NASA Contractor Report 145332

(NASA-CR-145332) REXOR 2 ROTORCRAFT SIMULATION MODEL. VOLUME 2: COMPUTER IMPLEMENTATION Final Technical Report (Lockheed-California Co., Burbank.) 177 p HC A09/MF A01 CSCL 01A G3/02 N78-30043 Unclas 28598

REXOR II ROTORCRAFT SIMULATION MODEL

VOLUME II - COMPUTER IMPLEMENTATION

J. S. Reaser,
P. H. Kretsinger

LOCKHEED-CALIFORNIA CO.
P.O. BOX 551
BURBANK, CALIF. 91520

CONTRACT NAS1-14570
JUNE 1978

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

FOREWORD

This report describes a nonlinear rotorcraft model and associated computer software which has been developed and documented for NASA, Langley Research Center, Hampton, Virginia under contract NAS1-14570 (July 1976). This work has been performed by the Lockheed-California Company, Burbank, California.

P. H. Kretsinger (Lockheed) performed the software implementation. W. D. Anderson and Fox Conner (both of Lockheed) assisted in preparation of the program.

PRECEDING PAGE BLANK NOT FILMED

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

REXOR II ROTORCRAFT SIMULATION MODEL*

Volume II — Computer Implementation

J. S. Reaser and P. H. Kretsinger

Lockheed-California Company

SUMMARY

This report describes a generalized format rotorcraft nonlinear simulation called REXOR II. The program models single main rotor vehicles with up to seven main rotor blades. Wings, two horizontal tail planes, and auxiliary thrustors may be included to model a variety of compound helicopter configurations.

Program output is primarily in the form of machine plotted time histories specified from a signal list. This list is, in turn, user selected from a set of computation variables used by the program.

LIST OF SYMBOLS

The symbols used in the REXOR II equations are quite numerous. In order to keep the catalog of symbols to manageable proportions a list of basic symbols is given, followed by subscripts, superscripts, and postscripts. Nonconforming cases of usage together with complicated or obscure subscripting are fully annotated in the basic list.

SYMBOLS

a	arbitrary vector
a_s	speed of sound, m/s
$\ddot{a}_0$	acceleration vector, m/s (ft/s^2)
a_1	longitudinal component of blade first harmonic flapping, rad
$[A]$	generalized mass element matrix
$A_{1,2,3}$	modal variables
A_{1n}	generalized displacement of nth blade, first mode

*The contract research effort which has lead to the results in this report was financially supported by USARTL (AVRADCOM) Structures Laboratory.

A_{2n}	generalized displacement of nth blade, second mode
A_{3n}	generalized displacement of nth blade, third mode
A_{1S}	cosine component of blade first harmonic cyclic, rad
b	number of main rotor blades; arbitrary vector
B	dissipation function
B_{1S}	sine component of blade first harmonic cyclic, rad
c	blade segment chord, m (ft)
[C]	damping matrix
C_D	aerodynamic drag coefficient
C_L	aerodynamic lift coefficient
C_M	aerodynamic pitching moment coefficient
C_P	power coefficient
C_T	thrust coefficient
$C_{X,Y,Z}$	linear damping, N/m/s (lb/ft/s)
$C_{\phi,\theta,\psi}$	rotary damping, N-m/rad/s (ft-lb/rad/s)
$C_{1,2,3}$	blade bending to feathering couplings
C(k)	lift deficiency function
d	infinitesimal increment
dr	increment in rotor, radius, m (ft)
dt	increment in time
d/dt	derivative with respect to time
$(d/e)_0$	swashplate to feather gear ratio, zero collective
$(d/e)_1$	swashplate to feather gear ratio slope with collective
e	pitch horn effective crank arm, m (ft)
EI	blade bending stiffness distribution, N-m^2 (lb-ft^2)
f_{iMR}	ground effect factor for main rotor

F	factor; force, N (lb)
$F_{X,Y,Z}$	force components along X,Y,Z directions, N (lb)
$F_{\phi,\theta,\psi}$	generalized force about ϕ, θ, ψ axis
$F_{\beta PH}$	feathering mode generalized force
g	gravity, m/s^2 (ft/s^2)
$g_{X,Y,Z}$	gravity components along X,Y,Z directions
G	gear ratio
$\{G\}$	generalized force vector
$\ddot{G}$	gyro angular acceleration partial product
GJ	blade torsional stiffness, $N\text{-}m^2$ ($lb - ft^2$)
I_X	$= \Sigma m_i X_i^2$, $kg\text{-}m^2$ ($slug\text{-}ft^2$)
I_Y	$= \Sigma m_i Y_i^2$, $kg\text{-}m^2$ ($slug\text{-}ft^2$)
I_Z	$= \Sigma m_i Z_i^2$, $kg\text{-}m^2$ ($slug\text{-}ft^2$)
I_{XX}	$= \Sigma m_i (Y_i^2 + Z_i^2)$, $kg\text{-}m^2$ ($slug\text{-}ft^2$)
I_{YY}	$= \Sigma m_i (X_i^2 + Z_i^2)$, $kg\text{-}m^2$ ($slug\text{-}ft^2$)
I_{ZZ}	$= \Sigma m_i (X_i^2 + Y_i^2)$, $kg\text{-}m^2$ ($slug\text{-}ft^2$)
I_{XY}	$= \Sigma m_i X_i Y_i$, $kg\text{-}m^2$ ($slug\text{-}ft^2$)
I_{XZ}	$= \Sigma m_i X_i Z_i$, $kg\text{-}m^2$ ($slug\text{-}ft^2$)
I_{YZ}	$= \Sigma m_i Y_i Z_i$, $kg\text{-}m^2$ ($slug\text{-}ft^2$)
i	unit vector
j	unit vector
J	advance ratio
k	number of blade radial stations; reduced frequency, rad/s; unit vector
$[K]$	spring matrix
K_{mj}	blade spring matrix element

$K_{X,Y,Z}$	spring constants along X,Y,Z direction, N/m (lb/ft)
$K_{\phi,\theta,\psi}$	spring rates about ϕ, θ, ψ axis, N-m/rad (ft-lb/rad)
l_{IB}	location inboard feather bearing, m (ft)
l_{OB}	location outboard feather bearing, m (ft)
l_p	radial location of intersection of precone and feather axis, m (ft)
L	rolling moment, N-m (ft-lb)
m	mass of element, kg (slugs)
m_F	summed fuselage coordinate mass, kg (slugs)
m_H	summed hub axis mass, kg (slugs)
m_i	mass of ith particle or blade segment, kg (slugs)
m_{SP}	swashplate summed mass, kg (slugs)
M	pitching moment, N-m (ft-lb); = Σm_i, kg (slugs); mach number
$[M]$	generalized mass matrix
M_{rk}	generalized mass matrix element
$M_{\overline{X}}$	= $\Sigma m_i X_i$, kg-m (slug-ft)
$M_{\overline{Y}}$	= $\Sigma m_i Y_i$, kg-m (slug-ft)
$M_{\overline{Z}}$	= $\Sigma m_i Z_i$, kg-m (slug-ft)
$M_{X,Y,Z}$	moments about X,Y,Z axis, N-m (ft-lb)
M_{φ}	blade torsional moment, N-m/m (ft-lb/ft)
N	number of system particles
p	angular velocity about X axis, rad/s; particle
p_{iMR}	main rotor pitch moment inflow, m/s (ft/s)
q	generalized coordinate; angular velocity about Y axis, rad/s

q_{iMR}	main rotor roll moment inflow, m/s (ft/s)
Q	generalized forcing function
Q_A	aerodynamic pressure times reference wing area, kg (lb)
QLOADS	total nonmain rotor aerodynamic loads matrix
Q_{TR}	tail rotor torque, N-m (ft-lb)
r	general vector; radius of curvature, ft; angular velocity about Z axis, rad/sec; notation for (X,Y,Z)
r_S	static blade shape
R	vector displacement of particle p in X,Y,Z axis system
R_0	vector displacement of x,y,z origin in X,Y,Z system
$R_{Z\phi,Z}$	gyro damper coupling ratios
S	Laplace variable, path of motion of particle p
S_{NA}	blade spline length along neutral axis locii, m (ft)
t	time
T	kinetic energy, N-m (ft-lb)
[T]	transformation of coordinates matrix
u	velocity in X direction, m/s (ft/s)
U	potential energy function, N-m (ft-lb); strain energy, N-m (ft-lb)
$U_{C,P,S,T}$	air velocity on blade element, m/s (ft/sec)
v	velocity in Y direction, m/s (ft/sec)
V_T	trajectory velocity
w	velocity in Z direction, m/s (ft/sec)
w_{iMR}	main rotor collective inflow, m/s (ft/sec)
w_{iTR}	tail rotor collective inflow, m/s (ft/sec)
x	motion in X direction, m (ft); blade span location

X coordinate direction; axis; deflection, m (ft); location, m (ft); cross product

X_{SW} blade radial station of sweep and jog, m (ft)

X_T trajectory path, m (ft)

X_{TR} tail rotor longitudinal force, N (lb)

y motion in Y direction, m (ft)

Y coordinate direction; axis; deflection, m (ft); location, m (ft)

$Y_{TTO_{1,2,3}}$ tension torsion pack outboard end modal coefficients

Y_{ONA} difference between Y direction locations of cg and neutral axis points of blade element, m (ft)

z motion in Z direction

Z coordinate direction; axis; deflection, m (ft); location, m (ft)

Z_{SP} relative swashplate vertical displacement with respect to the hub, m (ft)

$Z_{TTO_{1,2,3}}$ tension-torsion pack outboard end modal coefficients

Z_{OBL} teetering rotor undersling, m (ft)

Z_{OF} hub set distance above fuselage set, m (ft)

Z_{OSP} hub set distance above swashplate set, m (ft)

α angle of attack, rad

α_2 angle of attack with hub set, rad

β sideslip angle, rad

β_{FA} blade feathering angle, rad

β_{PHn} feathering/pitch-horn bending or dynamic torsion generalized coordinate displacement

β_0 blade droop relative to precone angle, rad

γ	blade sweep angle, rad; dynamic stall delay, s
γ_T	trajectory path angle with E set, rad
δ	limit deflection, rad; freeplay, rad; small increment
δ_{3TR}	tail rotor pitch - flap coupling
$\partial\epsilon/\partial\alpha$	downwash factor of wing on horizontal tail
ζ	vector notation of ϕ, θ, ψ
θ	rotation about Y axis, rad
θ_0	collective blade angle, rad
Λ	sideslip at blade element, rad
ρ	air density, kg/m^3, (slugs/ft^3)
τ	time constant, s; natural period, s
τ_0	feathering axis precone, rad
ϕ	rotation about X axis, rad
ϕ_F	feathering angle, rad
ϕ_{Fn}	feathering angle of blade element of nth blade, rad
ϕ_{REF}	blade root reference feather angle, rad
ϕ_T	blade torsion, rad
ϕ_T	sum of blade twist and torsion, rad
χ_{iMR}	wake angle of main rotor, rad, (deg)
ψ	rotation about Z axis, rad; sideslip angle with hub set, rad
ψ_c	control input axis rotation from swashplate, rad
ψ_{PH}	pitch lead angle, rad, (deg)
ψ_T	trajectory path yaw with E set, rad
ψ_W	main rotor apparent airflow angle, rad
ω	rotational speed, rad/s; angular velocity, rad/s; natural frequency, rad/s

∂ partial derivative, derivation

SUBSCRIPTS

a	arbitrary coordinate set a
A	due to aerodynamics
b	arbitrary coordinate set b
BEND	associated with blade elastic bending
BLE	blade element coordinate system
BLn	blade reference axis system for the nth blade
C	associated with pilot control input, chordwise
CG	associated with center of gravity location
CORR	corrective, correction
DW	referring to downwash
DYN	referring to dynamic component
E	earth axis
ENG	associated with powerplant - engine
EST	estimated
F	fuselage axis; associated with blade feathering
FA	referring to blade feather axis
FB	associated with feedback
Fn	associated with feathering of the nth blade
FR	due to friction
G	referring to gyro or gyro coordinate system
GEN	associated with gas generator section of powerplant
GFB	associated with gyro control feedback
GSP	gyro to swashplate connection

GUB	relating to gyro gimbal unbalance
H	referring to hub or principal reference axis system
HT	associated with horizontal tail
i	referring to inflow, particle
IB	referring to inboard feather bearing location
j	spring matrix index
jog	associated with blade attachment joggle
J	associated with gyro end of feedback rod linkage
Jn	associated with feedback rod coming from the nth blade
k	generalized mass index
LAG	associated with lead-lag damper
LIMIT	signifying limiting value
m	blade mode index, spring matrix index
MR	associated with main rotor
n	blade number index
NA	referring to blade segment neutral axis
NEW	newly determined value
NO	normal (to airflow) component
NR	pertaining to nonrotating value
OB	referring to outboard feather bearing location
OLD	value from previous time step
P	associated with propeller; perpendicular blade component
PH	referring to pitch horn
r	generalized mass index
R	referring to rotor axis system

REF	associated with blade feather reference value
RM	referring to control gyro feedback lever moment
S	referring to blade spanwise velocity; general mode; static; structural; shaft
SC	referring to blade segment shear center
SP	referring to swashplate
SP_c	command to swashplate
S, SP	referring to swashplate limit stop
STEADY	steady component
SW	referring to blade sweep angle location
T	associated with trajectory path relating to E axis; tangential blade component; blade torsion; blade twist
TR	associated with the tail rotor
TRIM	initial or trim value
TW	associated with blade twist (built in)
UB	relating to control gyro unbalance
UNSTEADY	associated with unsteady component
VT	associated with vertical tail
WING	associated with the wing
X	relating to component in X direction
Y	relating to component in Y direction
YA	relating to aerodynamic component in Y direction
Z	relating to component in Z direction
ZA	relating to aerodynamic component in Y direction

0	(nought) associated with collective value, coordinate axis value, with respect to principal reference axis, blade root summation
1,2,3	with respect to blade modes 1, 2, or 3
1S	first harmonic component shaft axis feathering
1/4 c	with respect to blade 1/4 chord
3/4 c	with respect to blade 3/4 chord
β_{PHn}	associated with the feathering mode of the nth blade
ϕ	relating to component in the ϕ direction
θ	relating to component in the θ direction
ψ	relating to component in the ψ direction

SUPERSCRIPTS

I	referring to inertial reference
T	matrix transpose
$(\bar{\ })$	(bar) average quantity
$(\ ')$	(prime) slope with respect to blade span
$(\dot{\ })$	(dot) time derivative of basic quantity
$(\ddot{\ })$	(double dot) second time derivative
(-1)	matrix inverse
$(\vec{\ })$	vector quantity

POSTSCRIPTS

(i)	blade radial station index
(n)	blade number index

1. INTRODUCTION

The presentation of the REXOR II software in this volume follows the multilevel hierarchial design of the program modules. A skeletal chart showing the major computational paths is presented in Figure 1.

The modules are discussed as follows:

- Section 2 — MAIN — Level 1 Executive
- Section 3 — FLY — Level 2 Executive
- Section 4 — ACCEL — Level 3 Executive
- Section 5 — GFORCE — Level 4 Executive
- Section 6 — MASSSP — Level 4 Executive

The design of REXOR II at the executive level is based on the classical technique input-process-output. The process can be characterized as a signal generator, with the output being a defined set of time-varying functions (signals) and the input block being what is necessary to define the physical model and control the process. The characterization of REXOR II as a signal generator provides a mechanism for extensive output processing. The signal set generated can be identified as input to a variety of output processors. Examples are:

- Printed time histories
- Graphic displays
- Harmonic analyses
- Frequency content analysis (FFT)

REXOR II provides run-time output abilities in printed form and CALCOMP plots of selected signals. Deferred signal processing is provided at a tradeoff cost, the cost being the peripheral storage required to save the signals for processing later. If the peripheral storage allocated for signal data is other than SCRATCH (i.e., disappears after the run), then an economic recovery mechanism is automatically provided if the central computer should fail during execution.

It should be noted that the concept of deferred processing does not have to be restricted to the type of signal analysis. The processing environment is also optional. A time-sharing computing environment with foreground access to the saved signal set presents an added dimension of effective use of information to a costly computer model such as REXOR II.

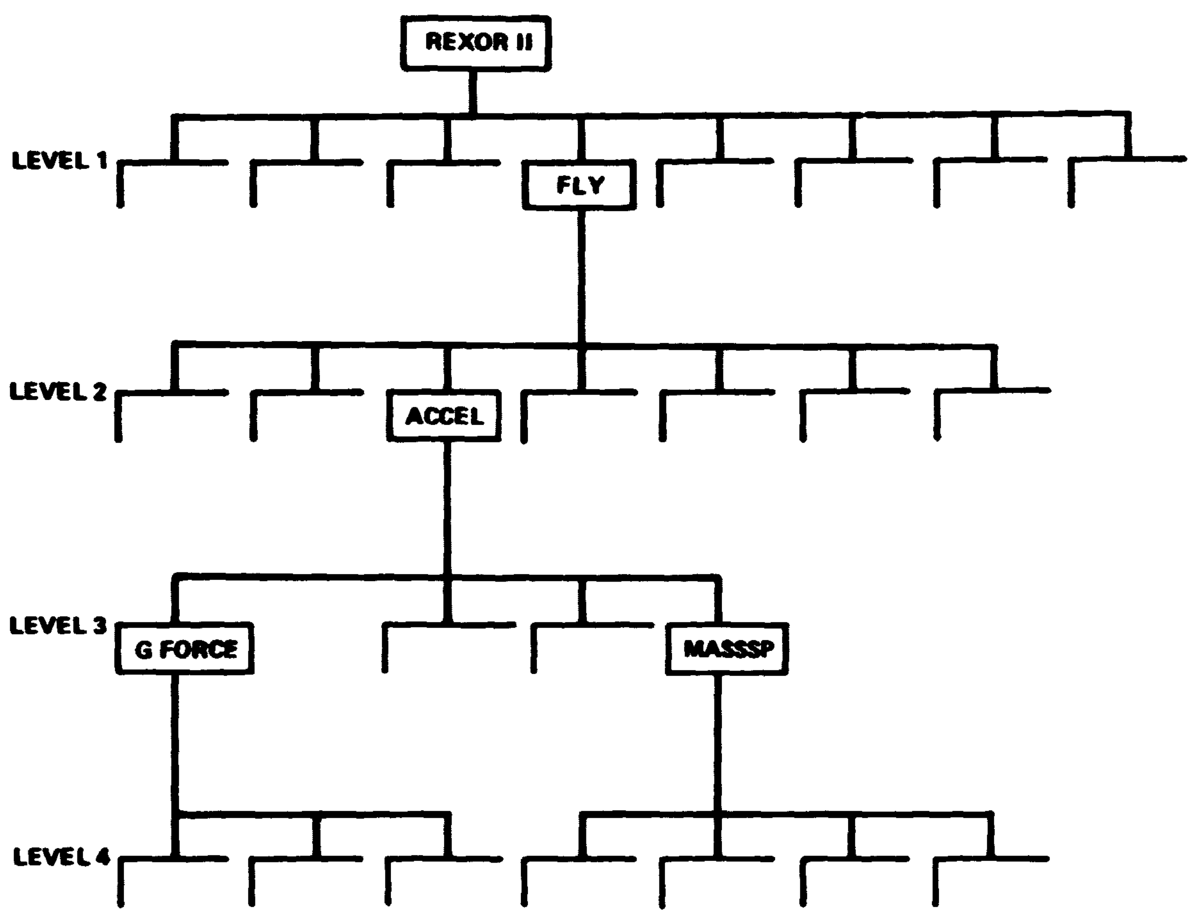


Figure 1. - Skeletal hierarchial chart.

2. MAIN — LEVEL 1 EXECUTIVE

The MAIN program is the REXOR II level 1 executive control program. The level 1 hierarchical chart presented in Figure 2 shows the separation and collection into input-process-output functions. The input routines include actual data-read routines, as well as certain data preparation and initialization routines. A flow diagram is presented in Figure 3. The loop structure depicted provides multiple case processing. The determination of a case to be processed is made in the input routine, READIN. The software development presented here is in relation to an IBM installation. Slight variations are required to effect the CDC overlay structure. These are discussed in section 2.7. The generalized coordinate system is defined for each case by a small set of input indicators. These indicators and required relationships for program control are described below.

Generalized Coordinate Selection

The elements of the generalized coord vector are determined by the user via input. The total system consists of subsystems which will be identified with some physical entity.

$$\underset{NTOTLx1}{\{q\}} = \left\{ \begin{array}{c} \{q_{BL}\} \\ \cdots \\ \{q_{SP}\} \\ \cdots \\ \{q_S\} \\ \cdots \\ \{q_{REF}\} \\ \cdots \\ \{q_R\} \end{array} \right\}$$

Each subsystem can be optionally included by proper identification at the subsystem level or in some cases, at the coordinate level, i.e., the selection of coordinates within a subsystem can be made. Required inputs and associated relationships are presented below.

2.1.1 Blade subsystem

- Input

NB	N_b	Number of main rotor blades $0 \le N_b \le 7$
IM1		=1 indicates presence of bending mode #1 =0 indicates absence
IM2		Bending mode #2 indicator
IM3		Bending mode #3 indicator

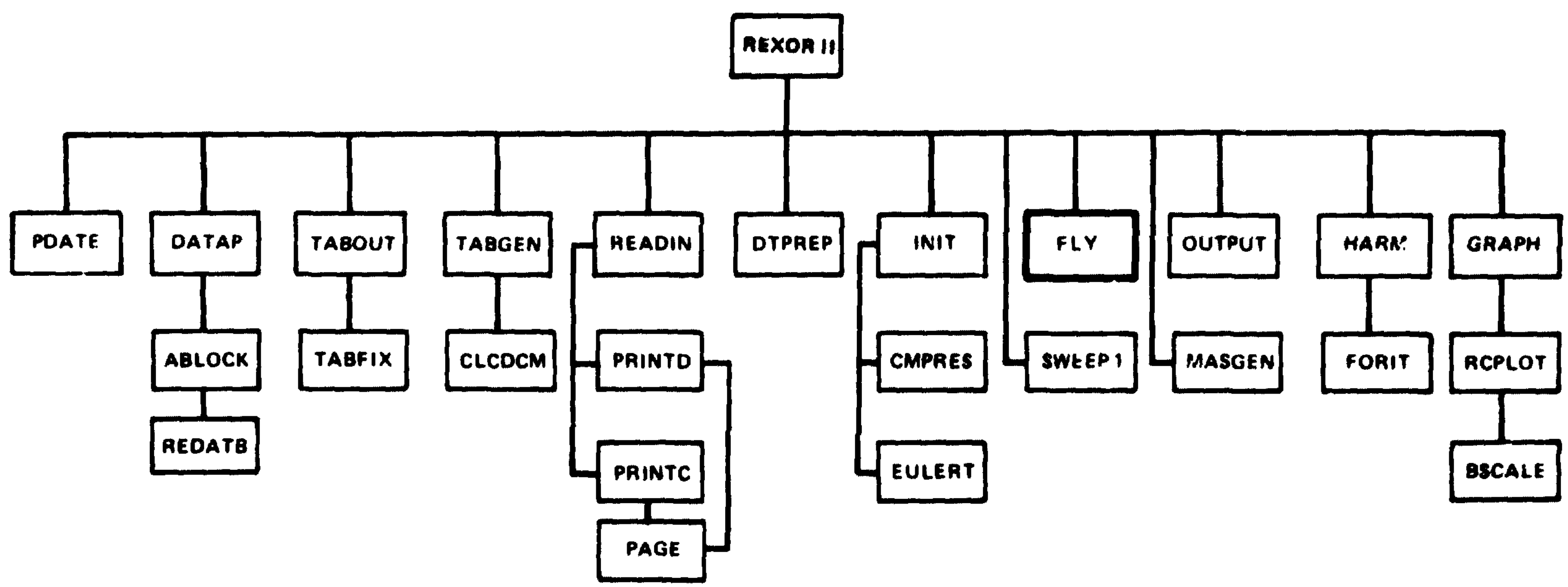


Figure 2. - REXOR II - level 1 hierarchical chart.

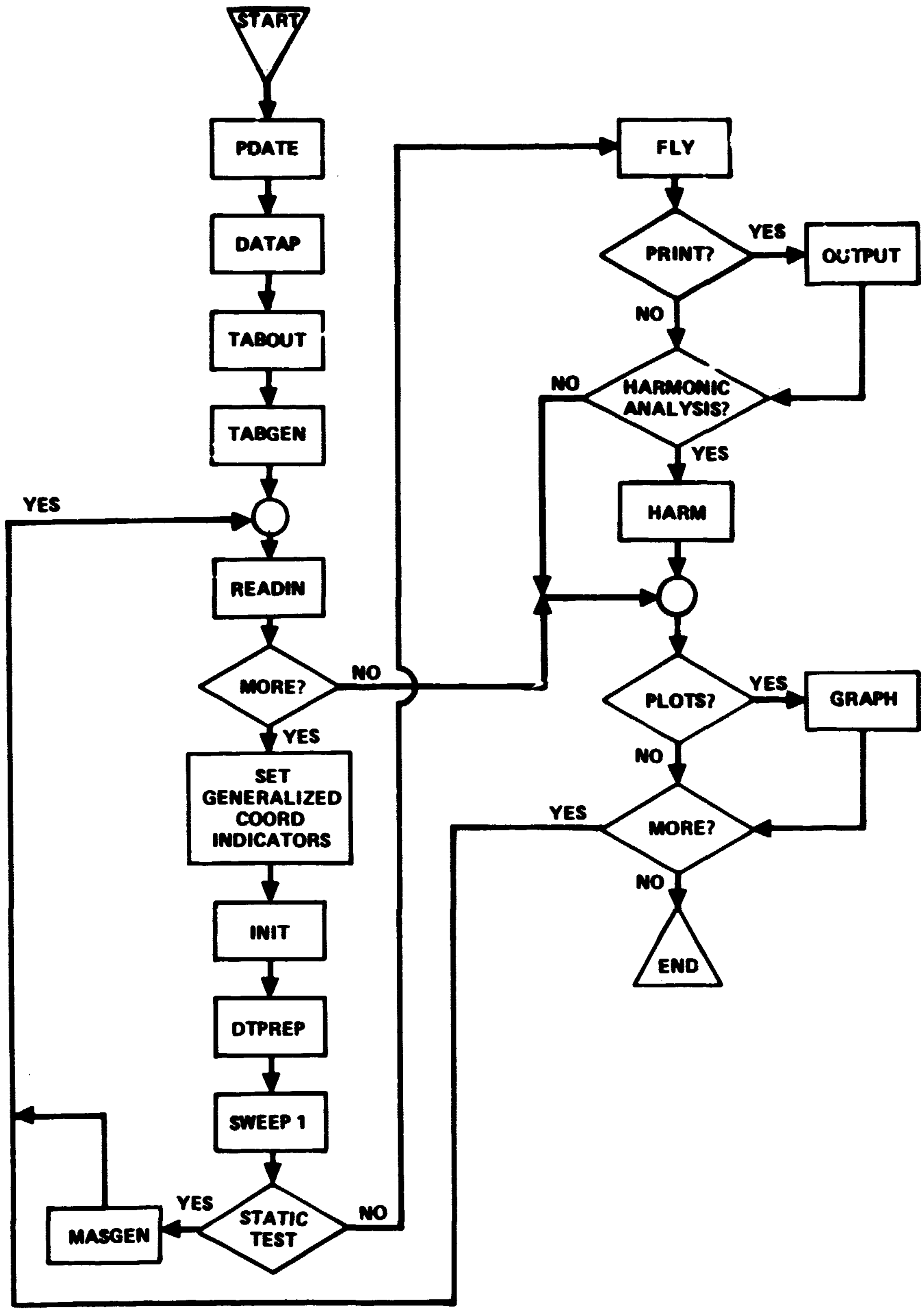


Figure 3. - REXOR II MAIN program.

NBP	$N_{B_{ph}}$	Pitch horn bending indicator (elastic feathering)
NPT	N_{ϕ_T}	Dynamic torsion indicator

- **Relationships**

Total number of bending modes per blade, N_{m_b}

$$NMB = IM1 + IM2 + IM3; \; 0 \le NMB \le 3$$

Total number of modes per blade, N_m.

$$NM = NMB + NBP + NPT$$

$$0 \le NBP + NPT \le 1 \qquad \text{(Mutually Exclusive)}$$

$$0 \le NM \le 4$$

Total number of coord in the blade subsystem, N_{BS}

$$NBS = NB \cdot NM \qquad 0 \le N_{BS} \le 28$$

2.1.2 Swashplate subsystem

- **Input**

NSP	N_{SP}	Number of swashplate coord.

- **Relationships**

Specific values identify specific subsets. Trivially,

$$N_{SP} = 0 \Rightarrow \text{No swashplate generalized coord.}$$

Continuing,

$$N_{SP} = 1 \Rightarrow \{q_{SP}\} = Z_{SP}$$

$$N_{SP} = 2 \Rightarrow \{q_{SP}\} = \begin{Bmatrix} \phi \\ \theta \end{Bmatrix}_{SP}$$

$$N_{SP} = 3 \Rightarrow \{q_{SP}\} = \begin{Bmatrix} \phi \\ \theta \\ Z \end{Bmatrix}_{SP}$$

$$0 \le N_{SP} \le 3$$

2.1.3 Reference subsystem

- Input

NREF N_{REF} Number of REF coordinates

$$N_{REF} = 0, 6$$

$$\{q_{REF}\} = \{X, Y, Z, \phi, \theta, \psi\}_{REF}$$

2.1.4 Shaft subsystem

All 6 shaft coord, $\{X, Y, Z, \phi, \theta, \psi\}_S$ are independently selectable.

- Input

IS1	X_S indicator
IS2	Y_S
IS3	Z_S
IS4	ϕ_S
IS5	θ_S
IS6	ψ_S

A value of 0 means exclude

A value of 1 means include

- Relationships

Total number of shaft coord, N_S

$$NS = IS1 + IS2 + IS3 + IS4 + IS5 + IS6$$

$$0 \leq N_S \leq 6$$

2.1.5 Rotor subsystem $\{\psi\}_R$

- Input

NR N_R Number of rotor coord $0 \leq N_R \leq 1$

Total number of generalized coord, N_{TOTL}

$$NTOTL = NBS + NSP + NS + NREF + NR$$

$$0 \le N_{TOTL} \le 44$$

Subsystem vector locations within generalized vector are:

LBS	L_{BS}	=	1
LSP	L_{SP}	=	$L_{BS} + N_{BS}$
LS	L_S	=	$L_{SP} + N_{SP}$
LREF	L_{REF}	=	$L_S + N_S$
LR	L_R	=	$L_{REF} + N_{REF}$

A summary of required inputs and the associated relative address (RA) numbers is given below

Input	RA
NB	1
IM1	2
IM2	3
IM3	4
NBP	5
NPT	6
NSP	7
NREF	8
IS1	9
IS2	10
IS3	11
IS4	12
IS5	13
IS6	14
NR	15

2.2 Input and Initialization

2.2.1 DATAP. - Subprogram DATAP is a data-preparation routine. It performs three functions.

- Lists the complete data deck
- Prepares a data stream for the input algorithm of subprogram READIN
- Senses and processes a special data block.

The first item is self-explanatory. The input data deck is listed as input except for the special data block to be discussed. The second function, to prepare a data stream, requires further comment. The input routine READIN reprocesses an input record more than once. Mechanisms to perform that task are traditionally installation and software dependent. DATAP builds a data stream with multiple copies of a given record where needed.

A normal data deck may be preceded by a special data block containing main rotor aerodynamic data. This data is handled by its own set of routines which are discussed in Section 2.3.

A flow diagram of DATAP is presented in Figure 4. It should be noted that the input stream is the standard FORTRAN input unit 5, and the prepared data stream is FORTRAN unit 12.

The subprogram argument, NUMSET, is an informational parameter which passes information from the aerodynamic data input routine ABLOCK back to MAIN for use in other aero routines. This is discussed further in Section 2.3.

2.2.2 READIN. - Subprogram READIN controls the data deck processing as described in Section 3.1. of Volume III. This processing includes data unit recognition; case data construction; the printing of master data and case data; the punching of an edited master data unit; setting signal, plot, and print default values; and finally the processing of signal label and title cards.

Arguments List

NDATE	8-character date, input
CASEID	8-character case identification, output
SAVED	Logical variable for control .F. - master data not saved .T. - master data saved

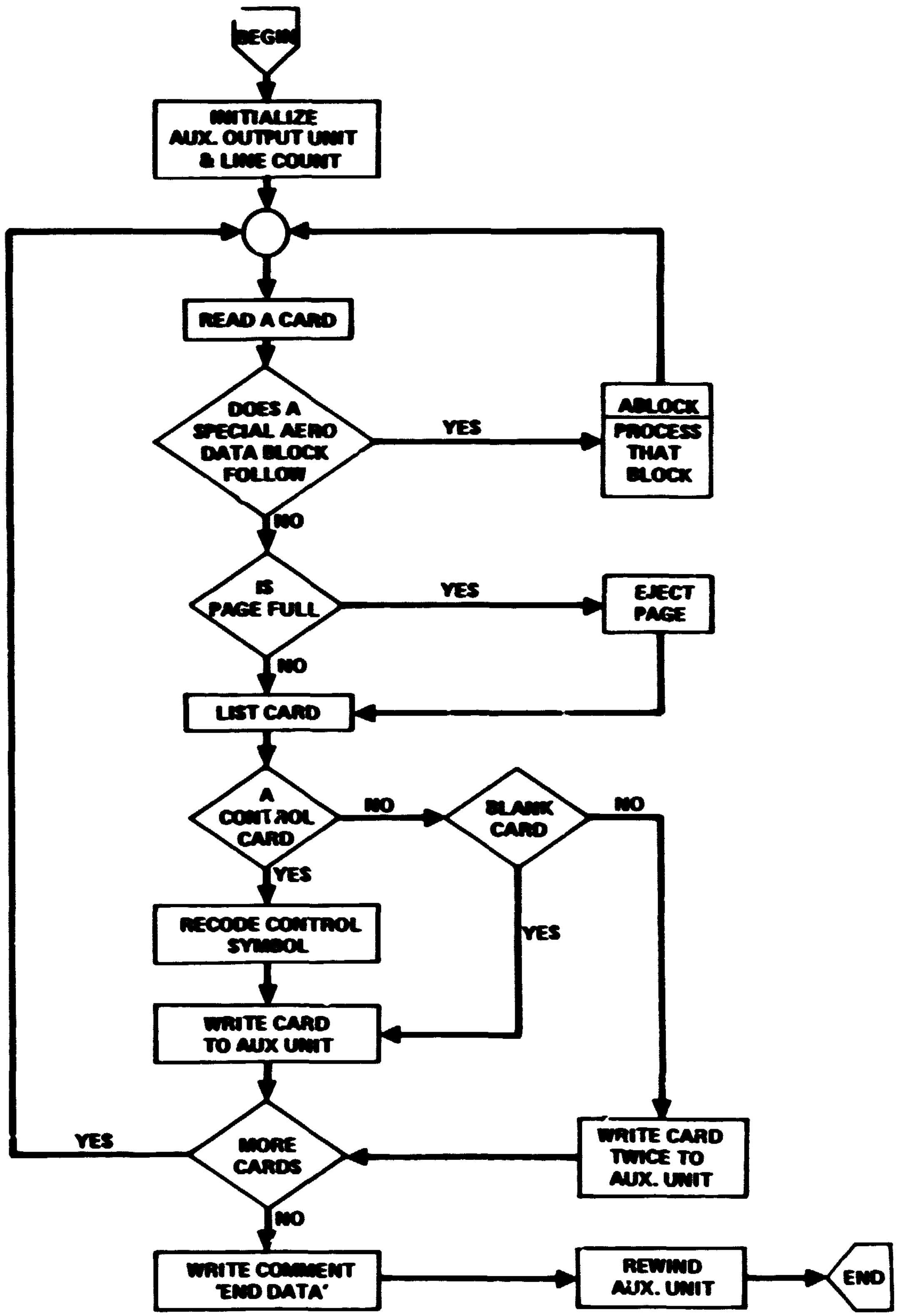


Figure 4. - DATAP flow diagram.

