(NASA-CR-160356) NUMERICAL SIMULATION OF DYNAMICS OF BRUSHLESS DC MOTORS FOR AEROSPACE AND OTHER APPLICATIONS. VOLUME 2: USER'S GUIDE TO COMPUTER EMA MODEL Final Report (Virginia Polytechnic Inst. and State G3/33 N80-14340 Unclas 46338

NASA CR-160356

Final Report on Contract No. NAS 9-15091

NUMERICAL SIMULATION OF DYNAMICS OF BRUSHLESS DC MOTORS FOR AEROSPACE AND OTHER APPLICATIONS

VOLUME (II) USER'S GUIDE TO COMPUTER EMA MODEL

Submitted to NASA-Johnson Space Center

Control Systems Development Division

Houston, Texas 77058

by

Principal Investigator

N. A. O. Demerdash

N. A. O. Demerdash

and

Research Associate

T. W. Nehl

T. W. Nehl

Department of Electrical Engineering

Virginia Polytechnic Institute and State University

Blacksburg, VA 24061

TABLE OF CONTENTS

1.0 INTRODUCTION

This report contains a description and user's guide of the computer program developed at VPI&SU under the NASA contract NAS9-15091. The major goal of this contract was the development of a FORTRAN computer program to simulate the dynamics of the NASA-DELCO Electromechanical Actuator for aerospace applications. The program developed under this contract had to meet the following criteria:

1. Adequate accounting of the effects of the stator phase currents on the permanent magnets of the rotor.

2. Reasonably accurate simulation of the voltage and current waveforms present in the power conditioner network during the motoring, regenerative braking, and plugging modes of operation. Such accuracy requires;

    a) Accounting for the major component nonlinearities such as the coupling or chopper inductor saturation and the diode and transistor nonlinearities,

    and b) inclusion of the machine nonlinearities (see item 1 above).

3. Reasonably accurate simulation of the overall EMA servo loop during a step response. This requires the inclusion of all major nonlinearities such as machine velocity saturation, machine torque limiting, etc.

These requirements have all been met by the computer model developed under this contract. The derivation of this model is developed in detail in the first volume of this report.

2.0 MODEL DEVELOPMENT

The four channel EM actuator shown in Figure (2-1) is usually operated with two machines or channels active while the other machines are braked. Therefore, one needs to consider only two channels at a time in the modeling process. These two channels do not operate independently of each other because of the velocity correction controller which keeps both machines running at approximately the same speed.

Since both channels operate under very similar conditions it was decided to model the one and two machine modes based upon the single channel mechanization diagram shown in Figure (2-2). Under this assumption the velocity correction signal shown in this diagram would always be zero. Figure (2-3) shows essentially the same mechanization diagram with the addition of the blocks representing the rotating masses. The differentiation between one and two channel operation is provided by the ficticious gear ratio NX. When only one machine is operating NX equals 2 since the machine has to rotate through twice the angular displacement when compared with the normal two machine mode. With two machines operating NX is reduced to 1.

Closer inspection of this diagram reveals that there are essentially three basic components to this one and two channel model. They are:

1. The power conditioner and machine
2. The velocity correction loop
3. The position correction-mechanical loop

Notice that these components have been listed in order of ascending time constants. A schematic of the power conditioner and machine network (item 1) is given in Figure (2-4). The electrical model is shown in Figure (2-5). Figure (2-6) displays the various components of the EMA which make up the system of rotating masses.

The models for each of the EMA components are derived in the standard state space form symbolized by

$$\dot{\underline{x}} = \underline{\underline{A}}\underline{x} + \underline{\underline{B}}\underline{u}$$

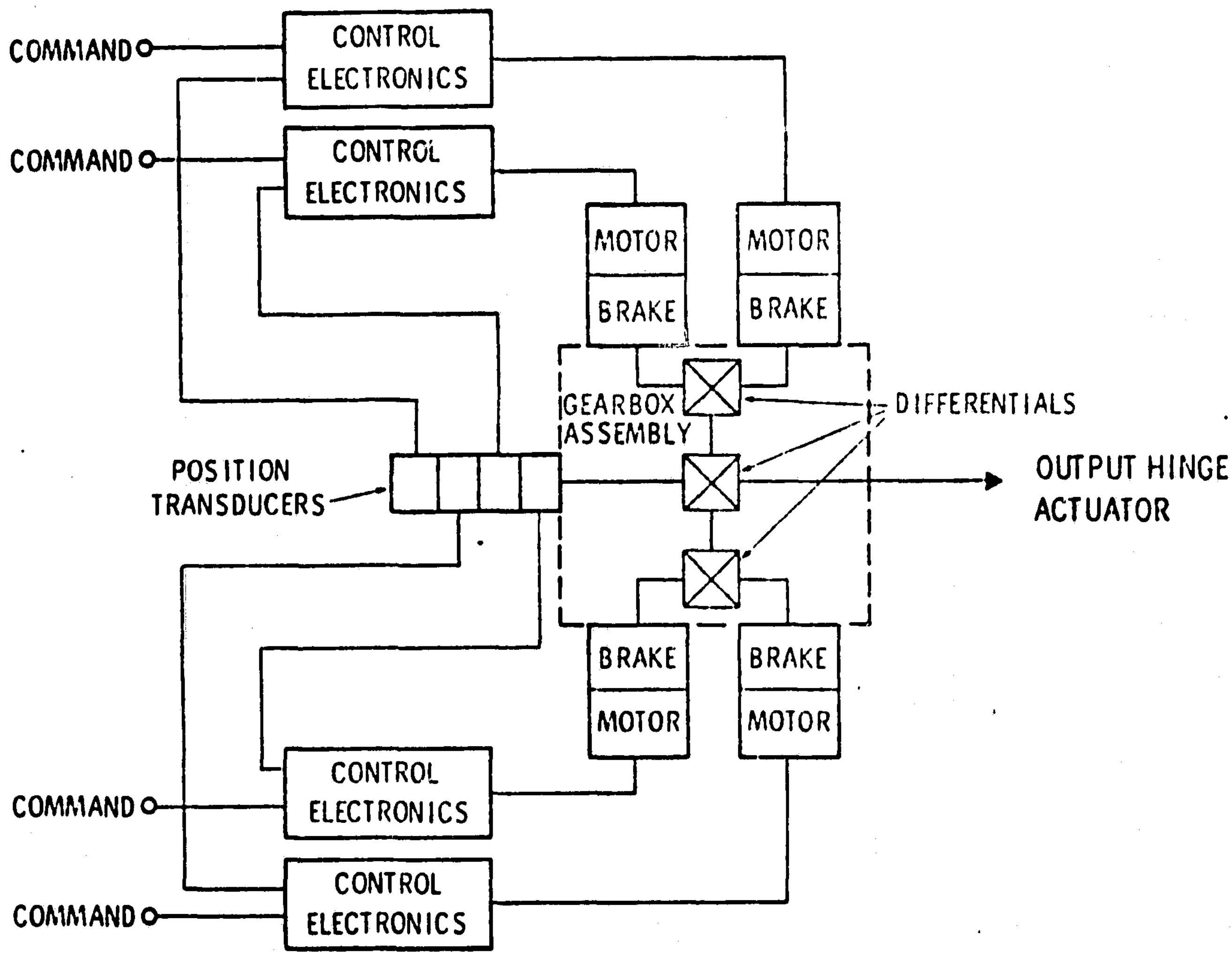
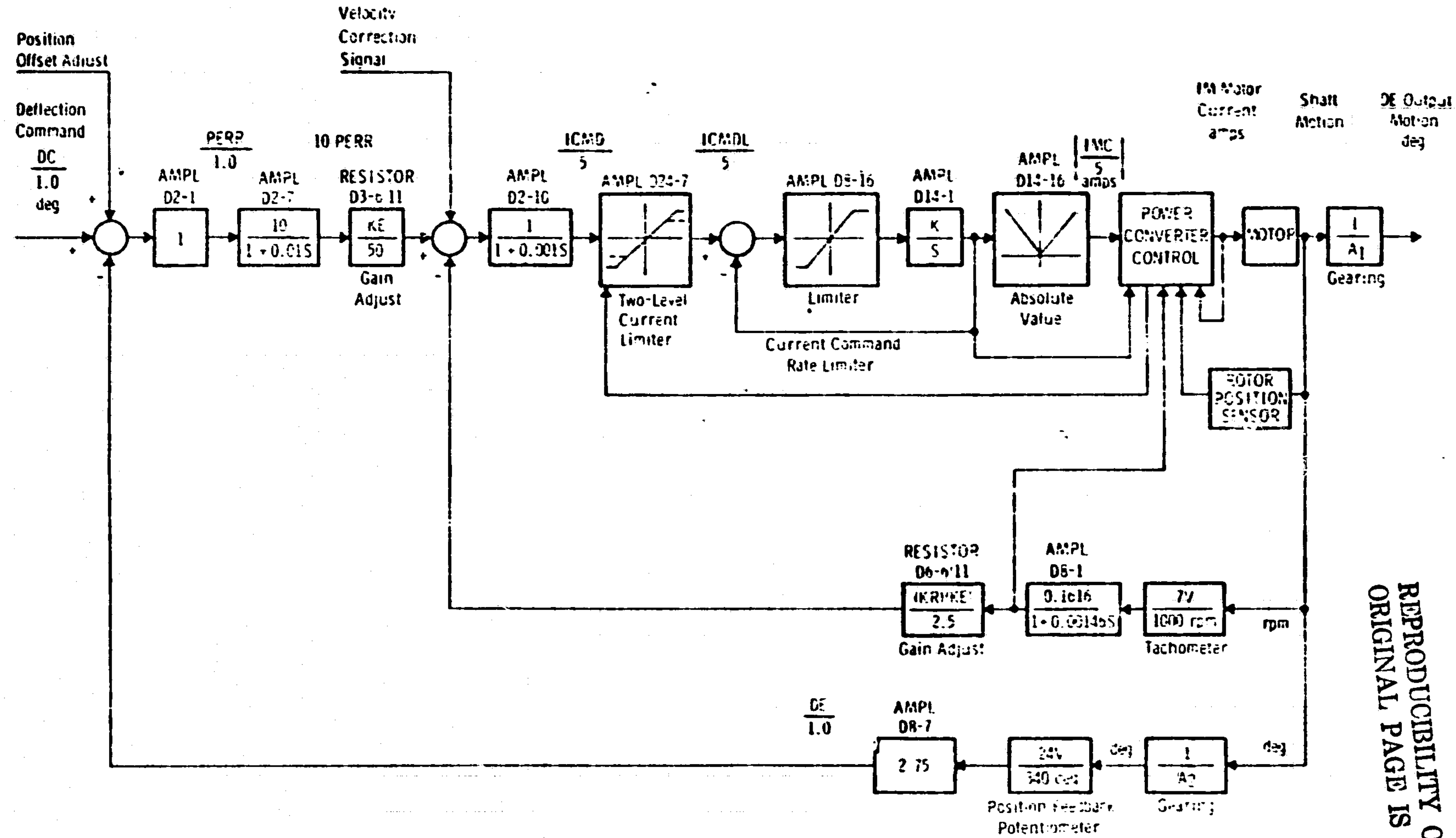
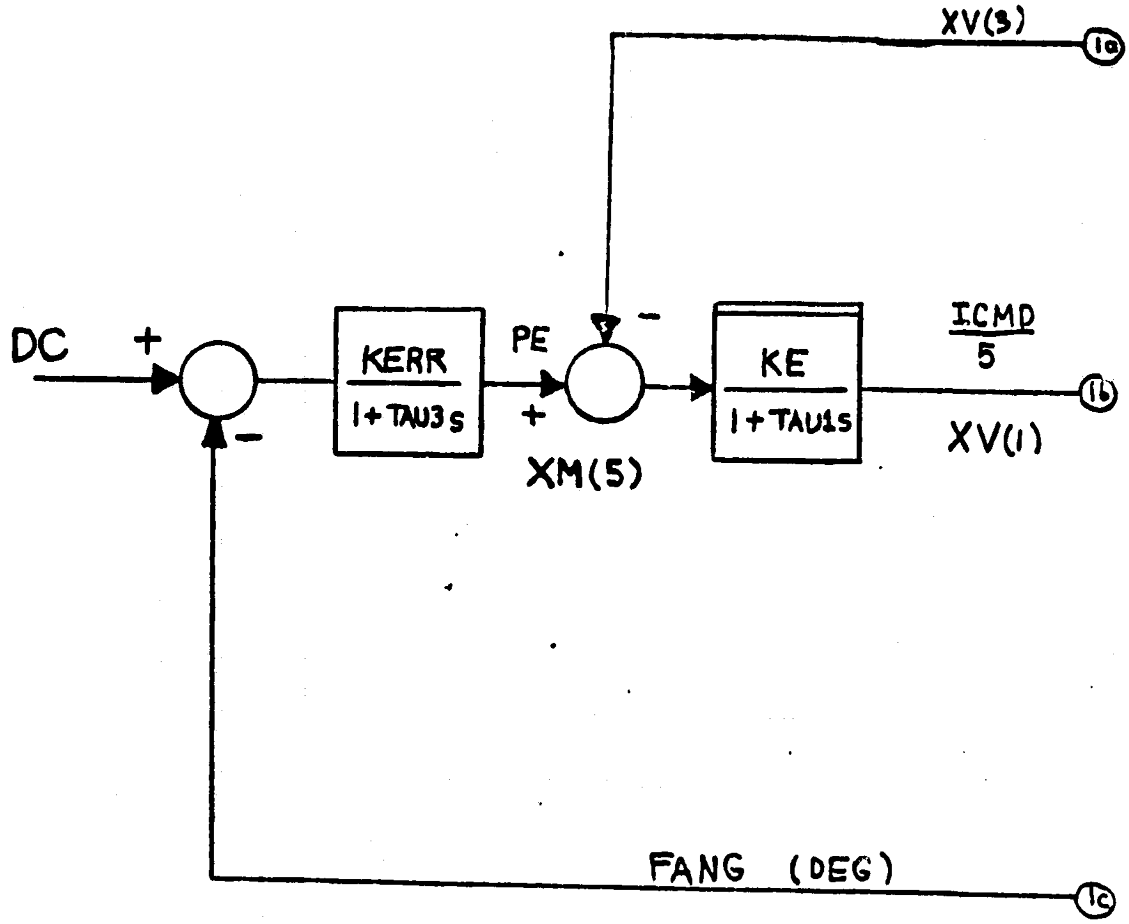


Figure 2-1. EMA Block Diagram



REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

Figure 2-2 EMA Mechanization Diagram



VELOCITY AND POSITION LOOP PARAMETERS

KP = 57.29577951 [deg/rad]

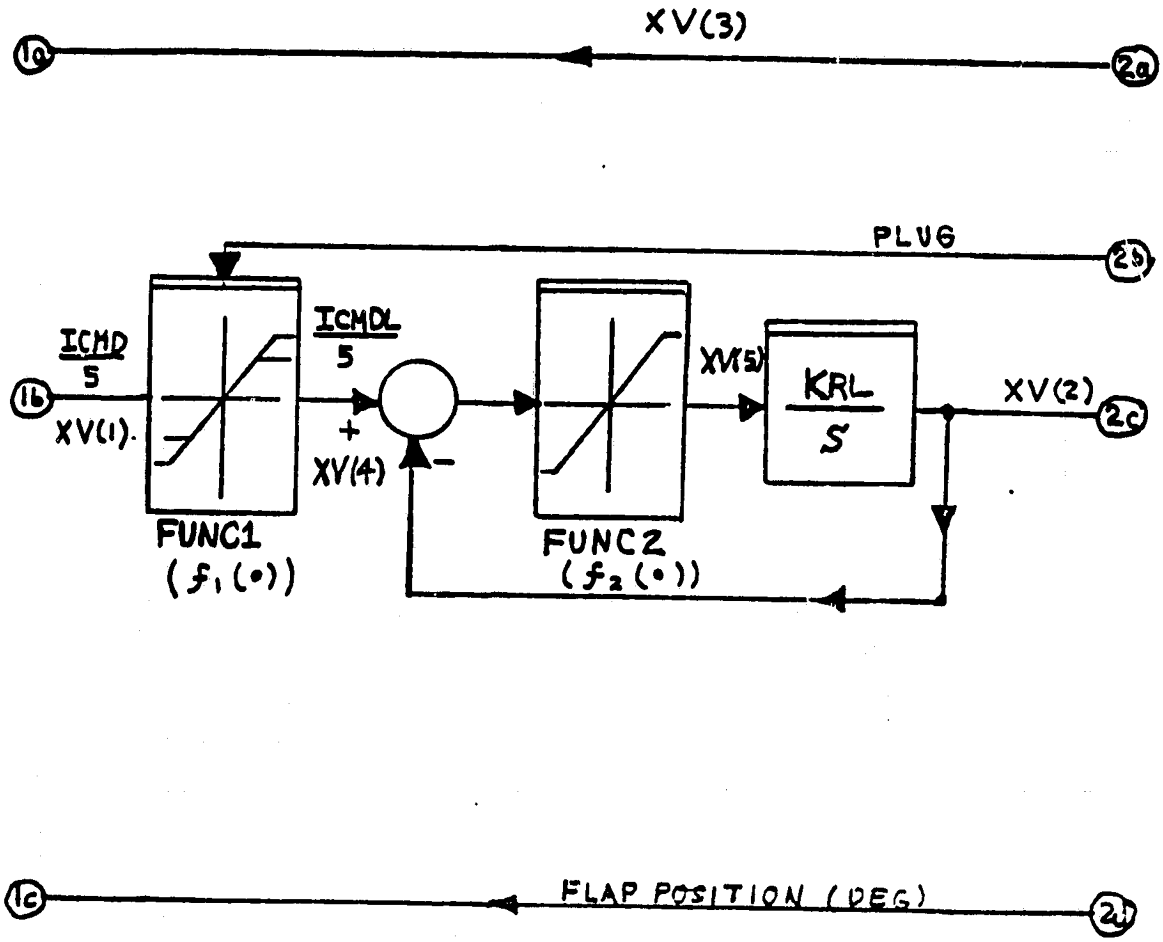
KE = 1. [Dimensionless]

KERR·KE = 85.1 [Amps/deg]

KV = 0.066845 [Amps/rad/S]

KRL = 379

Figure 2-3. Block Diagram of the One and Two Channel EMA Model



MECHANICAL LOOP PARAMETERS

KACT = 100,000 [in-lb/rad] = 11297.92 [NT-m/rad]

KF = 8.6×10^5 [in-lb/rad] = 97162.11 [NT-m/rad]

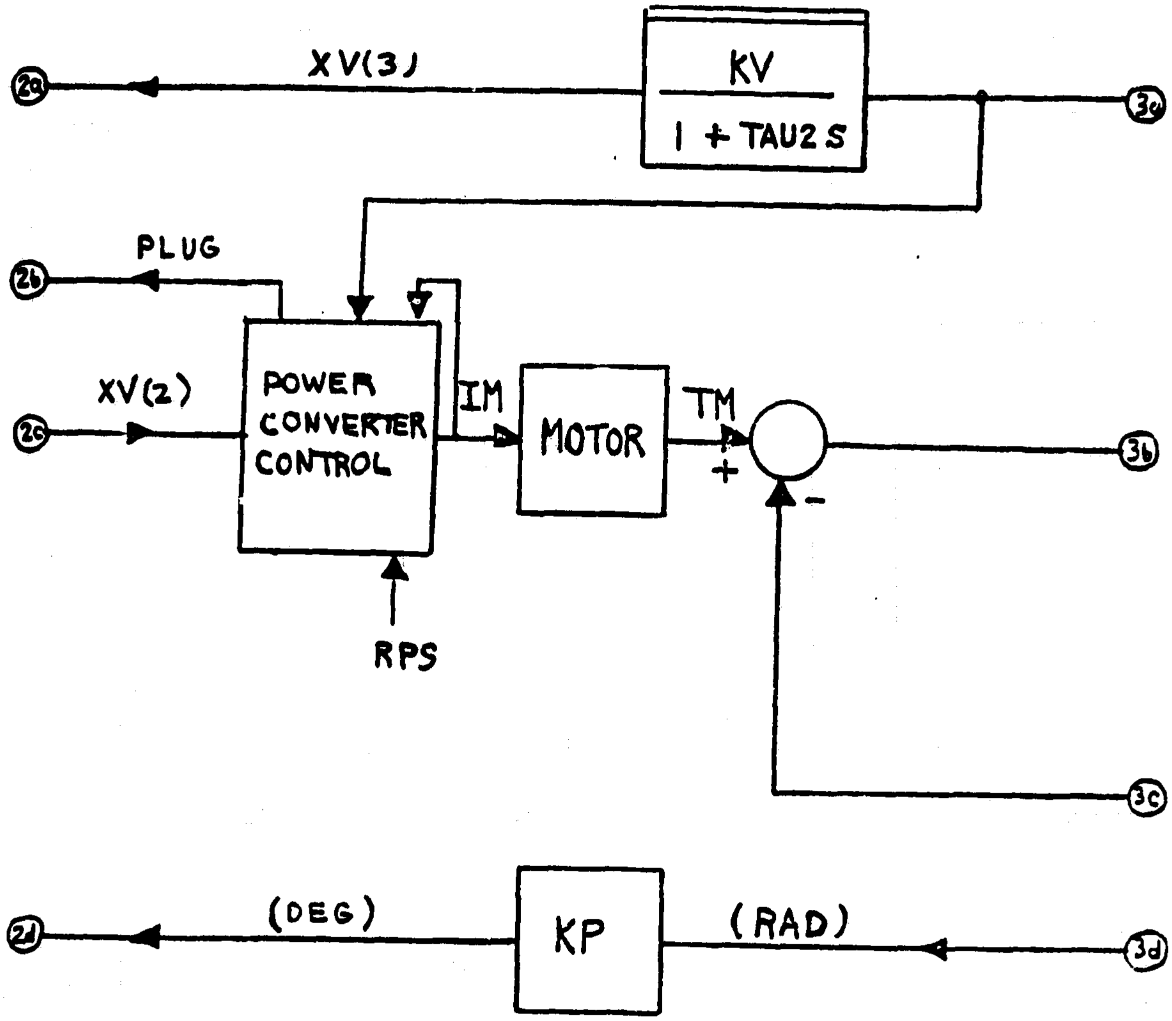
JM = .00842 [in-lb-s^2] = .00095126 [Kg-m^2]

JF = 96.46 [in-lb-s^2] = 10.8977 [Kg-m^2]

BF = 1821.6 [in-lb/rad/s] = 205.80291 [NT-m/rad/s]

N1 = 3.75   N2 = 1.5   N3 = 238.71

Figure 2-3. (cont)



CONVERSION FACTORS (ENGLISH TO MKS)

$T\,[NT\text{-}m] = 0.1129792\ T\,[in\text{-}lb]$

$B\,[NT\text{-}m/rad/s] = 0.1129792\ B\,[in\text{-}lb/rad/s]$

$K\,[NT\text{-}m/rad] = 0.1129792\ K\,[in\text{-}lb/rad]$

$J\,[Kg\text{-}m^2] = 0.1129792\ J\,[in\text{-}lb\text{-}s^2]$

Figure 2-3. (cont)

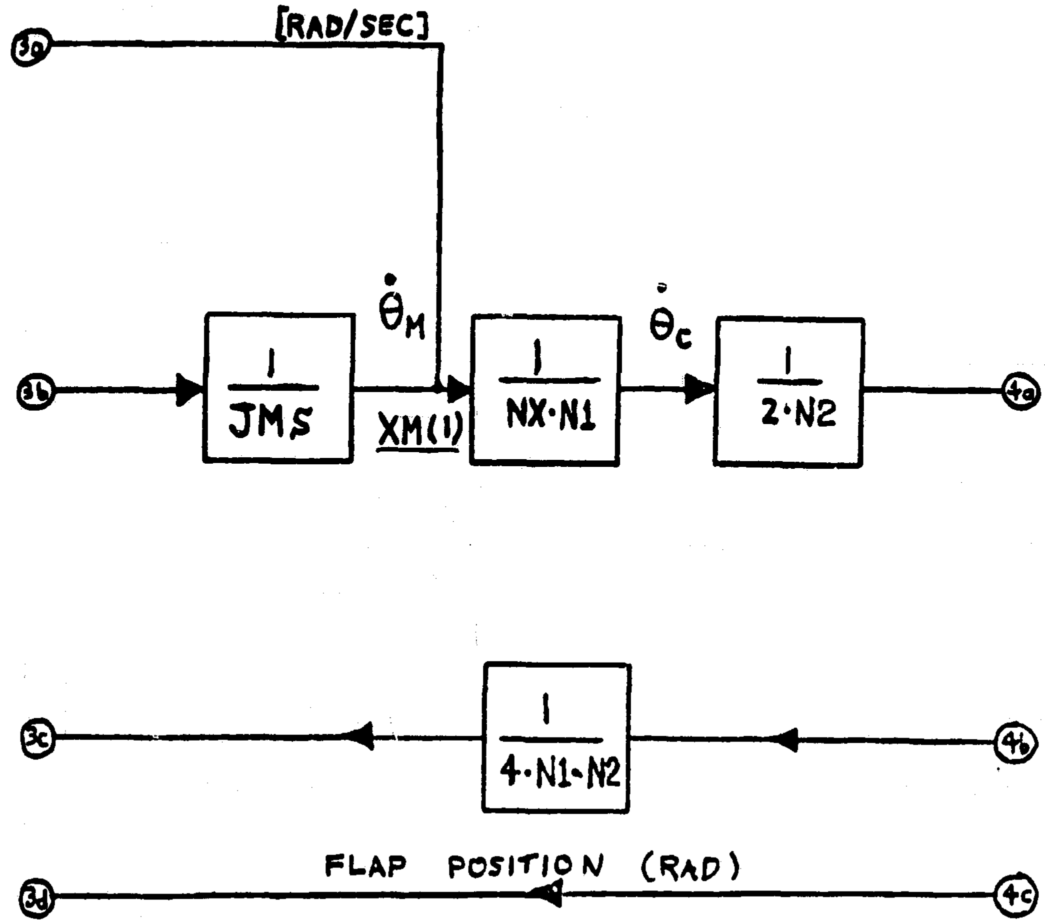


Figure 2-3. (cont)

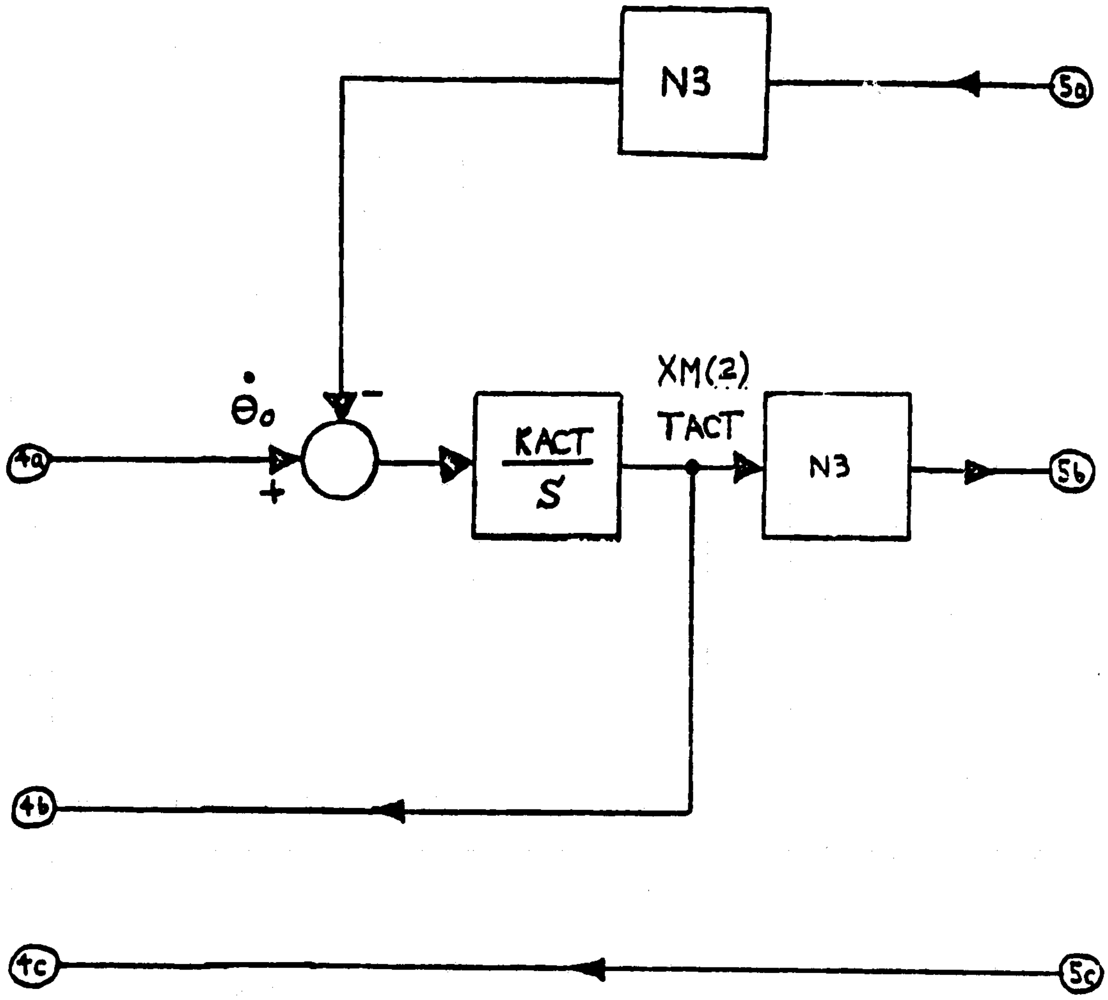


Figure 2-3. (cont)

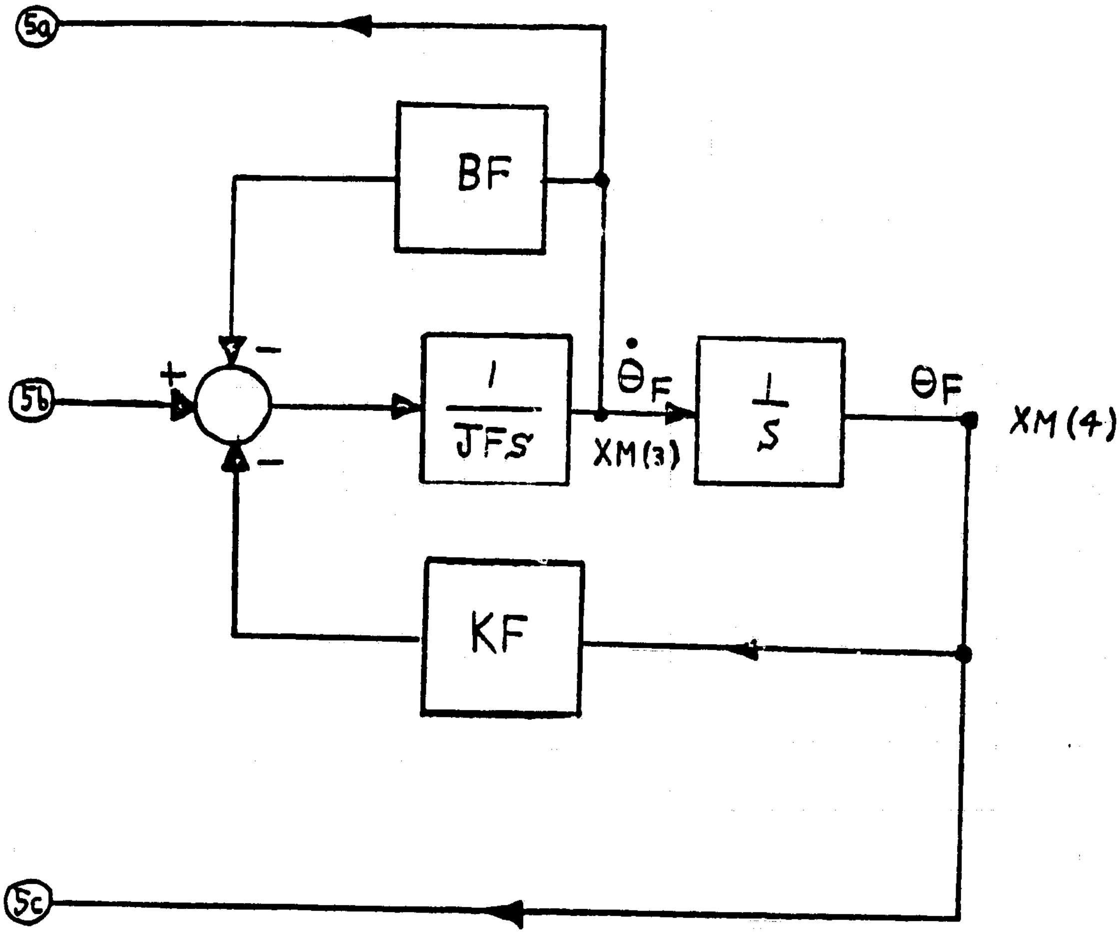


Figure 2-3. (cont)

