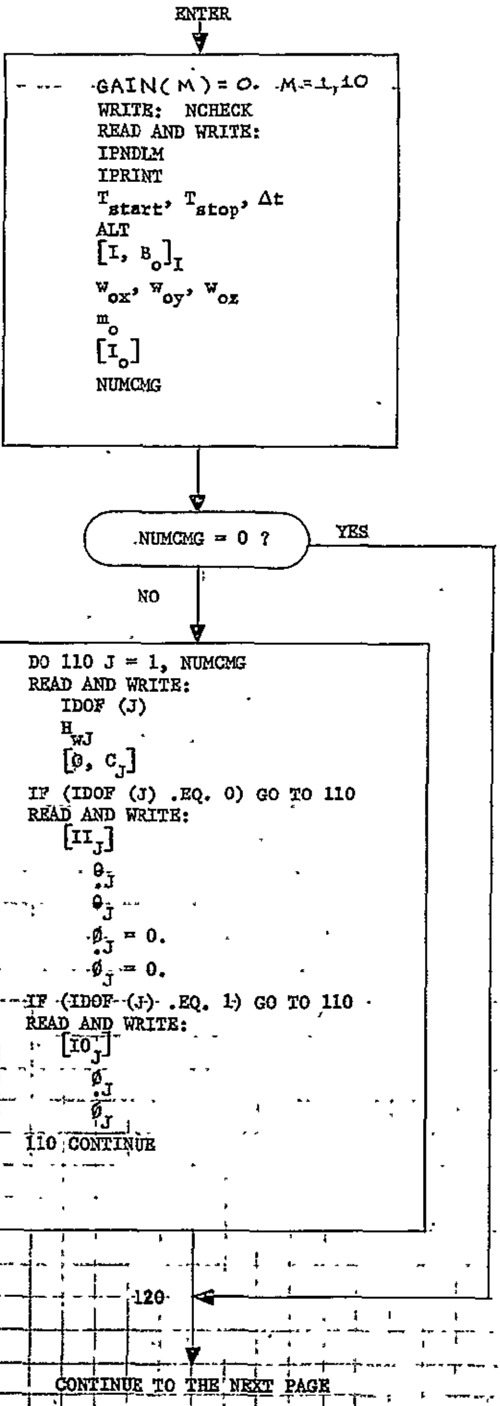


FLOW CHART & BLOCK DIAGRAM

FORM DEN 1103-01 (4-64)

Application INPUT MODULE Date OCTOBER 1970 Page 1 of 5

Procedure Drawn By GARY JOHNSON

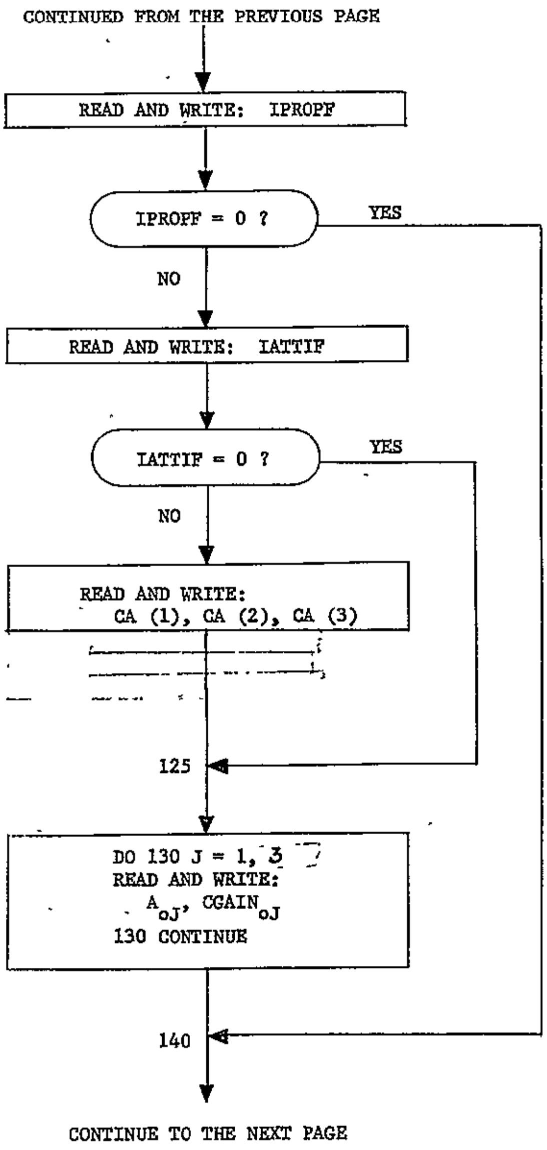


```
GAIN(M) = 0.  M=1,10
WRITE:  NCHECK
READ AND WRITE:
IPNDLM
IPRINT
$T_{start}$, $T_{stop}$, $\Delta t$
ALT
$[I, B_o]_I$
$w_{ox}$, $w_{oy}$, $w_{oz}$
$m_o$
$[I_o]$
NUMCMG

NUMCMG = 0 ?

DO 110 J = 1, NUMCMG
READ AND WRITE:
   IDOF (J)
   $H_{wJ}$
   $[\theta, C_J]$
IF (IDOF (J) .EQ. 0) GO TO 110
READ AND WRITE:
   $[II_J]$
     $\dot{\theta}_J$
     $\theta_J$
     $\dot{\phi}_J = 0.$
     $\phi_J = 0.$
IF (IDOF (J) .EQ. 1) GO TO 110
READ AND WRITE:
   $[IO_J]$
     $\dot{\phi}_J$
     $\phi_J$
110 CONTINUE
```

FLOW CHART & BLOCK DIAGRAM

Application INPUT MODULE CONTINUED Date OCTOBER 1970 Page 2 of 5

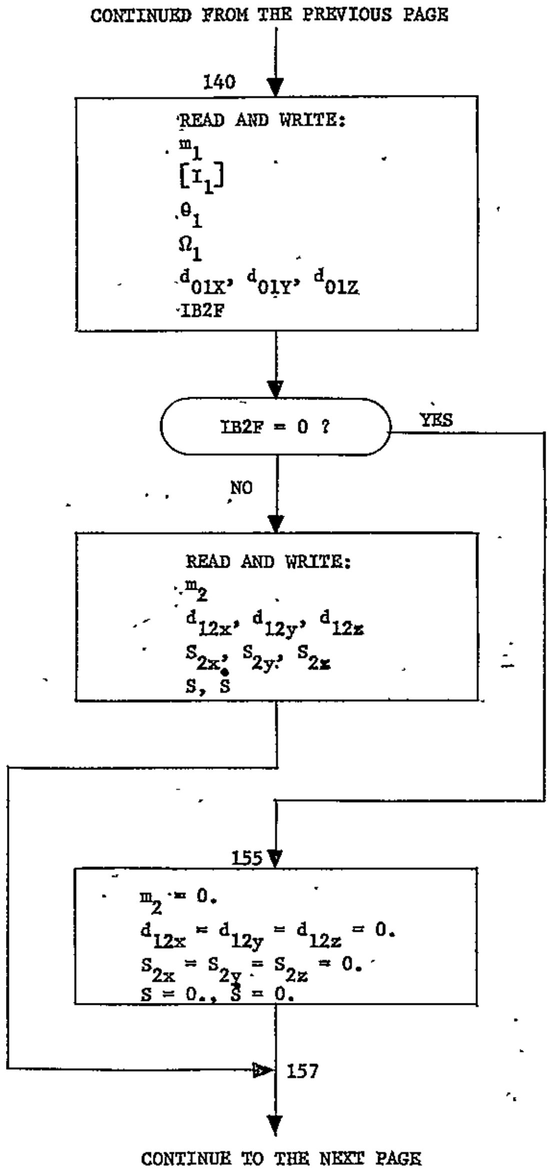
Procedure ________ Drawn By GARY JOHNSON



FLOW CHART & BLOCK DIAGRAM

Application INPUT MODULE CONTINUED Date OCTOBER 1970 Page 3 of 5

Procedure ______ Drawn By GARY JOHNSON

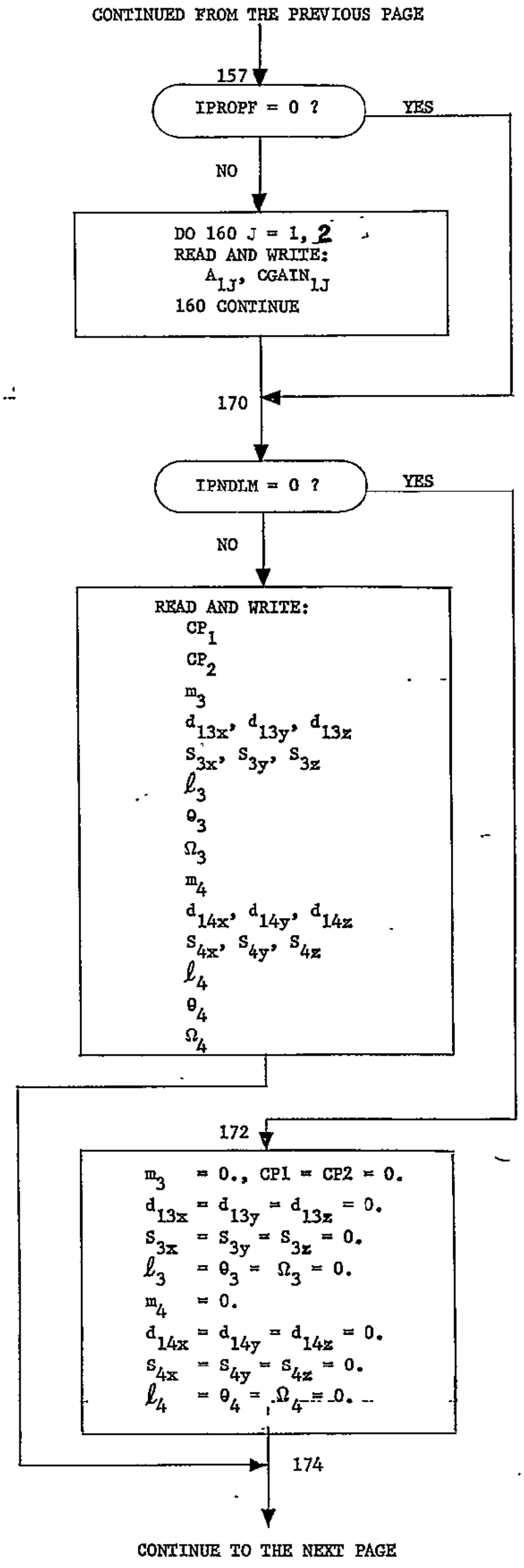

CONTINUED FROM THE PREVIOUS PAGE
140
READ AND WRITE:
m_1
$[I_1]$
θ_1
Ω_1
d_{01X}, d_{01Y}, d_{01Z}
IB2F
IB2F = 0 ?
YES
NO
READ AND WRITE:
m_2
d_{12x}, d_{12y}, d_{12z}
S_{2x}, S_{2y}, S_{2z}
S, $\dot{S}$
155
$m_2 = 0.$
$d_{12x} = d_{12y} = d_{12z} = 0.$
$S_{2x} = S_{2y} = S_{2z} = 0.$
$S = 0., \dot{S} = 0.$
157
CONTINUE TO THE NEXT PAGE

Fold to center line

FLOW CHART & BLOCK DIAGRAM

Application INPUT MODULE CONTINUED Date OCTOBER 1970 Page 4 of 5

Procedure Drawn By GARY JOHNSON

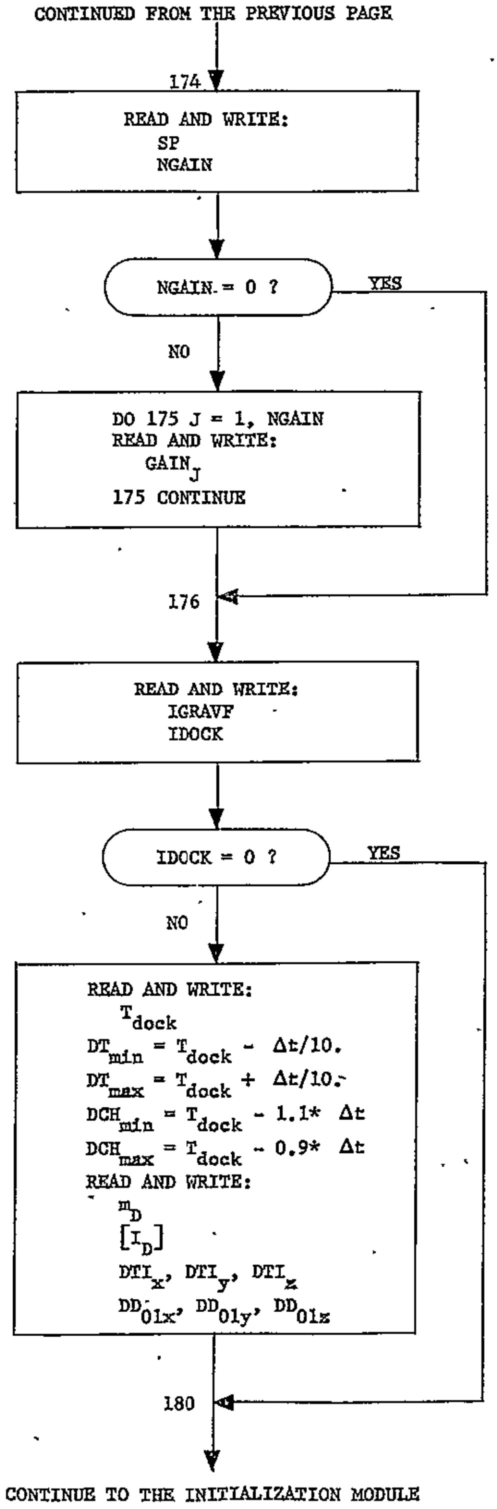


CONTINUED FROM THE PREVIOUS PAGE
157
IPROPF = 0 ?
YES
NO
DO 160 J = 1, 2
READ AND WRITE:
A_{1J}, $CGAIN_{1J}$
160 CONTINUE
170
IPNDLM = 0 ?
YES
NO
READ AND WRITE:
CP_1
CP_2
m_3
d_{13x}, d_{13y}, d_{13z}
S_{3x}, S_{3y}, S_{3z}
ℓ_3
θ_3
Ω_3
m_4
d_{14x}, d_{14y}, d_{14z}
S_{4x}, S_{4y}, S_{4z}
ℓ_4
θ_4
Ω_4
172
$m_3 = 0.$, $CP1 = CP2 = 0.$
$d_{13x} = d_{13y} = d_{13z} = 0.$
$S_{3x} = S_{3y} = S_{3z} = 0.$
$\ell_3 = \theta_3 = \Omega_3 = 0.$
$m_4 = 0.$
$d_{14x} = d_{14y} = d_{14z} = 0.$
$S_{4x} = S_{4y} = S_{4z} = 0.$
$\ell_4 = \theta_4 = \Omega_4 = 0.$
174
CONTINUE TO THE NEXT PAGE

FLOWCHART & BLOCK DIAGRAM

Application INPUT MODULE CONTINUED Date OCTOBER 1970 Page 5 of 5

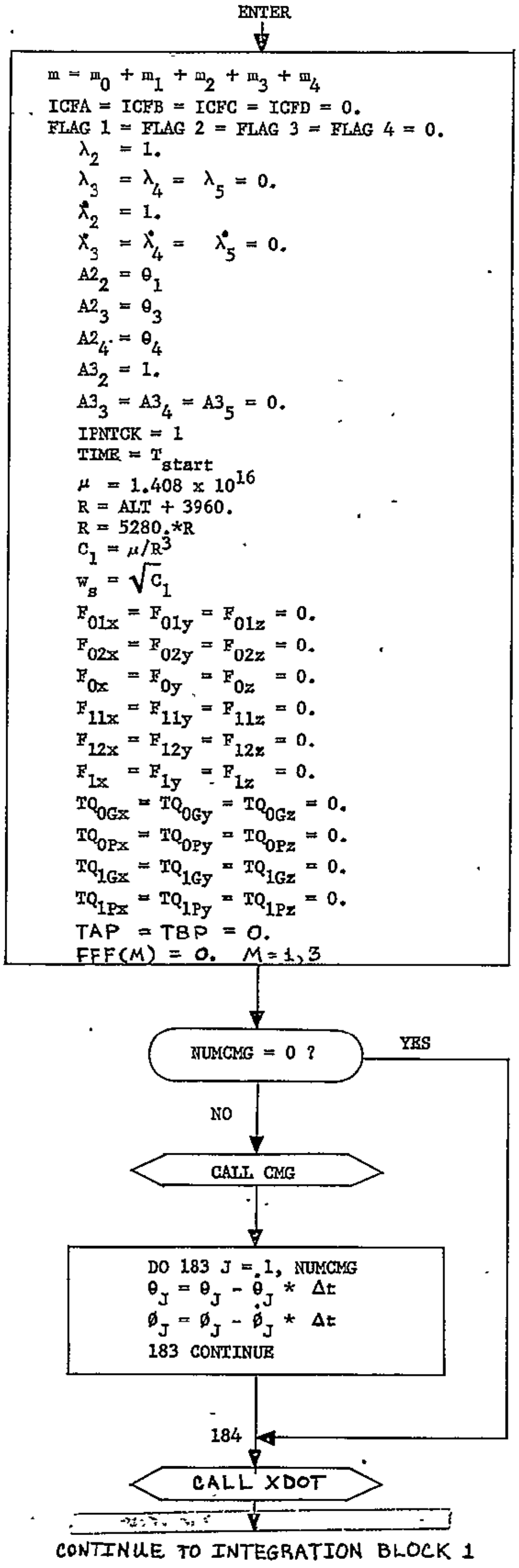
Procedure Drawn By GARY JOHNSON


CONTINUED FROM THE PREVIOUS PAGE
174
READ AND WRITE:
SP
NGAIN
NGAIN = 0 ?
YES
NO
DO 175 J = 1, NGAIN
READ AND WRITE:
$GAIN_J$
175 CONTINUE
176
READ AND WRITE:
IGRAVF
IDOCK
IDOCK = 0 ?
YES
NO
READ AND WRITE:
T_{dock}
$DT_{min} = T_{dock} - \Delta t/10.$
$DT_{max} = T_{dock} + \Delta t/10.$
$DCH_{min} = T_{dock} - 1.1* \Delta t$
$DCH_{max} = T_{dock} - 0.9* \Delta t$
READ AND WRITE:
m_D
$[I_D]$
DTI_x, DTI_y, DTI_z
$DD_{01x}, DD_{01y}, DD_{01z}$
180
CONTINUE TO THE INITIALIZATION MODULE

FLOW CHART & BLOCK DIAGRAM

Application INITIALIZATION MODULE Date OCTOBER 1970 Page 1 of 1

Procedure Drawn By GARY JOHNSON

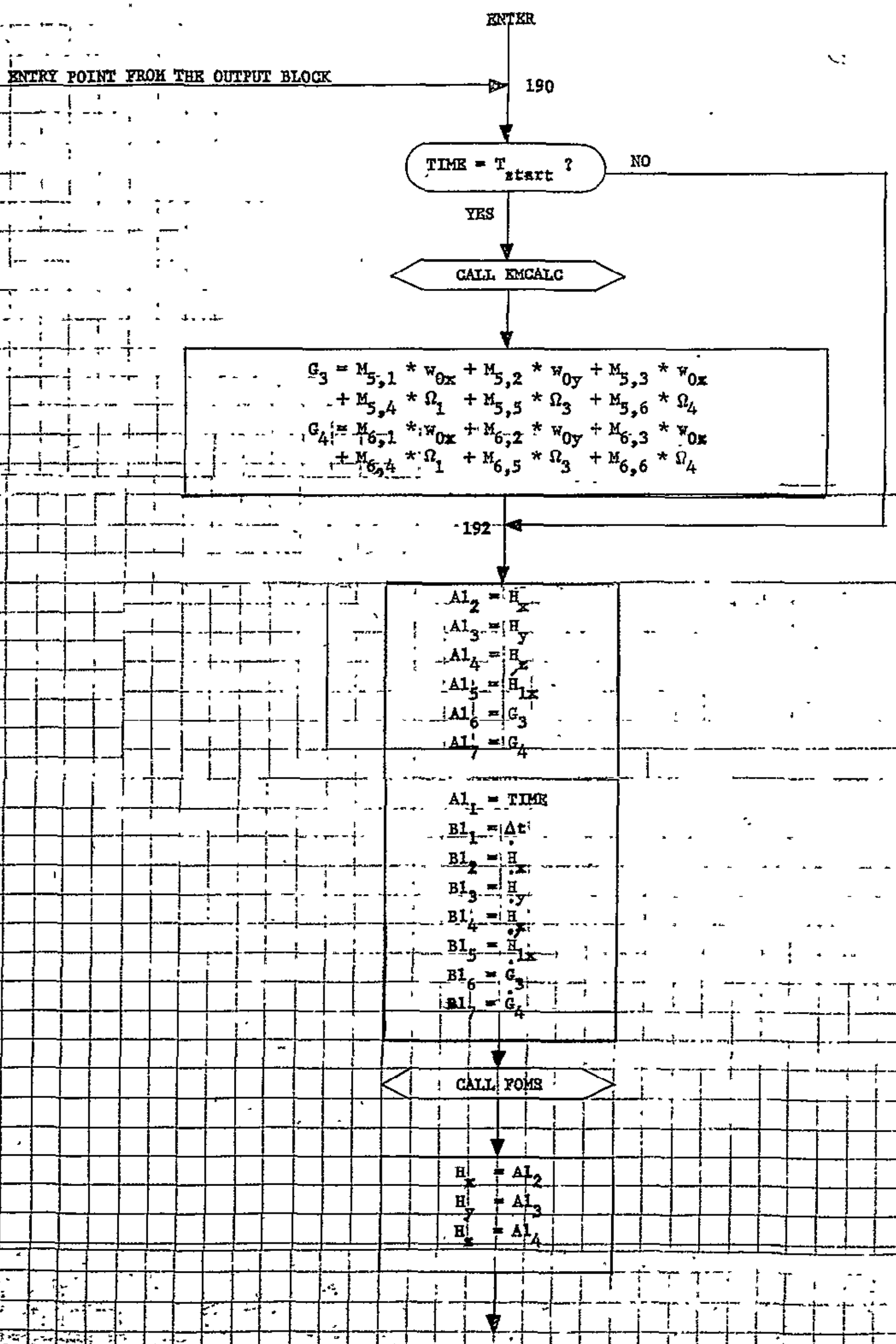

ENTER
$m = m_0 + m_1 + m_2 + m_3 + m_4$
ICFA = ICFB = ICFC = ICFD = 0.
FLAG 1 = FLAG 2 = FLAG 3 = FLAG 4 = 0.
$\lambda_2 = 1.$
$\lambda_3 = \lambda_4 = \lambda_5 = 0.$
$\dot{\lambda}_2 = 1.$
$\dot{\lambda}_3 = \dot{\lambda}_4 = \dot{\lambda}_5 = 0.$
$A2_2 = \theta_1$
$A2_3 = \theta_3$
$A2_4 = \theta_4$
$A3_2 = 1.$
$A3_3 = A3_4 = A3_5 = 0.$
IPNTCK = 1
$TIME = T_{start}$
$\mu = 1.408 \times 10^{16}$
R = ALT + 3960.
R = 5280.*R
$C_1 = \mu/R^3$
$w_s = \sqrt{C_1}$
$F_{01x} = F_{01y} = F_{01z} = 0.$
$F_{02x} = F_{02y} = F_{02z} = 0.$
$F_{0x} = F_{0y} = F_{0z} = 0.$
$F_{11x} = F_{11y} = F_{11z} = 0.$
$F_{12x} = F_{12y} = F_{12z} = 0.$
$F_{1x} = F_{1y} = F_{1z} = 0.$
$TQ_{0Gx} = TQ_{0Gy} = TQ_{0Gz} = 0.$
$TQ_{0Px} = TQ_{0Py} = TQ_{0Pz} = 0.$
$TQ_{1Gx} = TQ_{1Gy} = TQ_{1Gz} = 0.$
$TQ_{1Px} = TQ_{1Py} = TQ_{1Pz} = 0.$
TAP = TBP = 0.
FFF(M) = 0. M=1,3
NUMCMG = 0 ?
YES
NO
CALL CMG
DO 183 J = 1, NUMCMG
$\theta_J = \theta_J - \dot{\theta}_J * \Delta t$
$\emptyset_J = \emptyset_J - \dot{\emptyset}_J * \Delta t$
183 CONTINUE
184
CALL XDOT
CONTINUE TO INTEGRATION BLOCK 1

FLOW CHART & BLOCK DIAGRAM

FORM DEN 1103-01 (4-64)

Application INTEGRATION BLOCK 1 Date OCTOBER 1970 Page 1 of 2

Procedure Drawn By GARY JOHNSON


ENTER
ENTRY POINT FROM THE OUTPUT BLOCK
190
TIME = T_{start} ?
NO
YES
CALL EMCALC
$G_3 = M_{5,1} * w_{0x} + M_{5,2} * w_{0y} + M_{5,3} * w_{0x}$
$+ M_{5,4} * \Omega_1 + M_{5,5} * \Omega_3 + M_{5,6} * \Omega_4$
$G_4 = M_{6,1} * w_{0x} + M_{6,2} * w_{0y} + M_{6,3} * w_{0x}$
$+ M_{6,4} * \Omega_1 + M_{6,5} * \Omega_3 + M_{6,6} * \Omega_4$
192
$A1_2 = H_x$
$A1_3 = H_y$
$A1_4 = H_z$
$A1_5 = H_{1x}$
$A1_6 = G_3$
$A1_7 = G_4$
$A1_1 = TIME$
$B1_1 = \underset{\cdot}{\Delta t}$
$B1_2 = \underset{\cdot}{H}_x$
$B1_3 = \underset{\cdot}{H}_y$
$B1_4 = \underset{\cdot}{H}_z$
$B1_5 = \underset{\cdot}{H}_{1x}$
$B1_6 = \underset{\cdot}{G}_3$
$B1_7 = \underset{\cdot}{G}_4$
CALL FOMS
$H_x = A1_2$
$H_y = A1_3$
$H_z = A1_4$

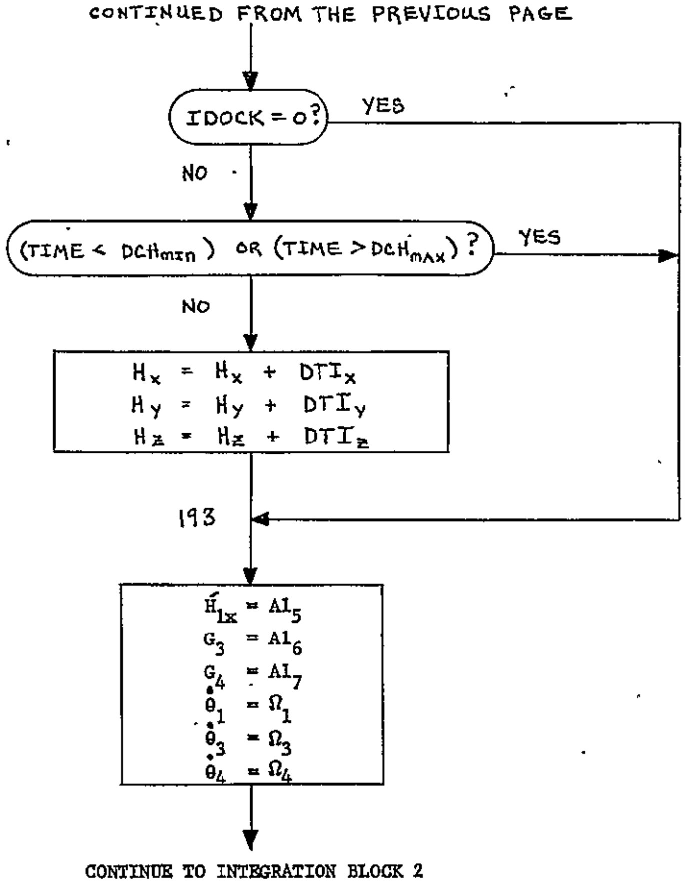
CONTINUE TO THE FOLLOWING PAGE

FLOW CHART & BLOCK DIAGRAM

FORM DEN 1103-01 (4-54)

Application INTEGRATION BLOCK 1 CONTINUED Date OCTOBER 1970 Page 2 of 2

Procedure Drawn By GARY JOHNSON

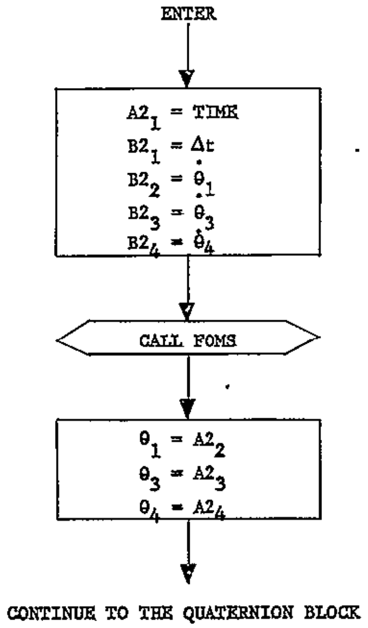

CONTINUED FROM THE PREVIOUS PAGE
IDOCK = 0?
YES
NO
(TIME < DCH$_{min}$) OR (TIME > DCH$_{MAX}$)?
YES
NO
$H_x = H_x + DTI_x$
$H_y = H_y + DTI_y$
$H_z = H_z + DTI_z$
193
$\acute{H}_{1x} = AI_5$
$G_3 = AI_6$
$G_4 = AI_7$
$\dot{\theta}_1 = \Omega_1$
$\dot{\theta}_3 = \Omega_3$
$\dot{\theta}_4 = \Omega_4$
CONTINUE TO INTEGRATION BLOCK 2

FLOW CHART & BLOCK DIAGRAM

Form DEN 1103-01 (4-64)