NOMORE   Logical variable for control

.F. - more data to process
.T. - no more data to process

CASE   Logical variable for control

.F. - not a case data unit
.T. - a case data unit

A flow diagram is presented in Figure 5.

2.2.3 DTPREP. - Subprogram DTPREP is a data preparation routine. Main rotor blade bending mode data is assembled for the modes desired. Mode data for up to three modes may be defined in the input addresses. The modes desired can be any subset of the defined modes. Also, modal data is defined in a left handed Z down coordinate system. This routine rotates the data into a Z up system.

Data associated with the shaft subsystem may require shuffling also, if a coordinate subsystem is chosen. Shaft spring and damper input matrices are row and column shuffled accordingly.

2.2.4 INIT. - Subprogram INIT performs a collection of loosely bound functions associated with the initialization of a case. These functions include the one time calculations of subsystem inertias and mass matrices. The units of some inputs are converted. Certain transformation matrices which will remain constant throughout the case are calculated. Finally, the generalized coordinates and the auxiliary coordinates are initialized. Certain equations of interest are presented below.

Main rotor blade mass:

$$m_{BL} = \int^{S} m(S)ds$$

Blade inertia:

$$I_{XX_{BL}} = \int^{S} I_{XX_{BLE}} (S)\, ds$$

$$I_{ZZ_{BL}} = \int^{S} m(S)\, X^2_{BLE} (S)\, ds$$

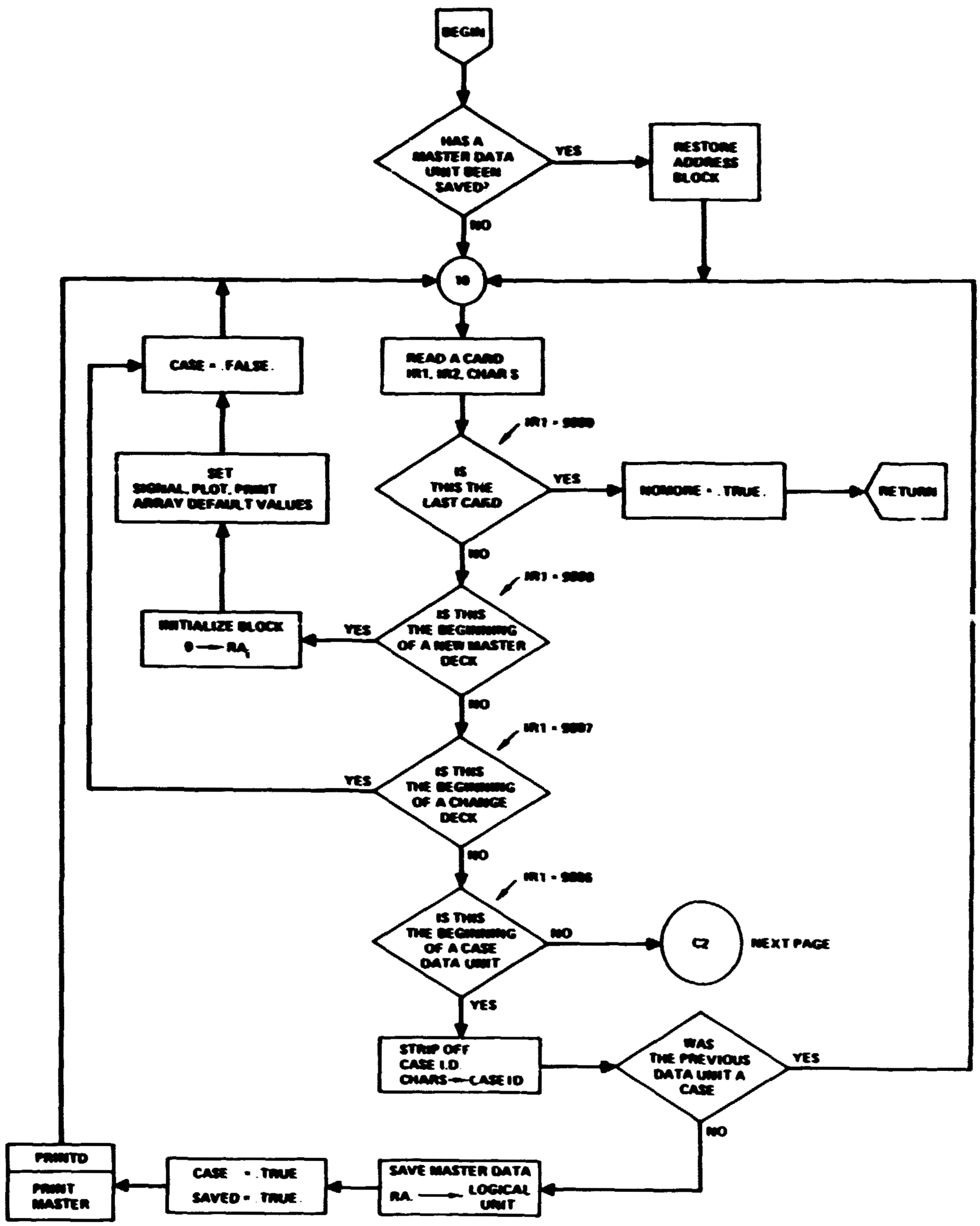


Figure 5. - READIN flow diagram.

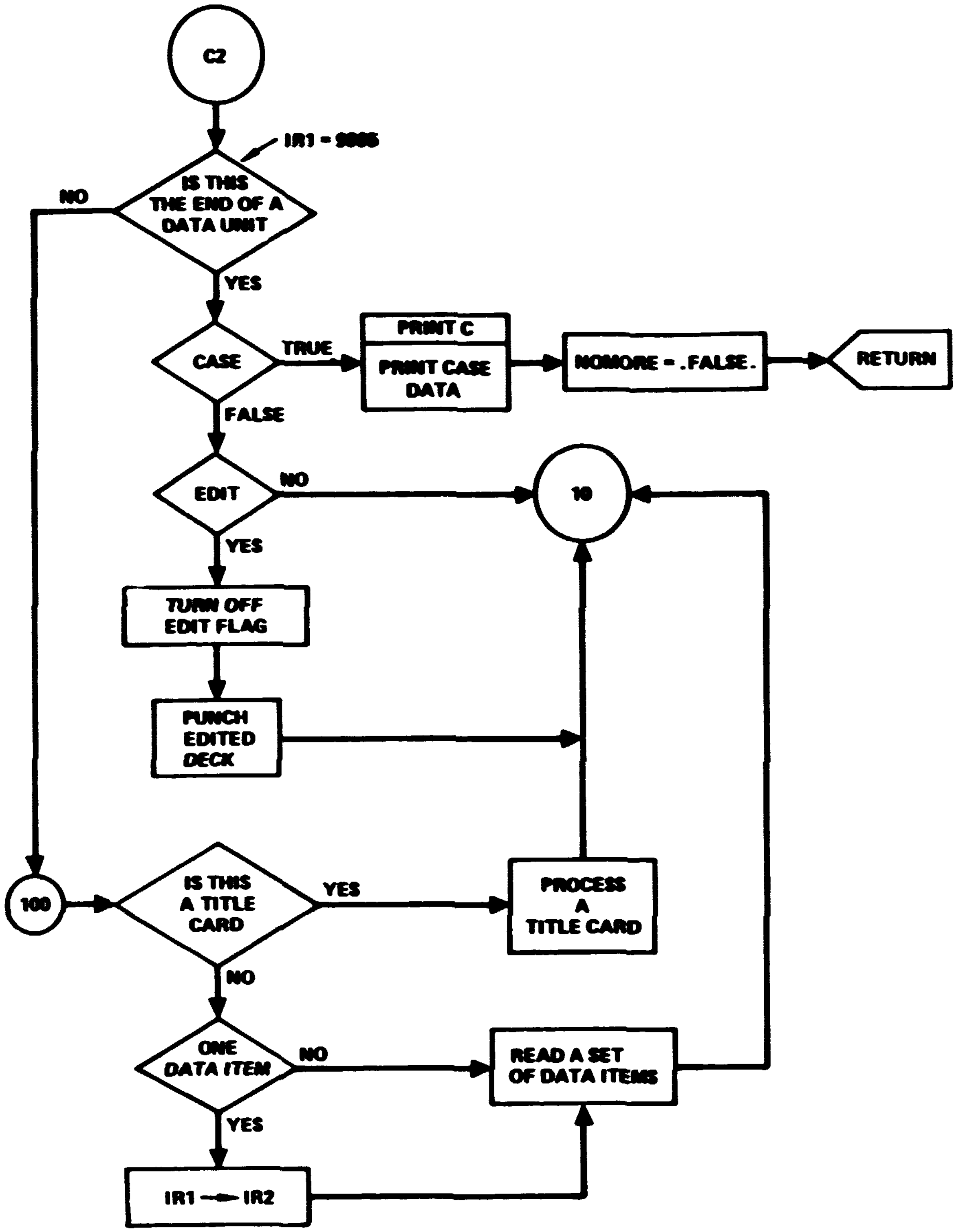


Figure 5. - Concluded.

Total inertia:

$$I_{ZZ_{BL}} = I_{ZZ_{BL}} + I_{XX_{BL}}$$

Mass of the transmission mount relative to the hub

$$\left[M_{O_T}\right]_H = \left[M_{O_T}\right] + \left[M_{O_H}\right]$$

where $\left[M_{O_H}\right]$ is given by equation 351, Volume I and $\left[M_{O_T}\right]$ is given by equation 353, Volume I. Also, the fuselage mass $\left[M_O\right]_F$, given by equation 362 of Volume I, is calculated. Transforms calculated in INIT include the transform from rotor to blade. For each blade, n,

$$\left[T_{R \rightarrow BL}\right]_n = \begin{bmatrix} \cos \psi_{BL} & \sin \psi_{BL} & 0 \\ -\sin \psi_{BL} & \cos \psi_{BL} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where

$$\psi_{BL_n} = -2\pi (n-1)/N_b.$$

The fuselage-to-transmission mount base is also computed

$$\left[T_{F-MB}\right] = T_E (\phi_{O_S}, \theta_{O_S}, \psi_{O_S}).$$

In the notation above, T_E, depicts the standard Euler transformation.

The remaining initialization details will not be repeated here. The reader should refer directly to the code.

2.3 Aerodynamic Data Input and Preparation Modules

The main rotor aerodynamics of REXOR II are handled via a table lookup method for the functions:

$$C_L = C_L (M, \alpha)$$

$$C_D = C_D (M, \alpha)$$

$$C_M = C_D (M, \alpha).$$

A representative set of data is built into the program. However, the user may provide his own tables to the program. Further, two sets of tables may be provided with a provision to shift from one set to the other as a function of blade station. Another requirement of this simulation is the compatibility of input format with those of the C-81 Rotor Program. Compatibility has been achieved by installing the necessary input and interpolation routines borrowed from the C-81 program. C-81 routines installed intact or in a slightly modified form include:

ABLOCK
REDATB
TABOUT
TABFIX
CLCDCM.

The function of these routines and others will be briefly described below.

2.3.1 ABLOCK. - The user can supply to the program on a once-per-run basis, either a replacement set 2 of aero data or an added set 1 to the existing set or both. The details of this mechanism are described in Volume III. If an aerodynamic data set is sensed by the routine DATAP, then ABLOCK is called. This routine controls the input of the aerodynamic block sensed by causing the data to be read into the proper tables via subprogram REDATB.

Arguments List

IN — FORTRAN input unit. Designated as 5.

SETID — Set I.D. as sensed and supplied by DATAP. It is an alphanumeric string of the form

' ^1 ^^ ' or ' ^2 ^^ '.

NUMSET — An identifier which will indicate which sets have been input.

=1 set one only
=2 set two only
=3 if both sets

2.3.2 REDATB. - Subprogram REDATB simply reads in an aerodynamic table. All tables, i.e., C_L, C_D, and C_M are of the same form. Subprogram ABLOCK, which calls REDATB, controls which table is read. The table formats are detailed in Volume III.

Arguments List

C	Table name
NX, NZ	Table size parameters
IN	FORTRAN input unit

2.3.3 TABOUT. - Subprogram TABOUT performs two functions. First, if tables have been input, TABOUT will list those tables on the standard output device. Second, all tables to be used by the program are further prepared for the interpolation routine. This preparation is initiated for the required tables by calls to routine TABFIX.

2.3.4 TABFIX. - Subprogram TABFIX scans the table specified and prepares associated information tables for later use by the interpolating routine, CLCDCM. The intent is to provide information which will speed up the argument bracketing process.

2.3.5 TABGEN. - Subprogram TABGEN uses the interpolation routine, CLCDCM to prepare two functions for use by the dynamic STALL routine. The two functions are

$$C_L = C_L(\alpha, M)\Big|_{\alpha=0}$$

$$\frac{\partial C_L}{\partial \alpha} = f(\alpha, M)\Big|_{\alpha=0}$$

The functions will be in table form suitable for fast interpolation by routine XTRP1.

2.4 Output Modules

2.4.1 PRINTD, PRINTC. - This multiple entry routine produces a formatted listing of the program inputs. The entry PRINTD produces a listing of master data and PRINTC produces a listing of case data in the form of an exception report.

2.4.2 OUTPUT. - This routine prints block time histories. The block size is up to six rows of eight variables each. The quantities to be printed are selected from the signal set with the print pointers, PP(48). The corresponding titles are gathered from the PLAB array, and the first eight characters of each signal title are used. The first and last points are always printed. An input, PFREQ, controls the print frequency. The user control of the block time history output generated by this routine is discussed in Volume III.

Arguments List

SIGUNI	The number of the FORTRAN unit which holds the signal data. The value is set in the calling routine, MAIN.
NPTS	The number of points saved in the signal set.
CASEID	8-character case identification.

2.4.3 HARM. - This subprogram will provide harmonic analysis output for up to 50 signals. The output will consist of the Fourier coefficients

$$a_o = \frac{1}{2\pi} \int_o^{2\pi} f(x)\, dx$$

$$a_n = \frac{1}{\pi} \int_o^{2\pi} f(x) \cos nx\, dx$$

$$b_n = \frac{1}{\pi} \int_o^{2\pi} f(x) \sin nx\, dx$$

for $n = 1, ---, 8$ (8 harmonics).

The data will also be presented as phase-amplitude information where:

$$A_n = \sqrt{a_n^2 + t_n^2}$$

$$\theta_n = \mathrm{Tan}^{-1}\left(\frac{b_n}{a_n}\right) \cdot \left(\frac{360}{2\pi}\right) \Big/ n$$

If $b_n < 0$

$$\theta_n = \theta_n + 360/n$$

Subroutine limitations are arbitrary and include:

- The period of the functions analyzed is 2π
- Eight and only eight harmonics
- A maximum of 50 functions can be analyzed. The functions must be defined in the signal set.

Arguments List

NPTS	Number of points in signal set
SIGUNI	The number of the FORTRAN unit which holds the signal data.
CASEID	8-character case identification for output
TCYCLE	Number of cycles consumed in TRIM

2.4.4 FORIT. - Subprogram FORIT is derived from the IBM Scientific Subroutine Package. It performs a Fourier analysis of a periodically tabulated function by computing the coefficients of the desired number of terms in the Fourier series series (Reference 1):

$$F(x) = a_o + \sum_{k=1}^{M} a_k \cos kx + b_k \sin kx.$$

Arguments List

FNT	Vector of tabulated function of length N
N	The number of samples over the interval $(0, 2\pi)$
M	Maximum order of harmonics to be fitted
A	Resultant vector of Fourier cosine coefficients of length M+1
B	Resultant vector of Fourier sine coefficients of length M+1
IER	Resultant error code when

IER = 0	no error
IER = 1	$N < 2M + 1$
IER = 2	$M < 0$

2.4.5 GRAPH. - Subprogram GRAPH controls the generation of CALCOMP plots. This includes the opening and closing of plots as required by CALCOMP software. Further, a logical decision is made as to whether there are enough points to plot. Plotting is not accomplished unless sufficient points are available. The plotting of signals is performed by a subordinate routine, RCPLOT.

Arguments List

NPTS	Number of points in signal set
SIGUNI	Logical unit number of signal set
MORE	Sign-off signal. A logical variable which if false then sign-off.
FENTRY	First entry signal. If .TRUE. then first entry, so open plots.
CASEID	8-character case identification.

A diagram is presented in Figure 6.

2.4.6 RCPLOT. - Subprogram RCPLOT is the CALCOMP plot generator. Up to 100 signals from the signal set can be selected for plotting. Only a few words concerning plot specifications are presented here. Details concerning user options and control are set forth in Volume III.

Although the number of points saved in the signal set is open-ended, dimension statements within this routine have arbitrarily set the maximum number of points per signal at 2048. This number can be extended.

Plotting proceeds in a frame-oriented manner. Signals are collected four at a time for plotting on a frame. A minimum frame size corresponding to a standard 8-1/2" x 11" page is produced so that CALCOMP continuous paper rolls can be cut page size. If time history signals are longer than the minimum, then the abscissa is continued the necessary amount. A sample frame can be found in Volume III.

Arguments List

NPTS	Number of points in signal set
TSCLE	Abscissa scale factor, sec/in.
NFREQ	Plotting frequency, i.e., =1, plot every point =2, plot every other point =3, plot every third point, etc.

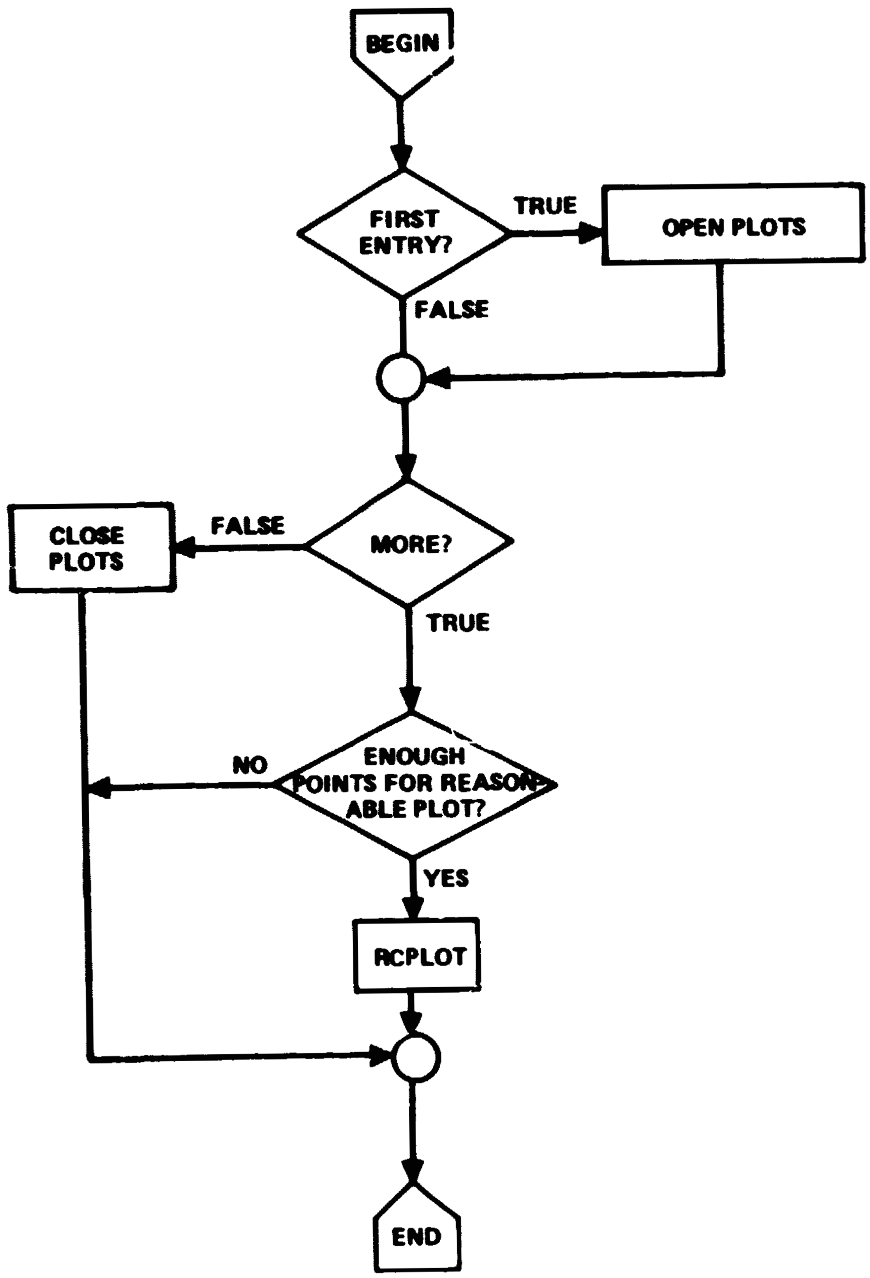


Figure 6. - GRAPH flow diagram.

SIGUNI	Logical unit containing signal set
CASEID	8-character case identification.

2.4.7 BSCALE. - Given a signal to be plotted, BSCALE determines a plot scale factor such that the signal is restricted to a 2-inch channel.

Arguments List

X	Array to be scaled
N	Number of points in array
DY	Scale factor of the form

$$DY = 1 \times 10^{na}$$

$$DY = 2 \times 10^{na}$$

$$DY = 4 \times 10^{na}$$

NA	Exponent above.

2.4.8 PDATE. - PDATE is an installation-dependent routine designed to provide the date in character form. The subroutine argument is an eight character variable of the form:

NDATE = MM-DD-YY

2.4.9 PAGE. - Subprogram PAGE performs a page eject on unit 6 when called. A page header is produced with the case identification, date, and REXOR II input titles 1 and 2. On exit, a line counter is set to 5.

Arguments List

ICT	Line count on exit
CASEID	8-character case identification
NDATE	8-character date
TITLES	REXOR II title array with dimension (15, 4)

2.5 MASGEN

Subprogram MASGEN is an alternate process control program. It is optionally invoked from MAIN. Its function is to generate a generalized mass matrix via perturbation techniques where

$$M_{k_j} = - \frac{\Delta F_k (\Delta \ddot{q}_j) - \Delta F_k}{\Delta \ddot{q}_j} .$$

A mass matrix generated by this method provides an excellent diagnostic tool for studying the correctness of the computed mass matrix when new equations are developed. MASGEN is not necessary to the operation, but is provided for possible future use. A flow diagram is presented in Figure 7.

2.6 Block Data Subprograms

A number of block data subprograms are present in the program. A list of the subprograms and the LABELED COMMON'S within each is presented.

NAME	COMMON LABEL	DESCRIPTION
BLOCKA	/TAB / /CLACLO/	Main rotor blade aerodynamics
BLOCKB	/CONSTS/	Constants
BLOCKC	/NAMES /	Input parameter names
BLOCKD	/LABELS/	Signal labe[illegible] default set
BLOCKE	/RAINIT/	Default relative addresses for the signal set, print, and plot tables

The labeled COMMON /CONSTS/ is expanded below.

NAME	VALUE
G	32.174
PI	3.141593
TWOPI	6.283185
PI02	1.570796
DTOR	57.29578
RTOD	0.01745329
RHOZ	0.00237689
VSZ	1116.45

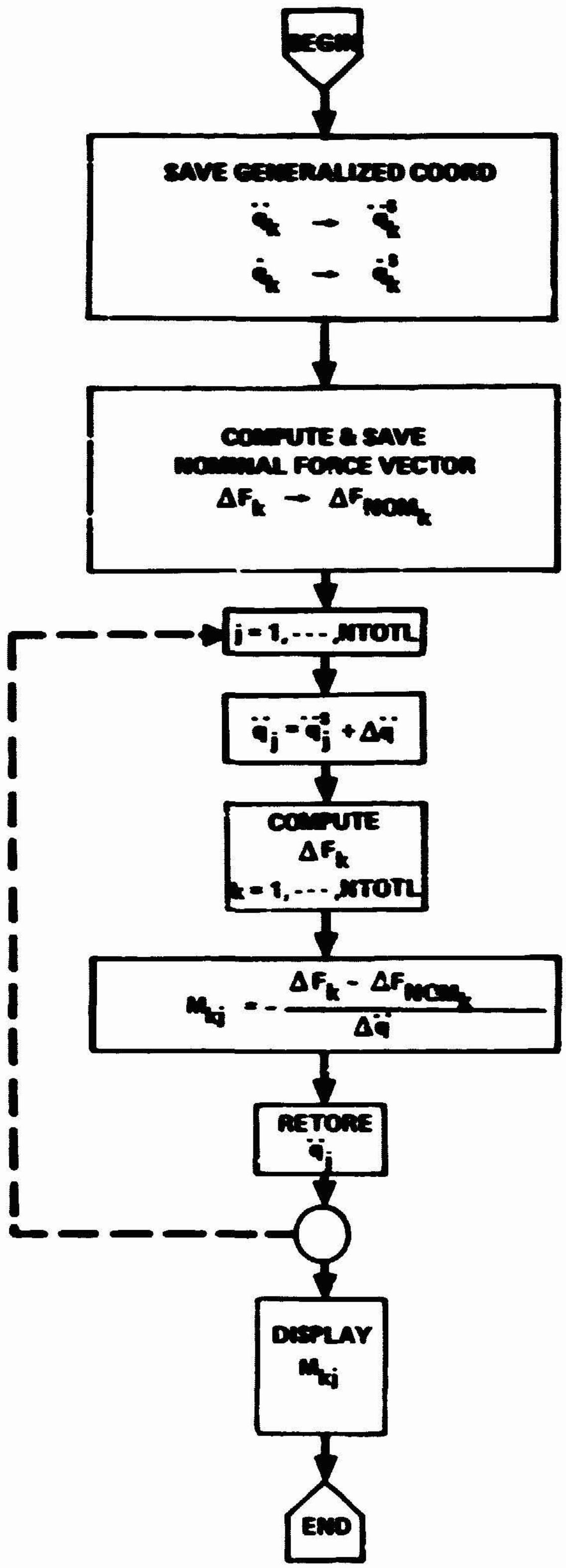


Figure 7. - MASGEN flow diagram.

2.7 CDC Overlay Implementation

The implementation of an overlay structure in a CDC environment requires the restructuring of the software into a series of programs. The execution flow must be explicitly defined within the source statements of the various programs. The changes required to convert IBM software as described here to CDC software is as follows. The reader can refer to Figure 3, REXOR II MAIN program. Subprogram READIN is elevated to program status and placed in primary overlay 1. It is invoked by a CALL OVERLAY (PFILE, 1, 0, RECALL) in place of CALL READIN.

The CALL SWEEP1, CALL MASGEN, CALL FLY sequence of the MAIN program are removed, and given program status as PROGRAM PROCESS. They are placed in primary overlay 2. Subroutine HARM is also given program status, and placed in primary overlay 3. Finally, subroutine RCPLOT is given program status, and placed in primary overlay 4. The respective calls have been altered. Data communications which existed through argument lists are handled through labelled COMMONS.

3. FLY — LEVEL 2 EXECUTIVE

The level 2 hierarchical chart for FLY is presented in Figure 8. FLY is a secondary executive routine which controls the calculation of the generalized coordinates, thus, generating all the desired time signals. FLY is primarily a loop structure built around the integrators as shown in Figure 9. The elements of the loop are described below.

3.1 FLY

A flow diagram of FLY is presented in Figure 10. A number of program counters and run controls are defined below for further clarification.

The program variable TRIMED is a logical variable which is initialized as false and under certain conditions is set to the value, true, within the TRIM subprogram. The variable TRIMED indicates if the vehicle trimmed. The variable MCOMP is the generalized mass matrix computation flag. It is initialized as true. The variable DISPMX flag is the display matrix flag and is initialized as false. Other variables are:

NPTS	Number of points saved in the signal set
PTIME	Trajectory time measured from end of TRIM. Its primary use is as the argument to the pilot control tables
IC ICNTAX	} Integration step counters used for logic control within the integration routines.
AZIMTH	Blade azimuth step counter within one rotor revolution
CYCLE	A count of completed rotor revolutions. Blade 1 is the reference
CTEST	Trim convergence test control counter
TCYCLE	A count of the number of cycles required to trim
LTICK	Main rotor pip counter. Counts blades crossing the fuselage
NCC	The number of second order differential equations in the generalized coordinate set to be integrated. This count is altered in TRIM
NE	The number of auxillary differential equations to be integrated.

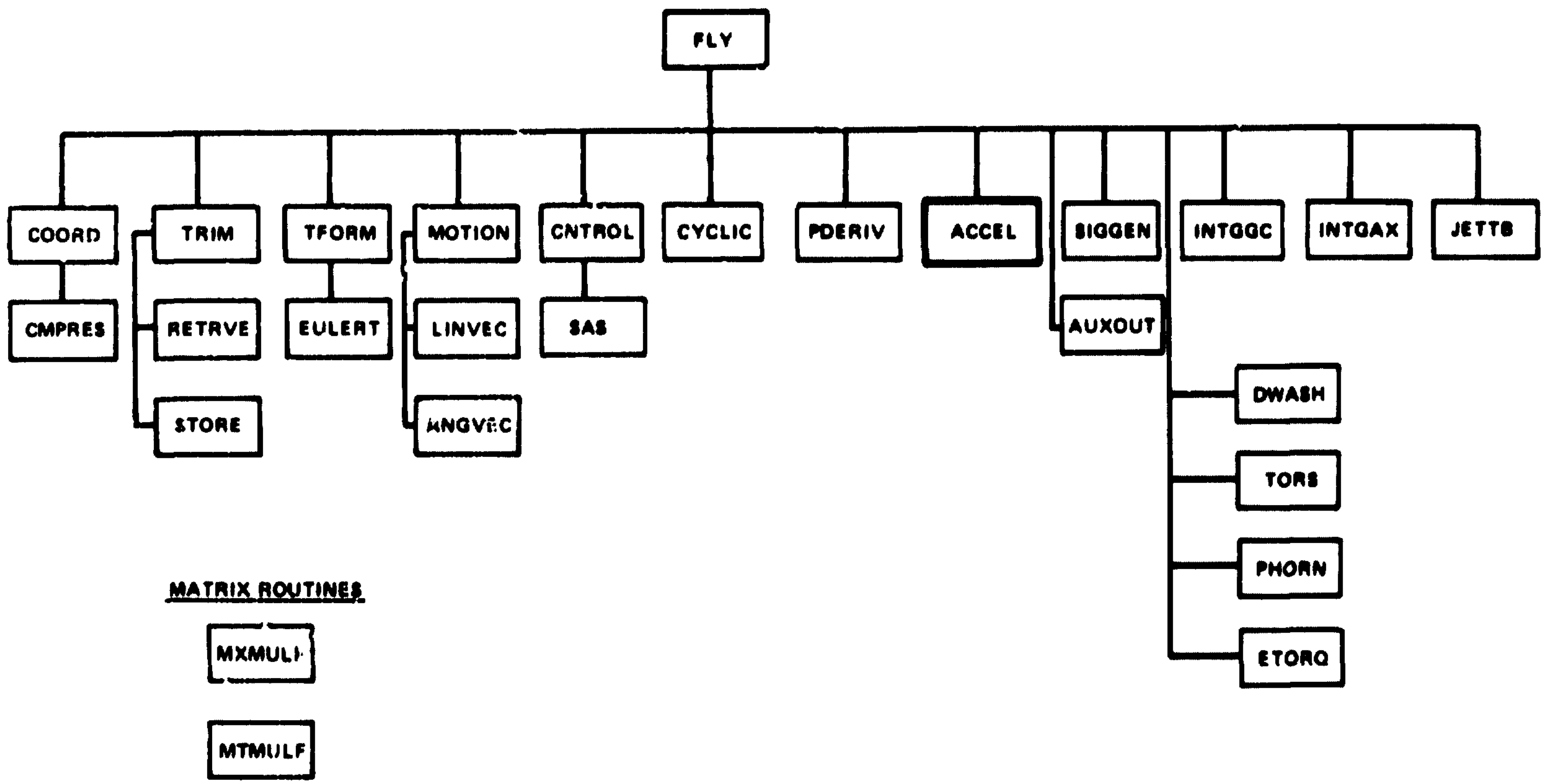


Figure 8. - FLY - Level 2 hierarchical chart.

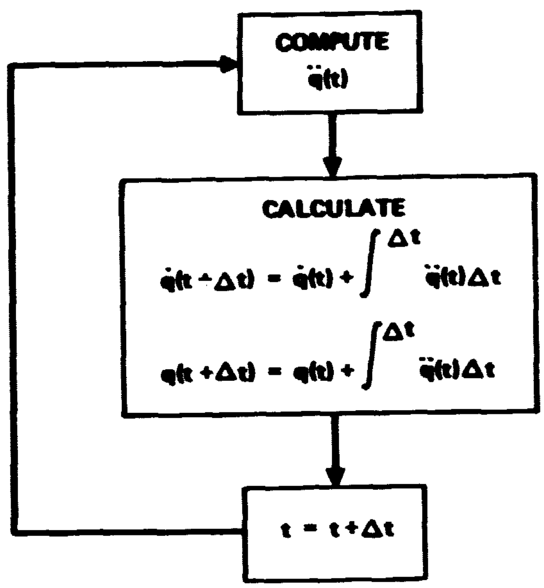


Figure 9. - FLY calculation loop.

One should note that if in the future new auxillary differential equations are defined, the variable, NE, defined above must be changed to the correct count. It should also be pointed out for future consideration that a fundamental assumption upon entry into FLY is that the generalized coordinate set has been defined and initial values for the vectors GC(I), GCD(I), and GCDD(I) have been made.

The arguments in the calling sequence to FLY are primarily output oriented.

CASEID	8 character case identification which originates in READIN
IER	Error indicator. Initialized in MAIN. Value set within ACCEL.
SIGUNI	FORTRAN logical unit for storage of output signals.