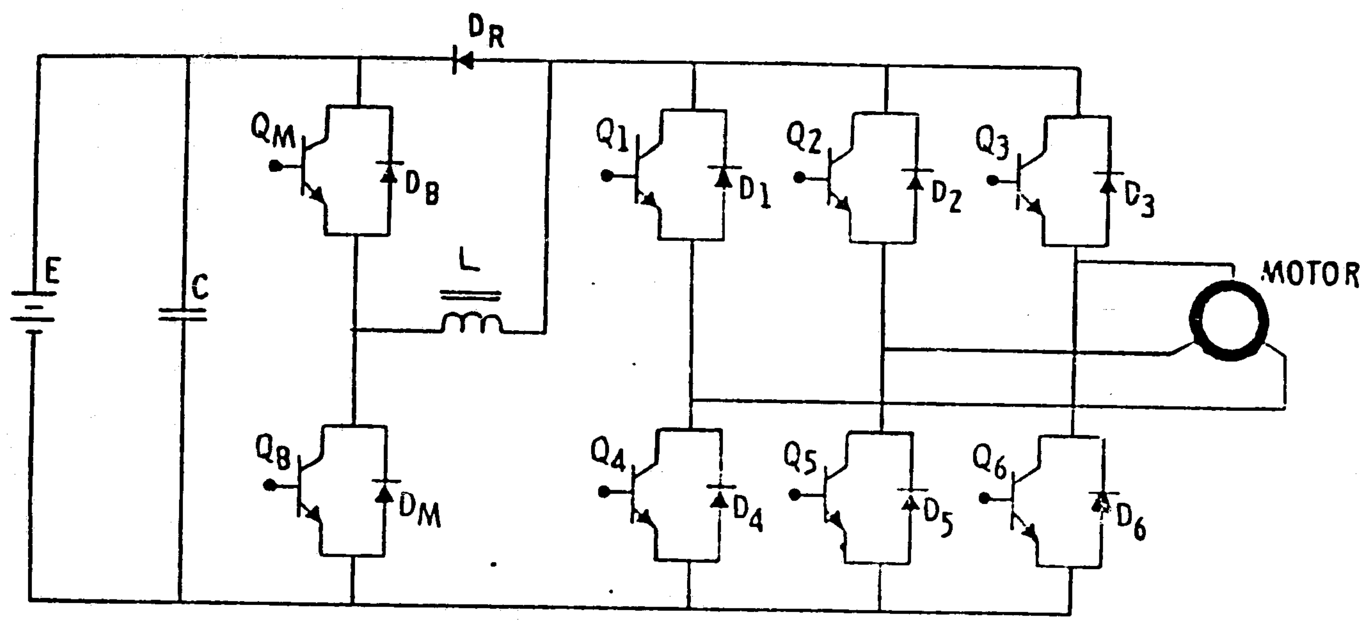


Figure 2-4. Machine and Power Conditioner schematic

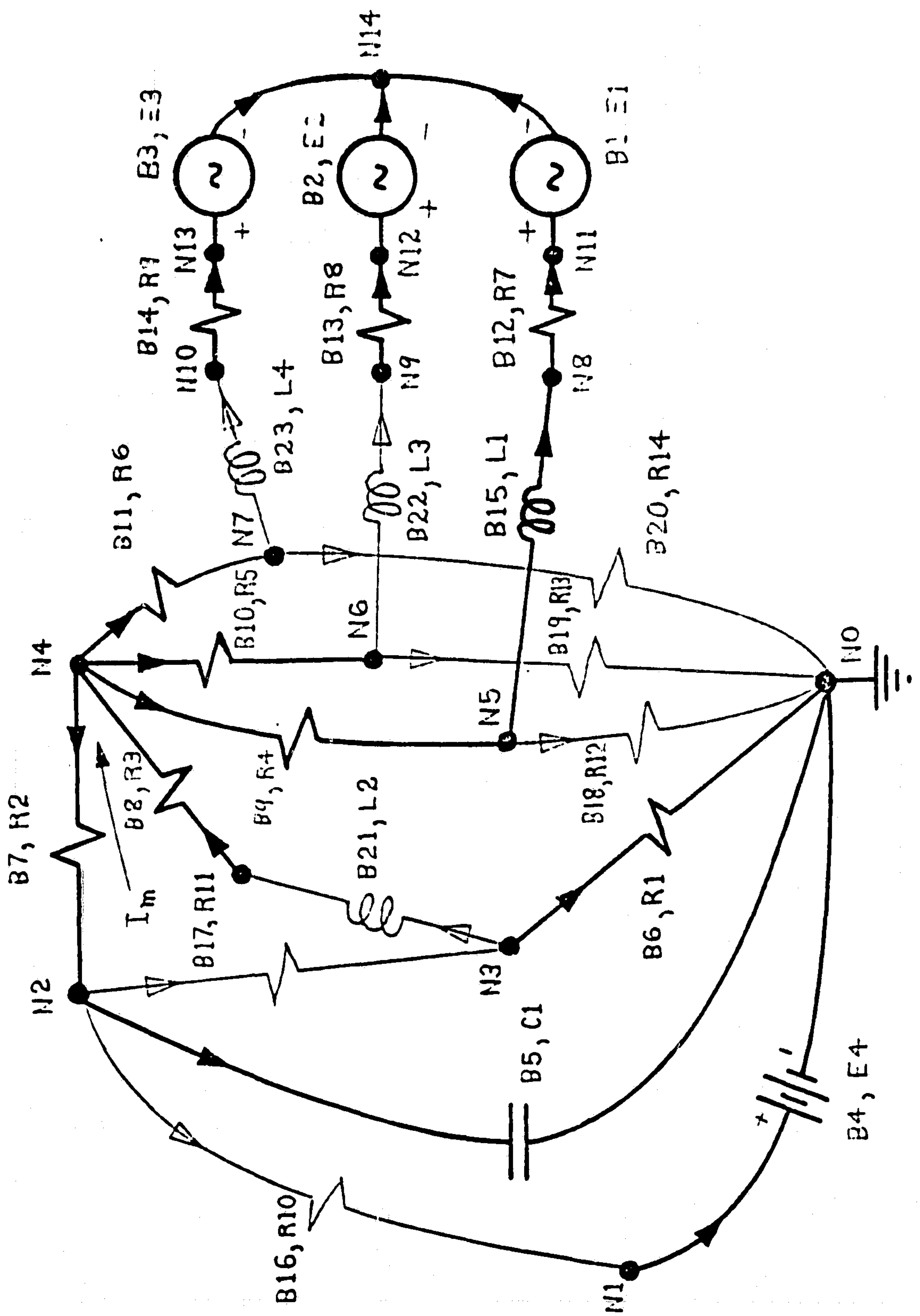


Figure 2-5. Machine - Power Conditioner Model

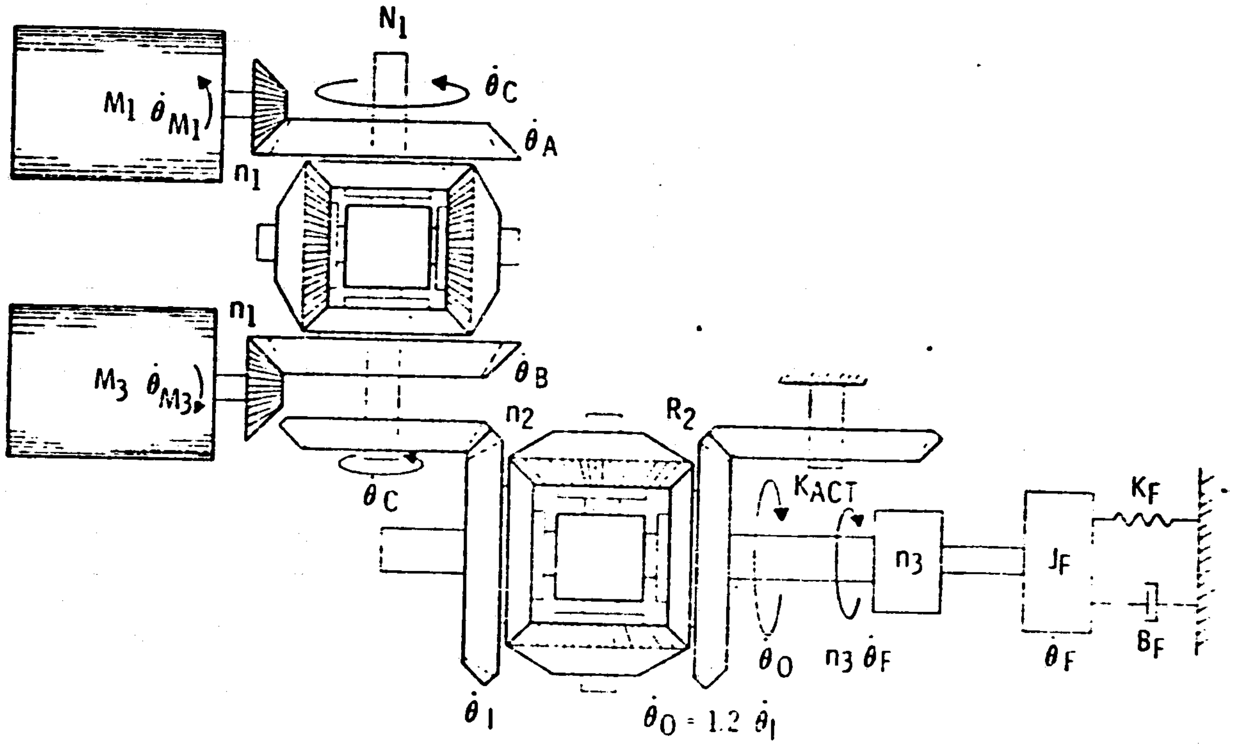


Figure 2-6. The Rotating Masses of the Delco EMA

POWER CONDITIONER AND MACHINE NETWORK STATE EQUATIONS

The state equations of the power conditioner and machine network can be written as follows:

$$\dot{\underline{x}}_n = \underline{A}_n \underline{x}_n + \underline{B}_n \underline{u}_n$$

where

$$\underline{A}_n = \underline{F}_n^{-1} \cdot \underline{G}_n$$

$$\underline{B}_n = \underline{F}_n^{-1} \cdot \underline{H}_n$$

These matrices and vectors are defined as follows:

$$\underline{x}_n = \begin{bmatrix} VB5 \\ IB21 \\ IB22 \\ IB23 \end{bmatrix} \qquad \dot{\underline{x}}_n = \begin{bmatrix} \dot{VB5} \\ \dot{IB21} \\ \dot{IB22} \\ \dot{IB23} \end{bmatrix} \qquad \underline{u} = \begin{bmatrix} E1 \\ E2 \\ E3 \\ E4 \end{bmatrix}$$

$$\underline{F}_n^{-1} = \begin{bmatrix} 1/C1 & 0 & 0 & 0 \\ 0 & 1/L1 & 0 & 0 \\ 0 & 0 & \dfrac{(L1+L4)}{L1L4+L3L1+L3L4} & \dfrac{-L1}{L1L4+L3L1+L3L4} \\ 0 & 0 & \dfrac{-L1}{L1L4+L3L1+L3L4} & \dfrac{L1+L3}{L1L4+L3L1+L3L4} \end{bmatrix}$$

$$
\underset{\sim}{G}_n = \left[\begin{array}{|c|c|c|c|}
\hline
\begin{array}{c} \frac{-1}{R10} \quad \frac{-1}{R1+R10} \\ -(C11+C21+C31) \end{array} & \begin{array}{c} 1 - \frac{R1}{R1+R11} \\ -(C12+C22+C32) \end{array} & -(C13+C23+C33) & -(C14+C24+C34) \\
\hline
\begin{array}{c} -1 + \frac{R1}{R1+R11} \\ +R2(C11+C21+C31) \end{array} & \begin{array}{c} \frac{R1R1}{R1+R11} \\ -(R1+R2+R3) \\ +R2(C12+C22+C32) \end{array} & R2(C13+C23+C33) & R2(C14+C24+C34) \\
\hline
C11R4-C21R5 & C12R4-C22R5 & \begin{array}{c} C13R4-C23R5 \\ -(R4+R5+R7+R8) \end{array} & \begin{array}{c} -(R4+R7) \\ C14R4-C24R5 \end{array} \\
\hline
C11R4-C31R6 & C12R4-C32R6 & \begin{array}{c} -(R4+R7) \\ +C13R4-C33R6 \end{array} & \begin{array}{c} C14R4-C34R6 \\ -(R4+R6+R7+R9) \end{array} \\
\hline
\end{array}\right]
$$

$$\underline{H}_n = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{R10} \\ 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \end{bmatrix}$$

The coefficients C11, C12, ··· C34 in matrix $[\underline{G}_n]$ relate the branch currents IB18, IB19, and IB20 to the state variables XSV as follows:

$$\begin{bmatrix} IB18 \\ IB19 \\ IB20 \end{bmatrix} = \begin{bmatrix} C11 & C12 & C13 & C14 \\ C21 & C22 & C23 & C24 \\ C31 & C32 & C33 & C34 \end{bmatrix} \cdot \begin{bmatrix} VB5 \\ IB21 \\ IB22 \\ IB23 \end{bmatrix}$$

VELOCITY LOOP STATE EQUATIONS

$$\begin{bmatrix} \dot{XV}(1) \\ \dot{XV}(2) \\ \dot{XV}(3) \\ \dot{XV}(4) \\ \dot{XV}(5) \end{bmatrix} = \begin{bmatrix} \frac{-1}{TAU1} & 0 & \frac{-KE}{TAU1} & 0 & 0 \\ 0 & 0 & 0 & 0 & KRL \\ 0 & 0 & \frac{-1}{TAU2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} XV(1) \\ XV(2) \\ XV(3) \\ XV(4) \\ XV(5) \end{bmatrix}$$

$$\begin{bmatrix} \frac{KE}{TAU1} & 0 \\ 0 & 0 \\ 0 & \frac{KV}{TAU2} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} PE \\ RVEL \end{bmatrix}$$

or in symbolic form:

$$\dot{\underline{x}}_v = \underline{A}_v \, \underline{x}_v + \underline{B}_v \, \underline{u}_v$$

MECHANICAL-POSITION LOOP STATE EQUATIONS

$$
\begin{bmatrix} \dot{XM}(1) \\ \dot{XM}(2) \\ \dot{XM}(3) \\ \dot{XM}(4) \\ \dot{XM}(5) \end{bmatrix}
=
\begin{bmatrix}
0 & \frac{-1}{4N1N2JM} & 0 & 0 & 0 \\
\frac{KACT}{2N1N2NX} & 0 & -N3\ KACT & 0 & 0 \\
0 & \frac{N3}{JF} & -\frac{BF}{JF} & -\frac{KF}{JF} & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & -\frac{KP \cdot KERR}{TAU3} & -\frac{1}{TAU3}
\end{bmatrix}
\cdot
\begin{bmatrix} XM(1) \\ XM(2) \\ XM(3) \\ XM(4) \\ XM(5) \end{bmatrix}
+
\begin{bmatrix}
\frac{1}{JM} & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & \frac{KERR}{TAU3}
\end{bmatrix}
\cdot
\begin{bmatrix} TM \\ DC \end{bmatrix}
$$

or in symbolic form:

$$\dot{\underline{x}}_p = \underline{A}_p\, \underline{x}_p + \underline{B}_p\, \underline{u}_p$$

3.0 PROGRAM DESCRIPTION

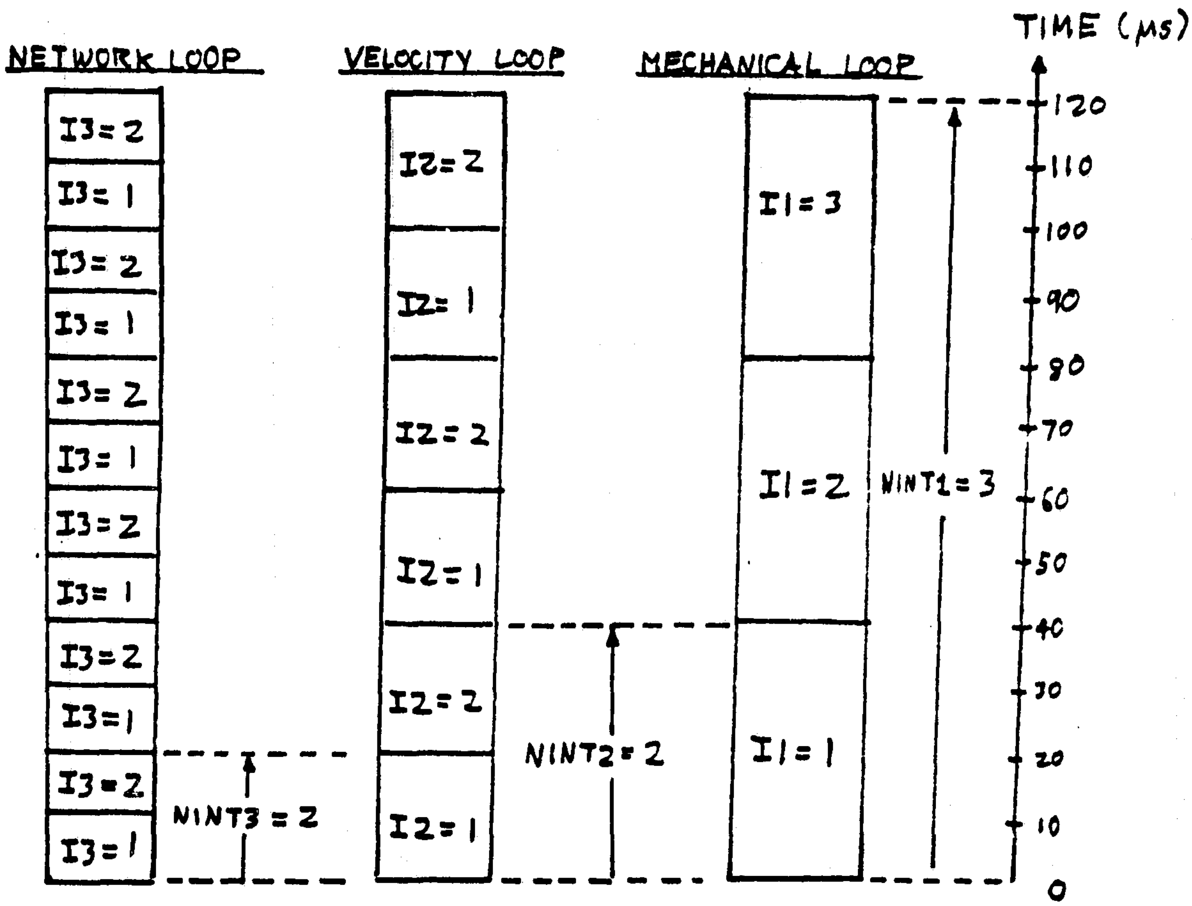
A program was written in FORTRAN to solve numerically the state equations of the three subsystems presented in the previous chapter. Since the time constants for these three components vary widely, it was decided to integrate each one of these loops independently. For example, the mechanical-position loop which varies the slowest would be integrated first. The velocity loop which has the next smallest time step would then be integrated NINT2 times for each of the NINT1 time steps of the mechanical-position loop. The velocity loop integrations use linear interpolation on the old and new state vectors of the mechanical loop in order to generate the inputs PE and RVEL. The same is true for the power conditioner and machine network. In this case, there are NINT3 integrations per velocity loop integration. This scheme eliminates having to integrate all 14 state variables at a rate determined by the smallest time step. The relationship between the three separate integration loops is given in the following example:

1) Let NINT1 = 3
   NINT2 = 2
   NINT3 = 2
   and TSNET = 10 μs

2) Then TSVLP = NINT3*TSNET = 20 μs

   and TSMLP = NINT2*TSVLP = 40 μs

3) Based upon this example, the relationship between the various integration loops is given below:



In order to allow some flexibility in the computer simulations it was decided to have three different modes of program execution. The first mode, PGMODE = 1, considers the power conditioner and machine network only. The other two modes include the entire servo loop but with a simple machine model for PGMODE = 2 and a detailed model for PGMODE = 3. A flow chart of the overall logic flow of this program is given in Figure (3-1).

PROGRAM MODE 1 (PGMODE = 1)

This mode is characterized by the following:

1. Fourth order machine and power conditioner model.
2. Constant machine speed (i.e., there are no mechanical loop integrations)
3. Constant torque (current) command (i.e., there are no velocity loop integrations)

Typical Applications

This mode is useful when it is desired to examine the behavior of the power conditioner and machine network at a specified speed, torque command and mode of operation (i.e., motoring, regenerative braking, and plugging). Total simulation time is generally below 10 ms (EMA time) which corresponds to 1000 integrations with a time step, TSNET, of 10 μs.

PROGRAM MODE 2 (PGMODE = 2)

This mode is characterized by the following:

1. Simple machine and power conditioner model consisting of:

    a) machine torque, TM = KT * IM, and

    b) a machine speed characteristic which is clamped at ±VMSAT (typically 9500 rpm)

2. Fifth order velocity loop model

3. Fifth order mechanical loop model

Typical Applications

This mode is used to obtain the step response of the EMA with one or two machines operating. The use of the simple machine model greatly reduces the total execution time and at the same time gives excellent results when compared to test.

PROGRAM MODE 3 (PGMODE = 3)

This mode is characterized by the following:

1. Fourth order machine and power conditioner model.
2. Fifth order velocity loop model.
3. Fifth order mechanical loop model.

Typical Applications

This mode is designed to obtain step responses of the EMA with either one or two machines operational. The fourth order power conditioner and machine model generates more accurate machine torques which serve as inputs to the mechanical-position loop model. Typical simulation times (EMA reference) are generally of the order of 40,000 integrations of the network state equations. This large number of integrations is necessary because of the rapid response of the power conditioner and machine model when compared with the velocity and mechanical loop integrations.

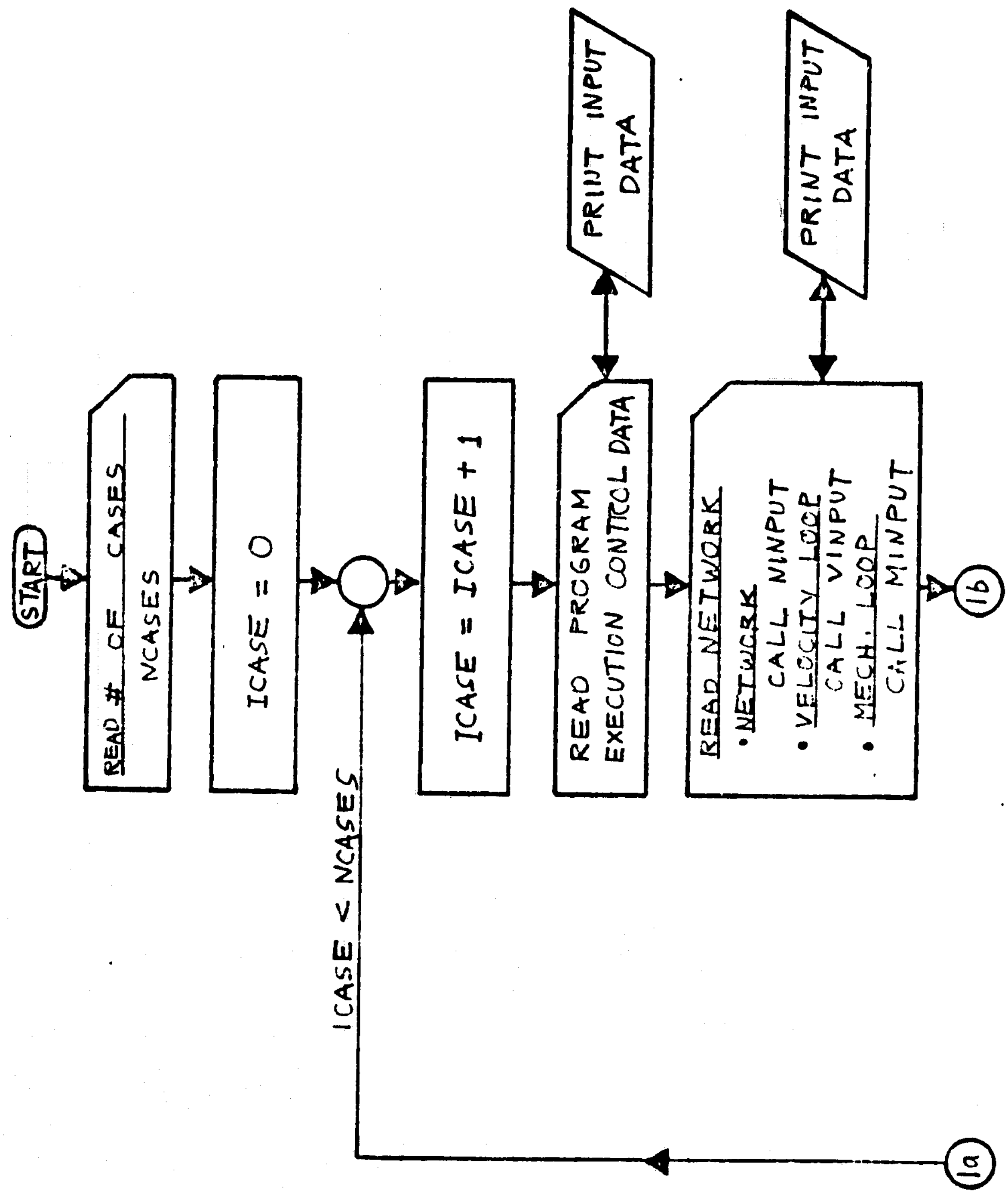


Figure 3-1. EMA Model Flow Chart

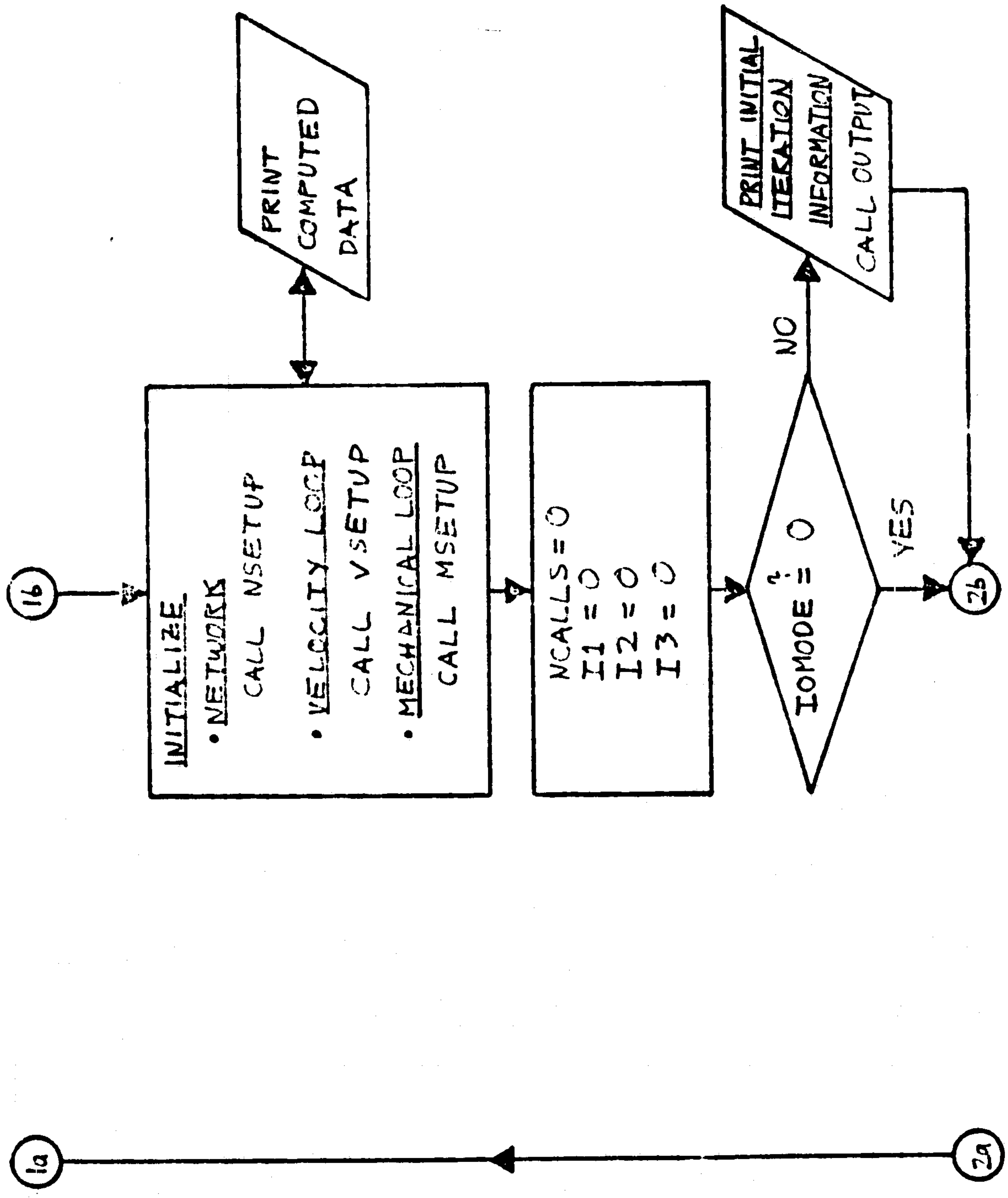


Figure 3-1. (cont)

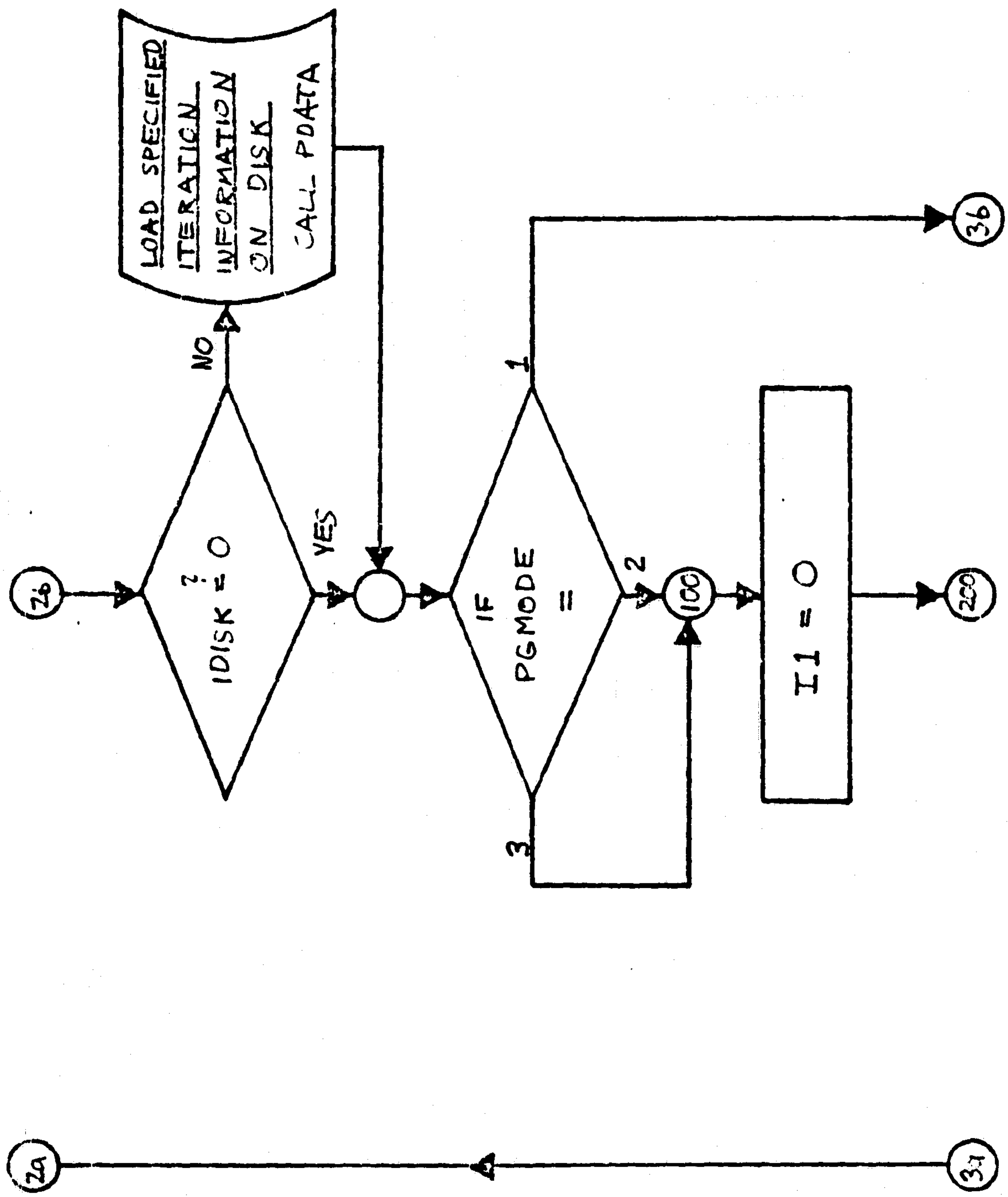


Figure 3-1. (cont)