Application INTEGRATION BLOCK 2 Date OCTOBER 1970 Page 1 of 1

Procedure ______ Drawn By GARY JOHNSON

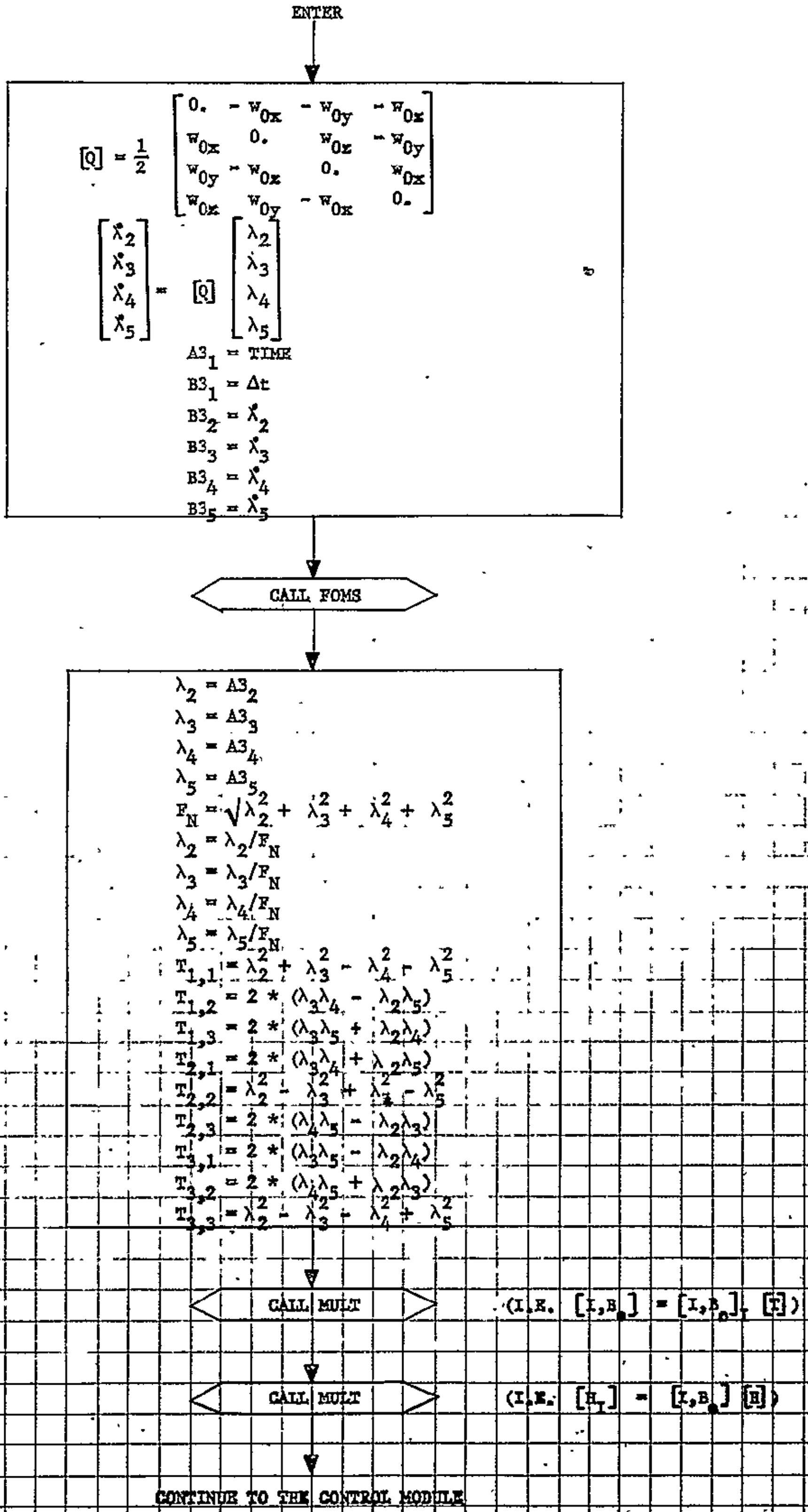


FLOW CHART & BLOCK DIAGRAM

FORM DEN 1103-01 (4-64)

Application QUATERNION BLOCK Date OCTOBER 1970 Page 1 of 1

Procedure Drawn By GARY JOHNSON



ENTER

$$[Q] = \frac{1}{2}\begin{bmatrix} 0. & -w_{0x} & -w_{0y} & -w_{0z} \\ w_{0x} & 0. & w_{0z} & -w_{0y} \\ w_{0y} & -w_{0z} & 0. & w_{0x} \\ w_{0z} & w_{0y} & -w_{0x} & 0. \end{bmatrix}$$

$$\begin{bmatrix} \dot{\lambda}_2 \\ \dot{\lambda}_3 \\ \dot{\lambda}_4 \\ \dot{\lambda}_5 \end{bmatrix} = [Q]\begin{bmatrix} \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \end{bmatrix}$$

$A3_1$ = TIME

$B3_1 = \Delta t$

$B3_2 = \dot{\lambda}_2$

$B3_3 = \dot{\lambda}_3$

$B3_4 = \dot{\lambda}_4$

$B3_5 = \dot{\lambda}_5$

CALL FOMS

$\lambda_2 = A3_2$

$\lambda_3 = A3_3$

$\lambda_4 = A3_4$

$\lambda_5 = A3_5$

$F_N = \sqrt{\lambda_2^2 + \lambda_3^2 + \lambda_4^2 + \lambda_5^2}$

$\lambda_2 = \lambda_2/F_N$

$\lambda_3 = \lambda_3/F_N$

$\lambda_4 = \lambda_4/F_N$

$\lambda_5 = \lambda_5/F_N$

$T_{1,1} = \lambda_2^2 + \lambda_3^2 - \lambda_4^2 - \lambda_5^2$

$T_{1,2} = 2 * (\lambda_3\lambda_4 - \lambda_2\lambda_5)$

$T_{1,3} = 2 * (\lambda_3\lambda_5 + \lambda_2\lambda_4)$

$T_{2,1} = 2 * (\lambda_3\lambda_4 + \lambda_2\lambda_5)$

$T_{2,2} = \lambda_2^2 - \lambda_3^2 + \lambda_4^2 - \lambda_5^2$

$T_{2,3} = 2 * (\lambda_4\lambda_5 - \lambda_2\lambda_3)$

$T_{3,1} = 2 * (\lambda_3\lambda_5 - \lambda_2\lambda_4)$

$T_{3,2} = 2 * (\lambda_4\lambda_5 + \lambda_2\lambda_3)$

$T_{3,3} = \lambda_2^2 - \lambda_3^2 - \lambda_4^2 + \lambda_5^2$

CALL MULT (I.E. $[I,B] = [I,B_0]\,[T]$)

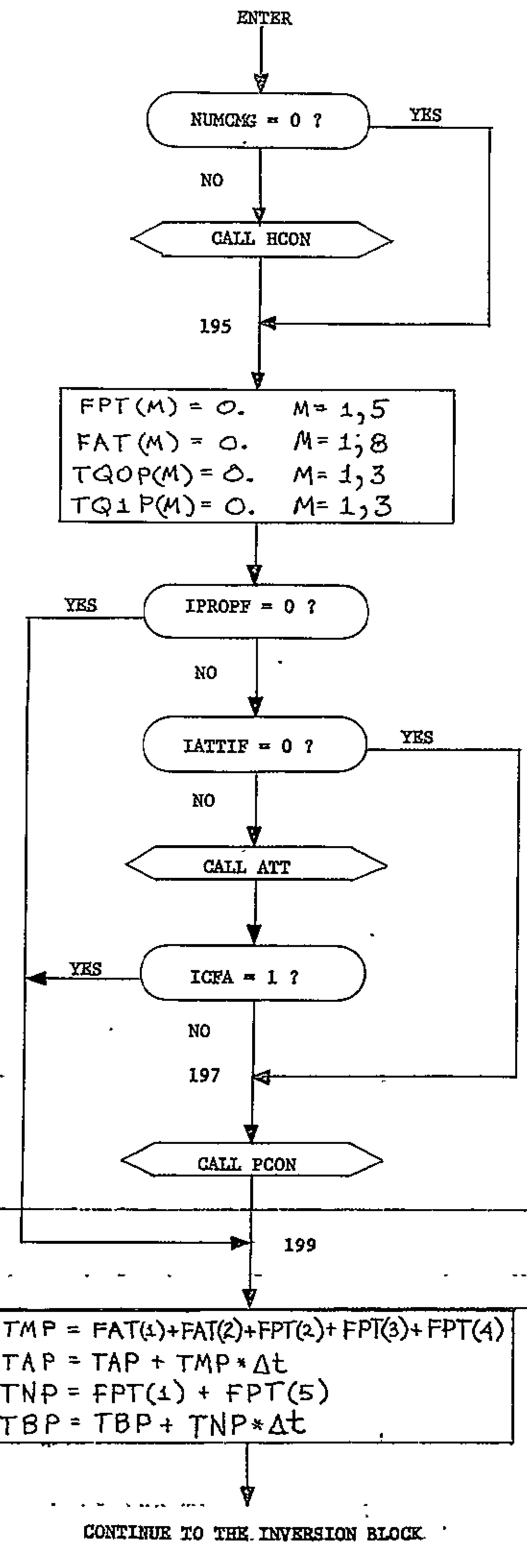
CALL MULT (I.E. $[H_I] = [I,B]\,[H]$)

CONTINUE TO THE CONTROL MODULE

FLOW CHART & BLOCK DIAGRAM

Application CONTROL MODULE Date OCTOBER 1970 Page 1 of 1

Procedure Drawn By GARY JOHNSON

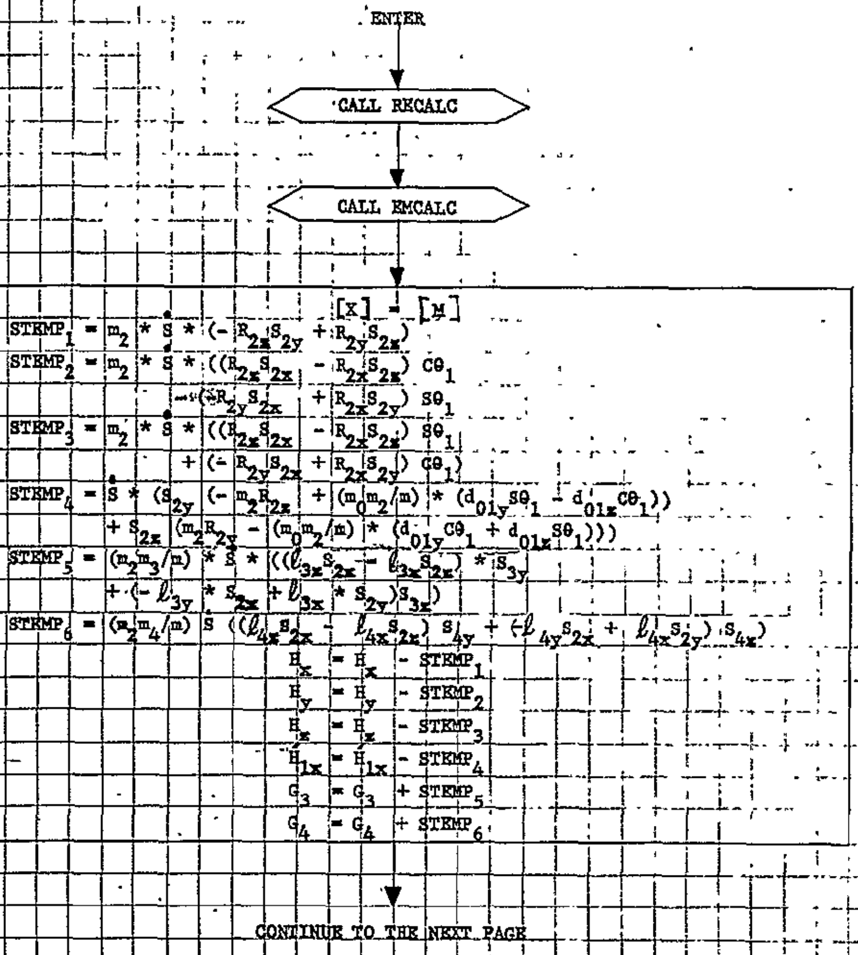


FLOW CHART & BLOCK DIAGRAM

FORM DEN 1103-01 (4-64)

Application INVERSION BLOCK Date OCTOBER 1970 Page 1 of 2

Procedure Drawn By GARY JOHNSON



$[X] = [M]$

$STEMP_1 = m_2 * \dot{S} * (-R_{2z}S_{2y} + R_{2y}S_{2z})$

$STEMP_2 = m_2 * \dot{S} * ((R_{2z}S_{2x} - R_{2x}S_{2z})\ C\theta_1$
$\quad - (-R_{2y}S_{2x} + R_{2x}S_{2y})\ S\theta_1)$

$STEMP_3 = m_2 * \dot{S} * ((R_{2z}S_{2x} - R_{2x}S_{2z})\ S\theta_1$
$\quad + (-R_{2y}S_{2x} + R_{2x}S_{2y})\ C\theta_1)$

$STEMP_4 = \dot{S} * (S_{2y}\ (-m_2R_{2z} + (m_0m_2/m) * (d_{01y}S\theta_1 - d_{01z}C\theta_1))$
$\quad + S_{2z}\ (m_2R_{2y} - (m_0m_2/m) * (d_{01y}C\theta_1 + d_{01z}S\theta_1)))$

$STEMP_5 = (m_2m_3/m) * \dot{S} * ((\ell_{3z}S_{2x} - \ell_{3x}S_{2z}) * S_{3y}$
$\quad + (-\ell_{3y} * S_{2z} + \ell_{3x} * S_{2y})S_{3z})$

$STEMP_6 = (m_2m_4/m)\ \dot{S}\ ((\ell_{4z}S_{2x} - \ell_{4x}S_{2z})\ S_{4y} + (-\ell_{4y}S_{2x} + \ell_{4x}S_{2y})\ S_{4z})$

$H_x = H_x - STEMP_1$

$H_y = H_y - STEMP_2$

$H_z = H_z - STEMP_3$

$H_{1x} = H_{1x} - STEMP_4$

$G_3 = G_3 + STEMP_5$

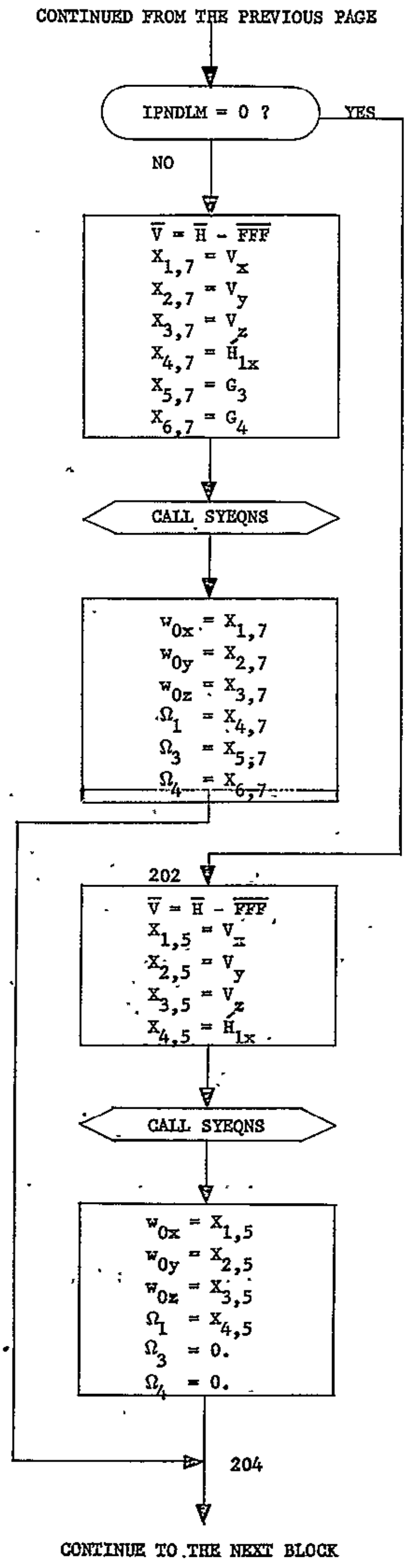
$G_4 = G_4 + STEMP_6$

CONTINUE TO THE NEXT PAGE

FLOW CHART & BLOCK DIAGRAM

Application INVERSION BLOCK CONTINUED Date OCTOBER 1970 Page 2 of 2

Procedure ____ Drawn By GARY JOHNSON

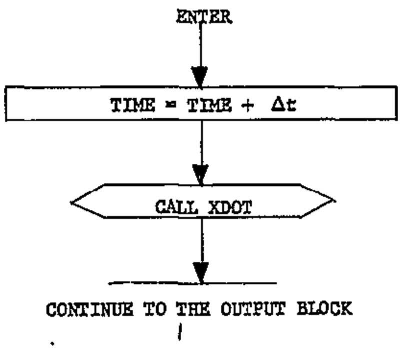


FLOW CHART & BLOCK DIAGRAM

Form DEN 1103-01 (4-64)

Application TIME UPDATE Date OCTOBER 1970 Page 1 of 1

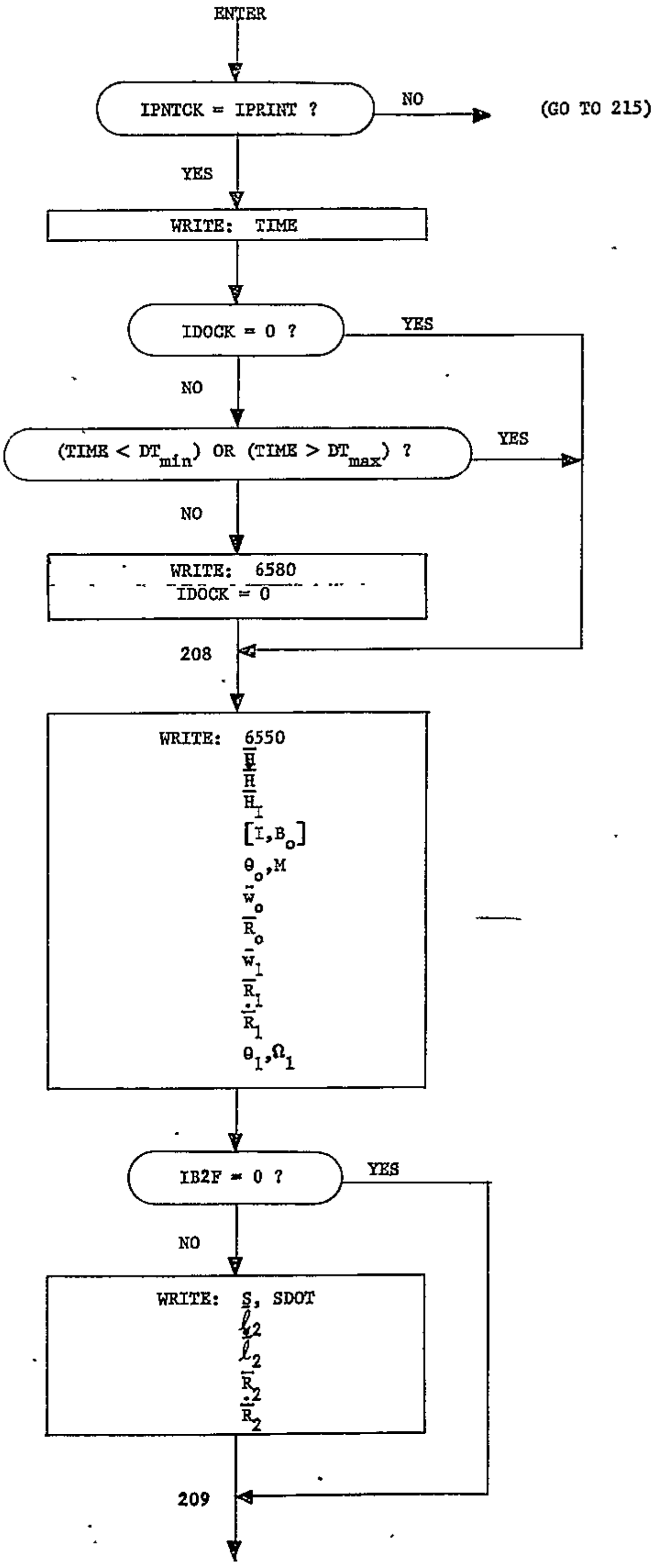
Procedure Drawn By GARY JOHNSON



FLOW CHART & BLOCK DIAGRAM

Application OUTPUT BLOCK Date OCTOBER 1970 Page 1 of 3

Procedure Drawn By GARY JOHNSON



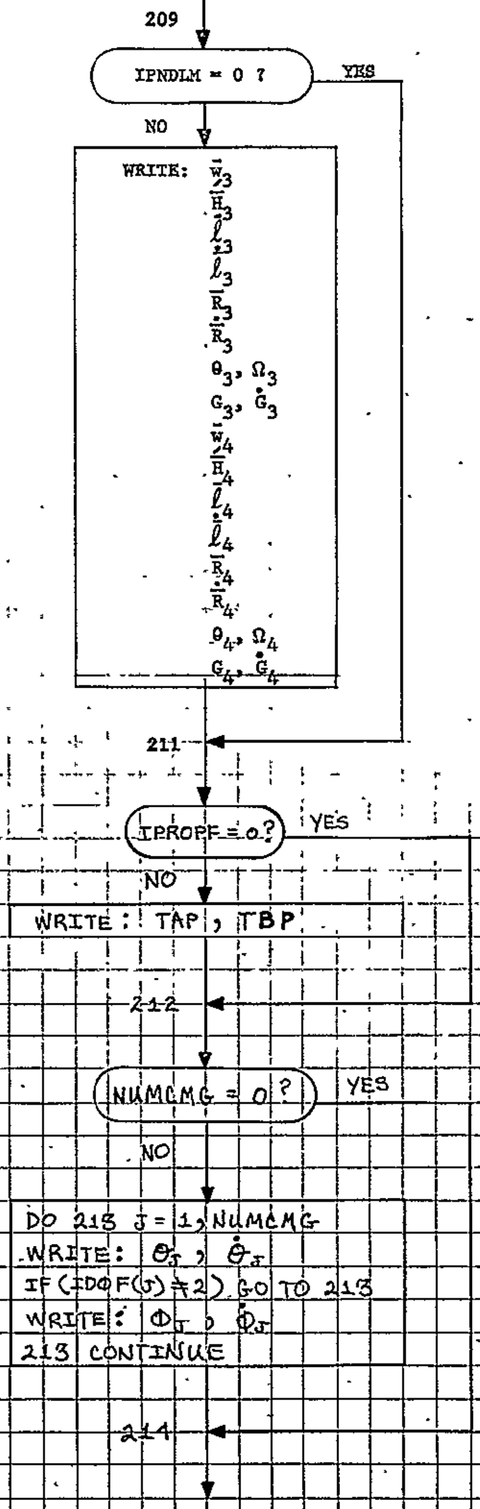
FLOW CHART & BLOCK DIAGRAM

FORM DEN 1103-01 (4-64)

Application OUTPUT BLOCK CONTINUED Date OCTOBER 1970 Page 2 of [illegible]

Procedure Drawn By GARY JOHNSON

CONTINUED FROM THE PREVIOUS PAGE



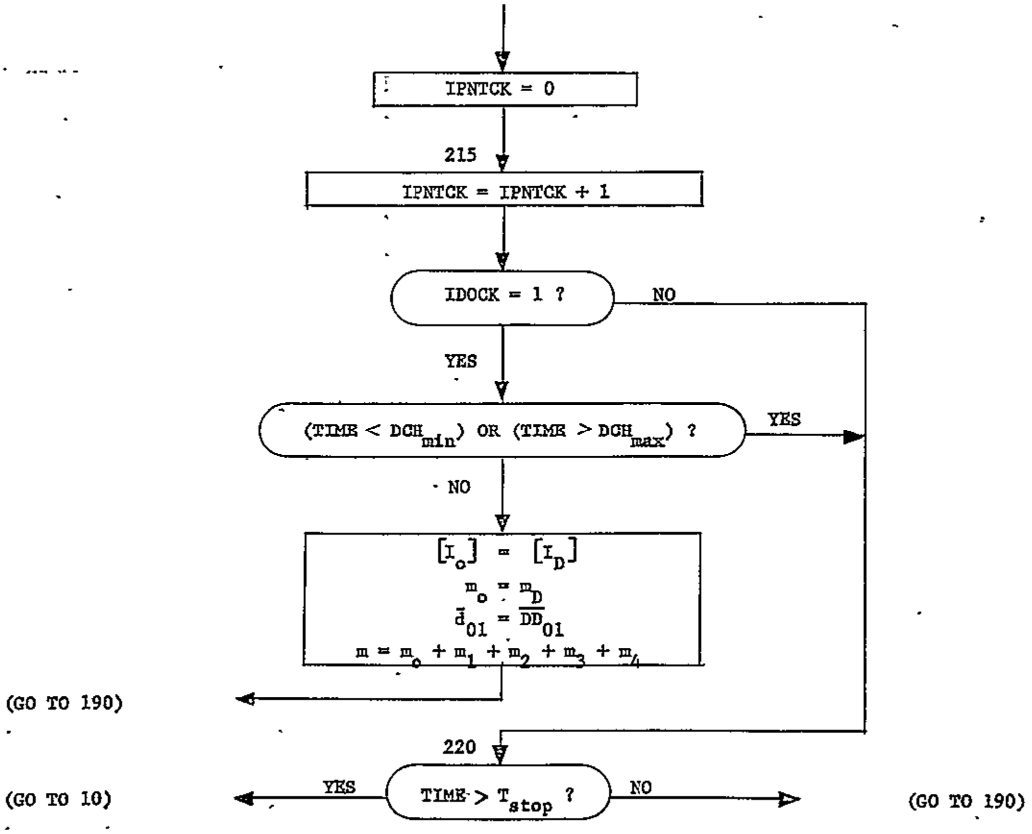
CONTINUE TO THE FOLLOWING PAGE

FLOW CHART & BLOCK DIAGRAM

Application OUTPUT BLOCK CONTINUED Date OCTOBER 1970 Page 3 of 3

Procedure Drawn By GARY JOHNSON

CONTINUED FROM THE PREVIOUS PAGE

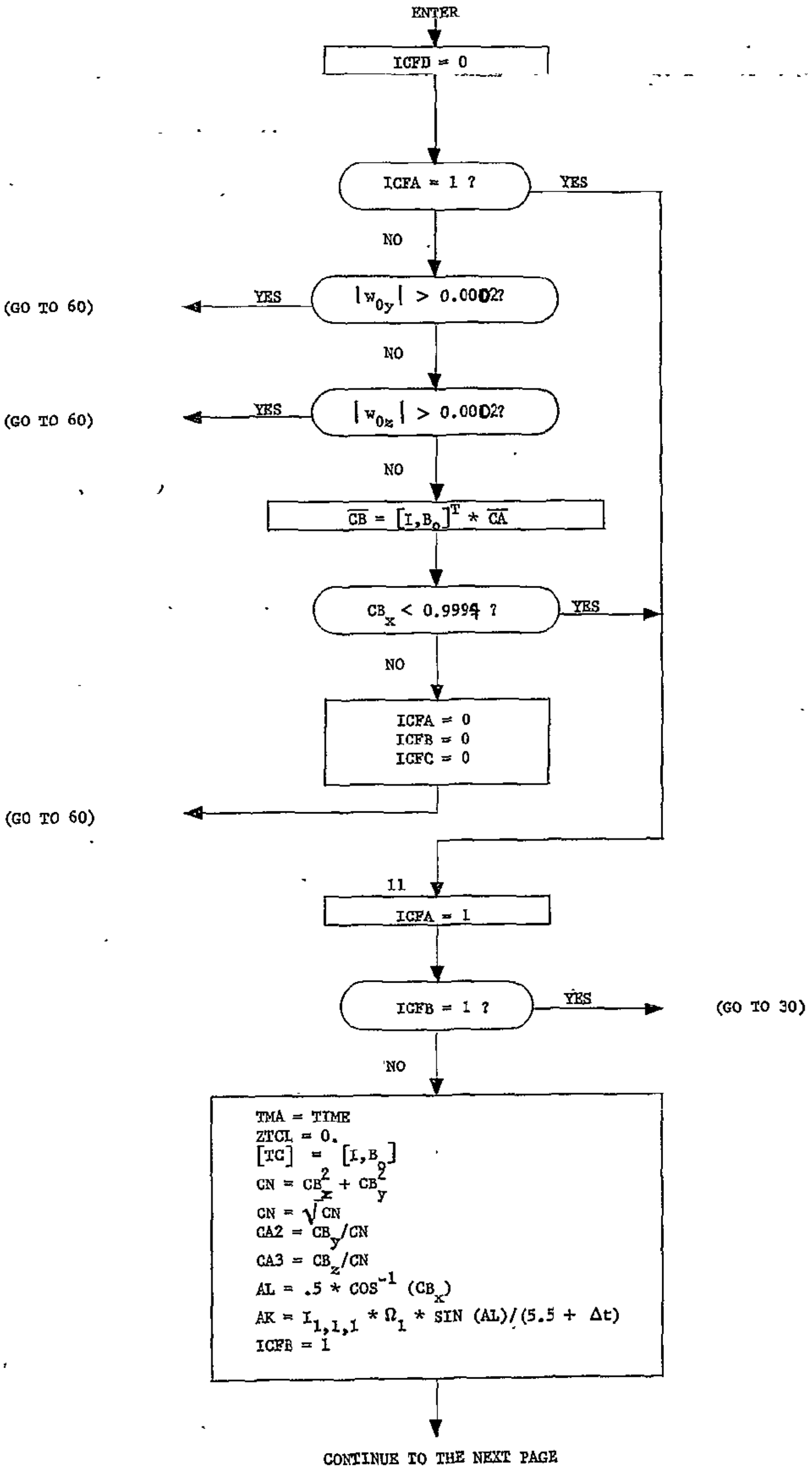


FLOW CHART & BLOCK DIAGRAM

FORM DEV 1102-01 (4-64)