The remaining arguments NPTS and TCYCLE are defined above.

3.2 COORD

Given, the generalized coordinate vectors, and the necessary indicators, this subprogram defines the program problem coordinates. The generalized coordinate vector is

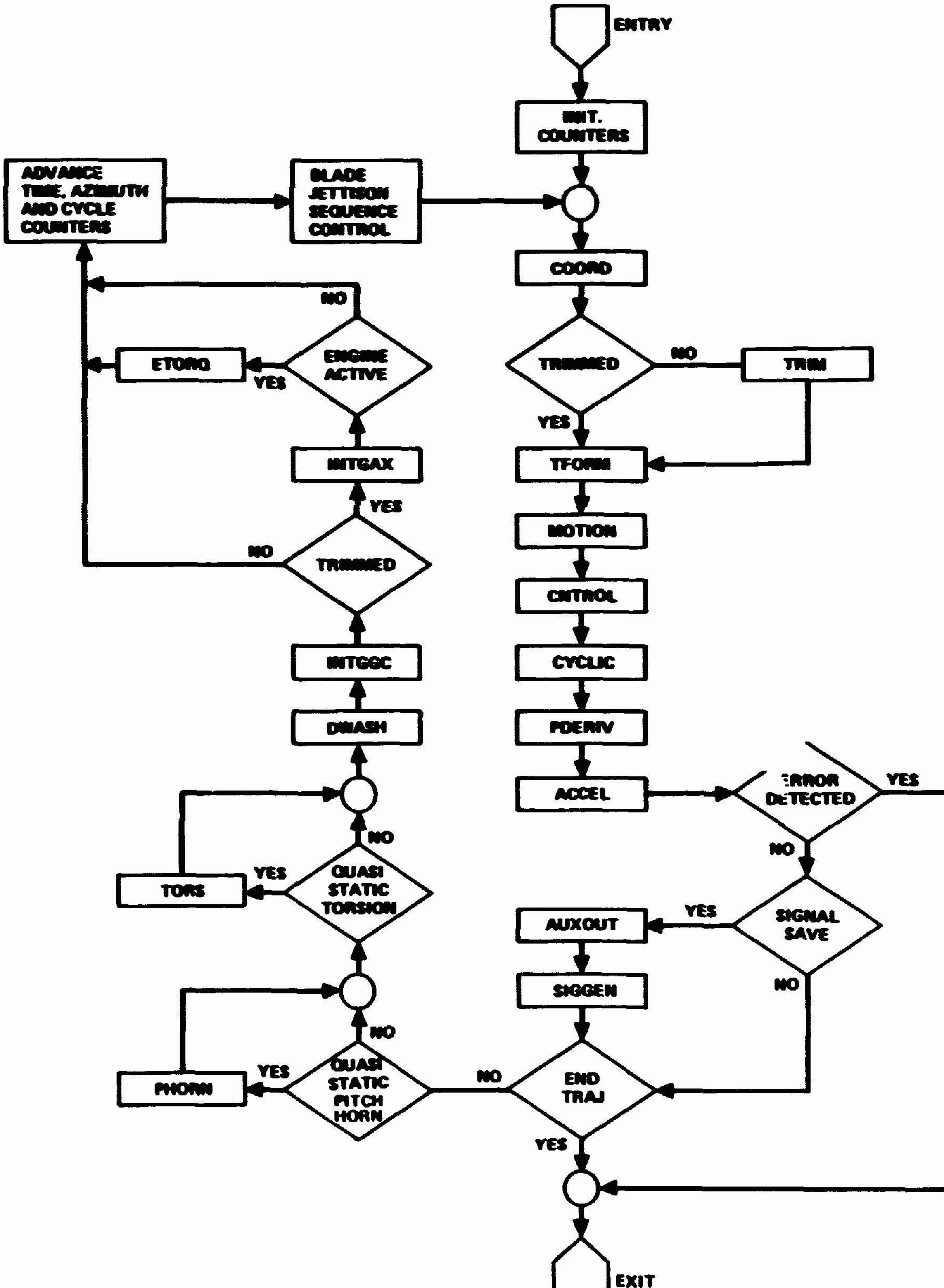


Figure 10. - FLY flow diagram.

$$\left\{q\right\} = \left\{\begin{matrix} q_{BS} \\ \cdots\cdots \\ q_{SP} \\ \cdots\cdots \\ q_S \\ \cdots\cdots \\ q_{REF} \\ \cdots\cdots \\ q_R \end{matrix}\right\}$$

similarly for

$$\left\{\dot{q}\right\}$$

and

$$\left\{\ddot{q}\right\}$$

The following definitions and coord options are available.

<u>Blade System</u>

$$\left\{\begin{matrix} \left\{q_{N_m}\right\}_1 \\ \left\{q_{N_m}\right\}_2 \\ \cdot \\ \cdot \\ \cdot \\ \left\{q_{N_m}\right\}_{N_b} \end{matrix}\right\}_{N_{BS}} = \left\{q\right\}_{N_{BS}}$$

For each blade, $n = 1, \ldots, N_b$

<u>Bending Modes</u>

$$\left\{A_m\right\}_n \leftarrow \left\{q_m\right\}_n \qquad m = 1, \ldots, N_{m_b}$$

Likewise for

$$\left\{\dot{A}_m\right\}_n$$

and

$$\left\{\ddot{A}_m\right\}_n$$

Pitch Horn Bending (Elastic Feathering)

If NBP = 1, then

$$P_n \leftarrow \left\{q_{N_{m_{b+1}}}\right\}_n$$

Likewise for

$$\dot{P}_n \text{ and } \ddot{P}_n$$

Also, since pitch horn bending and dynamic torsion are mutually exclusive,

$$\Phi_{T_b} = \dot{\Phi}_{T_n} = \ddot{\Phi}_{T_n} = 0$$

Dynamic Torsion

If NPT = 1, then

$$\left\{\Phi_T\right\}_n \longleftarrow \left\{q_{N_{mb+1}}\right\}_n$$

Likewise for

$$\left\{\dot{\Phi}_T\right\}_n$$

and

$$\left\{\ddot{\Phi}_T\right\}_n$$

Also

$$P_n = \dot{P}_n = \ddot{P}_n = 0$$

NOTE: Quasi-static pitch horn bending and torsion are available as alternative models.

NOTE: No provision has been made at this time for the special case where NBS = 0, i.e., no blade system.

Swashplate Options $N_{SP} = 0, 1, 2, 3$

$N_{SP} = 3$ All active

$$\phi_{SP} = q_{1\ SP}$$

$$\theta_{SP} = q_{2\ SP}$$

$$Z_{SP} = q_{3\ SP}$$

Similarly for

$\dot{\phi}_{SP}$ and $\ddot{\phi}_{SP}$

$\dot{\theta}_{SP}$ and $\ddot{\theta}_{SP}$

$\dot{Z}_{SP}$ and $\ddot{Z}_{SP}$

$N_{SP} = 2$

ϕ_{SP} and θ_{SP} as defined in Option ($N_{SP} = 3$)

$$Z_{SP} = \dot{Z}_{SP} = \ddot{Z}_{SP} = 0$$

$N_{SP} = 1$

$$Z_{SP} = q_{1\ SP},\ \dot{Z}_{SP} = \dot{q}_{1\ SP},\ \ddot{Z}_{SP} = \ddot{q}_{1\ SP}$$

$$\begin{Bmatrix} \phi_{SP} \\ \theta_{SP} \end{Bmatrix} = \begin{bmatrix} \cos\psi_c & -\sin\psi_c \\ \sin\psi_c & \cos\psi_c \end{bmatrix} \begin{Bmatrix} -K_{XC} X_C \\ K_{YC} Y_C \end{Bmatrix}$$

$$\begin{Bmatrix} \dot{\phi}_{SP} \\ \dot{\theta}_{SP} \end{Bmatrix} = \begin{bmatrix} \cos\psi_c & -\sin\psi_c \\ \sin\psi_c & \cos\psi_c \end{bmatrix} \begin{Bmatrix} -K_{XC} \dot{X}_C \\ K_{YC} \dot{Y}_C \end{Bmatrix}$$

$$\begin{Bmatrix} \ddot{\phi}_{SP} \\ \ddot{\theta}_{SP} \end{Bmatrix} = 0$$

$K_{XC} = \partial\phi_{SP} / X_C$ RA (123)

$K_{YC} = \partial\theta_{SP} / Y_C$ RA (124)

<u>$N_{SP} = 0$</u>

Z_{SP} is defined as in option ($N_{SP} = 2$)

ϕ_{SP} and Θ_{SP} are defined as in option ($N_{SP} = 1$)

<u>Reference</u> ($N_{REF} = 6$)

$$\begin{Bmatrix} X_o \\ Y_o \\ Z_o \\ \phi_o \\ \Theta_o \\ \psi_o \end{Bmatrix}_{REF} = \{q_{REF}\}$$

Likewise for

$$\begin{Bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix}_{REF}$$

and

$$\begin{Bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{Bmatrix}_{REF}$$

Shaft Subsystem Options N_S = 0, 1, 2, 3, 4, 5, 6

$N_S = 6$

$$\begin{Bmatrix} X \\ Y \\ Z \\ \phi \\ \theta \\ \psi \end{Bmatrix}_S = \left\{ q_S \right\}$$

Likewise for

$$\left\{ \dot{q}_S \right\}$$

and

$$\left\{ \ddot{q}_S \right\}$$

For options where $N_S < 6$, a special vector compress-decompress routine assigns the appropriate coordinates.

Rotor Subsystem

$N_R = 1$

$$\psi_R = \left\{ q_R \right\}$$

$$\dot{\psi}_R = \left\{ \dot{q}_R \right\}$$

$$\ddot{\psi}_R = \left\{ \ddot{q}_R \right\}$$

$\underline{N_R = 0}$

$$\psi_R = \Omega_{\dot{R}} t$$

$$\dot{\psi}_R = \Omega_R \quad \text{INPUT}$$

$$\dot{\psi}_R = 0$$

3.3 CMPRES

This subprogram will compress a vector, X, to a vector, Q;

$$\underset{(6x1)}{\{X\}} \longrightarrow \underset{(Nx1)}{\{Q\}}$$

or decompress a vector, Q, restoring X,

$$\underset{(Nx1)}{\{Q\}} \longrightarrow \underset{(6x1)}{\{X\}}$$

by using a mapping function, IND, of indicators. The indicators operate positionally. If an element is 1, the corresponding element of X is passed to Q. Also,

$$N = \sum_{i=1}^{6} IND_i$$

Description of Arguments

Input

COMPR logical variable
If COMPR = .TRUE., then X is compressed to Q.
If COMPR = .FALSE., then the vector X is reconstructed from Q.

IND Vector of indicators used in the mapping.
IND_i = 1, then element i is to be passed
IND_i = 0, then element i is to be ignored

Input/Output

If COMPR = .TRUE.

then X(6) is input
Q(N) is output.

If COMPR = .FALSE.

then Q(N) is input
X(6) is output.

Undefined X's will be set to zero.

3.4 TRIM

Subprogram TRIM provides a means by which the equations representing the vehicular system can be put in a so called state of equilibrium. Actually, a more general capability is available. Find x such that

$$g(x) = f(x) - c = 0.$$

The algorithm to solve this problem is simply stated

$$x^{n+1} = x^{n} - k \cdot h^{n}(x) \cdot dt$$

$$h^{n}(x) = h^{n-1}(x) + k^{*} (f^{n}(x) - c).$$

The user can choose up to ten functions and ten corresponding independent variables from a supplied table. The collection of functions is not treated as a system but rather as independent relationships. Convergence is determined not by monitoring the function, $g(x)$, but rather the relative change in the independent variable over one rotor revolution. Convergence is assumed if

$$x - x^{*} < \varepsilon$$

for all x. x^{*} is the value at the end of the previous cycle.

A general, not detailed, flow diagram is presented in Figure 11.

It should be noted that during trim the reference subsystem and the rotor subsystem are not integrated. Thus, as indicated on the diagram certain coordinate overrides must be performed.

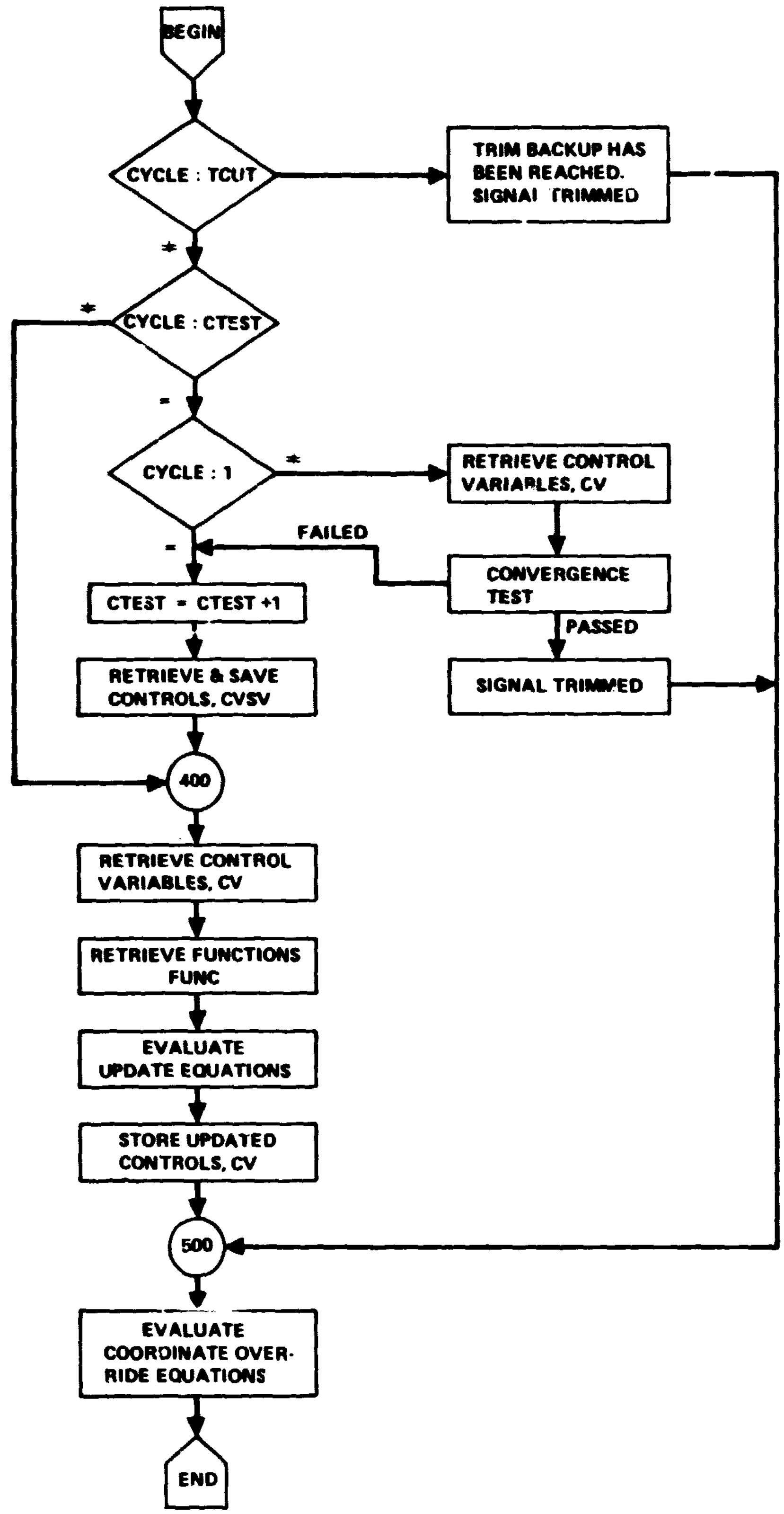


Figure 11. - TPIM flow diagram.

The coordinate overrides are

ϕ_E = input

$\theta_E = F(\phi_E, \gamma)$, γ - input

$\psi_E = 0$

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_{REF} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_F = f(V_T, \alpha_F, \beta_F). \quad V_T, \alpha_F, \beta_F \text{ are input.}$$

$\dot{\phi}_E = 0$

$\dot{\theta}_E = 0$

$\dot{\psi}_E = f(L_F)$, L_F = input

$$\begin{Bmatrix} p \\ q \\ r \end{Bmatrix}_{REF} = f(\psi_E, \phi_E, \theta_E)$$

$\ddot{\psi}_R = 0$

$\dot{\psi}_R = \Omega$

$\psi_R = \Omega t$

Other inputs which are central to the trim process are listed below.

Inputs	Description	RA
NC	Number of control laws	37
CVP(10)	Control variable pointers	1301-1310
FUNCP(10)	Control function pointers	1311-1320
CVE(10)	Convergence errors	1321-1330
CVK(10)	Control variable gains	1331-1340
FUNCD(10)	Desired function values	671-680
TCUT	Trim cutoff, cycles	36

The description of arguments are:

TRIMED	Logical variable indicating trim condition
CYCLE	Main rotor cycle count
CTEST	Trim convergence cycle indicator
IC	Generalized coordinate integrator control
NCC	Generalized coordinate count
TCYCLE	Cycle number when system declared trimmed

3.5 RETRVE, STORE

This is a multiple entry subprogram designed to retrieve and store trim parameters which are defined in various COMMONS. The process is table driven. The current trim parameter definitions are given in the following table.

Parameter	I.D. Number	Common Block Ident.						Block Offset
		Blank	/Input/					
u_{REF}	1	X						186
v_{REF}	2	X						187
w_{REF}	3	X						188
p_{REF}	4	X						363
q_{REF}	5	X						364
r_{REF}	6	X						365
$\%\phi_E$	7	X						153
$\%X_{C,T}$	8		X					53
$\%Y_{C,T}$	9		X					54
$\%r_{C,T}$	10		X					55
$\%\theta_{C,T}$	11		X					56
$\%\kappa_{P,T}$	12		X					57
α_E	13		X					58

Parameter	K.D. Number	Common Block Ident.						Block Offset
		Blank	/Input/					
γ_E	14		X					63
$M_{ZZ_{END}}$	15		X					92
$\ddot{\psi}_R$	16	X						320
M_{ZR}	17	X						963
M_{XR}	18	X						961
M_{YR}	19	X						962
F_{ZR}	20	X						960

The retrieve/store mechanism is depicted in Figure 12.

A description of the arguments is

N Number of parameters to retrieve/store

POINTR Vector of N pointers. The values are drawn from the ID numbers defined in the definition table.

V Vector of parameters retrieved or stored.

3.6 TFORM

This program computes all time varying transformation matrices.

Input Data

All necessary angles are in blank common

Quantity	Name
ψ_R	PSIR
ϕ_{SP}	PHSP
θ_{SP}	THSP
ϕ_S	PHIS ≡ DISPS(4)
θ_S	THTS ≡ DISPS(5)
ψ_S	PSIS ≡ DISPS(6)
ϕ_E	PHIE ≡ AUXE(1)
θ_E	THETE ≡ AUXE(2)
ψ_E	PSIE ≡ AUXE(3)

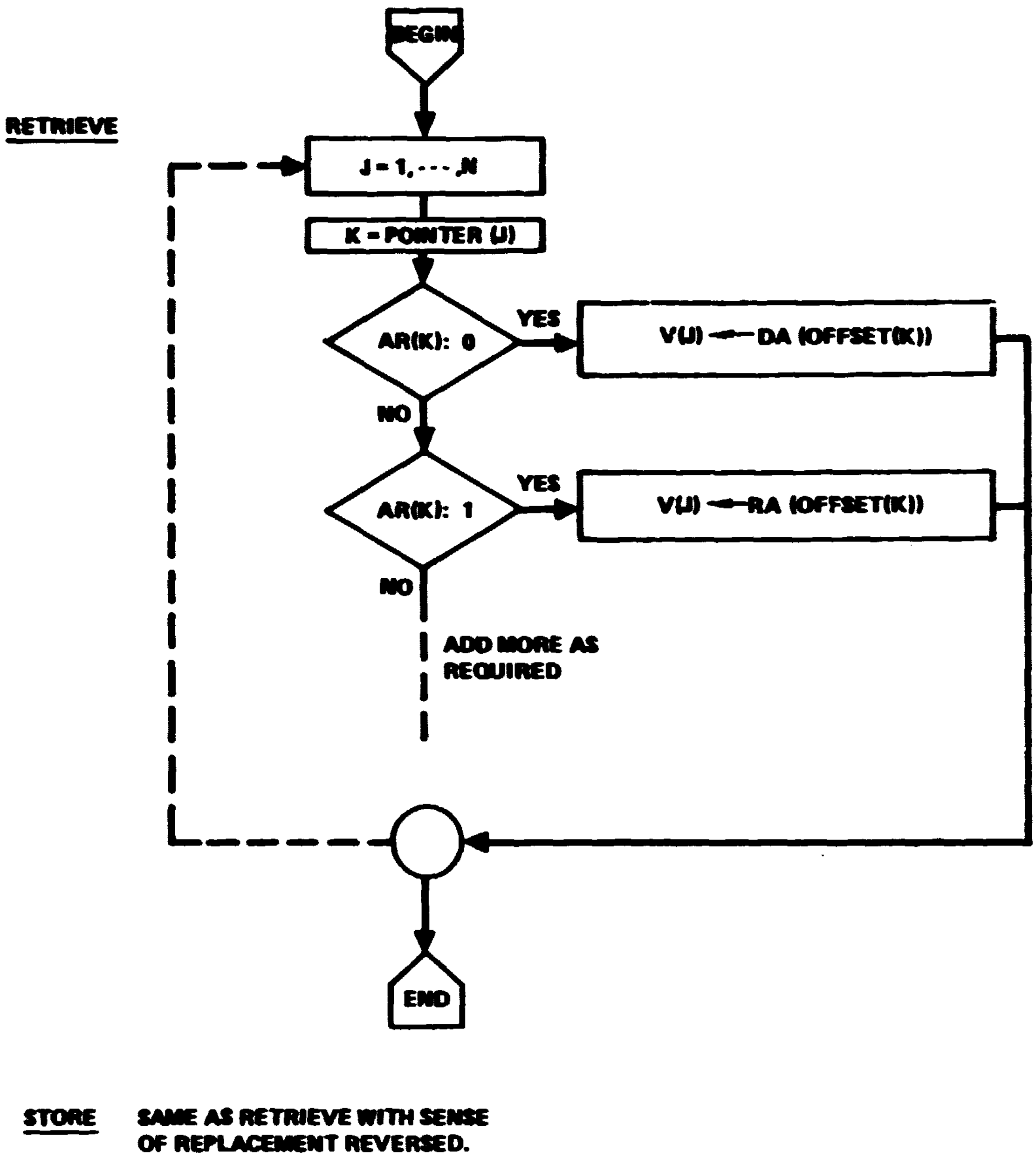


Figure 12. - RETRVE/STORE flow diagram.

N_b	NB	COMMON/DATAS1/ number of main rotor blades
$T_{R \to BL_n}$	TRBL(3,3,7)	COMMON/TFORMS/ transforms from rotor to blades, computed in MAIN.

Subprogram TFORM uses one subordinate routine, EULERT. EULERT evaluates a general Euler transformation. The use of EULERT is expressed functionally as

$$T_E(\phi, \theta, \psi)$$

Output

All transforms appear and are available thru labeled COMMON/TFORMS/. The transforms are defined below.

Hub to Rotor

$$\left[T_{R \leftarrow H}\right] = \begin{bmatrix} -\cos\psi_R & \sin\psi_R & 0 \\ \sin\psi_R & \cos\psi_R & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Hub to Blade n

$$\left[T_{BL_n \leftarrow H}\right] = \left[T_{BL_n \leftarrow R}\right]\left[T_{R \leftarrow H}\right]$$

for each blade n = 1, ..., N_b.

Hub to Swashplate (Rotating)

$$\left[T_{SP \leftarrow H}\right] = T_E(\phi_{SP}, \theta_{SP}, 0)$$

Mount Base to Hub (Transmission Isolation System)

$$\left[T_{S \to H}\right] \equiv \left[T_{MB \to H}\right] = T_E(\phi_s, \theta_s, \psi_s)$$

Reference to Earth

$$\left[T_{REF \rightarrow E}\right] = T_E\,(-\phi_E, -\theta_E, -\psi_E)$$

3.7 EULERT

This program calculates the Euler transformation matrix given by

$$T_E = T(\psi) \cdot T(\theta) \cdot T(\phi)$$

where

$$T(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$T(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

Input

ϕ - PHI , Radians

θ - THETA, Radians

ψ - PSI , Radians

Output

T_E - TE(3,3) 3 x 3 transformation matrix

3.8 MOTION

Given the basic reference motions, this routine computes all major linear and angular motions except blade and blade element motions. They are computed elsewhere. The equations of interest are presented below.

Gravity in Ref System

$$\begin{Bmatrix} g_x \\ g_y \\ g_z \end{Bmatrix}_{REF} = \left[T_{E \rightarrow REF} \right] \begin{Bmatrix} 0 \\ 0 \\ g \end{Bmatrix}$$

Reference Acceleration

$$\begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{Bmatrix}_{REF} = \begin{Bmatrix} \ddot{X}_o \\ \ddot{Y}_o \\ \ddot{Z}_o \end{Bmatrix}_{REF} + \begin{Bmatrix} \\ g \\ \\ \end{Bmatrix}_{REF} - \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}_{REF} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_{REF}$$

Earth Rates

$$\dot{\psi}_E = (q_{REF} \sin \phi_E + r_{REF} \cos \phi_E)/\cos \theta_E$$

$$\dot{\phi}_E = p_{REF} + \dot{\psi}_E \sin \theta_E$$

$$\dot{\theta}_E + q_{REF} \cos \phi_E - r_{REF} \sin \phi_E$$

Earth Motions

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_E = \left[T_{E \leftarrow REF} \right] \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_{REF}$$

and

AUXD(21) = u_E

AUXD(22) = $-v_E$

AUXD(23) = $-w_E$

Fuselage

Quantity	FORTRAN
$$\begin{Bmatrix} p \\ q \\ r \end{Bmatrix}_F = \begin{Bmatrix} p \\ q \\ r \end{Bmatrix}_{REF}$$	= VELRF (4) = VELRF (5) = VELRF (6)
$$\begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix}_F = \begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix}_{REF}$$	= ACCRF (4) = ACCRF (5) = ACCRF (6)
$$\begin{Bmatrix} \dot{X}_o \\ \dot{Y}_o \\ \dot{Z}_o \end{Bmatrix}_F^I = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_F = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_{REF}$$	= AUXE (4) = AUXE (5) = AUXE (6)

$$\begin{Bmatrix} \ddot{X}_o \\ \ddot{Y}_o \\ \ddot{Z}_o \end{Bmatrix}_F = \begin{Bmatrix} \ddot{X}_o \\ \ddot{Y}_o \\ \ddot{Z}_o \end{Bmatrix}_{REF} \quad \begin{matrix} = \text{ACCRF (1)} \\ = \text{ACCRF (2)} \\ = \text{ACCRF (3)} \end{matrix}$$

Velocity and acceleration of MB relative to fuselage

$$\begin{Bmatrix} \dot{X}_o \\ \dot{Y}_o \\ \dot{Z}_o \end{Bmatrix}_F^{REL} \quad \text{and} \quad \begin{Bmatrix} \ddot{X}_o \\ \ddot{Y}_o \\ \ddot{Z}_o \end{Bmatrix}_F^{REL}$$

use routine LINVEC with arguments

$$\{V_1\} = \begin{Bmatrix} X_o \\ Y_o \\ Z_o \end{Bmatrix}_{F-MB} \quad ; \quad \{\dot{V}_1\} = 0 \quad ; \quad \{\ddot{V}_1\} = 0$$

$$\{\dot{V}_2\} = \begin{Bmatrix} \dot{X}_o \\ \dot{Y}_o \\ \dot{Z}_o \end{Bmatrix}_F^{I} \; ; \; \{\ddot{V}_2\} = \begin{Bmatrix} \ddot{X}_o \\ \ddot{Y}_o \\ \ddot{Z}_o \end{Bmatrix}_F \; ; \; \{W\} = \begin{Bmatrix} p \\ q \\ r \end{Bmatrix}_F \; ; \; \{\dot{W}\} = \begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix}_F$$

<u>Mount Base</u> (Transmission support)

$$VMB, \begin{Bmatrix} \dot{X}_O \\ \dot{Y}_O \\ \dot{Z}_O \end{Bmatrix}^{I}_{MB} = \begin{bmatrix} T_{O_F} \rightarrow MB \end{bmatrix} \begin{Bmatrix} \dot{X}_O \\ \ddot{Y}_O \\ \dot{Z}_O \end{Bmatrix}^{REL}_{F}$$

$$VDMB, \begin{Bmatrix} \ddot{X}_O \\ \ddot{Y}_O \\ \ddot{Z}_O \end{Bmatrix}^{I}_{MB} = \begin{bmatrix} T_{O_F} \rightarrow MB \end{bmatrix} \begin{Bmatrix} \ddot{X}_O \\ \ddot{Y}_O \\ \ddot{Z}_O \end{Bmatrix}^{REL}_{F}$$

$$WMB, \begin{Bmatrix} p \\ q \\ r \end{Bmatrix}_{MB} = \begin{bmatrix} T_{O_F} \rightarrow MB \end{bmatrix} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix}_{F}$$

$$WDMB, \begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix}_{MB} = \begin{bmatrix} T_{O_F} \rightarrow MB \end{bmatrix} \begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix}_{F}$$

$$\text{DMH,} \quad \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_{MB-H} = \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_S + W(\phi_s, \theta_s, \psi_s) \begin{Bmatrix} X_o \\ Y_o \\ Z_o \end{Bmatrix}_{MB-H}$$

$$\text{VMH,} \quad \begin{Bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{Bmatrix}_{MB-H}^{REL} = \begin{Bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{Bmatrix}_S + W(\dot{\phi}_s, \dot{\theta}_s, \dot{\psi}_s) \begin{Bmatrix} X_o \\ Y_o \\ Z_o \end{Bmatrix}_{MB-H}$$

$$\text{AMH,} \quad \begin{Bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{Bmatrix}_{MB-H}^{REL} = \begin{Bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{Bmatrix}_S + W(\ddot{\phi}_s, \ddot{\theta}_s, \ddot{\psi}_s) \begin{Bmatrix} X_o \\ Y_o \\ Z_o \end{Bmatrix}_{MB-H}$$

<u>Hub</u>

Compute a set of intermediate vectors $\{VEC\}$ and $\{\dot{VEC}\}$ using LINVEC with

$$\{V_1\} = \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_{MB-H}$$

$$\{\dot{V}_1\} = \begin{Bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{Bmatrix}_{MB-H}^{REL}$$

$$\{\ddot{V}_1\} = \begin{Bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{Bmatrix}_{MB-H}^{REL}$$

$$\left\{ \dot{V}_2 \right\} = \left\{ \begin{matrix} \dot{X}_o \\ \dot{Y}_o \\ \dot{Z}_o \end{matrix} \right\}^{I}_{MB}$$

$$\left\{ \ddot{V}_2 \right\} = \left\{ \begin{matrix} \ddot{X}_o \\ \ddot{Y}_o \\ \ddot{Z}_o \end{matrix} \right\}^{I}_{MB}$$

$$\left\{ W \right\} = \left\{ \begin{matrix} p \\ q \\ r \end{matrix} \right\}_{MB}$$

$$\left\{ \dot{W} \right\} = \left\{ \begin{matrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{matrix} \right\}_{MB}$$

then

$$\text{VHUB}, \left\{ \begin{matrix} \dot{X}_o \\ \dot{Y}_o \\ \dot{Z}_o \end{matrix} \right\}^{I}_{H} = \left[T_{MB \to H} \right] \left\{ \text{VEC} \right\}$$

$$\text{VDHUB}, \left\{ \begin{matrix} \ddot{X}_o \\ \ddot{Y}_o \\ \ddot{Z}_o \end{matrix} \right\}_{N} = \left[T_{MB \to H} \right] \left\{ \dot{\text{VEC}} \right\}$$

Using program ANGVEC, compute

$$\begin{Bmatrix} p \\ q \\ r \end{Bmatrix}_H \quad \text{and} \quad \begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix}_H$$

The arguments are

$$\{W\} = \begin{Bmatrix} p \\ q \\ r \end{Bmatrix}_{MB}$$

$$\{\dot{W}\} = \begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix}_{MB}$$

$$\{A\} = \begin{Bmatrix} \phi \\ \theta \\ \psi \end{Bmatrix}_S$$

$$\{\dot{A}\} = \begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix}_S$$

$$\{\ddot{A}\} = \begin{Bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{Bmatrix}_S$$

Rotor

Quantity	FORTRAN

$$\begin{Bmatrix} \dot{X}_o \\ \dot{Y}_o \\ \dot{Z}_o \end{Bmatrix}_R^I = \left[T_{H \to R}\right] \begin{Bmatrix} \dot{X}_o \\ \dot{Y}_o \\ \dot{Z}_o \end{Bmatrix}_H^I \begin{matrix} = \text{VR(1)} \\ = \text{VR(2)} \\ = \text{VR(3)} \end{matrix}$$

$$\begin{Bmatrix} \ddot{X}_o \\ \ddot{Y}_o \\ \ddot{Z}_o \end{Bmatrix}_R = \left[T_{H \to R}\right] \begin{Bmatrix} \ddot{X}_o \\ \ddot{Y}_o \\ \ddot{Z}_o \end{Bmatrix}_H \begin{matrix} = \text{VDR(1)} \\ = \text{VDR(2)} \\ = \text{VDR(3)} \end{matrix}$$

$$\begin{Bmatrix} p \\ q \\ r \end{Bmatrix}_R \begin{matrix} = \text{WR(1)} \\ = \text{WR(2)} \\ = \text{WR(3)} \end{matrix}$$ Eq 74 of Vol. I

$$\begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix}_R \begin{matrix} = \text{WDR(1)} \\ = \text{WDR(2)} \\ = \text{WDR(3)} \end{matrix}$$ Eq 75 of Vol. I

Swashplate

Quantity	FORTRAN

$$\begin{Bmatrix} p \\ q \\ r \end{Bmatrix}_{SP} \begin{matrix} = \text{WSP(1)} \\ = \text{WSP(2)} \\ = \text{WSP(3)} \end{matrix}$$ Eq 159 of Vol. I

$$\begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix}_{SP} \begin{matrix} = \text{WDSP(1)} \\ = \text{WDSP(2)} \\ = \text{WDSP(3)} \end{matrix}$$ Eq 160 of Vol. I

Gravity in the rotor system

$$\{g\}_R = [T_{R-H}]\ [T_{H-MB}]\ [T_{MB-F}]\ \{g\}_{REF}$$

3.9 LINVEC

This subprogram computes general linear inertial velocity and acceleration vectors of the form

$$\{\dot{V}\} = \{\dot{V}_2\} + \{\dot{V}_1\} + [W]\ \{V_1\}$$

$$\{\ddot{V}\} = \{\ddot{V}_2\} + \{\ddot{V}_1\} + [W]\ [W]\ \{V_1\} + [\dot{W}]\ \{V_1\} + 2\ [W]\ \{\dot{V}_1\}$$

where

$$[W] = \begin{bmatrix} 0 & -W(3) & W(2) \\ W(3) & 0 & -W(1) \\ -W(2) & W(1) & 0 \end{bmatrix}$$

and similarly

$$[\dot{W}] = \begin{bmatrix} 0 & -\dot{W}(3) & \dot{W}(2) \\ \dot{W}(3) & 0 & -\dot{W}(1) \\ -\dot{W}(2) & \dot{W}(1) & 0 \end{bmatrix}$$

The subroutine argument list is

$$(V_1, \dot{V}_1, \ddot{V}_1, \dot{V}_2, \ddot{V}_2, W, \dot{W}, \dot{V}, \ddot{V})$$

all arguments are (3 x 1) vectors.