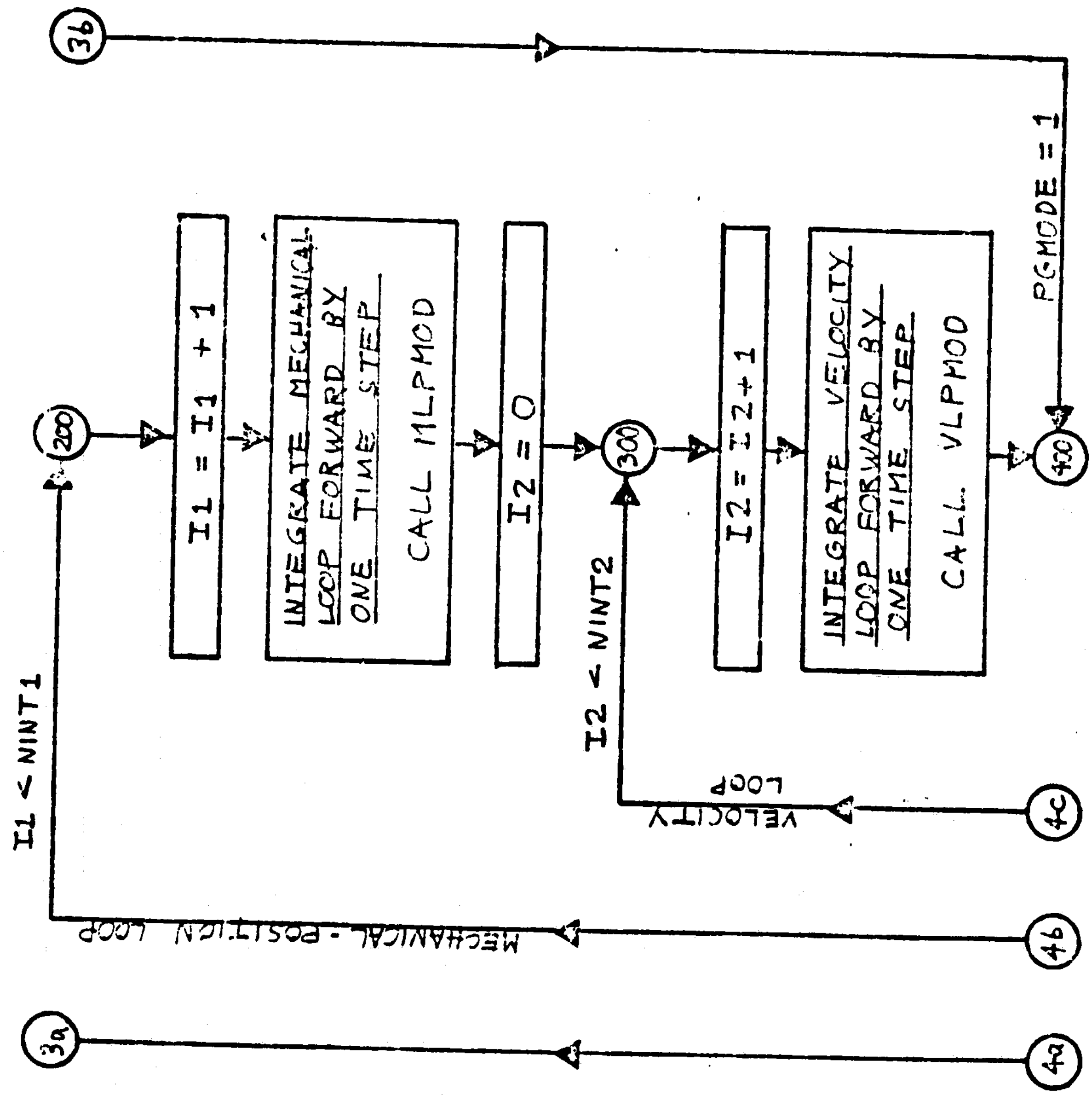


Figure 3-1. (cont)

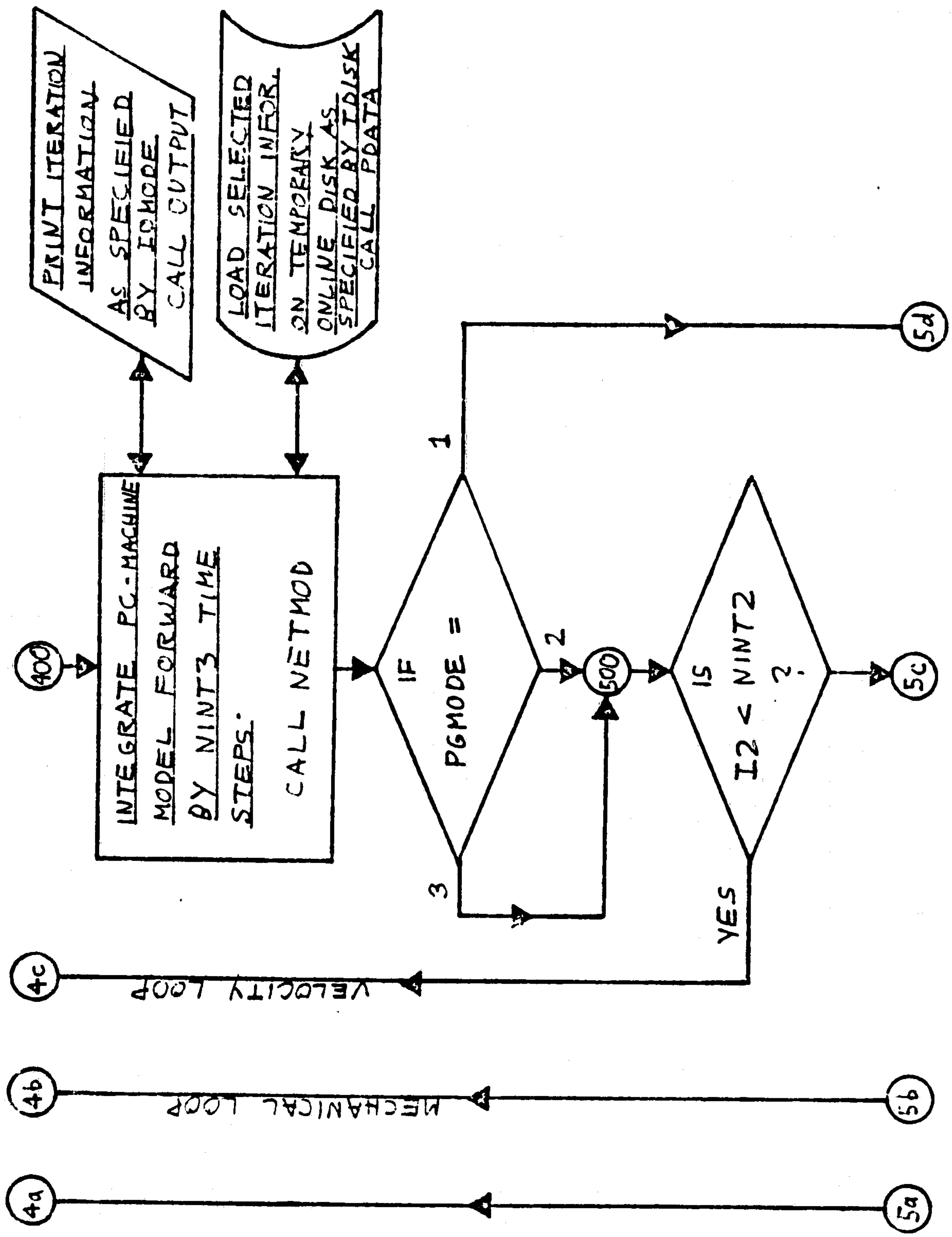


Figure 3-1. (cont)

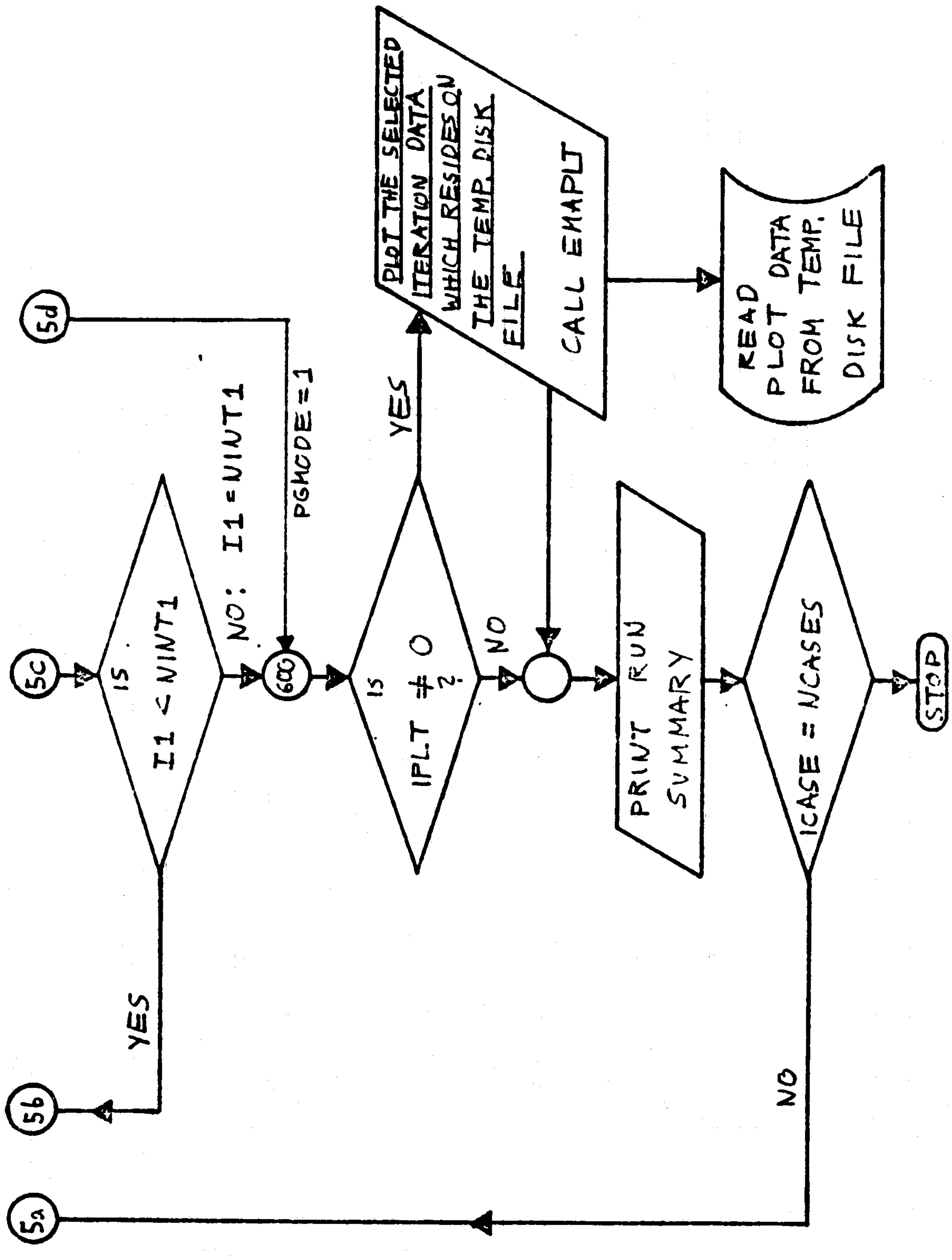
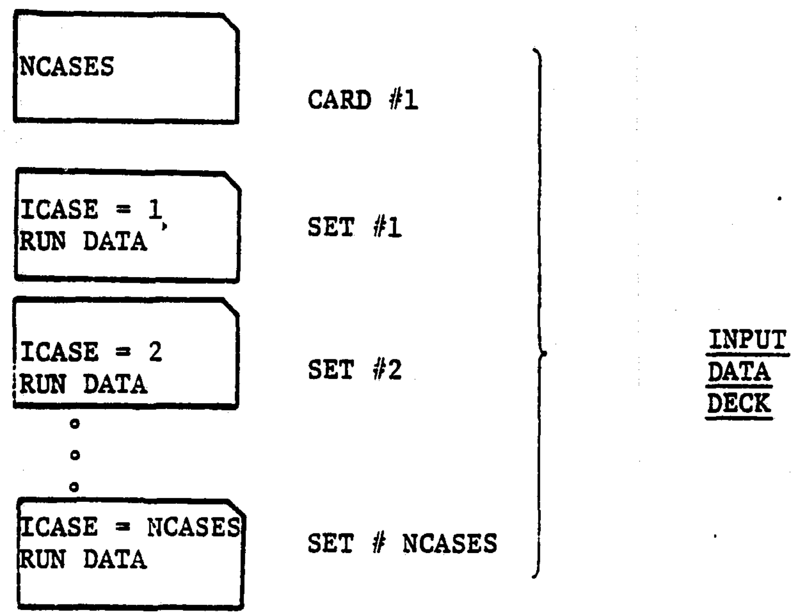


Figure 3-1. (cont)

4.0 RUN PREPARATION

This chapter describes the steps in the preparation of the input data cards for the EMA simulation program. The input structure has been designed to allow the execution of several cases per run. This input data structure can be represented pictorially as follows:



The total number of cases per run is given on the first data card and is represented by the integer variable NCASES in the program itself. The card sets which follow contain the input data for the cases specified by NCASES. In the main program each individual case is identified by the variable ICASE. The required variables and their input formats are the topics of the remainder of this chapter.

DESCRIPTION OF INPUT DATA

1. NCASES - Number of simulations or cases per run

2. PGMODE - Program Mode

$$\text{PGMODE} = \begin{cases} 1. & \text{Machine Model Only - 4th Order} \\ 2. & \text{Complete EMA Model - 10th Order (Simple Machine Model)} \\ 3. & \text{Complete EMA Model - 14th Order (Detailed Machine Model)} \end{cases}$$

3. IOMODE - Printing control

$$\text{IOMODE} = \begin{cases} 0 & \text{Do not print results of integrations} \\ 1 & \text{Print results of every integration} \\ & \vdots \\ N & \text{Print results of every Nth integration} \end{cases}$$

4. IDISK - Disk storage control

$$\text{IDISK} = \begin{cases} 0 & \text{Do not load integration data on disk} \\ 1 & \text{Load specified data for each integration on disk} \\ & \vdots \\ N & \text{Load specified data for every Nth integration on disk} \end{cases}$$

5. IPLT - Line printer plot control

$$\text{IPLT} = \begin{cases} 0 & \text{Do not plot} \\ 1 & \text{Plot data stored on disk} \end{cases}$$

6. NPLTS - Number of X-Y Plots (Excluding the plot of the logic signals)

7. IDX - Location in INDEX of the X variable for the X-Y Plots

IDX = NPLTS + 1

8. NPAGES - Number of pages per line printer plot

9. NVSTOR - Number of X-Y values stored on disk

NVSTOR = NPLTS + 1

10. INDEX - Vector which contains the ID numbers of the X-Y variables to be stored on disk. The ID numbers of the Y variables come first. The ID number of the X variable is placed immediately following the last Y variable. (See list of ID codes later in this report)

11. NINT3 - Number of Network Integrations:

$$\text{IF} = \begin{cases} \text{PGMODE} = 1 - \text{Total Number of Network Integrations} \\ \text{PGMODE} = 2 - \text{NINT3} = 0 \\ \text{PGMODE} = 3 - \text{Total Number of Network Integrations} \\ \qquad\qquad\quad \text{per Velocity Loop Integration.} \end{cases}$$

12. NTERMS - Control of the transition matrix calculation for the Network model

$$\text{NTERMS} = \begin{cases} 0 & \text{Use the error tolerance ETNET to control} \\ & \text{PHI and THETA} \\ \geq 1 & \text{Number of terms for PHI and THETA} \end{cases}$$

13. TSNET - Network integration time step (seconds)

$$5\ \mu s \leq \text{TSNET} \leq 20\ \mu s$$

14. ETNET - Maximum allowed difference in the last two terms of PHI for the PC-Machine Network

15. SHIFT - Commutation Switch Control (mech. rad.)

16. ITOL - Machine Line Current Tolerance (amps)

17. KT - Machine Torque Constant (nt - m/amp)

18. SAMPLE - Chopper Inductance Control

$$\text{SAMPLE} = \begin{cases} 0 \quad \text{Linear Case:} & \text{Do not update inductance} \\ & \text{of the chopper inductance} \\ 0 < \text{SAMPLE} < 1 & \text{e.g. SAMPLE} = 0.5\text{: Update} \\ & \text{Chopper inductance in 5\%} \\ & \text{increments} \end{cases}$$

19. XSV(1)···XSV(4) - Power Conditioner and Machine Network State Variables

20. E4 - Supply Battery Voltage (volts)

21. RD1···RDM - On resistances of diodes D1 - DM (ohms)

22. ROFF - Equivalent resistance of a cut off diode or transistor (ohms)

23. RQ1···RQM - On resistances of transistors Q1 - QM (ohms)

24. L1, L3, L4 - Leakage inductances of phases a, b, and c respectively. (Henries)

25. R10 - Battery Resistance (Ohms)

26. R3 - Chopper Inductor Resistance (ohms)

27. L2 - Chopper Inductor Inductance (henries)
(Maximum Value)

28. C1 - Capacitance of Filter Capacitor
(farads)

29. NBCHS - Number of Network Branches

30. NTWIGS - Number of Network Twigs

31. NLINKS - Number of Links

32. NETWIG - Number of Twig Voltage Sources

33. NCTWIG - Number of Twig Capacitors

34. NRTWIG - Number of Twig Resistors

35. NLTWIG - Number of Twig Inductors

36. NCLINK - Number of Link Capacitors

37. NRLINK - Number of Link Resistors

38. NLLINK - Number of Link Inductors

39. NDPTS - Number of data points in the curve of the chopper incremental inductance

40. XI(1)···XI(NDPTS) - X-values of the chopper incremental inductance curve (amps)

41. FXI(1)···FXI(NDPTS) - Y-values of the chopper incremental inductance curve (henries)

42. NINT2 - Number of velocity loop integrations per mechanical-position loop integration

IF PGMODE = 1, NINT2 = 0

43. NTERM2 - Control of the transition matrix calculation for the velocity loop model

$$\text{NTERM2} = \begin{cases} 0 & \text{Use the error tolerance ETVLP to control PHIV and THETAV} \\ \geq 1 & \text{Number of terms for PHIV and THETAV} \end{cases}$$

44. TSVLP - Time step of the velocity loop (seconds)

TSVLP = NINT3 * TSNET

45. ETVLP - Maximum allowed difference in the last two terms of PHIV for the velocity loop model

46. TAU1 - Position error amplifier time constant (seconds)

47. TAU2 - Velocity error FB amplifier time constant (seconds)

48. KE - Velocity error amplifier gain

49. KRL - Current error integrator gain

50. KV - Velocity FB amplifier gain
(amps/rotor radian/sec)

51. VHI - Maximum (+) rotor velocity during plugging (radians/sec)

52. VLO - Maximum (-) rotor velocity during plugging (radians/sec)

53. F1Y1 - Maximum plugging current divided by 5

54. F1Y2 - Maximum non-plugging current divided by 5

55. F2Y - Current rate limiter Y-coordinate (machine amps/5)

56. F2X - Current rate limiter X-coordinate (machine amps/5)

57. XV(1) - ICMD/5

58. XV(2) - ICMD1/5
IF PGMODE = 1 Set XV(2) equal to the desired machine line current divided by 5

59. XV(3) - Velocity FB-Loop Output

60. XV(4) - ICMDL/5

61. XV(5) - Current Rate Limiter Output
(machine amps * .0005277/sec)

62. NMACH - Number of machines active (1 or 2)

63. NINT1 - Number of integrations of the mechanical-position loop

64. NTERM1 - Control of the transition matrix calculation for the mechanical-position loop model

$$\text{NTERM1} = \begin{cases} 0 & \text{Use the error tolerance ETMLP to control PHIM and THETAM} \\ \geq 1 & \text{Number of terms for PHIM and THETAM} \end{cases}$$

65. TSMLP - Time step of the mechanical-position loop. (seconds)

66. ETMLP - Maximum allowed difference in the last two terms of PHIM for the mechanical-position loops

67. TAU3 - Position error amplifier time constant (seconds)

68. DC - Deflection command (flap degrees)

69. VMSAT - Rotor velocity limit (rotor radians/sec) Applies to PGMODE = 2 only

70. JM - Machine and reflected gear inertia ($kg - m^2$)

71. JF - Flap inertia ($kg - m^2$)

72. BF - Flap viscous damping coefficient (nt - m/flap rad/sec)

73. KACT - Actuator mount stiffness (nt - m/EMA output rad)

74. KF - Flap Stiffness Coefficient (nt - m/flap rad)

75. KP - Position FB-Loop Gain (deg/rad)

76. KERR - Position error amplifier gain (machine amps/5/degree flap)

77. N1 - Gear ratio

78. N2 - Gear ratio

79. N3 - Gear ratio

80. XM(1) - Rotor velocity (rad/sec)

81. XM(2) - EMA Reaction Torque (nt - m)

82. XM(3) - Flap Velocity (rad/sec)

83. XM(4) - Flap Position (radians)

84. XM(5) - Amplified Position Error Signal (machine amps/5)

85. XLAB(1)··· XLAB(10) - Contains the x-axis label for plots (40 characters max)

86. YLAB(1)··· YLAB(10) - Contains the y-axis label for plots (40 characters max)

87. TITLE(1)··· TITLE(10) - Contains the x-y plot title (40 characters max)

INPUT DATA FORMAT

The input data is read from 80 column computer cards. The formats for each variable are designated as follows:

VAR ← Variable Name

1	I10

This specifies the format code (I10 in this case)

Starting column of the 10 column field corresponding to format I10

The placement of integer, real single precision, real double precision and character data is described by the examples given below.

Integer data Input format: I10

e.g. IVAR = 10 and

1	I10

DATA CARD:

Col #	1	2	3	4	5	6	7	8	9	10	...	80
									1	0	...	

*RIGHT HAND JUSTIFY ALL INTEGER DATA

Real Single Precision Data Input Format E14.7

e.g. RVAR = 10. and

1	E14.7

Form 1:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	...	80
1	0	.												...	

The 10. can be placed anywhere within the 14 spaces

Form 2:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	...	80
1	.											E	1	...	

*Notice the exponent E1 must be right hand justified.

Real Double Precision Data Input Format D14.7

e.g. DVAR = 10.DO (i.e. 10) and

1	D14.7

1	2	3	4	5	6	7	8	9	10	11	12	13	14	...	80
1	.											D	1	...	

*Notice the exponent D1 must be right hand justified.

Character Data   Input Format 10A4

e.g. PLOT TITLE   and

1	10A4

1	2	3	4	5	6	7	8	9	10	...	40	...	80
P	L	O	T		T	I	T	L	E	...		...	

*Notice that character data is left hand justified.

FORMATS FOR THE INPUT DATA CARDS

1.

NCASES
1 I10

2.

PGMODE	IOMODE	IDISK	IPLT
1 I10	11 I10	21 I10	31 I10

3. (SKIP IF DISK = 0)

NPLTS	IDX	NPAGES	NVSTOR
1 I10	11 I10	21 I10	31 I10

4. (SKIP IF DISK -= 0)

INDEX(1) ···						
1 I10	11 I10	21 I10	31 I10	41 I10	51 I10	61 I10

5.

NINT3	NTERMS
1 I10	11 I10

6.

TSNET	ETNET	SHIFT	ITOL	KT
1 E14.7	15 E14.7	29 E14.7	43 E14.7	57 E14.7

7.

SAMPLE
1 E14.7

8.

XSV(1)	XSV(2)	XSV(3)	XSV(4)
1 D14.7	15 D14.7	29 D14.7	43 D14.7

9.

E4
1 E14.7

10.

RD1	RD2	RD3	RD4	RD5
1 E14.7	15 E14.7	29 E14.7	43 E14.7	57 E14.7

11.

RD6	RDR	RDB	RDM	ROFF
1 E14.7	15 E14.7	29 E14.7	43 E14.7	57 E14.7

12.

RQ1	RQ2	RQ3	RQ4	RQ5
1 E14.7	15 E14.7	29 E14.7	43 E14.7	57 E14.7

13.

RQ6	RQB	RQM
1 E14.7	15 E14.7	29 E14.7

14.

L1	L3	L4
1 E14.7	15 E14.7	29 E14.7

15.

R10	R3	L2	C1
1 E14.7	15 E14.7	29 E14.7	43 E14.7

16.

NBCHS	NTWIGS	NLINKS	NETWIG	NCTWIG	NRTWIG	NLTWIG
1 I10	11 I10	21 I10	31 I10	41 I10	51 I10	61 I10

NCLINK	NRLINK	NLLINK
1 I10	11 I10	21 I10

17.

NDPTS
1 I10

18.

XI(1)	XI(2)	XI(3)	XI(4)	XI(5)
1 D14.7	15 D14.7	20 D14.7	43 D14.7	57 D14.7

XI(6)	XI(7)	XI(8) ···		
1 D14.7	15 D14.7	29 D14.7	43 D14.7	57 D14.7

19.

FXI(1)	FXI(2)	FXI(3)	FXI(4)	FXI(5)
1 D14.7	15 D14.7	29 D14.7	43 D14.7	57 D14.7

FXI(6)	FXI(7)	FXI(8) ···		
1 D14.7	15 D14.7	29 D14.7	43 D14.7	57 D14.7

20.

NINT2	NTERM2
1 I10	11 I10

21.

TSVLP	ETVLP	TAU1	TAU2	KE
1 E14.7	15 E14.7	29 E14.7	43 E14.7	57 E14.7

22.

KRL	KV	VHI	VLO
1 E14.7	15 E14.7	29 E14.7	43 E14.7

23.

F1Y1		F1Y2	
1	E14.7	15	E14.7

24.

F2Y		F2X	
1	E14.7	15	E14.7

25.

XV(1)		XV(2)		XV(3)		XV(4)		XV(5)	
1	D14.7	15	D14.7	29	D14.7	43	D14.7	57	D14.7

26.

NMACH		NINT1		NTERM1	
1	I10	11	I10	21	I10

27.

TSMLP		ETMLP		TAU3		DC		VMSAT	
1	E14.7	15	E14.7	29	E14.7	43	E14.7	57	E14.7

28.

JM		JF		BF		KACT		KF	
1	E14.7	15	E14.7	29	E14.7	43	E14.7	57	E14.7

29.

XM(1)		XM(2)		XM(3)		XM(4)		XM(5)	
1	D14.7	15	D14.7	29	D14.7	43	D14.7	57	D14.7

30. (SKIP IF IDISK = 0 .OR. IPLT = 0)

(XLAB(I), I = 1, 10)

10A4

31. (SKIP IF IDISK = 0 .OR. IPLT = 0)

(TITLE(J), J = 1, 10) (YLAB(J), J = 1, 10)

10A4	10A4	1
⋮	⋮	⋮
10A4	10A4	NPLTS

STORAGE OF PLOT DATA ON DISK

All data which is to be plotted is first stored on a temporary disk file inorder to reduce the core requirements of the program. This data is loaded onto disk using the following formats:

1. 5E14.7 (Real data)
2. 80L1 (Logical Variables)

The unit number of this temporary disk file at VPI is 8. If this unit number is different at the USER's installation then the following changes must be made:

In subroutines

```
PDATA
EMAPLT
BOLPLT
```

all statements of the form

```
REWIND8 and
WRITE (8,     )
```

must be changed to the unit number valid at that installation.

For example if the unit number is 9 then these become:

```
REWIND9
WRITE (9, FORMAT #)
```

This data when read from disk by the plotting routines is loaded into plot vectors which presently have a dimension of 1001 words. This corresponds to the data produced by 1000 integrations (i.e. Initial Conditions + 1000 integrations). If more than a 1000 integrations are contemplated then some of the data must be skipped. This is accomplished by the input parameter IDISK defined earlier in this chapter.

IDENTIFICATION NUMBERS FOR VARIABLES WHICH MAY BE STORED ON THE TEMPORARY DISK FILE

(THESE ID NUMBERS ARE LOADED INTO THE VECTOR INDEX)

In addition to these ID numbers, each variable also has a restriction code number defined as follows:

RESTRICTION CODE	MEANING
0	No restrictions
1	Not available for PGMODE = 1
2	Not available for PGMODE = 2
3	Not available for PGMODE = 3
4	Not presently available
5	Do not plot or store on disk (meaningless)

The ID numbers and restriction codes for each variable are listed sequentially starting on the following page.

The loading of the variables identified in INDEX on to the temporary disk file is accomplished by SUBROUTINE PDATA.

VARIABLE NAME	ID NUMBER	RESTRICTION CODE	LOCATION
CIA	1	2	COMMON/BLK1/
CIB	2	2	COMMON/BLK1/
CIC	3	2	COMMON/BLK1/
VA	4	2	COMMON/BLK1/
VB	5	2	COMMON/BLK1/
VC	6	2	COMMON/BLK1/
VAB	7	2	COMMON/BLK1/
VBC	8	2	COMMON/BLK1/
VCA	9	2	COMMON/BLK1/
PA	10	2	COMMON/BLK1/
PB	11	2	COMMON/BLK1/
PC	12	2	COMMON/BLK1/
PCORE	13	0	COMMON/BLK1/
PML	14	0	COMMON/BLK1/
PTRM	15	2	COMMON/BLK1/
PSO	16	2	COMMON/BLK1/
PEM	17	2	COMMON/BLK1/
PGM	18	0	COMMON/BLK1/
PNM	19	0	COMMON/BLK1/
TEM	20	2	COMMON/BLK1/
TM	21	2	COMMON/BLK1/
RRANG	22	2	COMMON/BLK6/
RANG	23	0	COMMON/BLK6/
RVEL	24	0	COMMON/BLK6/
RACEL	25	4	COMMON/BLK6/
ICMDL	26	0	COMMON/BLK7/
ICMD1	27	0	COMMON/BLK7/
IMC	28	0	COMMON/BLK7/
IM	29	0	COMMON/BLK7/

VARIABLE NAME	ID NUMBER	RESTRICTION CODE	LOCATION
BOC(1)	30	2	COMMON/BLK12/
BOC(2)	31	2	COMMON/BLK12/
BOC(3)	32	2	COMMON/BLK12/
BGAP(1)	33	2	COMMON/BLK12/
BGAP(2)	34	2	COMMON/BLK12/
BGAP(3)	35	2	COMMON/BLK12/
FS(1)	36	2	COMMON/BLK12/
FS(2)	37	2	COMMON/BLK12/
FS(3)	38	2	COMMON/BLK12/
FM(1)	39	2	COMMON/BLK12/
FM(2)	40	2	COMMON/BLK12/
FM(3)	41	2	COMMON/BLK12/
FG(1)	42	2	COMMON/BLK12/
FG(2)	43	2	COMMON/BLK12/
FG(3)	44	2	COMMON/BLK12/
HM(1)	45	2	COMMON/BLK12/
HM(2)	46	2	COMMON/BLK12/
HM(3)	47	2	COMMON/BLK12/
HG(1)	48	2	COMMON/BLK12/
HG(2)	49	2	COMMON/BLK12/
HG(3)	50	2	COMMON/BLK12/
LMV(1)	51	2	COMMON/BLK12/
LMV(2)	52	2	COMMON/BLK12/
LMV(3)	53	2	COMMON/BLK12/
LGV(1)	54	2	COMMON/BLK12/
LGV(2)	55	2	COMMON/BLK12/
LGV(3)	56	2	COMMON/BLK12/

VARIABLE NAME	ID NUMBER	RESTRICTION CODE	LOCATION
QLOSS (1)	57	2	COMMON/BLK13/
(2)	58	2	COMMON/BLK13/
(3)	59	2	COMMON/BLK13/
(4)	60	2	COMMON/BLK13/
(5)	61	2	COMMON/BLK13/
(6)	62	2	COMMON/BLK13/
(7)	63	2	COMMON/BLK13/
(8)	64	2	COMMON/BLK13/
DLOSS (1)	65	2	COMMON/BLK13/
(2)	66	2	COMMON/BLK13/
(3)	67	2	COMMON/BLK13/
(4)	68	2	COMMON/BLK13/
(5)	69	2	COMMON/BLK13/
(6)	70	2	COMMON/BLK13/
(7)	71	2	COMMON/BLK13/
(8)	72	2	COMMON/BLK13/
(9)	73	2	COMMON/BLK13/
PELOSS	74	2	COMMON/BLK13/
FANG	75	1	COMMON/BLK28/
PE	76	1	COMMON/BLK28/
VE	77	1	COMMON/BLK28/
TACT	78	1	COMMON/BLK28/
TIME	79	0	COMMON/BLK27/

VARIABLE TIME		ID NUMBER	RESTRICTION CODE	LOCATION
BCHCUR	(1)	201	2	COMMON/BLK2/
	(2)	202	2	COMMON/BLK2/
	(3)	203	2	COMMON/BLK2/
	(4)	204	2	COMMON/BLK2/
	(5)	205	2	COMMON/BLK2/
	(6)	206	2	COMMON/BLK2/
	(7)	207	2	COMMON/BLK2/
	(8)	208	2	COMMON/BLK2/
	(9)	209	2	COMMON/BLK2/
	(10)	210	2	COMMON/BLK2/
	(11)	211	2	COMMON/BLK2/
	(12)	212	2	COMMON/BLK2/
	(13)	213	2	COMMON/BLK2/
	(14)	214	2	COMMON/BLK2/
	(15)	215	2	COMMON/BLK2/
	(16)	216	2	COMMON/BLK2/
	(17)	217	2	COMMON/BLK2/
	(18)	218	2	COMMON/BLK2/
	(19)	219	2	COMMON/BLK2/
	(20)	220	2	COMMON/BLK2/
	(21)	221	2	COMMON/BLK2/
	(22)	222	2	COMMON/BLK2/
	(23)	223	2	COMMON/BLK2/
BCHEMF	(1)	224	2	COMMON/BLK2/
	(2)	225	2	COMMON/BLK2/
	(3)	226	2	COMMON/BLK2/
	(4)	227	2	COMMON/BLK2/
	(5)	228	2	COMMON/BLK2/
	(6)	229	2	COMMON/BLK2/
	(7)	230	2	COMMON/BLK2/
	(8)	231	2	COMMON/BLK2/