Application SUBROUTINE ATT Date OCTOBER 1970 Page 1 of 3

Procedure Drawn By GARY JOHNSON

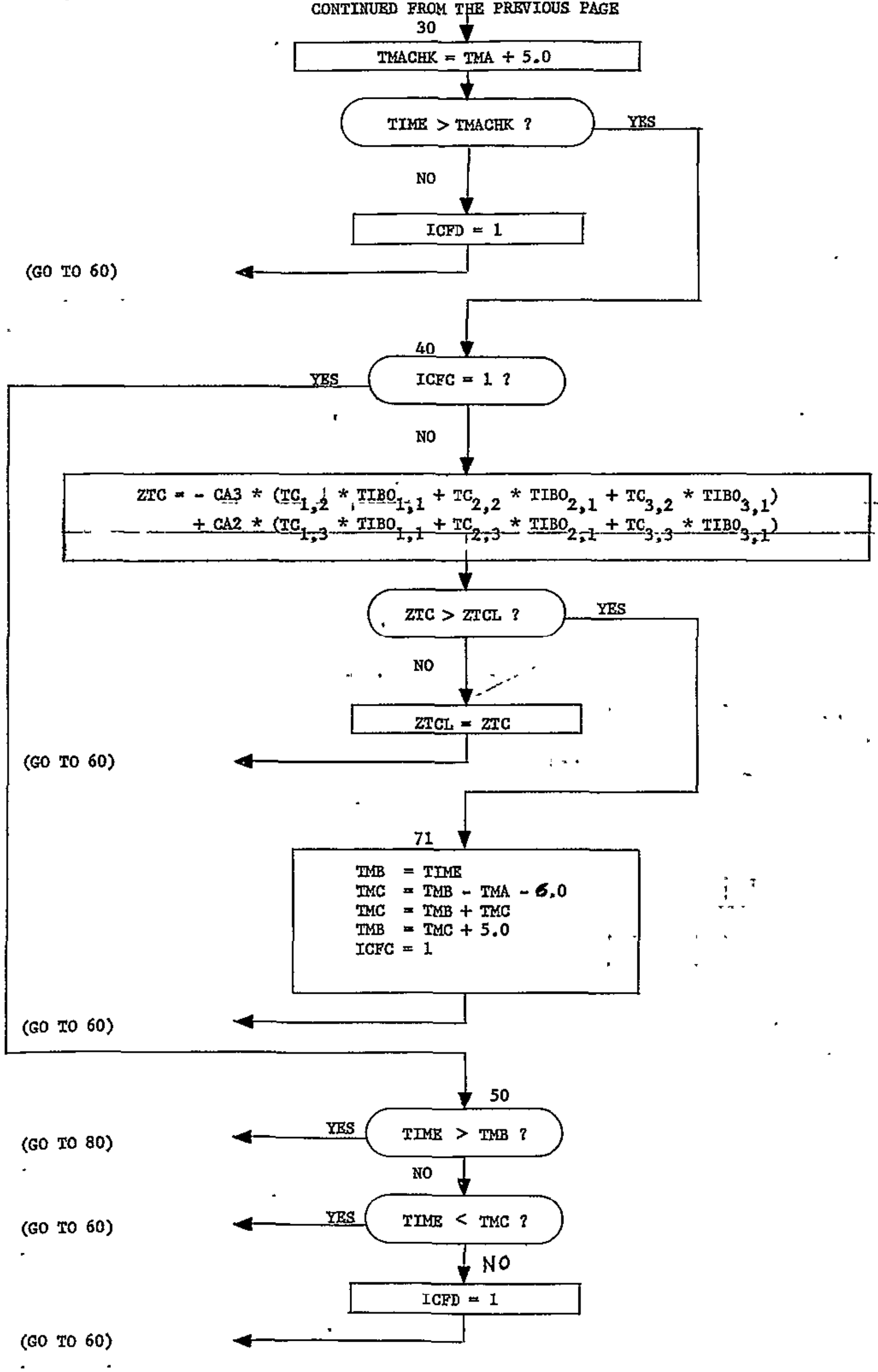


FLOW CHART & BLOCK DIAGRAM

Application SUBROUTINE ATT CONTINUED Date OCTOBER 1970 Page 2 of 3

Procedure Drawn By GARY JOHNSON

CONTINUED FROM THE PREVIOUS PAGE



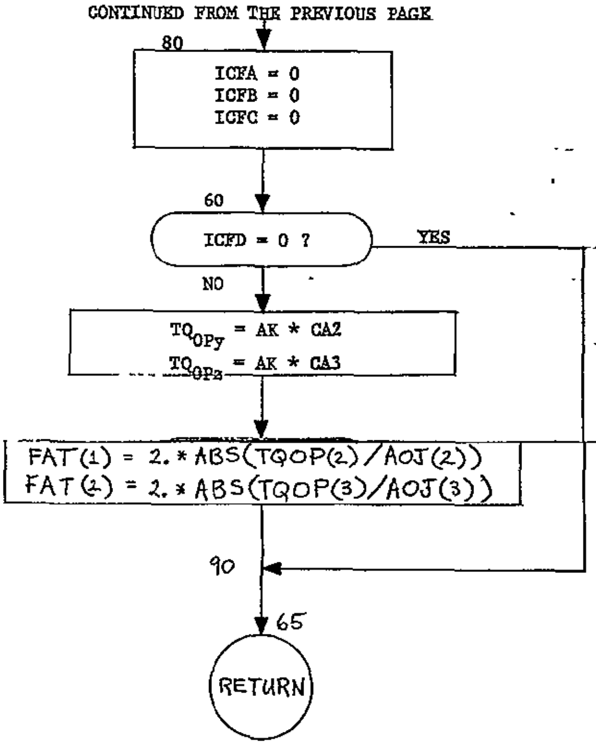
CONTINUE TO THE NEXT PAGE

FLOW CHART & BLOCK DIAGRAM

Form DEN 1103-01 (4-64)

Application SUBROUTINE ATT CONTINUED Date OCTOBER 1970 Page 3 of 3

Procedure ________ Drawn By GARY JOHNSON


CONTINUED FROM THE PREVIOUS PAGE
80
ICFA = 0
ICFB = 0
ICFC = 0
60
ICFD = 0 ?
YES
NO
TQOPy = AK * CA2
TQOPz = AK * CA3
FAT(1) = 2.*ABS(TQOP(2)/AOJ(2))
FAT(2) = 2.*ABS(TQOP(3)/AOJ(3))
90
65
RETURN

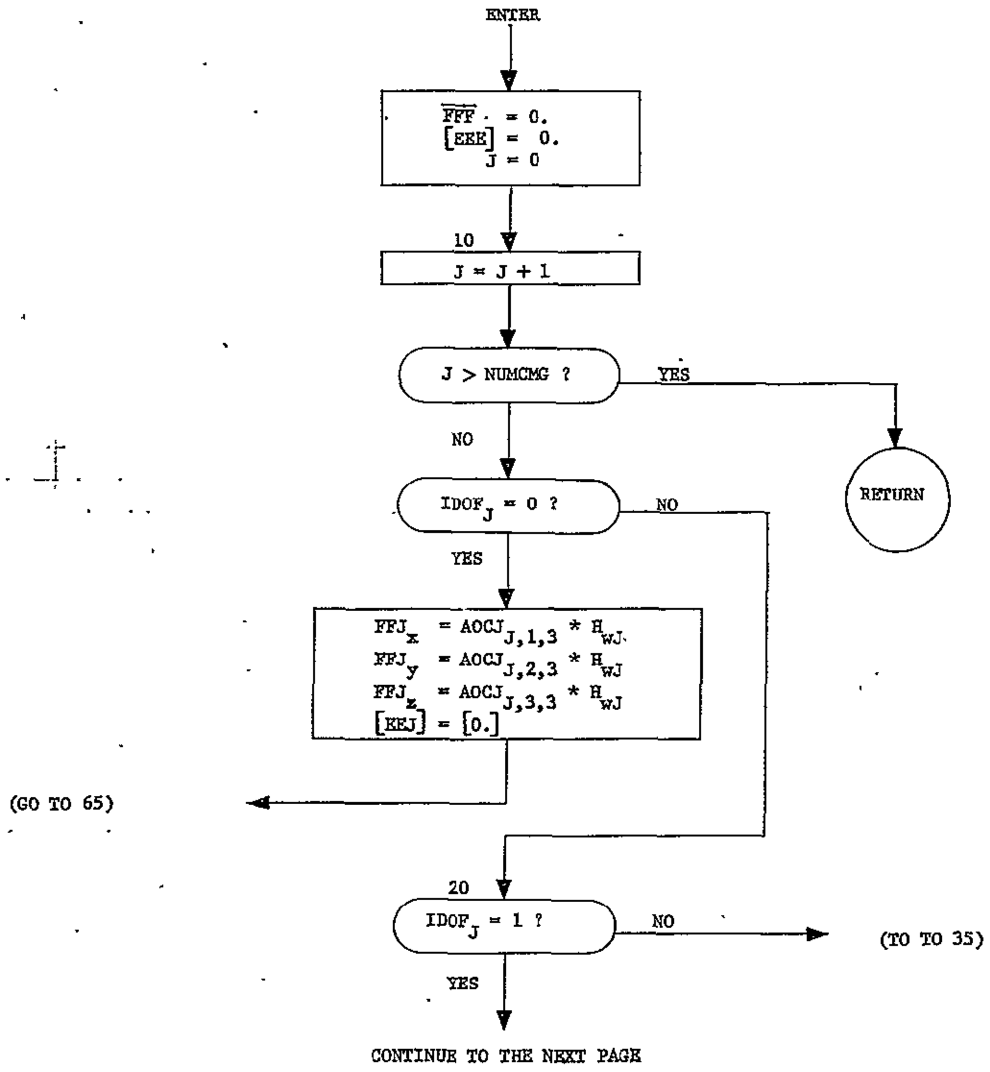
Fold to dotted line

FLOW CHART & BLOCK DIAGRAM

Form DEN 1103-01 (4-64)

Application SUBROUTINE CMG Date OCTOBER 1970 Page 1 of 4

Procedure ____ Drawn By GARY JOHNSON



Application SUBROUTINE CMG CONTINUED Date OCTOBER 1970 Page 2 of 4

Procedure Drawn By GARY JOHNSON

CONTINUED FROM THE PREVIOUS PAGE

$\text{TEMP12}_x = \text{C}\theta_J * \text{AII}_{J,1,2} * \dot{\theta}_J + \text{S}\theta_J * \text{AII}_{J,3,2} * \dot{\theta}_J + \text{S}\theta_J \text{H}_{wJ}$

$\text{TEMP12}_y = \text{SØ}_J * \text{S}\theta_J * \text{AII}_{J,1,2} * \dot{\theta}_J + \text{CØ}_J * \text{AII}_{J,2,2} * \dot{\theta}_J - \text{SØ}_J * \text{C}\theta_J * \text{AII}_{J,3,2} * \dot{\theta}_J$

$\text{TEMP12}_z = - \text{S}\theta_J * \text{CØ}_J * \text{AII}_{J,1,2} * \theta_J + \text{SØ}_J * \text{AII}_{J,2,2} * \theta_J$

$+ \text{CØ}_J * \text{C}\theta_J * \text{AII}_{J,3,2} * \dot{\theta}_J + \text{H}_{wJ} * \text{C}\theta_J$

$\overline{\text{FFJ}} = [0,\text{C}_J] * \overline{\text{TEMP12}}$

$\text{TEMP13}_{1,1} = \text{C}\theta_J * \text{AOCJ}_{J,1,1} + \text{SØ}_J * \text{S}\theta_J * \text{AOCJ}_{J,1,2} - \text{S}\theta_J * \text{CØ}_J * \text{AOCJ}_{J,1,3}$

$\text{TEMP13}_{1,2} = \text{C}\theta_J * \text{AOCJ}_{J,2,1} + \text{SØ}_J * \text{S}\theta_J * \text{AOCJ}_{J,2,2} - \text{S}\theta_J * \text{CØ}_J * \text{AOCJ}_{J,2,3}$

$\text{TEMP13}_{1,3} = \text{C}\theta_J * \text{AOCJ}_{J,3,1} + \text{SØ}_J * \text{S}\theta_J * \text{AOCJ}_{J,3,2} - \text{S}\theta_J * \text{CØ}_J * \text{AOCJ}_{J,3,3}$

$\text{TEMP13}_{2,1} = \text{CØ}_J * \text{AOCJ}_{J,1,2} + \text{SØ}_J * \text{AOCJ}_{J,1,3}$

$\text{TEMP13}_{2,2} = \text{CØ}_J * \text{AOCJ}_{J,2,2} + \text{SØ}_J * \text{AOCJ}_{J,2,3}$

$\text{TEMP13}_{2,3} = \text{CØ}_J * \text{AOCJ}_{J,3,2} + \text{SØ}_J * \text{AOCJ}_{J,3,3}$

$\text{TEMP13}_{3,1} = \text{S}\theta_J * \text{AOCJ}_{J,1,1} - \text{SØ}_J * \text{C}\theta_J * \text{AOCJ}_{J,1,2} + \text{CØ}_J * \text{C}\theta_J * \text{AOCJ}_{J,1,3}$

$\text{TEMP13}_{3,2} = \text{S}\theta_J * \text{AOCJ}_{J,2,1} - \text{SØ}_J * \text{C}\theta_J * \text{AOCJ}_{J,2,2} + \text{CØ}_J * \text{C}\theta_J \;\&\; \text{AOCJ}_{J,2,3}$

$\text{TEMP13}_{3,3} = \text{S}\theta_J * \text{AOCJ}_{J,3,1} - \text{SØ}_J * \text{C}\theta_J * \text{AOCJ}_{J,3,2} + \text{CØ}_J * \text{C}\theta_J * \text{AOCJ}_{J,3,3}$

$[\text{TEMP14}] = [\text{II}_J] * [\text{TEMP13}]$

$\text{TEMP15}_{1,1} = \text{C}\theta_J * \text{TEMP14}_{1,1} + \text{S}\theta_J * \text{TEMP14}_{3,1}$

$\text{TEMP15}_{1,2} = \text{C}\theta_J * \text{TEMP14}_{1,2} + \text{S}\theta_J * \text{TEMP14}_{3,2}$

$\text{TEMP15}_{1,3} = \text{C}\theta_J * \text{TEMP14}_{1,3} + \text{S}\theta_J * \text{TEMP14}_{3,3}$

$\text{TEMP15}_{2,1} = \text{SØ}_J * \text{S}\theta_J * \text{TEMP14}_{1,1} + \text{CØ}_J * \text{TEMP14}_{2,1} - \text{SØ}_J * \text{C}\theta_J * \text{TEMP14}_{3,1}$

$\text{TEMP15}_{2,2} = \text{SØ}_J * \text{S}\theta_J * \text{TEMP14}_{1,2} + \text{CØ}_J * \text{TEMP14}_{2,2} - \text{SØ}_J * \text{C}\theta_J * \text{TEMP14}_{3,2}$

$\text{TEMP15}_{2,3} = \text{SØ}_J * \text{S}\theta_J * \text{TEMP14}_{1,3} + \text{CØ}_J * \text{TEMP14}_{2,3} - \text{SØ}_J * \text{C}\theta_J * \text{TEMP14}_{3,3}$

$\text{TEMP15}_{3,1} = - \text{S}\theta_J * \text{CØ}_J * \text{TEMP14}_{1,1} + \text{SØ}_J * \text{TEMP14}_{2,1} + \text{CØ}_J * \text{C}\theta_J * \text{TEMP14}_{3,1}$

$\text{TEMP15}_{3,2} = - \text{S}\theta_J * \text{CØ}_J * \text{TEMP14}_{1,2} + \text{SØ}_J * \text{TEMP14}_{2,2} + \text{CØ}_J * \text{C}\theta_J * \text{TEMP14}_{3,2}$

$\text{TEMP15}_{3,3} = - \text{S}\theta_J * \text{CØ}_J * \text{TEMP14}_{1,3} + \text{SØ}_J * \text{TEMP14}_{2,3} + \text{CØ}_J * \text{C}\theta_J * \text{TEMP14}_{3,3}$

$[\text{EEJ}] = [0,\text{C}_J] * [\text{TEMP15}]$

(GO TO 60)

CONTINUE TO THE NEXT PAGE

FLOW CHART & BLOCK DIAGRAM

FORM DES 1103-01 (4-64)

Application SUBROUTINE CMG CONTINUED Date OCTOBER 1970 Page 3 of 4

Procedure Drawn By GARY JOHNSON

CONTINUED FROM THE PREVIOUS PAGE

35

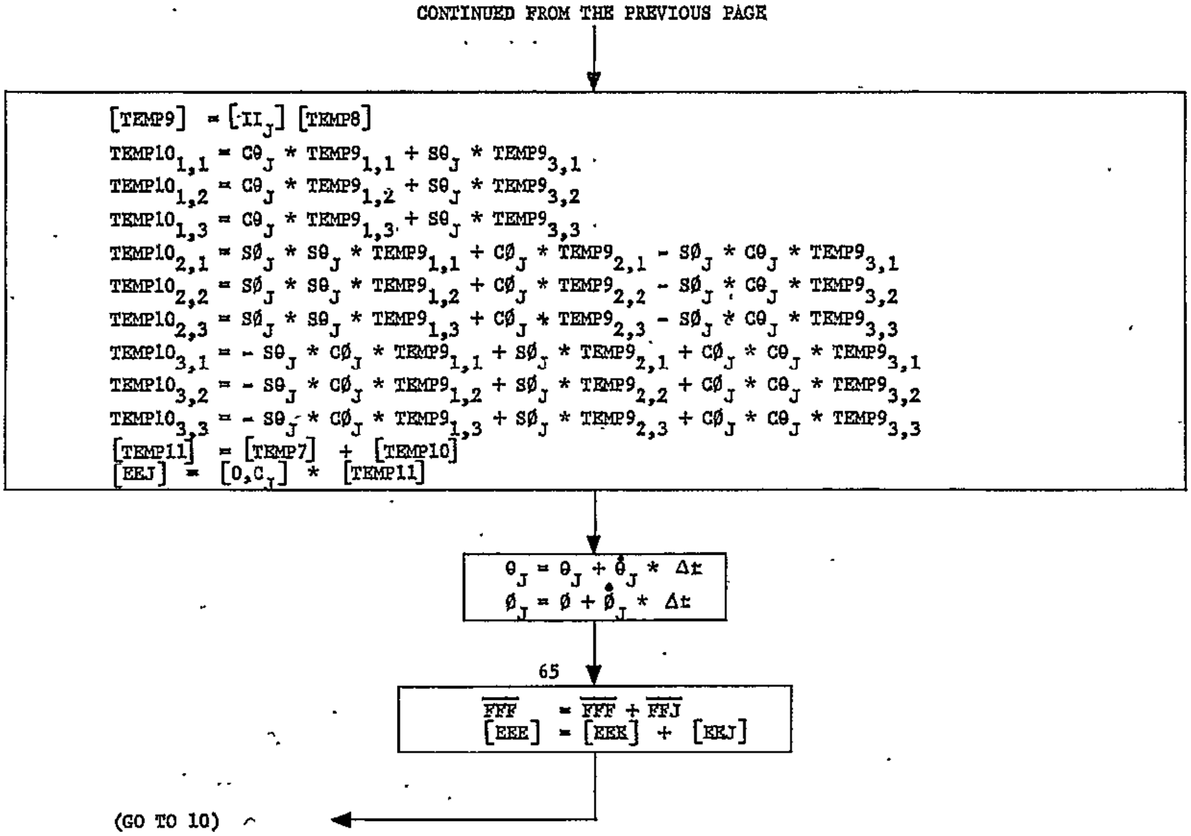
$$
\begin{aligned}
TEMP1_x &= AIO_{J,1,1} * \dot{\phi}_J \\
TEMP1_y &= C\phi_J * AIO_{J,2,1} * \dot{\phi}_J - S\phi_J * AIO_{J,3,1} * \dot{\phi}_J \\
TEMP1_z &= S\phi_J * AIO_{J,2,1} * \dot{\phi}_J + C\phi_J * AIO_{J,3,1} * \dot{\phi}_J \\
TEMP2_x &= C\theta_J * [AII_{J,1,1} * C\theta_J * \dot{\phi}_J + AII_{J,1,2} * \dot{\theta}_J + AII_{J,1,3} * S\theta_J * \dot{\phi}_J] \\
&\quad + S\theta_J * [AII_{J,3,1} * C\theta_J * \dot{\phi}_J + AII_{J,3,2} * \dot{\theta}_J + AII_{J,3,3} * S\theta_J * \dot{\phi}_J] \\
TEMP2_y &= S\phi_J * S\theta_J * [AII_{J,1,1} * C\theta_J * \dot{\phi}_J + AII_{J,1,2} * \dot{\theta}_J + AII_{J,1,3} * S\theta_J * \dot{\phi}_J] \\
&\quad + C\phi_J * [AII_{J,2,1} * C\theta_J * \dot{\phi}_J + AII_{J,2,2} * \dot{\theta}_J + AII_{J,2,3} * S\theta_J * \dot{\phi}_J] \\
&\quad - S\phi_J * C\theta_J * [AII_{J,3,1} * C\theta_J * \dot{\phi}_J + AII_{J,3,2} * \dot{\theta}_J + AII_{J,3,3} * S\theta_J * \dot{\phi}_J] \\
TEMP2_z &= - S\theta_J * C\phi_J * [AII_{J,1,1} * C\theta_J * \dot{\phi}_J + AII_{J,1,2} * \dot{\theta}_J + AII_{J,1,3} * S\theta_J * \dot{\phi}_J] \\
&\quad + S\phi_J * [AII_{J,2,1} * C\theta_J * \dot{\phi}_J + AII_{J,2,2} * \dot{\theta}_J + AII_{J,2,3} * S\theta_J * \dot{\phi}_J] \\
&\quad + C\phi_J * C\theta_J * [AII_{J,3,1} * C\theta_J * \dot{\phi}_J + AII_{J,3,2} * \dot{\theta}_J + AII_{J,3,3} * S\theta_J * \dot{\phi}_J] \\
TEMP3_x &= H_{wJ} * S\theta_J \\
TEMP3_y &= - H_{wJ} * C\theta_J * S\phi_J \\
TEMP3_z &= H_{wJ} * C\theta_J * C\phi_J \\
\overline{TEMP4} &= \overline{TEMP1} + \overline{TEMP2} + \overline{TEMP3} \\
\overline{FFJ} &= [0, C_J] * \overline{TEMP4} \\
TEMP5_{1,1} &= AOCJ_{J,1,1} \\
TEMP5_{1,2} &= AOCJ_{J,2,1} \\
TEMP5_{1,3} &= AOCJ_{J,3,1} \\
TEMP5_{2,1} &= C\phi_J * AOCJ_{J,1,2} + S\phi_J * AOCJ_{J,1,3} \\
TEMP5_{2,2} &= C\phi_J * AOCJ_{J,2,2} + S\phi_J * AOCJ_{J,2,3} \\
TEMP5_{2,3} &= C\phi_J * AOCJ_{J,3,2} + S\phi_J * AOCJ_{J,3,3} \\
TEMP5_{3,1} &= - S\phi_J * AOCJ_{J,1,2} + C\phi_J * AOCJ_{J,1,3} \\
TEMP5_{3,2} &= - S\phi_J * AOCJ_{J,2,2} + C\phi_J * AOCJ_{J,2,3} \\
TEMP5_{3,3} &= - S\phi_J * AOCJ_{J,3,2} + C\phi_J * AOCJ_{J,3,3} \\
\overline{TEMP6} &= [IO_J] * \overline{TEMP5} \\
TEMP7_{1,1} &= TEMP6_{1,1} \\
TEMP7_{1,2} &= TEMP6_{1,2} \\
TEMP7_{1,3} &= TEMP6_{1,3} \\
TEMP7_{2,1} &= C\phi_J * TEMP6_{2,1} - S\phi_J * TEMP6_{3,1} \\
TEMP7_{2,2} &= C\phi_J * TEMP6_{2,2} - S\phi_J * TEMP6_{3,2} \\
TEMP7_{2,3} &= C\phi_J * TEMP6_{2,3} - S\phi_J * TEMP6_{3,3} \\
TEMP7_{3,1} &= S\phi_J * TEMP6_{2,1} + C\phi_J * TEMP6_{3,1} \\
TEMP7_{3,2} &= S\phi_J * TEMP6_{2,2} + C\phi_J * TEMP6_{3,2} \\
TEMP7_{3,3} &= S\phi_J * TEMP6_{2,3} + C\phi_J * TEMP6_{3,3} \\
TEMP8_{1,1} &= C\theta_J * AOCJ_{J,1,1} + S\phi_J * S\theta_J * AOCJ_{J,1,2} - S\theta_J * C\phi_J * AOCJ_{J,1,3} \\
TEMP8_{1,2} &= C\theta_J * AOCJ_{J,2,1} + S\phi_J * S\theta_J * AOCJ_{J,2,2} - S\theta_J * C\phi_J * AOCJ_{J,2,3} \\
TEMP8_{1,3} &= C\theta_J * AOCJ_{J,3,1} + S\phi_J * S\theta_J * AOCJ_{J,3,2} - S\theta_J * C\phi_J * AOCJ_{J,3,3} \\
TEMP8_{2,1} &= C\phi_J * AOCJ_{J,1,2} + S\phi_J * AOCJ_{J,1,3} \\
TEMP8_{2,2} &= C\phi_J * AOCJ_{J,2,2} + S\phi_J * AOCJ_{J,2,3} \\
TEMP8_{2,3} &= C\phi_J * AOCJ_{J,3,2} + S\phi_J * AOCJ_{J,3,3} \\
TEMP8_{3,1} &= S\theta_J * AOCJ_{J,1,1} - S\phi_J * C\theta_J * AOCJ_{J,1,2} + C\phi_J * C\theta_J * AOCJ_{J,1,3} \\
TEMP8_{3,2} &= S\theta_J * AOCJ_{J,2,1} - S\phi_J * C\theta_J * AOCJ_{J,2,2} + C\phi_J * C\theta_J * AOCJ_{J,2,3} \\
TEMP8_{3,3} &= S\theta_J * AOCJ_{J,3,1} - S\phi_J * C\theta_J * AOCJ_{J,3,2} + C\phi_J * C\theta_J * AOCJ_{J,3,3}
\end{aligned}
$$

CONTINUE TO THE NEXT PAGE

FLOW CHART & BLOCK DIAGRAM

Application SUBROUTINE CMG CONTINUED Date OCTOBER 1970 Page 4 of 4

Procedure Drawn By GARY JOHNSON



CONTINUED FROM THE PREVIOUS PAGE

$[TEMP9] = [II_J]\ [TEMP8]$

$TEMP10_{1,1} = C\theta_J * TEMP9_{1,1} + S\theta_J * TEMP9_{3,1}$

$TEMP10_{1,2} = C\theta_J * TEMP9_{1,2} + S\theta_J * TEMP9_{3,2}$

$TEMP10_{1,3} = C\theta_J * TEMP9_{1,3} + S\theta_J * TEMP9_{3,3}$

$TEMP10_{2,1} = S\emptyset_J * S\theta_J * TEMP9_{1,1} + C\emptyset_J * TEMP9_{2,1} - S\emptyset_J * C\theta_J * TEMP9_{3,1}$

$TEMP10_{2,2} = S\emptyset_J * S\theta_J * TEMP9_{1,2} + C\emptyset_J * TEMP9_{2,2} - S\emptyset_J * C\theta_J * TEMP9_{3,2}$

$TEMP10_{2,3} = S\emptyset_J * S\theta_J * TEMP9_{1,3} + C\emptyset_J * TEMP9_{2,3} - S\emptyset_J * C\theta_J * TEMP9_{3,3}$

$TEMP10_{3,1} = - S\theta_J * C\emptyset_J * TEMP9_{1,1} + S\emptyset_J * TEMP9_{2,1} + C\emptyset_J * C\theta_J * TEMP9_{3,1}$

$TEMP10_{3,2} = - S\theta_J * C\emptyset_J * TEMP9_{1,2} + S\emptyset_J * TEMP9_{2,2} + C\emptyset_J * C\theta_J * TEMP9_{3,2}$

$TEMP10_{3,3} = - S\theta_J * C\emptyset_J * TEMP9_{1,3} + S\emptyset_J * TEMP9_{2,3} + C\emptyset_J * C\theta_J * TEMP9_{3,3}$

$[TEMP11] = [TEMP7] + [TEMP10]$

$[EEJ] = [0, C_J] * [TEMP11]$

$\theta_J = \theta_J + \dot{\theta}_J * \Delta t$

$\emptyset_J = \emptyset + \dot{\emptyset}_J * \Delta t$

65

$\overline{FFF} = \overline{FFF} + \overline{FFJ}$

$[EEE] = [EEK] + [EEJ]$

(GO TO 10)

FLOW CHART & BLOCK DIAGRAM

Application SUBROUTINE EMCALC Date OCTOBER 1970 Page 1 of 4

Procedure ____ Drawn By GARY JOHNSON

ENTER

$$
\begin{aligned}
M_{1,1} = {} & I_{0,1,1} + I_{1,1,1} + m_1 * d_{01y} * (R_{1y}C\theta_1 - R_{1z}S\theta_1) + m_1 * d_{01z} * (R_{1y}S\theta_1 + R_{1z}C\theta_1) \\
& + m_2 * (R_{2y}C\theta_1 - R_{2z}S\theta_1) * (d_{01y} + \ell_{2y}C\theta_1 - \ell_{2z}S\theta_1) \\
& + m_2 * (R_{2y}S\theta_1 + R_{2z}C\theta_1) * (d_{01z} + \ell_{2y}S\theta_1 + \ell_{2z}C\theta_1) \\
& + m_3 * (R_{3y}C\theta_1 - R_{3z}S\theta_1) * (d_{01y} + d_{13y}C\theta_1 - d_{13z}S\theta_1 + \ell_{3y}C\theta_1 - \ell_{3z}S\theta_1) \\
& + m_3 * (R_{3y}S\theta_1 + R_{3z}C\theta_1) * (d_{01z} + d_{13y}S\theta_1 + d_{13z}C\theta_1 + \ell_{3y}S\theta_1 + \ell_{3z}C\theta_1) \\
& + m_4 * (R_{4y}C\theta_1 - R_{4z}S\theta_1) * (d_{01y} + d_{14y}C\theta_1 - d_{14z}S\theta_1 + \ell_{4y}C\theta_1 - \ell_{4z}S\theta_1) \\
& + m_4 * (R_{4y}S\theta_1 + R_{4z}C\theta_1) * (d_{01z} + d_{14y}S\theta_1 + d_{14z}C\theta_1 + \ell_{4y}S\theta_1 + \ell_{4z}C\theta_1)
\end{aligned}
$$

$$
\begin{aligned}
M_{1,2} = {} & I_{0,1,2} + I_{1,1,2} * C\theta_1 - I_{1,1,3} * S\theta_1 - m_1 * R_{1x} * d_{01y} \\
& - m_2 * R_{2x} * (d_{01y} + \ell_{2y}C\theta_1 - \ell_{2z}S\theta_1) \\
& - m_3 * R_{3x} * (d_{01y} + d_{13y}C\theta_1 - d_{13z}S\theta_1 + \ell_{3y}C\theta_1 - \ell_{3z}S\theta_1) \\
& - m_4 * R_{4x} * (d_{01y} + d_{14y}C\theta_1 - d_{14z}S\theta_1 + \ell_{4y}C\theta_1 - \ell_{4z}S\theta_1)
\end{aligned}
$$

$$
\begin{aligned}
M_{1,3} = {} & I_{0,1,3} + I_{1,1,2} * S\theta_1 + I_{1,1,3} * C\theta_1 - m_1 * R_{1x} * d_{01z} \\
& - m_2 * R_{2x} * (d_{01z} + \ell_{2y}S\theta_1 + \ell_{2z}C\theta_1) \\
& - m_3 * R_{3x} * (d_{01z} + d_{13y}S\theta_1 + d_{13z}C\theta_1 + \ell_{3y}S\theta_1 + \ell_{3z}C\theta_1) \\
& - m_4 * R_{4x} * (d_{01z} + d_{14y}S\theta_1 + d_{14z}C\theta_1 + \ell_{4y}S\theta_1 + \ell_{4z}C\theta_1)
\end{aligned}
$$

$$
\begin{aligned}
M_{1,4} = {} & I_{1,1,1} - m_1 * d_{01z} * ((m_o/m) * d_{01z} - R_{1y}S\theta_1 - R_{1z}C\theta_1) \\
& + m_1 * d_{01y} * (R_{1y}C\theta_1 - R_{1z}S\theta_1 - (m_o/m) * d_{01y}) \\
& - m_2 * (d_{01z} + \ell_{2y}S\theta_1 + \ell_{2z}C\theta_1) * ((m_o/m) * d_{01z} - R_{2y}S\theta_1 - R_{2z}C\theta_1) \\
& + m_2 * (d_{01y} + \ell_{2y}C\theta_1 - \ell_{2z}S\theta_1) * (R_{2y}C\theta_1 - R_{2z}S\theta_1 - (m_o/m) * d_{01y}) \\
& - m_3 * (d_{01z} + (d_{13y} + \ell_{3y})S\theta_1 + (d_{13z} + \ell_{3z}) * C\theta_1) * ((m_o/m) * d_{01z} - R_{3y}S\theta_1 - R_{3z}C\theta_1) \\
& + m_3 * (d_{01y} + (d_{13y} + \ell_{3y})C\theta_1 - (d_{13z} + \ell_{3z}) * S\theta_1) * (R_{3y}C\theta_1 - R_{3z}S\theta_1 - (m_o/m) * d_{01y}) \\
& - m_4 * (d_{01z} + (d_{14y} + \ell_{4y})S\theta_1 + (d_{14z} + \ell_{4z}) * C\theta_1) * ((m_o/m) * d_{01z} - R_{4y}S\theta_1 - R_{4z}C\theta_1) \\
& + m_4 * (d_{01y} + (d_{14y} + \ell_{4y})C\theta_1 - (d_{14z} + \ell_{4z}) * S\theta_1) * (R_{4y}C\theta_1 - R_{4z}S\theta_1 - (m_o/m) * d_{01y})
\end{aligned}
$$

$$
\begin{aligned}
M_{1,5} = {} & - m_3 * R_{3x} * (S_{3z} * \ell_{3x} - S_{3x} * \ell_{3z}) \\
& + m_3 * R_{3y} * (- S_{3y} * \ell_{3x} + S_{3x} * \ell_{3y})
\end{aligned}
$$

$$
\begin{aligned}
M_{1,6} = {} & - m_4 * R_{4z} * (S_{4z} * \ell_{4x} - S_{4x} * \ell_{4z}) \\
& + m_4 * R_{4y} * (- S_{4y} * \ell_{4x} + S_{4x} * \ell_{4y})
\end{aligned}
$$

$$
\begin{aligned}
M_{2,1} = {} & I_{0,2,1} + I_{1,2,1} * C\theta_1 - I_{1,3,1} * S\theta_1 - m_1 * d_{01x} * (R_{1y}C\theta_1 - R_{1z}S\theta_1) \\
& - m_2 * (R_{2y}C\theta_1 - R_{2z}S\theta_1) * (d_{01x} + \ell_{2x}) \\
& - m_3 * (R_{3y}C\theta_1 - R_{3z}S\theta_1) * (d_{01x} + d_{13x} + \ell_{3x}) \\
& - m_4 * (R_{4y}C\theta_1 - R_{4z}S\theta_1) * (d_{01x} + d_{14x} + \ell_{4x})
\end{aligned}
$$