3.10 ANGVEC

The subroutine computes general angular velocity and acceleration vectors.

The argument list is (WA, WDA, A, AD, ADD, WB, WDB) where

Inputs

$$WA = \begin{Bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{Bmatrix}_a \quad \text{normally} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix}_a$$

$$WDA = \begin{Bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{Bmatrix}_a \quad \text{derivative of WA}$$

$$A = \begin{Bmatrix} \phi \\ \theta \\ \psi \end{Bmatrix} \quad \text{defines transformation from system } a \rightarrow b$$

$$\dot{A} = \begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix} \quad \text{derivative of A}$$

$$\ddot{A} = \begin{Bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{Bmatrix} \quad \text{derivative of } \dot{A}$$

Output

$$WB = \begin{Bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{Bmatrix}_b \rightarrow \text{normally} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix}_b$$

$$WDB = \begin{Bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{Bmatrix}_b \rightarrow \text{normally} \begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix}_b$$

Algorithm

$$T(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

$$\dot{T}(\phi) = \dot{\phi} \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\sin\phi & \cos\phi \\ 0 & -\cos\phi & -\sin\phi \end{bmatrix}$$

$$T(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$\dot{T}(\theta) = \dot{\theta} \begin{bmatrix} -\sin\theta & 0 & -\cos\theta \\ 0 & 0 & 0 \\ \cos\theta & 0 & -\sin\theta \end{bmatrix}$$

$$T(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\dot{T}(\psi) = \dot{\psi} \begin{bmatrix} -\sin\psi & \cos\psi & 0 \\ -\cos\psi & -\sin\psi & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\left\{ V_1 \right\} = \left\{ \begin{matrix} \dot{\phi} \\ 0 \\ 0 \end{matrix} \right\} + T(\phi) \left\{ W \right\}_a$$

$$\left\{ \dot{V}_1 \right\} = \left\{ \begin{matrix} \ddot{\phi} \\ 0 \\ 0 \end{matrix} \right\} + \dot{T}(\phi) \left\{ W \right\}_a + T(\phi) \left\{ \dot{W} \right\}_a$$

$$\left\{ V_2 \right\} = \left\{ \begin{matrix} 0 \\ \dot{\theta} \\ 0 \end{matrix} \right\} + T(\theta) \left\{ V_1 \right\}$$

$$\left\{ \dot{V}_2 \right\} \quad \left\{ \begin{matrix} 0 \\ \ddot{\theta} \\ 0 \end{matrix} \right\} + \dot{T}(\theta) \left\{ V_1 \right\} + T(\theta) \left\{ \dot{V}_1 \right\}$$

Finally,

$$\left\{ W \right\}_b = \left\{ \begin{matrix} 0 \\ 0 \\ \dot{\psi} \end{matrix} \right\} + T(\psi) \left\{ V_2 \right\}$$

and

$$\left\{ \dot{W} \right\}_b = \left\{ \begin{matrix} 0 \\ 0 \\ \ddot{\psi} \end{matrix} \right\} + \dot{T}(\psi) \left\{ V_2 \right\} + T(\psi) \left\{ \dot{V}_2 \right\}$$

3.11 CNTROL

Subprogram CNTROL assembles pilot controls with trim inputs and stability augmentation inputs. These combined inputs then operate the rotor and fixed surface panels. Controls are presented in Section 7 of Volume I. The general flow of the routine is depicted in Figure 13. Notice in the diagram that an abbreviated set of equations is used during TRIM. Whereas, during FLY, pilot control tables and SAS inputs are active.

During trim, the control equations are given by the following equations.

$\theta_o = K_{\theta FS} \cdot (\% \theta_{o,T})$ variable

$\delta_p = K_{PFS} \cdot (\% \delta_{p,T})$ variable

$\delta_{DB} = K_{DBFS} \cdot (\% \delta_{DB,IC})$ constant

$\delta_{FL} = K_{FLFS} \cdot (\% \delta_{FL,IC})$ constant

$\delta_{iw} = K_{iwFS} \cdot (\% \delta_{iw,IC})$ constant

$\delta i_{HT} = K_{iHTFS} \cdot (\% \delta_{iHT,IC})$ constant

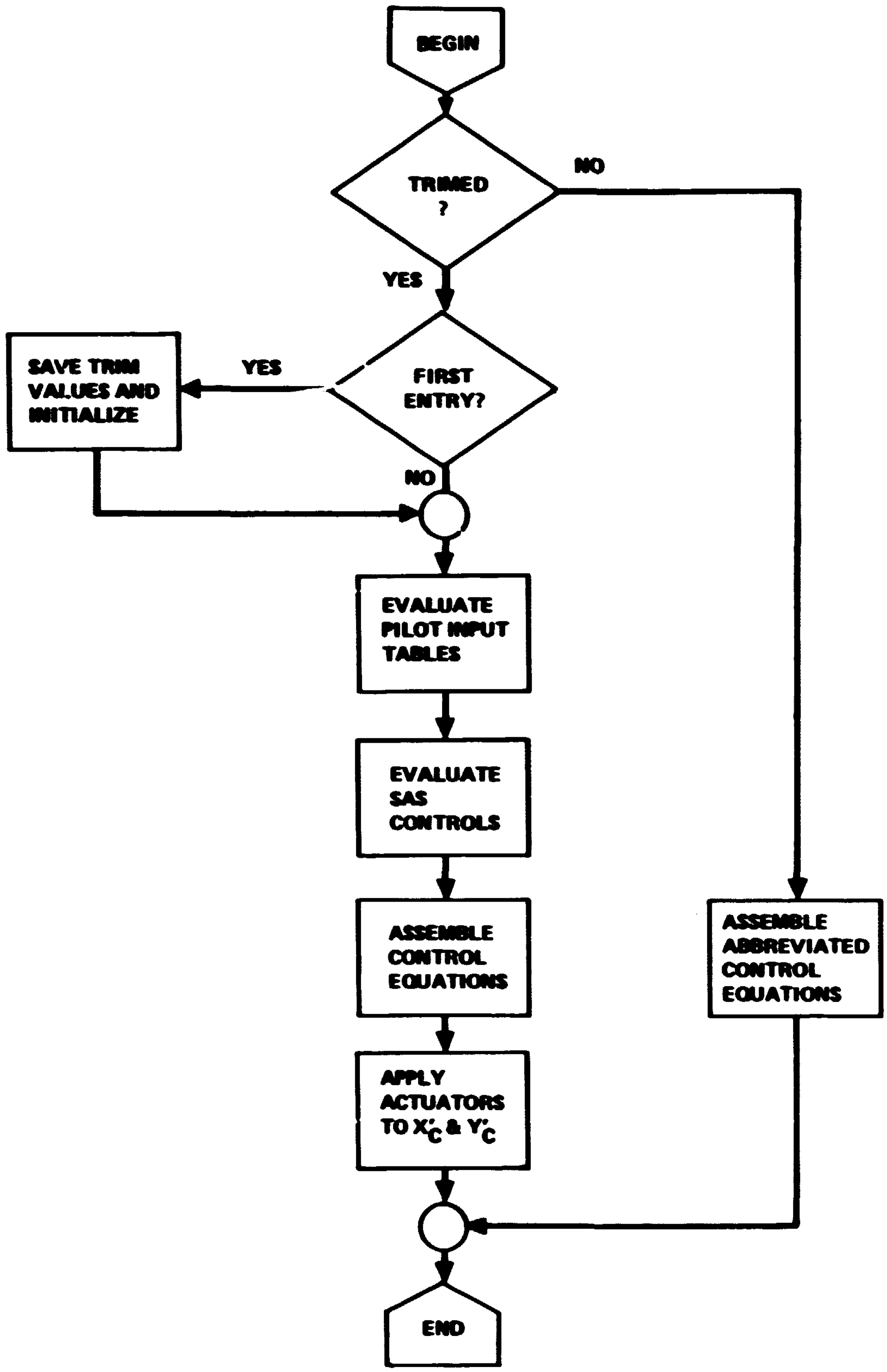


Figure 13. - CNTROL flow diagram.

$$\delta_{EL} = K_{ELFS} \cdot G_{EL} \cdot (\% \, x_{c,T}) + \delta_{EL,0}$$

$$\delta_{AIL} = K_{AILFS} \cdot G_{AIL} \cdot (\% \, y_{c,T}) + \delta_{AIL,0}$$

$$\delta_{RUD} = K_{RUDFS} \cdot G_{RUD} \cdot (\% \, r_{c,T}) + \delta_{RUD,0}$$

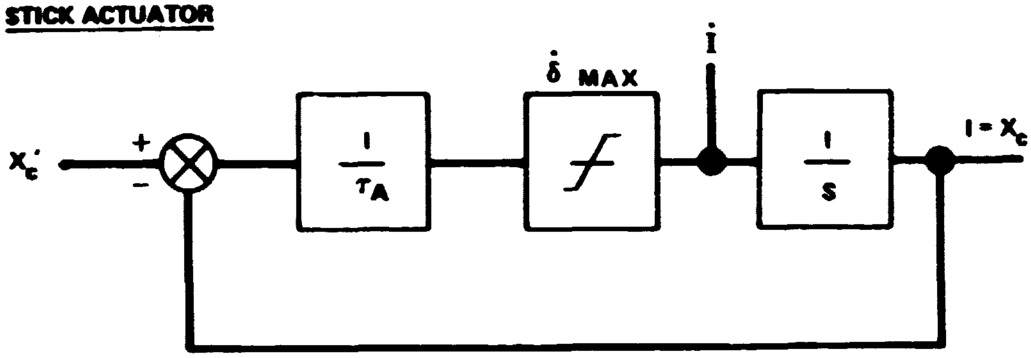
$$\theta_{o_{TR}} = K_{TRFS} \cdot G_{TR} \cdot (\% \, r_{c,T}) + \theta_{o_{TR,0}} + \frac{\partial \theta_{OTR}}{\partial \theta_o} \cdot \theta_o$$

$$X_c = K_{XCFS} \cdot G_c \cdot (\% \, x_{c,T}) + \frac{\partial x_c}{\partial \theta_o} \cdot \theta_o$$

$$Y_c = K_{YCFS} \cdot G_c \cdot (\% \, y_{c,T}) + \frac{\partial y_c}{\partial \theta_o} \cdot \theta_o$$

During FLY, a number of pilot controls may be input as a function of time. The controls available are Xc_p, Yc_p, rc_p, δp_p, θo_p, δDB_p, δFL_p, δiw_p, δiHT_p. These functions are evaluated via linear interpolation. Continuing, SAS inputs are available for six functions, Xc_{SAS}, Yc_{SAS}, δEL_{SAS}, δAL_{SAS}, θo_{TRSAS}, and δRUD_{SAS}. The SAS logic is found in routine, SAS. The control equations are straight forward and can be found in Volume I and in the program listing. Finally, stick actuators are applied to the longitudinal and lateral stick controls. Actuator logic is depicted in Figure 14 for the X-stick. Similar logic applies to the Y-stick.

STICK ACTUATOR



$$\dot{I} = \text{MIN}\left(|\dot{I}_L| , \dot{\delta}_{\text{MAX}} \right) \cdot \dot{I}_L / |\dot{I}_L|$$

$$\dot{I}_L = \frac{1}{\tau_A} \left(X_c' - I \right)$$

$$I_{IC} = X_c' \big|_{\text{TRIM}}$$

TWO CHANNELS X_c AND Y_c

INPUTS

NAME	FORTRAN	RA
τ_{xc}	TAUXC	282
τ_{yc}	TAUYC	283
$\dot{\delta}_{MYC}$	RATLXC	437
$\dot{\delta}_{MYC}$	RATLYC	438

Figure 14. Actuator logic.

3.12 SAS

This is a general SAS control routine. The SAS block diagram is presented in Figure 15. The digital implementation of that diagram is

$$S_1 = K_1 \cdot \varepsilon_a + K_2 \cdot \varepsilon_b + I_1$$

$$S_2 = K_3 \cdot \varepsilon_b - I_1$$

$$S_3 = S_1 - I_2$$

$$S_4 = \min(|S_3|, C_{LIM}) \cdot S_3/|S_3|$$

$$\dot{I}_1 = \frac{1}{\tau} \cdot S_2$$

$$\dot{I}_2 = K_4 \cdot S_3 \qquad K_4 = \frac{1}{\tau_1}$$

$$X_{SAS} = K_S \cdot S_4$$

All integrals have zero initial conditions

$$I_{1_{IC}} = I_{2_{IC}} = 0$$

The error signals are of the form

$$\varepsilon = q - q_{TRIM} \quad .$$

The routine is invoked by the call

```
CALL SAS(EA, EB, SASD, I1, I2, ID1, ID2, XSAS)
```

where

EA ε_a } Error signals
EB ε_b

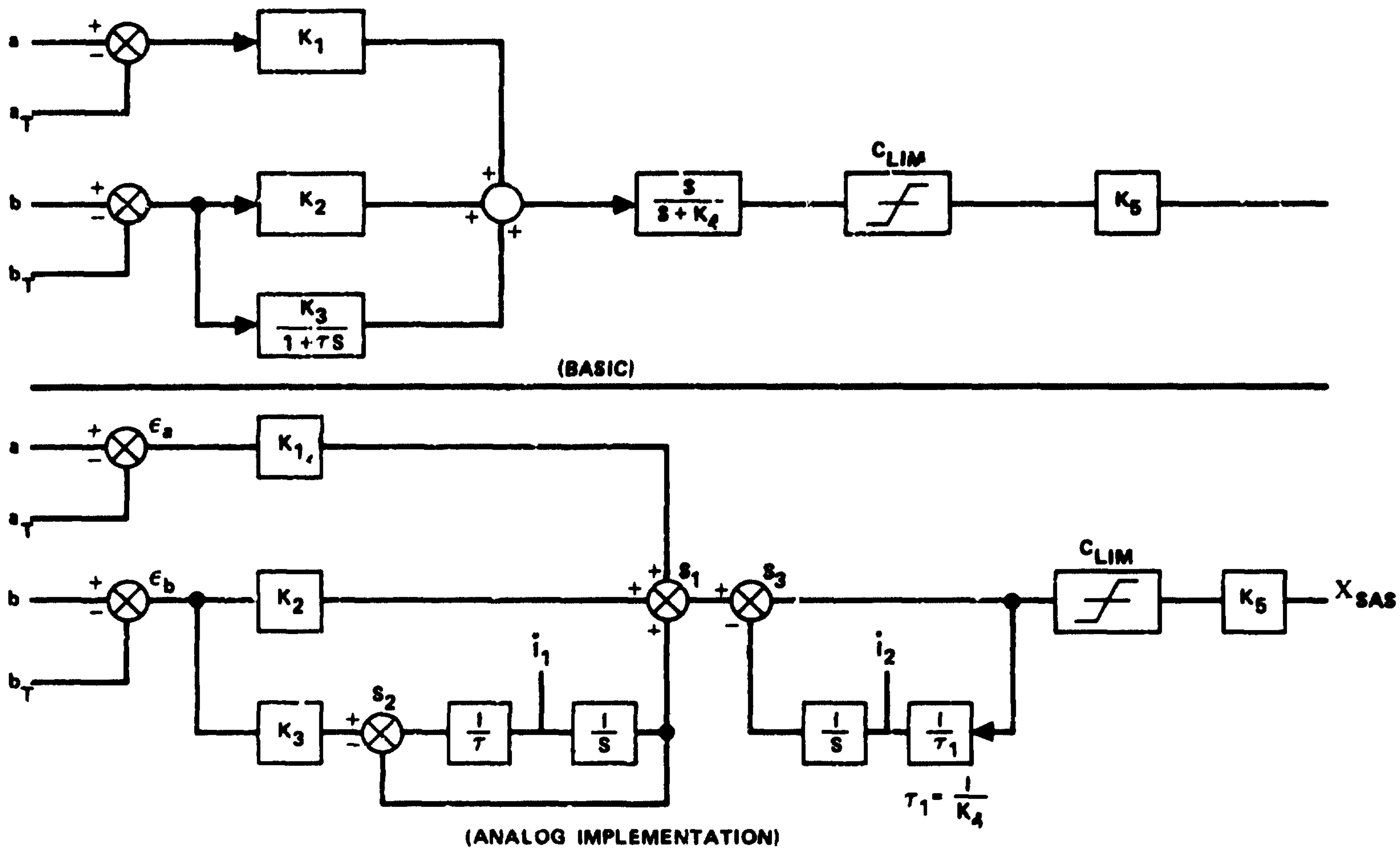


Figure 15. - SAS block diagram.

SASD Vector of constants for the particular control where

K_1, K_2 — Feedback gains

K_3 Filter gain

K_4 Frequency breakpoint

K_5 Output gain

τ Filter time constant

C_{LIM} Limiter

I1 I_1 Low pass filter integral

I2 I_2 Washout integral

ID1 $\dot{I}_1$, ID2 $\dot{I}_2$ — Integrands to be integrated elsewhere

XSAS Control output

3.13 CYCLIC

Subroutine CYCLIC provides the linkage between the swashplate and the main rotor.

The main rotor collective is determined in CNTROL. However, the associated derivatives are determined here.

$$\dot{\theta}_o = - \dot{Z}_{SP}/E$$

$$\ddot{\theta}_o = - \ddot{Z}_{SP}/E$$

The cyclic angles are

$$\left(\frac{d}{E}\right) = \left(\frac{d}{E}\right)_0 + \theta_o \left(\frac{d}{E}\right)_1$$

$$A_{1S} = \left(\frac{d}{E}\right) [\phi_{SP} \sin \beta_G + \theta_{SP} \cos \beta_G]$$

$$B_{1S} = \left(\frac{d}{E}\right) [\phi_{SP} \cos \beta_G - \theta_{SP} \sin \beta_G]$$

The derivatives are

$$\dot{A}_{1S} = \left(\frac{d}{E}\right) [\dot{\phi}_{SP} \sin \beta_G + \dot{\theta}_{SP} \cos \beta_G] + \left(\frac{d}{E}\right)_1 \dot{\theta}_o A_{1S}/(d/E)$$

$$\dot{B}_{1S} = \left(\frac{d}{E}\right) [\dot{\phi}_{SP} \cos \beta_G - \dot{\theta}_{SP} \sin \beta_G] + \left(\frac{d}{E}\right)_1 \dot{\theta}_o B_{1S}/(d/E)$$

$$\ddot{A}_{1S} = \left(\frac{d}{E}\right) [\ddot{\phi}_{SP} \sin \beta_G + \ddot{\theta}_{SP} \cos \beta_G]$$

$$+ 2 \left(\frac{d}{E}\right)_1 \dot{\theta}_o [\dot{\phi}_{SP} \sin \beta_G + \dot{\theta}_{SP} \cos \beta_G]$$

$$+ \left(\frac{d}{E}\right)_1 \ddot{\theta}_o A_{1S}/(d/E)$$

$$\ddot{B}_{1S} = \left(\frac{d}{E}\right) [\ddot{\phi}_{SP} \cos \beta_G - \ddot{\theta}_{SP} \sin \beta_G]$$

$$+2\left(\frac{d}{E}\right)_1 \dot{\theta}_o [\dot{\phi}_{SP} \cos \beta_G - \dot{\theta}_{SP} \sin \beta_G]$$

$$+ \left(\frac{d}{E}\right)_1 \ddot{\theta}_o B_{1S}/(d/E)$$

3.14 PDERIV

This routine computes partial derivative matrices relating a set of dependent coordinates, τ_O, to a set of independent coordinates, τ_I. Not all program partials are computed here. Those computed in this routine are

$$\left\{\frac{\partial \phi_f}{\partial q_{SP}}\right\}_n = \left\{\begin{matrix} -\left(\frac{d}{E}\right)\left[\sin(\psi_R + \psi_{BLn}) \sin\beta_G + \cos(\psi_R + \psi_{BLn}) \cos\beta_G\right] \\ -\left(\frac{d}{E}\right)\left[\sin(\psi_R + \psi_{BLn}) \cos\beta_G - \cos(\psi_R + \psi_{BLn}) \sin\beta_G\right] \\ -\left(1\right) - \left(\frac{d}{E}\right)_1 \theta_c / \left(\frac{d}{E}\right) \quad /E \end{matrix}\right\}$$

where

$$\theta_c = A_{1S} \sin(\psi_R + \psi_{BLn}) + \beta_{1S} \cos(\psi_R + \psi_{BLn})$$

$\left\{\frac{\partial \phi_f}{\partial q_{SP}}\right\}_n$ is an angle subset of $\left\{\frac{\partial \phi_f}{\partial \tau_{SP}}\right\}_n$

$$\left\{\frac{\partial \phi_f}{\partial \psi_R}\right\}_n = A_{1S} \sin(\psi_R + \psi_{BLn}) - B_{1S} \cos(\psi_R + \psi_{BLn})$$

$$\left\{\frac{\partial \tau_{OR}}{\partial \tau_H}\right\} = \left[\begin{array}{c:c} \left[T_{H-R}\right] & B_{H-R} \\ \hdashline 0 & \left[T_{H-R}\right] \end{array}\right]$$

$$B_{H-R} = \left[T_{H-R}\right] \begin{bmatrix} 0 & Z & -Y \\ -Z & 0 & X \\ Y & -X & 0 \end{bmatrix}_{H-R} \triangleq \left[0\right]$$

$$\left\{\frac{\partial \tau_{OH}}{\partial \tau_S}\right\} = \left[\begin{array}{c|ccc} \left[T_{MB \rightarrow H}\right] & & \left[B_{MB-H}\right] & \\ \hline & \cos\psi_s & \theta_s \quad \sin\psi_s & 0 \\ 0 & -\sin\psi_s\cos\theta_s & \cos\psi_s & 0 \\ & \sin\theta_s & 0 & 1 \end{array}\right]$$

$$\left[B_{MB-H}\right] = \left[T_{MB-H}\right] \begin{bmatrix} 0 & Z & -Y \\ -Z & 0 & X \\ Y & -X & 0 \end{bmatrix}_{MB-H}$$

$$\left\{\frac{\partial \tau_{o_H}}{\partial \tau_{o_{MB}}}\right\} = \left[\begin{array}{c|c} \left[T_{MB-H}\right] & \left[B_{MB-H}\right] \\ \hline 0 & \left[T_{MB-H}\right] \end{array}\right]$$

$$\left\{\frac{\partial \tau_{o_{MB}}}{\partial \tau_{REF}}\right\} = \left[\begin{array}{c:c} \left[T_{F \rightarrow MB}\right] & \left[B_{F-MB}\right] \\ \hdashline 0 & \left[T_{F \rightarrow MB}\right] \end{array}\right]$$

$$\left[B_{F-MB}\right] = \left[T_{F-MB}\right] \begin{bmatrix} 0 & Z & -Y \\ -Z & 0 & X \\ Y & -X & 0 \end{bmatrix}_{F-MB}$$

$$\left\{\frac{\partial \tau_{o_H}}{\partial \tau_{REF}}\right\} = \left\{\frac{\partial \tau_{o_H}}{\partial \tau_{o_{MB}}}\right\} \left\{\frac{\partial \tau_{o_{MB}}}{\partial \tau_{REF}}\right\}$$

3.15 AUXOUT

This routine performs auxiliary calculations for output. The quantities computed are stored in the vector, OUTD, which is defined in COMMON and is therefore available for signal definition. Definitions of the currently defined elements of OUTD follow.

Speed in the reference system

$$\text{OUTD}(1) = \left(u^2_{REF} + v^2_{REF} + w^2_{REF}\right)^{1/2} = V_{T_{REF}}$$

Angle of attack

$$\text{OUTD}(2) = \sin^{-1}\left[v_{REF} \Big/ (u^2_{REF} + w^2_{REF})^{1/2}\right] \cdot 57.29578$$

Angle of sideslip

$$\text{OUTD}(3) = \sin^{-1}\left[v_{REF} \Big/ V_{T_{REF}}\right] \cdot 57.29578$$

Flight path angle

$$OUTD(4) = \sin^{-1}\left[-w_E / V_{T_{REF}}\right] \cdot 57.29578$$

$$OUTD(5) = A_{1S} \cdot 57.29578$$

$$OUTD(6) = B_{1S} \cdot 57.29578$$

$$OUTD(7) = w_{I_{MR}}$$

$$OUTD(8) = p_{I_{MR}}$$

$$OUTD(9) = q_{I_{MR}}$$

$$OUTD(10) = w_{I_{TR}}$$

$$OUTD(11) = M_{ZZ_{END}}$$

Attitude gain

$$OUTD(12) = h_e - h_o \text{ (input)}$$

Seven functions at three blade stations on blade 1 are made available primarily for harmonic analysis. The routine will assemble data at the stations closest to 25, 50, and 75 percent blade radius.

The seven functions are

- Span force
- Inplane shear

- Flap shear
- Torsion
- Flap moment
- Inplane moment
- Torsion, aero only

The functions are equivalenced to the OUTD vector, OUTD(13) through OUTD(33).

3.16 SIGGEN

SIGGEN is the signal generator routine. It will collect, up to a maximum of 200, signals from blank COMMON. This is accomplished by a pointer array, SLOC, which is input. Each signal collected can be scaled with a unit converter, one for each signal input. These multipliers are also input, UC.

Arguments List

STIME	Trajectory time
NPTS	Number of turn points output on exit
SIGUNI	Logical unit on which the data is written

3.17 DWASH

Subprogram DWASH calculates main rotor and tail rotor inflow functions.

3.17.1 Main rotor equations

$$V_I = \left(u^2_{HUB} + v^2_{HUB} + \left(w_{HUB} - v_{I_{MR}} \right)^2 \right)^{1/2}$$

$$\dot{v}_{I_{MR}} = \frac{3}{4B \cdot R} \left[- \frac{F_{ZA_{MR,H}}}{2\rho (B \cdot R)^2} - \pi v_{I_{MR}} V_I \right]$$

$$F = \pi V_I \left[1 - v_{I_{MR}} \left(w_{HUB} - v_{I_{MR}} \right) \Big/ V_I^2 \right]$$

$$\dot{p}_{I_{MR}} = \frac{45}{32\ B\ R}\left[-\ \frac{2M_{XA_{MR,H}}}{\rho(B\cdot R)^4} - p_{I_{MR}}\quad F\right]$$

$$\dot{q}_{I_{MR}} = \frac{45}{32\ B\ R}\left[-\ \frac{2M_{YA_{MR,H}}}{(B\ R)^4} - q_{I_{MR}}\quad F\right]$$

The derivatives are integrated locally using a Euler integrator

$$p_{I_{MR}} = p_{I_{MR}} + \dot{p}_{I_{MR}} \cdot dt$$

$$q_{I_{MR}} = \ _{MR} + \dot{q}_{I_{MR}} \cdot dt$$

$$w_{I_{MR}} = w_{I_{MR}} + \dot{w}_{I_{MR}} \cdot dt$$

3.17.2 Tail rotor equations. - The tail rotor functions are handled by first order log equations.

Inflow

$$w_{I_{TR}}^{n+1} = w_{I_{TR}}^{n} \quad e^{-dt/\tau} + w_{I_{TR_N}}^{n} \quad \left(1 - e^{-dt/\tau}\right)$$

Longitudinal Flap Angle

$$A_{1_{TR}}^{n+1} = A_{1_{TR}}^{n} \quad e^{-dt/\tau} + A_{1_{TR_N}}^{n} \quad \left(1 - e^{-dt/\tau}\right)$$

The superscript n, stands for iteration number. The tail rotor functions $W_{I_{TR_N}}$ and $A_{1_{TR_N}}$ are computed in subprogram LOADS.

3.18 TORS

Quasi-static torsion is optionally computed by subprogram TORS. The function evaluated here is described in detail in Section 5.6.5 of Volume I.

3.19 PHORN

Quasi-static pitch horn bending is optionally computed by subprogram PHORN. The feathering angle increment due to pitch horn bending is determined by a first order log equation for each blade

$$\overset{n+1}{\phi_{f_{Ph}}} = \overset{n}{\phi_{f_{Ph}}} e^{-dt/\tau} + \overset{n}{\phi_{f_{Ph_1}}} \left(1 - e^{-dt/\tau}\right)$$

$$\overset{.n+1}{\phi_{f_{Ph}}} = \overset{.n}{\phi_{f_{Ph}}} e^{-dt/\tau} + \left(\overset{n}{\phi_{f_{Ph_1}}} - \overset{n-1}{\phi_{f_{Ph_1}}}\right) \Big/ \tau$$

where

$$\overset{n}{\phi_{f_{Ph_1}}} = M_{\phi_f} / K_{Ph}$$

The reader can refer to Section 5.6.6 of Volume I.

3.20 ETORQ

An engine can be modeled as described in Section 5.10 of Volume I. The engine equations are evaluated within subprogram ETORQ with the primary output being $M_{XA_{ENG}}$. The pertinent equations are

$$\dot{\psi}^{n}_{GEN} = \dot{\psi}^{n-1}_{GEN} + \left[K_{R1} \ddot{\psi}_R + K_{R2} (\dot{\psi}_{ENG} - \dot{\psi}_{ENG,TRIM}) - \dot{\psi}^{n-1}_{GEN} \right] \frac{dt}{\tau}$$

$$M_{XA_{ENG}} = M_{XA_{ENG,TRIM}} + \frac{\partial M_{ENG}}{\partial \dot{\psi}_{GEN}} \dot{\psi}_{GEN} - \frac{\partial M_{ENG}}{\partial \dot{\psi}_{ENG}} (\dot{\psi}_{ENG} - \dot{\psi}_{ENG,TRIM})$$

3.21 INTGGC

This subprogram provides the integration of the generalized coordinates. The integrator is an Adams-Bashforth 4-point open relation. An accumulation of back value information is required. Therefore, a Euler starter is used. It can be seen in Figure 16 that accelerations and velocities are separately integrated. This is for bookkeeping purposes.

A description of calling sequence arguments follow.

NE	Number of integrators to use
DT	Integration interval, sec
IC	Step counter. It will be increased by one on return.
Y	Displacement vector. Will contain updated values on return.
YD	Velocity vector. Will contain updated values on return.
YDD	Acceleration vector.

3.22 INTGAX

Subprogram INTGAX provides an integration mechanism for all auxiliary integrals. It is identical in design and form with INTGGC except that only one set of integrals are applied.

$$Y_i = \int^t Y_i \, dt + Y_{i/IC}$$

3.23 JETTB

Subprogram JETTB is designed to provide a main rotor blade jettison sequence.

This subprogram is special purpose in that it is only for a 5-bladed system with a specific jettison sequence. A blade is jettisoned by removing it from the generalized coordinate set, i.e., all generalized coordinates are redefined. The generalized coordinate integrator back values are also shifted so that integration is not interrupted. The rotor-to-blade transformations are also shifted, and are not redefined. Other blade dependent data affected is the quasi-static pitch horn and quasi-static torsion.

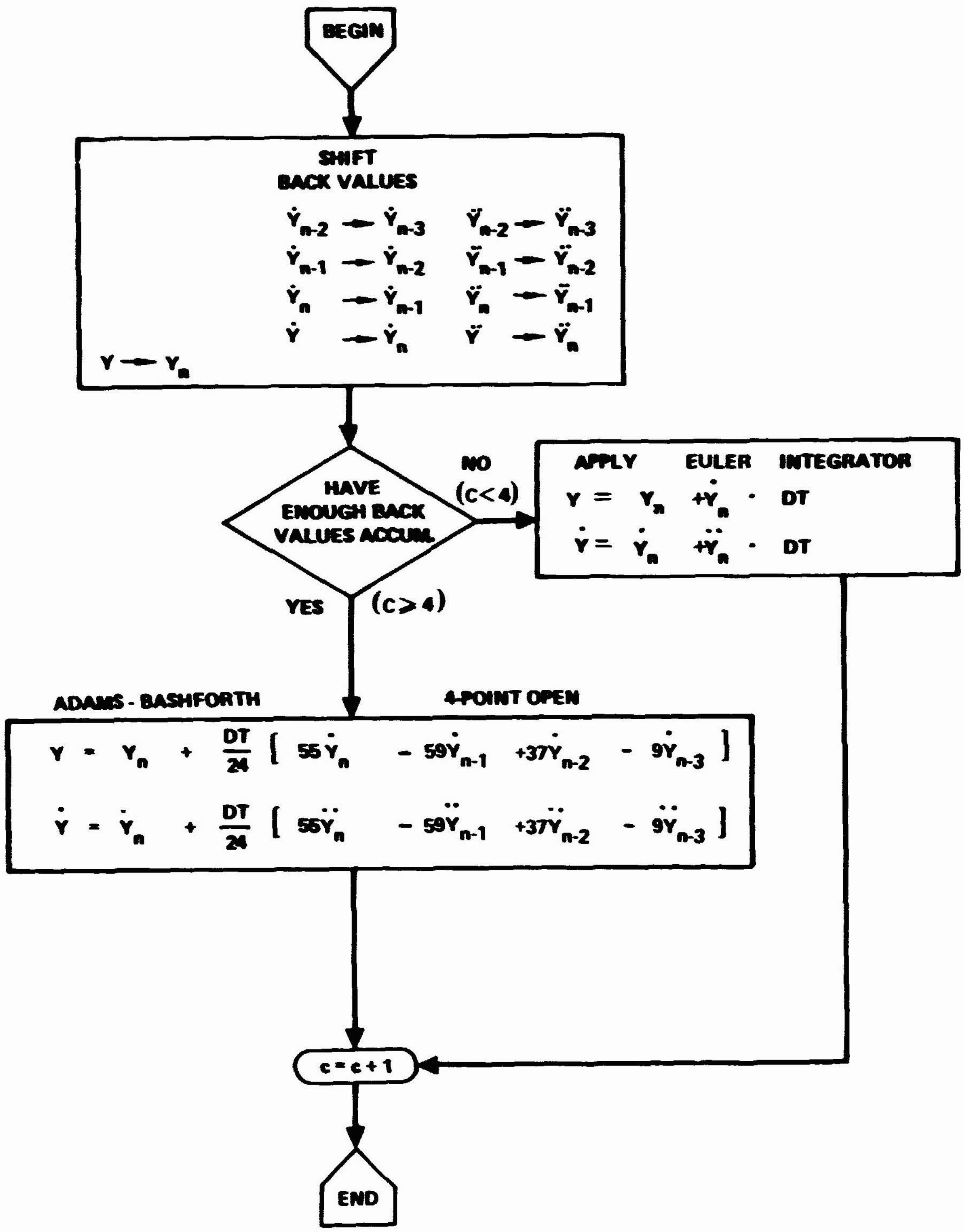


Figure 16. - INTGGC flow diagram.