VARIABLE NAME	ID NUMBER	RESTRICTION CODE	LOCATION
BCHEMF (9)	232	2	COMMON/BLK2/
(10)	233	2	COMMON/BLK2/
(11)	234	2	COMMON/BLK2/
(12)	235	2	COMMON/BLK2/
(13)	236	2	COMMON/BLK2/
(14)	237	2	COMMON/BLK2/
(15)	238	2	COMMON/BLK2/
(16)	239	2	COMMON/BLK2/
(17)	240	2	COMMON/BLK2/
(18)	241	2	COMMON/BLK2/
(19)	242	2	COMMON/BLK2/
(20)	243	2	COMMON/BLK2/
(21)	244	2	COMMON/BLK2/
(22)	245	2	COMMON/BLK2/
(23)	246	2	COMMON/BLK2/
BCHPOW (1)	247	2	COMMON/BLK2/
(2)	248	2	COMMON/BLK2/
(3)	249	2	COMMON/BLK2/
(4)	250	2	COMMON/BLK2/
(5)	251	2	COMMON/BLK2/
(6)	252	2	COMMON/BLK2/
(7)	253	2	COMMON/BLK2/
(8)	254	2	COMMON/BLK2/
(9)	255	2	COMMON/BLK2/
(10)	256	2	COMMON/BLK2/
(11)	257	2	COMMON/BLK2/
(12)	258	2	COMMON/BLK2/
(13)	259	2	COMMON/BLK2/
(14)	260	2	COMMON/BLK2/
(15)	261	2	COMMON/BLK2/
(16)	262	2	COMMON/BLK2/

VARIABLE NAME	ID NUMBER	RESTRICTION CODE		LOCATION
BCHPOW(17)	263	2		COMMON/BLK2/
(18)	264	2		COMMON/BLK2/
(19)	265	2		COMMON/BLK2/
(20)	266	2		COMMON/BLK2/
(21)	267	2		COMMON/BLK2/
(22)	268	2		COMMON/BLK2/
(23)	269	2		COMMON/BLK2/
XSV(1)	270	2		COMMON/BLK15/
XSV(2)	271	2		COMMON/BLK15/
XSV(3)	272	2		COMMON/BLK15/
XSV(4)	273	2		COMMON/BLK15/
U(1)	274	2		COMMON/BLK15/
U(2)	275	2		COMMON/BLK15/
U(3)	276	2		COMMON/BLK15/
U(4)	277	2		COMMON/BLK15/
XSVDOT(1)	278	2		COMMON/BLK15/
XSVDOT(2)	279	2		COMMON/BLK15/
XSVDOT(3)	280	2		COMMON/BLK15/
XSVDOT(4)	281	2		COMMON/BLK15/
XV(1)	282	1	DO NOT USE THESE UNLESS NINT2=1 AND NINT3=1	COMMON/BLK17/
(2)	283	1		COMMON/BLK17/
(3)	284	1		COMMON/BLK17/
(4)	285	1		COMMON/BLK17/
(5)	286	1		COMMON/BLK17/
UV(1)	287	1		COMMON/BLK17/
(2)	288	1		COMMON/BLK17/
(3)	289	1		COMMON/BLK17/
(4)	290	1		COMMON/BLK17/
(5)	291	1		COMMON/BLK17/

VARIABLE NAME	ID NUMBER	RESTRICTION CODE	LOCATION
XVOLD(1)	292	5	COMMON/BLK17/
(2)	293	5	COMMON/BLK17/
(3)	294	5	COMMON/BLK17/
(4)	295	5	COMMON/BLK17/
(5)	296	5	COMMON/BLK17/
UVOLD(1)	297	4	COMMON/BLK17/
(2)	298	4	COMMON/BLK17/
(3)	299	4	COMMON/BLK17/
(4)	300	4	COMMON/BLK17/
(5)	301	4	COMMON/BLK17/
XVI(1)	302	1	COMMON/BLK17/
(2)	303	1	COMMON/BLK17/
(3)	304	1	COMMON/BLK17/
(4)	305	1	COMMON/BLK17/
(5)	306	1	COMMON/BLK17/
UVI(1)	307	4	COMMON/BLK17/
(2)	308	4	COMMON/BLK17/
(3)	309	4	COMMON/BLK17/
(4)	310	4	COMMON/BLK17/
(5)	311	4	COMMON/BLK17/
XM(1)	312	1	COMMON/BLK20/
(2)	313	1	COMMON/BLK20/
(3)	314	1	COMMON/BLK20/
(4)	315	1	COMMON/BLK20/
(5)	316	1	COMMON/BLK20/
UM(1)	317	1	COMMON/BLK20/
(2)	318	1	COMMON/BLK20/
(3)	319	1	COMMON/BLK20/
(4)	320	1	COMMON/BLK20/
(5)	321	1	COMMON/BLK20/

DO NOT USE THESE UNLESS NINT2=1 AND NINT3=1

VARIABLE NAME	ID NUMBER	RESTRICTION CODE	LOCATION
XMOLD(1)	322	5	COMMON/BLK20/
(2)	323	5	COMMON/BLK20/
(3)	324	5	COMMON/BLK20/
(4)	325	5	COMMON/BLK20/
(5)	326	5	COMMON/BLK20/
UMOLD(1)	327	4	COMMON/BLK20/
(2)	328	4	COMMON/BLK20/
(3)	329	4	COMMON/BLK20/
(4)	330	4	COMMON/BLK20/
(5)	331	4	COMMON/BLK20/
XMI(1)	332	1	COMMON/BLK20/
(2)	333	1	COMMON/BLK20/
(3)	334	1	COMMON/BLK20/
(4)	335	1	COMMON/BLK20/
(5)	336	1	COMMON/BLK20/
UMI(1)	337	4	COMMON/BLK20/
(2)	338	4	COMMON/BLK20/
(3)	339	4	COMMON/BLK20/
(4)	340	4	COMMON/BLK20/
(5)	341	4	COMMON/BLK20/

Example:

Suppose one wishes to store on disk and later plot every 5th value of

TM, CIA, CIB, CIC, FANG, AND RVEL

versus TIME and that PGMODE = 3. From the preceeding table of variable ID numbers one obtains:

1. TM - ID # = 21
2. CIA - ID # = 1
3. CIB - ID # = 2
4. CIC - ID # = 3
5. FANG - ID # = 75
6. RVEL - ID # = 24
7. TIME - ID # = 79

Since none of these variables have a restriction code number corresponding to PGMODE = 3 the input data becomes:

IDISK = 5 (stores every 5th value)

IPLT = 1 (plot specified values)

NPLTS = 6 (number of x-y plots)

IDX = 7 (location of x variable in INDEX)***

***Always place the x variable ID # after the last y variable in INDEX i. e. IDX = NVSTOR.

NVSTOR = 7 (Total number of variables stored NVSTOR = NPLTS + 1)

INDEX =

21	1	2	3	75	24	79

APPENDIX A PROGRAM LISTING

```
C-----------------------------------------------------------------------
C
C     VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY  (VPISU)
C     DEPARTMENT OF ELECTRICAL ENGINEERING
C     BLACKSBURG, VA.  24061
C
C     PROGRAM WRITTEN BY T.W. NEHL
C
C-----------------------------------------------------------------------
C
C     PROGRAM TITLE:
C
C     DYNAMIC MODEL OF THE NASA-DELCO ELECTROMECHANICAL
C     ACTUATOR FOR AEROSPACE APPLICATIONS.
C
C     ORIGIN:
C
C     THIS PROGRAM WAS DEVELOPED AT VPISU UNDER
C     NASA CONTRACT NAS9-15091
C
C     PRINCIPAL INVESTIGATOR:  N.A.O. DEMERDASH
C     GRADUATE RESEARCH ASSISTANT:  T.W. NEHL
C
C-----------------------------------------------------------------------
      INTEGER PGMODE
      COMMON/BLK22 /  PGMODE,IOMODE,IDISK,IPLT
      COMMON/BLK23 /  NCALLS,NREC,NUMINT,NPTS
      COMMON/BLK24 /  NINT1,NINT2,NINT3,I1,I2,I3
      COMMON/BLK27 /  TSNET,ETNET,NTERM3,DIFMAX,ITER,TIME
      COMMON/BLK29 /  NPLTS,IDX,NPAGES,NVSTOR
      COMMON/BLK30 /  INDEX(100)
      COMMON/BLK33 /  SAMPLE,NNETSW
```

```
C.....................................................................
C     PRINT THE PROGRAM HEADING
C.....................................................................
      WRITE(6,6040)
      WRITE(6,6050)
C.....................................................................
C     READ IN THE NUMBER OF SIMULATION CASES
C.....................................................................
      READ (5,5000) NCASES
C.....................................................................
C     EXECUTE THE SIMULATION CASES
C.....................................................................
      DO 3000 ICASE=1,NCASES
C     ----------------------------------------------------------------
C     READ INPUT DATA FOR CASE NUMBER ICASE
C
      READ (5,5000) PGMODE,IOMODE,IDISK,IPLT
      IF(IDISK.NE.0) READ(5,5000) NPLTS,IDX,NPAGES,NVSTOR
      IF(IDISK.NE.0) READ(5,5000) (INDEX(I),I=1,NVSTOR)
      WRITE(6,6010) ICASE,NCASES
      WRITE(6,6060)
      WRITE(6,6080) PGMODE
      WRITE(6,6090) IOMODE
      WRITE(6,6100) IDISK
      WRITE(6,6110) IPLT
      IF(IDISK.EQ.0) GO TO 10
      WRITE(6,6120)
      WRITE(6,6130) NPLTS
      WRITE(6,6140) IDX
      WRITE(6,6150) NPAGES
      WRITE(6,6160) NVSTOR
      WRITE(6,6170) (I,INDEX(I),I=1,NVSTOR)
```

```
   10 CONTINUE
C     ----------------------------------------------------------------
C     READ INPUT DATA FOR THE EMA MODEL
C
      CALL NINPUT
      CALL VINPUT
      CALL MINPUT
C     ----------------------------------------------------------------
C     PERFORM INITIALIZATIONS
C
      CALL NSETUP
      CALL VSETUP
      CALL MSETUP
C     ----------------------------------------------------------------
C     PRINT INITIAL ITERATION INFORMATION IF SPECIFIED
C
      NCALLS=0
      I1=0
      I2=0
      I3=0
      IF(IOMODE.NE.0) CALL OUTPUT(PGMODE,0,NCALLS)
C     ----------------------------------------------------------------
C     LOAD INITIAL DATA ON DISK IF SPECIFIED
C
      NREC=0
      IF(IDISK.NE.0) CALL PDATA(PGMODE,NREC,1)
C     ----------------------------------------------------------------
C     START THE INTEGRATION LOOPS
C
      GO TO (400,100,100),PGMODE
  100 CONTINUE
C     ----------------------------------------------------------------
```

```
C     INTEGRATE THE MECHANICAL POSITION LOOP FORWARD
C
      I1=0
  200 CONTINUE
      I1=I1+1
      CALL MLPMOD
C     ------------------------------------------------------------
C     INTEGRATE THE VELOCITY LOOP FORWARD
C
      I2=0
  300 CONTINUE
      I2=I2+1
      CALL VLPMOD
C     ------------------------------------------------------------
C     INTEGRATE THE MACHINE-POWER CONDITIONER FORWARD
C
  400 CONTINUE
      CALL NETMOD
      GO TO (600,500,500),PGMODE
C     ------------------------------------------------------------
C     FINISHED WITH VELOCITY LOOP ITEGRATIONS?
C
  500 CONTINUE
      IF(I2.LT.NINT2) GO TO 300
C     ------------------------------------------------------------
C     FINISHED WITH MECHANICAL-POSITION LOOP INTEGRATIONS?
C
      IF(I1.LT.NINT1) GO TO 200
  600 CONTINUE
C     ------------------------------------------------------------
C     PLOT DATA IF SPECIFIED
C
```

```
      IF(IPLT.GE.1) CALL EMAPLT(NREC)
C     ------------------------------------------------------------------
C     PRINT THE RUN SUMMARY
C
      WRITE(6,6180)
      WRITE(6,6190) ICASE,NCASES,PGMODE
      WRITE(6,6200) NUMINT
      WRITE(6,6210) NINT1
      N12=NINT1*NINT2
      WRITE(6,6220) N12
      N123=NINT1*NINT2*NINT3
      WRITE(6,6230) N123
      WRITE(6,6240) TIME
      WRITE(6,6250) NREC
      WRITE(6,6260) NPTS
      WRITE(6,6270) NNETSW
 3000 CONTINUE
C.......................................................................
C     FORMATS
C.......................................................................
 5000 FORMAT(7I10)
 6010 FORMAT(1H1,130('.')/1H ,130('.')/1H ,55('.'),'     CASE',I3,' OF',
     1I3,'     ',54('.')/1H ,130('.')/1H ,130('.')////)
 6040 FORMAT(1H1,////////////////,1H-,T33,'DYNAMIC MODEL OF THE NASA ELE
     1CTROMECHANICAL FLIGHT CONTROL SURFACE ACTUATOR  '/,1H-,T33,'POSSIB
     2LE PROGRAM MODES'/,1H0,T33,'1) PGMODE=1:  FOURTH ORDER MACHINE AND
     3 ',                                                              '
     4POWER CONDITIONER MODEL'/,1H ,T52,'CONSTANT SPEED'/,1H ,T52,'CONST
     5ANT TORQUE (CURRENT) COMMAND'/,1H0,T33,'2) PGMODE=2:  TENTH ORDER
     6EMA MODEL'/,1H ,T52,'MACHINE AND POWER CONDITIONER MODEL:  IM=ICMD
     71,TM=KT*IM')
 6050 FORMAT(1H ,T52,'VELOCITY LOOP MODEL:  FIFTH ORDER'/,1H ,T52,'MECHA
```

```
     1NICAL-POSITION LOOP MODEL:  FIFTH ORDER'/,1H0,T33,'3) PGMODE=3:  F
     2ORTEETH ORDER EMA MODEL'/,1H ,T52,'MACHINE AND POWER CONDITIONER M
     3ODEL:  FOURTH ORDER'/,1H ,T52,'VELOCITY LOOP MODEL:  FIFTH ORDER'/
     4,1H ,T52,'MECHANICAL-POSITION LOOP MODEL:  FIFTH ORDER')
6060 FORMAT(1H0,T5,'THE MAIN PROGRAM CONTROL VARIABLES ARE:',/,1H ,T5,
     138('-'),//)
6080 FORMAT(1H0,T5,'PROGRAM MODE (PGMODE) = ',I1)
6090 FORMAT(1H0,T5,'OUTPUT (PRINTING) MODE (IOMODE) = ',I4)
6100 FORMAT(1H0,T5,'PLOT RECORD STORAGE MODE (IDISK) = ',I4)
6110 FORMAT(1H0,T5,'PLOT CONTROL (IPLT) = ',I2///)
6120 FORMAT(1H0,T5,'THE PLOTTER CONTROL VARIABLES ARE:'/,1H ,T5,33('-')
     1,//)
6130 FORMAT(1H0,T5,'NUMBER OF X-Y PLOTS (NPLTS) = ',I3)
6140 FORMAT(1H0,T5,'X-PLOT VECTOR INDEX (IDX) = ',I3,' (INDEX INTO VECT
     1OR REC)')
6150 FORMAT(1H0,T5,'NUMBER OF PAGES PER PLOT(NPAGES) = ',I2)
6160 FORMAT(1H0,T5,'NUMBER OF PLOT VARIABLES STORED ON DISK (NVSTOR) =
     1',I3)
6170 FORMAT(1H0,T5,'PLOT VARIABLE INDEX VECTOR (INDEX):',/,
     15('          (',I3,')  ',I3))
6180 FORMAT(1H1,T5,'RUN SUMMARY',/,1H ,T5,11('-'),//)
6190 FORMAT(1H0,T5,'CASE ',I3,' OF ',I3,'    PROGRAM OPTION (PGMODE) '
     1,I1)
6200 FORMAT(1H0,T5,'TOTAL NUMBER OF INTEGRATIONS (NUMINT):',I8)
6210 FORMAT(1H0,T5,'TOTAL NUMBER OF MECHANICAL-POSITION LOOP INTEGRATIO
     1NS (NINT1):',I8)
6220 FORMAT(1H0,T5,'TOTAL NUMBER OF VELOCITY LOOP INTEGRATIONS (NINT2*N
     1INT1):',I8)
6230 FORMAT(1H0,T5,'TOTAL NUMBER OF PC-MACHINE NETWORK INTEGRATIONS (NI
     1NT3*NINT2*NINT1):',I8)
6240 FORMAT(1H0,T5,'TOTAL SIMULATION TIME - EMA REFERENCE (TIME):  ',
     1 E14.7,' SEC.')
```

```
6250 FORMAT(1H0,T5,'TOTAL NUMBER OF ITERATION RECORDS STORED ON DISK (N
    1REC):',I8)
6260 FORMAT(1H0,T5,'TOTAL NUMBER OF DATA POINTS (NPTS):',I8)
6270 FORMAT(1H0,T5,'TOTAL NUMBER OF TIMES THE PC-MACHINE NETWORK CHANGE
    1D (NNETSW):',I8)
     STOP
     END
```

```
      SUBROUTINE NINPUT
C-----------------------------------------------------------------------
C
C     SUBROUTINE NINPUT READS AND PRINTS THE DATA FOR THE
C     MACHINE-POWER CONDITIONER NETWORK MODEL
C
C-----------------------------------------------------------------------
      DIMENSION RDOON(9),RTRON(9)
      DIMENSION        IQ(15,8),NIO1(10)
      DOUBLE PRECISION XSV(4),U(4),XSVDOT(4)
      DOUBLE PRECISION XI(20),FXI(20),HI(20),BI(20),RHI(20),YI2(20)
      LOGICAL          AA,BB,CC,FF,AN,BN,CN,FN,QAP1,QAN1,QBP1,QBN1,QCP1,
     1                 QCN1,QMON,QBON,QIOFF,Q(8),NETCHG,NDUMP,D(9)
     2                ,QBC,QMC
      REAL             BRANCH(23),L1,L2,L3,L4
      REAL KT,ITOL
      COMMON/BLK3  /   RQ1,RQ2,RQ3,RQ4,RQ5,RQ6,RQM,RQB,ROFF
      COMMON/BLK4  /   RD1,RD2,RD3,RD4,RD5,RD6,RDB,RDM,RDR
      COMMON/BLK5  /   XI,FXI,HI,BI,RHI,YI2,NDPTS
      COMMON/BLK6  /   RRANG,RANG,RVEL,RACEL
      COMMON/BLK8  /   AA,BB,CC,FF,AN,BN,CN,FN,QAP1,QAN1,QBP1,QBN1,QCP1,
     1               QCN1,QMON,QBON,QIOFF,Q,NETCHG,NDUMP,D,QBC,QMC
      COMMON/BLK9  /   E1,E2,E3,E4,C1,R1,R2,R3,R4,R5,R6,R7,R8,R9,L1,R10,
     1                 R11,R12,R13,R14,L2,L3,L4
      COMMON/BLK10 /   NBCHS,NTWIGS,NLINKS,NETWIG,NCTWIG,NRTWIG,NLTWIG,
     1                 NCLINK,NRLINK,NLLINK,IQ
      COMMON/BLK15 /   XSV,U,XSVDOT
      COMMON/BLK21 /   ITOL,SHIFT,KT
      COMMON/BLK24 /   NINT1,NINT2,NINT3,I1,I2,I3
      COMMON/BLK25 /   TSMLP,ETMLP,NTERM1,NMACH,N1,N2,N3,NX
      COMMON/BLK27 /   TSNET,ETNET,NTERM3,DIFMAX,ITER,TIME
      COMMON/BLK33 /   SAMPLE,NNETSW
```

```
      EQUIVALENCE      (RTRON(1),RO1),(RDDON(1),RD1)
      EQUIVALENCE      (BRANCH(1),E1)
      EQUIVALENCE      (NIGI(1),NBCHS)
C.......................................................................
C     READ AND PRINT THE NETWORK MODEL CONTROL PARAMETERS
C.......................................................................
      READ(5,5100) NINT3,NTERM3
      READ(5,5000) TSNET,ETNET,SHIFT,ITOL,KT
      READ(5,5000) SAMPLE
      WRITE(6,6800)
      WRITE(6,6810) NINT3
      WRITE(6,6820) NTERM3
      WRITE(6,6830) TSNET
      WRITE(6,6840) ETNET
      WRITE(6,6850) SHIFT
      WRITE(6,6860) ITOL
      WRITE(6,6870) KT
      WRITE(6,6875) SAMPLE
C.......................................................................
C     READ AND PRINT THE INITIAL CONDITIONS
C.......................................................................
      READ(5,5300) (XSV(I),I=1,4)
      READ(5,5000) E4
      WRITE(6,6880) XSV(1)
      WRITE(6,6890) XSV(2)
      WRITE(6,6900) XSV(3)
      WRITE(6,6910) XSV(4)
      WRITE(6,6920) E4
C.......................................................................
C     READ AND PRINT THE NETWORK PARAMETERS
C.......................................................................
      READ(5,5000) RD1,RD2,RD3,RD4,RD5
```

```
      READ(5,5000) RD6,RDR,RDB,RDM,ROFF
      READ(5,5000) RQ1,RQ2,RQ3,RQ4,RQ5
      READ(5,5000) RQ6,RQB,RQM
      READ(5,5000) L1,L3,L4
      READ(5,5000) R7,R8,R9
      READ(5,5000)  R10,R3,L2,C1
      WRITE(6,6000)
      WRITE(6,6100) RD1,RD2,RD3,RD4,RD5,RD6,RDR,RDB,RDM
      WRITE(6,6200) RQ1,RQ2,RQ3,RQ4,RQ5,RQ6,RQB,RQM,ROFF
      WRITE(6,6300) R10,R3,L2,C1
      WRITE(6,6400) L1,L3,L4
      WRITE(6,6500) R7,R8,R9
C.....................................................................
C     READ AND PRINT THE NETWORK TOPOLOGY
C.....................................................................
      READ(5,5100) (NIO1(I),I=1,10)
      WRITE(6,6010)
      WRITE(6,6020) (NIO1(I),I=1,10)
C.....................................................................
C     READ AND PRINT THE CURRENT SOURCE INDUCTOR SATURATION CURVE.
C.....................................................................
      READ(5,5100) NDPTS
      READ(5,5400) (XI(I),I=1,NDPTS)
      READ(5,5400) (FXI(I),I=1,NDPTS)
      WRITE(6,6610)
      WRITE(6,6620) (I,XI(I),FXI(I),I=1,NDPTS)
C.....................................................................
C     READ FORMATS
C.....................................................................
 5000 FORMAT(5E14.7)
 5100 FORMAT(7I10)
 5300 FORMAT(5D14.7)
```

```
 5400 FORMAT(5D14.7)
C.......................................................................
C    WRITE FORMATS
C.......................................................................
 6800 FORMAT(1H1,T5,'THE POWER CONDITIONER AND MACHINE MODEL CONTROL PAR
     1AMETERS ARE:',/,1H ,T5,62('-'),//)
 6810 FORMAT(1H0,T5,'NUMBER OF INTEGRATIONS PER CYCLE 2 (NINT3) = ',I5)
 6820 FORMAT(1H0,T5,'STATE TRANSITION SERIES CONTROL (INTERM3) = ',I3)
 6830 FORMAT(1H0,T5,'INTEGRATION TIME STEP (TSNET) = ',E14.7,' SEC.')
 6840 FORMAT(1H0,T5,'STATE TRANSITION ERROR TOLERANCE (ETNET) = ',E14.7)
 6850 FORMAT(1H0,T5,'COMMUTATION SHIFT ANGLE (SHIFT) = ',E14.7,' ROTOR M
     1ECHANICAL RADIANS')
 6860 FORMAT(1H0,T5,'CHOPPER CURRENT TOLERANCE (ITOL) = ',E14.7,' AMPS')
 6870 FORMAT(1H0,T5,'MACHINE TORQUE CONSTANT (KT) = ',E14.7,' NT-M/AMP')
 6875 FORMAT(1H0,T5,'CHOPPER INDUCTANCE SAMPLING OPTION (SAMPLE): ',E14.
     17,' (SAMPLE.EQ.0-LINEAR CASE/SAMPLE.GT.0-INDUCTANCE STEP SIZE)')
 6880 FORMAT(1H0,T5,'INITIAL CAPACITOR VOLTAGE (XSV(1)) = ',D14.7,' VOLT
     1S')
 6890 FORMAT(1H0,T5,'INITIAL CHOPPER CURRENT (XSV(2)) = ',D14.7,' AMPS')
 6900 FORMAT(1H0,T5,'INITIAL PHASE B CURRENT (XSV(3)) = ',D14.7,' AMPS')
 6910 FORMAT(1H0,T5,'INITIAL PHASE C CURRENT (XSV(4)) = ',D14.7,' AMPS')
 6920 FORMAT(1H0,T5,'BATTARY VOLTAGE   (E4) = ',E14.7,' VOLTS')
 6000 FORMAT(1H1,T5,'THE NETWORK PARAMETERS IN OHMS,HENRIES AND FARADS
     1ARE:',/,1H ,T5,54('-'),//)
 6100 FORMAT(1H0,T5,'RD1 = ',E14.7,2X,'RD2 = ',E14.7,2X,
     1'RD3 = ',E14.7,2X,'RD4 = ',E14.7,2X,'RD5 = ',E14.7,//,
     2 1H ,T5,'RD6 = ',E14.7,2X,
     3                     'RDR = ',E14.7,2X,'RDB = ',E14.7,2X,
     4  'RDM = ',E14.7,///)
 6200 FORMAT(1H0,T5,'RQ1 = ',E14.7,2X,'RQ2 = ',E14.7,2X,'RQ3 = ',
     1 E14.7,2X,'RQ4 = ',E14.7,2X,'RQ5 = ',E14.7,//,1H ,T5,'RQ6 = ',
     2E14.7,2X,'RQB = ',E14.7,2X,'RQM = ',E14.7,2X,'RHI = ',E14.7,///)
```

```
6300 FORMAT(1H2,T5,'RE = ',E14.7,2X,'RCH = ',E14.7,2X,'LCH = ',
    1  E14.7,2X,'C = ',E14.7,///)
6400 FORMAT(1H0,T5,'L1 (PHASE A) = ',E14.7,'  L3 (PHASE B) = ',
    1 E14.7,'  L4 (PHASE C) = ',E14.7,//)
6500 FORMAT(1H0,T5,'R7 (PHASE A) = ',E14.7,'  R8 (PHASE B) = ',E14.7,
    1 '  R9 (PHASE C) = ',E14.7,////)
6010 FORMAT(1H0,T5,'THE NETWORK TOPOLOGY',/,1H ,T5,20('-')//)
6020 FORMAT(1H0,T5,'NBCHS = ',I3,'   NTWIGS = ',I3,'   NLINKS = ',I3,
    1'   NETWIG = ',I3,'   NCTWIG = ',I3,'   NRTWIG = ',I3,'   NLTWIG =
    2 ',I3,//,1H ,T5,   'NCLINK = ',I3,'   NRLINK = ',I3,
    3'   NLLINK = ',I3,///)
6610 FORMAT(1H0,T5,'THE CURVES OF THE NONLINEAR NETWORK PARAMETERS ARE:
    1',/,1H ,T5,50('-'),//)
6620 FORMAT(1H0,T5,'THE CHOPPER INDUCTOR CURRENT-INDUCTANCE DATA POINTS
    1 (AMPS,HENRIES) ARE:',//,
    2 (1H ,T5,'(',I3,')',2D16.7,'  (',I3,')',2D16.7))
     RETURN
     END
```

```
      SUBROUTINE VINPUT
C-----------------------------------------------------------------------
C
C     SUBROUTINE VINPUT READS AND PRINTS THE DATA FOR THE MECHANICAL-
C     POSITION LOOP.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION XV(5),UV(5),XVOLD(5),UVOLD(5),XVI(5),UVI(5)
      REAL KE,KRL,KV
      COMMON/BLK17 /  XV,UV,XVOLD,UVOLD,XVI,UVI
      COMMON/BLK24 /  NINT1,NINT2,NINT3,I1,I2,I3
      COMMON/BLK26 /  TSVLP,ETVLP,NTERM2
      COMMON/VLP   /  TAU1,TAU2,KE,KRL,KV,F1Y1,F1Y2,F1X1,F1X2,F2Y,F2X,
      READ(5,5000) TSVLP,ETVLP,TAU1,TAU2,KE
      READ(5,5000) KRL,KV,VHI,VLO
      READ(5,5000) F1Y1,F1Y2
      READ(5,5000) F2Y,F2X
      READ(5,5200) (XV(I),I=1,5)
C.......................................................................
C     PRINT VELOCITY LOOP DATA
C.......................................................................
      WRITE(6,6700)
      WRITE(6,6710) NINT2
      WRITE(6,6720) NTERM2
      WRITE(6,6730) TSVLP
      WRITE(6,6740) ETVLP
      WRITE(6,6750) TAU1
      WRITE(6,6760) TAU2
      WRITE(6,6770) KE
      WRITE(6,6780) KRL
      WRITE(6,6790) KV
      WRITE(6,6800) VHI
```

```
      WRITE(6,6810) VLO
      WRITE(6,6820) F1Y1
      WRITE(6,6830) F1Y2
      WRITE(6,6860) F2Y
      WRITE(6,6870) F2X
      WRITE(6,6875)
      WRITE(6,6880) XV(1)
      WRITE(6,6890) XV(2)
      WRITE(6,6900) XV(3)
      WRITE(6,6910) XV(4)
      WRITE(6,6920) XV(5)
C.......................................................................
C     READ FORMATS
C.......................................................................
 5000 FORMAT(5E14.7)
 5100 FORMAT(7I10)
 5200 FORMAT(5D14.7)
C.......................................................................
C     WRITE FORMATS
C.......................................................................
 6700 FORMAT(1H1,T5,'THE VELOCITY LOOP CONTROL PARAMETERS ARE:',/,1H ,T5
     1,40('-'),//)
 6710 FORMAT(1H0,T5,'NUMBER OF INTEGRATIONS PER CYCLE 1 (NINT2) = ',I5)
 6720 FORMAT(1H0,T5,'STATE TRANSITION SERIES CONTROL (NTERM2) = ',I3)
 6730 FORMAT(1H0,T5,'INTEGRATION TIME STEP (TSVLP) = ',E14.7,' SEC.')
 6740 FORMAT(1H0,T5,'STATE TRANSITION ERROR TOLERANCE (ETVLP) = ',E14.7)
 6750 FORMAT(1H0,T5,'POSITION ERROR AMPLIFIER TIME CONSTANT (TAU1) = ',
     1E14.7,' SEC.')
 6760 FORMAT(1H0,T5,'VELOCITY ERROR FB AMPLIFIER TIME CONSTANT (TAU2) =
     1',E14.7,' SEC.')
 6770 FORMAT(1H0,T5,'VELOCITY ERROR AMPLIFIER GAIN (KE) = ',E14.7)
 6780 FORMAT(1H0,T5,'CURRENT ERROR INTEGRATOR GAIN (KRL) = ',E14.7)
```

```
6790 FORMAT(1H0,T5,'VELOCITY FB AMPLIFIER GAIN (KV) = ',E14.7,' AMPS/RO
    1TOR RADIAN/SEC.')
6800 FORMAT(1H0,T5,'MAX (+) ROTOR VEL. DURING PLUGGING (VHI) = ',E14.7,
    1' RADIANS/SEC.')
6810 FORMAT(1H0,T5,'MAX (-) ROTOR VEL. DURING PLUGGING (VLO) = ',E14.7,
    1' RADIANS/SEC.')
6820 FORMAT(1H0,T5,'(MAX PLUGGING CURRENT)/5    (F1Y1) = ',E14.7,' AMPS'
    1)
6830 FORMAT(1H0,T5,'(MAX MOTORING OR REGENERATION CURRENT)/5    (F1Y2) =
    1 ',E14.7,' AMPS')
6860 FORMAT(1H0,T5,'CURRENT RATE LIMITER Y-COOR. (F2Y) = ',E14.7,' MACH
    1INE AMPS/5')
6870 FORMAT(1H0,T5,'CURRENT RATE LIMITER X-COOR. (F2X) = ',E14.7,' MACH
    1INE AMPS/5')
6875 FORMAT(1H0,/,1H0,T5,'THE INITIAL VELOCITY LOOP STATE VECTOR IS',/,
    11H ,T5,41('-'),//)
6880 FORMAT(1H0,T5,'ICMD/5 (XV(1)) = ',E14.7,' MACHINE AMPS/5')
6890 FORMAT(1H0,T5,'RATE AND MAGNITUDE LIMITED CURRENT COMMAND (ICMD1/5
    1=XV(2)) = ',E14.7,' MACHINE AMPS/5')
6900 FORMAT(1H0,T5,'VELOCITY FB-LOOP OUTPUT (XV(3)) = ',D14.7,' MACHINE
    1 AMPS/5')
6910 FORMAT(1H0,T5,'MAGNITUDE LIMITED COMMAND CURRENT (ICMDL/5=XV(4)) =
    1 ',D14.7,' MACHINE AMPS/5')
6920 FORMAT(1H0,T5,'CURRENT RATE LIMITER OUTPUT (XV(5)) = ',D14.7,' MAC
    1HINE AMPS/5')
     RETURN
     END
```

```
      SUBROUTINE MINPUT
C-----------------------------------------------------------------------
C
C     SUBROUTINE MINPUT READS AND PRINTS THE DATA FOR THE MECHANICAL-
C     POSITION LOOP
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  XM(5),UM(5),XMOLD(5),UMOLD(5),XMI(5),UMI(5)
      REAL N1,N2,N3
      REAL JM,JF,KACT,KF,KP,KERR
      COMMON/BLK20 /  XM,UM,XMOLD,UMOLD,XMI,UMI
      COMMON/BLK24 /  NINT1,NINT2,NINT3,I1,I2,I3
      COMMON/BLK25 /  TSMLP,ETMLP,NTERM1,NMACH,N1,N2,N3,NX
      COMMON/MLP   /  DC,VMSAT,TAU3,JM,JF,BF,KACT,KF,KP,KERR
C.......................................................................
C     READ MECHANICAL-POSITION LOOP DATA
C.......................................................................
      READ(5,5100) NMACH,NINT1,NTERM1
      READ(5,5000) TSMLP,ETMLP,TAU3,DC,VMSAT
      READ(5,5000) JM,JF,BF,KACT,KF
      READ(5,5000) KP,KERR,N1,N2,N3
      READ(5,5200) (XM(I),I=1,5)
C.......................................................................
C     PRINT MECHANICAL-POSITION LOOP DATA
C.......................................................................
      WRITE(6,6700)
      WRITE(6,6710) NMACH
      WRITE(6,6720) NINT1
      WRITE(6,6730) NTERM1
      WRITE(6,6740) TSMLP
      WRITE(6,6750) ETMLP
      WRITE(6,6760) TAU3
```

```
      WRITE(6,6770) DC
      WRITE(6,6780) VMSAT
      WRITE(6,6790) JH
      WRITE(6,6800) JF
      WRITE(6,6810) BF
      WRITE(6,6820) KACT
      WRITE(6,6830) KF
      WRITE(6,6840) KP
      WRITE(6,6850) KERR
      WRITE(6,6860) N1
      WRITE(6,6870) N2
      WRITE(6,6880) N3
      WRITE(6,6890)
      WRITE(6,6900) XM(1)
      WRITE(6,6910) XM(2)
      WRITE(6,6920) XM(3)
      WRITE(6,6930) XM(4)
      WRITE(6,6940) XM(5)
C.....................................................................
C     READ FORMATS
C.....................................................................
 5000 FORMAT(5E14.7)
 5100 FORMAT(7I10)
 5200 FORMAT(5D14.7)
C.....................................................................
C     WRITE FORMATS
C.....................................................................
 6700 FORMAT(1H1,T5,'THE MECHANICAL-POSITION LOOP CONTROL PARAMETERS ARE
     1:',/,1H ,T5,51('-'),//)
 6710 FORMAT(1H0,T5,'NUMBER OF ACTIVE MACHINES (NMACH=1.OR.2) = ',I1)
 6720 FORMAT(1H0,T5,'NUMBER OF INTEGRATIONS (NINT1) = ',I5)
 6730 FORMAT(1H0,T5,'STATE TRANSITION SERIES CONTROL (NTERM1) = ',I3)
```

```
6740 FORMAT(1H0,T5,'INTEGRATION TIME STEP (TSMLP) = ',E14.7,' SEC.')
6750 FORMAT(1H0,T5,'STATE TRANSITION ERROR TOLERANCE (ETMLP) = ',E14.7)
6760 FORMAT(1H0,T5,'POSITION ERROR AMPLIFIER TIME CONSTANT (TAU3) = ',
    1E14.7,' SEC.')
6770 FORMAT(1H0,T5,'DEFLECTION COMMAND (DC) = ',E14.7,' FLAP DEGREES')
6780 FORMAT(1H0,T5,'ROTOR VELOCITY LIMIT-SIMPLE MACHINE MODEL (VMSAT) =
    1 ',E14.7,' ROTOR RADIANS/SEC.')
6790 FORMAT(1H0,T5,'MACHINE AND REFLECTED GEAR INERTIA (JM) = ',E14.7,
    1' KG-M**2')
6800 FORMAT(1H0,T5,'FLAP INERTIA (JF) = ',E14.7,' KG-M**2')
6810 FORMAT(1H0,T5,'FLAP VISCOUS DAMPING COEFFICIENT (BF) = ',E14.7,
    1' NT-M/FLAP RAD/SEC')
6820 FORMAT(1H0,T5,'ACTUATOR MOUNT STIFFNESS COEFFICIENT (KACT) = ',
    1E14.7,' NT-M/EMA OUTPUT RAD')
6830 FORMAT(1H0,T5,'FLAP STIFFNESS COEFFICIENT (KF) = ',E14.7,
    1 ' NT-M/FLAP RAD')
6840 FORMAT(1H0,T5,'POSITION FB-LOOP GAIN (KP) = ',E14.7,' DEG/RAD')
6850 FORMAT(1H0,T5,'POSITION ERROR AMPLIFIER GAIN (KERR) = ',E14.7,
    1 ' MACHINE AMPS/5/DEGREE FLAP')
6860 FORMAT(1H0,T5,'GEAR RATIO (N1) = ',E14.7)
6870 FORMAT(1H0,T5,'GEAR RATIO (N2) = ',E14.7)
6880 FORMAT(1H0,T5,'GEAR RATIO (N3) = ',E14.7//)
6890 FORMAT(1H0,T5,'THE INITIAL MECHANICAL-POSITION LOOP STATE VECTOR
    1 ',/,1H ,T5,49('-'),//)
6900 FORMAT(1H0,T5,'ROTOR VELOCITY (XM(1)) = ',D14.7,' RADIANS/SEC.')
6910 FORMAT(1H0,T5,'EMA ACTUATOR OUTPUT TORQUE (XM(2)) = ',E14.7,' NT-M
    1')
6920 FORMAT(1H0,T5,'FLAP VELOCITY (XM(3)) = ',D14.7,' RADIANS/SEC.')
6930 FORMAT(1H0,T5,'FLAP POSITION (XM(4)) = ',D14.7,' RADIANS')
6940 FORMAT(1H0,T5,'AMPLIFIED POSITION ERROR SIGNAL (XM(5)) = ',D14.7,
    1' MACHINE AMPS/5')
     RETURN
```