CONTINUE TO THE NEXT PAGE

FLOW CHART & BLOCK DIAGRAM

Application SUBROUTINE EMCALC CONTINUED Date: OCTOBER 1970 Page 2 of 4

Procedure Drawn By GARY JOHNSON

CONTINUED FROM THE PREVIOUS PAGE

$$\begin{aligned}
M_{2,2} = {} & I_{0,2,2} + I_{1,2,2}C\theta_1^2 - I_{1,2,3}C\theta_1 S\theta_1 - I_{1,3,2} * C\theta_1 S\theta_1 + I_{1,3,3} * S\theta_1^2 \\
& + m_1 * R_{1x} * d_{01x} + m_1 * d_{01z} * (R_{1y}S\theta_1 + R_{1z}C\theta_1) \\
& + m_2 * R_{2x} * (d_{01x} + \ell_{2x}) + m_2 * (R_{2y}S\theta_1 + R_{2z}C\theta_1) * (d_{01z} + \ell_{2y}S\theta_1 + \ell_{2z}C\theta_1) \\
& + m_3 * R_{3x} (d_{01x} + d_{13x} + \ell_{3x}) \\
& + m_3 * (R_{3y}S\theta_1 + R_{3z}C\theta_1) * (d_{01z} + d_{13y}S\theta_1 + d_{13z}C\theta_1 + \ell_{3y}S\theta_1 + \ell_{3z}C\theta_1) \\
& + m_4 * R_{4y} * (d_{01x} + d_{14x} + \ell_{4x}) \\
& + m_4 * (R_{4y}S\theta_1 + R_{4z}C\theta_1) * (d_{01z} + d_{14y}S\theta_1 + d_{14z}C\theta_1 + \ell_{4y}S\theta_1 + \ell_{4x}C\theta_1)
\end{aligned}$$

$$\begin{aligned}
M_{2,3} = {} & I_{0,2,3} + I_{1,2,2}C\theta_1 S\theta_1 + I_{1,2,3} * C\theta_1^2 - I_{1,3,2} * S\theta_1^2 - I_{1,3,3} * C\theta_1 S\theta_1 \\
& - m_1 * d_{01z} * (R_{1y}C\theta_1 - R_{1z}S\theta_1) \\
& - m_2 * (R_{2y}C\theta_1 - R_{2z}S\theta_1) * (d_{01z} + \ell_{2y}S\theta_1 + \ell_{2z}C\theta_1) \\
& - m_3 * (R_{3y}C\theta_1 - R_{3z}S\theta_1) * (d_{01x} + d_{13y}S\theta_1 + d_{13z}C\theta_1 + \ell_{3y}S\theta_1 + \ell_{3z}C\theta_1) \\
& - m_4 * (R_{4y}C\theta_1 - R_{4z}S\theta_1) * (d_{01x} + d_{14y}S\theta_1 + d_{14z}C\theta_1 + \ell_{4y}S\theta_1 + \ell_{4x}C\theta_1)
\end{aligned}$$

$$\begin{aligned}
M_{2,4} = {} & I_{1,1,2} * C\theta_1 - I_{1,1,3} * S\theta_1 - m_1 * d_{01x} * R_{1y}C\theta_1 \\
& + m_1 * d_{01x} * R_{1z}S\theta_1 + (m_o/m) * m_1 * d_{01x} * d_{01y} \\
& - m_2 * (d_{01x} + \ell_{2x}) * (R_{2y}C\theta_1 - R_{2z}S\theta_1 - (m_o/m) * d_{01y}) \\
& - m_3 * (d_{01x} + d_{13x} + \ell_{3x}) * (R_{3y}C\theta_1 - R_{3z}S\theta_1 - (m_o/m) * d_{01y}) \\
& - m_4 * (d_{01x} + d_{14x} + \ell_{4x}) * (R_{4y}C\theta_1 - R_{4z}S\theta_1 - (m_o/m) * d_{01y})
\end{aligned}$$

$$\begin{aligned}
M_{2,5} = {} & (m_3 * R_{3z} * (- S_{3z} * \ell_{3y} + S_{3y} * \ell_{3z}) \\
& - m_3 * R_{3x} * (--S_{3y} * \ell_{3x} + S_{3x} * \ell_{3y})) * C\theta_1 \\
& - (- m_3 * R_{3y} * (- S_{3z} * \ell_{3y} + S_{3y} * \ell_{3z}) \\
& + m_3 * R_{3x} * (S_{3z} * \ell_{3x} - S_{3x} * \ell_{3z})) * S\theta_1
\end{aligned}$$

$$\begin{aligned}
M_{2,6} = {} & (m_4 * R_{4z} * (- S_{4z} * \ell_{4y} + S_{4y} * \ell_{4z}) \\
& - m_4 * R_{4x} * (- S_{4y} * \ell_{4x} + S_{4x} * \ell_{4y})) * C\theta_1 \\
& - (- m_4 * R_{4y} * (- S_{4z} * \ell_{4y} + S_{4y} * \ell_{4z}) \\
& + m_4 * R_{4x} * (S_{4z} * \ell_{4x} - S_{4x} * \ell_{4z})) * S\theta_1
\end{aligned}$$

$$\begin{aligned}
M_{3,1} = {} & I_{0,3,1} + I_{1,2,1} * S\theta_1 + I_{1,3,1} * C\theta_1 - m_1 * d_{01x} * (R_{1y}S\theta_1 + R_{1z}C\theta_1) \\
& - m_2 * (d_{01x} + \ell_{2x}) * (R_{2y}S\theta_1 + R_{2z}C\theta_1) \\
& - m_3 * (d_{01x} + d_{13x} + \ell_{3x}) * (R_{3y}S\theta_1 + R_{3z}C\theta_1) \\
& - m_4 * (d_{01x} + d_{14x} + \ell_{4x}) * (R_{4y}S\theta_1 + R_{4z}C\theta_1)
\end{aligned}$$

$$\begin{aligned}
M_{3,2} = {} & I_{0,3,2} + I_{1,2,2} * C\theta_1 * S\theta_1 - I_{1,2,3} * S\theta_1^2 + I_{1,3,2} * C\theta_1^2 - I_{1,3,3} * C\theta_1 * S\theta_1 \\
& - m_1 * d_{01y} * (R_{1y}S\theta_1 + R_{1z}C\theta_1) \\
& - m_2 * (R_{2y}S\theta_1 + R_{2z}C\theta_1) * (d_{01y} + \ell_{2y}C\theta_1 - \ell_{2z}S\theta_1) \\
& - m_3 * (R_{3y}S\theta_1 + R_{3x}C\theta_1) * (d_{01y} + d_{13y}C\theta_1 - d_{13z}S\theta_1 + \ell_{3y}C\theta_1 - \ell_{3z}S\theta_1) \\
& - m_4 * (R_{4y}S\theta_1 + R_{4z}C\theta_1) * (d_{01y} + d_{14y}C\theta_1 - d_{14z}S\theta_1 + \ell_{4y}C\theta_1 - \ell_{4z}S\theta_1)
\end{aligned}$$

CONTINUE TO THE NEXT PAGE

CONTINUED FROM THE PREVIOUS PAGE

$$\begin{aligned}
M_{3,3} = {} & I_{0,3,3} + I_{1,2,2} * S\theta_1^2 + I_{1,2,3} * C\theta_1 * S\theta_1 + I_{1,3,2}C\theta_1 * S\theta_1 + I_{1,3,3} * C\theta_1^2 \\
& + m_1 * R_{1x} * d_{01x} + m_1 * (R_{1y}C\theta_1 - R_{1z}S\theta_1) * d_{01y} + m_2 * R_{2x} * (d_{01x} + \ell_{2x}) \\
& + m_2 * (R_{2y}C\theta_1 - R_{2z}S\theta_1) * (d_{01y} + \ell_{2y}C\theta_1 - \ell_{2z}S\theta_1) \\
& + m_3 * R_{3x} * (d_{01x} + d_{13x} + \ell_{3x}) \\
& + m_3 * (R_{3y}C\theta_1 - R_{3z}S\theta_1) * (d_{01y} + d_{13y}C\theta_1 - d_{13z}S\theta_1 + \ell_{3y}C\theta_1 - \ell_{3z}S\theta_1) \\
& + m_4 * R_{4x} * (d_{01x} + d_{14x} + \ell_{4x}) \\
& + m_4 * (R_{4y}C\theta_1 - R_{4z}S\theta_1) * (d_{01y} + d_{14y}C\theta_1 - d_{14z}S\theta_1 + \ell_{4y}C\theta_1 - \ell_{4z}S\theta_1)
\end{aligned}$$

$$\begin{aligned}
M_{3,4} = {} & I_{1,1,2} * S\theta_1 + I_{1,1,3} * C\theta_1 \\
& + m_1 * d_{01x} * ((m_o/m) * d_{01z} - R_{1y}S\theta_1 - R_{1z}C\theta_1) \\
& + m_2 * (d_{01x} + \ell_{2x}) * ((m_o/m) * d_{01z} - R_{2y}S\theta_1 - R_{2z}C\theta_1) \\
& + m_3 * (d_{01x} + d_{13x} + \ell_{3x}) * ((m_o/m) * d_{01z} - R_{3y}S\theta_1 - R_{3z}C\theta_1) \\
& + m_4 * (d_{01x} + d_{14x} + \ell_{4x}) * ((m_o/m) * d_{01z} - R_{4y}S\theta_1 - R_{4z}C\theta_1)
\end{aligned}$$

$$\begin{aligned}
M_{3,5} = {} & (m_3 * R_{3z} * (- S_{3z} * \ell_{3y} + S_{3y} * \ell_{3z}) \\
& - m_3 * R_{3x} * (- S_{3y} * \ell_{3x} + S_{3x} * \ell_{3y})) * S\theta_1 \\
& + (- m_3 * R_{3y} * (- S_{3z} * \ell_{3y} + S_{3y} * \ell_{3z}) \\
& + m_3 * R_{3x} * (S_{3z} * \ell_{3x} - S_{3x} * \ell_{3z})) * C\theta_1
\end{aligned}$$

$$\begin{aligned}
M_{3,6} = {} & (m_4 * R_{4z} * (- S_{4z} * \ell_{4y} + S_{4y} * \ell_{4z}) \\
& - m_4 * R_{4x} * (- S_{4y} * \ell_{4x} + S_{4x} * \ell_{4y})) * S\theta_1 \\
& +(-m_4 * R_{4y} * (- S_{4z} * \ell_{4y} + S_{4y} * \ell_{4z}) \\
& + m_4 * R_{4x} * (S_{4z} * \ell_{4x} - S_{4x} * \ell_{4z})) * C\theta_1
\end{aligned}$$

DEFINE SOME REOCCURRING TERMS

$$\begin{aligned}
SR_3 &= m_3 * (d_{13y} + \ell_{3y}) \\
SR_4 &= m_3 * (d_{13z} + \ell_{3z}) \\
SR_5 &= m_4 * (d_{14y} + \ell_{4y}) \\
SR_6 &= m_4 * (d_{14z} + \ell_{4z}) \\
SR_1 &= I_{1,1,2} - m_2 * \ell_{2y} * R_{2x} - SR_3 * R_{3x} - SR_5 * R_{4x} \\
SR_2 &= I_{1,1,3} - m_2 * \ell_{2z} * R_{2x} - SR_4 * R_{3x} - SR_6 * R_{4x}
\end{aligned}$$

$$M_{4,1} = I_{1,1,1} + m_2 * (\ell_{2z} * R_{2z} + \ell_{2y} * R_{2y}) + SR_4 * R_{3z} + SR_3 * R_{3y} + SR_6 * R_{4z} + SR_5 * R_{4y}$$

$$M_{4,2} = SR_1 * C\theta_1 - SR_2 * S\theta_1$$

$$M_{4,3} = SR_1 * S\theta_1 + SR_2 * C\theta_1$$

REDEFINE SR_1 AND SR_2

$$\begin{aligned}
SR_1 &= (m_o/m) * (d_{01y} * C\theta_1 + d_{01z} * S\theta_1) \\
SR_2 &= (m_o/m) * (- d_{01y} * S\theta_1 + d_{01z} * C\theta_1)
\end{aligned}$$

CONTINUE TO THE NEXT PAGE

FLOW CHART & BLOCK DIAGRAM

Application SUBROUTINE EMCALC CONTINUED Date OCTOBER 1970 Page 4 of 4

Procedure Drawn By GARY JOHNSON

CONTINUED FROM THE PREVIOUS PAGE

$$\begin{aligned}M_{4,4} = {} & I_{1,1,1} + m_2 * (\ell_{2z} * (R_{2z} - SR_2) + \ell_{2y} * (R_{2y} - SR_1)) \\ & + SR_4 * (R_{3z} - SR_2) + SR_3 * (R_{3y} - SR_1) + SR_6 * (R_{4z} - SR_2) + SR_5 * (R_{4y} - SR_1)\end{aligned}$$

$$\begin{aligned}M_{4,5} = {} & m_3 * (((R_{3z} - SR_2) * \ell_{3z} + (R_{3y} - SR_1) * \ell_{3y}) * S_{3x} \\ & - (R_{3y} - SR_1) * \ell_{3x} * S_{3y} - (R_{3z} - SR_2) * \ell_{3x} * S_{3z})\end{aligned}$$

$$\begin{aligned}M_{4,6} = {} & m_4 * (((R_{4z} - SR_2) * \ell_{4z} + (R_{4y} - SR_1) * \ell_{4y}) * S_{4x} \\ & - (R_{4y} - SR_1) * \ell_{4x} * S_{4y} - (R_{4z} - SR_2) * \ell_{4x} * S_{4z})\end{aligned}$$

$$M_{5,1} = - m_3 * \ell_{3x} * (R_{3y} * S_{3y} + R_{3z} * S_{3z})$$

$$\begin{aligned}M_{5,2} = {} & m_3 * (((\ell_{3x} * R_{3z} + \ell_{3x} * R_{3x}) * C\theta_1 + \ell_{3z} * R_{3y}S\theta_1) * S_{3y} \\ & + (-\ell_{3y} * R_{3z}C\theta_1 - (\ell_{3y} * R_{3y} + \ell_{3x} * R_{3x}) * S\theta_1) * S_{3z})\end{aligned}$$

$$\begin{aligned}M_{5,3} = {} & m_3 * (((\ell_{3z} * R_{3z} + \ell_{3x} * R_{3x}) * S\theta_1 - \ell_{3z} * R_{3y} * C\theta_1) * S_{3y} \\ & + (- \ell_{3y} * R_{3z} * S\theta_1 + (\ell_{3y} * R_{3y} + \ell_{3x} * R_{3x}) * C\theta_1) * S_{3z})\end{aligned}$$

$$M_{5,4} = m_3 * \ell_{3x} * ((- R_{3y} + SR_1) * S_{3y} + (- R_{3z} + SR_2) * S_{3z})$$

$$\begin{aligned}M_{5,5} = {} & m_3 * (1. - m_3/TOTMAS) * (((\ell_{3z}^2 + \ell_{3x}^2) * S_{3y} \\ & - \ell_{3z} * \ell_{3y} * S_{3z}) * S_{3y} + ((\ell_{3y}^2 + \ell_{3x}^2) * S_{3z} - \ell_{3y} * \ell_{3z} * S_{3y}) * S_{3z})\end{aligned}$$

$$\begin{aligned}M_{5,6} = {} & (m_3 * m_4/m) * ((-(\ell_{3z} * \ell_{4z} + \ell_{3x} * \ell_{4x}) * S_{4y} + \ell_{3z} * \ell_{4y} * S_{4z}) * S_{3y} \\ & + (-(\ell_{3y} * \ell_{4y} + \ell_{3x} * \ell_{4x}) * S_{4z} + \ell_{3y} * \ell_{4z} * S_{4y}) * S_{3z})\end{aligned}$$

$$M_{6,1} = - m_4 * \ell_{4x} * (R_{4y} * S_{4y} + R_{4z} * S_{4z})$$

$$\begin{aligned}M_{6,2} = {} & m_4 * (((\ell_{4z} * R_{4z} + \ell_{4x} * R_{4x}) * C\theta_1 + \ell_{4z} * R_{4y}S\theta_1) * S_{4y} \\ & + (- \ell_{4y} * R_{4z} * C\theta_1 - (\ell_{4y} * R_{4y} + \ell_{4x} * R_{4x}) * S\theta_1) * S_{4z})\end{aligned}$$

$$\begin{aligned}M_{6,3} = {} & m_4 * (((\ell_{4z} * R_{4z} + \ell_{4x} * R_{4x}) * S\theta_1 - \ell_{4z} * R_{4y} * C\theta_1) * S_{4y} \\ & + (- \ell_{4y} * R_{4z} * S\theta_1 + (\ell_{4y} * R_{4y} + \ell_{4x} * R_{4x}) * C\theta_1) * S_{4z})\end{aligned}$$

$$M_{6,4} = m_4 * \ell_{4x} * ((- R_{4y} + SR_1) * S_{4y} + (- R_{4z} + SR_2) * S_{4z})$$

$$\begin{aligned}M_{6,5} = {} & (m_3 * m_4/m) * ((-(\ell_{4z} * \ell_{3z} + \ell_{4x} * \ell_{3x}) * S_{3y} + \ell_{4z} * \ell_{3y} * S_{3z}) * S_{4y} \\ & + (-(\ell_{4y} * \ell_{3y} + \ell_{4x} * \ell_{3x}) * S_{3z} + \ell_{4y} * \ell_{3z} * S_{3y}) * S_{4z})\end{aligned}$$

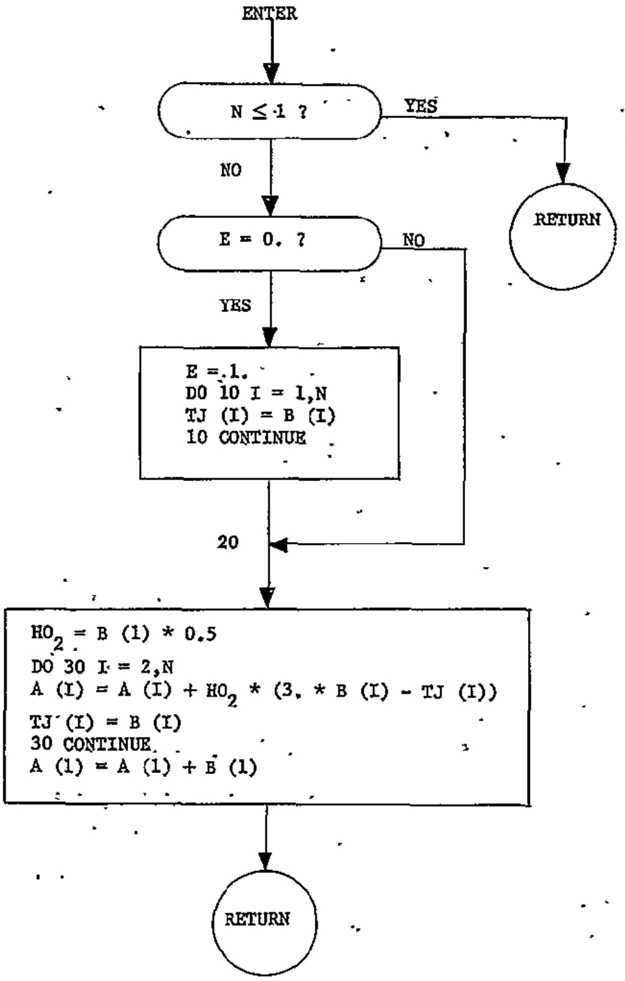
$$\begin{aligned}M_{6,6} = {} & m_4 (1. - m_4/m) * (((\ell_{4z}^2 + \ell_{4x}^2) * S_{4y} - \ell_{4z} * \ell_{4y} * S_{4z}) * S_{4y} \\ & + ((\ell_{4y}^2 + \ell_{4x}^2) * S_{4z} - \ell_{4y} * \ell_{4z} * S_{4y}) * S_{4z})\end{aligned}$$

$$[M] = [M] + [EKE]$$

RETURN

Application SUBROUTINE FOMS Date OCTOBER 1970 1 1

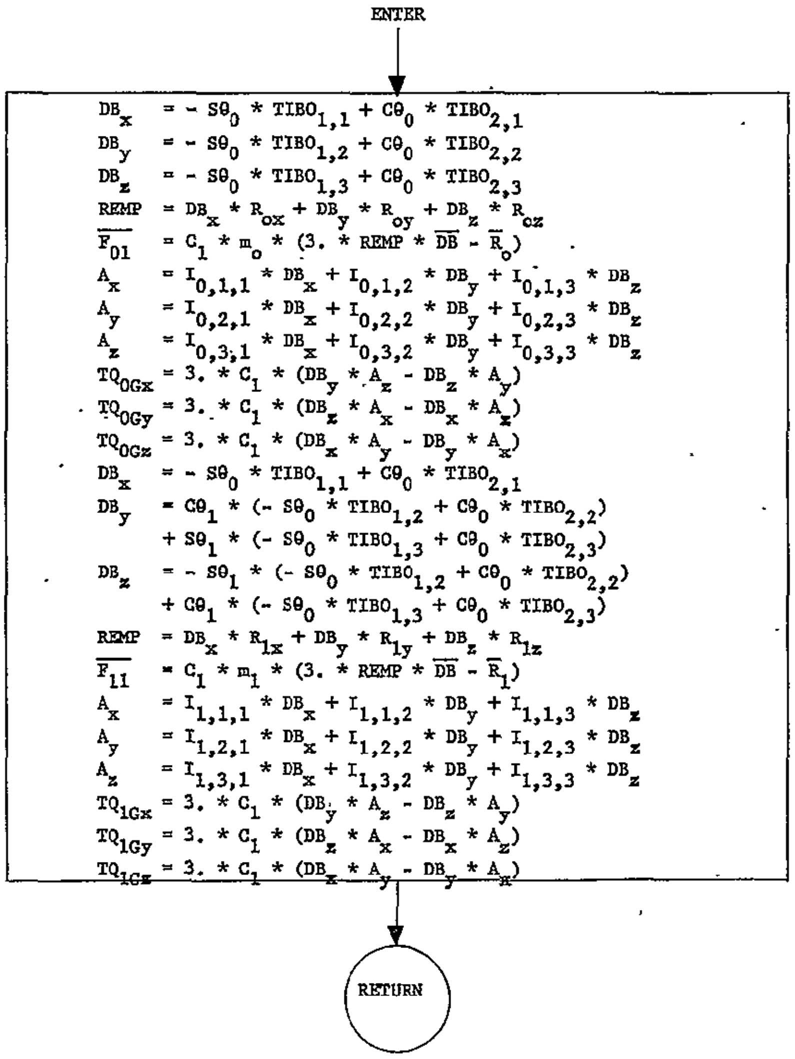
Procedure Drawn By GARY JOHNSON



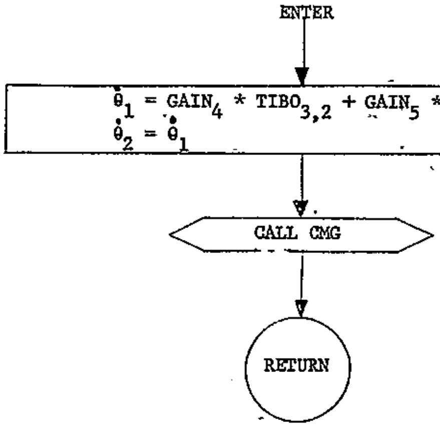
FLOW CHART & BLOCK DIAGRAM

Application SUBROUTINE GGRAD Date OCTOBER 1970 Page 1 of 1

Procedure Drawn By GARY JOHNSON



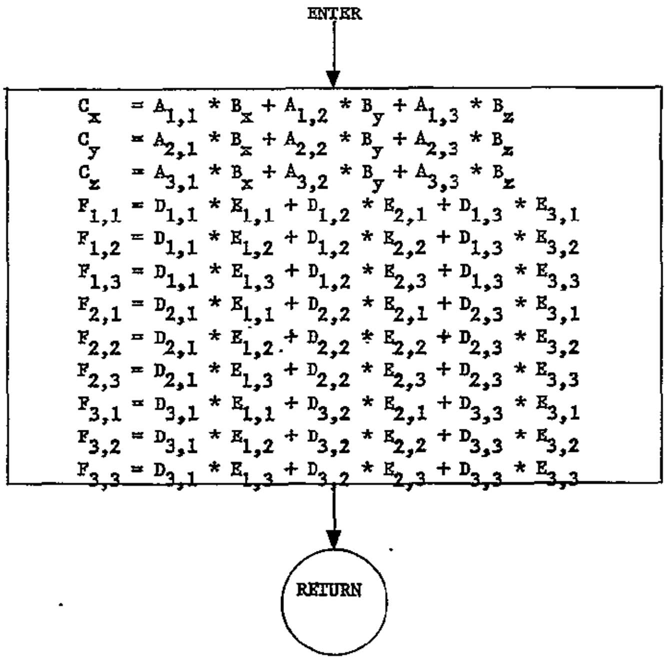
$$
\begin{aligned}
DB_x &= - S\theta_0 * TIBO_{1,1} + C\theta_0 * TIBO_{2,1} \\
DB_y &= - S\theta_0 * TIBO_{1,2} + C\theta_0 * TIBO_{2,2} \\
DB_z &= - S\theta_0 * TIBO_{1,3} + C\theta_0 * TIBO_{2,3} \\
REMP &= DB_x * R_{ox} + DB_y * R_{oy} + DB_z * R_{oz} \\
\overline{F_{01}} &= C_1 * m_o * (3. * REMP * \overline{DB} - \overline{R}_o) \\
A_x &= I_{0,1,1} * DB_x + I_{0,1,2} * DB_y + I_{0,1,3} * DB_z \\
A_y &= I_{0,2,1} * DB_x + I_{0,2,2} * DB_y + I_{0,2,3} * DB_z \\
A_z &= I_{0,3,1} * DB_x + I_{0,3,2} * DB_y + I_{0,3,3} * DB_z \\
TQ_{0Gx} &= 3. * C_1 * (DB_y * A_z - DB_z * A_y) \\
TQ_{0Gy} &= 3. * C_1 * (DB_z * A_x - DB_x * A_z) \\
TQ_{0Gz} &= 3. * C_1 * (DB_x * A_y - DB_y * A_x) \\
DB_x &= - S\theta_0 * TIBO_{1,1} + C\theta_0 * TIBO_{2,1} \\
DB_y &= C\theta_1 * (- S\theta_0 * TIBO_{1,2} + C\theta_0 * TIBO_{2,2}) \\
&\quad + S\theta_1 * (- S\theta_0 * TIBO_{1,3} + C\theta_0 * TIBO_{2,3}) \\
DB_z &= - S\theta_1 * (- S\theta_0 * TIBO_{1,2} + C\theta_0 * TIBO_{2,2}) \\
&\quad + C\theta_1 * (- S\theta_0 * TIBO_{1,3} + C\theta_0 * TIBO_{2,3}) \\
REMP &= DB_x * R_{1x} + DB_y * R_{1y} + DB_z * R_{1z} \\
\overline{F_{11}} &= C_1 * m_1 * (3. * REMP * \overline{DB} - \overline{R}_1) \\
A_x &= I_{1,1,1} * DB_x + I_{1,1,2} * DB_y + I_{1,1,3} * DB_z \\
A_y &= I_{1,2,1} * DB_x + I_{1,2,2} * DB_y + I_{1,2,3} * DB_z \\
A_z &= I_{1,3,1} * DB_x + I_{1,3,2} * DB_y + I_{1,3,3} * DB_z \\
TQ_{1Gx} &= 3. * C_1 * (DB_y * A_z - DB_z * A_y) \\
TQ_{1Gy} &= 3. * C_1 * (DB_z * A_x - DB_x * A_z) \\
TQ_{1Gz} &= 3. * C_1 * (DB_x * A_y - DB_y * A_x)
\end{aligned}
$$


ENTER
$\dot{\theta}_1 = GAIN_4 * TIBO_{3,2} + GAIN_5 * w_{0x}$
$\dot{\theta}_2 = \dot{\theta}_1$
CALL CMG
RETURN