Operationally, JETTB is activated from FLY at the beginning of a prescribed rotor cycle (JETCYC) measured from trim. Upon this first entry into JETTB, main rotor blades 1, 3, and 4 are removed from the generalized coordinate set as described above. Upon exit, the number of blades (NB) has been reset to 2. At a prescribed number of integration steps later, n = NAZ/5, JETTB is reentered and the remaining blades are jettisoned. At that time quasi-static pitch horn and quasi-static torsion will be turned off. Also, all swashplate coordinates will be turned off.

The user is reminded that some auxiliary outputs are a function of blade 1 data. During a jettison sequence, blade 1 is redefined. Thus, mixed data may be processed.

4. ACCEL - LEVEL 3 EXECUTIVE

The level 3 hierarchical chart for ACCEL is presented in Figure 17. ACCEL, as a control program, is simple and straight forward. It directs the calculation of the components of the force-mass-acceleration equation and evaluates the recursion relation

$$\ddot{q}^{t} = \Delta\ddot{q}^{t} + \ddot{q}^{t-\Delta t}$$

which is fundamental to development of the REXOR II model. Further details concerning the subprogram ACCEL are presented in the next section. The functions GFORCE and MASSSP depicted in bold-face in Figure 17 are treated later as level 4 executives. GMASS is discussed in this section and the matrix routines CHOSKY and MXOUT are discussed in Section 7.

4.1 ACCEL

The subprogram ACCEL computes the generalized acceleration at time, t, by applying the recursion relation

$$\ddot{q}^{t} = \ddot{q}^{t-\Delta t} + \Delta\ddot{q}^{t}$$

The incremental acceleration is provided by solving the linear equation

$$\left[M\right]\left\{\Delta\ddot{q}\right\} = \left\{\Delta F\right\}$$

The elements of this equation, the mass matrix and the force vector, are provided by subordinate routines.

The variables in the argument list are:

CASEID 8 character case title for output purposes

MCOMP Logical variable which controls the computation of the mass matrix

DISPMX A logical variable used to control the printing of the mass matrix

IER Program error indicator

The details of ACCEL are depicted in Figure 18.

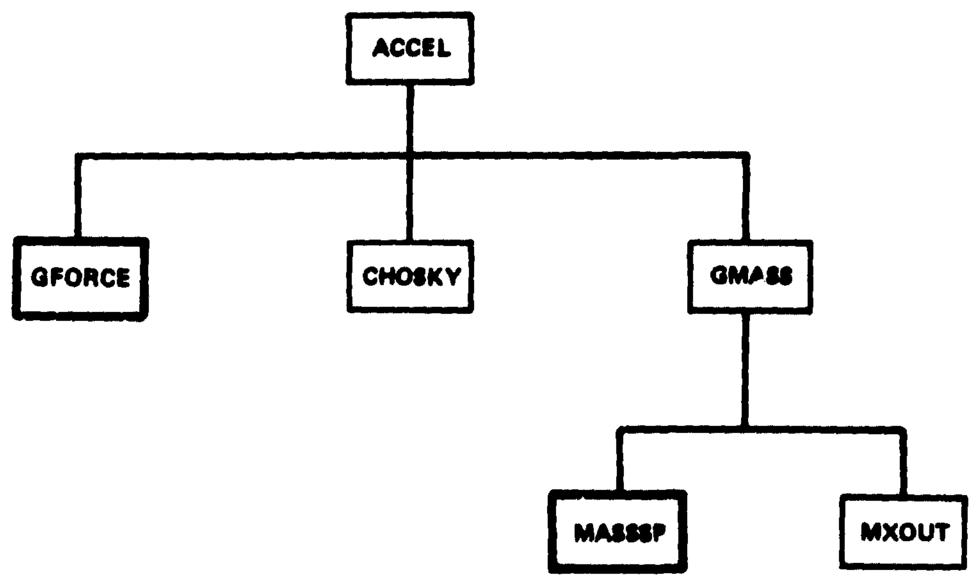


Figure 17. - ACCEL level 3 hierarchical chart

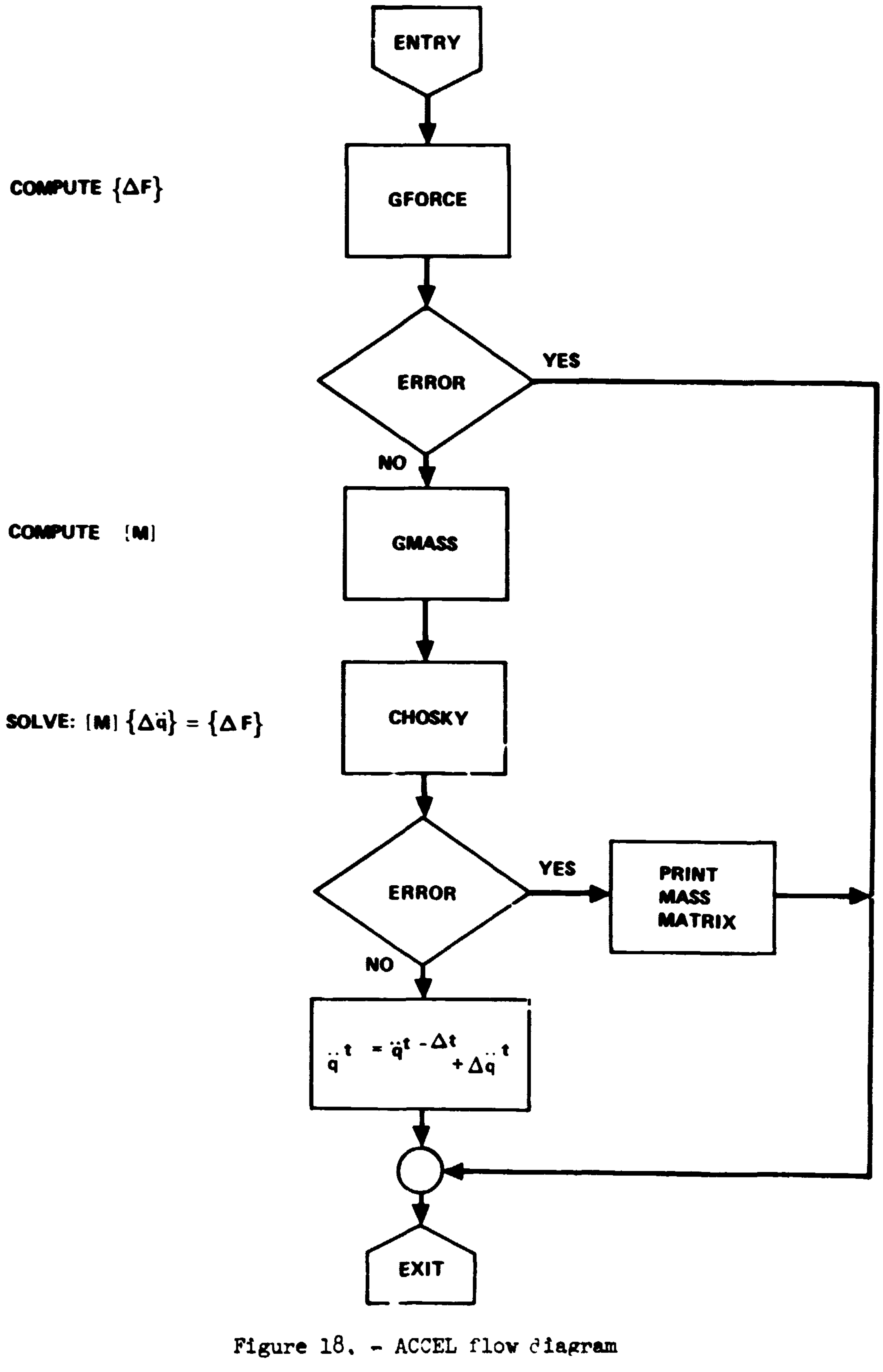


Figure 18. - ACCEL flow diagram

4.2 GMASS

Subprogram GMASS provides three functions. It determines whether a mass matrix is to be computed and calls MASSSP if required. It shifts the mass matrix to working storage. This is necessary because the coefficient matrix is destroyed on exit from CHOSKY. Finally, GMASS will list the mass matrix if so directed. All of these functions are depicted in Figure 19.

The system solution routine, CHOSKY, operates on double precision data in the I.B.M. software. Double precision is not required in C.D.C. software because of the extended word size.

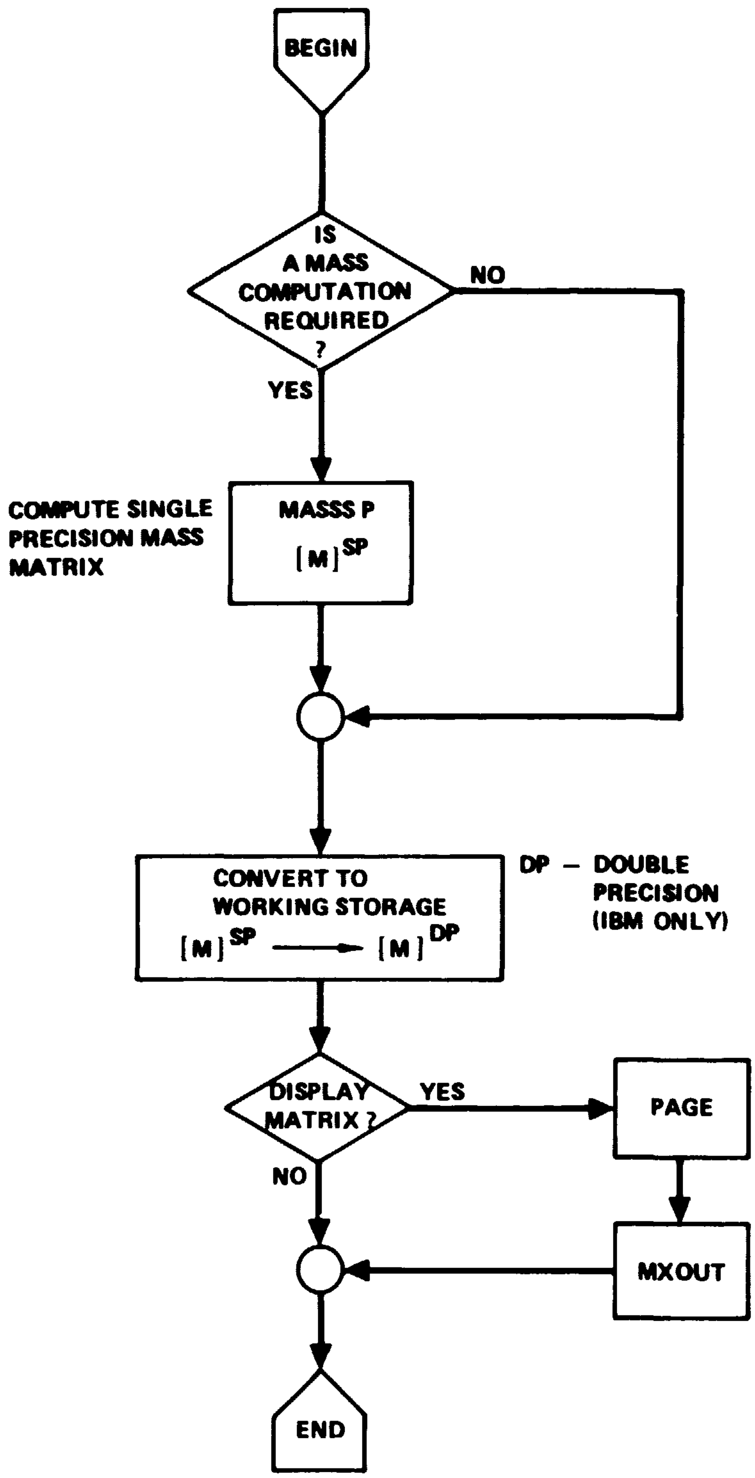


Figure 19. - GMASS flow diagram

5. GFORCE - LEVEL 4 EXECUTIVE

GFORCE is a fourth level executive control program. Its primary output is the generalized force vector, ΔF, which is defined in COMMON as DELF. The elements of GFORCE are depicted in Figure 20. GFORCE and its subordinate routines are presented in this chapter.

5.1 GFORCE

This subprogram assembles the generalized force vector by optionally computing the required subsystem force vectors. This is best depicted in Figure 21. The assembled force vector would look like

$$\underset{NTOTL \times 1}{\{\Delta F\}} = \left\{ \begin{matrix} \{\Delta Q\}_{BS} \\ \cdots\cdots \\ \{\Delta Q\}_{SP} \\ \cdots\cdots \\ \{\Delta Q\}_{S} \\ \cdots\cdots \\ \{\Delta Q\}_{REF} \\ \cdots\cdots \\ \{\Delta Q\}_{R} \end{matrix} \right\} \quad \begin{matrix} NBS \times 1 \\ \\ NSP \times 1 \\ \\ NS \times 1 \\ \\ NREF \times 1 \\ \\ NR \times 1 \end{matrix}$$

if all subsystems were triggered.

Subprogram GFORCE has only one argument, IER. If an error is detected in SWEEP, IER is set to 999 and control returns to ACCEL.

5.2 Main Rotor Analysis

Subprogram SWEEP and its initialization entry SWEEP1, computer main rotor blade forces and generalized mass elements. To this end blade element motions, partials, and aerodynamic forces are also computed. This is the major program computation sequence, and will only be outlined here. References to Volume I will be made where required. Other subordinate routines are presented in this section also.

5.2.1 <u>SWEEP1, SWEEP</u>. - The chief output of SWEEP is the blade root forces and generalized mass integrals. The forces can be presented as

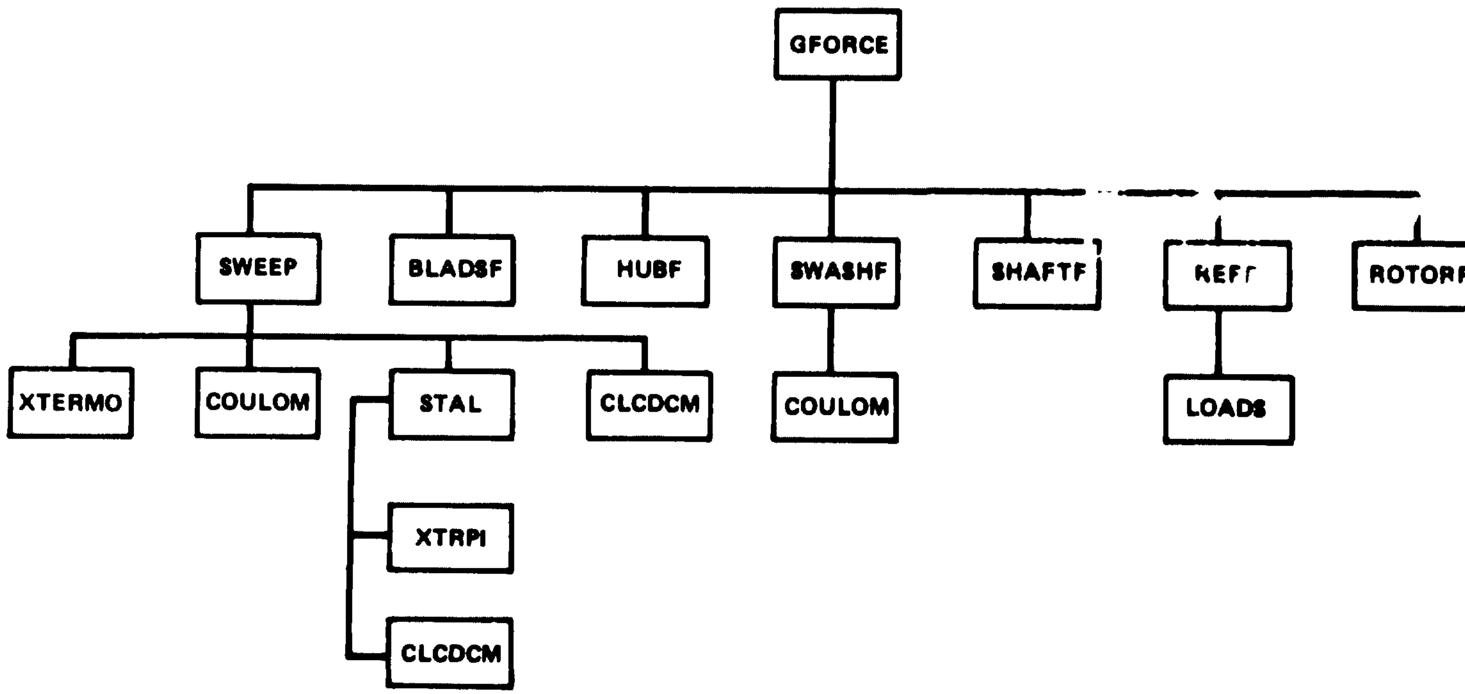


Figure 20. - GFORCE Level 4 hierarchical chart.

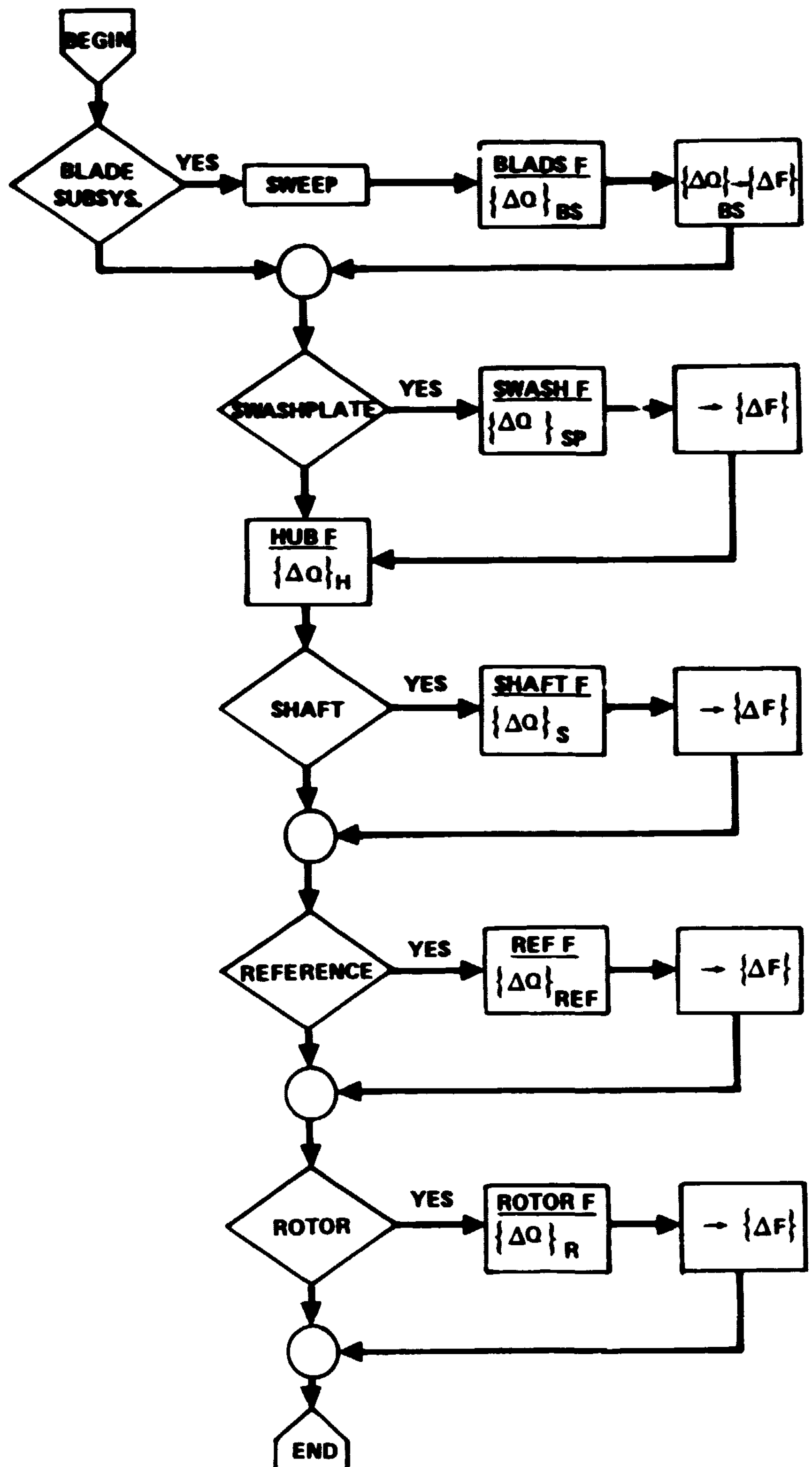


Figure 21. - GFORCE flow diagram.

$$\{\Delta F\}_{BL} = \begin{Bmatrix} \Delta F_{A_1} \\ \vdots \\ \Delta F_{A_{N_{mb}}} \\ \Delta F_{\Phi_T} \\ \Delta F_{\Phi_f} \\ \Delta F_x \\ \Delta F_y \\ \Delta F_z \\ \Delta F_{\phi_{BL}} \\ \Delta F_{\theta_{BL}} \\ \Delta F_{\psi_{BL}} \end{Bmatrix} = \int^{K} \left[P^k\right]^T \left[\Delta F_{BLE}^{\ k}\right] ds$$

where the integration is over all blade stations, k. The above set of integrals are generated for each blade n=1, ---, N_B.

Further, the mass integrals can be presented as

$$\left[M\right] = \int^{K} \left\{ m_{BLE}^{(k)} \left[P_1^{\ k}\right]^T \left[P_1^{\ k}\right] + I_{XX_{BLE}}^{(k)} \left[P_2^{\ k}\right]^T \left[P_2^{\ k}\right] \right\} ds$$

where the integration is over the blade stations and for each blade. Also, the matrix $[M]$ is symmetric.

The $[P_i]$ matrices are subsets of the partial derivative matrix defined below. For each blade and station

$$[P] = \left[\begin{array}{c:c:c:c:c} \left[\dfrac{\partial r_{BLE}}{A_{N_{mb}}}\right] & \left[\dfrac{\partial r_{BLE}}{\partial \phi_T}\right] & \left[\dfrac{\partial r_{BLE}}{\partial \phi_f}\right] & \left[I\right] & \left[\dfrac{\partial r_{BLE}}{\partial \xi_{BL}}\right] \\ \hdashline \left[\dfrac{\partial \xi_{BLE}}{\partial A_{N_{mb}}}\right] & \dfrac{\partial \phi_{BLE}}{\partial \phi_T} & \left[\dfrac{\partial \xi_{BLE}}{\partial \phi_f}\right] & \left[0\right] & \left[\dfrac{\partial \xi_{BLE}}{\partial \xi_{BL}}\right] \end{array}\right]$$

Identify the first four rows of the partial derivative matrix as $\left[P_3\right]$

$$\left[P_3\right] = \left[\begin{array}{c} \left[P_1\right] \\ \hdashline \left[P_2\right] \end{array}\right] = \left[\begin{array}{ccccc} \left[\dfrac{\partial r_{BLE}}{\partial A_{N_{mb}}}\right] & \left[\dfrac{\partial r_{BLE}}{\partial \phi_T}\right] & \left[\dfrac{\partial r_{BLE}}{\partial \phi_f}\right] & \left[I\right] & \left[\dfrac{\partial r_{BLE}}{\partial \xi_{BL}}\right] \\ \hdashline \left[\dfrac{\partial \phi_{BLE}}{\partial A_{N_{mb}}}\right] & \left[\dfrac{\partial \phi_{BLE}}{\partial \phi_T}\right] & \left[\dfrac{\partial \phi_{BLE}}{\partial \phi_f}\right] & \left[0\right] & \left[\dfrac{\partial \phi_{BLE}}{\partial \xi_{BL}}\right] \end{array}\right]$$

also define the blade forces

$$\left\{\Delta F_{BLE}{}^{(k)}\right\}_{BL_n} = \left\{\begin{array}{l} F_{X_{A_{BLE_{BL}}}}{}^{(k)} - m_{BLE}{}^{(k)}\ \ddot{x}_{BLE_{BL}}{}^{(k)} \\ F_{Y_{A_{BLE_n}}}{}^{(k)} - m_{BLE}{}^{(k)}\ \ddot{y}_{BLE_{BL}}{}^{(k)} \\ F_{Z_{A_{BLE_{SL}}}}{}^{(k)} - m_{BLE}{}^{(k)}\ z_{BLE_{BL}}{}^{(k)} \\ M_{XX_{BLE_{BL}}}{}^{(k)} \\ 0 \\ M_{ZZ_{BLE_{BL}}}{}^{(k)} \end{array}\right\}$$

The evaluation of the above partials and forces require the evaluation of the blade element motions and aerodynamic forces

$$\left\{{}^{\tau}O_{BLE}\right\}, \left\{{}^{\dot{\tau}}O_{BLE}\right\}, \left\{{}^{\ddot{\tau}}O_{BLE}\right\}, \left\{{}^{F}A_{BLE}\right\}$$

Blade element motions are treated in detail in Section 4.5.5 of Volume I. Those equations will not be repeated. Rather, program names for certain key quantities will be presented.

$$\left\{r_{BLE}\right\}_{BL_n} = \begin{matrix} X \\ Y \\ Z \end{matrix}$$

$$\left\{\dot{r}_{BLE}\right\}_{BL_n} = \begin{matrix} XD \\ YD \\ ZD \end{matrix}$$

$$\left\{\ddot{r}_{BLE}\right\}_{BL_n} = \begin{matrix} XDD \\ YDD \\ ZDD \end{matrix}$$

Blade element freestream absolute acceleration

$$\left\{\begin{matrix} \ddot{X}_{BLE} \\ \ddot{Y}_{BLE} \\ \ddot{Z}_{BLE} \end{matrix}\right\}^{I}_{BL_n} = \begin{matrix} DDXN \\ DDYN \\ DDZN \end{matrix}$$

Blade element angular velocities and accelerations

$$\left\{\begin{matrix} p \\ q \\ r \end{matrix}\right\}_{BLE} = ANGBL(3)$$

$$\left\{\begin{matrix} \dot{p} \\ q \\ \dot{r} \end{matrix}\right\}_{BLE} = ANGBLD(3)$$

Partial derivative matrices are identified as

$$\left[\frac{\partial r_{BLE}}{\partial A_{N_{mb}}}\right] = \text{PTAUBA(3,3)}$$

$$\left[\frac{\partial \xi_{BLE}}{\partial A_{N_{mb}}}\right] = \text{PZETBA(3,3)}$$

$$\left[\frac{\partial r_{BLE}}{\partial \xi_{BL}}\right] = \text{PRBLZ(3,3)}$$

$$\left[\frac{\partial \xi_{BLE}}{\partial \xi_{BL}}\right] = \left[T_{BL \rightarrow BLE}\right] = \left[T_{BLE \rightarrow BL}\right]^T$$

where

$$\left[T_{BLE \rightarrow BL}\right] = \text{TBLR(3,3)}$$

$$\left[\frac{\partial r_{BLE}}{\partial \phi_f}\right] = \text{PRZBF(3)}$$

$$\left[\frac{\partial \xi_{BLE}}{\partial \phi_f}\right] = \text{PZETBF(3)} = [T49]^T [TYB] [T51] \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$$

$$\left[\frac{\partial r_{BLE}}{\partial \phi_T}\right] = \text{PRZBP(3)}$$

$$\left[\frac{\partial \phi_{BLE}}{\partial \phi_T}\right] = \text{PFBLP}$$

The aerodynamic force vector is

$$\left\{F_{A_{BLE}}\right\} = \begin{Bmatrix} 0 \\ FC \\ FNO \end{Bmatrix}$$

Mode shape data is available in array form

$$\begin{bmatrix} Y_1 & Y_2 & Y_3 \\ Z_1 & Z_2 & Z_3 \end{bmatrix}_k$$

where the array dimension definition is

(2, 3, 20)   (COORD, MODE, STATION)

MCOEF (2, 3, 20)
MCOEFP (2, 3, 20)   Prime values (derivatives)

FBIB (2, 3)   Feather bearing inboard
FBOB (2, 3)   Feather bearing outboard

5.2.1.1 Rotor Blade Spatial Integration. - All force and mass integrals over the blade are computed from the inboard to the outboard stations. The numerical algorithm used is trapezoidal integration. Let FI stand for the function (integrand) to be integrated and F for the integral.

Then the algorithm proceeds as follows.

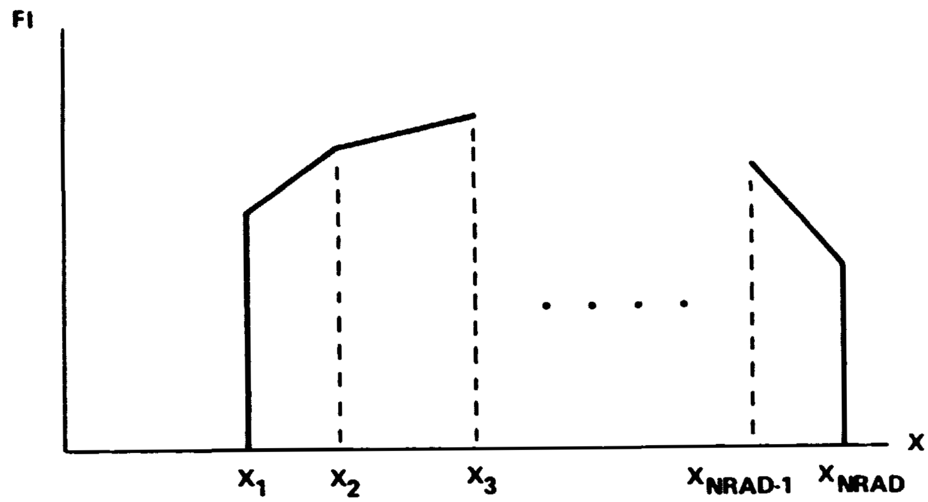
- Evaluate FI^1 i.e. $FI(X_1)$
- Set F^1 i.e. $F(X_1) = 0$
- Integrate from k=2, ---, N_{STA}

$$F^k = F^{k-1} + \left(\frac{x^k - x^{k-1}}{2}\right)\left[FI^k + FI^{k-1}\right]$$

This process is pictured in Figure 22.

The integral equation presented above for force and mass terms are straight. However, the current subprogram treats each function separately. Therefore the correspondence between program symbol and function is presented. There are currently 62 functions defined for each blade.

The array size is F(66, 7). the integrands for each function are defined in the FI array. The definitions of F for each blade are



$$X_{NRAD-1} = R - (2B - 1)$$

where

R = blade radius
B = tip loss factor

Figure 22. - SWEEP Integration.

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix} = \begin{Bmatrix} \Delta F_{A_{N_{mb}}} \end{Bmatrix}_n$$

$$\begin{Bmatrix} F_4 \\ F_5 \\ F_6 \end{Bmatrix} = \begin{Bmatrix} F_{XO_{BL}} \\ F_{YO_{BL}} \\ F_{ZO_{BL}} \end{Bmatrix}_n$$

$$\begin{Bmatrix} F_7 \\ F_8 \\ F_9 \end{Bmatrix} = \begin{Bmatrix} M_{X_{BL}} \\ M_{Y_{BL}} \\ M_{Z_{BL}} \end{Bmatrix}_n$$

$$\begin{Bmatrix} F_{10} \\ F_{11} \\ F_{12} \end{Bmatrix} = \begin{Bmatrix} F_{X_{OA_{BL}}} \\ F_{Y_{OA_{BL}}} \\ F_{Z_{OA_{BL}}} \end{Bmatrix}_n \text{ aerodynamic loading only}$$

$$\begin{Bmatrix} F_{13} \\ F_{14} \\ F_{15} \end{Bmatrix} = \begin{Bmatrix} M_{X_{OA_{BL}}} \\ M_{Y_{OA_{BL}}} \\ M_{Z_{OA_{BL}}} \end{Bmatrix}_n \text{ aerodynamic loading only}$$

$$F_{16} = \Delta F_{\phi_f\, n}$$

$$F_{17} = \Delta F_{\phi_T\, n}$$

$$F_{18} = M_{X_{BLE\, n}}$$

Mass Integrals follow.

$$F_{19} = M_{\phi_f \phi_f}$$

$$F_{20} = M_{\phi_f A_1}$$

$$F_{21} = M_{\phi_f A_2}$$

$$F_{22} = M_{\phi_f A_3}$$

$$F_{23} = M_{A_1 A_1}$$

$$F_{24} = M_{A_2 A_1}$$

$$F_{25} = M_{A_2 A_2}$$

Continuing Mass Integrals:

$$F_{26} = M_{A_3A_1}$$

$$F_{27} = M_{A_3A_2}$$

$$F_{28} = M_{A_3A_3}$$

$$F_{29} = M_{\psi_{BLA_1}}$$

$$F_{30} = M_{\psi_{BLA_2}}$$

$$F_{31} = M_{\psi_{BLA_3}}$$

$$F_{32} = M_{\psi_{BL\ f}}$$

$$F_{33} = M_{X_{BLA_1}}$$

$$F_{34} = M_{X_{BLA_2}}$$

$$F_{35} = M_{X_{BLA_3}}$$

$$F_{36} = M_{Y_{BLA_1}}$$

$$F_{37} = M_{Y_{BLA_2}}$$

$$F_{38} = M_{Y_{BLA_3}}$$

$$F_{39} = M_{Y_{BL\phi_f}}$$

$$F_{40} = M_{Z_{BLA_1}}$$

$$F_{41} = M_{Z_{BLA_2}}$$

$$F_{42} = M_{Z_{BLA_3}}$$

$$F_{43} = M_{Z_{BL\phi_f}}$$

$$F_{44} = M_{\phi_{BL}A_1}$$

$$F_{45} = M_{\phi_{BL}A_2}$$

$$F_{46} = M_{\phi_{BL}A_3}$$

$$F_{47} = M_{\phi_{BL}\phi_f}$$

$$F_{48} = M_{\theta_{BLA_1}}$$

$$F_{49} = M_{\theta_{BLA_2}}$$

$$F_{50} = M_{\theta_{BLA_3}}$$

$$F_{51} = M_{\theta_{BL}\phi_f}$$

$$F_{52} = M_{\phi_{BL}\psi_{BL}}$$

$$F_{53} = M_{\theta_{BL}\psi_{BL}}$$

$$F_{54} = M_{Y_{BL}\psi_{BL}}$$

$$F_{55} = M_{Z_{BL}\phi_{BL}}$$

$$F_{56} = M_{X_{BL}\theta_{BL}}$$

$$F_{57} = M_{\phi_T A_1}$$

$$F_{58} = M_{\phi_T A_2}$$

$$F_{59} = M_{\phi_T A_3}$$

$$F_{60} = M_{\phi_T \phi_T}$$

$$F_{61} = M_{\phi_T \phi_f}$$

$$F_{62} = M_{X_{BL}\phi_f}$$

5.2.2 XTERMO. - This is a specialized matrix routine, subordinate to SWEEP. Within SWEEP, the equations for

$$\left\{r_{BLE}\right\}_{BL_n}, \left\{\dot{r}_{BLE}\right\}_{BL_n}, \text{ and } \left\{\ddot{r}_{BLE}\right\}_{BL_n}$$

require the assembly of complex matrix terms which are made up of transforms of three angles Z'_{FA}, Y'_{FA}, and $\Delta\phi_f$. Subprogram XTERMO and its multiple entries provide the calculation of these terms. The program code has been computer generated by an application of IBM's FORMAC program where the number of calculations is minimized. The subprogram entries are described below.