END

```
      SUBROUTINE NSETUP
C-----------------------------------------------------------------------
C
C     SUBROUTINE NSETUP GENERATES THE SYSTEM AND TRANSITION MATRICES FOR
C     THE MACHINE-POWER CONDITIONER MODEL.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION XV(5),UV(5),XVOLD(5),UVOLD(5),XVI(5),UVI(5)
      DOUBLE PRECISION  XM(5),UM(5),XMOLD(5),UMOLD(5),XMI(5),UMI(5)
      INTEGER PGMODE
      REAL N1,N2,N3,KT
      REAL ICMDL,ICMD1,IMC,ITOL,IM
      COMMON/BLK1  /  CIA,CIB,CIC,VA,VB,VC,VAB,VBC,VCA,PA,PB,PC,PCORE,
     1                PML,PTRM,PSO,PEM,PGM,PNM,TEM,TM
      COMMON/BLK6  /  RRANG,RANG,RVEL,RACEL
      COMMON/BLK7/  ICMDL,ICMD1,IMC,IM
      COMMON/BLK17 /  XV,UV,XVOLD,UVOLD,XVI,UVI
      COMMON/BLK20 /  XM,UM,XMOLD,UMOLD,XMI,UMI
      COMMON/BLK21 /  ITOL,SHIFT,KT
      COMMON/BLK22 /  PGMODE,IOMODE,IDISK,IPLT
      COMMON/BLK25 /  TSMLP,ETMLP,NTERM1,NMACH,N1,N2,N3,NX
      COMMON/BLK27 /  TSNET,ETNET,NTERM3,DIFMAX,ITER,TIME
      COMMON/BLK32 /  NTRMS1,NTRMS2,NTRMS3,DIFMX1,DIFMX2,DIFMX3
      COMMON/BLK33 /  SAMPLE,NNETSW
C.......................................................................
C     INITIALIZE:  TIME,NNETSW,NTRMS3,DIFMX3
C.......................................................................
      TIME=0.
      NNETSW=0
      NTRMS3=0
      DIFMX3=0.
C.......................................................................
```

```
C     CALCULATE:   ICMDL,ICMD1,IMC,NX,RANG,RVEL
C......................................................................
      ICMDL=XV(2)*5.
      ICMD1=XV(4)*5.
      IMC=ABS(ICMD1)
      NX=1
      IF(NMACH.EQ.1) NX=2
      RANG=XM(4)*2.*N1*N2*N3*NX
      RVEL=XM(1)
C......................................................................
C     DETERMINE EMA MODE
C......................................................................
      CALL EMAMOD
C......................................................................
C     INITIALIZE THE SIMPLE NETWORK MODEL
C......................................................................
      IM=IMC
      TM=KT*ICMD1
      IF(PGMODE.EQ.2) RETURN
C......................................................................
C     INITIALIZE THE DETAILED NETWORK MODEL
C......................................................................
      CALL INITL
C......................................................................
C     DETERMINE THE INITIAL NETWORK CONFIGURATION AND GENERATE THE
C     SYSTEM AND TRANSITION MATRICES.
C......................................................................
      CALL NETWRK
      CALL NTJUMP
      CALL TORPOW
      CALL SWLOSS
      RETURN
```

END

```
      SUBROUTINE VSETUP
C-----------------------------------------------------------------------
C
C     SUBROUTINE VSETUP GENERATES THE SYSTEM AND TRANSITION MATRICES
C     FOR THE VELOCITY LOOP.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  AV(5,5),BV(5,5),PHIV(5,5),THETAV(5,5)
      DOUBLE PRECISION XV(5),UV(5),XVOLD(5),UVOLD(5),XVI(5),UVI(5)
      DOUBLE PRECISION  XM(5),UM(5),XMOLD(5),UMOLD(5),XMI(5),UMI(5)
      INTEGER PGMODE
      LOGICAL PLUG
      LOGICAL SPDHI,SPDLO,SPDPOS,SPDNEG,TPOS,TNEG,MTRG1,RGN4,PLUG4,
     1        MTRG3,RGN2,PLUG2
      REAL N1,N2,N3,KT
      REAL ICMDL,ICMD1,IMC,IM
      REAL KE,KRL,KV
      COMMON/BLK7/  ICMDL,ICMD1,IMC,IM
      COMMON/BLK16 /  AV,BV,PHIV,THETAV
      COMMON/BLK17 /  XV,UV,XVOLD,UVOLD,XVI,UVI
      COMMON/BLK20 /  XM,UM,XMOLD,UMOLD,XMI,UMI
      COMMON/BLK26 /  TSVLP,ETVLP,NTERM2
      COMMON/BLK32 /  NTRMS1,NTRMS2,NTRMS3,DIFMX1,DIFMX2,DIFMX3
      COMMON/BLK33 /  SAMPLE,NNETSW
      COMMON/MODE  /  SPDHI,SPDLO,SPDPOS,SPDNEG,TPOS,TNEG,MTRG1,RGN4,
     1                PLUG4,MTRG3,RGN2,PLUG2
      COMMON/VLP   /  TAU1,TAU2,KE,KRL,KV,F1Y1,F1Y2,F1X1,F1X2,F2Y,F2X,
     1                VHI,VLO
C.......................................................................
C     INITIALIZE AV,BV,PHIV,THETAV AND UV TO ZERO
C     INITIALIZE XVOLD TO XV
C     INITIALIZE XVI TO XV
```

```
C.......................................................................
      DO 10 I=1,5
      UV(I)=0.D0
      XVOLD(I)=XV(I)
      XVI(I)=XV(I)
      DO 10 J=1,5
      AV(I,J)=0.D0
      BV(I,J)=0.D0
      PHIV(I,J)=0.D0
      THETAV(I,J)=0.D0
   10 CONTINUE
C.......................................................................
C     FORM AV AND BV
C.......................................................................
      AV(1,1)=-1./TAU1
      AV(1,3)=-KE/TAU1
      AV(2,5)=KRL
      AV(3,3)=-1./TAU2
      BV(1,1)=KE/TAU1
      BV(3,2)=KV/TAU2
C.......................................................................
C     INITIALIZE THE FORCING FUNCTION VECTOR UV
C.......................................................................
      UV(1)=XM(5)
      UV(2)=XM(1)
C.......................................................................
C     JUMP THE CLAMPED VARIABLES FROM TIME=0-   TO  TIME=0+
C.......................................................................
      PLUG=PLUG2.OR.PLUG4
      ARG1=XV(1)
      ARG2=XV(4)-XV(2)
      XV(4)=FUNC1(ARG1,F1Y1,F1Y2,PLUG)
```

```
      XV(5)=FUNC2(ARG2,F2Y,F2X)
C.......................................................................
C     GENERATE PHIV AND THETAV
C.......................................................................
      CALL STRAN(TSVLP,PHIV,THETAV,AV,BV,5,5,5,2,1,NTERM2,ETVLP,
     1 DIFMX2,NTRMS2,IFLAG)
C.......................................................................
C     PRINT COMPUTED RESUTLS
C.......................................................................
      WRITE(6,6000)
      WRITE(6,6010) TSVLP
      WRITE(6,6020) ETVLP
      WRITE(6,6030) DIFMX2
      WRITE(6,6040) NTRMS2
      CALL DMIO(AV,5,5,5,5,1,2)
      CALL DMIO(BV,5,5,5,5,2,2)
      CALL DMIO(PHIV,5,5,5,5,3,2)
      CALL DMIO(THETAV,5,5,5,5,4,2)
      WRITE(6,6050)
      CALL DVIO(XV,5,5,5,2)
      CALL DVIO(UV,5,5,6,2)
C.......................................................................
C     FORMATS
C.......................................................................
 6000 FORMAT(1H1,T5,'THE CALCULATED VELOCITY LOOP DATA',/,1H ,T5,33('-')
     1,//)
 6010 FORMAT(1H0,T5,'INTEGRATION TIME STEP (TSVLP) = ',E14.7,' SEC.')
 6020 FORMAT(1H0,T5,'STATE TRANSITION ERROR TOLERANCE (ETVLP) = ',E14.7)
 6030 FORMAT(1H0,T5,'ACTUAL MAXIMUM STATE TRANSITION ERROR (DIFMX2) = ',
     1E14.7)
 6040 FORMAT(1H0,T5,'NUMBER OF TERMS (NTRMS2) USED TO ACHIEVE ERROR TOLE
     1RANCE = ',I3)
```

```
6050 FORMAT(1H1)
     RETURN
     END
```

```
      SUBROUTINE MSETUP
C-----------------------------------------------------------------------
C
C     SUBOUTINE MSETUP GENERATES THE SYSTEM AND TRANSITION MATRICES
C     FOR THE MECHANICAL-POSITION LOOPS.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  AM(5,5),BM(5,5),PHIM(5,5),THETAM(5,5)
      DOUBLE PRECISION  XM(5),UM(5),XMOLD(5),UMOLD(5),XMI(5),UMI(5)
      DOUBLE PRECISION XV(5),UV(5),XVOLD(5),UVOLD(5),XVI(5),UVI(5)
      INTEGER PGMODE
      REAL N1,N2,N3
      REAL JM,JF,KACT,KF,KP,KERR
      COMMON/BLK1  /  CIA,CIB,CIC,VA,VB,VC,VAB,VBC,VCA,PA,PB,PC,PCORE,
     1                PML,PTRM,PSO,PEM,PGM,PNM,TEM,TM
      COMMON/BLK17 /  XV,UV,XVOLD,UVOLD,XVI,UVI
      COMMON/BLK18 /  AM,BM,PHIM,THETAM
      COMMON/BLK20 /  XM,UM,XMOLD,UMOLD,XMI,UMI
      COMMON/BLK25 /  TSMLP,ETMLP,NTERM1,NMACH,N1,N2,N3,NX
      COMMON/BLK28 /  FANG,PE,VE,TACT
      COMMON/BLK32 /  NTRMS1,NTRMS2,NTRMS3,DIFMX1,DIFMX2,DIFMX3
      COMMON/MLP   /  DC,VMSAT,TAU3,JM,JF,BF,KACT,KF,KP,KERR
C.......................................................................
C     INITIALIZE AM,BM,PHIM,THETAM AND UM TO ZERO
C     INITIALIZE XMOLD TO XM
C     INITIALIZE XMI TO XM
C.......................................................................
      DO 10 I=1,5
      UM(I)=0.D0
      XMOLD(I)=XM(I)
      XMI(I)=XM(I)
      DO 10 J=1,5
```

```
      AM(I,J)=0.D0
      BM(I,J)=0.D0
      PHIM(I,J)=0.D0
      THETAM(I,J)=0.D0
   10 CONTINUE
C.....................................................................
C    CALCULATE:  NX,FANG,PE,TACT,VE
C.....................................................................
      NX=1
      IF(NMACH.EQ.1) NX=2
      FANG=XM(4)*57.29578
      PE=XM(5)
      TACT=XM(2)
      VE=PE-XV(3)
C.....................................................................
C    FORM AM AND BM
C.....................................................................
      AM(1,2)=-1./(4*N1*N2*JM)
      AM(2,1)=KACT/(2.*N1*N2*NX)
      AM(2,3)=-N3*KACT
      AM(3,2)=N3/JF
      AM(3,3)=-BF/JF
      AM(3,4)=-KF/JF
      AM(4,3)=1.D0
      AM(5,4)=-KP*KERR/TAU3
      AM(5,5)=-1./TAU3
      BM(1,1)=1./JM
      BM(5,2)=KERR/TAU3
C.....................................................................
C    INITIALIZE THE FORCING FUNCTION VECTOR UM
C.....................................................................
      UM(1)=TM
```

```
      UM(2)=DC
C.......................................................................
C     GENERATE PHIM AND THETAM
C.......................................................................
      CALL STRAN(TSMLP,PHIM,THETAM,AM,BM,5,5,5,2,1,NTERM1,ETMLP,
     1 DIFMX1,NTRMS1,IFLAG)
C.......................................................................
C     PRINT COMPUTED RESULTS
C.......................................................................
      WRITE(6,6000)
      WRITE(6,6010) TSMLP
      WRITE(6,6020) ETMLP
      WRITE(6,6030) DIFMX1
      WRITE(6,6040) NTRMS1
      CALL DMIO(AM,5,5,5,5,7,2)
      CALL DMIO(BM,5,5,5,5,8,2)
      CALL DMIO(PHIM,5,5,5,5,9,2)
      CALL DMIO(THETAM,5,5,5,5,10,2)
      WRITE(6,6050)
      CALL DVIO(XM,5,5,11,2)
      CALL DVIO(UM,5,5,12,2)
C.......................................................................
C     FORMATS
C.......................................................................
 6000 FORMAT(1H1,T5,'THE CALCULATED POSITION AND MECHANICAL LOOP DATA',/
     1 1H ,T5,48('-')//)
 6010 FORMAT(1H0,T5,'INTEGRATION TIME STEP (TSMLP) = ',E14.7,' SEC.')
 6020 FORMAT(1H0,T5,'STATE TRANSITION ERROR TOLERANCE (ETMLP) = ',E14.7)
 6030 FORMAT(1H0,T5,'ACTUAL MAXIMUM STATE TRANSITION ERROR (DIFMX1) = ',
     1E14.7)
 6040 FORMAT(1H0,T5,'NUMBER OF TERMS USED TO ACHIEVE ERROR TOLERANCE (NT
     1RMS1) = ',I3)
```

```
6050 FORMAT(1H1)
     RETURN
     END
```

```
      SUBROUTINE EMAMOD
C-----------------------------------------------------------------------
C
C     SUBROUTINE EMAMOD DETERMINES THE EMA MODE OF OPERATION
C
C-----------------------------------------------------------------------
      LOGICAL SPDHI,SPDLO,SPDPOS,SPDNEG,TPOS,TNEG,MTRG1,RGN4,PLUG4,
     1        MTRG3,RGN2,PLUG2
      REAL KE,KRL,KV
      REAL ICMDL,ICMD1,IMC,IM
      COMMON/BLK6  /  RRANG,RANG,RVEL,RACEL
      COMMON/BLK7/  ICMDL,ICMD1,IMC,IM
      COMMON/MODE  /  SPDHI,SPDLO,SPDPOS,SPDNEG,TPOS,TNEG,MTRG1,RGN4,
     1                PLUG4,MTRG3,RGN2,PLUG2
      COMMON/VLP   /  TAU1,TAU2,KE,KRL,KV,F1Y1,F1Y2,F1X1,F1X2,F2Y,F2X,
     1                VHI,VLO
C.......................................................................
C     DETERMINE THE COMPARATOR OUTPUTS
C.......................................................................
      SPDHI=RVEL.GE.VHI
      SPDLO=RVEL.LE.VLO
      SPDPOS=RVEL.GE.0.
      SPDNEG=RVEL.LT.0.
      TPOS=ICMD1.GE.0.
      TNEG=ICMD1.LT.0.
C.......................................................................
C     DETERMINE THE MODE
C.......................................................................
      MTRG1=TPOS.AND.SPDPOS
      RGN4=TNEG.AND.SPDHI
      PLUG4=TNEG.AND.SPDPOS.AND.(.NOT.SPDHI)
      MTRG3=TNEG.AND.SPDNEG
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
RUN2=TPOS.AND.SPDLO
PLUG2=TPOS.AND.SPDNEG.AND.(.NOT.SPDLO)
RETURN
END
```

```
      SUBROUTINE MLPMOD
C-----------------------------------------------------------------------
C
C     SUBROUTINE MLPMOD INTEGRATES THE MECHANICAL LOOP MODEL FORWARD IN
C     TIME.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  AM(5,5),BM(5,5),PHIM(5,5),THETAM(5,5)
      DOUBLE PRECISION  XM(5),UM(5),XMOLD(5),UMOLD(5),XMI(5),UMI(5)
      INTEGER PGMODE
      REAL JM,JF,KACT,KF,KP,KERR
      COMMON/BLK1   /  CIA,CIB,CIC,VA,VB,VC,VAB,VBC,VCA,PA,PB,PC,PCORE,
     1                 PML,PTRM,PSO,PEM,PGM,PNM,TEM,TM
      COMMON/BLK18 /   AM,BM,PHIM,THETAM
      COMMON/BLK20 /   XM,UM,XMOLD,UMOLD,XMI,UMI
      COMMON/BLK22 /   PGMODE,IOMODE,IDISK,IPLT
      COMMON/MLP   /   DC,VMSAT,TAU3,JM,JF,BF,KACT,KF,KP,KERR
C.......................................................................
C     UPDATE UM
C.......................................................................
      UM(1)=TM
      UM(2)=DC
C.......................................................................
C     SAVE THE PRESENT STATE VARIABLES AND INPUTS
C.......................................................................
      DO 10 I=1,5
      UMOLD(I)=UM(I)
      XMOLD(I)=XM(I)
   10 CONTINUE
C.......................................................................
C     INTEGRATE THE MECHANICAL LOOP FORWARD BY ONE TIME STEP
C.......................................................................
```

```
      CALL FUTURE(XM,UM,PHIM,THETAM,5,5,5,5,5,5,5,2,1)
C.......................................................................
C     CLAMP THE MOTOR VELOCITY AT (+ OR -) VMSAT IF THE SIMPLE MACHINE
C     MODEL IS USED.
C.......................................................................
      IF(PGMODE.NE.2) RETURN
      IF(XM(1).LT.-VMSAT) XM(1)=-VMSAT
      IF(XM(1).GT.VMSAT) XM(1)=VMSAT
      RETURN
      END
```

```
      SUBROUTINE VLPMOD
C-----------------------------------------------------------------------
C
C     SUBROUTINE VLPMOD INTEGRATES THE VELOCITY LOOP MODEL FORWARD IN
C     TIME.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  AV(5,5),BV(5,5),PHIV(5,5),THETAV(5,5)
      DOUBLE PRECISION XV(5),UV(5),XVOLD(5),UVOLD(5),XVI(5),UVI(5)
      DOUBLE PRECISION  XM(5),UM(5),XMOLD(5),UMOLD(5),XMI(5),UMI(5)
      LOGICAL PLUG
      LOGICAL SPDHI,SPDLO,SPDPOS,SPDNEG,TPOS,TNEG,MTRG1,RGN4,PLUG4,
     1        MTRG3,RGN2,PLUG2
      REAL KE,KRL,KV
      COMMON/BLK16 /  AV,BV,PHIV,THETAV
      COMMON/BLK17 /  XV,UV,XVOLD,UVOLD,XVI,UVI
      COMMON/BLK20 /  XM,UM,XMOLD,UMOLD,XMI,UMI
      COMMON/MODE  /  SPDHI,SPDLO,SPDPOS,SPDNEG,TPOS,TNEG,MTRG1,RGN4,
     1                PLUG4,MTRG3,RGN2,PLUG2
      COMMON/VLP   /  TAU1,TAU2,KE,KRL,KV,F1Y1,F1Y2,F1X1,F1X2,F2Y,F2X,
     1                VHI,VLO
C.......................................................................
C     UPDATE THE INPUT VECTOR UV
C.......................................................................
      CALL SVINTP(1)
      UV(1)=XMI(5)
      UV(2)=XMI(1)
C.......................................................................
C     SAVE THE PRESENT STATE VARIABLES AND INPUTS
C.......................................................................
      DO 10 I=1,5
      UVOLD(I)=UV(I)
```

```
      XVOLD(I)=XV(I)
   10 CONTINUE
C.....................................................................
C     INTEGRATE THE VELOCITY LOOP FORWARD BY ONE TIME STEP
C.....................................................................
       CALL FUTURE(XV,UV,PHIV,THETAV,5,5,5,5,5,5,5,2,1)
C.....................................................................
C     JUMP THE CLAMPED VARIABLES FORM TIME=K-  TO   TIME=K+
C.....................................................................
      PLUG=PLUG2.OR.PLUG4
      ARG1=XV(1)
      ARG2=XV(4)-XV(2)
      XV(4)=FUNC1(ARG1,F1Y1,F1Y2,PLUG)
      XV(5)=FUNC2(ARG2,F2Y,F2X)
      RETURN
      END
```

```
      SUBROUTINE NETMOD
C-----------------------------------------------------------------------
C
C     SUBROUTINE NETMOD INTEGRATES THE SIMPLIFIED OR DETAILED MACHINE-
C     POWER CONDITIONER NETWORK MODEL FORWARD IN TIME.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION A(4,4),B(4,4),PHI(4,4),THETA(4,4)
      DOUBLE PRECISION XSV(4),U(4),XSVDOT(4)
      DOUBLE PRECISION XV(5),UV(5),XVOLD(5),UVOLD(5),XVI(5),UVI(5)
      DOUBLE PRECISION  XM(5),UM(5),XMOLD(5),UMOLD(5),XMI(5),UMI(5)
      INTEGER PGMODE
      REAL  ICMDL,ICMD1,IMC,IM,ITOL
      REAL N1,N2,N3,KT
      COMMON/BLK1  /  CIA,CIB,CIC,VA,VB,VC,VAB,VBC,VCA,PA,PB,PC,PCORE,
     1                PML,PTRM,PSO,PEM,PGM,PNM,TEM,TM
      COMMON/BLK6  /  RRANG,RANG,RVEL,RACEL
      COMMON/BLK7/  ICMDL,ICMD1,IMC,IM
      COMMON/BLK14 /  A,B,PHI,THETA
      COMMON/BLK15 /  XSV,U,XSVDOT
      COMMON/BLK17 /  XV,UV,XVOLD,UVOLD,XVI,UVI
      COMMON/BLK20 /  XM,UM,XMOLD,UMOLD,XMI,UMI
      COMMON/BLK21 /  ITOL,SHIFT,KT
      COMMON/BLK22 /  PGMODE,IOMODE,IDISK,IPLT
      COMMON/BLK23 /  NCALLS,NREC,NUMINT,NPTS
      COMMON/BLK24 /  NINT1,NINT2,NINT3,I1,I2,I3
      COMMON/BLK25 /  TSMLP,ETMLP,NTERM1,NMACH,N1,N2,N3,NX
      COMMON/BLK26 /  TSVLP,ETVLP,NTERM2
      COMMON/BLK27 /  TSNET,ETNET,NTERM3,DIFMAX,ITER,TIME
      COMMON/BLK28 /  FANG,PE,VE,TACT
C.......................................................................
C     CHOOSE BETWEEN THE SIMPLE AND DETAILED NETWORK MODELS
```

```
C.......................................................................
      GO TO (200,100,200),PGMODE
C.......................................................................
C     SIMPLIFIED NETWORK MODEL
C.......................................................................
  100 CONTINUE
C     ------------------------------------------------------------------
C     UPDATE:  ICMDL,ICMD1,IMC,IM,TM,TEM,FANG,PE,TACT,VE,TIME,NPTS,NUMINT
C              RVEL,RANG
C
      CALL SVINTP(1)
      ICMDL=XV(2)*5.
      ICMD1=XV(4)*5.
      IMC=ABS(ICMD1)
      IM=IMC
      TM=KT*ICMD1
      TEM=TM
      FANG=XMI(4)*57.29578
      PE=XMI(5)
      TACT=XMI(2)
      VE=PE-XV(3)
      TIME=(I1-1)*TSMLP+I2*TSVLP
      NPTS=(I1-1)*NINT2+I2+1
      NUMINT=NPTS-1
      RVEL=XMI(1)
      RANG=RVEL*TIME
C     ------------------------------------------------------------------
C     DETERMINE EMA MODE
C
      CALL EMAMOD
C     ------------------------------------------------------------------
C     PRINT RESULTS OF THE I2-ND TIME SLICE IF SPECIFIED
```

```
C
      IF(IOMODE) 50,50,30
   30 IREM=MOD(NUMINT,IOMODE)
      IF(IREM.NE.0) GO TO 50
      CALL OUTPUT(PGMODE,NUMINT,NCALLS)
   50 CONTINUE
C     ------------------------------------------------------------
C     LOAD PLOT DATA ON DISK IF SPECIFIED
C
      IF(IDISK) 90,90,70
   70 IREM=MOD(NUMINT,IDISK)
      IF(IREM.NE.0) RETURN
      CALL PDATA(PGMODE,NREC,0)
   90 CONTINUE
      RETURN
C.................................................................
C     DETAILED NETWORK
C.................................................................
  200 CONTINUE
      DO 1000 K=1,NINT3
      I3=K
C     ------------------------------------------------------------
C     INTEGRATE NETWORK
C
      CALL FUTURE(XSV,U,PHI,THETA,4,4,4,4,4,4,4,4,1)
C     ------------------------------------------------------------
C     UPDATE:  TIME,NPTS,NUMINT
C
      IF(PGMODE.EQ.1) TIME=I3*TSNET
      IF(PGMODE.EQ.1) NPTS=I3+1
      IF(PGMODE.EQ.3) TIME=(I1-1)*TSMLP+(I2-1)*TSVLP+I3*TSNET
      IF(PGMODE.EQ.3) NPTS=(I1-1)*NINT2*NINT3+(I2-1)*NINT3+I3+1
```

```
      NUMINT=NPTS-1
C     ---------------------------------------------------------------
C     UPDATE MACHINE PHASE CURRENTS
C
      CIA=-(XSV(3)+XSV(4))
      CIB=XSV(3)
      CIC=XSV(4)
C     ---------------------------------------------------------------
C     UPDATE:  RANG,RVEL,ICMDL,ICMD1,IMC,FANG,PE,TACT,VE
C     USING LINEAR INTRA-SAMPLE INTERPOLATION OF XM AND XV
C
      GO TO (500,400,400),PGMODE
  400 CONTINUE
      CALL SVINTP(2)
      RANG=XMI(4)*2*N1*N2*N3*NX
      RVEL=XMI(1)
      ICMDL=XVI(2)*5.
      ICMD1=XVI(4)*5.
      IMC=ABS(ICMD1)
      FANG=XMI(4)*57.29578
      PE=XMI(5)
      TACT=XMI(2)
      VE=PE-XVI(3)
      GO TO 550
  500 CONTINUE
      RANG=RVEL*TIME
  550 CONTINUE
C     ---------------------------------------------------------------
C     DETERMINE EMA MODE
C
      CALL EMAMOD
C     ---------------------------------------------------------------
```

```
C     JUMP NETWORK FORM TIME=I3-  TO  TIME=I3+
C
      CALL NTJUMP
C     ------------------------------------------------------------
C     DETERMINE MACHINE TORQUES AND POWERS
C
      CALL TORPOW
C     ------------------------------------------------------------
C     DETERMINE POWER LOSSES IN POWER DIODES AND TRANSISTORS
C     DURING CONDUCTION
C
      CALL SWLOSS
C     ------------------------------------------------------------
C     PRINT RESULTS OF I3-RD TIME SLICE IF SPECIFIED
C
      IF(IOMODE) 700,700,675
  675 IREM=MOD(NUMINT,IOMODE)
      IF(IREM.NE.0) GO TO 700
      CALL OUTPUT(PGMODE,NUMINT,NCALLS)
  700 CONTINUE
C     ------------------------------------------------------------
C     LOAD PLOT DATA ON DISK IF SPECIFIED
C
      IF(IDISK) 900,900,875
  875 IREM=MOD(NUMINT,IDISK)
      IF(IREM.NE.0) GO TO 900
      IF(IDISK) 900,900,800
  800 CALL PDATA(PGMODE,NREC,0)
  900 CONTINUE
 1000 CONTINUE
      RETURN
      END
```

```
      REAL FUNCTION FUNC1(X,F1Y1,F1Y2,PLUG)
C-----------------------------------------------------------------------
C
C     FUNCTION FUNC1 SIMULATES THE COMMAND CURRENT MAGNITUDE LIMITER
C
C-----------------------------------------------------------------------
      LOGICAL PLUG
      IF(PLUG) GO TO 500
      FUNC1=X
      IF(X.GE.F1Y2) FUNC1=F1Y2
      IF(X.LE.-F1Y2) FUNC1=-F1Y2
      RETURN
  500 CONTINUE
      FUNC1=X
      IF(X.GE.F1Y1) FUNC1=F1Y1
      IF(X.LE.-F1Y1) FUNC1=-F1Y1
      RETURN
      END
```