FLOW CHART & BLOCK DIAGRAM

Application SUBROUTINE MULT Date OCTOBER 1970 Page 1 of 1

Procedure Drawn By GARY JOHNSON

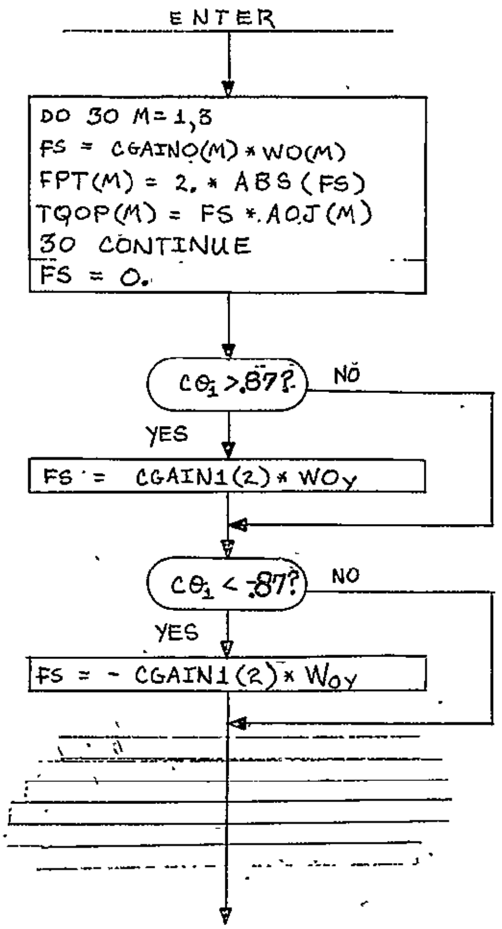


$$
\begin{aligned}
C_x &= A_{1,1} * B_x + A_{1,2} * B_y + A_{1,3} * B_z \\
C_y &= A_{2,1} * B_x + A_{2,2} * B_y + A_{2,3} * B_z \\
C_z &= A_{3,1} * B_x + A_{3,2} * B_y + A_{3,3} * B_z \\
F_{1,1} &= D_{1,1} * E_{1,1} + D_{1,2} * E_{2,1} + D_{1,3} * E_{3,1} \\
F_{1,2} &= D_{1,1} * E_{1,2} + D_{1,2} * E_{2,2} + D_{1,3} * E_{3,2} \\
F_{1,3} &= D_{1,1} * E_{1,3} + D_{1,2} * E_{2,3} + D_{1,3} * E_{3,3} \\
F_{2,1} &= D_{2,1} * E_{1,1} + D_{2,2} * E_{2,1} + D_{2,3} * E_{3,1} \\
F_{2,2} &= D_{2,1} * E_{1,2} + D_{2,2} * E_{2,2} + D_{2,3} * E_{3,2} \\
F_{2,3} &= D_{2,1} * E_{1,3} + D_{2,2} * E_{2,3} + D_{2,3} * E_{3,3} \\
F_{3,1} &= D_{3,1} * E_{1,1} + D_{3,2} * E_{2,1} + D_{3,3} * E_{3,1} \\
F_{3,2} &= D_{3,1} * E_{1,2} + D_{3,2} * E_{2,2} + D_{3,3} * E_{3,2} \\
F_{3,3} &= D_{3,1} * E_{1,3} + D_{3,2} * E_{2,3} + D_{3,3} * E_{3,3}
\end{aligned}
$$

FLOW CHART & BLOCK DIAGRAM

Ar... SUBROUTINE PCON Date OCTOBER 1970 Page 1 of 2

Procedure Drawn By GARY JOHNSON



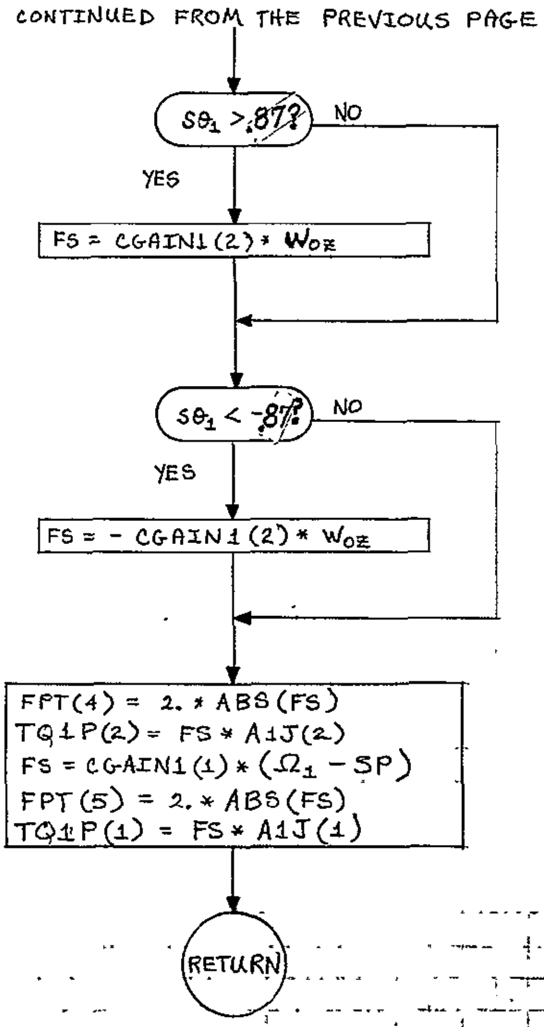
CONTINUE TO THE FOLLOWING PAGE

FLOW CHART & BLOCK DIAGRAM

FORM DEN 1103-01 (4-64)

Application SUBROUTINE PCON CONTINUED Date OCTOBER 1970 Page 2 of 2

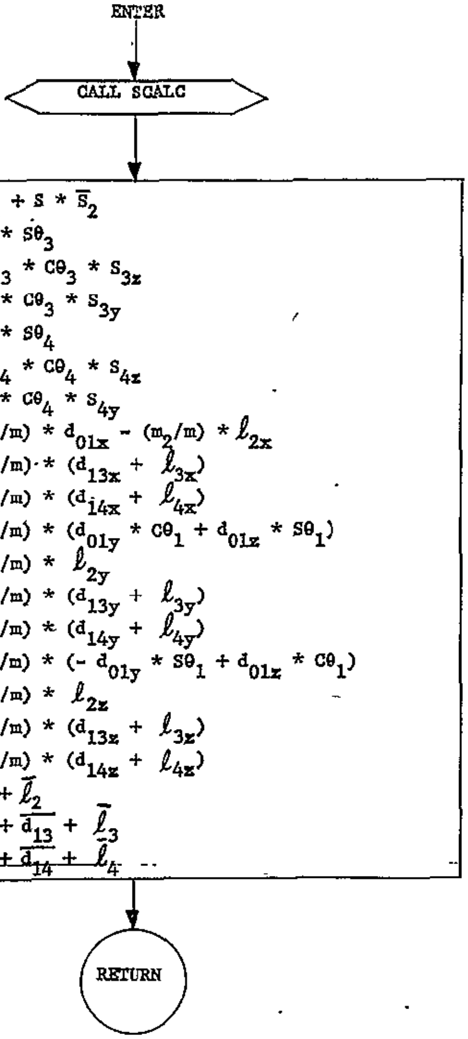
Procedure Drawn By GARY JOHNSON



FLOW CHART & BLOCK DIAGRAM

Application SUBROUTINE RECALC Date OCTOBER 1970 Page 1 of 1

Procedure Drawn By GARY JOHNSON

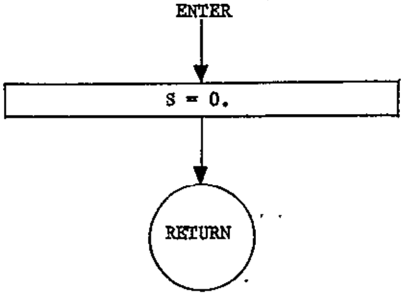


$$
\begin{aligned}
\bar{\ell}_2 &= \bar{d}_{12} + S * \bar{S}_2 \\
\ell_{3x} &= \ell_3 * S\theta_3 \\
\ell_{3y} &= -\ell_3 * C\theta_3 * S_{3z} \\
\ell_{3z} &= \ell_3 * C\theta_3 * S_{3y} \\
\ell_{4x} &= \ell_4 * S\theta_4 \\
\ell_{4y} &= -\ell_4 * C\theta_4 * S_{4z} \\
\ell_{4z} &= \ell_4 * C\theta_4 * S_{4y} \\
R_{1x} &= (m_o/m) * d_{01x} - (m_2/m) * \ell_{2x} \\
&\quad - (m_3/m) * (d_{13x} + \ell_{3x}) \\
&\quad - (m_4/m) * (d_{14x} + \ell_{4x}) \\
R_{1y} &= (m_o/m) * (d_{01y} * C\theta_1 + d_{01z} * S\theta_1) \\
&\quad - (m_2/m) * \ell_{2y} \\
&\quad - (m_3/m) * (d_{13y} + \ell_{3y}) \\
&\quad - (m_4/m) * (d_{14y} + \ell_{4y}) \\
R_{1z} &= (m_o/m) * (- d_{01y} * S\theta_1 + d_{01z} * C\theta_1) \\
&\quad - (m_2/m) * \ell_{2z} \\
&\quad - (m_3/m) * (d_{13z} + \ell_{3z}) \\
&\quad - (m_4/m) * (d_{14z} + \ell_{4z}) \\
\bar{R}_2 &= \bar{R}_1 + \bar{\ell}_2 \\
\bar{R}_3 &= \bar{R}_1 + \overline{d_{13}} + \bar{\ell}_3 \\
\bar{R}_4 &= \bar{R}_1 + \overline{d_{14}} + \bar{\ell}_4
\end{aligned}
$$

FLOW CHART & BLOCK DIAGRAM

Form DEN 1103-01 (4-64)

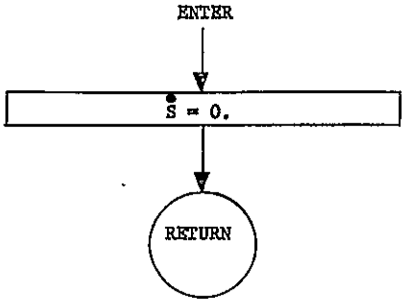
Application SUBROUTINE SCALC	Date OCTOBER 1970	Page 1 of 1
Procedure	Drawn By GARY JOHNSON	



FLOW CHART & BLOCK DIAGRAM

Application SUBROUTINE SDCALC Date OCTOBER 1970 Page 1 of 1

Procedure Drawn By GARY JOHNSON



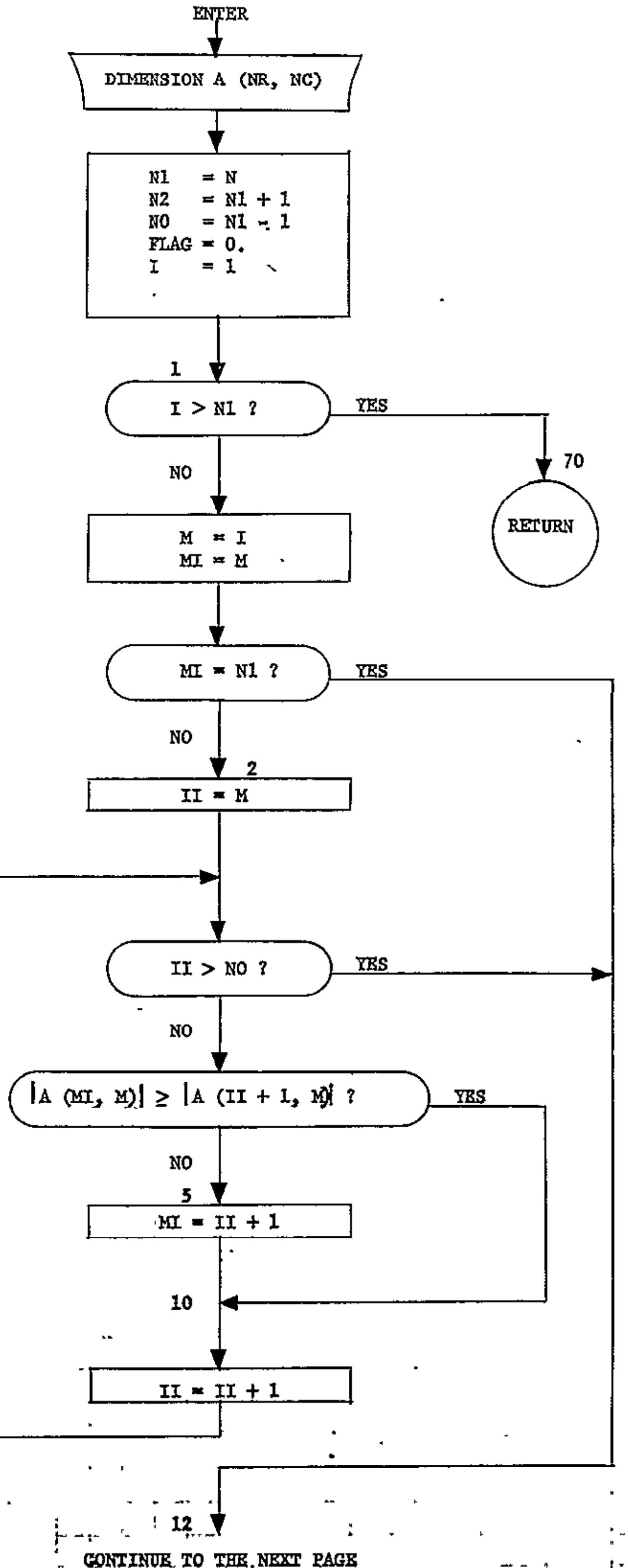
FLOW CHART & BLOCK DIAGRAM

Form DEN 1103-01 (4-64)

Application SUBROUTINE SYEQNS Date OCTOBER 1970 Page 1 of 2

Procedure Drawn By GARY JOHNSON

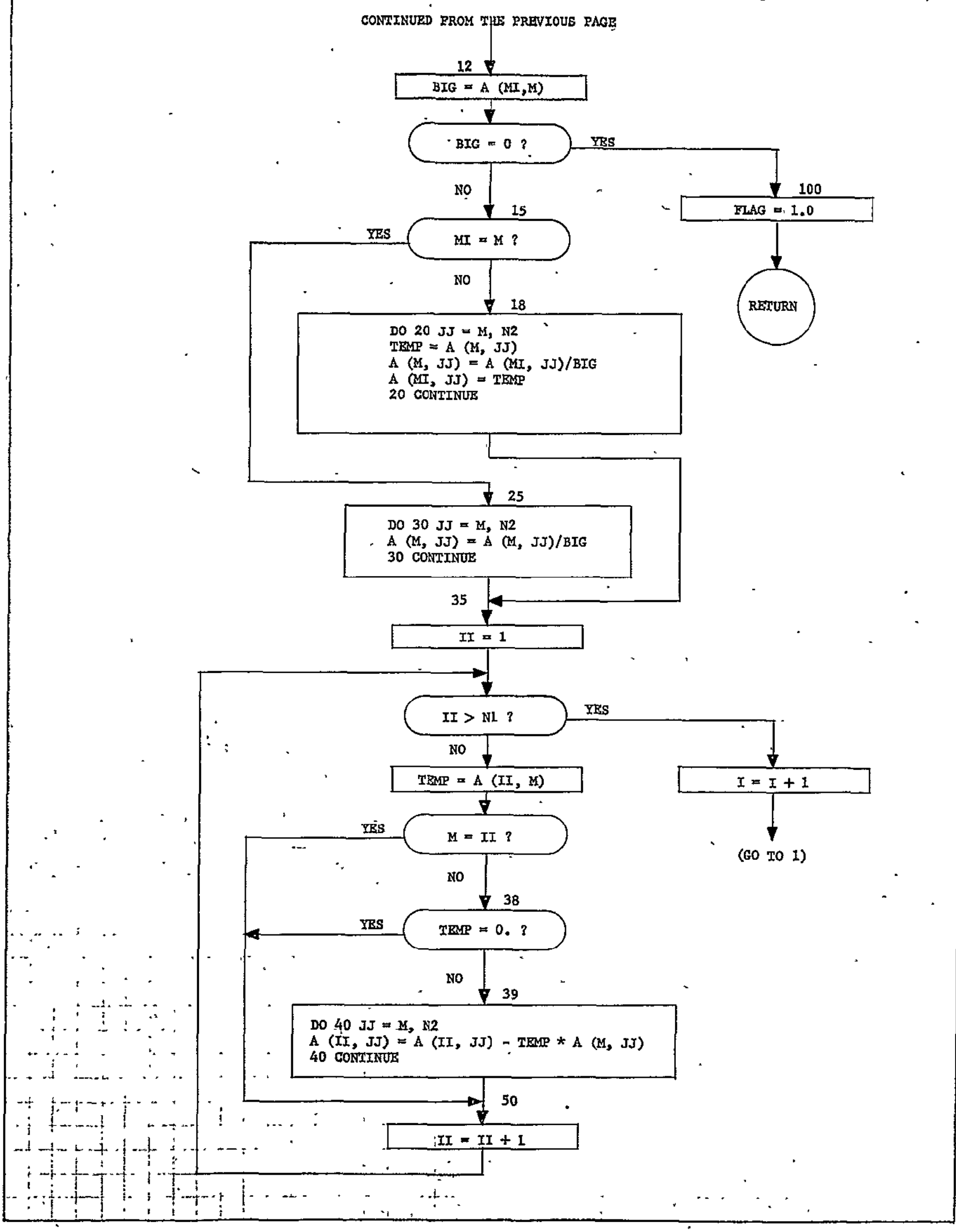
SUBROUTINE SYEQNS (A, N, NR, NC, FLAG)



FLOW CHART & BLOCK DIAGRAM

Application SUBROUTINE SYEQNS CONTINUED Date OCTOBER 1970 Page 2 of 2

Procedure Drawn By GARY JOHNSON

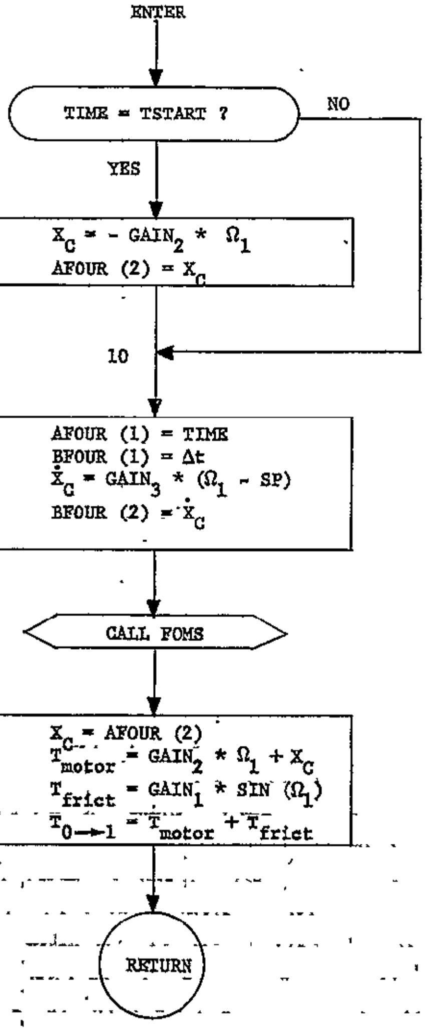


FLOW CHART & BLOCK DIAGRAM

FORM GEN 1103-01 (4-64)

Application SUBROUTINE TORK01 Date OCTOBER 1970 Page 1 of 1

Procedure Drawn By GARY JOHNSON

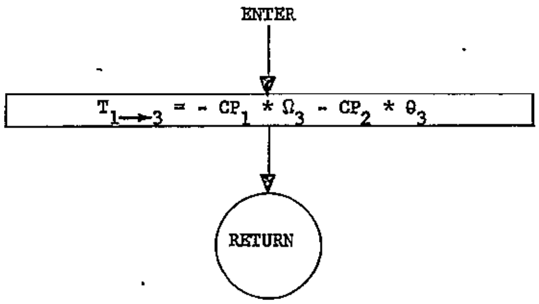

ENTER
TIME = TSTART ?
NO
YES
$X_C = - GAIN_2 * \Omega_1$
AFOUR (2) = X_C
10
AFOUR (1) = TIME
BFOUR (1) = Δt
$\dot{X}_C = GAIN_3 * (\Omega_1 - SP)$
BFOUR (2) = $\dot{X}_C$
CALL FOMS
X_C = AFOUR (2)
$T_{motor} = GAIN_2 * \Omega_1 + X_C$
$T_{frict} = GAIN_1 * SIN (\Omega_1)$
$T_{0 \rightarrow 1} = T_{motor} + T_{frict}$
RETURN

FLOW CHART & BLOCK DIAGRAM

Application SUBROUTINE TORK13 Date OCTOBER 1970 Page 1 of 1

Procedure Drawn By GARY JOHNSON

SUBROUTINE TORK13 (T13, CP1, CP2, THETA3, OMEGA3)

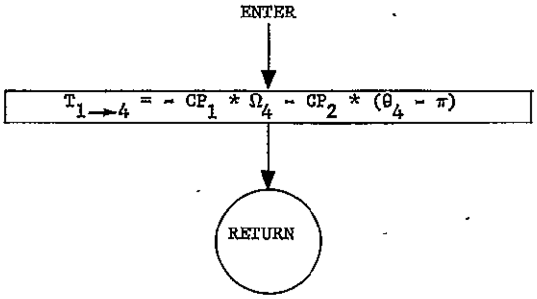

ENTER
$T_{1\longrightarrow 3} = - CP_1 * \Omega_3 - CP_2 * \theta_3$
RETURN

FLOW CHART & BLOCK DIAGRAM

Application SUBROUTINE TORK14 Date OCTOBER 1970 Page 1 of 1

Procedure ____ Drawn By GARY JOHNSON

SUBROUTINE TORK14 (T14, CP1, CP2, THETA4, OMEGA4)

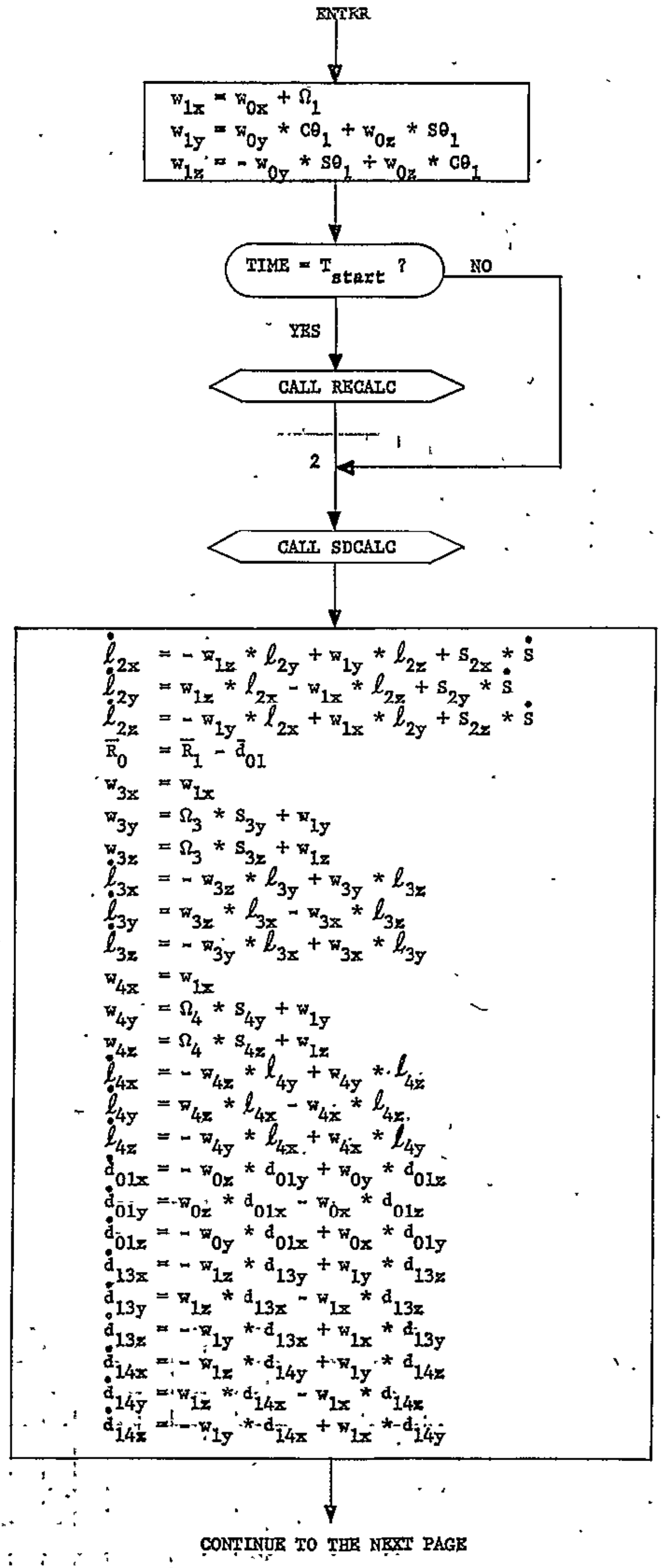

ENTER
$T_{1\rightarrow 4} = - CP_1 * \Omega_4 - CP_2 * (\theta_4 - \pi)$
RETURN

FLOW CHART & BLOCK DIAGRAM

Form DEN 1103-01 (4-64)

Application SUBROUTINE XDOT Date OCTOBER 1970 Page 1 of 5

Procedure ____ Drawn By ____



$$
\begin{aligned}
w_{1x} &= w_{0x} + \Omega_1 \\
w_{1y} &= w_{0y} * C\theta_1 + w_{0z} * S\theta_1 \\
w_{1z} &= - w_{0y} * S\theta_1 + w_{0z} * C\theta_1
\end{aligned}
$$

$$
\begin{aligned}
\dot{\ell}_{2x} &= - w_{1z} * \ell_{2y} + w_{1y} * \ell_{2z} + S_{2x} * \dot{S} \\
\dot{\ell}_{2y} &= w_{1z} * \ell_{2x} - w_{1x} * \ell_{2z} + S_{2y} * \dot{S} \\
\dot{\ell}_{2z} &= - w_{1y} * \ell_{2x} + w_{1x} * \ell_{2y} + S_{2z} * \dot{S} \\
\bar{R}_0 &= \bar{R}_1 - \bar{d}_{01} \\
w_{3x} &= w_{1x} \\
w_{3y} &= \Omega_3 * S_{3y} + w_{1y} \\
w_{3z} &= \Omega_3 * S_{3z} + w_{1z} \\
\dot{\ell}_{3x} &= - w_{3z} * \ell_{3y} + w_{3y} * \ell_{3z} \\
\dot{\ell}_{3y} &= w_{3z} * \ell_{3x} - w_{3x} * \ell_{3z} \\
\dot{\ell}_{3z} &= - w_{3y} * \ell_{3x} + w_{3x} * \ell_{3y} \\
w_{4x} &= w_{1x} \\
w_{4y} &= \Omega_4 * S_{4y} + w_{1y} \\
w_{4z} &= \Omega_4 * S_{4z} + w_{1z} \\
\dot{\ell}_{4x} &= - w_{4z} * \ell_{4y} + w_{4y} * \ell_{4z} \\
\dot{\ell}_{4y} &= w_{4z} * \ell_{4x} - w_{4x} * \ell_{4z} \\
\dot{\ell}_{4z} &= - w_{4y} * \ell_{4x} + w_{4x} * \ell_{4y} \\
\dot{d}_{01x} &= - w_{0z} * d_{01y} + w_{0y} * d_{01z} \\
\dot{d}_{01y} &= w_{0z} * d_{01x} - w_{0x} * d_{01z} \\
\dot{d}_{01z} &= - w_{0y} * d_{01x} + w_{0x} * d_{01y} \\
\dot{d}_{13x} &= - w_{1z} * d_{13y} + w_{1y} * d_{13z} \\
\dot{d}_{13y} &= w_{1z} * d_{13x} - w_{1x} * d_{13z} \\
\dot{d}_{13z} &= - w_{1y} * d_{13x} + w_{1x} * d_{13y} \\
\dot{d}_{14x} &= - w_{1z} * d_{14y} + w_{1y} * d_{14z} \\
\dot{d}_{14y} &= w_{1z} * d_{14x} - w_{1x} * d_{14z} \\
\dot{d}_{14z} &= - w_{1y} * d_{14x} + w_{1x} * d_{14y}
\end{aligned}
$$

FLOW CHART & BLOCK DIAGRAM

Application SUBROUTINE XDOT CONTINUED Date OCTOBER 1970 Page 2 of 5

Procedure ______ Drawn By GARY JOHNSON

CONTINUED FROM THE PREVIEUS PAGE

$$\dot{R}_{1x} = (m_o/m) * \dot{d}_{01x} - (m_2/m) * \dot{\ell}_{2x}$$
$$- (m_3/m) * (\dot{d}_{13x} + \dot{\ell}_{3x}) - (m_4/m) * (\dot{d}_{14x} + \dot{\ell}_{4x})$$
$$\dot{R}_{1y} = (m_o/m) * (\dot{d}_{01y} * C\theta_1 + \dot{d}_{01z} * S\theta_1) - (m_2/m) * \dot{\ell}_{2y}$$
$$- (m_3/m) * (\dot{d}_{13y} + \dot{\ell}_{3y}) - (m_4/m) * (\dot{d}_{14y} + \dot{\ell}_{4y})$$
$$\dot{R}_{1z} = (m_o/m) * (- \dot{d}_{01y} * S\theta_1 + \dot{d}_{01z} * C\theta_1) - (m_2/m) * \dot{\ell}_{2z}$$
$$- (m_3/m) * (\dot{d}_{13z} + \dot{\ell}_{3z}) - (m_4/m) * (\dot{d}_{14z} + \dot{\ell}_{4z})$$

$$\dot{\bar{R}}_2 = \dot{\bar{R}}_1 + \dot{\bar{\ell}}_2$$

$$\dot{\bar{R}}_3 = \dot{\bar{R}}_1 + \dot{\bar{d}}_{13} + \dot{\bar{\ell}}_3$$

$$\dot{\bar{R}}_4 = \dot{\bar{R}}_1 + \dot{\bar{d}}_{14} + \dot{\bar{\ell}}_4$$

CALL MULT (I.E. $\bar{H}_0 = [I_0] * \bar{w}_0$)