<u>XTERM0</u> - an initialization entry where trignometric terms of the three transformation angles are computed.

The three basic transformation are, in argument order; $(Z'_{FA}, Y'_{FA}, \Delta\phi_f)$

$$[T_{Z'FA}] = \begin{bmatrix} \cos Z'_{FA} & 0 & \sin Z'_{FA} \\ 0 & 1 & 0 \\ -\sin Z'_{FA} & 0 & \cos Z'_{FA} \end{bmatrix}$$

$$[T_{Y'FA}] = \begin{bmatrix} \cos Y'_{FA} & \sin Y'_{FA} & 0 \\ -\sin Y'_{FA} & \cos Y'_{FA} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[T_{\Delta\phi_f}] = \begin{bmatrix} 1 & 0 & \\ 0 & \cos \Delta\phi_f & \sin \Delta\phi_f \\ 0 & -\sin \Delta\phi_f & \cos \Delta\phi_f \end{bmatrix}$$

The transforms computed on the various entries are

<u>XTERM1</u>

$$[T] = [T'_{ZFA}]^T \, [T'_{YFA}]^T \, [T_{\Delta\phi_f}]^T \, [T'_{YFA}] \, [T'_{ZFA}]$$

<u>XTERM2</u>

$$[T] = [T'_{ZFA}]^T \, [T'_{YFA}]^T \, [\dot{T}_{\Delta\phi_f}]^T \, [T'_{YFA}] \, [T'_{ZFA}]$$

The dot over the matrix indicates differentiation with respect to the angle.

XTERM3

$$[T] = [T'_{ZFA}]^T [T'_{YFA}]^T [T_{\Delta\phi_f}]^T [\ddot{T}'_{YFA}] [T'_{ZFA}]$$

XTERM4

$$[T] = [\dot{T}'_{ZFA}]^T [T'_{YFA}]^T [T_{\Delta\phi_f}]^T [T'_{YFA}] [T'_{ZFA}]$$

$$+ [T'_{ZFA}]^T [T'_{YFA}]^T [T_{\Delta\phi_f}]^T [T'_{YFA}] [\dot{T}'_{ZFA}]$$

XTERM5

$$[T] = [T'_{ZFA}]^T [\dot{T}'_{YFA}]^T [T_{\Delta\phi_f}]^T [T'_{YFA}] [T'_{ZFA}]$$

$$+ [T'_{ZFA}]^T [T'_{YFA}]^T [T_{\Delta\phi_f}]^T [\dot{T}'_{YFA}] [T'_{ZFA}]$$

XTERM6

$$[T] = [T'_{ZFA}]^T [\dot{T}'_{YFA}]^T [\dot{T}_{\Delta\phi_f}]^T [T'_{YFA}] [T'_{ZFA}]$$

$$+ [T'_{ZFA}]^T [T'_{YFA}]^T [\dot{T}_{\Delta\phi_f}]^T [\dot{T}'_{YFA}] [T'_{ZFA}]$$

XTERM7

$$[T] = [\ddot{T}'_{ZFA}]^T [T'_{YFA}]^T [T_{\Delta\phi_f}]^T [T'_{YFA}] [T'_{ZFA}]$$

$$+ 2 [\dot{T}'_{ZFA}]^T [T'_{YFA}]^T [T_{\Delta\phi_f}]^T [T'_{YFA}] [\dot{T}'_{ZFA}]$$

$$+ [T'_{ZFA}]^T [T'_{YFA}]^T [T_{\Delta\phi_f}]^T [T'_{YFA}] [\ddot{T}'_{ZFA}]$$

XTERM8

$$[T] = [\dot{T}'_{ZFA}]^T [T'_{YFA}]^T [\dot{T}_{\Delta\phi_f}]^T [T'_{YFA}] [T'_{ZFA}]$$

$$+ [T'_{ZFA}]^T [T'_{YFA}]^T [\dot{T}_{\Delta\phi_f}]^T [T'_{YFA}] [\dot{T}'_{ZFA}]$$

XTERM9

$$[T] = [T'_{ZFA}]^T [\ddot{T}'_{YFA}]^T [T_{\Delta\phi_f}]^T [T'_{YFA}] [T'_{ZFA}]$$

$$+ 2 [T'_{ZFA}]^T [\dot{T}'_{YFA}]^T [T_{\Delta\phi_f}]^T [\dot{T}'_{YFA}] [T'_{ZFA}]$$

$$+ [T'_{ZFA}]^T [T'_{YFA}]^T [T_{\Delta\phi_f}]^T [\ddot{T}'_{YFA}] [T'_{ZFA}]$$

XTERMA

$$[T] = [\dot{T}'_{ZFA}]^T [\dot{T}'_{YFA}]^T [T_{\Delta\phi_f}]^T [T'_{YFA}] [T'_{ZFA}]$$

$$+ [\dot{T}'_{ZFA}]^T [T'_{YFA}]^T [T_{\Delta\phi_f}]^T [\dot{T}'_{YFA}] [T'_{ZFA}]$$

$$+ [T'_{ZFA}]^T [\dot{T}'_{YFA}]^T [T_{\Delta\phi_f}]^T [T'_{YFA}] [\dot{T}'_{ZFA}]$$

$$+ [T'_{ZFA}]^T [T'_{YFA}]^T [T_{\Delta\phi_f}]^T [\dot{T}'_{YFA}] [\dot{T}'_{ZFA}]$$

XTERMC

$$[T] = [T_{\Delta\phi_f}] [T'_{YFA}] [T'_{ZFA}]$$

XTERMD

$$[T] = [\dot{T}_{\Delta\phi_f}] [T'_{YFA}] [T'_{ZFA}]$$

XTERME

$$[T] = [T'_{ZFA}]^T \, [T'_{YFA}]^T$$

5.2.3 COULOM. - Coulomb friction is calculated by subprogram COULOM. COULOM is used for swashplate and feather bearing friction.

The output function is of the form shown in Figure 23.

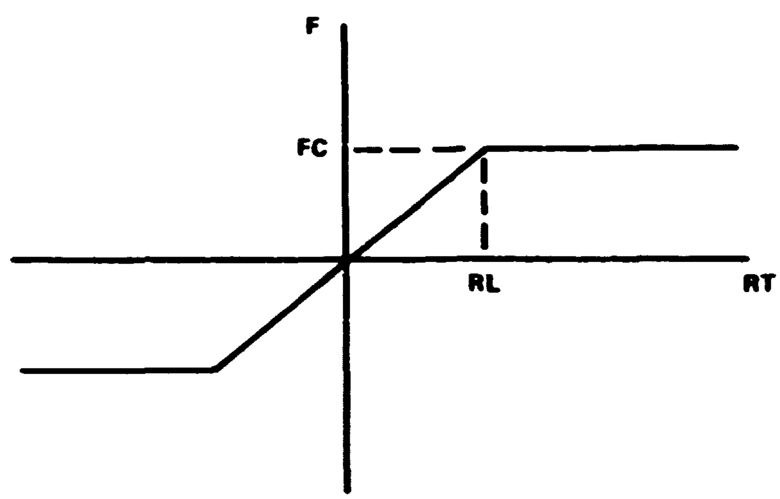


Figure 23. - Coulomb friction.

The routine is executed by

CALL COULOM (F, FC, RL, RT)

F - output friction

FC - friction limit

RT - argument such of $\dot{\phi}_f$

RL - argument limit

5.2.4 Main rotor blade aerodynamics. - The following three routines, CLCDCM, STALL, and XTRP1 provide main rotor aerodynamic coefficient calculations.

5.2.4.1 CLCDCM. - This routine calculates the aerodynamic coefficients

$$C_L = C_L(\alpha, M)$$

$$C_D = C_D(\alpha, M)$$

$$C_M = C_M(\alpha_M, M)$$

by linear interpolation. Up to two sets of data tables are available in labeled COMMON/TAB/. This routine is derived from a similar routine in the Army's C-81 computer program.

Arguments List

IDT	Data table set number. 1 or 2
ALPHLD	Angle of attack for CL and CD curves. Units are degrees
ALPHCM	Angle of attack for CM tables
MACH	Mach number
CL	Output coefficients
CD	Output coefficients
CM	Output coefficients

5.2.4.2 STALL. - Subprogram STALL applies the dynamic stall model as presented in section 6.2.3.4 of Volume I. These equations are distilled in Figure 24.

Arguments List

ISET	Aero table set number
A	Angle of attack, rad.
M	Mach number
AYAW	Λ radial flow angle, $Tan^{-1}(U_S/U_C)$
K	Blade element reduced frequency $K - C\dot{\alpha}/2V$

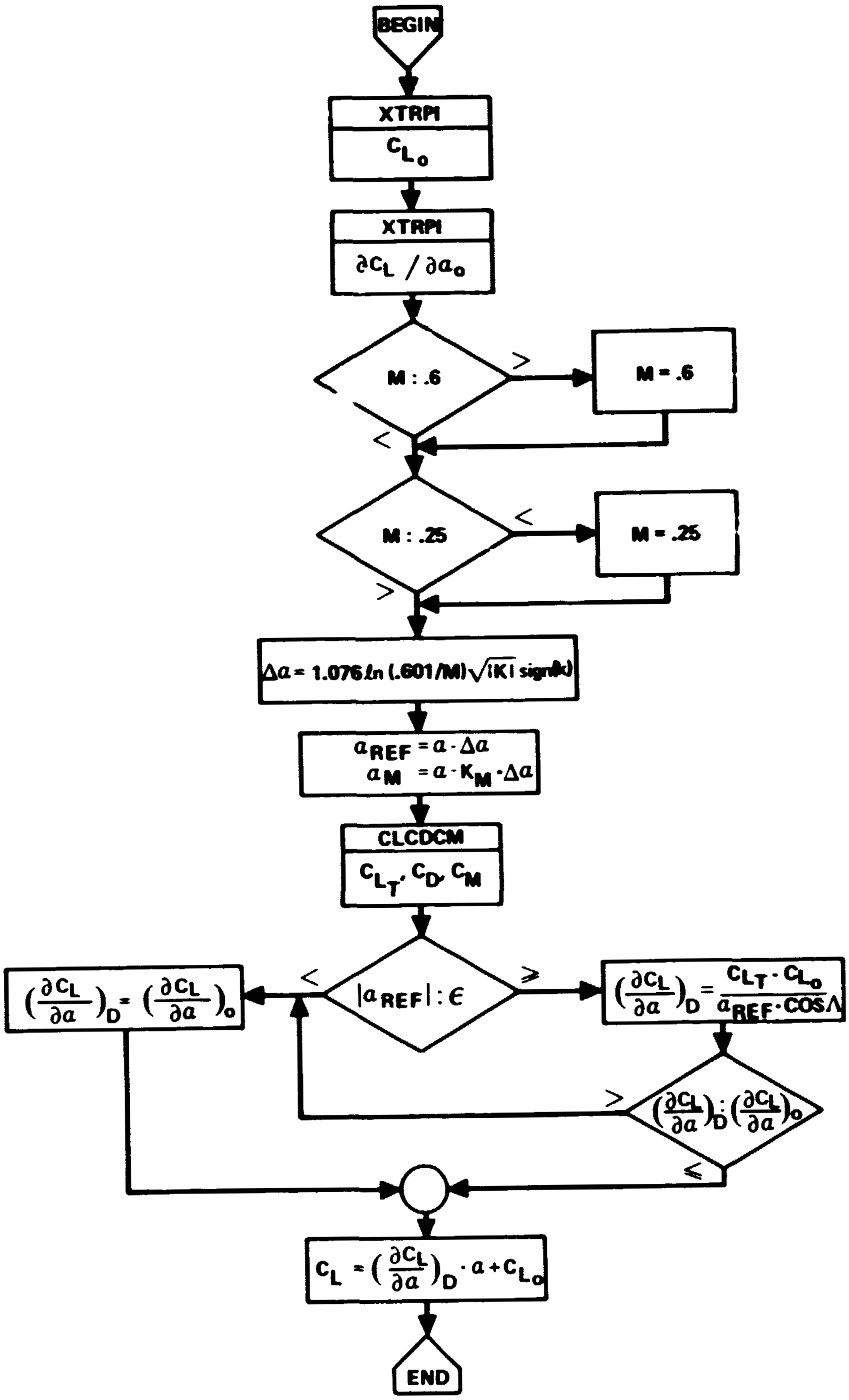


Figure 24. - STALL flow diagram.

Arguments List

CLR, CDR, CMR — Output aero coefficients

5.2.4.3 XTRP1. - This is a special linear interpolation routine designed for speed. The interpolation technique deos not require the classic table search to find the data of interest. This routine does require that the tabulated function be evaluated at constant argument intervals.

Algorithm - If x_o is the first argument value and the data is evaluated at Δx intervals, then the location within the table of bracketing function values for a value of the argument x is n where

$$n = \left[\frac{x - x_o}{\Delta x} \right]$$

The brackets indicate the largest integer whose magnitude does not exceed the magnitude of $(x - x_o)/\Delta x$.

The interpolating formula, based on the point-slope formula, is given as

$$y = y_n + dx \cdot (y_{n+1} - y_n)$$

where

$$dx = \frac{x - x_o}{\Delta x} - n$$

This is shown to be correct if

$$x - x_o = n \cdot \Delta x + (x - x_n)$$

is substituted in the equation for dx. The quantities x_o and Δx are stored with the table of function values. The table, T, must be of the form

T(1)	x_o
T(2)	Number of x breakpoints - 1
T(3)	$1/\Delta x$
T(4), ...	Table data of increasing x.

5.3 BLADSF

Subprogram BLADSF computes the generalized force vector for the blade subsystem. The single argument to the routine, DELQBS, is that force vector. The blade subsystem force vector is defined

$$\{\Delta Q\}_{BS} = \begin{Bmatrix} \Delta Q_{BL_1} \\ \cdots\cdots \\ \Delta Q_{BL_2} \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ \Delta Q_{BL_n} \end{Bmatrix}$$

where for each blade n

$$\left\{\Delta Q_{BL}\right\}_n = \begin{Bmatrix} \Delta F_{Am} \\ \cdots\cdots \\ \Delta F_{\beta} \end{Bmatrix}_n - \begin{Bmatrix} \frac{\partial U}{\partial Am} \\ \cdots\cdots \\ \frac{\partial U}{\partial \beta} \end{Bmatrix}_n - \begin{Bmatrix} \frac{\partial B}{\partial \dot{A}m} \\ \cdots\cdots \\ \frac{\partial B}{\partial \dot{\beta}} \end{Bmatrix}_n$$

for all modes $m = 1, \ldots, N_{MB}$

Consider first, the bending mode equation. The $\left\{\Delta F_{Am}\right\}$ vector has been computed in SWEEP where

$$\begin{Bmatrix} \Delta F_{A_1} \\ \Delta F_{A_2} \\ \Delta F_{A_3} \end{Bmatrix}_n = \begin{matrix} F_{1,n} \\ F_{2,n} \\ F_{3,n} \end{matrix} \quad \text{(elements of the array F(66, 7))}$$

The potential energy term is given by

$$\left\{\frac{\partial U}{\partial Am}\right\}_n = \sum_{j=1}^{N_{mb}} K_{mj} \cdot A_{jn} \qquad , m = 1, \ldots, N_{mb}$$

Two damping terms are computed. The structural damping is given by

$$\Delta\left\{\frac{\partial B}{\partial \dot{A}m}\right\}_n = C_S \sum_{j=1}^{N_{mb}} K_{mj} \cdot \dot{A}_{jn}$$

The proportionality factor , C_S, is controlled via input. The spring constants, K_{mj}, are also input.

A lag damper with relief valve is also modeled. For each blade n

$$\dot{y}'_{nc} = \sum_{j=1}^{N_{mb}} \left(\frac{\partial Y'_c}{\partial A_j}\right) \dot{A}_{jn}$$

Now if

$$\left|\dot{y}_{n'_c}\right| < \dot{y}'_1$$

then

$$M_{LAG} = C_{LAG1}\, \dot{y}'_{n_c}$$

otherwise

$$M_{LAG} = C_{LAG_1} \dot{y}'_B + C_{LAG_2} \left(\dot{y}'_{n_c} - \dot{y}'_B\right)$$

where

$$\dot{y}'_B = \dot{y}'_1 \left|\dot{y}'_{n_c}\right| / \dot{y}'_{n_c} \quad .$$

Finally, the damping term is

$$\Delta \left(\frac{\partial B}{\partial \dot{A}m}\right)_n = M_{LAG} \left(\frac{\partial y'_c}{\partial \dot{A}m}\right)_n \quad .$$

In the above equations, the following parameters are input

$$\frac{\partial \dot{y}_c}{\partial A_m} \equiv \begin{matrix} YP(1) \\ YP(2) \\ YP(3) \end{matrix} \qquad m = 1, 2, 3$$

$$\dot{y}'_1 \equiv YDPI$$

$$C_{LAG1} \equiv CLAG1$$

$$C_{LAG2} \equiv CLAG2$$

The fourth mode in the set can be pitch horn bending or dynamic torsion. If $N_{BP} \neq 0$ then

$$\Delta Q_{\beta_n} = M_{F_n} \left(\frac{\partial \phi_f}{\partial \beta_{PH}}\right)_n - K_{\beta_{PH_n}} \beta_{PH_n}$$

If $N_{\Phi_T} \neq 0$ then

$$\Delta Q_{\beta_n} = F_{17,n} - K_{\Phi_T} \cdot \Phi_{T_n} - C_{\dot{\Phi}_T} \cdot \dot{\Phi}_{T_n}$$

5.4 HUBF

Subprogram HUBF computes the force vector relative to the hub, DELQH. The hub is not a generalized coordinate subsystem, but does provide a transition of forces from the main rotor to other subsystems. Equations of interest follow.

$$\{\Delta Q\}_H = \{\Delta Q\}_{MR_H} - \left[M_{OH} + M_{TH}\right] \begin{Bmatrix} \ddot{X}_o \\ \ddot{Y}_o \\ \ddot{Z}_o \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix}_H$$

$$- \left\{ \begin{matrix} \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}_H \begin{bmatrix} \text{3 x 3 from upper right quad of } M_{OH} + M_{TH} \end{bmatrix} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix}_H \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}_H \begin{bmatrix} \text{3 x 3 from lower right quad of } M_{DN} + M_{TN} \end{bmatrix} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix}_H \end{matrix} \right\}$$

$$+ \{\Delta F\}_{SP_H}$$

and

$$
\left\{\Delta F\right\}_{SP} = \left\{\frac{\partial \tau_{OSP}}{\partial \tau_{OH}}\right\}^{T} \left[- \left[M_{OSP}\right] \begin{Bmatrix} \ddot{X}_o \\ \ddot{Y}_o \\ \ddot{Z}_o \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix}_{SP} - \begin{Bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \\ \cdots\cdots\cdots \\ \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}_{SP} \begin{bmatrix} I_{XX} & & \\ & I_{YY} & \\ & & I_{ZZ} \end{bmatrix}_{SP} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix}_{SP} \end{Bmatrix} \right]
$$

Other equations included are:

$$
\left\{\frac{\partial \tau_{OSP}}{\partial \tau_{OH}}\right\} = \left[\begin{array}{c:ccc} & 0 & Z_{o_{SP}} & 0 \\ [I] & -Z_{o_{SP}} & 0 & 0 \\ & 0 & 0 & 0 \\ \hdashline 0 & & \left[T_{H \to SP}\right] & \end{array} \right]
$$

$$\begin{Bmatrix} \ddot{X}_o \\ \ddot{Y}_o \\ \ddot{Z}_o \end{Bmatrix}_{SP} = \begin{Bmatrix} \ddot{X}_o \\ \ddot{Y}_o \\ \ddot{Z}_o \end{Bmatrix}_{H} + \begin{Bmatrix} 0 \\ 0 \\ \ddot{Z} \end{Bmatrix}_{SP} + W_H\, W_H \begin{Bmatrix} 0 \\ 0 \\ Z_o \end{Bmatrix}_{SP} + \dot{W}_H \begin{Bmatrix} 0 \\ 0 \\ Z_o \end{Bmatrix}_{SP} + 2\, W_H \begin{Bmatrix} 0 \\ 0 \\ \dot{Z} \end{Bmatrix}_{SP}$$

$$W_H = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}_H$$

The main term in the hub force equation is the main rotor force relative to the hub.

$$\left\{\Delta Q\right\}_{MR_H} = \left\{\frac{\partial \tau_{OR}}{\partial \tau_H}\right\}^T \left\{\Delta Q\right\}_{MR_R}$$

$$\left\{\Delta Q\right\}_{MR_R} = \sum_{n=1}^{N_b} \left\{\frac{\partial \tau_{OBL}}{\partial \tau_R}\right\}_n^T \left\{\Delta Q\right\}_{BL_n}$$

The blade generalized forces are computed in SWEEP as elements of the F array.

$$\left\{\Delta Q\right\}_{BL_n} = \begin{Bmatrix} F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \end{Bmatrix}_n$$

5.5 SWASHF

Subprogram SWASHF calculates the generalized force vector for the swashplate subsystem, DELQSP. The swashplate subsystem consists of a maximum of three coordinates.

$$q_{SP} = (\phi, \theta, Z)_{SP}$$

The force equation is given as

$$\left\{\Delta Q\right\}_{SP} = \sum_{n=1}^{N_b} \left\{\frac{\partial \phi_f}{\partial q_{SP}}\right\}_n^T M_{\phi_{f_n}} - \left\{\frac{\partial U}{\partial q_{SP}}\right\} - \left\{\frac{\partial B}{\partial \dot{q}_{SP}}\right\} - \left\{M_{FR_{SP}}\right\} - \left\{\Delta F_{SP}\right\} .$$

$M_{\phi_{f_n}}$ is the feathering moment computed in SWEEP. The frictional force vector is

$$\left\{M_{FR_{SP}}\right\} = \begin{Bmatrix} f(\dot{\phi}_{SP}) \\ f(\dot{\theta}_{SP}) \\ 0 \end{Bmatrix}$$

where the frictional functions are determined by entries to the subprogram COULOM.

The term $\left\{\Delta F\right\}_{SP}$ as written above is a (3 x 1) vector with the stated coordinate ordering. The definition is given below in the standard (6 x 1) form for clarity.

$$
\left\{\Delta F\right\}_{SP} = - \left\{\frac{\partial\tau_{OSP}}{\partial\tau_{SP}}\right\}^{T} \left\{ \left[M_{OSP}\right] \begin{Bmatrix} \ddot{X} \\ \dot{Y} \\ \ddot{Z} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix}_{SP} \right.
$$

$$
\left. + \left[\begin{array}{c} \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}_{SP} \begin{bmatrix} I_{XX} & & \\ & I_{YY} & \\ & & I_{ZZ} \end{bmatrix}_{SP} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix}_{SP} \end{array} \right] \right\}
$$

$$
\left\{\Delta F\right\}_{SP} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \left\{\Delta F\right\}_{SP}
$$

$$
= \begin{Bmatrix} (\cos\theta \; (I_{XX}\dot{p} + qr\,(I_{ZZ} - I_{YY})) + \sin\theta \cdot I_{ZZ} \cdot \dot{r})_{SP} \\ (I_{YY}q + pr \cdot (I_{XX} - I_{ZZ}))_{SP} \\ (m\,\ddot{Z})_{SP} \end{Bmatrix}
$$

The potential energy and dissipation functions are presented below. The angular coordinates are treated separately from Z_{SP}.

Let

$$[K_{SP}] = \left[\begin{array}{c:c} K_{\phi_{SP}} \cos^2\psi_c + K_{\theta_{SP}} \sin^2\psi_c & \left(K_{\phi_{SP}} - K_{\theta_{SP}}\right) \sin\psi_c \cos\psi_c \\ \hdashline K_{\phi_{SP}} - K_{\theta_{SP}} \quad \sin\psi_c \cos\psi_c & K_{\psi_{SP}} \sin^2\psi_c + K_{\theta_{SP}} \cos^2\psi_c \end{array}\right]$$

and

$$[C_{SP}] = \left[f\left(C_{\phi_{SP}}, C_{\theta_{SP}}\right)\right]$$

$$\left[T_{\psi_c}\right] = \begin{bmatrix} \cos\psi_c & \sin\psi_c \\ -\sin\psi_c & \cos\psi_c \end{bmatrix} .$$

Define a control load

$$\left\{LD\right\}_c = \begin{Bmatrix} -K_{X_{CS}} \cdot X_{CS} \\ K_{Y_{CS}} \cdot Y_{CS} \end{Bmatrix} .$$

Now

$$\left\{\frac{\partial U}{\partial q}\right\}_{SP_1} = [K_{SP}] \begin{Bmatrix} \phi_{SP} \\ \theta_{SP} \end{Bmatrix} - \left[T_{\psi_c}\right]^T \left\{LD\right\}_c$$

$$\left\{\frac{\partial B}{\partial q}\right\}_{SP} = [C_{SP}] \begin{Bmatrix} \dot{\phi}_{SP} \\ \dot{\theta}_{SP} \end{Bmatrix} .$$

A stop load is modeled.

If $\left(\phi_{SP}^{\,2} + \theta_{SP}^{\,2}\right)^{1/2} > \delta_{s,\,SP}$ then

$$\left\{L_s\right\}_{SP} = L \cdot \begin{Bmatrix} \phi_{SP} \\ \theta_{SP} \end{Bmatrix}$$

where

$$L = K_{s,SP} \left[\left(\phi_{SP}^{\,2} + \theta_{SP}^{\,2}\right)^{1/2} - \delta_{s,SP}\right] \Big/ \left(\phi_{SP}^{\,2} + \theta_{SP}^{\,2}\right)^{1/2}$$

otherwise

$$\left\{L_s\right\}_{SP} = 0 \qquad ;$$

and

$$\left\{\frac{\partial U}{\partial q}\right\}_{SP} = \left\{\frac{\partial U}{\partial q}\right\}_{SP_1} + \left\{L_s\right\}_{SP}$$

then Z_s coordinate loads are given by the following equations. If $\left|Z_{SP}\right| \leq Z_{G1}$

$$\frac{\partial U}{\partial Z_{SP}} = K_{1Z_{SP}} \cdot Z_{SP} + F_c$$

Otherwise

$$\frac{\partial U}{\partial Z_{SP}} = K_{1Z_{SP}} \cdot Z_{1SP}' + K_{2Z_{SP}} \left(Z_{SP} - Z_{1SP}'\right) + F_c$$

where

$$Z'_{1SP} = \text{sign}(Z_{SP}) \cdot Z_{G1}$$

$$\frac{\partial B}{\partial Z_{SP}} = C_{Z_{SP}} \dot{Z}_{SP} - R_{Z_\phi} \left[C_{1,1}\right]_{SP} \dot{\phi}_{SP} \phi_{SP} - R_{Z_\theta} \left[C_{2,2}\right]_{SP} \cdot \dot{\theta}_{SP} \cdot \theta_{SP}$$

The above equations contain a great deal of data. The equation symbols, corresponding program symbols and input addresses are listed.

Symbol	FORTRAN	RA
ψ_c	CHI	119
K_{XCS}	QKXCS	123
K_{YCS}	QKYCS	124
$K_{\phi SP}$	KPHCON	376
$K_{\theta SP}$	KTHCON	377
$C_{\phi_{SP}}$	CPHCON	378
$C_{\theta_{SP}}$	CTHCON	379
$\delta_{s,SP}$	GASTOP	1276
$K_{s,SP}$	GKSTOP	1277
$\dot{\phi}_{SP,BR}$, $\dot{\theta}_{SP,BR}$	RLG	116
M_{FR}, ϕ_{SP}; M_{FR}, θ_{SP}	FCG	117
K_{1ZSP}	QKGZ1	137
Z_{1SP}	ZG1	141
$K_{2Z_{SP}}$	QKGZ2	140
F_c	FIDDLE	1494
$C_{Z_{SP}}$	QCGZ	138
R_{Z_ϕ}, R_{Z_θ}	DGDHG	1263

The reader can reference Section 5.9 of Volume I of this report for additional control system explanation.

5.6 SHAFTF

Subprogram SHAFTF computes the generalized force vector, DELQS. The force equation is

$$\left\{\Delta Q\right\}_S = \left\{\frac{\partial\tau_{OH}}{\partial\tau_S}\right\}^T \left\{\Delta Q\right\}_H - \left[K_s\right] \left\{q\right\}_S - \left[C_s\right] \left\{\dot{q}\right\}_S$$

The hub force vector is computed in HUBF and passed as an input argument. Since the shaft coordinate set can be a subset of the standard six coordinates, the K_S and C_S coefficient matrices have been assembled in the data preparation subprogram DTPREP. The matrices are N_S x N_S symmetric. The reader can reference Section 5.7 of Volume I for further information on the shaft force equations.

5.7 REFF

Subprogram REFF computes the reference system generalized force vector, DELQF. The force equation is

$$\left\{\Delta Q\right\}_{REF} = \left\{\frac{\partial\tau_{OH}}{\partial\tau_{REF}}\right\}^T \left\{\Delta Q\right\}_H + \left\{\frac{\partial\tau_{OF}}{\partial\tau_{REF}}\right\}^T \left\{\Delta F\right\}_F + \left\{\Delta F_{TR}\right\}_{REF} + \left\{\Delta F_{ENG}\right\}_{REF}$$

The hub force vector is computed in HUBF and passed to REFF via the argument list. The fuselage force vector is given by

$$\left\{\Delta F\right\}_F = \left\{\Delta F_A\right\}_F - \left[M_{OF}\right] \begin{Bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix}_F$$

$$- \left[\begin{matrix} \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}_F \begin{bmatrix} \text{(3 x 3)} \\ \text{upper right} \\ \text{quad of} \\ M_{OF} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}_F \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}_F \begin{bmatrix} \text{(3 x 3)} \\ \text{lower right} \\ \text{quad cf} \\ M_{OF} \end{bmatrix} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix}_F \end{matrix} \right]$$

The fuselage aero loads depicted by $\{\Delta F_A\}_F$ are computed in subprogram LOADS. The tail rotor and engine contributions are

$$\left\{\Delta F_{TR}\right\}_{REF} + \left\{\Delta F_{ENG}\right\}_{REF} = - \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ I_{YY_{TR}} \; G_{T_R} \; \dot{r}_R \\ 0 \end{Bmatrix} - \begin{Bmatrix} 0 \\ 0 \\ 0 \\ -G_{ENG} \; \dot{r}_R \\ -G_{ENG} \; r_R \; r_{REF} \\ G_{ENG} \; q_{REF} \; r_R \end{Bmatrix}$$

where

$$r_R = \ddot{\psi}_R$$

$$\dot{r}_R = \ddot{\psi}_R$$

The above equations can be referenced in Section 5.8 of Volume I.

5.8 LOADS

Subprogram LOADS computes a total vector of nonmain rotor aerodynamic loads in fuselage coordinates. This force vector is the primary vector of the reference system force vector. The loads summation is given by

$$\left\{\Delta F_A\right\}_F = \left(\left\{L_B\right\}_F + \left\{L_{TR}\right\}_F + \left\{L_P\right\}_F\right) \rho/\rho_o$$

where $\{L_B\}_F$ stands for total body loads, $\{L_{TR}\}$ is tail rotor loads, and $\{L_P\}_F$ is the thrustor loads vector. Program symbols are

$\left\{\Delta F_A\right\}_F$	QLOADS(6)
$\left\{L_B\right\}_F$	FN(6)
$\left\{L_{TR}\right\}_F$	FTR(6)
$\left\{L_P\right\}_F$	FP(6)
ρ/ρ_o	SIGMA (density ratio)

The equations are straight forward but extensive. The reader should refer to Sections 6.3, 6.4, 6.5, and 6.6 of Volume I for equation development.

5.9 ROTORF

Subprogram ROTORF computes the generalized force for the rotor (engine) subsystem, DELQR. The system is a one-coordinate system and the equation is given by

$$\Delta Q_R = M_{Z_{MR_R}} - \left(M_{X_I} + M_{X_A}\right)_{ENG} \frac{\partial \phi_{ENG}}{\partial \psi_R}$$

$$+ \left(M_{Y_I} + M_{Y_A}\right)_{TR} \frac{\partial \theta_{TR}}{\partial \psi_R} - \left(I_{ZZ_H} + I_{ZZ_{SP}}\right) (\ddot{\psi}_R)$$

$$+ M_{\phi_f} - M_{ZZ_{END}}$$

where

$$M_{X_{I_{ENG}}} = - I_{XX_{ENG}} \left(\dot{p}_F - \frac{\partial \phi_{ENG}}{\partial \psi_R} \ddot{\psi}_R\right)$$

$$M_{X_{A_{ENG}}} = 0$$

$$M_{Y_{I_{TR}}} = - I_{XX_{TR}} \left(\dot{q}_F + \frac{\partial \theta_{TR}}{\partial \psi_R} \ddot{\psi}_R\right)$$

$$M_{Y_{A_{TR}}} = M_{Y_{A_{TR}}} \cdot \rho/\rho_o$$

$$M_{\phi_f} = \sum_{n=1}^{N_b} \left(\frac{\partial \phi_f}{\partial \psi_R}\right)^T_n M_{\phi_{f_n}}$$

The reader can refer to Section 5.10 of Volume I for equation development.

6. MASSSP — LEVEL 4 EXECUTIVE

MASSSP is a level 4 executive control program. Its primary output is the generalized mass matrix. The elements of MASSSP are depicted in Figure 25. As in the development of the force vector, a subsystem approach is followed.

The generalized mass matrix is symmetric positive definite. Thus, only half the matrix is computed. A schematic of a complete mass matrix is presented in Figure 26.

This subprogram controls the assembly of the generalized mass matrix. Each model subsystem is developed individually as required. The required subsystem coupling matrices are stored in the generalized mass matrix via a utility routine, RMSTOR. The subsystem diagonal matrices are stored by the utility routine, DMSTOR. The blade subsystem mass matrix is handled directly by its development routine, BLADSM.

The subsystem matrices, their sizes, and the location of each within the generalized mass matrix are listed below.