C-2

```
      REAL FUNCTION FUNC2(X,F2Y,F2X)
C-----------------------------------------------------------------------
C
C     FUNCTION FUNC2 SIMULATES THE COMMAND CURRENT RATE LIMITER
C
C-----------------------------------------------------------------------
      SLOPE=F2Y/F2X
      FUNC2=SLOPE*X
      IF(X.GE.F2X) FUNC2=F2Y
      IF(X.LE.-F2Y) FUNC2=-F2Y
      RETURN
      END
```

```
      SUBROUTINE INITL
C-----------------------------------------------------------------------
C
C     SUBROUTINE INITL INITIALIZES THE REQUIRED PARAMETERS OF THE
C     MACHINE-POWER CONDITIONER NETWORK
C
C-----------------------------------------------------------------------
      DIMENSION        IQ(15,8),NIO1(10)
      DOUBLE PRECISION BCHCUR(23),BCHEMF(23),BCHPOW(23)
      DOUBLE PRECISION XSV(4),U(4),XSVDOT(4)
      DOUBLE PRECISION A(4,4),B(4,4),PHI(4,4),THETA(4,4)
      INTEGER PGMODE
      LOGICAL          AA,BB,CC,FF,AN,BN,CN,FN,QAP1,QAN1,QBP1,QBN1,QCP1,
     1                 QCN1,QMON,QBON,QIOFF,Q(8),NETCHG,NDUMP,D(9)
     2                ,QBC,QMC
      REAL             BRANCH(23),L1,L2,L3,L4
      REAL  ITOL,KT
      COMMON/BLK1  /   CIA,CIB,CIC,VA,VB,VC,VAB,VBC,VCA,PA,PB,PC,PCORE,
     1                 PML,PTRM,PSO,PEM,PGM,PNM,TEM,TM
      COMMON/BLK2  /   BCHCUR,BCHEMF,BCHPOW
      COMMON/BLK3  /   RQ1,RQ2,RQ3,RQ4,RQ5,RQ6,RQM,RQB,ROFF
      COMMON/BLK8  /   AA,BB,CC,FF,AN,BN,CN,FN,QAP1,QAN1,QBP1,QBN1,QCP1,
     1                 QCN1,QMON,QBON,QIOFF,Q,NETCHG,NDUMP,D,QBC,QMC
      COMMON/BLK9  /   E1,E2,E3,E4,C1,R1,R2,R3,R4,R5,R6,R7,R8,R9,L1,R10,
     1                 R11,R12,R13,R14,L2,L3,L4
      COMMON/BLK10 /   NBCHS,NTWIGS,NLINKS,NETWIG,NCTWIG,NRTWIG,NLTWIG,
     1                 NCLINK,NRLINK,NLLINK,IQ
      COMMON/BLK14 /   A,B,PHI,THETA
      COMMON/BLK15 /   XSV,U,XSVDOT
      COMMON/BLK21 /   ITOL,SHIFT,KT
      COMMON/BLK27 /   TSNET,ETNET,NTERM3,DIFMAX,ITER,TIME
      COMMON/BLK31 /   CDMC,CDBC,XDM,XDB,ISM,ISB
```

```
      COMMON/BLK32 /  NTRMS1,NTRMS2,NTRMS3,DIFMX1,DIFMX2,DIFMX3
      EQUIVALENCE     (BRANCH(1),E1)
      EQUIVALENCE     (NIO1(1),NBCHS)
C...................................................................
C     INITIALIZE THE CHOPPER HYSTERESIS LOOPS
C...................................................................
      CALL SQBC(DUMMY1,XOB,ISB,ITOL,QBC,1)
      CALL SQMC(DUMMY2,XOM,ISM,ITOL,QMC,1)
C...................................................................
C     LOAD THE CUTSET MATRIX
      DO 120 I=1,15
      DO 120 J=1,8
  120 IQ(I,J)=0
      IQ(1,7)=1
      IQ(1,8)=1
      IQ(2,7)=-1
      IQ(3,8)=-1
      IQ(4,1)=-1
      IQ(5,1)=1
      IQ(5,2)=1
      IQ(5,3)=1
      IQ(5,4)=1
      IQ(5,5)=1
      IQ(5,6)=-1
      IQ(6,2)=-1
      IQ(6,6)=1
      IQ(7,3)=1
      IQ(7,4)=1
      IQ(7,5)=1
      IQ(7,6)=-1
      IQ(8,6)=-1
      IQ(9,3)=-1
```

```
      IQ(9,7)=1
      IQ(9,8)=1
      IQ(10,4)=-1
      IQ(10,7)=-1
      IQ(11,5)=-1
      IQ(11,8)=-1
      IQ(12,7)=1
      IQ(12,8)=1
      IQ(13,7)=-1
      IQ(14,8)=-1
      IQ(15,7)=1
      IQ(15,8)=1
C.....................................................................
C    INITIALIZE VECTORS
C.....................................................................
      DO 140 I=1,23
      BCHEMF(I)=0.D0
      BCHCUR(I)=0.D0
      BCHPOW(I)=0.D0
  140 CONTINUE
      CIA=-(XSV(3)+XSV(4))
      CIB=XSV(3)
      CIC=XSV(4)
      VA=0.
      VB=0.
      VC=0.
      U(4)=E4
C.....................................................................
C    TURN ALL SWITCHES OFF
C.....................................................................
      RHI=ROFF
      RI=RHI
```

```
      R2=RHI
      R4=RHI
      R5=RHI
      R6=RHI
      R11=RHI
      R12=RHI
      R13=RHI
      R14=RHI
      DO 200 I=1,9
  200 D(I)=.FALSE.
      DO 300 I=1,8
  300 Q(I)=.FALSE.
C.....................................................................
C     CALCULATE THE INITIAL A,B,PHI AND THETA
C.....................................................................
      CALL NETWRK
      CALL STRAN(TSNET,PHI,THETA,A,B,4,4,4,4,1,NTERM3,ETNET,
     1 DIFMAX,ITER,IFLAG)
      NTRMS3=ITER
      DIFMX3=DIFMAX
C.....................................................................
C     CALCULATE THE CUBIC SPLINE PARAMETERS
C.....................................................................
      CALL SPLPRM
      RETURN
      END
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      SUBROUTINE NETWRK
C-----------------------------------------------------------------------
C
C     SUBROUTINE NETWRK CALCULATES THE STATE SPACE EQUATIONS
C     OF THE MACHINE-POWER CONDITIONER NETWORK
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  A11,A12,A13,A21,A22,A23,A31,A32,A33,DETA,
     1                  B11,B12,B13,B21,B22,B23,B31,B32,B33,
     2                  C11,C12,C13,C14,C21,C22,C23,C24,C31,C32,C33,C34,
     3                  G11,G12,G13,G14,G21,G22,G23,G24,G31,G32,G33,G34,
     4                  G41,G42,G43,G44,DETF,F33,F34,F43,F44,SUM
      DOUBLE PRECISION A(4,4),B(4,4),PHI(4,4),THETA(4,4)
      DOUBLE PRECISION XSV(4),U(4),XSVDOT(4)
      DOUBLE PRECISION TEMP(53)
      REAL             BRANCH(23),L1,L2,L3,L4
      COMMON/BLK9  /   E1,E2,E3,E4,C1,R1,R2,R3,R4,R5,R6,R7,R8,R9,L1,R10,
     1                 R11,R12,R13,R14,L2,L3,L4
      COMMON/BLK11 /    A11,A12,A13,A21,A22,A23,A31,A32,A33,DETA,
     1                  B11,B12,B13,B21,B22,B23,B31,B32,B33,
     2                  C11,C12,C13,C14,C21,C22,C23,C24,C31,C32,C33,C34,
     3                  G11,G12,G13,G14,G21,G22,G23,G24,G31,G32,G33,G34,
     4                  G41,G42,G43,G44,DETF,F33,F34,F43,F44,SUM
      COMMON/BLK14 /   A,B,PHI,THETA
      COMMON/BLK15 /   XSV,U,XSVDOT
      EQUIVALENCE      (BRANCH(1),E1)
      EQUIVALENCE (A11,TEMP(1))
C.......................................................................
C     GENERATE MATRIX COEFFICIENTS
C.......................................................................
      A11=R2+R4+R12
      A12=R2
```

```
      A13=R2
      A21=R2
      A22=R2+R5+R13
      A23=R2
      A31=R2
      A32=R2
      A33=R2+R6+R14
      DETA=A11*(A22*A33-A32*A23)-A12*(A21*A33-A31*A23)
     1 +A13*(A21*A32-A31*A22)
      B11=(A22*A33-A32*A23)/DETA
      B12=-(A12*A33-A32*A13)/DETA
      B13=(A12*A23-A22*A13)/DETA
      B21=-(A21*A33-A31*A23)/DETA
      B22=(A11*A33-A31*A13)/DETA
      B23=-(A11*A23-A21*A13)/DETA
      B31=(A21*A32-A31*A22)/DETA
      B32=-(A11*A32-A31*A12)/DETA
      B33=(A11*A22-A21*A12)/DETA
      C11=B11+B12+B13
      C12=R2*(B11+B12+B13)
      C13=B11*R4-B12*R5
      C14=B11*R4-B13*R6
      C21=B21+B22+B23
      C22=R2*(B21+B22+B23)
      C23=B21*R4-B22*R5
      C24=B21*R4-B23*R6
      C31=B31+B32+B33
      C32=R2*(B31+B32+B33)
      C33=B31*R4-B32*R5
      C34=B31*R4-B33*R6
      G11=-1./R10-1./(R1+R11)-(C11+C21+C31)
      G12=1.-R1/(R1+R11)-(C12+C22+C32)
```

```
      G13=-C13-C23-C33
      G14=-C14-C24-C34
      G21=-1.+R1/(R1+R11)+R2*(C11+C21+C31)
      G22=R1*R1/(R1+R11)-(R1+R2+R3)+R2*(C12+C22+C32)
      G23=R2*(C13+C23+C33)
      G24=R2*(C14+C24+C34)
      G31=C11*R4-C21*R5
      G32=C12*R4-C22*R5
      G33=C13*R4-C23*R5-(R4+R5+R7+R8)
      G34=-(R4+R7)+C14*R4-C24*R5
      G41=C11*R4-C31*R6
      G42=C12*R4-C32*R6
      G43=-(R4+R7)+C13*R4-C33*R6
      G44=C14*R4-C34*R6-(R4+R6+R7+R9)
      DETF=L1*L4+L3*L1+L3*L4
      F33=(L1+L4)/DETF
      F34=-L1/DETF
      F43=-L1/DETF
      F44=(L1+L3)/DETF
C.................................................................
C     FORM THE A MATRIX
C.................................................................
      A(1,1)=G11/C1
      A(1,2)=G12/C1
      A(1,3)=G13/C1
      A(1,4)=G14/C1
      A(2,1)=G21/L2
      A(2,2)=G22/L2
      A(2,3)=G23/L2
      A(2,4)=G24/L2
      A(3,1)=G31*F33+G41*F34
      A(3,2)=G32*F33+G42*F34
```

```
      A(3,3)=G33*F33+G43*F34
      A(3,4)=G34*F33+G44*F34
      A(4,1)=G31*F43+G41*F44
      A(4,2)=G32*F43+G42*F44
      A(4,3)=G33*F43+G43*F44
      A(4,4)=G34*F43+G44*F44
C.......................................................................
C     FORM THE B MATRIX
C.......................................................................
      B(1,1)=0.D0
      B(1,2)=0.D0
      B(1,3)=0.D0
      B(1,4)=1./(CI*R10)
      B(2,1)=0.D0
      B(2,2)=0.D0
      B(2,3)=0.D0
      B(2,4)=0.D0
      B(3,1)=F33+F34
      B(3,2)=-F33
      B(3,3)=-F34
      B(3,4)=0.D0
      B(4,1)=F43+F44
      B(4,2)=-F43
      B(4,3)=-F44
      B(4,4)=0.D0
      RETURN
      END
```

```
      SUBROUTINE NTJUMP
C-----------------------------------------------------------------------
C
C     SUBROUTINE NTJUMP ITERATES ON THE NETWORK R, L AND C PARAMETERS AT
C     THE INTEGRATION SAMPLE POINTS K.  IN OTHER WORDS, THE NETWORK
C     IS 'JUMPED' FROM TIME=K-  TO  TIME=K+
C
C-----------------------------------------------------------------------
      DIMENSION        ROLD(9),NLBCH(9)
      DIMENSION RDOON(9),RTRON(9)
      DOUBLE PRECISION A(4,4),B(4,4),PHI(4,4),THETA(4,4)
      DOUBLE PRECISION XSV(4),U(4),XSVDOT(4)
      LOGICAL          AA,BB,CC,FF,AN,BN,CN,FN,QAP1,QAN1,QBP1,QBN1,QCP1,
     1                 QCN1,QMON,QBON,QIOFF,Q(8),NETCHG,NDUMP,D(9)
     2                 ,QBC,QMC
      LOGICAL          DDD(10,9),DOLD(9),QOLD(9),L2CHG
      REAL             BRANCH(23),L1,L2,L3,L4
      COMMON/BLK1  /   CIA,CIB,CIC,VA,VB,VC,VAB,VBC,VCA,PA,PB,PC,PCORE,
     1                 PML,PTRM,PSO,PEM,PGM,PNM,TEM,TM
      COMMON/BLK3  /   RQ1,RQ2,RQ3,RQ4,RQ5,RQ6,RQM,RQB,ROFF
      COMMON/BLK4  /   RD1,RD2,RD3,RD4,RD5,RD6,RDB,RDM,RDR
      COMMON/BLK8  /   AA,BB,CC,FF,AN,BN,CN,FN,QAP1,QAN1,QBP1,QBN1,QCP1,
     1               QCN1,QMON,QBON,QIOFF,Q,NETCHG,NDUMP,D,QBC,QMC
      COMMON/BLK9  /   E1,E2,E3,E4,C1,R1,R2,R3,R4,R5,R6,R7,R8,R9,L1,R10,
     1                 R11,R12,R13,R14,L2,L3,L4
      COMMON/BLK14 /   A,B,PHI,THETA
      COMMON/BLK15 /   XSV,U,XSVDOT
      COMMON/BLK27 /   TSNET,ETNET,NTERM3,DIFMAX,ITER,TIME
      COMMON/BLK32 /   NTRMS1,NTRMS2,NTRMS3,DIFMX1,DIFMX2,DIFMX3
      COMMON/BLK33 /   SAMPLE,NNETSW
      EQUIVALENCE      (BRANCH(1),E1)
      EQUIVALENCE      (RTRON(1),RQ1),(RDOON(1),RD1)
```

```
      DATA NLBCH/9,10,11,18,19,20,17,6,7/,NNLR,NTRANS/9,8/
C......................................................................
C     SAVE THE OLD NETWORK STATUS AT TIME=K-
C......................................................................
      DO 100 I=1,9
      DOLD(I)=D(I)
      IF(I.EQ.9) GO TO 100
      QOLD(I)=Q(I)
  100 CONTINUE
C......................................................................
C     UPDATE THE MACHINE BACK EMFS
C......................................................................
      CALL EPHASE
C......................................................................
C     DETERMINE IF THE NETWORK R,L AND C PARAMETERS HAVE CHANGED
C     FROM TIME=K-  TO  TIME=K+
C......................................................................
      L2CHG=.FALSE.
      IF(SAMPLE.EQ.0.) GO TO 200
      XSP=XSV(2)
      XCH=ABS(XSP)
      CALL SPLINE(XCH,XL2)
      DIFL2=ABS((L2-XL2)/L2)
      L2CHG=DIFL2.GE.SAMPLE
      IF(.NOT.L2CHG) GO TO 200
      L2=XL2
      CALL NETWRK
  200 CONTINUE
      CALL BCHCVP
      CALL SWITCH
      NDUMP=NETCHG.CR.L2CHG
      IF(NETCHG.AND.(SAMPLE.NE.0.)) L2=XL2
```

```
C     ---------------------------------------------------------------
C     RETURN IF THE NETWORK HAS NOT CHANGED
C
      IF(.NOT.NDUMP) RETURN
C     ---------------------------------------------------------------
C     THE NETWORK HAS CHANGED.  INCREMENT NNETSW
C
      NNETSW=NNETSW+1
C................................................................
C     ITERATE ON THE NONLINEAR BRANCH PARAMETERS
C................................................................
      DO 400 I=1,10
      CALL NETWRK
      CALL BCHCVP
      CALL SWITCH
      DO 350 K=1,9
  350 DDD(I,K)=D(K)
      IF(.NOT.NETCHG) GO TO 500
  400 CONTINUE
C................................................................
C     NETWORK ITERATIONS HAVE NOT SETTLED.
C     ADJUST THE NETWORK CONFIGURATION
C................................................................
      DO 450 I=1,9
      ID=NLBCH(I)
      IF(.NOT.(.NOT.DDD(9,I).AND.DDD(10,I).OR.DDD(9,I).AND.
     1 (.NOT.DDD(10,I)))) GO TO 450
      IF(I.EQ.9) GO TO 460
      D(I)=DOLD(I).AND.(.NOT.Q(I))
      GO TO 480
  460 CONTINUE
      D(I)=DOLD(I)
```

```
  480 CONTINUE
      BRANCH(ID)=ROFF
      IF(D(I)) BRANCH(ID)=RODON(I)
  450 CONTINUE
  500 CONTINUE
C.....................................................................
C     FORM PHI AND THETA
C.....................................................................
      CALL STRAN(TSNET,PHI,THETA,A,B,4,4,4,4,1,NTERM3,ETNET,DIFMAX,
     1 ITER,IFLAG)
      NTRMS3=ITER
      DIFMX3=DIFMAX
      RETURN
      END
```

```
      SUBROUTINE SQBC(X,XOB,ISB,ITOL,QBC,IRESET)
C-----------------------------------------------------------------------
C
C     SUBROUTINE SQBC GENERATES THE COMMAND QBC TO TURN ON QB
C
C-----------------------------------------------------------------------
      REAL ITOL
      LOGICAL QBC,LSL,LSH,LXH,LXL,LXGTXO,LXLTXO,LXEQXO
C.......................................................................
C     RESET THE HYSTERESIS IF IRESET=1
C.......................................................................
      IF(IRESET.NE.1) GO TO 10
      ISB=2
      XOB=0.
      QBC=.TRUE.
      RETURN
   10 CONTINUE
C.......................................................................
C   SET COMPARATORS
C.......................................................................
      LSL=X.EQ.(-ITOL)
      LSH=X.EQ.ITOL
      LXH=X.GT.ITOL
      LXL=X.LT.(-ITOL)
      LXGTXO=X.GT.XOB
      LXLTXO=X.LT.XOB
      LXEQXO=X.EQ.XOB
      GO TO (1000,2000),ISB
C.......................................................................
C     XOB IN HYSTERESIS SEGMENT ISB=1
C.......................................................................
 1000 CONTINUE
```

```
      IF(.NOT.(LXEQXO.OR.LXGTXO)) GO TO 1100
      ISB=1
      GO TO 3000
 1100 CONTINUE
      IF(LSL.OR.LXL) GO TO 1200
      ISB=1
      XOB=X
      GO TO 3000
 1200 CONTINUE
      ISB=2
      XOB=-ITOL
      GO TO 3000
C.....................................................................
C     XOB IN HYSTERESIS SEGMENT ISB=2
C.....................................................................
 2000 CONTINUE
      IF(.NOT.(LXEQXO.OR.LXLTXO)) GO TO 2100
      ISB=2
      GO TO 3000
 2100 CONTINUE
      IF(LXH.OR.LSH) GO TO 2200
      ISB=2
      XOB=X
      GO TO 3000
 2200 CONTINUE
      ISB=1
      XOB=ITOL
 3000 CONTINUE
C.....................................................................
C     SET QBC
C.....................................................................
      IF(ISB.EQ.1) QBC=.FALSE.
```

```
IF(ISB.EQ.2) QBC=.TRUE.
RETURN
END
```

```
      SUBROUTINE SQMC(X,XOM,ISM,ITOL,QMC,IRESET)
C-----------------------------------------------------------------------
C
C     SUBROUTINE SQMC GENERATES THE COMMAND QMC TO TURN ON QM
C
C-----------------------------------------------------------------------
      LOGICAL QMC,LSL,LSH,LXH,LXL,LXGTXO,LXLTXO,LXEQXO
      REAL ITOL
C.......................................................................
C     RESET THE HYSTERESIS IF IRESET=1
C.......................................................................
      IF(IRESET.NE.1) GO TO 10
      ISM=2
      XOM=0.
      QMC=.TRUE.
      RETURN
C.......................................................................
C     SET COMPARATORS
C.......................................................................
   10 CONTINUE
      LSL=X.EQ.(-ITOL)
      LSH=X.EQ.ITOL
      LXH=X.GT.ITOL
      LXL=X.LT.(-ITOL)
      LXGTXO=X.GT.XOM
      LXLTXO=X.LT.XOM
      LXEQXO=X.EQ.XOM
      GO TO (1000,2000),ISM
C.......................................................................
C     XOM IN HYSTERESIS SEGMENT ISM=1
C.......................................................................
 1000 CONTINUE
```

```
      IF(.NOT.(LXEQXO.OR.LXLTXO)) GO TO 1100
      ISM=1
      GO TO 3000
 1100 CONTINUE
      IF(LXH.OR.LSH) GO TO 1200
      ISM=1
      XOM=X
      GO TO 3000
 1200 CONTINUE
      ISM=2
      XOM=ITOL
      GO TO 3000
C.....................................................................
C     XOM IN HYSTERESIS SEGMENT ISM=2
C.....................................................................
 2000 CONTINUE
      IF(.NOT.(LXEQXO.OR.LXGTXO)) GO TO 2100
      ISM=2
      GO TO 3000
 2100 CONTINUE
      IF(LSL.OR.LXL) GO TO 2200
      ISM=2
      XOM=X
      GO TO 3000
 2200 CONTINUE
      ISM=1
      XOM=-ITOL
 3000 CONTINUE
C.....................................................................
C     SET QMC
C.....................................................................
      IF(ISM.EQ.1) QMC=.FALSE.
```

```
IF(ISM.EQ.2)  QMC=.TRUE.
RETURN
END
```

```
      SUBROUTINE SWITCH
C-----------------------------------------------------------------------
C
C     SUBROUTINE SWITCH DETERMINES THE STATUS (ON/OFF) OF THE POWER
C     CONDITIONER TRANSISTORS AND DIODES.
C
C-----------------------------------------------------------------------
      DIMENSION RDDON(9),RTRON(9)
      DIMENSION          RGLD(9),NLBCH(9)
      DOUBLE PRECISION BCHCUR(23),BCHEMF(23),BCHPOW(23)
      LOGICAL            AA,BB,CC,FF,AN,BN,CN,FN,QAP1,QAN1,QBP1,QBN1,QCP1,
     1                   QCN1,QMON,QBON,QIOFF,Q(8),NETCHG,NDUMP,D(9)
     2                  ,QBC,QMC
      LOGICAL SPDHI,SPDLO,SPDPOS,SPDNEG,TPOS,TNEG,MTRG1,RGN4,PLUG4,
     1        MTRG3,RGN2,PLUG2
      LOGICAL RGN
      REAL ICMDL,ICMD1,IMC,ITOL,IM,KT
      REAL               BRANCH(23),L1,L2,L3,L4
      COMMON/BLK2  /     BCHCUR,BCHEMF,BCHPOW
      COMMON/BLK3  /     RQ1,RQ2,RQ3,RQ4,RQ5,RQ6,RQM,RQB,ROFF
      COMMON/BLK4  /     RD1,RD2,RD3,RD4,RD5,RD6,RDB,RDM,RDR
      COMMON/BLK6  /     RRANG,RANG,RVEL,RACEL
      COMMON/BLK7/   ICMDL,ICMD1,IMC,IM
      COMMON/BLK8  /     AA,BB,CC,FF,AN,BN,CN,FN,QAP1,QAN1,QBP1,QBN1,QCP1,
     1                  QCN1,QMON,QBON,QIOFF,Q,NETCHG,NDUMP,D,QBC,QMC
      COMMON/BLK9  /     E1,E2,E3,E4,C1,R1,R2,R3,R4,R5,R6,R7,R8,R9,L1,R10,
     1                   R11,R12,R13,R14,L2,L3,L4
      COMMON/BLK21 /     ITOL,SHIFT,KT
      COMMON/BLK31 /     CDMC,CDBC,XOM,XOB,ISM,ISB
      COMMON/MODE  /     SPDHI,SPDLO,SPDPOS,SPDNEG,TPOS,TNEG,MTRG1,RGN4,
     1                   PLUG4,MTRG3,RGN2,PLUG2
      EQUIVALENCE        (BRANCH(1),E1)
```

```
      EQUIVALENCE      (RTRON(1),RQ1),(RDDON(1),RD1)
      DATA NLBCH/9,10,11,18,19,20,17,6,7/,NNLR,NTRANS/9,8/
C.....................................................................
C     SAVE THE PRESENT NONLINEAR RESISTANCES
C.....................................................................
      DO 100 I=1,NNLR
      ID=NLBCH(I)
      ROLD(I)=BRANCH(ID)
  100 CONTINUE
C.....................................................................
C     DETERMINE THE STATUS OF THE TRANSISTORS
C.....................................................................
      RGN=RGN2.OR.RGN4
      CDMC=ABS(IMC)-IM
      CDBC=-ABS(ICMDL)-IM
      CALL SQMC(CDMC,XOM,ISM,ITOL,QMC,0)
      CALL SQBC(CDBC,XOB,ISB,ITOL,QBC,0)
      QMON=QMC.AND.(.NOT.RGN)
      QBON=QBC.AND.RGN
      AA=SIN((SHIFT+RANG)*4.).GE.0.
      BB=SIN((SHIFT-.5235988+RANG)*4.).GE.0.
      CC=SIN((SHIFT-1.047198+RANG)*4.).GE.0.
      AN=.NOT.AA
      BN=.NOT.BB
      CN=.NOT.CC
      FN=TNEG
      FF=TPOS
      QIOFF=RGN
      QAP1=(AA.AND.BN.AND.FF).OR.(AN.AND.CC.AND.FN)
      QAN1=(AN.AND.BB.AND.FF).OR.(AA.AND.CN.AND.FN)
      QBP1=(BB.AND.CN.AND.FF).OR.(AA.AND.BN.AND.FN)
      QBN1=(BN.AND.CC.AND.FF).OR.(AN.AND.BB.AND.FN)
```

```
      QCP1=(.NOT.QAP1).AND.(.NOT.QBP1)
      QCN1=(.NOT.QAN1).AND.(.NOT.QBN1)
      Q(1)=QAP1.AND.(.NOT.QIOFF)
      Q(4)=QAN1.AND.(.NOT.QIOFF)
      Q(2)=QBP1.AND.(.NOT.QIOFF)
      Q(5)=QBN1.AND.(.NOT.QIOFF)
      Q(3)=QCP1.AND.(.NOT.QIOFF)
      Q(6)=QCN1.AND.(.NOT.QIOFF)
      Q(7)=QMON
      Q(8)=QBON
C.....................................................................
C     UPDATE THE NONLINEAR RESISTANCES
C.....................................................................
      DO 200 I=1,8
      D(I)=.FALSE.
      ID=NLBCH(I)
      VDIODE=-BCHEMF(ID)
      BRANCH(ID)=ROFF
      IF(Q(I)) BRANCH(ID)=RTRON(I)
      IF(Q(I)) GO TO 200
      IF(VDIODE.GT.0.) BRANCH(ID)=RDDON(I)
      D(I)=VDIODE.GT.0.
  200 CONTINUE
C.....................................................................
C     UPDATE RDR
C.....................................................................
      BRANCH(7)=ROFF
      VDIODE=BCHEMF(7)
      IF(VDIODE.GT.0) BRANCH(7)=RDDON(9)
      D(9)=VDIODE.GT.0.
C.....................................................................
C     DETERMINE IF ANY OF THE NONLINEAR RESISTANCES HAVE CHANGED
```

```
C.......................................................................
      NETCHG=.FALSE.
      DO 300 I=1,NNLR
      ID=NLBCH(I)
      IF(BRANCH(ID).NE.ROLD(I)) NETCHG=.TRUE.
  300 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BCHCVP
C-----------------------------------------------------------------------
C
C     SUBROUTINE BCHCVP CALCULATES THE BRANCH VOLTAGES, CURRENTS AND
C     POWERS.
C
C-----------------------------------------------------------------------
      DIMENSION        IQ(15,8),NIO1(10)
      DOUBLE PRECISION BCHCUR(23),BCHEMF(23),BCHPOW(23)
      DOUBLE PRECISION  A11,A12,A13,A21,A22,A23,A31,A32,A33,DETA,
     1                  B11,B12,B13,B21,B22,B23,B31,B32,B33,
     2                  C11,C12,C13,C14,C21,C22,C23,C24,C31,C32,C33,C34,
     3                  G11,G12,G13,G14,G21,G22,G23,G24,G31,G32,G33,G34,
     4                  G41,G42,G43,G44,DETF,F33,F34,F43,F44,SUM
      DOUBLE PRECISION A(4,4),B(4,4),PHI(4,4),THETA(4,4)
      DOUBLE PRECISION XSV(4),U(4),XSVDOT(4)
      REAL             BRANCH(23),L1,L2,L3,L4
      REAL ICMDL,ICMD1,IMC,IM
      COMMON/BLK1  /  CIA,CIB,CIC,VA,VB,VC,VAB,VBC,VCA,PA,PB,PC,PCORE,
     1                PML,PTRM,PSO,PEM,PGM,PNM,TEM,TM
      COMMON/BLK2  /  BCHCUR,BCHEMF,BCHPOW
      COMMON/BLK7/  ICMDL,ICMD1,IMC,IM
      COMMON/BLK9  /  E1,E2,E3,E4,C1,R1,R2,R3,R4,R5,R6,R7,R8,R9,L1,R10,
     1                R11,R12,R13,R14,L2,L3,L4
      COMMON/BLK10 /  NBCHS,NTWIGS,NLINKS,NETWIG,NCTWIG,NRTWIG,NLTWIG,
     1                NCLINK,NRLINK,NLLINK,IQ
      COMMON/BLK11 /    A11,A12,A13,A21,A22,A23,A31,A32,A33,DETA,
     1                  B11,B12,B13,B21,B22,B23,B31,B32,B33,
     2                  C11,C12,C13,C14,C21,C22,C23,C24,C31,C32,C33,C34,
     3                  G11,G12,G13,G14,G21,G22,G23,G24,G31,G32,G33,G34,
     4                  G41,G42,G43,G44,DETF,F33,F34,F43,F44,SUM
      COMMON/BLK14 /  A,B,PHI,THETA
```

```
      COMMON/BLK15 /  XSV,U,XSVDOT
      EQUIVALENCE     (BRANCH(1),E1)
      EQUIVALENCE     (NIO1(1),NBCHS)
C..................................................................
C      CALCULATE XSVDOT USING THE LATEST S.V.'S XSV
C..................................................................
      DO 300 I=1,4
      SUM=0.0D0
      DO 200 J=1,4
      SUM=SUM+A(I,J)*XSV(J)+B(I,J)*U(J)
  200 CONTINUE
      XSVDOT(I)=SUM
  300 CONTINUE
C    ------------------------------------------------------------
C    LOAD S.V. INDUCTOR CURRENTS INTO BCHCUR
C
      BCHCUR(21)=XSV(2)
      BCHCUR(22)=XSV(3)
      BCHCUR(23)=XSV(4)
C    ------------------------------------------------------------
C    LOAD EMFS INTO BCHEMF
C
      BCHEMF(1)=U(1)
      BCHEMF(2)=U(2)
      BCHEMF(3)=U(3)
      BCHEMF(4)=U(4)
C..................................................................
C    FIND THE VOLTAGE ACROSS THE TWIG INDUCTOR
C..................................................................
      BCHEMF(15)=-BRANCH(15)*(XSVDOT(3)+XSVDOT(4))
C..................................................................
C    FIND THE VOLTAGE ACROSS THE LINK INDUCTORS
```

```
C.....................................................................
      BCHEMF(21)=BRANCH(21)*XSVDOT(2)
      BCHEMF(22)=BRANCH(22)*XSVDOT(3)
      BCHEMF(23)=BRANCH(23)*XSVDOT(4)
C.....................................................................
C    FIND THE RESISTOR LINK CURRENTS
C.....................................................................
      BCHCUR(16)=(XSV(1)-U(4))/R10
      BCHCUR(17)=(XSV(1)+XSV(2)*R1)/(R1+R11)
      BCHCUR(18)=C11*XSV(1)+C12*XSV(2)+C13*XSV(3)+C14*XSV(4)
      BCHCUR(19)=C21*XSV(1)+C22*XSV(2)+C23*XSV(3)+C24*XSV(4)
      BCHCUR(20)=C31*XSV(1)+C32*XSV(2)+C33*XSV(3)+C34*XSV(4)
C.....................................................................
C    FIND THE RESISTOR LINK VOLTAGES
C.....................................................................
      DO 350 I=16,20
      BCHEMF(I)=BCHCUR(I)*BRANCH(I)
  350 CONTINUE
C.....................................................................
C    FIND THE TWIG CURRENTS
C.....................................................................
      DO 500 M=1,NTWIGS
      BCHCUR(M)=0.D0
      DO 400 N=1,NLINKS
      IGO=IQ(M,N)+2
      GO TO (410,400,420),IGO
  410 CONTINUE
      BCHCUR(M)=BCHCUR(M)+BCHCUR(N+NTWIGS)
      GO TO 400
  420 CONTINUE
      BCHCUR(M)=BCHCUR(M)-BCHCUR(N+NTWIGS)
  400 CONTINUE
```

```
  500 CONTINUE
C     ---------------------------------------------------------------
C     LOAD THE TWIG CAPACITOR VOLTAGE INTO BCHEMF
C
      BCHEMF(5)=XSV(1)
C.....................................................................
C     DETERMINE THE TWIG RESISTOR VOLTAGES
C.....................................................................
      DO 600 M=1,NRTWIG
      MM=M+NETWIG+NCTWIG
      BCHEMF(MM)=BCHCUR(MM)*BRANCH(MM)
  600 CONTINUE
C.....................................................................
C     DETERMINE THE BRANCH POWERS
C.....................................................................
      DO 700 I=1,23
      BCHPOW(I)=BCHEMF(I)*BCHCUR(I)
  700 CONTINUE
C.....................................................................
C     DETERMINE THE PHASE VOLTAGES
C.....................................................................
      VA=BCHEMF(1)+BCHEMF(12)+BCHEMF(15)
      VB=BCHEMF(2)+BCHEMF(13)+BCHEMF(22)
      VC=BCHEMF(3)+BCHEMF(14)+BCHEMF(23)
C.....................................................................
C     DETERMINE MACHINE LINE TO LINE VOLTAGES
C.....................................................................
      VAB=VA-VB
      VBC=VB-VC
      VCA=VC-VA
C.....................................................................
C     DETERMINE INSTANTANEOUS PHASE POWERS
```

```
C.....................................................................
      PA=CIA*VA
      PB=CIB*VB
      PC=CIC*VC
C.....................................................................
C     DETERMINE THE D.C. LINE CURRENT
C.....................................................................
      IM    =BCHCUR(9)+BCHCUR(10)+BCHCUR(11)
      RETURN
      END
```

```
      SUBROUTINE EPHASE
C----------------------------------------------------------------------
C
C     SUBROUTINE EPHASE DETERMINES THE MACHINE PHASE BACK EMFS
C
C----------------------------------------------------------------------
      DIMENSION        ANGLE(3)
      DOUBLE PRECISION XSV(4),U(4),XSVDOT(4)
      REAL             BRANCH(23),L1,L2,L3,L4
      REAL             LMV(3),LGV(3)
      REAL LG,LM,LT,BR,HC,BRM,HCM,UO
      COMMON/BLK1  /   CIA,CIB,CIC,VA,VB,VC,VAB,VBC,VCA,PA,PB,PC,PCORE,
     1                 PML,PTKM,PSO,PEM,PGM,PNM,TEM,TM
      COMMON/BLK6  /   RKANG,RANG,RVEL,RACEL
      COMMON/BLK9  /   E1,E2,E3,E4,C1,R1,R2,R3,R4,R5,R6,R7,R8,R9,L1,R10,
     1                 R11,R12,R13,R14,L2,L3,L4
      COMMON/BLK12 /   BOC(3),BGAP(3),FS(3),FM(3),FG(3),HM(3),HG(3),LMV,
     1                 LGV
      COMMON/BLK15 /   XSV,U,XSVDOT
      EQUIVALENCE      (BRANCH(1),E1)
      DATA ANGLE/.3926991,.9162978,1.43989/
      DATA        NPOLES,NSLOTS,NUMANG,RCOND,CONDL/  8,24,3,.0236,.1335/
      DATA NCPSLT,LT,BRM,HCM,UO/11,.0097155,.86,-636220.,12.56637E-7/
C......................................................................
C     CALCULATE THE VELOCITY OF THE STATOR CONDUCTORS W.R.T.
C     THE FIELD
C......................................................................
      VEL=-RVEL*RCOND
C......................................................................
C     CALCULATE BGAP AT THE ABC PHASE AXIS
C......................................................................
      CALL SANGLE
```

```
      CALL SBGAP(BGAP,ANGLE,RRANG,NUMANG)
C.......................................................................
C     DETERMINE THE STATCR MMFS
C.......................................................................
      CALL STMMF(ANGLE,CIA,CIB,CIC,NCPSLT,FS)
C.......................................................................
C     DEMAGNETIZE THE MAGNETS
C.......................................................................
      DO 100 K=1,NUMANG
      BR=BRM
      HC=HCM
      IF(BGAP(K).LT.0.) BR=-BRM
      IF(BGAP(K).LT.0.) HC=-HCM
      LG=LT/(1.+BR*BGAP(K)/(UO*HC*(BGAP(K)-BR)))
      LM=LT-LG
      LMV(K)=LM
      LGV(K)=LG
      BOC(K)=BGAP(K)
      BGAP(K)=BR*(HC*LM-FS(K))/(HC*LM-BR*LG/UO)
      HM(K)=(BR-BGAP(K))*HC/BR
      HG(K)=BGAP(K)/UO
      FM(K)=HM(K)*LM
      FG(K)=HG(K)*LG
  100 CONTINUE
C.......................................................................
C     CALCULATE THE PHASE EMFS  USING EMF=-BLV
C.......................................................................
      E1=-BGAP(1)*CONDL*VEL*NPOLES*NCPSLT
      E2=-BGAP(2)*CONDL*VEL*NPOLES*NCPSLT
      E3=-BGAP(3)*CONDL*VEL*NPOLES*NCPSLT
      U(1)=E1
      U(2)=E2
```

```
U(3)=E3
RETURN
END
```

```
      SUBROUTINE TURPOW
C-----------------------------------------------------------------------
C
C     SUBROUTINE TURPOW CALCULATES THE MACHINE TORQUES AND POWERS
C
C-----------------------------------------------------------------------
      REAL   ICMDL,ICMD1,IMC,IM,ITOL,KT
      REAL              BRANCH(23),L1,L2,L3,L4
      COMMON/BLK1  /    CIA,CIB,CIC,VA,VB,VC,VAB,VBC,VCA,PA,PB,PC,PCORE,
     1                  PML,PTRM,PSO,PEM,PGM,PNM,TEM,TM
      COMMON/BLK6  /    RRANG,RANG,RVEL,RACEL
      COMMON/BLK7/   ICMDL,ICMD1,IMC,IM
      COMMON/BLK9  /    E1,E2,E3,E4,C1,R1,R2,R3,R4,R5,R6,R7,R8,R9,L1,R10,
     1                  R11,R12,R13,R14,L2,L3,L4
      COMMON/BLK21 /    ITOL,SHIFT,KT
      EQUIVALENCE       (BRANCH(1),E1)
      EA=E1
      EB=E2
      EC=E3
      RA=R7
      RB=R8
      RC=R9
C.......................................................................
C     ELECTROMAGNETIC POWER
C.......................................................................
      PEM=CIA*EA+CIB*EB+CIC*EC
C.......................................................................
C     TERMINAL POWER
C.......................................................................
      PTRM=VA*CIA+VB*CIB+VC*CIC
C.......................................................................
C     OHMIC LOSSES IN THE STATOR
```

```
C.....................................................................
      PSO=(CIA**2)*RA+(CIB**2)*RB+(CIC**2)*RC
C.....................................................................
C     ELECTROMAGNETIC TORQUE
C.....................................................................
      IF(RVEL.EQ.0.) GO TO 1000
      TEM=PEM/RVEL
      TM=TEM
      RETURN
C.....................................................................
C     MACHINE AT REST--CALCULATE TEM AND TM
C.....................................................................
 1000 CONTINUE
      TEM=KT*ICMD1
      TM=TEM
      RETURN
      END
```

```
      SUBROUTINE SWLOSS
C-----------------------------------------------------------------------
C
C     THIS SUBROUTINE CALCULATES THE INSTANTANEOUS POWER LOSSES IN
C     THE TRANSISTOR AND DIODE SWITCHES DURING CONDUCTION.
C
C     LOSSES WHILE THESE SWITCHES ARE OFF ARE NEGLECTED.
C
C-----------------------------------------------------------------------
      DIMENSION NLBCH(9)
      DOUBLE PRECISION BCHCUR(23),BCHEMF(23),BCHPOW(23)
      LOGICAL          AA,BB,CC,FF,AN,BN,CN,FN,QAP1,QAN1,QBP1,QBN1,QCP1,
     1                 QCN1,QMON,QBON,QIOFF,Q(8),NETCHG,NDUMP,D(9)
     2                ,QBC,QMC
      COMMON/BLK2  /   BCHCUR,BCHEMF,BCHPOW
      COMMON/BLK8  /   AA,BB,CC,FF,AN,BN,CN,FN,QAP1,QAN1,QBP1,QBN1,QCP1,
     1               QCN1,QMON,QBON,QIOFF,Q,NETCHG,NDUMP,D,QBC,QMC
      COMMON/BLK13 /   QLOSS(8),DLOSS(9),PELOSS
      DATA NLBCH/9,10,11,18,19,20,17,6,7/
C.......................................................................
C     DETERMINE THE TRANSISTOR LOSSES
C.......................................................................
      DO 100 I=1,8
      ID=NLBCH(I)
      QLOSS(I)=0.