CALL MULT (I.E. $\bar{H}_1 = [I_1] * \bar{w}_1$)

$$h'_{3x} = m_3 * (- \ell_{3z} * \dot{R}_{3y} + \ell_{3y} * \dot{R}_{3z})$$
$$h'_{3y} = m_3 * (\ell_{3z} * \dot{R}_{3x} - \ell_{3x} * \dot{R}_{3z})$$
$$h'_{3z} = m_3 * (- \ell_{3y} * \dot{R}_{3x} + \ell_{3x} * \dot{R}_{3y})$$
$$h'_{4x} = m_4 * (- \ell_{4z} * \dot{R}_{4y} + \ell_{4y} * \dot{R}_{4z})$$
$$h'_{4y} = m_4 * (\ell_{4z} * \dot{R}_{4x} - \ell_{4x} * \dot{R}_{4z})$$
$$h'_{4z} = m_4 * (- \ell_{4y} * \dot{R}_{4x} + \ell_{4x} * \dot{R}_{4y})$$
$$h'_{1y} = H_{1y} + h'_{3y} + h'_{4y}$$
$$- m_2 * (- \ell_{2y} * \dot{R}_{2x} + \ell_{2x} * \dot{R}_{2z})$$
$$- m_3 * (- d_{13z} * \dot{R}_{3x} + d_{13x} * \dot{R}_{3z})$$
$$- m_4 * (- d_{14z} * \dot{R}_{4x} + d_{14x} * \dot{R}_{4z})$$
$$h'_{1z} = H_{1z} + h'_{3z} + h'_{4z}$$
$$- m_2 * (\ell_{2y} * \dot{R}_{2x} - \ell_{2x} * \dot{R}_{2y})$$
$$- m_3 * (d_{13y} * \dot{R}_{3x} - d_{13x} * \dot{R}_{3y})$$
$$- m_4 * (d_{14y} * \dot{R}_{4x} - d_{14x} * \dot{R}_{4y})$$

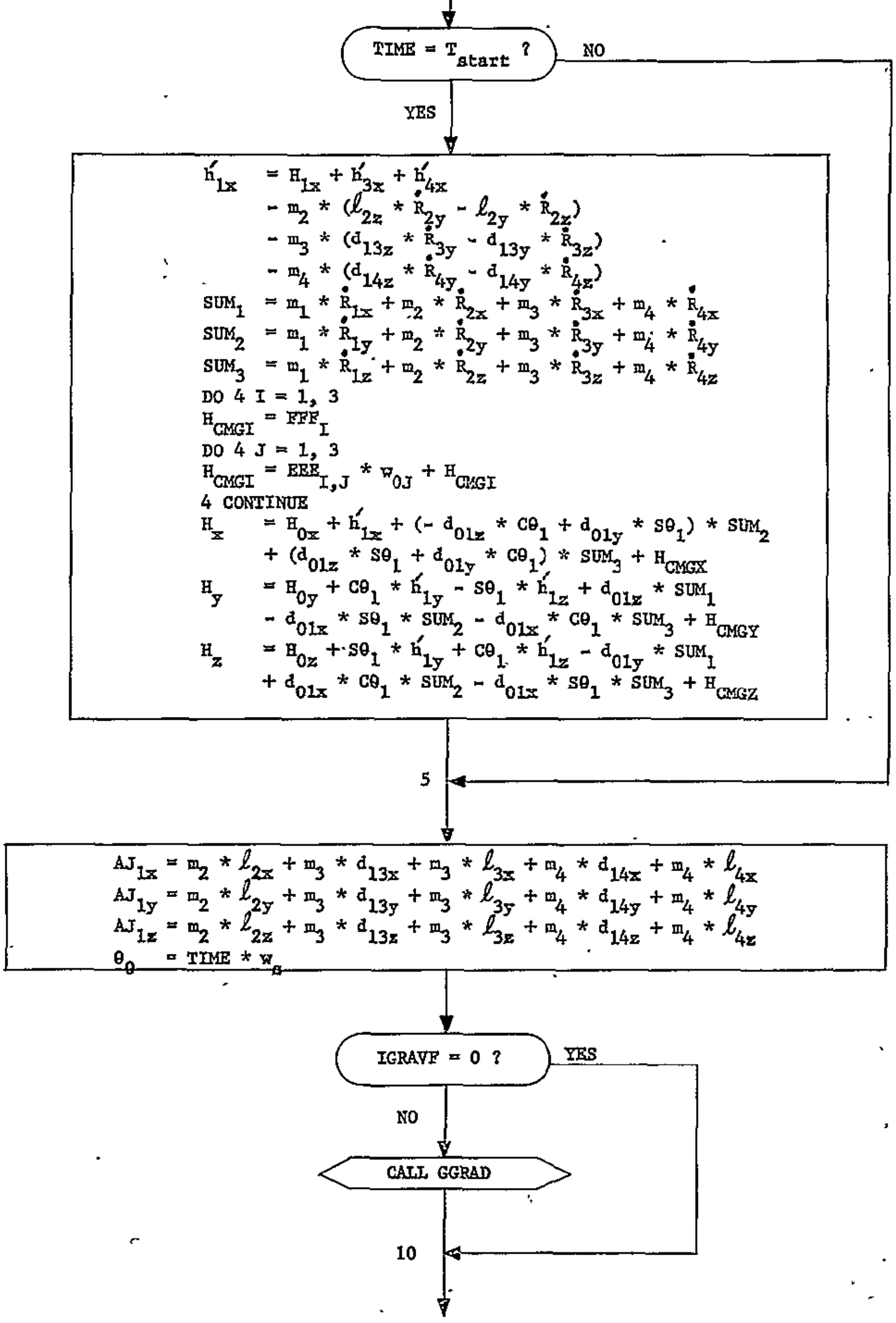
CONTINUE TO THE NEXT PAGE

FLOW CHART & BLOCK DIAGRAM

Application SUBROUTINE XDOT CONTINUED Date OCTOBER 1970 Page 3 of 5

Procedure Drawn By GARY JOHNSON

CONTINUED FROM THE PREVIOUS PAGE



$$\begin{aligned}
h'_{1x} &= H_{1x} + h'_{3x} + h'_{4x} \\
&- m_2 * (\ell_{2z} * \dot{R}_{2y} - \ell_{2y} * \dot{R}_{2z}) \\
&- m_3 * (d_{13z} * \dot{R}_{3y} - d_{13y} * \dot{R}_{3z}) \\
&- m_4 * (d_{14z} * \dot{R}_{4y} - d_{14y} * \dot{R}_{4z}) \\
SUM_1 &= m_1 * \dot{R}_{1x} + m_2 * \dot{R}_{2x} + m_3 * \dot{R}_{3x} + m_4 * \dot{R}_{4x} \\
SUM_2 &= m_1 * \dot{R}_{1y} + m_2 * \dot{R}_{2y} + m_3 * \dot{R}_{3y} + m_4 * \dot{R}_{4y} \\
SUM_3 &= m_1 * \dot{R}_{1z} + m_2 * \dot{R}_{2z} + m_3 * \dot{R}_{3z} + m_4 * \dot{R}_{4z}
\end{aligned}$$

```
DO 4 I = 1, 3
```

$H_{CMGI} = FFF_I$

```
DO 4 J = 1, 3
```

$H_{CMGI} = EEE_{I,J} * w_{0J} + H_{CMGI}$

```
4 CONTINUE
```

$$\begin{aligned}
H_x &= H_{0x} + h'_{1x} + (- d_{01z} * C\theta_1 + d_{01y} * S\theta_1) * SUM_2 \\
&+ (d_{01z} * S\theta_1 + d_{01y} * C\theta_1) * SUM_3 + H_{CMGX} \\
H_y &= H_{0y} + C\theta_1 * h'_{1y} - S\theta_1 * h'_{1z} + d_{01z} * SUM_1 \\
&- d_{01x} * S\theta_1 * SUM_2 - d_{01x} * C\theta_1 * SUM_3 + H_{CMGY} \\
H_z &= H_{0z} + S\theta_1 * h'_{1y} + C\theta_1 * h'_{1z} - d_{01y} * SUM_1 \\
&+ d_{01x} * C\theta_1 * SUM_2 - d_{01x} * S\theta_1 * SUM_3 + H_{CMGZ}
\end{aligned}$$

$$\begin{aligned}
AJ_{1x} &= m_2 * \ell_{2x} + m_3 * d_{13x} + m_3 * \ell_{3x} + m_4 * d_{14x} + m_4 * \ell_{4x} \\
AJ_{1y} &= m_2 * \ell_{2y} + m_3 * d_{13y} + m_3 * \ell_{3y} + m_4 * d_{14y} + m_4 * \ell_{4y} \\
AJ_{1z} &= m_2 * \ell_{2z} + m_3 * d_{13z} + m_3 * \ell_{3z} + m_4 * d_{14z} + m_4 * \ell_{4z} \\
\theta_0 &= TIME * w_0
\end{aligned}$$

CONTINUE TO THE FOLLOWING PAGE

FLOW CHART & BLOCK DIAGRAM

Application SUBROUTINE XDOT CONTINUED Date OCTOBER 1970 Page 4 of 5

Procedure Drawn By GARY JOHNSON

CONTINUED FROM THE PREVIOUS PAGE

$\overline{F}_0 = \overline{F}_{01}$

$\overline{F}_1 = \overline{F}_{11}$

$\overline{T}_{0EF} = \overline{TQ}_{0G} + \overline{TQ}_{0P}$

$\overline{T}_{1EF} = \overline{TQ}_{1G} + \overline{TQ}_{1P}$

$\text{TERM1}(1) = (m_o - m) * d_{01x} - AJ_{1x}$

$\text{TERM1}(2) = (m_o - m) * d_{01y} - C\theta_1 * AJ_{1y} + S\theta_1 * AJ_{1z}$

$\text{TERM1}(3) = (m_o - m) * d_{01z} - S\theta_1 * AJ_{1y} - C\theta_1 * AJ_{1z}$

$\text{TERM2}(1) = m_o * d_{01x} - AJ_{1x}$

$\text{TERM2}(2) = m_o * (C\theta_1 * d_{01y} + S\theta_1 * d_{01z}) - AJ_{1y}$

$\text{TERM2}(3) = m_o * (- S\theta_1 * d_{01y} + C\theta_1 * d_{01z}) - AJ_{1z}$

$\text{ATCPT2}_{1,1} = 0.$

$\text{ATCPT2}_{1,2} = - \text{TERM2}(3)$

$\text{ATCPT2}_{1,3} = \text{TERM2}(2)$

$\text{ATCPT2}_{2,1} = C\theta_1 * \text{TERM2}(3) + S\theta_1 * \text{TERM2}(2)$

$\text{ATCPT2}_{2,2} = - S\theta_1 * \text{TERM2}(1)$

$\text{ATCPT2}_{2,3} = - C\theta_1 * \text{TERM2}(1)$

$\text{ATCPT2}_{3,1} = S\theta_1 * \text{TERM2}(3) - C\theta_1 * \text{TERM2}(2)$

$\text{ATCPT2}_{3,2} = C\theta_1 * \text{TERM2}(1)$

$\text{ATCPT2}_{3,3} = - S\theta_1 * \text{TERM2}(1)$

$\dot{H}_x = w_{0z} * H_y - w_{0y} * H_z + (- \text{TERM1}(3) * F_{0y} + \text{TERM1}(2) * F_{0z})/m$
$\quad + (\text{ATCPT2}_{1,1} * F_{1x} + \text{ATCPT2}_{1,2} * F_{1y} + \text{ATCPT2}_{1,3} * F_{1z})/m$
$\quad + T_{0EFX} + T_{1EFX}$

$\dot{H}_y = - w_{0z} * H_x + w_{0x} * H_z + (\text{TERM1}(3) * F_{0x} - \text{TERM1}(1) * F_{0z})/m$
$\quad + (\text{ATCPT2}_{2,1} * F_{1x} + \text{ATCPT2}_{2,2} * F_{1y} + \text{ATCPT2}_{2,3} * F_{1z})/m$
$\quad + T_{0EFY} + C\theta_1 * T_{1EFY} - S\theta_1 * T_{1EFZ}$

$\dot{H}_z = w_{0y} * H_x - w_{0x} * H_y + (- \text{TERM1}(2) * F_{0x} + \text{TERM1}(1) * F_{0y})/m$
$\quad + (\text{ATCPT2}_{3,1} * F_{1x} + \text{ATCPT2}_{3,2} * F_{1y} + \text{ATCPT2}_{3,3} * F_{1z})/m$
$\quad + T_{0EFZ} + S\theta_1 * T_{1EFY} + C\theta_1 * T_{1EFZ}$

CALL TORK01

$\dot{h}'_{1x} = - w_{1y} * h'_{1z} + w_{1z} * h'_{1y}$
$\quad + \dot{R}_{1y} * (- m_2 * \dot{\ell}_{2z} - m_3 * (\dot{d}_{13z} + \dot{\ell}_{3z}) - m_4 * (\dot{d}_{14z} + \dot{\ell}_{4z}))$
$\quad - \dot{R}_{1z} * (- m_2 * \dot{\ell}_{2y} - m_3 * (\dot{d}_{13y} + \dot{\ell}_{3y}) - m_4 * (\dot{d}_{14y} + \dot{\ell}_{4y}))$
$\quad - AJ_{1y} * (- F_{0y} * S\theta_1/m + F_{0z} * C\theta_1/m)$
$\quad + AJ_{1z} * (F_{0y} * C\theta_1/m + F_{0z} * S\theta_1/m)$
$\quad + AJ_{1z} * F_{1y}/m - AJ_{1y} * F_{1z}/m + T_{1EFX} + T_{0 \rightarrow 1}$

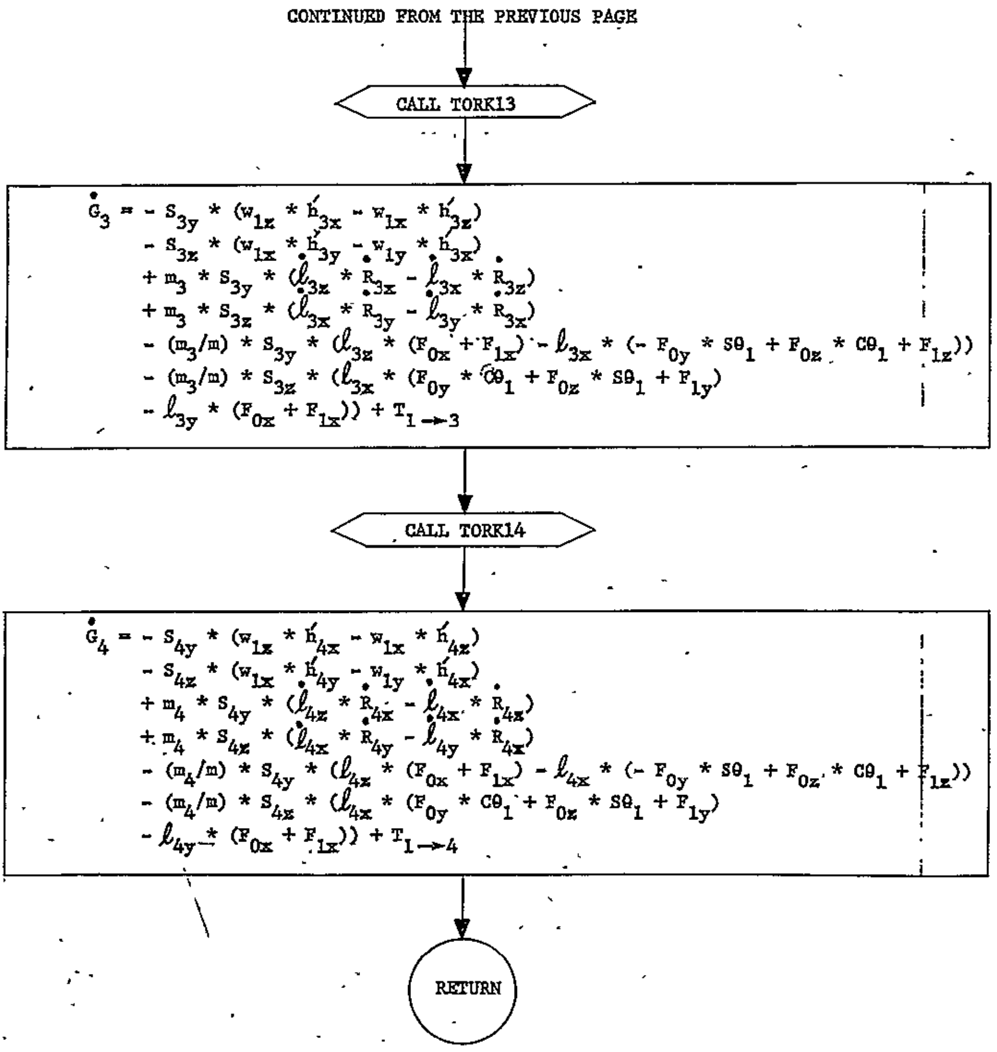
CONTINUE TO THE FOLLOWING PAGE

FLOW CHART & BLOCK DIAGRAM

FORM DEV 1103-01 (4-64)

Application SUBROUTINE XDOT CONTINUED Date OCTOBER 1970 Page 5 of 5

Procedure Drawn By GARY JOHNSON



CONTINUED FROM THE PREVIOUS PAGE

CALL TORK13

$$\begin{aligned}\dot{G}_3 = &- S_{3y} * (w_{1z} * h'_{3x} - w_{1x} * h'_{3z}) \\ &- S_{3z} * (w_{1x} * h'_{3y} - w_{1y} * h'_{3x}) \\ &+ m_3 * S_{3y} * (\dot{\ell}_{3z} * \dot{R}_{3x} - \dot{\ell}_{3x} * \dot{R}_{3z}) \\ &+ m_3 * S_{3z} * (\dot{\ell}_{3x} * \dot{R}_{3y} - \dot{\ell}_{3y} * \dot{R}_{3x}) \\ &- (m_3/m) * S_{3y} * (\ell_{3z} * (F_{0x} + F_{1x}) - \ell_{3x} * (- F_{0y} * S\theta_1 + F_{0z} * C\theta_1 + F_{1z})) \\ &- (m_3/m) * S_{3z} * (\ell_{3x} * (F_{0y} * C\theta_1 + F_{0z} * S\theta_1 + F_{1y}) \\ &- \ell_{3y} * (F_{0x} + F_{1x})) + T_{1 \rightarrow 3}\end{aligned}$$

CALL TORK14

$$\begin{aligned}\dot{G}_4 = &- S_{4y} * (w_{1z} * h'_{4x} - w_{1x} * h'_{4z}) \\ &- S_{4z} * (w_{1x} * h'_{4y} - w_{1y} * h'_{4x}) \\ &+ m_4 * S_{4y} * (\dot{\ell}_{4z} * \dot{R}_{4x} - \dot{\ell}_{4x} * \dot{R}_{4z}) \\ &+ m_4 * S_{4z} * (\dot{\ell}_{4x} * \dot{R}_{4y} - \dot{\ell}_{4y} * \dot{R}_{4x}) \\ &- (m_4/m) * S_{4y} * (\ell_{4z} * (F_{0x} + F_{1x}) - \ell_{4x} * (- F_{0y} * S\theta_1 + F_{0z} * C\theta_1 + F_{1z})) \\ &- (m_4/m) * S_{4z} * (\ell_{4x} * (F_{0y} * C\theta_1 + F_{0z} * S\theta_1 + F_{1y}) \\ &- \ell_{4y} * (F_{0x} + F_{1x})) + T_{1 \rightarrow 4}\end{aligned}$$

RETURN

APPENDIX E, SUBROUTINE DESCRIPTIONS

Subroutine: CMG

Purpose: This subroutine computes the angular momenta produced by the CMGs located on body 0. Either 2 degree of freedom, 1 degree of freedom or reaction wheels can be accommodated.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	NUMCMG	N_{CMG}	The number of CMGs located on body 0
I	IDOF(K)	I_{DOF}	The number of degrees of freedom for the <u>kth</u> CMG
I	HW(K)	$\bar{H}$	The momentum of the <u>kth</u> wheel
I	AII(J,K,M)	I_I	The inertia matrix of the inner gimbal including wheel. The subscript J refers to the CMG being referenced, K, and M are dimensioned 3 and accommodate the inertia matrix
I	AIO(J,K,M)	I_o	The inertia matrix of the outer gimbal. The meaning of the subscripts are the same as AII(J,K,M)
I	THATA(J)	θ_j	The inner gimbal angle
I	FEE(J)	ϕ_j	The outer gimbal angle
I	THATA(J)	$\dot{\theta}_j$	The time-derivative of θ_j
I	FEED(J)	$\dot{\phi}_j$	The time derivative of ϕ_j
0	FFF(M)	f	The total angular momentum of the CMGs which is not a function of the angular rates
0	EEE(M,N)	E	The total angular momentum of the CMGs which is a function of the angular rates. For further discussion see the Appendix

Subroutines required: None

Discussion: None

Subroutine ATT

Purpose: This routine simulates the action of an attitude control system using reaction jets.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	WO(3)	$\bar{\omega}_0$	Angular rates of Body 0 (stator)
I	TIBO(3,3)	$[I,B]_0$	Transformation from Body 0 to inertial frame
I	CA(3)	$\overline{CA}$	Direction cosines of reference direction
I	TIME	t	Time
I	BODY1I(1,1)	$I_{1_{11}}$	Moment of inertia around spin axis of rotor
I	OMEGA1	Ω_1	Gimbal rate of rotor
I	DELTAT	Δt	Time increment per step
0	TQOP(3)		Control torques
0	FAT(8)		Control forces
I	AOJ(3)		Reaction jet lever arms

Subroutines required: none

Equations programmed: $AK = \Omega_1 I_{1_{11}} \omega \sin\beta/2 / (5.5 + \Delta t)$
coordinate transformations

Discussion: A complete description of this routine is given in the final report.

In order to activate the attitude control function it is necessary to set both IPROPF and IATTIF in the input data.

The attitude section also requires designation of the three direction cosines (CA(1), CA(2), CA(3)) of the direction in inertial space at which control is desired. For a reorientation maneuver the initial orientation of the spin axis is inertial space read in (TIBOI(1,1), TIBOI(2,1), TIBOI(2,3) are the initial direction cosines of the spin axis in inertial space) can be specified different from CA. It is necessary, however, to ensure that the angle between the initial direction of the spin axis and the direction to which it is commanded to be redirected be not greater than 60° for the attitude control routine employed in this program.

The propulsion control section must be supplied with appropriate jet couple lengths and control gains for removal of transverse angular rates (see discussion on PCON subroutine). The control can be on either Body 0 or Body 1 or on both.

Subroutine: EMCALC

Purpose: Subroutine EMCALC assembles the M matrix used in the calculations from which the angular velocities are computed.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	THETA1	θ_1	Angular displacement between bodies 0 and 1
I	R1(3)	$\bar{r}_1$	Vector distance from system c.m. to the c.m. of body 1
I	R2(3)	$\bar{r}_2$	Vector distance from system c.m. to the c.m. of body 2
I	R3(3)	$\bar{r}_3$	Vector distance from system c.m. to the c.m. of body 3
I	R4(3)	$\bar{r}_4$	Vector distance from system c.m. to the c.m. of body 4
I	EL2(3)	$\bar{\ell}_2$	Vector position of the movable mass
I	EL3(3)	$\bar{\ell}_3$	Vector position of body 3 from hinge line s_3
I	EL4(3)	$\bar{\ell}_4$	Vector position of body 4 from hinge line s_4
I	D01(3)	$\bar{d}_{01}$	Vector distance from the c.m. of body 0 to the hinge line of body 1
I	D13(3)	$\bar{d}_{13}$	Vector distance from the c.m. of body 1 to the hinge line of body 3
I	D14(3)	$\bar{d}_{14}$	Vector distance from the c.m. of body 1 to the hinge line of body 4
I	BOMASS	m_0	Mass of body 0

I	B2MASS	m_2	Mass of body 2
I	B3MASS	m_3	Mass of body 3
I	B4MASS	m_4	Mass of body 4
I	TOTMAS	m_3	Mass of composite body
I		A_1	Coordinate transformation from body 0 to body 1
I	R1(3)	$\bar{r}_1$	Vector distance from system c.m. to the c.m. of body 1
I	R2(3)	$\bar{r}_2$	Vector distance from the system c.m. to the c.m. of body 2
I	R3(3)	$\bar{r}_3$	Vector distance from the system c.m. to the c.m. of body 3
I	BODY0I(3,3)	I_0	The inertia matrix of body 0
I	BODY1I(3,3)	I_1	The inertia matrix of body 1
I	S3(3)	$\bar{s}_3$	The hinge line of body 3
I	S4(3)	$\bar{s}_4$	The hinge line of body 4
Ø	EM(6,6)	m_{ij}	The M matrix

Subroutines required: None

Equations programmed: The M matrix is related to the angular momenta and the angular velocities by the equation shown below:

$$\begin{bmatrix} H_x \\ H_y \\ H_z \\ h'_{1x} \\ h'_3 \cdot s_3 \\ h'_4 \cdot s_4 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} & M_{15} & M_{16} \\ M_{21} & M_{22} & M_{23} & M_{24} & M_{25} & M_{26} \\ M_{31} & M_{32} & M_{33} & M_{34} & M_{35} & M_{36} \\ M_{41} & M_{42} & M_{43} & M_{44} & M_{45} & M_{46} \\ M_{51} & M_{52} & M_{53} & M_{54} & M_{55} & M_{56} \\ M_{61} & M_{62} & M_{63} & M_{64} & M_{65} & M_{66} \end{bmatrix} \begin{bmatrix} \omega_{0x} \\ \omega_{0y} \\ \omega_{0z} \\ \Omega_1 \\ \Omega_3 \\ \Omega_4 \end{bmatrix}$$

Subroutine FOMS

Purpose: This integration routine calculates values of a set of variables at time t_{n+1} from their values at t_n and their derivatives at times t_n and t_{n-1}.

Calling Sequence: FOMS (A, B, N, E, TJ)

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	A(I)	$x_{i,n}$	ith variable at time t_n; i = 1, 2 . . .n; I = i + 1
0	A(I)	$x_{i,n+1}$	ith variable at time t_{n+1}
I	B(I)	$\dot{x}_{i,n}$	ith derivative at time t_n
I	N	m	Number of variables to be integrated; m = N-1
I,O	E		Flag to use current derivatives for past derivatives on first integration
I	TJ(I)	$\dot{x}_{i,n-1}$	ith derivative at time t_{n-1}
0	TJ(I)	$\dot{x}_{i,n}$	ith derivative at time t_n, storage for next step

Subroutines required: none

Equations programmed: $x_{n+1} = x_n + 1/2\ \Delta t(3\dot{x}_n - \dot{x}_{n-1})$

Discussion: The variables and derivatives are indexed over values of I from 2 to N. B(1) is used to bring in Δt. A(1) is set up to be used for time but is not used in this program. On the first integration the current derivatives are also used as past derivatives since no past derivatives are available.

Subroutine GGRAD

Purpose: This subroutine calculates the forces and torques due to gravity gradient on Body 0 and Body 1.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	THETO	θ_0	True anomaly
I	TIBO(3,3)	$[I,B]_0$	Transformation from Body 0 to inertial frame
I	BODY0I(3,3)	$[I_0]$	Moment of inertia matrix of Body 0
I	BODY1I(3,3)	$[I_1]$	Moment of inertia matrix of Body 1
0	F01(3)	$\overline{F}_{01}$	Force on Body 0 due to gravity gradient
0	F11(3)	$\overline{F}_{11}$	Force on Body 1 due to gravity gradient
0	TQ0G(3)		Torque on Body 0 due to gravity gradient
0	TQ1G (3)		Torque on Body 1 due to gravity gradient
I	C1		Gravitation constant and earth radius factor
I	B0MASS	m_0	Mass of Body 0
I	B1MASS	m_1	Mass of Body 1
I	THETA1	θ_1	Gimbal angle of Body 1
I	R0 (3)	r_0	Distance between centers of mass of Body 0 and system
I	R1 (3)	r_1	Distance between centers of mass of Body 1 and system

Equations programmed:

Torque: $$T_G = -\frac{3\mu}{R_0^{\,5}}\,\overline{R}_0 \times \left[\overline{\overline{I}}\cdot\overline{R}_0\right]$$

Force: $$F_G = \frac{\mu m}{R_0^{\,3}}\left(\bar{r} - \frac{3\overline{R}_0\cdot\bar{r}}{R_0^{\,2}}\,\overline{R}_0\right)$$

where R_0 = distance to center of earth

r = distance between center of mass of body and center of mass of system

Discussion: The translational motion is limited to circular orbits in the X-Y inertial plane in order to simplify transformations and relationships in this subroutine. Gravity gradient effects on Body 2, Body 3 and Body 4 are neglected.

Subroutine: HCON

Purpose: The subroutine contains the control laws in terms of gimbal angle rates for the CMGs. In most cases the user must furnish his own control law.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I			Any program variables to be used in the control law
0	FEED(J)	$\dot{\phi}_J$	Outer gimbal rate of the Jth CMG
0	THATAD(J)	$\dot{\theta}_J$	Inner gimbal rate of the Jth CMG

Subroutines required: CMG

Discussion: None

Subroutine MULT

Purpose: Subroutine MULT multiplies either and 3 x 3 matrix by a 3 x 1 matrix or a 3 x 3 matrix by a 3 x 3 matrix.

Calling sequence: CALL MUTL (C, A, B, F, D, E, MTYPE)

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	A		A 3 x 3 matrix used in the matrix multiplication C = A x B
I	B		A 3 x 1 matrix used in the matrix multiplication C = A x B
Ø	C		A 3 x 1 matrix which is the result of A x B
I	D		A 3 x 3 matrix used in the matrix multiplication F = D x E
I	E		A 3 x 3 matrix used in the matrix multiplication F = D x E
Ø	F		A 3 x 3 matrix which is the result of D x E
I	MTYPE		A flag which determines the type of matrix multiplication being performed. If MTYPE = 1 then C = A x B is performed, if MTYPE ≠ 1 then D = E x F is performed.

Discussion: When subroutine MULT is used to multiply a 3 x 3 matrix by a 3 x 1 matrix variables F, D, E are dummies, when used to multiply two 3 x 3 matrices then variables C, A, B are dummies.

Subroutine PCON

Purpose: To simulate a reaction jet control system for removing transverse components of angular velocity, maintaining the spin rate of the rotor and removing spin axis angular rates of the stator.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	WO	$\bar{\omega}_0$	Angular rates of Body 0
I	THETA1	θ_1	Angle of Body 1 with respect to Body 0
I	OMEGA1	Ω_1	Angular velocity of Body 1 with respect to Body 0
I	CGAIN0(3)		Control gains of stator jets
I	AOJ(3)		Stator jet couple arm length
I	CGAIN1(2)		Control gains of rotor jets
I	A1J(2)		Rotor jet couple arm length
0	TQ0P(3)		Control torques on stator
0	TQ1P(3)		Control torques on rotor
0	FPT(5)		Control forces

Subroutines required: none

Equations programmed: none

Discussion: Certain restrictions on the configuration of the reaction jets have been assumed.

Associated with the torque around an axis there are four jets. One pair is in a pure couple to produce torque in one direction. The other pair, identical in location and strength, is oppositely directed. No torque is produced around axes other than the one designated.

In setting up input data for the reaction jet controls, the control gain of the jets of the couple and the distance between the two jets forming the couple must be specified. Since the couples are similar for the two directions, this data is read only once for each axis.

The firing of jets on Body 1 (rotor) must be timed according to alignment with stator axes. Gimbal angle sensing is used in this routine to provide this timing.