Subsystem	Matrix Size n, m	Location Within Gen. Matrix, M(i, j)	
		i	j
Swashplate	M_{SPBL} (NSP, NBS)	LSP	1
Shaft	M_{SBL} (NS, NBS)	LS	1
	M_{SSP} (NS, NSP)	LS	LSP
Reference	M_{REFBL} (NREF, NBS)	LREF	1
	M_{REFSP} (NREF, NSP)	LREF	LSP
	M_{REFS} (NREF, NS)	LREF	LS
Rotor	M_{RBL} (NR, NBS)	LR	1
	M_{RSP} (NR, NSP)	LR	LSP
	M_{RS} (NR, NS)	LR	LS
	M_{RREF} (NR, NREF)	LR	LREF

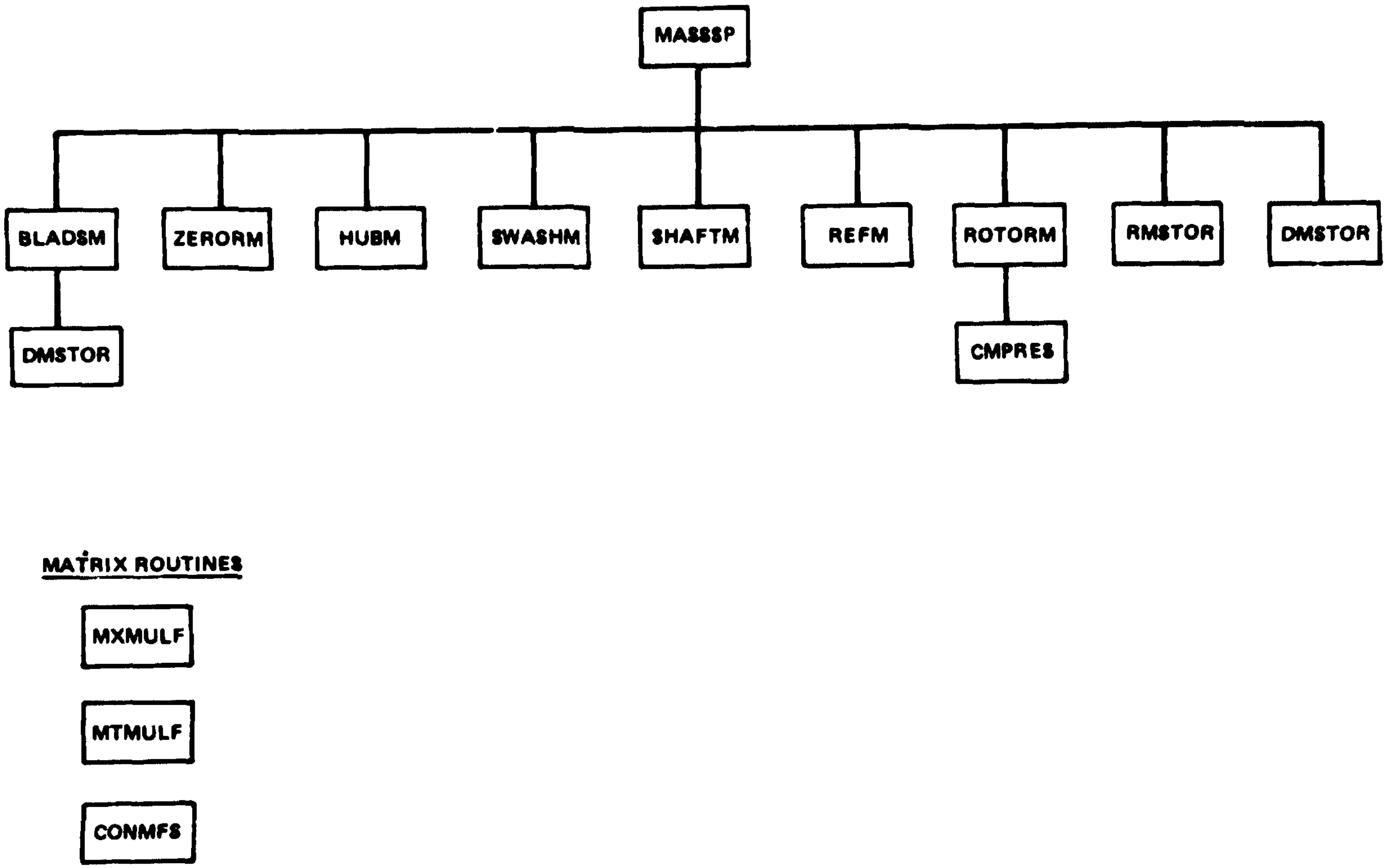


Figure 25. -- MASSSP - level 4 hierarchieval chart.

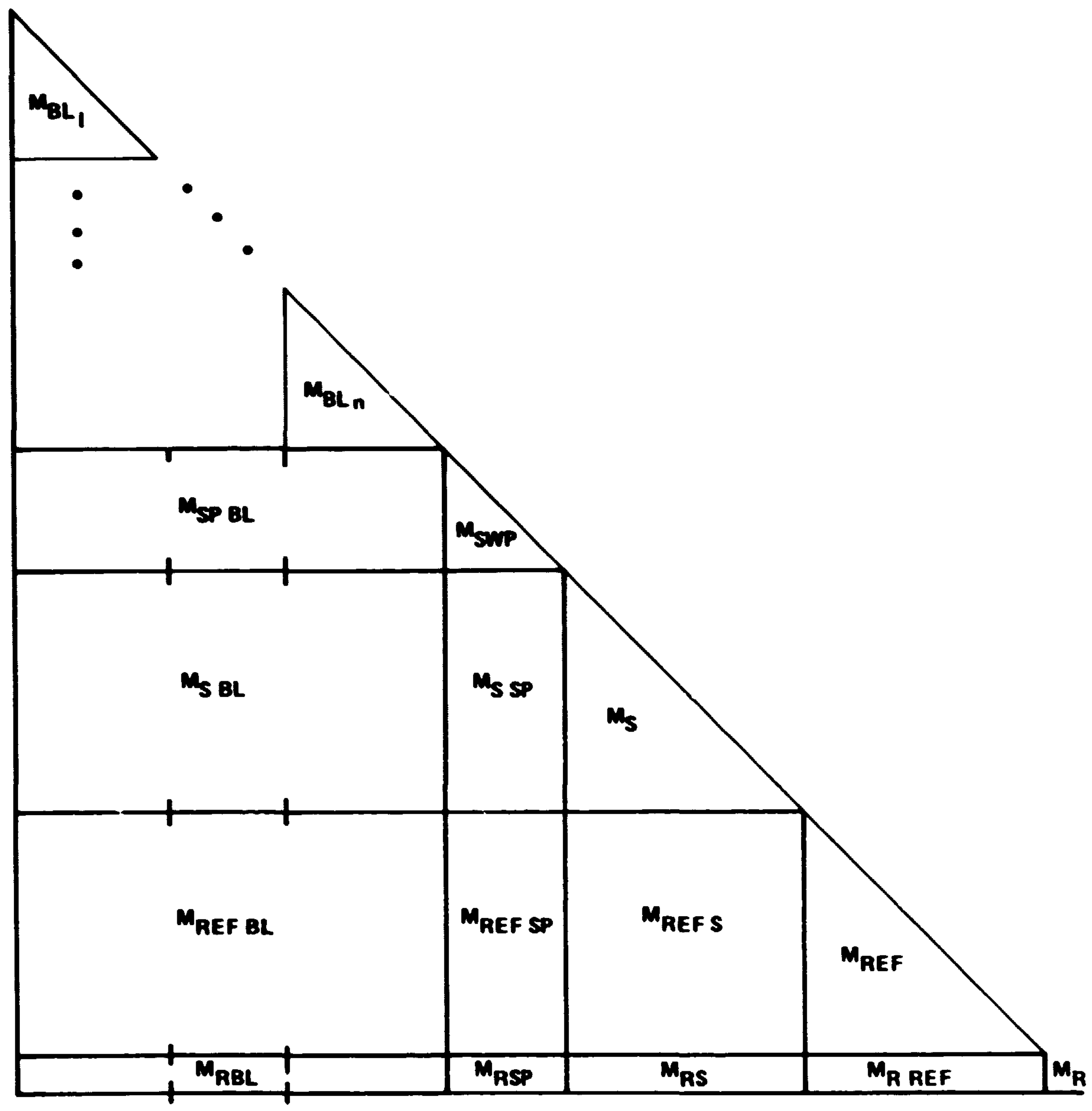


Figure 26. - Generalized mass matrix.

The flow of MASSSP is depicted in Figure 27. Routines for each model subsystem are easily recognized. Two other routines identified are ZERORM and HUBM. These routines compute intermediate mass matrices which are used in developing the required coupling matrices.

6.1 DMSTOR

This routine stores a subsystem diagonal mass matrix into the generalized mass matrix. The subsystem and generalized mass matrix are in symmetric storage mode form.

Arguments List

L	Subsystem coordinate vector location within the generalized coord. vector
N	Subsystem size
MSUB	Subsystem diagonal mass matrix
MSP	Generalized mass matrix.

Subroutine DMSTOR is depicted in Figure 28.

6.2 RMSTOR

This subprogram stores a subsystem rectangular coupling matrix $M_{SUB}(n, m)$ into the proper locations within the generalized mass matrix which is in symmetric storage mode.

Arguments List

N M	Submatrix dimensions $M_{SUB}(N, M)$
NMAX	Row definition in DIMENSION statement for MSUB
MSUB	Submatrix
I J	Logical location of MSUB(1, 1) within the generalized mass matrix
MSP	Generalized mass matrix.

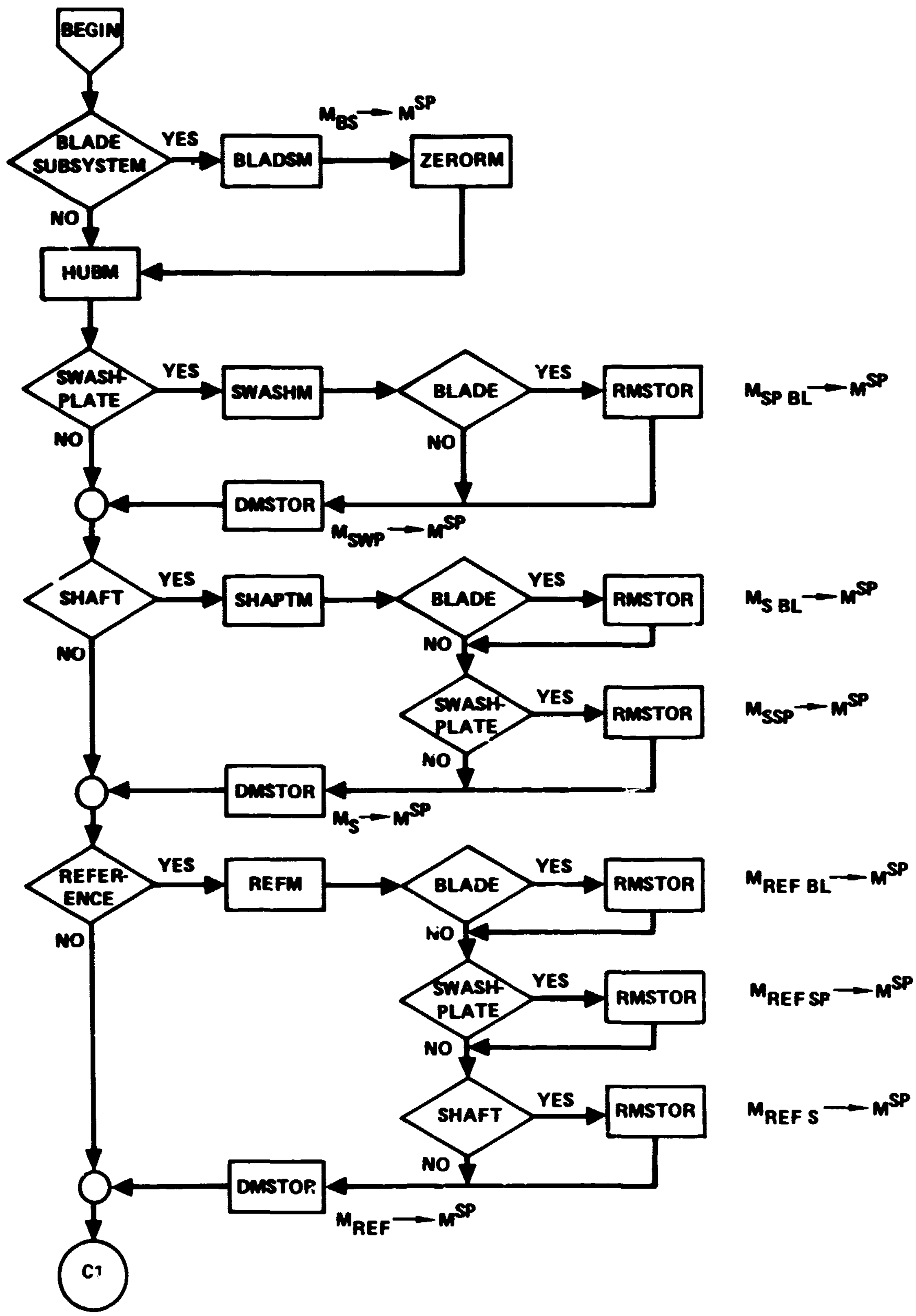


Figure 27. - MASSSP flow diagram.

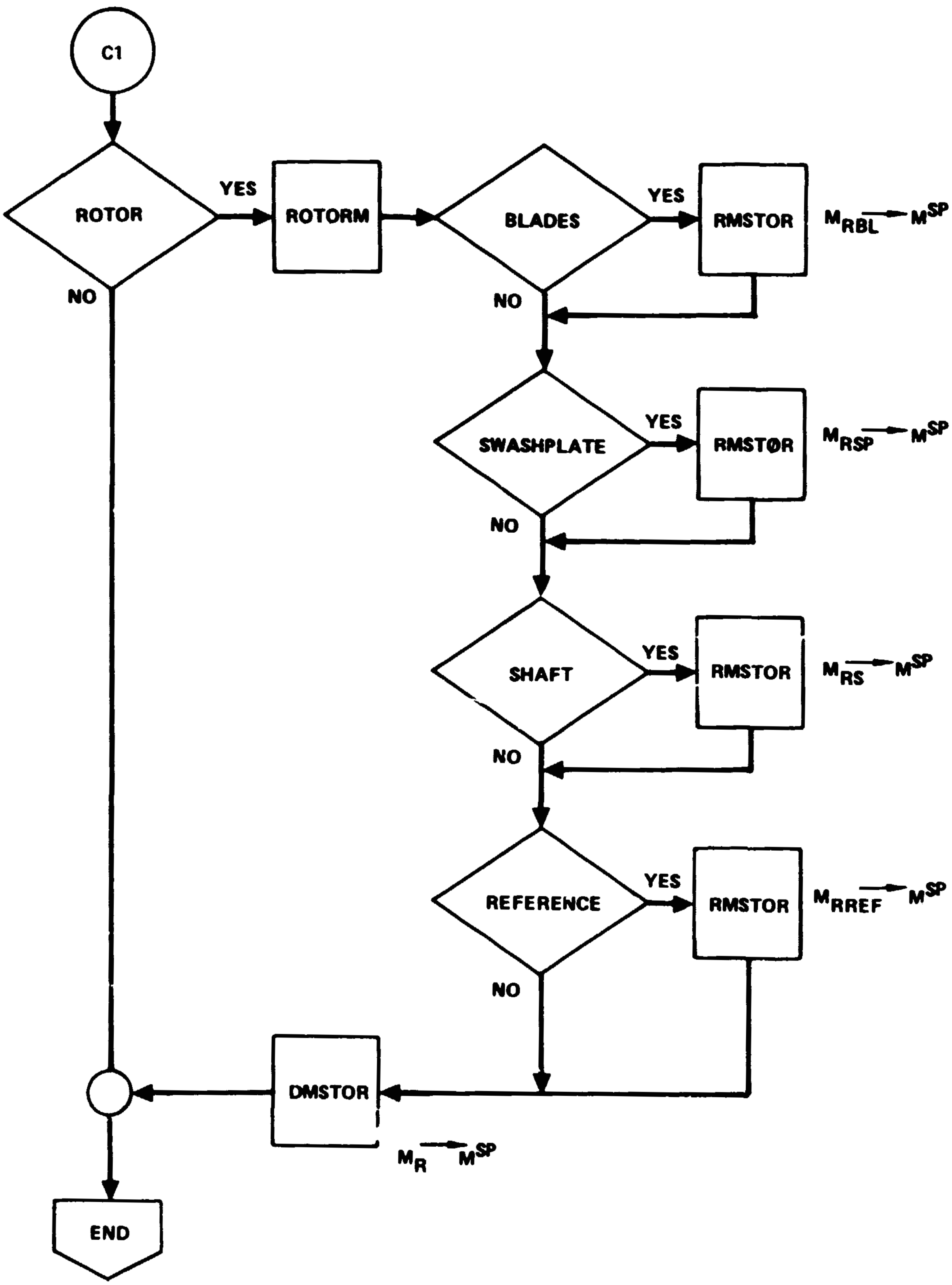


Figure 27. - Concluded.

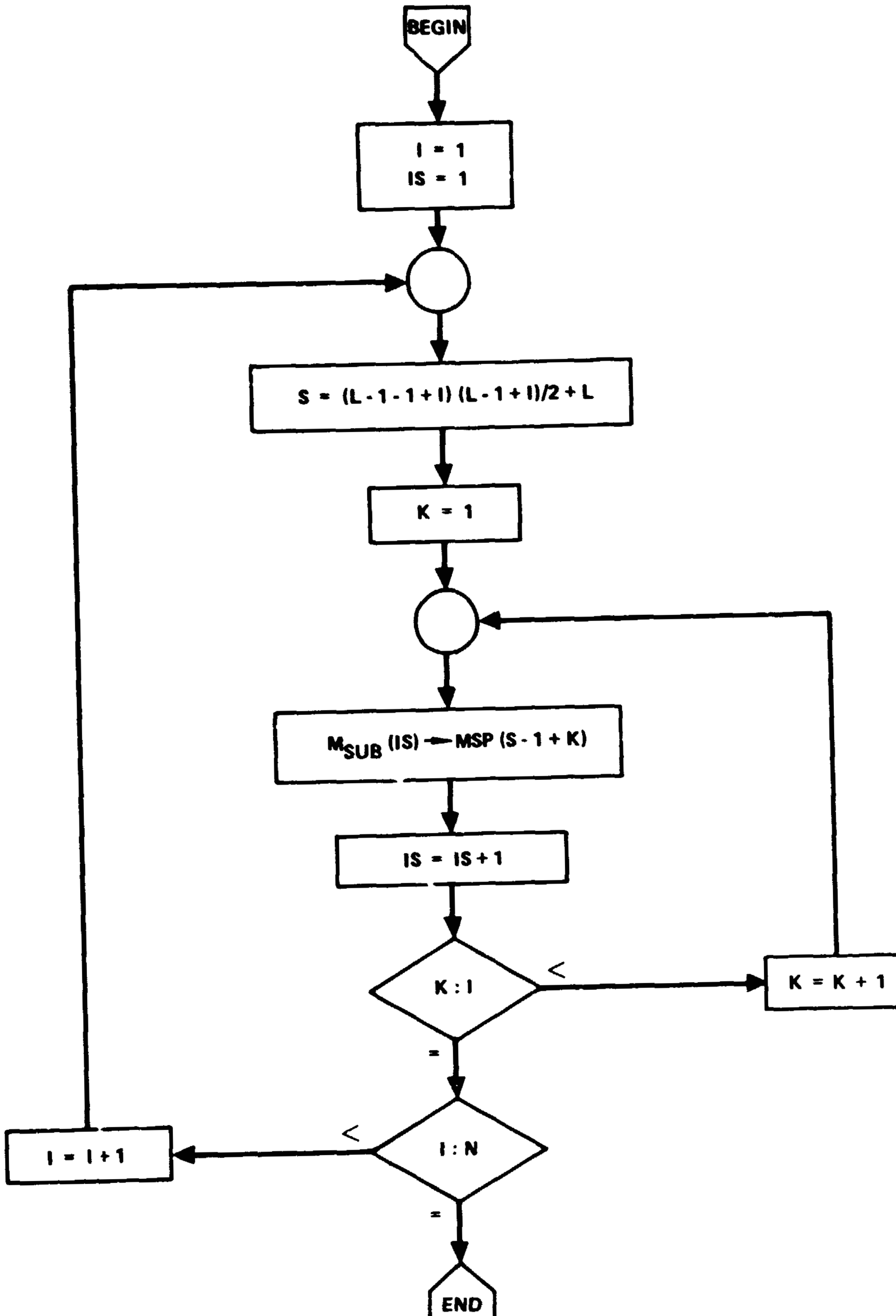


Figure 28. - DMSTOR flow diagram.

The problem is, given the size of the rectangular submatrix (n, m), to locate the origin within the logical generalized matrix, M_{ij}. Then

$$M_{SUB}(k, h) \rightarrow M(i - 1 + k, j - 1 + k)$$

$$k = 1, \ldots, n$$

$$h = 1, \ldots, m.$$

The real mass matrix is in symmetric storage mode, i.e., in vector form. However, an element, S, of the vector corresponds to the matrix element qr where

$$S = (q)(q - 1)/2 + r$$

The final algorithm is then

$$M_{SUB}(k, h) \rightarrow V(t - 1 + h)$$

where

$$t = (i - 1 + k - 1)(i - 1 + k)/2 + j$$

for

$k = 1, \ldots, n$

$h = 1, \ldots, m$

This is depicted in the Figure 29.

6.3 BLADSM

The subprogram BLADSM assembles the elements of the blade subsystem generalized mass. The use of intermediate matrices is minimized because of the specialized form of the subsystem. Data is stored directly into the generalized mass matrix, MSP, which is the only quantity in the argument list.

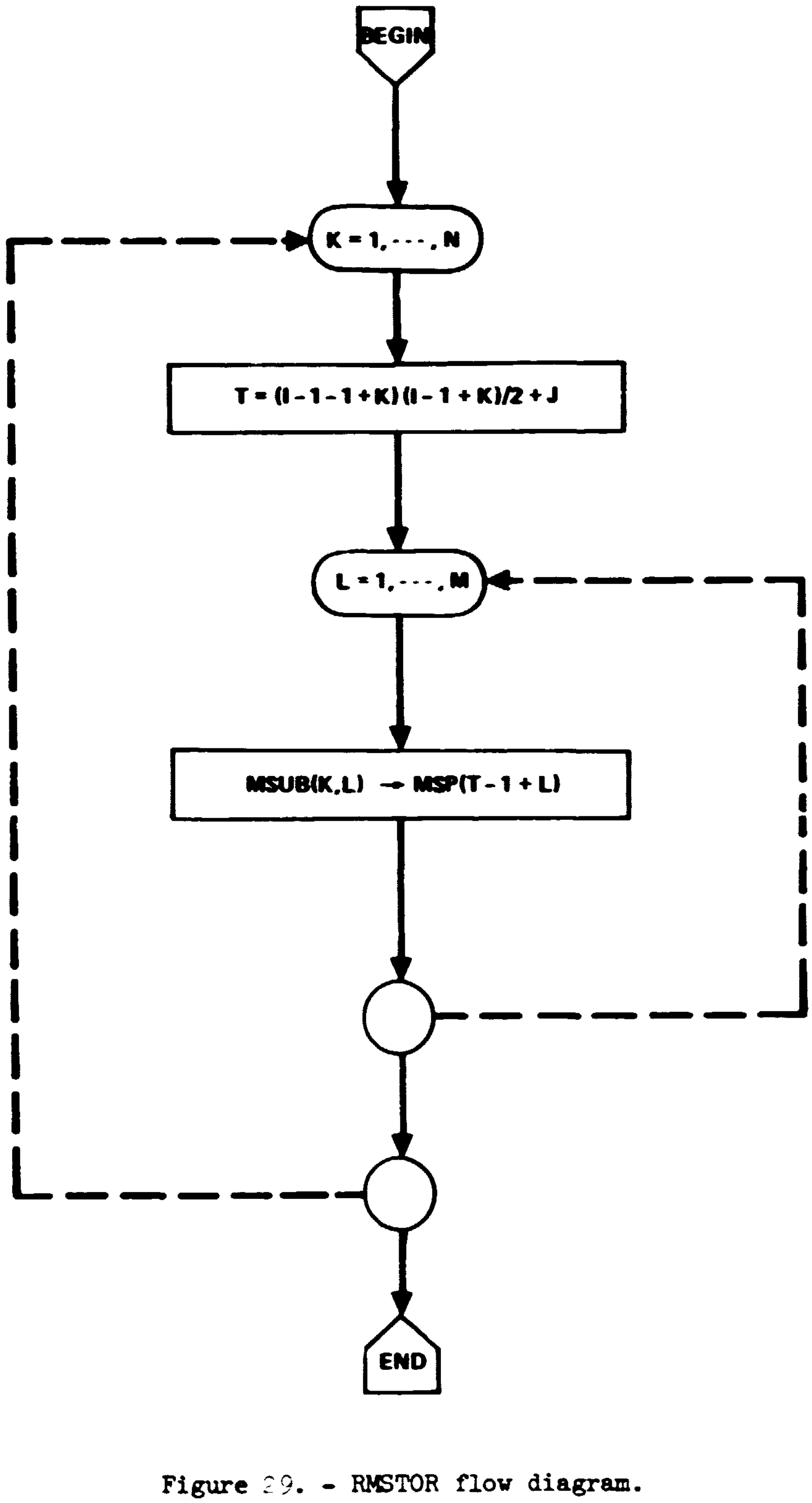


Figure 29. - RMSTOR flow diagram.

The mass matrix for the total blade system can be depicted as

$$M_{BLN} = \begin{bmatrix} M_{BL_1} & & & \bigcirc \\ & M_{BL_2} & & \\ & & \ddots & \\ \bigcirc & & & M_{BL_{N_b}} \end{bmatrix}$$

The symmetric mass matrix for each blade, n, can be further expanded as

$$\left[M_{BL}\right]_n = \begin{bmatrix} M_A & \\ M_{\beta A} & M_{\beta\beta} \end{bmatrix} = \left[\begin{array}{ccc|c} M_{A_{11}} & & & \\ M_{A21} & M_{A22} & & \\ M_{A31} & M_{A32} & M_{A33} & \\ \hline M_{\beta A}, & M_{\beta A_2} & M_{\beta A_3} & M_{\beta\beta} \end{array}\right]$$

The subscript, A, identifies blade bending modes. The number of bending modes, N_{mb}, is optional. The addition of another mode, β, is also optional. If the input NBP = 1, then $\beta = \beta_{Ph}$, where β_{Ph} stands for pitch horn bending (elastic feathering). If the input NPT = 1, then $\beta = \phi_T$, where ϕ_T stands for dynamic torsion. Also, note, that the number of blades, N_b, is optional.

Current limits on blade coordinates are

N_b	NB	7 blades
N_{mb}	NMB	3 bending modes per blade

$N_{\beta P}$ NBP 1 elastic feathering mode per blade

$N_{\phi T}$ NPT 1 dynamic torsion mode per blade.

$N_{\beta P}$ and $N_{\phi T}$ are mutually exclusive.

The mass matrix definitions to be given below are in terms of mass integrals developed in subprogram SWEEP. Blade force and mass integrals are stored in an array $F_{i,\ n}$. Expanded definitions can be found in the section on SWEEP.

For each blade, n, the elements of the symmetric matrix $[M_{BL}]$ in symmetric storage mode are

$$M_{Aj} = F_{22+j,n} \qquad j = 1, \ldots, N_{mb}$$

and if $N_{\beta P} \neq 0$

$$M_{\beta A_j} = \left(\frac{\partial \phi_f}{\partial \beta_{Ph}}\right) \cdot F_{19+j,n} \qquad j = 1, \ldots, N_{mb}$$

$$M_{\beta\beta} = \left(\frac{\partial \phi_f}{\partial \beta_{Ph}}\right)^2 \cdot F_{19,n}$$

or if $N_{\phi_T} \neq 0$

$$M_{\beta A_j} = F_{56+j,n} \qquad j = 1, \ldots, N_{mb}$$

$$M_{\beta\beta} = F_{60,n}$$

otherwise

$$M_{\beta A} \equiv 0$$

$$M_{\beta\beta} \equiv 0$$

The utility routine, DMSTOR, is used to load each $[M_{BL}]$ into $[M]^{SF}$ as each blade is evaluated.

6.4 ZERORM

The subprogram ZERORM computes a set of intermediate matrices for use by other routines. These include M_{OR} and M_{ORBL}. The rationale is to collect all blade-dependent data in a pseudo system which is called zero of the rotor, thus the subscript (OR). The matrix definitions are

$$\underset{(6x6)}{\left[M_{OR}\right]} = \sum_{n=1}^{N_b} \left\{\frac{\partial \tau_{OBL}}{\partial \tau_{OR}}\right\}^T \left[M_{BLO}\right]_n \left\{\frac{\partial \tau_{OBL}}{\partial \tau_{OR}}\right\}$$

where

$$\left[M_{BLO}\right]_n = \left[\begin{array}{ccc|ccc} M_{BL} & 0 & 0 & 0 & F_{56} & -F_{55} \\ 0 & M_{BL} & 0 & -F_{56} & 0 & F_{54} \\ 0 & 0 & M_{BL} & F_{55} & -F_{54} & 0 \\ \hline 0 & -F_{56} & F_{55} & I_{XXBL} & 0 & F_{52} \\ F_{56} & 0 & -F_{54} & 0 & I_{YYBL} & F_{53} \\ -F_{55} & F_{44} & 0 & F_{52} & F_{53} & I_{ZZBL} \end{array}\right]_n$$

and

$$\left[M_{ORBL}\right] = \left[M_{ORBL_1} \,\middle|\, \cdots \,\middle|\, M_{ORBLn}\right], \quad (6 \times N_{BS})$$

where

$$\underset{(6 \times N_m)}{\left[M_{ORBL}\right]_n} = \underset{(6 \times 6)}{\left\{\frac{\partial \tau_{O_{BL}}}{\partial \tau_{OR}}\right\}^T} \underset{(6 \times N_m)}{\left[M_{OBLBL}\right]_n} .$$

Continuing with the definitions,

$$\left[M_{OBLBL}\right]_n = \left[M_{OBLAm} \vdots M_{OBL_\beta}\right]_n$$

where for all blade bending modes, m,

$$\left[M_{OBLAm}\right]_n = \begin{bmatrix} F_{33} & F_{34} & F_{35} \\ F_{36} & F_{37} & F_{38} \\ F_{40} & F_{41} & F_{42} \\ F_{44} & F_{45} & F_{46} \\ F_{48} & F_{49} & F_{50} \\ F_{29} & F_{30} & F_{31} \end{bmatrix}_n \quad , m = 1, 2, 3$$

Actually, the columns used are j = 1, ..., N_{mb}. If $N_{\beta P} \neq 0$ then

$$\left[M_{OBL\beta}\right]_n = \left(\frac{\partial \phi_f}{\partial \beta_{Ph}}\right)_n \begin{bmatrix} F_{62} \\ F_{39} \\ F_{43} \\ F_{47} \\ F_{51} \\ F_{32} \end{bmatrix}_n \quad .$$

If $N_{\phi_T} \neq 0$ then

$$\left[M_{OBL\beta}\right]_n = [0]$$

Otherwise

$$\left[M_{OBL\beta}\right]_n \text{ is not defined.}$$

The partial derivative $\left\{\partial\tau_{OBL}/\partial\tau_{OR}\right\}$ is of the special form

$$\left\{\frac{\partial\tau_{OBL}}{\partial\tau_{OR}}\right\} = \left[\begin{array}{c|c} \left[T_{R-BL}\right]_n & 0 \\ \hline 0 & \left[T_{R-BL}\right]_n \end{array}\right] .$$

Therefore, the implementation of $[M_{OR}]$ and $[M_{ORBL}]$ proceeds as follows. Re-define

$$\left[M_{OR}\right] = \sum_{n=1}^{N_b} \left[M'_{OR}\right]_n$$

where

$$\left[M'_{OR}\right]_n = \left[\begin{array}{c|c} P_{11} & P_{12} \\ \hline P_{21} & P_{22} \end{array}\right]$$

and for each blade, n,

$$P_{11} = m_{BL} \left[T_{R-BL}\right]_n^T \left[T_{R-BL}\right]_n$$

$$P_{12} = \left[T_{R-BL}\right]_n^T \left[M_{BLO_{12}}\right]_n \left[T_{R-BL}\right]_n$$

$$P_{21} = - P_{12}$$

$$P_{22} = \left[T_{R-BL}\right]_n^T \left[M_{BLO_{22}}\right]_n \left[T_{R-BL}\right]_n$$

$$\left[M_{BLO_{12}}\right]_n = \begin{bmatrix} 0 & F_{56} & -F_{55} \\ -F_{56} & 0 & F_{54} \\ F_{55} & -F_{54} & 0 \end{bmatrix}_n$$

$$\left[M_{BLO_{22}}\right]_n = \begin{bmatrix} I_{XX_{BL}} & 0 & F_{52} \\ 0 & I_{YY_{BL}} & F_{53} \\ F_{52} & F_{53} & I_{ZZ_{BL}} \end{bmatrix}_n$$

Finally, partition

$$\left[M_{OBLBL}\right]_n = \begin{bmatrix} C_1 \ (3 \times N_m) \\ \hline C_2 \ (3 \times N_m) \end{bmatrix}_n$$

then

$$\left[M_{ORBL}\right]_n = \begin{bmatrix} \left[T_{R-BL}\right]^T \left[C_1\right] \\ \hline \left[T_{R-BL}\right]^T \left[C_2\right] \end{bmatrix}_n$$

Note that the subscripted F's are mass integrals developed in subprogram SWEEP. The blade mass, m_{BL}, and inertias $I_{XX_{BL}}$, $I_{YY_{BL}}$, $I_{ZZ_{BL}}$ are calculated in subprogram INIT as well as the transformation matrices from rotor to blade, $[T_{R-BL}]_n$ and the partial derivative $(\partial\phi_f/\partial\beta_{Ph})$.