      IF(Q(I)) QLOSS(I)=BCHPOW(ID)
  100 CONTINUE
C.......................................................................
C     DETERMINE THE DIODE LOSSES
C.......................................................................
      DO 200 I=1,9
      ID=NLBCH(I)
```

```
      DLOSS(I)=0.
      IF(D(I)) DLOSS(I)=BCHPOW(ID)
  200 CCNTINUE
C.......................................................................
C    DETERMINE THE TOTAL LOSS
C.......................................................................
      PELOSS=0.
      DO 300 I=1,8
      PELOSS=PELOSS+DLOSS(I)+QLOSS(I)
  300 CONTINUE
      PELOSS=PELOSS+DLOSS(9)
      RETURN
      END
```

```
      SUBROUTINE STMMF(ANGLE,CIA,CIB,CIC,NCPSLT,SMMF)
C-----------------------------------------------------------------------
C
C     THIS ROUTINE CALCULATES THE STATOR MMFS AT THE STATOR CONDUCTORS
C
C-----------------------------------------------------------------------
      LOGICAL AMF,BMF,CMF
      DIMENSION ANGLE(3),SMMF(3)
C.......................................................................
C     CALCULATE THE SLOT AMPERE TURNS
C.......................................................................
      AT1=NCPSLT*CIA
      AT2=NCPSLT*CIB
      AT3=NCPSLT*CIC
      DO 1000 K=1,3
      THETA=ANGLE(K)
C.......................................................................
C     DETERMINE THE MMF
C.......................................................................
      AMF=-COS(4.*THETA).GE.0.
      BMF=-COS(4.*(THETA-.5235988)).GE.0.
      CMF=-COS(4.*(THETA-1.047198)).GE.0.
      IAMMF=-1
      IBMMF=-1
      ICMMF=-1
      IF(AMF) IAMMF=1
      IF(BMF) IBMMF=1
      IF(CMF) ICMMF=1
      GO TO (100,200,300),K
  100 CONTINUE
      SMMF(K)=IBMMF*AT2+ICMMF*AT3
      GO TO 400
```

```
 200 CONTINUE
     SMMF(K)=IAMMF*AT1+ICMMF*AT3
     GO TO 400
 300 CONTINUE
     SMMF(K)=IAMMF*AT1+IBMMF*AT2
 400 CONTINUE
1000 CONTINUE
     RETURN
     END
```

```
      SUBROUTINE SANGLE
C-----------------------------------------------------------------------
C
C     SUBROUTINE SANGLE NORMALIZES THE ABSOLUTE ROTOR ANGLE
C     RANG TO WITHIN TWO POLE PITCHES.
C
C-----------------------------------------------------------------------
      COMMON/BLK6  /  RRANG,RANG,RVEL,RACEL
      PI=3.141593
      MOD= RANG*2./PI
      ANGMOD=MOD*PI/2.
      RRANG=RANG-ANGMOD
      IF(RRANG.LT.O.) RRANG=RRANG+PI/2.
      RETURN
      END
```

```
      SUBROUTINE SBGAP(BGAP,ANGLE,RRANG,NUMANG)
C-----------------------------------------------------------------------
C
C     SUBROUTINE SBGAP DETERMINES THE NO LOAD AIR GAP FLUX DENSITIES
C     AT THE PHASE CONDUCTORS.
C
C-----------------------------------------------------------------------
      DIMENSION ANGLE(3),BGAP(3)
      BPEAK=0.536
      DO 1000 I=1,NUMANG
      ANG=(RRANG-ANGLE(I))*4.
      BGAP(I)=BPEAK*(1.111111*(COS(ANG))
     1         -.0819017*COS(3.*ANG)
     2         -.0807241*COS(5.*ANG)
     3         +.0405518*COS(7.*ANG))
 1000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE STRAN(TAU,PHI,THETA,A,B,NA,NB,MA,MB,MODE,NTERMS,
     1 TOL,DIFMAX,ITER,IFLAG)
C-----------------------------------------------------------------------
C
C    SUBROUTINE STRAN CALCULATES THE PHI AND THETA STATE
C    TRANSITION MATRICES USING THE EXPONENTIAL MATRIX METHOD.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION C1,TS
      DOUBLE PRECISION PHI(NA,NA),THETA(NA,NB),A(NA,NA),B(NA,NB),
     1 WORK1(10,10),WORK2(10,10),DUMMY(1,1),FAC1,CON1,FAC2,CON2
C.......................................................................
C    INITIALIZE
C.......................................................................
      C1=1.D0
      TS=TAU
      DO 4 I=1,MA
      DO 2 J=1,MA
      WORK1(I,J)=0.0D0
      WORK2(I,J)=0.0D0
      PHI(I,J)=0.0D0
    2 CONTINUE
      WORK1(I,I)=1.D0
      WORK2(I,I)=TAU
      PHI(I,I)=1.D0
    4 CONTINUE
      DO 6 I=1,MA
      DO 6 J=1,MB
    6 THETA(I,J)=0.D0
      DIFMAX=1.E8
      NDO=50
      IF(NTERMS.GT.0) NDO=NTERMS
```

```
      FAC1=1.D0
      IFLAG=0
      DO 1000 I=1,NDO
C..........................................................................
C    FORM THE MATRIX PRODUCT WORK=WORK*A*TAU
C..........................................................................
      CALL MXMUL(DUMMY,WORK1,A,TS ,1,1,10,10,NA,NA,MA,MA,MA,MA,1)
C..........................................................................
C    ADD THE NEXT TERM TO PHI AND FIND DIFMAX (IF NTERMS=0)
C..........................................................................
      FAC1=FAC1*I
      CON1=1./FAC1
      IF(NTERMS.EQ.0.AND.I.GE.4) CALL SERROR(PHI,WORK1,CON1,NA,NA,
     1  10,10,MA,MA,DIFMAX)
      CALL MXADD(DUMMY,PHI,WORK1,C1,CON1,1,1,NA,NA,10,10,MA,MA,1)
C..........................................................................
C    ADD THE NEXT TERM TO THETA IF MODE =1
C..........................................................................
      IF(MODE.EQ.2) GO TO 500
      FAC2=FAC1*(I+1)
      CON2=TAU/FAC2
C     ---------------------------------------------------------------------
C    ADD THE NEXT TERM TO THETA
C
      CALL MXADD(DUMMY,WORK2,WORK1,C1,CON2,1,1,10,10,10,10,MA,MA,1)
  500 CONTINUE
C..........................................................................
C    CHECK FOR CONVERGENCE AFTER THE THIRD TERM
C..........................................................................
      ITER=I
      IF(NTERMS.GT.0.OR.I.LT.4) GO TO 1000
      IF(DIFMAX.LE.TOL) GO TO 1100
```

```
 1000 CONTINUE
 1100 CONTINUE
C......................................................................
C     FINISH THETA IF MODE = 1
C......................................................................
      IF(MODE.EQ.1) CALL MXMUL(THETA,WORK2,B,C1,NA,NB,10,10,NA,NB,
     1 MA,MA,MA,MB,2)
C......................................................................
C     DETERMINE WHETHER OR NOT THE SERIES HAS CONVERGED
C......................................................................
      IF(ITER.EQ.NDO.AND.NTERMS.EQ.0) IFLAG=1
      RETURN
      END
```

```
      SUBROUTINE SERROR(AMX,BMX,CCC,IA,JA,IB,JB,IDO,JDO,DIFMAX)
C-----------------------------------------------------------------------
C
C     SUBROUTINE SERROR DETERMINES THE MAXIMUM MISMATCH BETWEEN
C     TWO SUCESSIVE TERMS OF THE STATE TRANSITION EXPONENTIAL
C     MATRIX SERIES.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION AMX(IA,JA),BMX(IB,JB),CCC
      DIFMAX=1.E-30
      DO 100 I=1,IDO
      DO 50 J=1,JDO
      IF(AMX(I,J).EQ.0.D0) GO TO 50
      CHANGE=DABS(BMX(I,J)*CCC/AMX(I,J))
      IF(CHANGE.GT.DIFMAX) DIFMAX=CHANGE
   50 CONTINUE
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FUTURE(X,V,PHI,THETA,LP,MP,LT,MT,IP,JP,IT,JT,MODE)
C-----------------------------------------------------------------------
C
C     SUBROUTINE FUTURE INTEGRATES THE STATE EQUATIONS FORWARD
C     IN TIME BY ONE TIME STEP USING THE TRANSITION MATRICES
C     CALCULATED IN SUBROUTINE STRAN.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION X(MP),V(MT),PHI(LP,MP),THETA(LT,MT),
     1 TEMP(20),SUM
C.......................................................................
C     MULTIPLY X BY PHI (X=PHI*X)
C.......................................................................
      DO 20 I=1,IP
      SUM=0.0D0
      DO 10 J=1,JP
   10 SUM=SUM+PHI(I,J)*X(J)
   20 TEMP(I)=SUM
      DO 30 I=1,IP
   30 X(I)=TEMP(I)
C.......................................................................
C     MULTIPLY V BY THETA (V=THETA*V) IF MODE=1
C.......................................................................
      IF(MODE.EQ.2) RETURN
      DO 60 I=1,IT
      SUM=0.0D0
      DO 50 J=1,JT
   50 SUM=SUM+THETA(I,J)*V(J)
C.......................................................................
C     ADD THETA*V TO X
C.......................................................................
      X(I)=X(I)+SUM
```

```
60 CONTINUE
   RETURN
   END
```

```
      SUBROUTINE MXADD(RMX,AMX,BMX,ACC,BCC,IR,JR,IA,JA,IB,JB,IDO,JDO,
     1MODE)
C-----------------------------------------------------------------------
C
C     SUBROUTINE MXADD ADDS TWO DOUBLE PRECISICN MATRICES TOGETHER.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION AMX(IA,JA),BMX(IB,JB),RMX(IR,JR),ACC,BCC
C.......................................................................
C     CHECK FOR IMPROPER MATRICES
C.......................................................................
      IF(IA.LT.IDO.OR.JA.LT.JDO)  GO TO 999
      IF(IB.LT.IDO.OR.JB.LT.JDO) GO TO 999
      GO TO (10,100),MODE
C.......................................................................
C     MODE=1  AMX=AMX*ACC+BMX*BCC
C.......................................................................
   10 CONTINUE
      DO 50 I=1,IDO
      DO 50 J=1,JDO
   50 AMX(I,J)=AMX(I,J)*ACC+BMX(I,J)*BCC
      GO TO 300
C.......................................................................
C     MODE=2  RMX=AMX*ACC+BMX*BCC
C.......................................................................
  100 CONTINUE
C     ------------------------------------------------------------------
C     CHECK FOR IMPROPER DIMENSIONS OF RMAT
C
      IF(IR.LT.IDO.OR.JR.LT.JDO) GO TO 999
      DO 200 I=1,IDO
      DO 200 J=1,JDO
```

```
  200 RMX(I,J)=AMX(I,J)*ACC+BMX(I,J)*BCC
  300 RETURN
C.......................................................................
C     ARRANGE ABORT
C.......................................................................
  959 CONTINUE
      WRITE(6,6000)
 6000 FORMAT(1H0,'***ERROR IN ROUTINE MXADD:  MATRICES HAVE WRONG DIMENS
     1IONS***')
      RETURN
      END
```

```
      SUBROUTINE MXMUL(RMX,AMX,BMX,CCC,IR,JR,IA,JA,IB,JB,IDA,JDA,IDB,JDB
     1,MODE)
C-----------------------------------------------------------------------
C
C     SUBROUTINE MXMUL MULTIPLIES TWO DOUBLE PRECISION MATRICES TOGETHER.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION AMX(IA,JA),BMX(IB,JB),RMX(IR,JR),TEMP(20),SUM,
     1 CCC
C.......................................................................
C     CHECK FOR IMPROPER MATRICES
C.......................................................................
      IF(IDA*IDB*JDA*JDB.GT.IA*IB*JA*JB) GO TO 999
      IF(JDB.GT.JDA) GO TO 999
      GO TO (10,210),MODE
C.......................................................................
C     MODE=1   AMX=AMX*BMX*CCC
C.......................................................................
   10 DO 100 I=1,IDA
      DO 20 L=1,JDA
   20 TEMP(L)=AMX(I,L)
      DO 80 J=1,JDB
      SUM=0.0D0
      DO 40 K=1,JDA
   40 SUM=SUM+TEMP(K)*BMX(K,J)
      AMX(I,J)=SUM*CCC
   80 CONTINUE
  100 CONTINUE
      GO TO 600
C.......................................................................
C     MODE=2  RMX=AMX*BMX*CCC
C.......................................................................
```

```
  210 CONTINUE
C     ------------------------------------------------------------------
C     CHECK FOR IMPROPER RMX DIMENSIONS
C
      IF(IR.LT.IDA.OR.JR.LT.JDB) GO TO 999
C     ------------------------------------------------------------------
C     MULTIPLY MATRICES
C
      DO 400 I=1,IDA
      DO 380 J=1,JDB
      SUM=0.0D0
      DO 340 K=1,JDA
  340 SUM=SUM+AMX(I,K)*BMX(K,J)
      RMX(I,J)=SUM*CCC
  380 CONTINUE
  400 CONTINUE
  600 RETURN
C.......................................................................
C     ARRANGE ABORT
C.......................................................................
  999 CONTINUE
      WRITE(6,6000)
 6000 FORMAT(1H0,'***ERROR IN ROUTINE MXMUL:  MATRICES HAVE WRONG DIMENS
     1IONS***')
      RETURN
      END
```

```
      SUBROUTINE SPLPRM
C-----------------------------------------------------------------------
C
C     THIS SUBROUTINE CALCULATES THE PARAMETERS FOR THE CUBIC SPLINE
C     INTERPOLATION PROGRAM.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION XI(20),FXI(20),YI(20),HI(20),BI(20),RHI(20),
     1 YI2(20),TEMP(20)
      COMMON/BLK5  /  XI,FXI,HI,BI,RHI,YI2,NDPTS
      N1=NDPTS-1
      DO 1000 I=1,N1
      HI(I)=XI(I+1)-XI(I)
 1000 CONTINUE
      DO 1010 I=2,N1
      BI(I-1)=0.6D1*((FXI(I+1)-FXI(I))/HI(I)-
     1 (FXI(I)-FXI(I-1))/HI(I-1))
 1010 CONTINUE
      DO 1020 I=2,N1
      RHI(I-1)=0.2D1*(HI(I)+HI(I-1))
 1020 CONTINUF
      YI2(1)=0.0D0
      YI2(NDPTS)=0.0D0
      N2=NDPTS-2
      CALL TRDIAG(HI,RHI,HI,TEMP,BI,N2,IERROR)
      DO 1030 I=2,N1
      YI2(I)=TEMP(I-1)
 1030 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRDIAG(A,B,C,X,F,N,IERROR)
C-----------------------------------------------------------------------
C
C     SUBROUTINE TRDIAG SOLVES A TRI DIAGONAL SYSTEM OF EQUATIONS
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION A(20),B(20),C(20),F(20),U(20),Y(20),X(20),L(20)
      INTEGER N ,IERROR
      U(1)=B(1)
      DO 1 I=2,N
           IF(U(I-1).EQ.0) GO TO 4
           L(I)=A(I-1)/U(I-1)
           U(I)=B(I)-L(I)*C(I-1)
    1      CONTINUE
      Y(1)=F(1)
      DO 2 I=2,N
           Y(I)=F(I)-L(I)*Y(I-1)
    2  CONTINUE
      IF (U(N).EQ.0) GO TO 4
      X(N)=Y(N)/U(N)
      NM1=N-1
      DO 3 I=1,NM1
           X(N-I)=(Y(N-I)-C(N-I)  *X(N+1-I))/U(N-I)
    3      CONTINUE
      IERROR=1
      RETURN
    4 IERROR=2
      RETURN
      END
```

```
      SUBROUTINE SPLINE(X,Y)
C-----------------------------------------------------------------------
C
C     THIS SUBROUTINE USES CUBIC SPLINES TO FIND Y FOR A GIVEN X
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION XI(20),FXI(20),YI(20),HI(20),BI(20),RHI(20),
     1 YI2(20)
      COMMON/BLK5  /  XI,FXI,HI,BI,RHI,YI2,NDPTS
      N1=NDPTS-1
      IF(X.LE.XI(1)) GO TO 9991
      IF(X.GE.XI(NDPTS)) GO TO 9992
      J=1
 1000 IF(X.GE.XI(J).AND.X.LE.XI(J+1)) GO TO 2000
      J=J+1
      GO TO 1000
 2000 CONTINUE
      Y=YI2(J)*(XI(J+1)-X)**3/(6.0D0*HI(J))
     1  +YI2(J+1)*(X-XI(J))**3/(6.0D0*HI(J))
     2  +(FXI(J+1)/HI(J)-(YI2(J+1)*HI(J))/6.)*
     3  (X-XI(J))+(FXI(J)/HI(J)-YI2(J)*HI(J)/6.)
     4  *(XI(J+1)-X)
      RETURN
 9991 Y=FXI(1)
      RETURN
 9992 Y=FXI(NDPTS)
      RETURN
      END
```

```
      SUBROUTINE SVINTP(MTYPE)
C-----------------------------------------------------------------------
C
C     SUBROUTINE SVINTP INTERPOLATES XM AND XV DURING THE INTRA-SAMPLE
C     PERIOD.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION XV(5),UV(5),XVOLD(5),UVOLD(5),XVI(5),UVI(5)
      DOUBLE PRECISION  XM(5),UM(5),XMOLD(5),UMOLD(5),XMI(5),UMI(5)
      COMMON/BLK17 /  XV,UV,XVOLD,UVOLD,XVI,UVI
      COMMON/BLK20 /  XM,UM,XMOLD,UMOLD,XMI,UMI
      COMMON/BLK24 /  NINT1,NINT2,NINT3,I1,I2,I3
      COMMON/BLK25 /  TSMLP,ETMLP,NTERM1,NMACH,N1,N2,N3,NX
      COMMON/BLK26 /  TSVLP,ETVLP,NTERM2
      COMMON/BLK27 /  TSNET,ETNET,NTERM3,DIFMAX,ITER,TIME
      GO TO (1000,2000),MTYPE
 1000 CONTINUE
      RATIO1=I2*TSVLP/TSMLP
      DO 5 I=1,5
      XVI(I)=XV(I)
      XMI(I)=XMOLD(I)+(XM(I)-XMOLD(I))*RATIO1
    5 CONTINUE
      RETURN
 2000 CONTINUE
      RATIO1=((I2-1)*TSVLP+I3*TSNET)/TSMLP
      RATIO2=I3*TSNET/TSVLP
      DO 100 I=1,5
      XVI(I)=XVOLD(I)+(XV(I)-XVOLD(I))*RATIO2
      XMI(I)=XMOLD(I)+(XM(I)-XMOLD(I))*RATIO1
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE OUTPUT(PGMODE,NUMINT,NCALLS)
C-----------------------------------------------------------------------
C
C     SUBROUTINE OUTPUT PRINTS THE COMPUTED DATA WHEN SPECIFIED.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION BCHCUR(23),BCHEMF(23),BCHPOW(23)
      DOUBLE PRECISION XSV(4),U(4),XSVDOT(4)
      DOUBLE PRECISION XV(5),UV(5),XVOLD(5),UVOLD(5),XVI(5),UVI(5)
      DOUBLE PRECISION  XM(5),UM(5),XMOLD(5),UMOLD(5),XMI(5),UMI(5)
      INTEGER PGMODE
      LOGICAL          AA,BB,CC,FF,AN,BN,CN,FN,QAP1,QAN1,QBP1,QBN1,QCP1,
     1                 QCN1,QMON,QBON,QIOFF,Q(8),NETCHG,NDUMP,D(9)
     2                ,QBC,QMC
      LOGICAL SPDHI,SPDLO,SPDPOS,SPDNEG,TPOS,TNEG,MTRG1,RGN4,PLUG4,
     1        MTRG3,RGN2,PLUG2
      REAL             LMV(3),LGV(3)
      REAL ICMDL,ICMD1,IMC,IM
      REAL             BRANCH(23),L1,L2,L3,L4
      REAL JM,JF,KACT,KF,KP,KERR
      REAL             N1,N2,N3
      COMMON/BLK1  /   CIA,CIB,CIC,VA,VB,VC,VAB,VBC,VCA,PA,PB,PC,PCORE,
     1                 PML,PTRM,PSO,PEM,PGM,PNM,TEM,TM
      COMMON/BLK2  /   BCHCUR,BCHEMF,BCHPOW
      COMMON/BLK6  /   RRANG,RANG,RVEL,RACEL
      COMMON/BLK7/   ICMDL,ICMD1,IMC,IM
      COMMON/BLK8  /   AA,BB,CC,FF,AN,BN,CN,FN,QAP1,QAN1,QBP1,QBN1,QCP1,
     1                QCN1,QMON,QBON,QIOFF,Q,NETCHG,NDUMP,D,QBC,QMC
      COMMON/BLK9  /   E1,E2,E3,E4,C1,R1,R2,R3,R4,R5,R6,R7,R8,R9,L1,R10,
     1                 R11,R12,R13,R14,L2,L3,L4
      COMMON/BLK12 /   BOC(3),BGAP(3),FS(3),FM(3),FG(3),HM(3),HG(3),LMV,
     1                 LGV
```

```
      COMMON/BLK13 /  QLOSS(8),DLOSS(9),PELOSS
      COMMON/BLK15 /  XSV,U,XSVDOT
      COMMON/BLK17 /  XV,UV,XVOLD,UVOLD,XVI,UVI
      COMMON/BLK20 /  XM,UM,XMOLD,UMOLD,XMI,UMI
      COMMON/BLK24 /  NINT1,NINT2,NINT3,I1,I2,I3
      COMMON/BLK25 /  TSMLP,ETMLP,NTERM1,NMACH,N1,N2,N3,NX
      COMMON/BLK27 /  TSNET,ETNET,NTERM3,DIFMAX,ITER,TIME
      COMMON/BLK28 /  FANG,PE,VE,TACT
      COMMON/BLK32 /  NTRMS1,NTRMS2,NTRMS3,DIFMX1,DIFMX2,DIFMX3
      COMMON/MLP   /  DC,VMSAT,TAU3,JM,JF,BF,KACT,KF,KP,KERR
      COMMON/MODE  /  SPDHI,SPDLO,SPDPOS,SPDNEG,TPOS,TNEG,MTRG1,RGN4,
     1                PLUG4,MTRG3,RGN2,PLUG2
      EQUIVALENCE     (BRANCH(1),F1)
C..........................................................................
C     PRINT ITERATION DATA
C..........................................................................
      NCALLS=NCALLS+1
      NPPG=2
      IF(PGMODE.EQ.2) NPPG=4
      IF(MOD(NCALLS,NPPG).EQ.1) WRITE(6,6010)
      IF(MOD(NCALLS,NPPG).NE.1) WRITE(6,6011)
      WRITE(6,6020) PGMODE,NUMINT,TIME,I1,I2,I3,RANG,RVEL
      IF(PGMODE.NE.1) WRITE(6,6030) DC,FANG,PE,VE,TACT,NMACH
      WRITE(6,6040) IM,IMC,ICMDI,ICMDL,TM
      WRITE(6,6050)
      IF(PGMODE.EQ.1) GO TO 10
      WRITE(6,6060) (XMI(I),I=1,5),NTRMS1
      WRITE(6,6070) (UM(I),I=1,5),DIFMX1
      WRITE(6,6080) (XVI(I),I=1,5),NTRMS2
      WRITE(6,6090) (UV(I),I=1,5),DIFMX2
   10 CONTINUE
      IF(PGMODE.EQ.2) GO TO 20
```

```
      WRITE(6,6100) (XSV(I),I=1,4),CIA,CIB,CIC
      WRITE(6,6110) (U(I),I=1,4),VA,VB,VC
   20 CONTINUE
      WRITE(6,6050)
      WRITE(6,6120) SPDHI,SPDLO,SPDPOS,SPDNEG,TPOS,TNEG
      WRITE(6,6130) MTRG1,RGN4,PLUG4,MTRG3,RGN2,PLUG2
      IF(PGMODE.EQ.2) GO TO 30
      WRITE(6,6140) AA,BB,CC,(Q(I),I=1,8),(D(I),I=1,9)
      WRITE(6,6050)
      WRITE(6,6150) PELOSS,PTRM,PEM,PSO,TEM,L2
      WRITE(6,6160) (BOC(I),I=1,3),(BGAP(I),I=1,3)
      WRITE(6,6170) (FS(I),I=1,3),(FM(I),I=1,3),NTRMS3
      WRITE(6,6180) (FG(I),I=1,3),(LMV(I),I=1,3),DIFMX3
      WRITE(6,6050)
      WRITE(6,6190) (BCHEMF(I),I=1,8)
      WRITE(6,6200) (BCHEMF(I),I=9,16)
      WRITE(6,6210) (BCHEMF(I),I=17,23)
      WRITE(6,6220) (BCHCUR(I),I=1,8)
      WRITE(6,6230) (BCHCUR(I),I=9,16)
      WRITE(6,6240) (BCHCUR(I),I=17,23)
      WRITE(6,6250) (BCHPOW(I),I=1,8)
      WRITE(6,6260) (BCHPOW(I),I=9,16)
      WRITE(6,6270) (BCHPOW(I),I=17,23)
   30 CONTINUE
      WRITE(6,6011)
C.......................................................................
C     WRITE FORMATS
C.......................................................................
 6010 FORMAT(1H1,130('.'))
 6011 FORMAT(1H ,130('.'))
 6020 FORMAT(1H ,T4,'PGMODE = ',I1,'  NUMINT = ',I5,'  TIME = ',E12.5,
     1'  I1 = ',I5,'  I2 = ',I5,'  I3 = ',I5,'  RANG = ',E12.5,
```

```
    2 '  RVEL = ',E12.5)
6030 FORMAT(1H ,T4,'DC = ' ,E12.5,'  FANG   = ',E12.5,'  PE     = ',
    1E12.5,'  VE     = ',E12.5,'  TACT   = ',E12.5,'  NMACH = ',I1)
6040 FORMAT(1H ,T4,'IM      = ',E12.5,'  IMC = ',E12.5,'  ICMD1  = ',
    1E12.5,'  ICMDL  = ',E12.5,'  TM     = ',E12.5)
6050 FORMAT(1H ,130('-'))
6060 FORMAT(1H ,T4,'XM  =',5D20.12,' NTRMS1 = ',I2)
6070 FORMAT(1H ,T4,'UM  =',5D20.12,' DIFMX1 = ',E14.7)
6080 FORMAT(1H ,T4,'XV  =',5D20.12,' NTRMS2 = ',I2)
6090 FORMAT(1H ,T4,'UV  =',5D20.12,' DIFMX2 = ',E14.7)
6100 FORMAT(1H ,T4,'XSV =',4D20.12,'  CIA,CIB,CIC=',3F8.3)
6110 FORMAT(1H ,T4,'U   =',4D20.12,'  VA,VB,VC   =',3F8.3)
6120 FORMAT(1H ,T4,'SPDHI = ',L1,'  SPDLO = ',L1,'  SPDPOS = ',L1,
    1'  SPDNEG = ',L1,'  TPOS = ',L1,'  TNEG = ',L1)
6130 FORMAT(1H ,T4,'MTRG1 = ',L1,'  RGN4  = ',L1,'  PLUG4  = ',L1,
    1'  MTRG3  = ',L1,'  RGN2 = ',L1,'  PLUG2 = ',L1)
6140 FORMAT(1H ,T4,'AA,BB,CC =',3L2,T26,'Q1-Q6  =',3L2,1X,3L2,T50,
    1'QMON = ',L1,'  QBON  = ',L1,T75,'D1-D6 = ',3L2,1X,3L2,'    DB = '
    2,L1,'  DM = ',L1,'  DR = ',L1)
6150 FORMAT(1H ,T4,'PELOSS = ',E14.7,'  PTRM = ',E14.7,' PEM  = ',E14.7
    1,' PSO = ',E14.7,' TEM = ',E14.7,' L2=',F10.7)
6160 FORMAT(1H ,T4,'BDC    = ',E14.7,2X,E14.7,2X,E14.7,'  BGAP = ',
    1E14.7,2X,E14.7,2X,E14.7)
6170 FORMAT(1H ,T4,'FS     = ',E14.7,2X,E14.7,2X,E14.7,'  FM   = ',
    1E14.7,2X,E14.7,2X,E14.7,' NTRMS3=',I2)
6180 FORMAT(1H ,T4,'FG     = ',E14.7,2X,E14.7,2X,E14.7,'  LM   = ',
    1 E14.7,2X,E14.7,2X,E14.7,' DIFMX3=',E10.3)
6190 FORMAT(1H ,'VB1 -VB4  =',4D13.5,' VB5 -VB8 =',4D13.5)
6200 FORMAT(1H ,'VB9 -VB12 =',4D13.5,' VB13-VB16=',4D13.5)
6210 FORMAT(1H ,'VB17-VB20 =',4D13.5,' VB21-VB23=',4D13.5)
6220 FORMAT(1H ,'IB1 -IB4  =',4D13.5,' IB5 -IB8 =',4D13.5)
6230 FORMAT(1H ,'IB9 -IB12 =',4D13.5,' IB13-IB16=',4D13.5)
```

```
6240 FORMAT(1H ,'IB17-IB20 =',4D13.5,' IB21-IB23=',4D13.5)
6250 FORMAT(1H ,'PB1 -PB4  =',4D13.5,' PB5 -PB8 =',4D13.5)
6260 FORMAT(1H ,'PB9 -PB12 =',4D13.5,' PB13-PB16=',4D13.5)
6270 FORMAT(1H ,'PB17-PB20 =',4D13.5,' PB21-PB23=',4D13.5)
     RETURN
     END
```

```
      SUBROUTINE DVIO(RVEC,IDIM,NUM,ID,INOUT)
C-----------------------------------------------------------------------
C
C     SUBROUTINE DVIO READS OR PRINTS OR BOTH THE DOUBLE PRECISION
C     VECTOR RVEC
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION NAMES(12),RVEC(IDIM)
      DATA      NAMES(1)/'AV       '/,NAMES(2)/'BV       '/,
     1          NAMES(3)/'PHIV     '/,NAMES(4)/'THETAV   '/,
     2          NAMES(5)/'XV       '/,NAMES(6)/'UV       '/,
     3          NAMES(7)/'AM       '/,NAMES(8)/'BM       '/,
     4          NAMES(9)/'PHIM     '/,NAMES(10)/'THETAM   '/,
     5          NAMES(11)/'XM       '/,NAMES(12)/'UM       '/
C.......................................................................
C     INOUT=1/2  READ AND PRINT RVEC/PRINT RVEC
C.......................................................................
      GO TO (10,20),INOUT
   10 READ(5,15) (RVEC(I),I=1,NUM)
   20 WRITE(6,30) NAMES(ID),(I,RVEC(I),I=1,NUM)
C.......................................................................
C     FORMATS
C.......................................................................
   15 FORMAT(3D21.14)
   30 FORMAT(1H ,///,1H ,'************',/,1H ,'* ',A8,' *',/,
     1  1H ,'************',//,(4(' (',I4,')',D21.14,3X)))
      RETURN
       END
```

```
      SUBROUTINE DMIO(RMAT,IDIM,JDIM,IO,JO,ID,INOUT)
C-----------------------------------------------------------------------
C
C     SUBROUTINE DMIO READS OR PRINTS OR BOTH  THE DOUBLE PRECISION
C     MATRIX DMIO.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION NAMES(12)
      DOUBLE PRECISION RMAT(IDIM,JDIM)
      DATA    NAMES(1)/'AV      '/,NAMES(2)/'BV      '/,
     1        NAMES(3)/'PHIV    '/,NAMES(4)/'THETAV  '/,
     2        NAMES(5)/'XV      '/,NAMES(6)/'UV      '/,
     3        NAMES(7)/'AM      '/,NAMES(8)/'BM      '/,
     4        NAMES(9)/'PHIM    '/,NAMES(10)/'THETAM  '/,
     5        NAMES(11)/'XM      '/,NAMES(12)/'UM      '/
C.......................................................................
C     INOUT=1/2  READ AND PRINT RMAT/PRINT RMAT
C.......................................................................
      GO TO (100,200),INOUT
  100 DO 120 I=1,IO
      READ(5,150) (RMAT(I,J),J=1,JO)
  120 CONTINUE
  200 WRITE(6,250) NAMES(ID)
      DO 400 I=1,IO
      WRITE(6,350) I,(J,RMAT(I,J),J=1,JO)
  400 CONTINUE
C.......................................................................
C     FORMATS
C.......................................................................
  150 FORMAT(3D21.14)
  250 FORMAT(1H1,///,1H ,'************',/,1H ,'* ',A8,' *',/,
     1 1H ,'************',/)
```

```
350 FORMAT(1H0,'ROW',I5,/,' ---------',/,(4(' (',I4,')',D21.14,3X)))
    RETURN
    END
```

```
      SUBROUTINE PDATA(PGMODE,NREC,IREWND)
C-----------------------------------------------------------------------
C
C     SUBROUTINE PDATA LOADS THE PLOT DATA ONTO A SEQUENTIAL DISK FILE.
C
C-----------------------------------------------------------------------
      DIMENSION  EBLK1(21),EBLK6(4),EBLK7(4),EBLK12(27),EBLK13(18),
     1           EBLK28(4)
      DOUBLE PRECISION  EBLK2(69),EBLK15(12),EBLK17(30),EBLK20(30)
      DOUBLE PRECISION BCHCUR(23),BCHEMF(23),BCHPOW(23)
      DOUBLE PRECISION XSV(4),U(4),XSVDOT(4)
      DOUBLE PRECISION XV(5),UV(5),XVOLD(5),UVOLD(5),XVI(5),UVI(5)
      DOUBLE PRECISION  XM(5),UM(5),XMOLD(5),UMOLD(5),XMI(5),UMI(5)
      INTEGER PGMODE
      LOGICAL           AA,BB,CC,FF,AN,BN,CN,FN,QAP1,QAN1,QBP1,QBN1,QCP1,
     1                  QCN1,QMON,QBON,QIOFF,Q(8),NETCHG,NDUMP,D(9)
     2                 ,QBC,QMC
      LOGICAL SPDHI,SPDLO,SPDPOS,SPDNEG,TPOS,TNEG,MTRG1,RGN4,PLUG4,
     1        MTRG3,RGN2,PLUG2
      REAL              LMV(3),LGV(3)
      REAL  ICMDL,ICMD1,IMC,IM,REC(100)
      COMMON/BLK1  /    CIA,CIB,CIC,VA,VB,VC,VAB,VBC,VCA,PA,PB,PC,PCORE,
     1                  PML,PTRM,PSO,PEM,PGM,PNM,TEM,TM
      COMMON/BLK6  /    RRANG,RANG,RVEL,RACEL
      COMMON/BLK7/   ICMDL,ICMD1,IMC,IM
      COMMON/BLK8  /    AA,BB,CC,FF,AN,BN,CN,FN,QAP1,QAN1,QBP1,QBN1,QCP1,
     1               QCN1,QMON,QBON,QIOFF,Q,NETCHG,NDUMP,D,QBC,QMC
      COMMON/BLK12 /    BOC(3),BGAP(3),FS(3),FM(3),FG(3),HM(3),HG(3),LMV,
     1                  LGV
      COMMON/BLK13 /    QLOSS(8),DLOSS(9),PELOSS
      COMMON/BLK30 /    INDEX(100)
      COMMON/BLK27 /    TSNET,ETNET,NTERM3,DIFMAX,ITER,TIME
```

```
      COMMON/BLK28 /  FANG,PE,VE,TACT
      COMMON/BLK29 /  NPLTS,IDX,NPAGES,NVSTOR
      COMMON/BLK2  /  BCHCUR,BCHEMF,BCHPOW
      COMMON/BLK15 /  XSV,U,XSVDOT
      COMMON/BLK17 /  XV,UV,XVOLD,UVOLD,XVI,UVI
      COMMON/BLK20 /  XM,UM,XMOLD,UMOLD,XMI,UMI
      COMMON/MODE  /  SPDHI,SPDLO,SPDPOS,SPDNEG,TPOS,TNEG,MTRG1,RGN4,
     1                PLUG4,MTRG3,RGN2,PLUG2
      EQUIVALENCE  (EBLK1(1),CIA)
      EQUIVALENCE  (EBLK6(1),RRANG)
      EQUIVALENCE  (EBLK7(1),ICMDL)
      EQUIVALENCE  (EBLK12(1),BOC(1))
      EQUIVALENCE  (EBLK13(1),QLOSS(1))
      EQUIVALENCE  (EBLK28(1),FANG)
      EQUIVALENCE  (EBLK2(1),BCHCUR(1))
      EQUIVALENCE  (EBLK15(1),XSV(1))
      EQUIVALENCE  (EBLK17(1),XV(1))
      EQUIVALENCE  (EBLK20(1),XM(1))
C.....................................................................
C    REWIND DISK FILE IF SPECIFIED
C.....................................................................
      IF(IREWND.EQ.1) REWIND8
C.....................................................................
C    LOAD SINGLE AND DOUBLE PRECISION DATA
C.....................................................................
      NREC=NREC+1
      DO 2000 I=1,NVSTOR
      ID=INDEX(I)
      IF(ID.LE.21) REC(I)=EBLK1(ID)
      IF(ID.GE.22.AND.ID.LE.25) REC(I)=EBLK6(ID-21)
      IF(ID.GE.26.AND.ID.LE.29) REC(I)=EBLK7(ID-25)
      IF(ID.GE.30.AND.ID.LE.56) REC(I)=EBLK12(ID-29)
```

```
      IF(ID.GE.57.AND.ID.LE.74) REC(I)=EBLK13(ID-56)
      IF(ID.GE.75.AND.ID.LE.78) REC(I)=EBLK28(ID-74)
      IF(ID.EQ.79) REC(I)=TIME
      IF(ID.GE.201.AND.ID.LE.269) REC(I)=EBLK2(ID-200)
      IF(ID.GE.270.AND.ID.LE.281) REC(I)=EBLK15(ID-269)
      IF(ID.GE.282.AND.ID.LE.311) REC(I)=EBLK17(ID-281)
      IF(ID.GE.312.AND.ID.LE.341) REC(I)=EBLK20(ID-311)
 2000 CONTINUE
      WRITE(8,6100) (REC(I),I=1,NVSTOR)
C......................................................................
C    LOAD THE LOGICAL DATA
C......................................................................
      IF(PGMODE.NE.2) WRITE(8,6200)
     1 SPDHI,SPDLO,SPDPOS,SPDNEG,TPOS,TNEG,MTRG1,RGN4,PLUG4,MTRG3,RGN2,
     2 PLUG2,AA,BB,CC,(Q(I),I=1,8),(D(I),I=1,9)
      IF(PGMODE.EQ.2) WRITE(8,6200)
     1 SPDHI,SPDLO,SPDPOS,SPDNEG,TPOS,TNEG,MTRG1,RGN4,PLUG4,MTRG3,RGN2,
     2 PLUG2
C......................................................................
C    FORMATS
C......................................................................
 6100 FORMAT(5E14.7)
 6200 FORMAT(80L1)
      RETURN
      END
```

```
      SUBROUTINE EMAPLT(NREC)
C-----------------------------------------------------------------------
C
C     SUBROUTINE EMAPLT PLOTS THE DESIRED LOGICAL, SINGLE PRECISION AND
C     DOUBLE PRECISION VARIABLES.
C
C-----------------------------------------------------------------------
      INTEGER PGMODE
      LOGICAL LREC(32)
      REAL YMINV(5),YMAXV(5),REC(100)
      REAL XLAB(10),YLAB(10),TITLE(10),X(1001),Y(1001),PVEC(1001,5)
      COMMON/BLK22 /  PGMODE,IOMODE,IDISK,IPLT
      COMMON/BLK29 /  NPLTS,IDX,NPAGES,NVSTOR
C.......................................................................
C     READ AND PRINT THE PLOT CONTROL CARDS AND PLOT HEADINGS.
C.......................................................................
      READ(5,5300) (XLAB(I),I=1,10)
C     ------------------------------------------------------------------
C     READ IN X FOR SUBROUTINE BOLPLT.
C
      XMIN=1.E20
      XMAX=-1.E20
      REWIND8
      NLOG=32
      IF(PGMODE.EQ.2) NLOG=12
C     ------------------------------------------------------------------
C     FIND XMIN AND XMAX
C
      DO 100 I=1,NREC
      READ(8,5100) (REC(J),J=1,NVSTOR)
      X(I)=REC(IDX)
      IF(X(I).LT.XMIN) XMIN=X(I)
```

```
      IF(X(I).GT.XMAX) XMAX=X(I)
      READ(8,5200) (LREC(J),J=1,NLOC)
  100 CONTINUE
C.......................................................................
C    PLOT THE LOGICAL VARIABLES
C.......................................................................
      CALL BOLPLT(PGMODE,X,XMIN,XMAX,NREC,NVSTOR,NPAGES)
      IF(NPLTS.EQ.0) RETURN
C.......................................................................
C    PLOT THE SINGLE AND DOUBLE PRECISION DATA
C.......................................................................
      NCDIM=(NPLTS-1)/5+1
      NREM=MOD(NPLTS,5)
      IE=5
      IF(NPLTS.LT.5) IE=NREM
      NDO=12
      IF(PGMODE.NE.2) NDO=32
      DO 2000 L=1,NCDIM
      DO 150 I=1,5
      YMINV(I)=1.E20
      YMAXV(I)=-1.E20
  150 CONTINUE
      REWIND8
      IF(L.EQ.NCDIM.AND.NREM.NE.0) IE=NREM
      DO 1000 K=1,NREC
      READ(8,5100) (REC(I),I=1,NVSTOR)
      READ(8,5200) (LREC(I),I=1,NDO)
      DO 500 I=1,IE
      ID=(L-1)*5+I
      PVEC(K,I)=REC(ID)
      IF(PVEC(K,I).LT.YMINV(I)) YMINV(I)=PVEC(K,I)
      IF(PVEC(K,I).GT.YMAXV(I)) YMAXV(I)=PVEC(K,I)
```

```
  500 CONTINUE
 1000 CONTINUE
      DO 1500 I=1,IE
      READ(5,5300) (TITLE(J),J=1,10),(YLAB(J),J=1,10)
      DO 1300 J=1,K
 1300 Y(J)=PVEC(J,I)
      YMIN=YMINV(I)
      YMAX=YMAXV(I)
      CALL PGPLOT(X,Y,K,NPAGES,XMIN,XMAX,YMIN,YMAX,XLAB,YLAB,TITLE)
 1500 CONTINUE
 2000 CONTINUE
C.....................................................................
C     FORMATS
C.....................................................................
 5100 FORMAT(5E14.7)
 5200 FORMAT(80L1)
 5300 FORMAT(20A4)
      RETURN
      END
```

```
      SUBROUTINE BOLPLT(PGMODE,X,XMIN,XMAX,NPTS,NVSTOR,NPAGES)
C-----------------------------------------------------------------------
C
C     SUBROUTINE BOLPLT PLOTS THE EMA LOGIC SIGNALS VERSUS TIME.
C
C-----------------------------------------------------------------------
      INTEGER PGMODE
      LOGICAL LREC(32),LPLOT(32)
      LOGICAL             AA,BB,CC,FF,AN,BN,CN,FN,QAP1,QAN1,QBP1,QBN1,QCP1,
     1                    QCN1,QMON,QBON,QIOFF,Q(8),NETCHG,NDUMP,D(9)
     2                   ,QBC,QMC
      LOGICAL SPDHI,SPDLO,SPDPOS,SPDNEG,TPOS,TNEG,MTRG1,RGN4,PLUG4,
     1        MTRG3,RGN2,PLUG2
      REAL X(1001),LGNAME(6,32),REC(100)
      REAL LAB1(72),LAB2(60),LAB3(60)
      COMMON/BLK8  /  AA,BB,CC,FF,AN,BN,CN,FN,QAP1,QAN1,QBP1,QBN1,QCP1,
     1               QCN1,QMON,QEON,QIOFF,Q,NETCHG,NDUMP,D,QBC,QMC
      COMMON/MODE  /  SPDHI,SPDLO,SPDPOS,SPDNEG,TPOS,TNEG,MTRG1,RGN4,
     1                PLUG4,MTRG3,RGN2,PLUG2
      DATA LAB1
     1/'S','P','D','H','I',' ',  'S','P','D','L','O',' ',
     2 'S','P','D','P','O','S',  'S','P','D','N','E','G',
     3'T','P','O','S',' ',' ',  'T','N','E','G',' ',' ',
     4 'H','T','R','G','1',' ',  'R','G','N','4',' ',' ',
     5 'P','L','U','G','4',' ',  'M','T','R','G','3',' ',
     6 'R','G','N','2',' ',' ',  'P','L','U','G','2',' '/
      DATA LAB2
     1/'A','A',' ',' ',' ',' ',  'B','B',' ',' ',' ',' ',
     2 'C','C',' ',' ',' ',' ',  'Q','1',' ',' ',' ',' ',
     3 'Q','2',' ',' ',' ',' ',  'Q','3',' ',' ',' ',' ',
     4 'Q','4',' ',' ',' ',' ',  'Q','5',' ',' ',' ',' ',
     5 'Q','6',' ',' ',' ',' ',  'Q','M','O','N',' ',' '/
```