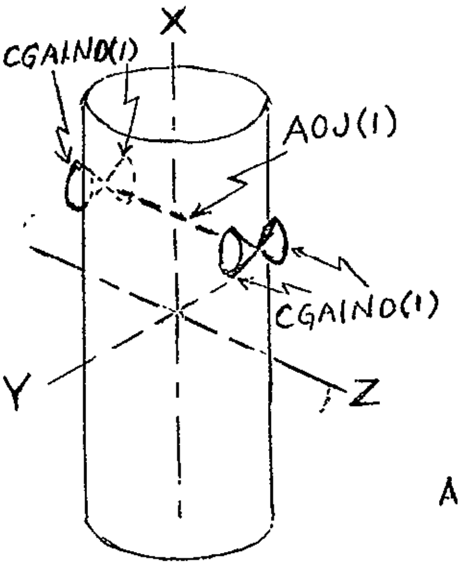
The PCON subroutine simulates a reaction jet control on angular rates. Transverse control torques are made proportional and opposite to transverse velocities. Such a control is similar in effect to an external frictional force acting against transverse motion. For a transverse torque T_T related to the magnitude of transverse rate ω_T by $T_T = - K \ \omega_T$ some estimate of the decrease in magnitude of transverse rate over a time interval t is given by $\omega_T \approx \omega_{To} \ e^{- K/I_T \ t}$ where ω_{To} is the initial transverse angular rate and $I_T = \sqrt{(I_{0_{22}} + I_{1_{22}}) \ (I_{0_{33}} + I_{1_{33}})}$. This estimate will be good for jets on Body 0 where both transverse axes are controlled. For the single axis jets on the rotor the control will be much slower due to the time that the rotor spends in unfavorable positions.

In Figure   jet configurations are illustrated. Note that it is not necessary for jets to be symmetrically located with respect to any coordinate axes or planes. It is necessary, however, that the X-jet couple arm be parallel to the Y-Z plane, the Y-jet couple arm be parallel to the X-Z plane, etc. The data deck is simplified by requiring only magnitudes of jet couple arms without regard for actual directions or components.

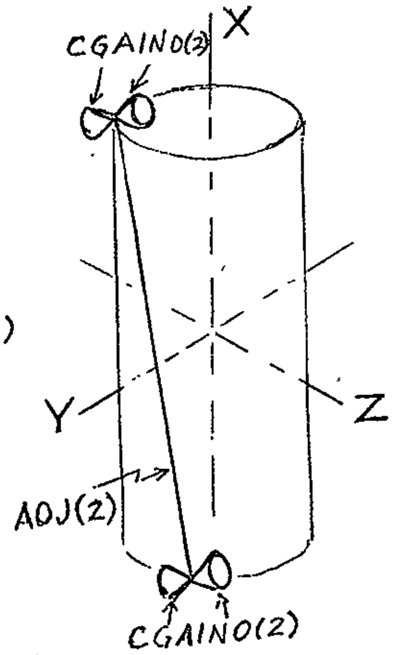
The reaction jets are activated by setting IPROPF = 1 and supplying control gain and couple arm length data. If it is desired to leave a certain axis uncontrolled, the corresponding control gain is set to zero, but the couple arm length should be given some non-zero value in order to prevent division by zero in calculation of impulse contributions.

Body 0

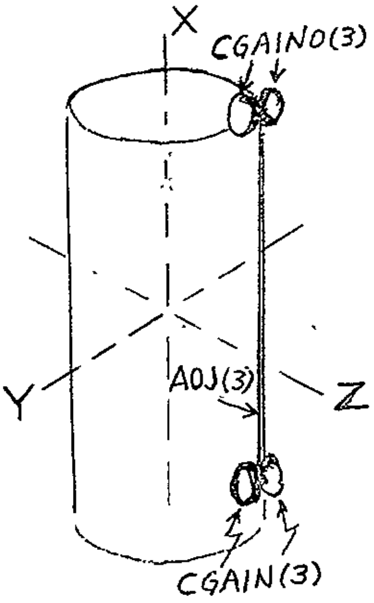
X-Axis Jets



Y-Axis Jets

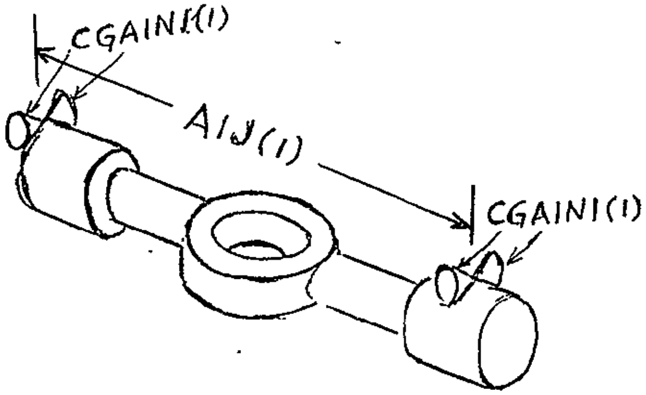


Z-Axis Jets

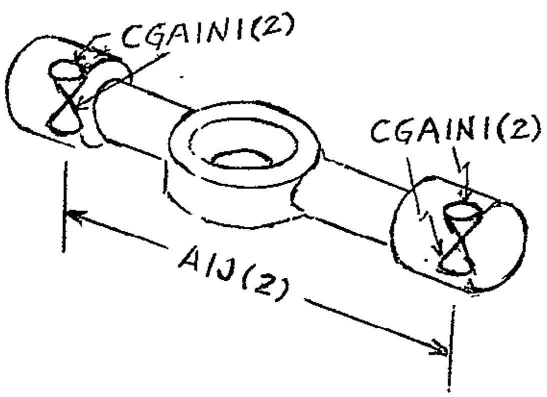


Body 1

X-Axis Jets



Y-Axis Jets



Subroutine: RECALC

Purpose: The purpose of this subroutine is to compute the distance from combined system center of mass to the center of mass of the various bodies.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	BOMASS	m_0	Mass of body 0
I	B2MASS	m_2	Mass of body 2
I	B3MASS	m_3	Mass of body 3
I	B4MASS	m_4	Mass of body 4
I	TOTMASS	m	Mass of composite body
I	D01(3)	$\bar{d}_{01}$	Vector distance from the c.m. of body 0 to the hinge line of body 1
I	EL2(3)	$\bar{\ell}_2$	Vector position of the movable mass
I	D13(3)	$\bar{d}_{13}$	Vector distance from the c.m. of body 1 to the hinge line of body 3
I	D14(3)	$\bar{d}_{14}$	Vector distance from the c.m. of body 1 to the hinge line of body 4
I	EL3(3)	$\bar{\ell}_3$	Vector position of body 3 from the hinge line s_3
I	EL4(3)	$\bar{\ell}_4$	Vector position of body 4 from the hinge line s_4
0	R1(3)	$\bar{r}_1$	Vector distance from system c.m. to the c.m. of body 1
0	R2(3)	$\bar{r}_2$	Vector distance from the system c.m. to the c.m. of body 2

0	R3(3)	$\bar{r}_3$	Vector distance from the system c.m. to the c.m. of body 3
0	R4(4)	$\bar{r}_4$	Vector distance froι the system c.m. to the c.m. of body 4

Subroutines required: None

Equations programmed:

$$r_1 = \frac{m_0}{m} \bar{d}_{01} - \frac{m_2}{m} \bar{\ell}_2 - \frac{m_3}{m} (\bar{d}_{13} + \bar{\ell}_3) - \frac{m_4}{m} (\bar{d}_{14} + \ell_4)$$

$$r_2 = \frac{m_0}{m} d_{01} + (1 - \frac{m_2}{m}) \ell_2 - \frac{m_3}{m} (d_{13} + \ell_3) - \frac{m_4}{m} (d_{14} + \ell_4)$$

$$r_3 = \frac{m_0}{m} d_{10} - \frac{m_2}{m} \ell_2 + (1 - \frac{m_3}{m}) (d_{13} + \ell_3) - \frac{m_4}{m} (d_{14} + \ell_4)$$

$$r_4 = \frac{m_0}{m} d_{01} - \frac{m_2}{m} \ell_2 - \frac{m_3}{m} (d_{13} + \ell_3) + (1 - \frac{m_4}{m}) (d_{14} + \ell_4)$$

m_0, m_2, m_3, m_4, m,

d_{10}, d_{13}, d_{14}

ℓ_2, ℓ_3, ℓ_4

r_1, r_2, r_3, r_4

Subroutine SCALC

Purpose: This routine supplies the position of the movable mass, Body 2.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
O	S	S	Distance of Body 2 from D12 along direction S2

Note: Other I/O variables may be employed depending on the formulation of the subroutine.

Subroutines required: Not specified

Equations programmed: Not specified

Discussion: This subroutine will be constructed to suit the needs of the user. Any variables appearing in the common region can be employed as input/output variables.

Subroutine SDCALC

Purpose: This routine supplies the speed of the movable mass, Body 2.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
0	SDOT	$\dot{s}$	Magnitude of velocity of Body 2 along direction S2.

Note: Other I/O variables may be employed depending on the formulation of the subroutine.

Subroutines required: Not specified

Equations programmed: Not specified

Discussion: This subroutine will be constructed to suit the needs of the user. Any variables appearing in the common region can be employed as input/output variables.

Subroutine: SYEQNS(A,N,NR,NC,FLAG)

Purpose: Subroutine SYEQNS solves a set of linear simultaneous equations Ax = c to determine the column vector x.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I/O	A	A	A is the system matrix as depicted above. Also the answer x will appear as the N+1 at column of matrix A
I	N	N	N is the number of linear equations to be solved
I	NR	NR	NR is the number of rows in A
I	NC	NC	NC is the number of columns in A
0	FLAG	FLAG	If FLAG = 0 as solution exists if FLAG = 1 no solution exists

Subroutines required: None

Discussion: None

Subroutine: TORK01

Purpose: The purpose of this subroutine is to compute the torque acting between bodies 0 and 1. The present version of TORK01 contains a frictional torque as well as torque motor with appropriate control law.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I			Any of the variables carried through common
0	T01	T_{01}	The torque acting between body 0 and body 1

Discussion: None

Subroutine: TORK13

Purpose: This subroutine computes the torque between body 1 and body 3.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	THETA3	θ_3	Angular displacement of body 3 about the hinge line s_3
I	OMEGA3	3	Angular velocity of body 3 about the hinge line s_3
I	CP1	CP1	Gain for ω_3
I	CP2	CP2	Gain for θ_3
0	T13	T_{13}	The torque acting between bodies 1 and 3

Subroutines required: None

Equations programmed:

$$T_{1 \rightarrow 3} = - \mathrm{CP1} \mathrm{\ x\ } \omega_3 - \mathrm{CP2} \mathrm{\ x\ } \theta_3$$

Subroutine: TORK14

Purpose: This subroutine computes the torque acting between body 1 and body 4.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	THETA4	θ_4	Angular displacement of body 4 about the hinge line s_4
I	ØMEGA4	4	Angular velocity o: body 4 about the hinge line s_4
I	CP1	CP_1	Gain for ω_4
I	CP2	CP_2	Gain for θ_3
0	T14	T_{14}	The torque acting between bodies 1 and 4

Subroutines required: None

Equation programmed:

$$T_{1 \rightarrow 4} = - \mathrm{CP1} \times \omega_4 - \mathrm{CP2} \times (\theta_4 - \pi)$$

Subroutine XDOT

Purpose: To compute the derivative of the unconstrained components of angular momenta as well as the variable required for these calculations.

Segment 1

Purpose: To compute the angular velocity of body 1.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	OMEGA1	Ω_1	Angular velocity between bodies 1 and 0
I	W0(3)	$\bar{\omega}_0$	Angular velocity of body 0
Ø	W1(3)	$\bar{\omega}_1$	Angular velocity of body 1
I	THETA1	θ_1	Angular displacement between bodies 1 and 0

Subroutines required: None

Equations Programmed:

$$\begin{array}{r|l|l}
\Omega_1 \rightarrow & \omega_{1x} = \omega_{0x} \qquad\qquad\qquad\qquad\qquad\qquad + \Omega_1 & \\
\bar{\omega}_0 \rightarrow & \omega_{1y} = \qquad \omega_{0y}\cos\theta_1 + \omega_{0z}\sin\theta_1 & \rightarrow \bar{\omega}_1 \\
 & \omega_{1z} = \quad -\omega_{0y}\sin\theta_1 + \omega_{0z}\cos\theta_1 &
\end{array}$$

Discussion: The angular velocity of body 0 ($\bar{\omega}_0$) is transformed to the body 1 coordinate system and is added to the primary gimbal rate to obtain the angular velocity of body l ($\bar{\omega}_1$).

Segment 2.

Purpose: To compute the vector position of the movable mass from the center-of-mass (c.m.) of body 1.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	S	s	Movable mass travel, a specified scalar function of time
I	D12(3)	$\bar{d}_{12}$	Fixed vector (in body 1 coordinates) locating the path of movable mass from c.m. of body 1
I	S2(3)	$\bar{s}_2$	Unit vector defining direction in which the movable mass travels
Ø	EL2(3)	$\bar{\ell}_2$	Position of movable mass (in body 1 coordinates) from c.m. of body 1

Subroutines required: None

Equations programmed:

$$\begin{array}{c} \xrightarrow{s} \\ \bar{d}_{12} \\ \bar{s}_2 \end{array} \boxed{\begin{array}{l} \ell_{2x} = d_{12x} + ss_{2x} \\ \ell_{2y} = d_{12y} + ss_{2y} \\ \ell_{2z} = d_{12z} + ss_{2z} \end{array}} \xrightarrow{\bar{\ell}_2}$$

Discussion: The position of the movable mass is computed from the vector equation $\bar{\ell}_2 = \bar{d}_{12} + \bar{s}_{2s}$.

Segment 3

Purpose: To compute the positions of bodies 3 and 4.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	THETA3	θ_3	Angular position of pendulum 3
I	S3(3)	$\bar{s}_3$	Hinge line of pendulum 3
I	PEND3L	ℓ_3	Scalar length of pendulum 3
Ø	EL3(3)	$\bar{\ell}_3$	Vector position of pendulum 3 from hinge line $\bar{s}_3$
I	THETA4	θ_4	Angular position of pendulum 4
I	S4(3)	$\bar{s}_4$	Hinge line of pendulum 4 in body 1 coordinated (no dimensions)
I	PEND4L	ℓ_4	Scalar length of pendulum 4
Ø	EL4(3)	$\bar{\ell}_4$	Vector position of pendulum 4

Subroutines required: None

Equations programmed:

body 1 $\theta_3 \rightarrow$ $\bar{s}_3 \rightarrow$ $\ell_3 \rightarrow$

$$\boxed{\begin{aligned} \ell_{3x} &= \ell_3 \sin\theta_3 \\ \ell_{3y} &= -\ell_3 \cos\theta_3 \, s_{3z} \\ \ell_{3z} &= \ell_3 \cos\theta_3 \, s_{3y} \end{aligned}} \xrightarrow{\bar{\ell}_3}$$

body 1 coords.

$\theta_4 \rightarrow$ $\bar{s}_4 \rightarrow$ $\ell_4 \rightarrow$

$$\boxed{\begin{aligned} \ell_{4x} &= \ell_4 \sin\theta_4 \\ \ell_{4y} &= -\ell_4 \cos\theta_4 \, s_{4z} \\ \ell_{4z} &= \ell_4 \cos\theta_4 \, s_{4y} \end{aligned}} \xrightarrow{\bar{\ell}_4}$$

body 1 coords.

Discussion: None

Segment 4

Purpose: To compute the center of mass equations for bodies 0, 1, 2 and 3. $\bar{r}_j$ is defined as the vector distance from the system center of mass to the center of mass of the jth body.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	EL2(3)	$\bar{\ell}_2$	Vector position of the movable mass
I	EL3(3)	$\bar{\ell}_3$	Vector position of body 3 from hinge line s_3
I	EL4(3)	$\bar{\ell}_4$	Vector position of body 4 from hinge line s_4
I	D01(3)	$\bar{d}_{01}$	Vector distance from the c.m. of body 0 to the hinge line of body 1
I	D13(3)	$\bar{d}_{13}$	Vector distance from the c.m. of body 1 to the hinge line of body 3
I	D14(3)	$\bar{d}_{14}$	Vector distance from the c.m. of body 1 to the hinge line of body 4
I	B0MASS	m_0	Mass of body 0 (slugs)
I	B2MASS	m_2	Mass of body 2
I	B3MASS	m_3	Mass of body 3
I	B4MASS	m_4	Mass of body 4
I	TOTMAS	m	Mass of composite body
I		A_1	Coordinate transformation from body 0 to body 1

Ø	R1(3)	$\bar{r}_1$	Vector distance from system c.m. to the c.m. of body 1
Ø	R2(3)	$\bar{r}_2$	Vector distance from the system c.m. to the c.m. of body 2
Ø	R3(3)	$\bar{r}_3$	Vector distance from the system c.m. to the c.m. of body 3
Ø	R4(3)	$\bar{r}_4$	Vector distance from the system c.m. to the c.m. of body 4

Subroutines required: None

Equations programmed:

$$\begin{bmatrix} r_{1x} \\ r_{1y} \\ r_{1z} \end{bmatrix} = \frac{m_0}{m} A_1 \begin{bmatrix} d_{01x} \\ d_{01y} \\ d_{01z} \end{bmatrix} - \frac{m_2}{m} \begin{bmatrix} \ell_{2x} \\ \ell_{2y} \\ \ell_{2z} \end{bmatrix} - \frac{m_3}{m} \begin{bmatrix} d_{13x} + \ell_{3x} \\ d_{13y} + \ell_{3y} \\ d_{13z} + \ell_{3z} \end{bmatrix} - \frac{m_4}{m} \begin{bmatrix} d_{14x} + \ell_{4x} \\ d_{14y} + \ell_{4y} \\ d_{14z} + \ell_{4z} \end{bmatrix}$$

$$\begin{bmatrix} r_{2x} \\ r_{2y} \\ r_{2z} \end{bmatrix} = \begin{bmatrix} r_{1x} \\ r_{1y} \\ r_{1z} \end{bmatrix} + \begin{bmatrix} \ell_{2x} \\ \ell_{2y} \\ \ell_{2z} \end{bmatrix}$$

$$\begin{bmatrix} r_{3x} \\ r_{3y} \\ r_{3z} \end{bmatrix} = \begin{bmatrix} r_{1x} \\ r_{1y} \\ r_{1z} \end{bmatrix} + \begin{bmatrix} d_{13x} \\ d_{13y} \\ d_{13z} \end{bmatrix} + \begin{bmatrix} \ell_{3x} \\ \ell_{3y} \\ \ell_{3z} \end{bmatrix}$$

$$\begin{bmatrix} r_{4x} \\ r_{4y} \\ r_{4z} \end{bmatrix} = \begin{bmatrix} r_{1x} \\ r_{1y} \\ r_{1z} \end{bmatrix} + \begin{bmatrix} d_{14x} \\ d_{14y} \\ d_{14z} \end{bmatrix} + \begin{bmatrix} \ell_{4x} \\ \ell_{4y} \\ \ell_{4z} \end{bmatrix}$$

↑ $\bar{\ell}_2, \bar{\ell}_3, \bar{\ell}_4$

$\bar{d}_{01}, \bar{d}_{13}, \bar{d}_{14}$

A_1

↓ $\bar{r}_1, \bar{r}_2, \bar{r}_3, \bar{r}_4$

(body 1 coords.)

Discussion: None

Segment 5

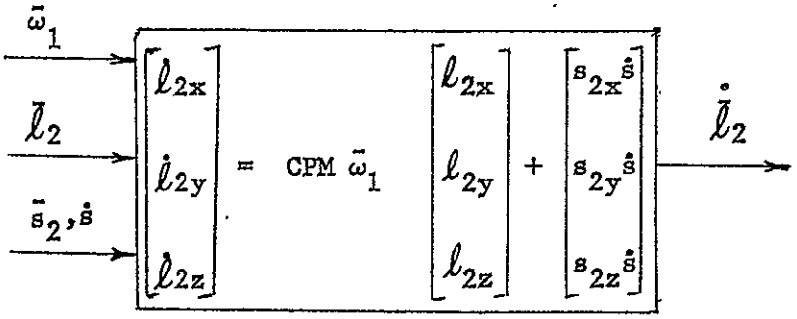
Purpose: To compute the derivative of $\bar{\ell}_2$, the rate at which to movable mass in moving.

Input/output:

I/O	Fortran Statement	Math Symbol	Definition
I	W1(3)	ω_1	Angular velocity of body 1
I	EL2(3)	$\bar{\ell}_2$	Vector position of the movable mass
I	s2(3)	$\bar{s}_2$	Unit vector defining the direction of travel of the movable mass (body 2)
I	S	s	Movable mass travel, a specified scalar function of time
I	sDOT	$\dot{s}$	The time derivative of s.
Ø	EL2DOT(3)	$\dot{\bar{\ell}}_2$	The time derivative of $\bar{\ell}_2$

Subroutines required: SDCALC

Equations programmed:



$$\begin{bmatrix} \dot{\ell}_{2x} \\ \dot{\ell}_{2y} \\ \dot{\ell}_{2z} \end{bmatrix} = \text{CPM } \bar{\omega}_1 \begin{bmatrix} \ell_{2x} \\ \ell_{2y} \\ \ell_{2z} \end{bmatrix} + \begin{bmatrix} s_{2x}\dot{s} \\ s_{2y}\dot{s} \\ s_{2z}\dot{s} \end{bmatrix}$$

where

$$\mathrm{CPM}\bar{\omega}_1 = \begin{bmatrix} 0 & -\mathrm{W1(3)} & \mathrm{W1(2)} \\ \mathrm{W1(3)} & 0 & -\mathrm{W1(1)} \\ -\mathrm{W1(2)} & \mathrm{W1(1)} & 0 \end{bmatrix}$$

Discussion: None

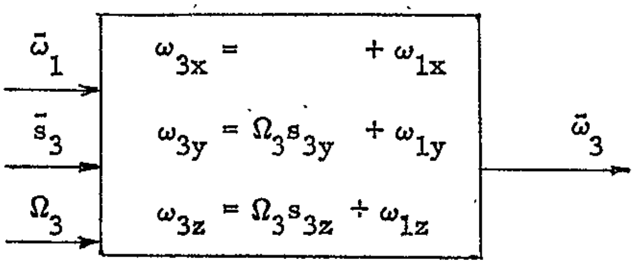
Segment 6

Purpose: To compute the angular velocity of body 3 ($\bar{\omega}_3$).

I/O	Fortran Name	Math Symbol	Definition
I	W1(3)	$\bar{\omega}_1$	Angular velocity of body 1
I	OMEGA3	Ω_3	Angular velocity of body 3 about the hinge line s_3
I	s3(3)	$\bar{s}_3$	The hinge line about which body 3 rotates
Ø	W3(3)	$\bar{\omega}_3$	The angular velocity of body 3

Subroutines required: None

Equations programmed:



$$\omega_{3x} = \qquad + \omega_{1x}$$

$$\omega_{3y} = \Omega_3 s_{3y} + \omega_{1y}$$

$$\omega_{3z} = \Omega_3 s_{3z} + \omega_{1z}$$

Discussion: None

Segment 7

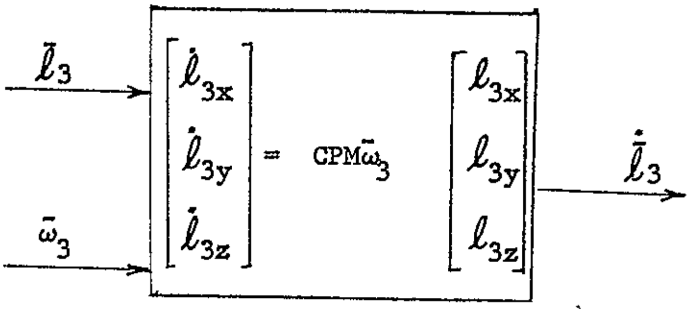
Purpose: To compute the derivative of $\bar{\ell}_3$.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	EL3(3)	$\bar{\ell}_3$	Vector position of body 3 from hinge line $\bar{s}_3$
I	W3(3)	$\bar{\omega}_3$	Angular velocity of body 3
Ø	EL3DOT(3)	$\dot{\bar{\ell}}_3$	Time derivative of $\bar{\ell}_3$

Subroutines required: None

Equations programmed:



$$\begin{bmatrix} \dot{\ell}_{3x} \\ \dot{\ell}_{3y} \\ \dot{\ell}_{3z} \end{bmatrix} = \text{CPM}\bar{\omega}_3 \begin{bmatrix} \ell_{3x} \\ \ell_{3y} \\ \ell_{3z} \end{bmatrix}$$

Discussion: None

Segment 8

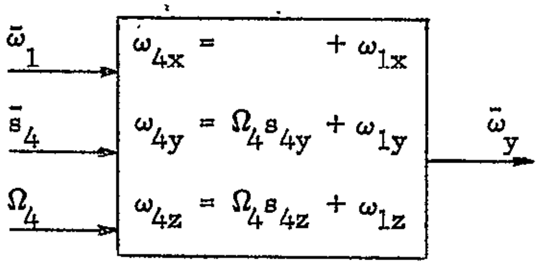
Purpose: To compute the angular velocity of body 4 $\bar{\omega}_4$.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	W1(3)	$\bar{\omega}_1$	Angular velocity of body 1
I	OMEGA4	Ω_4	Angular velocity of body 4 about the hinge line $\bar{s}_4$
I	S4(3)	$\bar{s}_4$	The hinge line about which body 4 rotates
Ø	W4(3)	$\bar{\omega}_4$	Angular velocity of body 4

Subroutines required: None

Equations programmed:



$$\omega_{4x} = \qquad + \omega_{1x}$$

$$\omega_{4y} = \Omega_4 s_{4y} + \omega_{1y}$$

$$\omega_{4z} = \Omega_4 s_{4z} + \omega_{1z}$$

Discussion: None

Segment 9

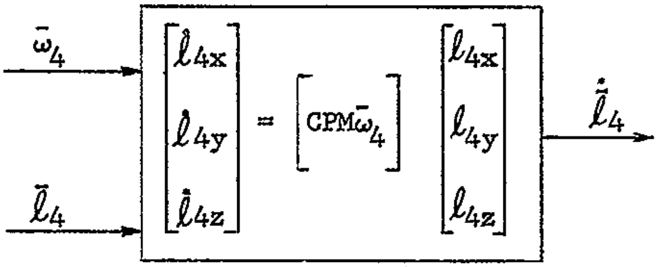
Purpose: To compute the derivative of $\bar{\ell}_4$.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	EL4(3)	$\bar{\ell}_4$	Vector position of body 4 from the hinge line $\bar{s}_4$
I	W4(3)	$\bar{\omega}_4$	Angular velocity of body 4
Ø	EL4DOT(3)	$\dot{\bar{\ell}}_4$	Time derivative of $\bar{\ell}_4$

Subroutines required: None

Equations programmed:



$$\begin{bmatrix} \ell_{4x} \\ \ell_{4y} \\ \dot{\ell}_{4z} \end{bmatrix} = \begin{bmatrix} \mathrm{CPM}\bar{\omega}_4 \end{bmatrix} \begin{bmatrix} \ell_{4x} \\ \ell_{4y} \\ \ell_{4z} \end{bmatrix}$$

Discussion: None

Segment 10

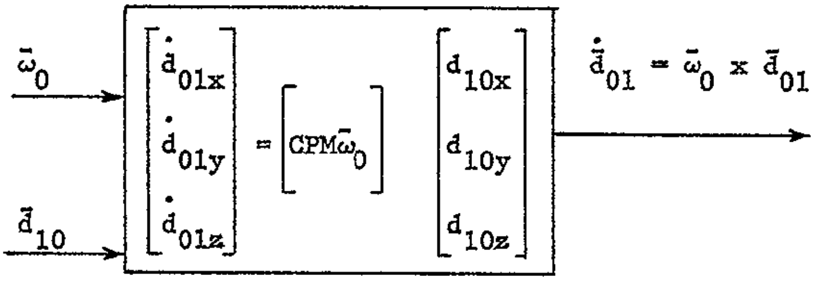
Purpose: To compute the time derivatives of $\bar{d}_{01}$, $\bar{d}_{13}$, and $\bar{d}_{14}$.

Input/output:

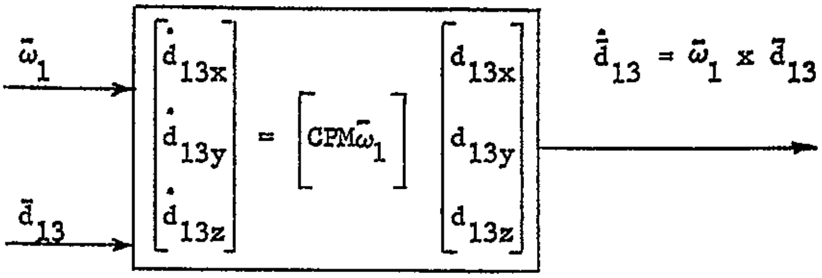
I/O	Fortran Statement	Math Symbol	Definition
I	D01(3)	$\bar{d}_{01}$	Vector distance from the c.m. of body 0 to the hinge line of body 1
I	D13(3)	$\bar{d}_{13}$	Vector distance from the c.m. of body 1 to the hinge line of body 3
I	D14(3)	$\bar{d}_{14}$	Vector distance from the c.m. of body 1 to the hinge line of body 4
I	W1(3)	$\bar{\omega}_1$	Angular velocity of body 1
Ø	D01DOT(3)	$\dot{\bar{d}}_{01}$	Time derivative of $\bar{d}_{01}$
Ø	D13DOT(3)	$\dot{\bar{d}}_{13}$	Time derivative of $\bar{d}_{13}$
Ø	D14DOT(3)	$\dot{\bar{d}}_{14}$	Time derivative of $\bar{d}_{14}$

Subroutines required: None

Equations programmed:

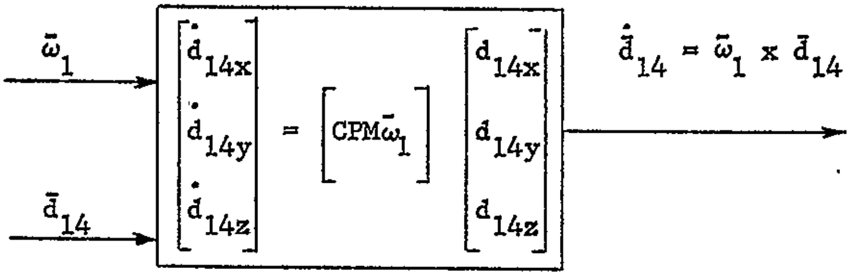


$$\begin{bmatrix} \dot{d}_{01x} \\ \dot{d}_{01y} \\ \dot{d}_{01z} \end{bmatrix} = \begin{bmatrix} \mathrm{CPM}\bar{\omega}_0 \end{bmatrix} \begin{bmatrix} d_{10x} \\ d_{10y} \\ d_{10z} \end{bmatrix}$$



$$\begin{bmatrix} \dot{d}_{13x} \\ \dot{d}_{13y} \\ \dot{d}_{13z} \end{bmatrix} = \begin{bmatrix} CPM\bar{\omega}_1 \end{bmatrix} \begin{bmatrix} d_{13x} \\ d_{13y} \\ d_{13z} \end{bmatrix}$$

$$\dot{\bar{d}}_{13} = \bar{\omega}_1 \times \bar{d}_{13}$$



$$\begin{bmatrix} \dot{d}_{14x} \\ \dot{d}_{14y} \\ \dot{d}_{14z} \end{bmatrix} = \begin{bmatrix} CPM\bar{\omega}_1 \end{bmatrix} \begin{bmatrix} d_{14x} \\ d_{14y} \\ d_{14z} \end{bmatrix}$$

$$\dot{\bar{d}}_{14} = \bar{\omega}_1 \times \bar{d}_{14}$$

Discussion: None

Segment 11

Purpose: To compute the time derivatives to the center of mass variables $\bar{r}_1$, $\bar{r}_2$, $\bar{r}_3$ and $\bar{r}_4$.