Pgs. A-110 thru A-119
PRECEDING PAGE BLANK NOT FILMED

```
      DATA LAB3
     1/'Q','B','O','N',' ',' ',' ',  'D','1',' ',' ',' ',' ',
     2 'D','2',' ',' ',' ',' ',' ',  'D','3',' ',' ',' ',' ',
     3 'D','4',' ',' ',' ',' ',' ',  'D','5',' ',' ',' ',' ',
     4 'D','6',' ',' ',' ',' ',' ',  'D','B',' ',' ',' ',' ',
     5 'D','M',' ',' ',' ',' ',' ',  'D','R',' ',' ',' ',' '/
C.......................................................................
C     REWIND THE DISK
C.......................................................................
      REWIND8
C.......................................................................
C     INITIALIZE
C.......................................................................
      K=0
      DO 10 J=1,12
      DO 10 I=1,6
      K=K+1
      LGNAME(I,J)=LAB1(K)
   10 CONTINUE
      K=0
      DO 20 J=13,22
      DO 20 I=1,6
      K=K+1
      LGNAME(I,J)=LAB2(K)
   20 CONTINUE
      K=0
      DO 30 J=23,32
      DO 30 I=1,6
      K=K+1
      LGNAME(I,J)=LAB3(K)
   30 CONTINUE
      IRUN=1
```

```
      NCOUNT=0
      NROWS=50*NPAGES
      XINC=ABS(XMAX-XMIN)/(NROWS-1)
      IF(NPTS.LT.NROWS) NPAGES=1+NPTS/50
C.....................................................................
C     START THE PAGE PLOTS (50 ROWS/PAGE)
C.....................................................................
      DO 4000 IPAGE=1,NPAGES
C     ----------------------------------------------------------------
C     SELECT PROPER PLOTTING ROUTINE
C
      GO TO (1000,2000,1000),PGMODE
C.....................................................................
C     PLOT MODE,INVERTER AND CHOPPER LOGIC SIGNALS
C.....................................................................
 1000 CONTINUE
C     ----------------------------------------------------------------
C     PLOT HEADING
C
      WRITE(6,6010) IPAGE,NPAGES
      DO 1050 I=1,6
      WRITE(6,6020) (LGNAME(I,J),J=1,32)
 1050 CONTINUE
      WRITE(6,6030)
      DO 1500 L=1,50
      IF(NCOUNT.EQ.NPTS) GO TO 4500
      XX=(IROW-1)*XINC+XMIN
      ISTART=NCOUNT+1
C     ----------------------------------------------------------------
C     LOAD PLOT VECTOR
C
      DO 1100 I=1,32
```

```
 1100 LPLOT(I)=.FALSE.
      DO 1300 I=ISTART,NPTS
      IF(X(I).GT.(XX+XINC/2)) GO TO 1400
      NCOUNT=NCOUNT+1
C     ------------------------------------------------------------------
C     READ LOGICAL DATA OFF THE DISK
C
      READ(8,5100) (REC(J),J=1,NVSTOR)
      READ(8,5200) (LREC(J),J=1,32)
      DO 1200 J=1,32
 1200 LPLOT(J)=LPLOT(J).OR.LREC(J)
 1300 CONTINUE
 1400 CONTINUE
      IROW=IROW+1
C     ------------------------------------------------------------------
C     PRINT THE PLOT VECTOR
C
      WRITE(6,6040) L,XX,(LPLOT(I),I=1,32)
 1500 CONTINUE
      GO TO 3000
C......................................................................
C     PLOT ONLY THE EMA MODE DECODER LOGIC SIGNALS
C......................................................................
 2000 CONTINUE
C     ------------------------------------------------------------------
C     PLOT HEADING
C
      WRITE(6,6050) IPAGE,NPAGES
      DO 2050 I=1,6
      WRITE(6,6060) (LGNAME(I,J),J=1,12)
 2050 CONTINUE
      WRITE(6,6070)
```

```
      DO 2500 L=1,50
      IF(NCOUNT.EQ.NPTS) GO TO 4500
      XX=(IROW-1)*XINC+XMIN
      ISTART=NCOUNT+1
C     ----------------------------------------------------------------
C     LOAD PLOT VECTOR
C
      DO 2100 I=1,12
 2100 LPLOT(I)=.FALSE.
      DO 2300 I=ISTART,NPTS
      IF(X(I).GT.(XX+XINC/2)) GO TO 2400
      NCOUNT=NCOUNT+1
C     ----------------------------------------------------------------
C     READ LOGICAL DATA OFF THE DISK
C
      READ(8,5100) (REC(J),J=1,NVSTOR)
      READ(8,5200) (LREC(J),J=1,12)
      DO 2200 J=1,12
 2200 LPLOT(J)=LPLOT(J).OR.LREC(J)
 2300 CONTINUE
 2400 CONTINUE
      IROW=IROW+1
C     ----------------------------------------------------------------
C     PRINT THE PLOT VECTOR
C
      WRITE(6,6080) L,XX,(LPLOT(I),I=1,12)
 2500 CONTINUE
 3000 CONTINUE
 4000 CONTINUE
 4500 CONTINUE
C.................................................................
C     FORMATS
```

```
C.......................................................................
 5100 FORMAT(5E14.7)
 5200 FORMAT(80L1)
 6010 FORMAT(1H1,T50,'EMA LOGIC SIGNALS VERSUS TIME',T90,
     1 'PAGE',I3,' OF ',I3,//)
 6020 FORMAT(1H ,T28,32(A1,2X))
 6030 FORMAT(1H ,T28,32('+',2X))
 6040 FORMAT(1H ,T6,I2,3X,E14.7,T28,32(L1,2X))
 6050 FORMAT(1H1,T45,'EMA MODE DECODER LOGIC SIGNALS',T90,
     1 'PAGE',I3,' OF ',I3,//)
 6060 FORMAT(1H ,T58,12(A1,2X))
 6070 FORMAT(1H ,T58,12('+',2X))
 6080 FORMAT(1H ,T36,I2,3X,E14.7,T58,12(L1,2X))
      RETURN
      END
```

```
      SUBROUTINE PGPLOT(X,Y,NPTS,NPAGES,XMIN,XMAX,YMIN,YMAX,XLAB,YLAB,
     1TITLE)
C-----------------------------------------------------------------------
C
C     SUBROUTINE PGPLOT PRODUCES A PAGE PLOT OF Y VERSUS X
C
C-----------------------------------------------------------------------
      DIMENSION X(1001),Y(1001),XLAB(10),YLAB(10),TITLE(10),YTIC(5)
      DIMENSION PLINE(101)
C.......................................................................
C     INITIALIZE
C.......................................................................
      IROW=1
      NCOUNT=0
      NROWS=50*NPAGES
      XINC=ABS(XMAX-XMIN)/(NROWS-1)
C.......................................................................
C      START THE PAGE PLOTS
C.......................................................................
      DO 3000 IPAGE=1,NPAGES
C     ------------------------------------------------------------------
C     PLOT HEADINGS
C
      WRITE(6,6001) (TITLE(I),I=1,10),IPAGE,NPAGES
 6001 FORMAT(1H1,T20,'PLOT TITLE:  ',10A4,T90,'PAGE',I3,' OF ',I3)
      WRITE(6,6002) (XLAB(I),I=1,10)
 6002 FORMAT(1H0,T33,'X-AXIS LABEL:  ',10A4)
      WRITE(6,6003) (YLAB(I),I=1,10)
 6003 FORMAT(1H ,T33,'Y-AXIS LABEL:  ',10A4)
C     ------------------------------------------------------------------
C     DETERMINE THE Y-AXIS INCREMENT
C
```

```
      YINC=ABS(YMAX-YMIN)/4.
      DO 100 I=1,5
      YTIC(I)=(I-1)*YINC+YMIN
  100 CONTINUE
C     ------------------------------------------------------------------
C     PRINT THE Y-AXIS TIC MARK LABELS
C
      WRITE(6,6004) (YTIC(I),I=1,5)
 6004 FORMAT(1H0,T5,5(11X,E14.7))
C     ------------------------------------------------------------------
C     GENERATE AND PRINT THE FIRST ROW
C
      CALL LINE(1,PLINE)
      WRITE(6,6010) (PLINE(I),I=1,101)
 6010 FORMAT(' ',T20,101A1)
C     ------------------------------------------------------------------
C     GENERATE AND PRINT THE REMAINING 50 ROWS
C
      DO 2000 I=1,5
      DO 1000 J=1,9
      CALL LINE(2,PLINE)
      XX=(IROW-1)*XINC+XMIN
      IF(NCOUNT.LT.NPTS)
     1CALL YLOAD(IROW,NCOUNT,NPTS,X,Y,XMIN,XINC,YMIN,YMAX,XX,PLINE)
      IROW=IROW+1
      WRITE(6,6020) XX,(PLINE(K),K=1,101)
 6020 FORMAT(' ',T5,E14.7,T20,101A1)
 1000 CONTINUE
      CALL LINE(1,PLINE)
      XX=(IROW-1)*XINC+XMIN
      IF(NCOUNT.LT.NPTS)
     1CALL YLOAD(IROW,NCOUNT,NPTS,X,Y,XMIN,XINC,YMIN,YMAX,XX,PLINE)
```

```
      IROW=IROW+1
      WRITE(6,6020) XX,(PLINE(K),K=1,101    )
 2000 CONTINUE
 3000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  YLOAD(IROW,NCOUNT,NPTS,X,Y,XMIN,XINC,YMIN,YMAX,
     1 XX,PLINE)
C-----------------------------------------------------------------------
C
C     SUBROUTINE YLOAD LOADS THE PLOT POINTS INTO THE PLOT VECTOR PLINE
C
C-----------------------------------------------------------------------
      DIMENSION X(1001),Y(1001),PLINE(101)
      DATA BLNK,PLUS,EYE,STAR,DOT,DASH
     1 /' ','+','I','*','.','-'/
      ISTART=NCOUNT+1
      DO 1000 I=ISTART,NPTS
      IF(X(I).GT.(XX+XINC/2)) RETURN
      NCOUNT=NCOUNT+1
      IYPOS=ABS((Y(I)-YMIN)/(YMAX-YMIN))*100+1
      PLINE(IYPOS)=STAR
 1000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LINE(LTYPE,PLINE)
C-----------------------------------------------------------------------
C
C     GENERATE HORIZONTAL PLUS-DASH LINE (LINE TYPE=1)
C     GENERATE HORIZONTAL I-BLANK LINE (LINE TYPE=2)
C
C-----------------------------------------------------------------------
      DATA BLNK,PLUS, EYE,STAR,DOT,DASH
     1 /' ','+','I','*','.','-'/
      DIMENSION PLINE(101)
      C1=PLUS
      C2=DASH
      IF(LTYPE.EQ.2) C1= EYE
      IF(LTYPE.EQ.2) C2=BLNK
      IY=1
      DO 400 I=1,4
      PLINE(IY)=C1
      ISTART=IY+1
      ISTOP=IY+24
      DO 200 J=ISTART,ISTOP
      IY=J
      PLINE(IY)=C2
  200 CONTINUE
      IY=IY+1
  400 CONTINUE
      PLINE(IY)=C1
      RETURN
      END
```