Input/output:

I/O	Fortran Statement	Math Symbol	Definition
I	D01DOT(3)	$\dot{\bar{d}}_{01}$	Time derivatives of $\bar{d}_{01}$
I	EL2DOT(3)	$\dot{\bar{\ell}}_2$	Time derivative of $\bar{\ell}_2$
I	D13DOT(3)	$\dot{\bar{d}}_{13}$	Time derivative of $\bar{d}_{13}$
I	EL3DOT(3)	$\dot{\bar{\ell}}_3$	Time derivative of $\bar{\ell}_3$
I	D14DOT(3)	$\dot{\bar{d}}_{14}$	Time derivative of $\bar{d}_{14}$
I	EL4DOT(3)	$\dot{\bar{\ell}}_4$	Time derivative of ℓ_4
I	R1DOT(3)	$\dot{\bar{r}}_1$	Time derivative of $\bar{r}_1$
I	B0MASS	m_0	Mass of body 0
I	B2MASS	m_2	Mass of body 2
I	B3MASS	m_3	Mass of body 3
I	B4MASS	m_4	Mass of body 4
I	TOTMAS	m	Mass of composite body (total mass)
I		A_1	Coordinate transformation from body 0 to body 1
Ø	R1DOT(3)	$\dot{\bar{r}}_1$	Time derivative of $\bar{r}_1$
Ø	R2DOT(3)	$\dot{\bar{r}}_2$	Time derivative of $\bar{r}_2$
Ø	R3DOT(3)	$\dot{\bar{r}}_3$	Time derivative of $\bar{r}_3$
Ø	R4DOT(3)	$\dot{\bar{r}}_4$	Time derivative of $\bar{r}_4$

Subroutines required: None

Equations programmed:

$$\begin{bmatrix} \dot{r}_{1x} \\ \dot{r}_{1y} \\ \dot{r}_{1z} \end{bmatrix} = \frac{m_0}{m} \begin{bmatrix} A_1 \end{bmatrix} \begin{bmatrix} \dot{d}_{01x} \\ \dot{d}_{01y} \\ \dot{d}_{01z} \end{bmatrix} - \frac{m_2}{m} \begin{bmatrix} \dot{\ell}_{2x} \\ \dot{\ell}_{2y} \\ \dot{\ell}_{2z} \end{bmatrix} - \frac{m_3}{m} \begin{bmatrix} \dot{d}_{13x} + \dot{\ell}_{3x} \\ \dot{d}_{13y} + \dot{\ell}_{3y} \\ \dot{d}_{13y} + \dot{\ell}_{3z} \end{bmatrix} - \frac{m_4}{m} \begin{bmatrix} \dot{d}_{14x} + \dot{\ell}_{4x} \\ \dot{d}_{14y} + \dot{\ell}_{4y} \\ \dot{d}_{14z} + \dot{\ell}_{4z} \end{bmatrix}$$

$$\begin{bmatrix} \dot{r}_{2x} \\ \dot{r}_{2y} \\ \dot{r}_{2z} \end{bmatrix} = \begin{bmatrix} \dot{r}_{1x} \\ \dot{r}_{1y} \\ \dot{r}_{1z} \end{bmatrix} + \begin{bmatrix} \dot{\ell}_{2x} \\ \dot{\ell}_{2y} \\ \dot{\ell}_{2z} \end{bmatrix}$$

$$\begin{bmatrix} \dot{r}_{3x} \\ \dot{r}_{3y} \\ \dot{r}_{3z} \end{bmatrix} = \begin{bmatrix} \dot{r}_{1x} \\ \dot{r}_{1y} \\ \dot{r}_{1z} \end{bmatrix} + \begin{bmatrix} \dot{d}_{13x} + \dot{\ell}_{3x} \\ \dot{d}_{13y} + \dot{\ell}_{3y} \\ \dot{d}_{13z} + \dot{\ell}_{3z} \end{bmatrix}$$

$$\begin{bmatrix} \dot{r}_{4x} \\ \dot{r}_{4y} \\ \dot{r}_{4z} \end{bmatrix} = \begin{bmatrix} \dot{r}_{1x} \\ \dot{r}_{1y} \\ \dot{r}_{1z} \end{bmatrix} + \begin{bmatrix} \dot{d}_{14x} + \dot{\ell}_{4x} \\ \dot{d}_{14y} + \dot{\ell}_{4y} \\ \dot{d}_{14z} + \dot{\ell}_{4z} \end{bmatrix}$$

↑ $\dot{\bar{d}}_{01}$ (body 0 coordinates)

$\dot{\bar{d}}_{10}$, $\dot{\bar{d}}_{13}$, $\dot{\bar{d}}_{14}$, $\dot{\bar{\ell}}_2$, $\dot{\bar{\ell}}_3$, $\dot{\bar{\ell}}_4$, (body 1 coordinates)

A_1

↓ $\dot{\bar{r}}_1$, $\dot{\bar{r}}_2$, $\dot{\bar{r}}_3$, $\dot{\bar{r}}_4$

(body 1 coords.)

Discussion: None

Segment 12

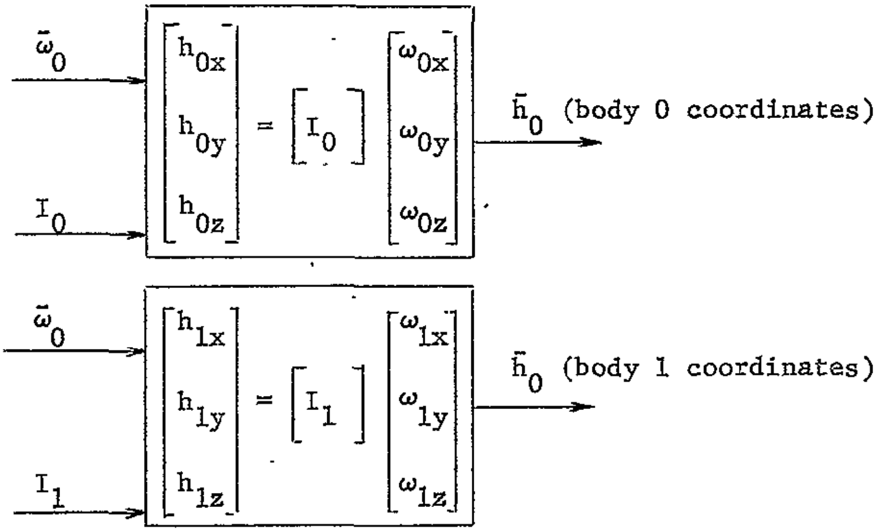
Purpose: To compute the angular momentum of body 0 and body 1.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	W0	$\bar{\omega}_0$	Angular velocity of body 0
I	W1	$\bar{\omega}_1$	Angular velocity of body 1
I	BODY0I	I_0	The inertia of body 0
I	BODY1I	I_1	The inertia of body 1
0	H0	$\bar{h}_0$	The angular momentum of body 0
0	H1	$\bar{h}_1$	The angular momentum of body 1

Subroutines required: MULT

Equations programmed:



$$\begin{bmatrix} h_{0x} \\ h_{0y} \\ h_{0z} \end{bmatrix} = \begin{bmatrix} I_0 \end{bmatrix} \begin{bmatrix} \omega_{0x} \\ \omega_{0y} \\ \omega_{0z} \end{bmatrix}$$

$$\begin{bmatrix} h_{1x} \\ h_{1y} \\ h_{1z} \end{bmatrix} = \begin{bmatrix} I_1 \end{bmatrix} \begin{bmatrix} \omega_{1x} \\ \omega_{1y} \\ \omega_{1z} \end{bmatrix}$$

$\bar{\omega}_0$
I_0
$\bar{h}_0$ (body 0 coordinates)
$\bar{\omega}_0$
I_1
$\bar{h}_0$ (body 1 coordinates)

Discussion: None

Segment 13

Purpose: To compute the primed angular momentum of bodies 1, 3 and 4.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	B3MASS	m_3	The mass of body 3
I	EL3(3)	$\bar{\ell}_3$	Vector position of body 3 from the hinge line $\bar{s}_3$
I	R3DOT(3)	$\dot{\bar{r}}_3$	The time derivative of $\bar{r}_3$
0	H3PRIM(3)	$\bar{h}'_3$	The primed angular momentum of body 3
I	B4MASS	m_4	The mass of body 4
I	EL4(3)	$\bar{\ell}_4$	Vector position of body 4 from the hinge line $\bar{s}_4$
I	R4DOT(3)	$\dot{\bar{r}}_4$	The time derivative of $\bar{r}_4$
0	H4PRIM(3)	$\bar{\ }'_4$	The primed angular momentum of body 4
I	H1(3)	$\bar{h}_1$	The angular momentum of body 1
I	H3PRIM(3)	$\bar{h}'_3$	The primed angular momentum of body 3
I	H4PRIM(3)	$\bar{h}'_4$	The primed angular momentum of body 4
I	B2MASS	m_2	The mass of body 2
I	EL2(3)	$\bar{\ell}_2$	Vector position of the movable mass
I	R2DOT(3)	$\dot{\bar{r}}_2$	The time derivative of $\bar{r}_2$
I	B3MASS	m_3	The mass of body 3

I	R3DOT(3)	$\dot{\bar{r}}_3$	The time derivative of $\bar{r}_3$
I	B4MASS	m_4	The mass of body 4
I	R4DOT(3)	$\dot{\bar{r}}_4$	The time derivative of $\bar{r}_4$
I	D13(3)	$\bar{d}_{13}$	Vector distance from the c.m. of body 1 to the hinge line of body 3
I	D14(3)	$\bar{d}_{14}$	Vector distances from the c.m. of body 1 to the hinge line of body 4
0	H1PRIM(3)	$\bar{h}'_1$	The primed angular momentum of body 1

Subroutines required: None

Equations programmed:

$$\xrightarrow[\text{(body 1 coord.)}]{\bar{\ell}_3,\ \dot{\bar{r}}_3} \boxed{\begin{bmatrix} h'_{3x} \\ h'_{2y} \\ h'_{3z} \end{bmatrix} = m_3 \left[\text{CPM}\bar{\ell}_3 \right] \begin{bmatrix} \dot{r}_{3x} \\ \dot{r}_{3y} \\ \dot{r}_{3z} \end{bmatrix}} \xrightarrow{\bar{h}'_3 \text{ (body 1 coords.)}}$$

$$\xrightarrow[\text{(body 1 coord.)}]{\bar{\ell}_4,\ \dot{\bar{r}}_4} \boxed{\begin{bmatrix} h'_{4x} \\ h'_{4y} \\ h'_{4z} \end{bmatrix} = m_4 \left[\text{CPM}\bar{\ell}_4 \right] \begin{bmatrix} \dot{r}_{4x} \\ \dot{r}_{4y} \\ \dot{r}_{4z} \end{bmatrix}} \xrightarrow{\bar{h}'_4 \text{ (body 1 coords.)}}$$

$$\begin{bmatrix} h'_{1x} \\ h'_{1y} \\ h'_{1z} \end{bmatrix} = \begin{bmatrix} h_{1x} + h'_{3x} + h'_{4x} \\ h_{1y} + h'_{3y} + h'_{4y} \\ h_{1z} + h'_{3z} + h'_{4z} \end{bmatrix} - m_2 \left[\mathrm{CPM}(-\bar{\ell}_2) \right] \begin{bmatrix} \dot{r}_{2x} \\ \dot{r}_{2y} \\ \dot{r}_{2z} \end{bmatrix} - m_3 \left[\mathrm{CPM}(-\bar{\ell}_3) \right] \begin{bmatrix} \dot{r}_{3x} \\ \dot{r}_{3y} \\ \dot{r}_{3z} \end{bmatrix}$$

$$- m_4 \left[\mathrm{CPM}(-\bar{d}_{14}) \right] \begin{bmatrix} \dot{r}_{4x} \\ \dot{r}_{4y} \\ \dot{r}_{4z} \end{bmatrix}$$

↑ $\bar{h}_1, \bar{h}'_3, \bar{h}'_4, \bar{\ell}_2, \bar{\ell}_3, \dot{\bar{r}}_2, \dot{\bar{r}}_3, \dot{\bar{r}}_4$

(body 1 coords.)

↓ $\bar{h}'_1$ (body 1 coords.)

Discussion: None

Segment 14

Purpose: To compute the angular momentum of the composite vehicle.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	HO(3)	$\bar{h}_0$	The angular momentum of body 0
I	H1PRIM(3)	$\bar{h}_1'$	The primed angular momentum of body 1
I	D01(3)	$\bar{d}_{01}$	Vector distance from the c.m. of body 0 to the hinge line of body 1
I		A_0	Coordinate transformation from body 1 to body 0
I	B1MASS	m_1	Mass of body 1
I	B2MASS	m_2	Mass of body 2
I	B3MASS	m_3	Mass of body 3
I	B4MASS	m_4	Mass of body 4
I	R1DOT(3)	$\dot{\bar{r}}_1$	Time derivative of $\bar{r}_1$
I	R2DOT(3)	$\dot{\bar{r}}_2$	Time derivative of $\bar{r}_2$
I	R3DOT(3)	$\dot{\bar{r}}_3$	Time derivative of $\bar{r}_3$
I	R4DOT(3)	$\dot{\bar{r}}_4$	Time derivative of $\bar{r}_4$
0	H(3)	$\bar{H}$	The angular momentum of the composite vehicle

Subroutines required: None

Equations programmed:

$$\begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix} = \begin{bmatrix} h_{0x} \\ h_{0y} \\ h_{0z} \end{bmatrix} + \begin{bmatrix} A_0 \end{bmatrix} \begin{bmatrix} h'_{1x} \\ h'_{1y} \\ h'_{1z} \end{bmatrix} + \begin{bmatrix} CPM(\bar{d}_{01}) \end{bmatrix} \begin{bmatrix} A_0 \end{bmatrix} \begin{bmatrix} m_1\dot{r}_{1x} + m_2\dot{r}_{2x} + m_3\dot{r}_{3x} \\ m_1\dot{r}_{1y} + m_2\dot{r}_{2y} + m_3\dot{r}_{3y} \\ m_1\dot{r}_{1z} + m_2\dot{r}_{2z} + m_3\dot{r}_{3z} \end{bmatrix}$$

$\bar{h}_0$ (body 0 coords.)

$\bar{h}'_1$, $\dot{\bar{r}}_1$, $\dot{\bar{r}}_2$, $\dot{\bar{r}}_3$, $\dot{\bar{r}}_4$ (body 1 coords.)

$\bar{H}$ (body 0 coords.)

Discussion: None

Segment 15

Purpose: To compute in body 1 coordinates, the unit vector $\bar{j}$. This is done as a matter of computational convenience and is used in setting up the angular momentum derivatives.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	EL2(3)	$\bar{\ell}_2$	Vector position of the movable mass
I	D13(3)	$\bar{d}_{13}$	Vector distance from the c.m. of body 1 to the hinge line of body 3
I	EL3(3)	$\bar{\ell}_3$	Vector position of body 3
I	D14(3)	$\bar{d}_{14}$	Vector distance from the c.m. of body 1 to the hinge line of body 4
I	EL4(3)	$\bar{\ell}_4$	Vector position of body 4
I	BMASS2	m_2	Mass of body 2
I	BMASS3	m_3	Mass of body 3
I	BMASS4	m_4	Mass of body 4
0	AJ1(3)	$\bar{j}_1$	Unit vector defined in body 1

Subroutines required: None

Equations programmed:

$$\begin{array}{l} \bar{\ell}_2, \bar{\ell}_3, \bar{\ell}_4 \rightarrow \\ m_2, m_3, m_4 \rightarrow \\ \bar{d}_{13}, \bar{d}_{14} \rightarrow \end{array} \boxed{\begin{bmatrix} j_{1x} \\ j_{1y} \\ j_{1z} \end{bmatrix} = m_2 \begin{bmatrix} \ell_{2x} \\ \ell_{2y} \\ \ell_{2z} \end{bmatrix} + m_3 \begin{bmatrix} d_{13x} + \ell_{3x} \\ d_{13y} + \ell_{3y} \\ d_{13z} + \ell_{3z} \end{bmatrix} + m_4 \begin{bmatrix} d_{14x} + \ell_{4x} \\ d_{14y} + \ell_{4y} \\ d_{14z} + \ell_{4z} \end{bmatrix}} \rightarrow \begin{array}{l} j_1 \\ \text{(body 1} \\ \text{coords.)} \end{array}$$

Discussion: None

Segment 16

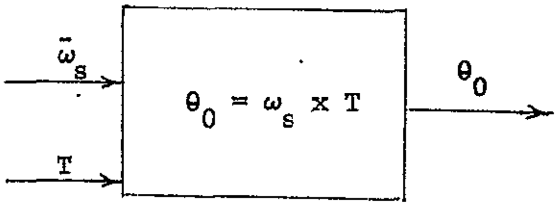
Purpose: To update the orbit angle θ_0 and to compute the external forces and moments.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	TIME	T	The time in seconds
I	Ws	ω_s	The orbital rate
0	THETO	θ_0	The orbital angular position
0	TQ1G(3)	TQ_{1G}	The gravity gradient torques
0	TQ1P(3)	TQ_{1P}	The propulsion torques

Subroutines required: GGRAD, PCON

Equations programmed:



Discussion: None

Segment 17

Purpose: To sum to forces and moments acting on bodies 0 and 1.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	F01(3)	F_{01}	Force acting on body 0 due to gravity gradient
I	F11(3)	F_{11}	Force acting on body 1 due to gravity gradient
I	TQ0G(3)	TQ_{0G}	The torque acting on body 0 due to gravity gradient
I	TQ0P(3)	TQ_{0P}	The torque acting on body 0 due to propulsion forces
I	TQ1G(3)	TQ_{1G}	The torque acting on body 1 due to gravity gradient
I	TQ1P(3)	TQ_{1P}	The torque acting on body 1 due to propulsion forces
Ø	T0EF(3)	T_{0eF}	The summation of the torques acting on body 0
Ø	T1EF(3)	T_{1eF}	The summation of the torques acting on body 1

Subroutines required: None

Equations programmed:

Segment 18

Purpose: To calculate the time derivatives of the angular momentum of the composite vehicle.

Input/output:

I/0	Fortran Name	Math Symbol	Definition
I	WO(3)	$\bar{\omega}_0$	Angular velocity of body 0
I	H(3)	$\bar{H}$	Angular momentum of the composite body
I	TOTMAS	m	Mass of the composite body
I	BOMASS	m_0	Mass of body 0
I	D01(3)	$\bar{d}_{01}$	Vector distance from the c.m. of body 0 to the hinge line of body 1
I	AJ1(3)	$\bar{j}_1$	A vector, defined in body 1 coordinates used only for computational convenience
I	F0(3)	$\bar{F}_0$	Summation of the forces acting on body 0
		A_0	Coordinate transformation from body 1 to body 0
I	T0EF(3)	$\bar{T}_{0EF}$	Summation of the torques acting on body 0
I	T1EF(3)	$\bar{T}_{1EF}$	Summation of the torques acting on body 1
0	HDOT(3)	$\frac{dH}{dt}$	The time derivative of H

Subroutines required: None

Equations programmed:

$$\begin{bmatrix} \frac{dH_x}{dt} \\ \frac{dH_y}{dt} \\ \frac{dH_z}{dt} \end{bmatrix} + \text{CPM}(\bar{\ }_0) \begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix} = \frac{1}{m} \left[\text{CPM} \left\{ (m_0 - m) \begin{bmatrix} d_{01x} \\ d_{01y} \\ d_{01z} \end{bmatrix} - A_0 \begin{bmatrix} j_{1x} \\ j_{1y} \\ j_{1z} \end{bmatrix} \right\} \right] \begin{bmatrix} F_{0x} \\ F_{0y} \\ F_{0z} \end{bmatrix}$$

$$+ \frac{1}{m} A_0 \quad \text{CPM} \left\{ m_0 \begin{bmatrix} A_1 \end{bmatrix} \begin{bmatrix} d_{01x} \\ d_{01y} \\ d_{01z} \end{bmatrix} - \begin{bmatrix} j_{1x} \\ j_{1y} \\ j_{1z} \end{bmatrix} \right\} \cdot \begin{bmatrix} F_{1x} \\ F_{1y} \\ F_{1z} \end{bmatrix} + \begin{bmatrix} T_{0EFx} \\ T_{0EFy} \\ T_{0EFz} \end{bmatrix} + A_0 \begin{bmatrix} T_{1EFx} \\ T_{1EFy} \\ T_{1EFz} \end{bmatrix}$$

$\bar{\omega}_0$, $\bar{H}$, $\bar{F}_0$, $\bar{T}_{0EF}$, $\bar{d}_{01}$ (body 0 coords.)

$\bar{j}_1$, F_1, T_{1EF} (body 1 coords.)

A_1, A_0

$\frac{d\bar{H}}{dt}$ (body 0 coords.)

INTEGRATION SUBROUTINE

Discussion: The following vector equation is programmed in this segment:

$$\frac{d\bar{H}}{dt} + \bar{\omega}_0 \times \bar{H} = \left\{ -(m-m_0)\bar{d}_{01} - \bar{j}_2 \right\} \times \frac{\bar{F}_0}{m} + \left\{ m_0 \bar{d}_{01} - \bar{j}_1 \right\} \times \frac{\bar{F}_0}{m} + \bar{T}_{0EF} + \bar{T}_{1EF}$$

Segment 19

Purpose: To calculate the time derivative of $\bar{h}'_1$ about the unconstrained axis $\underline{\bar{x}}_1$.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	W1(3)	$\bar{\omega}_1$	Angular velocity of body 1
I	H1PRIM(3)	$\bar{h}_1$	The primed angular momentum of body 1
I	B2MASS	m_2	Mass of body 2
I	EL2DOT(3)	$\dot{\bar{\ell}}_2$	Time derivative of $\bar{\ell}_2$
I	R2DOT(3)	$\dot{\bar{r}}_2$	Time derivative of $\bar{r}_2$
I	B3MASS	m_3	Mass of body 3
I	D13DOT(3)	$\dot{\bar{d}}_{13}$	Time derivative of $\bar{d}_{13}$
I	EL3DOT(3)	$\dot{\bar{\ell}}_3$	Time derivative of $\bar{\ell}_3$
I	R3DOT(3)	$\dot{\bar{r}}_3$	Time derivative of $\bar{r}_3$
I	B4MASS	m_4	Mass of body 4
I	D14DOT(3)	$\dot{\bar{d}}_{14}$	Time derivative of $\bar{d}_{14}$
I	EL4DOT(3)	$\dot{\bar{\ell}}_4$	Time derivative of $\bar{\ell}_4$
I	R4DOT(3)	$\dot{\bar{r}}_4$	Time derivative of $\bar{r}_4$
I	AJ1(3)	$\bar{j}_1$	A vector, defined in body 1 coordinates, used for computational convenience
I	T1EF(3)	$\bar{T}_{1EF}$	Summation of the torques acting on body 1
I	T01	$\bar{T}_{0\rightarrow1}$	The torque acting between bodies 0 and 1
0	H1PDOT(1)	$\dot{h}'_{1x}$	Time derivative of h'_{1x}

Subroutines required: TORK01

Equations programmed:

Unconstrained component along $\bar{\underline{x}}_0 = \bar{\underline{x}}_1$ (axis)

$$\boxed{\begin{aligned}\frac{dh'_{1x}}{dt} + \omega_{1y}h'_{1z} - \omega_{1z}h'_{1y} = {} & \dot{r}_{1y}\,(-m_2\ell_{2z} - m_3\,(\dot{d}_{13z} + \ell_{3z}) - m_4\,(\dot{d}_{14z} + \dot{\ell}_{4z}) \\ & - \dot{r}_{1z}\,(-m_2\ell_{2y} - m_3\,(\dot{d}_{13y} + \dot{\ell}_{3y}) - m_4\,(\dot{d}_{14y} + \dot{\ell}_{4y}) \\ & + j_{1y}\,(\frac{F_{0y}}{m}\sin\theta_1 - \frac{F_{0z}}{m}\cos\theta_1) + j_{1z}\,(\frac{F_{0y}}{m}\cos\theta_1 + \frac{F_{0z}}{m}\sin\theta_1) \\ & + j_{1z}\frac{F_{1y}}{m} - j_{1y}\frac{F_{1z}}{m} + T_{1EFx} + T_{(0\rightarrow 1)x}\end{aligned}}$$

$\bar{h}'_1$, $\bar{\omega}_1$, $\dot{\bar{r}}_1$, $\dot{\bar{r}}_2$, $\dot{\bar{r}}_3$, $\dot{\bar{r}}_4$, $\dot{\bar{\ell}}_2$, $\dot{\bar{\ell}}_3$, $\dot{\bar{\ell}}_4$, $\bar{j}_1$, $\bar{F}_1$, $\bar{T}_{1EF}$, $\bar{T}_{(0\rightarrow 1)_x}$
(body 1 coords.)

$\bar{F}_0$ (body 0 coords.)

h'_{1x} (body 1 coords.)

Discussion: None

Segment 20

Purpose: To calculate the time derivatives of the primed angular momenta about the unconstrained axes $\bar{s}_3$ and $\bar{s}_4$.

Input/output:

I/O	Fortran Name	Math Symbol	Definition
I	H3PRIM(3)	$\bar{h}'_3$	The primed angular momentum of body 3
I	S3(3)	$\bar{s}_3$	The hinge line of body 3
I	W1(3)	$\bar{\omega}_1$	The angular velocity of body 1
I	B3MASS	m_3	Mass of body 3
I	EL3DOT(3)	$\dot{\bar{\ell}}_3$	Time derivative of $\bar{\ell}_3$
I	R3DOT(3)	$\dot{\bar{r}}_3$	Time derivative of $\bar{r}_3$
I	EL3(3)	$\bar{\ell}_3$	Vector position of body 3 from the hinge line $\bar{s}_3$
I	B4MASS	m_3	Mass of body 4
I	TOTMAS	m	Mass of composite vehicle
I	F0(3)	$\bar{F}_0$	Summation of the forces acting on body 0
I	F1(3)	$\bar{F}_1$	Summation of the forces acting on body 1
I	T13	$\bar{T}_{1\rightarrow 3}$	Torque acting between bodies 1 and 3
Ø	G3DOT	$\frac{d}{dt}(\bar{h}'_3\cdot\bar{s}_3)$	Time derivative of the unconstrained component of $\bar{h}'_3$
I	H4PRIM(3)	$\bar{h}'_4$	The primed angular momentum of body 4

I	S4(3)	$\bar{s}_4$	The hinge line of body 4
I	EL4DOT(3)	$\dot{\bar{\ell}}_4$	Time derivative of $\bar{\ell}_4$
I	R4DOT(3)	$\dot{\bar{r}}_4$	Time derivative of $\bar{r}_4$
I	EL4(3)	$\bar{\ell}_4$	Vector position of body 4 from the hinge line $\bar{s}_4$
I	T14	$\bar{T}_{1\rightarrow 4}$	Torque acting between bodies 1 and 4
0	G4DOT	$\frac{d}{dt}(\bar{h}_4' \cdot \bar{s}_4)$	Time derivative of the unconstrained component of $\bar{h}_4'$

Subroutines required: TORK13, TORK14

Equations programmed:

$$\frac{d}{dt}(\bar{h}'_3\cdot\bar{s}_3) + s_{3y}(\omega_{1z}h'_{3x} - \omega_{1x}h'_{3z}) + s_{3z}(\omega_{1x}h'_{3y} - \omega_{1y}h'_{3x}) =$$

$$m_3 s_{3y}(\dot{\ell}_{3z}\dot{r}_{3x} - \dot{\ell}_{3x}\dot{r}_{3z}) + m_3 s_{3z}(\dot{\ell}_{3x}\dot{r}_{3y} - \dot{\ell}_{3y}\dot{r}_{3x})$$

$$+\frac{m_3}{m} s_{3y}\ \ell_{3z}(F_{0x} + F_{1x}) - \ell_{3x}(-F_{0y}\sin\theta_1 + F_{02}\cos\theta_1 + F_{1z})$$

$$+\frac{m_3}{m} s_{3z}\ \ell_{3z}(F_{0y}\cos\theta_1 + F_{0z}\sin\theta_1 + F_{1y}) - \ell_{3y}(F_{0x} + F_{1x})$$

$$\frac{d(\bar{h}'_4\cdot\bar{s}_4}{dt} + s_{4y}(\omega_{1z}h'_{4x} - \omega_{1x}h'_{4z}) + s_{4z}(\omega_{1x}h'_{4y} - \omega_{1y}h'_{4x})$$

$$= m_4 s_{4y}(\dot{\ell}_{4z}\dot{r}_{4x} - \dot{\ell}_{4x}\dot{r}_{4z}) + m_4 s_{4y})\dot{\ell}_{4x}\dot{r}_{4y} - \dot{\ell}_{4y}r_{4x})$$

$$+\frac{m_4}{m} s_{4y}\ \ell_{4z}(F_{0x} + F_{1x}) - \ell_{4x}(-F_{0y}\sin\theta_1 + F_{0z}\cos\theta_1 + F_{1z})$$

$$+\frac{m_4}{m} s_{4z}\ \ell_{4x}(F_{0y}\cos\theta_1 + F_{0z}\sin\theta_1 + F_{1y}) - \ell_{4y}(F_{0x} + F_{1x}) \quad + T_{1\to 4}$$

$\bar{\omega}_1, \bar{h}'_3, \bar{h}'_4, \bar{\ell}_3, \bar{\ell}_4, \dot{\bar{\ell}}_3, \dot{\bar{\ell}}_4$

$\dot{\bar{r}}_3, \dot{\bar{r}}_4, \bar{F}_1, \bar{T}_1\ {}_3, \bar{T}_{1\to 4}$

(body 1 coords.)

$\bar{F}_0$ (body 0 coords.)

$$\frac{d}{dt}(\bar{h}'_3\cdot\bar{s}_3) = \frac{d}{dt}(h'_{3y}s_{3y} + h'_{3z}s_{3z}$$

$$\frac{d}{dt}(\bar{h}'_4\cdot\bar{s}_4) = \frac{d}{dt}(h'_{4y}s_{4y} + h'_{4z}s_{4z})$$

(body 1 coords.)

Discussion: None