6.5 HUBM

The main rotor hub serves as another pseudo system for gathering information used by other subsystems. Subprogram HUBM accepts two matrices as input directly through the calling sequence $[M_{OR}]$ and $[M_{ORBL}]$. It computes and outputs three matrices $[M_{HBL}]$, $[M_H]$, and $[M_{ORH}]$. The definitions of these matrices follow.

$$\left[M_{HBL}\right] = \left\{\frac{\partial\tau_{OR}}{\partial\tau_H}\right\}^T \left[M_{ORBL}\right]$$

$$\left[M_H\right] = \left\{\frac{\partial\tau_{OR}}{\partial\tau_H}\right\}^T \left[M_{ORH}\right] + \left[M_{OT}\right]_H + \left[M_{OSP}\right]_H$$

$$\left[M_{ORH}\right] = \left[M_{OR}\right] \left\{\frac{\partial\tau_{OR}}{\partial\tau_H}\right\}$$

The transmission mount mass matrix, $[M_{OT}]_H$ is constant and computed once in subprogram INIT. The mass of the swashplate in hub coordinates is

$$\left[M_{OSP}\right]_H = \left\{\frac{\partial\tau_{OSP}}{\partial\tau_H}\right\}^T \left[M_{OSP}\right] \left\{\frac{\partial\tau_{OSP}}{\partial\tau_H}\right\}$$

$$\left[M_{OSP}\right] = \begin{bmatrix} m & & & & & 0 \\ & m & & & & \\ & & m & & & \\ & & & I_{XX} & & \\ & 0 & & & I_{YY} & \\ & & & & & I_{ZZ} \end{bmatrix}_{SP}$$

$$\left\{\frac{\partial\tau 0_{SP}}{\partial\tau_H}\right\} = \left[\begin{array}{c|ccc} & 0 & Z_{OSP} & 0 \\ I & -Z_{OSP} & 0 & 0 \\ \hline & 0 & 0 & 0 \\ 0 & \multicolumn{3}{c}{\left[T_H \rightarrow SP\right]} \end{array}\right]$$

The partial $\{\partial\tau_{OR}/\partial\tau_H\}$ coupling the rotor to the hub is computed in subprogram PDERIV. Other definitions include

$$Z_{OSP} = Z'_{OSP} + Z_{SP}$$

$$Z'_{OSP} \equiv \text{ZZSP} \equiv \text{RA}(1469)$$

$$m_{SP} \equiv \text{SPMASS} \equiv \text{RA}(139)$$

$$I_{XX_{SP}} \equiv \text{IXXSP} \equiv \text{RA}(361)$$

$$I_{YY_{SP}} \equiv \text{IYYSP} = \text{IXXSP}$$

$$I_{ZZ_{SP}} \equiv \text{IZZSP} \equiv \text{RA}(118)$$

6.6 SWASHM

This routine computes the necessary swashplate mass matrices. These include the swashplate-blade system coupling matrix and the swashplate-swashplate symmetric mass matrix. The coupling matrices $[M_{ORSP}]$ and $[M_{HSP}]$ are also computed. The matrix definitions follow.

$$\left[M_{SPBL}\right] = \left[M_{SPBL_1} \,\middle|\, \cdots \,\middle|\, M_{SPBL_{Nb}}\right]$$

$$\left[M_{SPBL}\right]_n = \left\{\frac{\partial\phi_f}{\partial q_{SP}}\right\}^T \left[M_{\phi_f BL}\right]_n$$

$$(n_{SP} \times 1) \quad (1 \times N_m)$$

where

$$\left[M_{\phi_f BL}\right]_n = \left[M_{\phi_f A_m'} \,\vdots\, M_x\right]_n \quad .$$

If $N_{BP} \neq 0$ (pitch-horn bending)

$$M_x = \frac{\partial\phi_f}{\partial\beta_{Ph}} M_{\phi_f \phi_f}$$

or if $N_{\phi_T} \neq 0$ (dynamic torsion)

$$M_x = M_{\phi_f \phi_T}$$

otherwise

$$M_x \equiv 0 \quad .$$

Further definitions include for each blade n

$$M_{\phi_f A_m} = F_{19+m,n} \quad , m = 1, \ldots, N_{mb}$$

$$M_{\phi_f \phi_f} = F_{19,n}$$

$$F_{\phi_f \phi_T} = F_{61,n} \quad .$$

The following partial derivatives are also required.

$$\left\{\frac{\partial \phi_f}{\partial q_{SP}}\right\}_n$$ the partial of feathering angle wrt the generalized swashplate coordinates

If $N_{SP} = 1$,

$$\left\{\frac{\partial \phi_f}{\partial q_{SP}}\right\} = \left\{\frac{\partial \phi_f}{\partial Z_{SP}}\right\}$$

If $N_{SP} = 2$,

$$\left\{\frac{\partial \phi_f}{\partial q_{SP}}\right\} = \left\{\frac{\partial \phi_f}{\partial \phi_{SP}}\ ,\ \frac{\partial \phi_f}{\partial \theta_{SP}}\right\}$$

If $N_{SP} = 3$,

$$\left\{\frac{\partial \phi_f}{\partial q_{SP}}\right\} = \left\{\frac{\partial \phi_f}{\partial \phi_{SP}}\ ,\ \frac{\partial \phi_f}{\partial \theta_{SP}}\ ,\ \frac{\partial \phi_f}{\partial Z_{SP}}\right\}$$

Also, the partial of the feathering angle wrt the generalized rotor coordinate is

$$\left\{\frac{\partial \phi_f}{\partial qr}\right\}_n = \left\{0,\ 0,\ 0,\ 0,\ 0,\ \frac{\partial \phi_f}{\partial \psi_R}\right\}_n$$

where

$$\frac{\partial \phi_f}{\partial \Psi_R} = A_{1S} \sin\left(\psi_{BL_n} + \psi_R\right) - B_{1S} \cos\left(\psi_{BL_n} + \psi_R\right)$$

The swashplate - swashplate matrix is

$$\left[M_{SWP}\right] = \left[M_{SPBS}\right] + \left[M_{SP}\right]_0$$

$$\left[M_{SPBS}\right] = \sum_{n=1}^{N_b} \left\{\frac{\partial \phi_f}{\partial q_{SP}}\right\}_n^T \left[M_{\phi_f \phi_f}\right] \left\{\frac{\partial \phi_f}{\partial q_{SP}}\right\}$$

$$\left[M_{SP}\right]_0 = \left\{\frac{\partial \rho_{SP}}{\partial q_{SP}}\right\}^T \left[M_{OSP}\right] \left\{\frac{\partial \rho_{SP}}{\partial q_{SP}}\right\}$$

The swashplate coordinate set is not standard nor are they grouped in a standard order. With this in mind, the following mass matrix and partial derivative matrix are based on the coordinate definitions:

$$\rho_{SP} = \left\{\phi,\ \theta,\ \psi,\ Z\right\}_{SP}$$

$$\tau_{SP} = \left\{\phi,\ \theta,\ Z\right\}_{SP} \quad .$$

$$\left[M_{OSP}\right] = \begin{bmatrix} I_{XX_{SP}} & & & 0 \\ & I_{YY_{SP}} & & \\ & & I_{ZZ_{SP}} & \\ 0 & & & m_{SP} \end{bmatrix}$$

$$\left\{\frac{\partial \rho_{SP}}{\partial \tau_{SP}}\right\} = \begin{bmatrix} \cos\theta_{SP} & 0 & 0 \\ 0 & 1 & 0 \\ \sin\theta_{SP} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The program implementation of the above definitions follows. If $N_{SP} = 1$, i.e., Z_{SP} coordinate only, then

$$\underset{(1 \times 1)}{\left[M_{SP}\right]_0} = m_{SP}$$

$$\underset{(1 \times 1)}{\left[M_{SPBS}\right]} = \sum_{n=1}^{N_b} \left(\frac{\partial \phi_f}{\partial Z_{SP}}\right)^2 \left[M_{\phi_f \phi_f}\right]_n$$

otherwise *in symmetric storage mode*

$$\left[M_{SP}\right]_0 = M \begin{bmatrix} (1) & & \\ (2) & (3) & \\ (4) & (5) & (6) \end{bmatrix} ; \quad \begin{aligned} (1) &= I_{XX_{SP}} \cos^2 \theta_{SP} + I_{ZZ_{SP}} \sin^2 \theta_{SP} \\ (2) &= 0 \\ (3) &= I_{YY_{SP}} \triangleq I_{XX_{SP}} \\ (4) &= 0 \\ (5) &= 0 \\ (6) &= m_{SP} \end{aligned}$$

In symmetric storage mode, M_{SPBS} *is a vector of length* $N_{SP} \cdot (N_{SP} + 1)/2$

$$M_{SPBS}(L) = \sum_{n=1}^{N_b} \left[\frac{\partial \phi_f}{\partial q_{SP}}\right]^T_{j,n} \left[M_{\phi_f \phi_f}\right] \left[\frac{\partial \phi_f}{\partial q_{SP}}\right]_{k,n}$$

$$j = 1, \ldots, N_{SP}$$

$$k = 1, \ldots, j$$

$$L = j \cdot (j - 1)/2 + k$$

Finally

$$\left[M_{SWP}\right]_m = \left[M_{SP_0}\right]_m + \left[M_{SPBS}\right]_m$$

$$m = 1, \ldots, N_{SP}(N_{SP} + 1)/2$$

The remaining output matrices are

$$\left[M_{HSP}\right] = \left[M'_{HSP}\right] + \left\{\frac{\partial \tau_{OR}}{\partial \tau_H}\right\}^T \left[M_{ORSP}\right]$$

$$\left[M_{ORSP}\right] = \sum_{n=1}^{N_b} \left\{\frac{\partial \tau_{OBL}}{\partial \tau_{OR}}\right\}^T \left[M_{OBL}\ \phi_f\right]_n \left\{\frac{\partial \phi_f}{\partial q_{SP}}\right\}_n$$

$$+ \sum_{n=1}^{N_b} \left\{\frac{\partial \phi_f}{\partial qr}\right\}_n^T \left[M_{\phi_f \phi_f}\right]_n \left\{\frac{\partial \phi_f}{\partial q_{SP}}\right\}_n$$

$$\left[M'_{HSP}\right] = \left\{\frac{\partial \rho_{SP}}{\partial \tau_H}\right\}^T \left[M_{OSP}\right] \left\{\frac{\partial \rho_{SP}}{\partial q_{SP}}\right\}$$

$$\left\{\frac{\partial \rho_{SF}}{\partial \tau_H}\right\} = \left[\begin{array}{ccc|ccc} & 0 & & & \left[T_H \rightarrow SP\right] & \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right]$$

The partials are not defined explicitly. The particular form of $[M'_{HSP}]$ will be determined as follows.

If NSP = 1 then

$$\left[M'_{HSP}\right] = \left\{\frac{\partial \rho_{SP}}{\partial \tau_H}\right\}^T \left[M_{OSP}\right] \left\{\frac{\partial \rho_{SP}}{\partial q_{SP}}\right\} = \left\{\begin{matrix} 0 \\ 0 \\ m_{SP} \\ 0 \\ 0 \\ 0 \end{matrix}\right\}$$

otherwise define

$$\underset{(6 \times 3)}{\left[M''_{HSP}\right]} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & m_{SP} \\ I_{XX_{SP}} T_{HSP_{11}} \cos\theta_{SP} + I_{ZZ_{SP}} T_{HSP_{31}} \sin\theta_{SP} & I_{YY_{SP}} T_{HSP21} & 0 \\ I_{XX_{SP}} T_{HSP_{12}} \cos\theta_{SP} + I_{ZZ_{SP}} T_{HSP_{32}} \sin\theta_{SP} & I_{YY_{SP}} T_{HSP22} & 0 \\ I_{XX_{SP}} T_{HSP_{13}} \cos\theta_{SP} + I_{ZZ_{SP}} T_{HSP_{33}} \sin\theta_{SP} & I_{YY_{SP}} T_{HSP23} & 0 \end{bmatrix}$$

and the final

$$\underset{6 \times N_{SP}}{\left[M'_{HSP}\right]} = \underset{6 \times N_{SP}}{\left[M''_{HSP}\right]} \quad .$$

The matrix $[M_{OBL}\Phi_f]$ is defined in ZERORM. The partial $\{\partial\tau_{OPL}/\partial\tau_{OR}\}_n$ is not available explicitly but is defined as

$$\left\{ \frac{\partial \tau_{OBL}}{\partial \tau_{OR}} \right\}_n = \left[\begin{array}{c|c} \left[T_{R-BL} \right]_n & 0 \\ \hline 0 & \left[T_{R-BL} \right]_n \end{array} \right]$$

6.7 SHAFTM

Subprogram SHAFTM computes the mass matrices required to represent the shaft as a model subsystem. Matrices input include $[M_{HBL}]$, $[M_{HSP}]$, $[M_{ORH}]$ and $[M_H]$. Matrices output are defined below.

$$\left[M_{SBL} \right] = \left\{ \frac{\partial \tau_{OH}}{\partial \tau_S} \right\}^T \left[M_{HBL} \right]$$

$$\left[M_{SSP} \right] = \left\{ \frac{\partial \tau_{OH}}{\partial \tau_S} \right\}^T \left[M_{HSP} \right]$$

$$\left[M_S \right] = \left\{ \frac{\partial \tau_{OH}}{\partial \tau_S} \right\}^T \left[M_{HS} \right]$$

$$\left[M_{HS} \right] = \left[M_H \right] \left\{ \frac{\partial \tau_{OH}}{\partial \tau_S} \right\}$$

$$\left[M_{ORS} \right] = \left[M_{ORH} \right] \left\{ \frac{\partial \tau_{OH}}{\partial \tau_S} \right\}$$

The partial $\{\partial \tau_{OH} / \partial \tau_S\}$ is defined in PDERIV.

6.8 REFM

Subprogram REFM computes the mass matrices necessary to describe the reference subsystem. Input matrices include $[M_{HBL}]$, $[M_{HSP}]$, $[M_{HS}]$, $[M_{ORH}]$, $[M_H]$, and $[M_O]_F$. Output matrices are defined below.

$$\left[M_{REFBL}\right] = \left\{\frac{\partial\tau_{OH}}{\partial\tau_{REF}}\right\}^T \left[M_{HBL}\right]$$

$$\left[M_{REFSP}\right] = \left\{\frac{\partial\tau_{OH}}{\partial\tau_{REF}}\right\}^T \left[M_{HSP}\right]$$

$$\left[M_{REFS}\right] = \left\{\frac{\partial\tau_{OH}}{\partial\tau_{REF}}\right\}^T \left[M_{HS}\right]$$

$$\left[M_{REF}\right] = \left\{\frac{\partial\tau_{OF}}{\partial\tau_{REF}}\right\}^T \left[M_O\right]_F \left\{\frac{\partial\tau_{OF}}{\partial\tau_{REF}}\right\} + \left\{\frac{\partial\tau_{OH}}{\partial\tau_{REF}}\right\}^T \left[M\right]_{HREF}$$

$$\left[M_{ORREF}\right] = \left[M_{ORH}\right] \left\{\frac{\partial\tau_{OH}}{\partial\tau_{REF}}\right\}$$

$$\left[M_{HREF}\right] = \left[M_H\right] \left\{\frac{\partial\tau_{OH}}{\partial\tau_{REF}}\right\}$$

The matrix $[M_O]_F$ is a constant and computed once in INIT. The partial $\{\partial\tau_{OH} / \partial\tau_{REF}\}$ is computed in PDERIV. The partial connecting the fuselage to the reference system is defined as

$$\left\{\frac{\partial \tau_{OF}}{\partial \tau_{REF}}\right\} = [I]$$

6.9 ROTORM

Subprogram ROTORM computes the mass matrices necessary to describe the rotor subsystem. The subsystem matrices are defined below followed by implementation information.

$$\left[M_{RBL}\right] = \left\{\frac{\partial \tau_{OR}}{\partial q_R}\right\}^T \left[M_{ORBL}\right]$$

$$\left[M_{RSP}\right] = \left\{\frac{\partial \tau_{OR}}{\partial q_R}\right\}^T \left[M_{ORSP}\right] + \left[M'_{RSP}\right]$$

where

$$\left[M'_{RSP}\right] = \left\{\frac{\partial \tau_{OSP}}{\partial q_r}\right\}^T \left[M_{OSP}\right] \left\{\frac{\partial \tau_{OSP}}{\partial q_{SP}}\right\}$$

$$\left[M_{RS}\right] = \left\{\frac{\partial \tau_{OR}}{\partial q_R}\right\}^T \left[M_{ORS}\right] + \left\{\frac{\partial \tau_{OH}}{\partial q_R}\right\}^T \left[M_{OH}\right] \left\{\frac{\partial \tau_{OH}}{\partial \tau_S}\right\}$$

$$+ \left\{\frac{\partial \tau_{OSP}}{\partial q_R}\right\}^T \left[M_{OSP}\right] \left\{\frac{\partial \tau_{OSP}}{\partial \tau_S}\right\}$$

$$\left[M_{RREF}\right] = \left\{\frac{\partial\tau_{OR}}{\partial q_R}\right\}^T \left[M_{ORREF}\right] + \left\{\frac{\partial\tau_{OH}}{\partial q_R}\right\}^T \left[M_{OH}\right] \left\{\frac{\partial\tau_{OH}}{\partial\tau_{REF}}\right\}$$

$$+ \left\{\frac{\partial\tau_{OSP}}{\partial q_R}\right\}^T \left[M_{OSP}\right] \left\{\frac{\partial\tau_{OSP}}{\partial\tau_{REF}}\right\}$$

$$+ \left\{\frac{\partial\tau_{ENG}}{\partial q_R}\right\}^T \left[M_{ENG}\right] \left\{\frac{\partial\tau_{ENG}}{\partial\tau_{REF}}\right\} + \left\{\frac{\partial\tau_{TR}}{\partial q_F}\right\}^T \left[M_{TR}\right] \left\{\frac{\partial\tau_{TR}}{\partial\tau_{REF}}\right\}$$

$$\left[M_R\right] = \left\{\frac{\partial\tau_{OR}}{\partial q_R}\right\}^T \left[M_{OR}\right] \left\{\frac{\partial\tau_{OR}}{\partial q_R}\right\} + \left\{\frac{\partial\tau_{OH}}{\partial q_R}\right\}^T \left[M_{OH}\right] \left\{\frac{\partial\tau_{OH}}{\partial q_R}\right\}$$

$$+ \left\{\frac{\partial\tau_{OSP}}{\partial q_R}\right\}^T \left[M_{OSP}\right] \left\{\frac{\partial\tau_{OSP}}{\partial q_R}\right\} + \left\{\frac{\partial\tau_{ENG}}{\partial q_R}\right\}^T \left[M_{ENG}\right] \left\{\frac{\partial\tau_{ENG}}{\partial q_R}\right\}$$

$$+ \left\{\frac{\partial\tau_{TR}}{\partial q_R}\right\}^T \left[M_{TR}\right] \left\{\frac{\partial\tau_{TR}}{\partial q_R}\right\}$$

The following matrices are input, $[M_{ORBL}]$, $[M_{ORSP}]$, $[M_{ORS}]$, $[M_{ORREF}]$, and $[M_{OR}]$. Other matrices and partial derivatives are defined below. Finally, the implementation of the individual terms in the above equations are presented. Partials of interest include

$$\left\{\frac{\partial\tau_{OR}}{\partial q_R}\right\} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{Bmatrix} \quad , \left\{\frac{\partial\tau_{OH}}{\partial q_R}\right\} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{Bmatrix}$$

$$\left\{\frac{\partial \tau_{ENG}}{\partial q_R}\right\} = \left\{\begin{matrix} 0 \\ 0 \\ 0 \\ -G_{ENG} \\ 0 \\ 0 \end{matrix}\right\} \quad , \quad \left\{\frac{\partial \tau_{TR}}{\partial q_R}\right\} = \left\{\begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ G_{TR} \\ 0 \end{matrix}\right\}$$

$$\left\{\frac{\partial \tau_{OSP}}{\partial q_R}\right\} = \left\{\frac{\partial \tau_{OSP}}{\partial \tau_H}\right\} \left\{\frac{\partial \tau_{OH}}{\partial q_R}\right\} = \left\{\begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -\cos \phi_{SP} \cos \theta_{SP} \end{matrix}\right\}$$

$$\left\{\frac{\partial \tau_{OSP}}{\partial \tau_S}\right\} = \left\{\frac{\partial \tau_{OH}}{\partial \tau_S}\right\} \left\{\frac{\partial \tau_{OSP}}{\partial \tau_H}\right\}$$

$$\left\{\frac{\partial \tau_{ENG}}{\partial \tau_{REF}}\right\} = [I] \quad , \quad \left\{\frac{\partial \tau_{T}}{\partial \tau_{REF}}\right\} = [I] \quad .$$

Mass definitions include

$$\left[M_{ENG}\right] = \begin{bmatrix} 0 & & & & & \\ & 0 & & & & \\ & & 0 & & & \\ & & & I_{XX_{ENG}} & & \\ & & & & 0 & \\ & & & & & 0 \end{bmatrix}$$

$$\left[M_{TR} \right] = \begin{bmatrix} 0 & & & & & \\ & 0 & & & & \\ & & 0 & & & \\ & & & 0 & & \\ & & & & I_{YY_{TR}} & \\ & & & & & 0 \end{bmatrix} .$$

Terms are now defined as

$$\left\{ \frac{\partial \tau_{OSP}}{\partial q_R} \right\}^T \left[M_{OSP} \right] \left\{ \frac{\partial \tau_{OSP}}{\partial \tau_S} \right\} = \left\{ 0,\ 0,\ 0,\ 0,\ 0,\ - I_{ZZ} \cos \phi_{SP} \cos \theta_{SP} \right\}$$

$$\left[\begin{array}{c|c} \left[T_{MB-H} \right] & \left[T_{MB-H} \right] \begin{bmatrix} 0 & -2 & -Y \\ -2 & 0 & X \\ Y & -X & 0 \end{bmatrix}_{M_{B-H}} \\ \hline [0] & \begin{bmatrix} \cos \psi_S \cos \theta_S & \sin \psi_S & 0 \\ -\sin \psi_S \cos \theta_S & \cos \psi_S & 0 \\ \sin \theta_S & 0 & 1 \end{bmatrix} \end{array} \right] \left[\begin{array}{c|c} [I] & \left[\quad \right] \\ \hline [0] & \left[T_{SP-H} \right] \end{array} \right]$$

$$= \left\{ 0,\ 0,\ 0,\ -I_{ZZ_{SP}} \sin \theta_S \cos \phi_{SP} \cos \theta_{SP} \sin \theta_{SP},\ 0,\ -I_{ZZ_{SP}} \cos^2 \phi_{SP} \cos^2 \theta_{SP} \right\}$$

Note the forth element is approximately Zero.

$$\left\{ \frac{\partial \tau_{OH}}{\partial q_R} \right\}^T \left[M_{OH} \right] \left\{ \frac{\partial \tau_{OH}}{\partial \tau_S} \right\} = \left\{ 0,\ 0,\ 0,\ 0,\ 0,\ - I_{ZZ_H} \right\} \left\{ \frac{\partial \tau_{OH}}{\partial \tau_S} \right\}$$

$$= \left\{ 0,\ 0,\ 0,\ 0,\ 0,\ - I_{ZZ_H} \right\}$$

$$\left[M'_{RSP}\right] = \left\{- I_{ZZ_{SP}} \cos\phi_{SP} \sin\theta_{SP} \cos\theta_{SP}, 0, 0\right\}$$

$$\left\{\frac{\partial\tau_{ENG}}{\partial q_R}\right\}^T \left[M_{ENG}\right] \left\{\frac{\partial\tau_{ENG}}{\partial\tau_{REF}}\right\} = \left\{0, 0, 0, -G_{ENG} I_{XX_{ENG}}, 0, 0\right\}$$

$$\left\{\frac{\partial\tau_{TR}}{\partial q_R}\right\}^T \left[M_{TR}\right] \left\{\frac{\partial\tau_{TR}}{\partial\tau_{REF}}\right\} = \left\{0, 0, 0, 0, G_{TR} I_{YY_{TR}}, 0\right\}$$

$$\left\{\frac{\partial\tau_{ENG}}{\partial q_R}\right\}^T \left[M_{ENG}\right] \left\{\frac{\partial\tau_{ENG}}{\partial q_R}\right\} = G_{ENG}^2 I_{XX_{ENG}}$$

$$\left\{\frac{\partial\tau_{TR}}{\partial q_R}\right\}^T \left[M_{TR}\right] \left\{\frac{\partial\tau_{TR}}{\partial q_R}\right\} = G_{TR}^2 I_{YY_{TR}}$$

$$\left\{\frac{\partial\tau_{OSP}}{\partial q_R}\right\}^T \left[M_{OSP}\right] \left\{\frac{\partial\tau_{OSP}}{\partial q_R}\right\} = I_{ZZ_{SP}} \cos^2\phi_{SP} \cos^2\theta_{SP}$$

$$\left\{\frac{\partial\tau_{OH}}{\partial q_R}\right\}^T \left[M_{OH}\right] \left\{\frac{\partial\tau_{OH}}{\partial q_R}\right\} = I_{ZZ_H}$$

$$\left\{\frac{\partial\tau_{OSP}}{\partial q_R}\right\}^T \left[M_{OSP}\right] \left\{\frac{\partial\tau_{OSP}}{\partial\tau_{REF}}\right\} = \Bigg\{0, 0, 0, 0, 0,$$

$$- I_{ZZ_{SP}} \cos\phi_{SP} \cos\theta_{SP}\Bigg\}$$

$$\left\{\frac{\partial\tau_{OSP}}{\partial\tau_H}\right\} \left\{\frac{\partial\tau_{OH}}{\partial\tau_{RF}}\right\}$$

$$= \left\{0, 0, 0, 0, 0, - I_{ZZ_{SP}} \cos^2\phi_{SP} \cos^2\theta_{SP} \cos\theta_{OS} \cos\theta_{OS} \cos\theta_S \cos\phi_S\right\}$$

$$\left\{\frac{\partial \tau_{OH}}{\partial q_R}\right\}^T \left[M_{OH}\right]\left\{\frac{\partial \tau_{OH}}{\partial \tau_{REF}}\right\} = \Big\{0,\ 0,\ 0,\ 0,\ 0,$$

$$- I_{ZZ_H} \cos \phi_S \cos \theta_S \cos \phi_{OS} \cos \theta_{OS}\Big\}$$

7. MATRIX ROUTINES

Throughout the documentation, reference is made to Full Storage Mode and Symmetric Storage Mode. This refers to the manner in which a matrix resides in core storage. A matrix stored in Full Storage Mode assumes the normal storage attributes as dictated by FORTRAN for two-dimensional arrays.

Symmetric Storage Mode is used in order to minimize memory requirements. Only the elements on and below the main diagonal of symmetric matrices are stored. The order and occurence of these elements in core are as follows: Assume [A] is a symmetric matrix and [B] is a vector. Then

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \longrightarrow \begin{bmatrix} a_{11} & & \\ a_{21} & a_{22} & \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \longrightarrow \begin{bmatrix} a_{11} \\ a_{21} \\ a_{22} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix} = [B]$$

An n by n matrix in Symmetric Storage Mode is reduced to a vector of length $n(n + 1)/2$ where the element ij can be found as the element k of the vector B, where $k = (i\ (i - 1)/2) + j$ for $i \geq j$. Storing a matrix in this manner effects a savings of $n\ (n - 1)/2$ core locations.

The Cholesky decomposition routine, CHOSKY, is specifically designed to operate on a matrix in Symmetric Storage Mode as is the output routine, MXOUT.

The routine, CONMFS, will convert a matrix from Full to Symmetric Storage Mode. The remaining routines, MXMULF and MTMULF are standard matrix multiply routines.

7.1 CHOSKY

Within REXOR II, the equations of motion are stated

$$\left\{ \ddot{q} \right\} = \left\{ \Delta \ddot{q} \right\} + \left\{ \ddot{q}_{EST} \right\}$$

where

$$\{\Delta\ddot{q}\} = [M]^{-1} \{\Delta F\} \quad ,$$

M and ΔF are given. $\Delta\ddot{q}$ can be computed by first inverting M then performing the indicated multiplication. A more efficient method is to solve for the product directly by solving the linear system

$$[M] \{\Delta\ddot{q}\} = \{\Delta F\}$$

for $\{\Delta\ddot{q}\}$. It is further known that the mass matrix, [M], is positive definite and symmetric. Subprogram CHOSKY is a general algorithm for the solution of simultaneous equations of the form

$$[A] \{X\} = \{b\}$$

where the coefficient matrix is positive definite, symmetric. The algorithm is a Cholesky decomposition of [A]; followed by a forward-backward substitution. The algorithm is presented below.

Cholesky method for symmetric, positive definite matrices.

Theorem: Let A be symmetric, positive definite. Then A can be factored in the form

$$LL^T = A$$

where L is a lower triangular matrix (i.e., $L = (l_{ij})$ where $l_{ij} = 0$ for $j > i$).

Cholesky Method: Let A be n x n, symmetric, positive definite

$$A = (a_{ij}) \quad , a_{ij} = a_{ji} \quad .$$

Assume A is factorable

$$A = LU \quad .$$

Then

$$a_{ij} = \sum_{k=1}^{n} \ell_{ik}\, u_{kj} \quad .$$

If

$$U = L^T,\ u_{kj} = \ell_{jk}$$

then

$$a_{ij} = \sum_{k=1}^{n} \ell_{ik}\, \ell_{jk} \qquad , i = 1, \ldots, n,\ j = 1, \ldots, n \quad .$$

I t L is lower triangular which implies $\ell_{ik} = 0$ for $k > i$. Therefore,

$$a_{ij} = \sum_{k=1}^{i} \ell_{ik}\, \ell_{jk} \quad ; \ i = 1, \ldots, n,\ j = i, \ldots, n \quad .$$

This equation forms the basis of the decomposition. The elements of L are found as follows.

$$\ell_{11} = \sqrt{a_{11}}$$

also

$$a_{1j} = \ell_{11}\, \ell_{j1}$$

leads to

$$\ell_{j1} = a_{1j}/\ell_{11} \qquad , j = 2, \ldots, n$$

and

$$a_{ii} = \sum_{k=1}^{i} \ell^2_{ik} = \sum_{k=1}^{i-1} \ell^2_{ik} + \ell^2_{ii} \qquad , i = 2, \ldots, n$$

can be solved for ℓ_{ii}. Therefore

$$\ell_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} \ell^2_{ik}}$$

Finally,

$$a_{ij} = \sum_{k=1}^{i-1} \ell_{ik} \, \ell_{jk} + \ell_{ii} \, \ell_{ji} \qquad , i = 2, \ldots, n$$

giving

$$\ell_{ji} = \left(a_{ij} - \sum_{k=1}^{i-1} \ell_{ik} \, \ell_{jk} \right) \Big/ \ell_{ii} \qquad , j = i + 1, \ldots, n$$

With the decomposition of A accomplished, the system of interest can be solved as follows.

Given

$$A X = b \quad .$$

Substitute

$$A = LL^T$$

giving

$$LL^T X = b \quad .$$

Define

$$g = L^T X$$

Solve $Lg = b$ by forward substitution.

Namely

$$g_1 = b_1/\ell_{11}$$

and

$$g_i = \left(b_i - \sum_{k=1}^{i-1} \ell_{ik}\, g_k\right) \Big/ \ell_{ii}, \qquad i = 2, \ldots, n$$

Finally, X can be found by backward substitution.

$$X_n = g_n/\ell_{nn}$$

$$X_j = \left(g_j - \sum_{k=j+1}^{n} \ell_{kj}\, X_k\right) \Big/ \ell_{jj} \qquad , j = r - 1, \ldots, 1$$

The above equations form the algorithm.

The arguments in the CHOSKY subprogram calling sequence are

Arguments List

A	An input vector of length $N(N + 1)/2$ containing the $N \times N$ positive definite symmetric matrix stored in symmetric storage mode form.
B	Input matrix of dimension $N \times L$ containing the right hand sides of the equation $AX = B$. On output, the $N \times L$ solution matrix (X) replaces (B).
N	Order of [A] and the number of rows of B.
M	Row definition of B in calling program.
L	Number of right hand sides (columns in B).
IER	Error indicator = 0 normal return > 0 indicates the input matrix is algorithmically not positive definite. IER with contain the index of the row which failed.

The positive definite test is as follows. While forming the elements of the decomposed matrix, L, the argument of the square root function is tested. If

$$a_{ii} - \sum_{k=1}^{i-1} \ell_{ik}^2 < 0$$

then the procedure is terminated. The coefficient matrix, A, is destroyed on exit.

The subprogram CHOSKY is computed in double precision on IBM hardware. This means the matrices A and B are double precision. The CDC version is single precision.

7.2 CONMFS

This routine is a matrix storage mode conversion routine. It converts a full symmetric matrix to a matrix in symmetric storage mode.

<u>Arguments List</u>

A	Input matrix of dimension N x N.
N	Order of matrix A.
IA	Row dimension of A as specified in the calling program.
B	Output vector of dimension N · (N + 1)/2 containing matrix A in symmetric storage mode.

An n x n matrix in symmetric storage mode is reduced to a vector of length $n(n + 1)/2$ where the element ij can be found as the element k of the vector B, where $k = (i\ (i - 1)/2) + j$ for $i \geq j$.

7.3 MXOUT

This routine prints a matrix which is stored in symmetric storage mode.

<u>Arguments List</u>

A	Name of matrix to be printed.
M	Number of rows and columns of A.
K	Number of lines already printed.

Output is to FORTRAN unit 6. The routine will page eject after 55 lines.

7.4 MXMULF

MXMULF is a matrix multiplication routine, where the matrices are in Full Storage Mode.

$$[C] = [A] [B]$$

<u>Arguments List</u>

A	(LXM) matrix
B	(MXN) matrix

L	Maximum value of first subscript of A
M	Maximum value of second subscript of A and first subscript of B
N	Maximum value of second subscript of B
IA	Value of first index in DIMENSION statement for A
IB	Value of first index in DIMENSION statement for B
C	Resultant matrix, (LXN)
IC	Value of first index in DIMENSION statement for C

The multiplication is performed in double precision.

7.5 MTMULF

This subprogram performs a matrix multiplication of the transpose of a matrix A by a matrix B. The matrices are in Full Storage Mode.

$$[C] = [A]^T [B]$$

Arguments List

A	(LXM) input matrix
B	(LXN) input matrix
L	Number of rows in A and B
M	Number of columns in A and rows in C
N	Number of columns in B and C
IA	Row dimension of A as specified in calling program
IB	Row dimension of B
C	Resultant matrix (MXN)
IC	Row dimension of C in calling program

The multiplication is performed in double precision.

8. PROGRAM MODIFICATION HINTS

This section is not intended to be a comprehensive programming guide, but rather a few hints concerning the extension of the program in the area of input/output.

8.1 Adding Parameters to Input Set

All program data which is user changeable via input resides in a data array, RA, which is assigned to the labeled COMMON/INPUT/ as follows

```
COMMON/INPUT/RA(3000), TITLES(15,4)
```

Not all of the 3000 addresses have been assigned to input. This can be seen in Table 3-1 of Volume III, where all addresses are defined. Those which are undefined as of this writing are indicated as "open". If one wishes to define a new input, find an undefined address in the table and equivalence the new parameter to the appropriate address. For example, assume a new input is required, XNEW, and an open address is found, 20. Then simply declare

```
EQUIVALENCE (RA(20), XNEW)
```

in those routines necessary. A copy of the labeled COMMON statement

```
COMMON/INPUT/RA(3000), TITLES(15,4)
```

is also required. Another fact should be pointed out. The array, RA, is a REAL variable and all inputs are read as REAL data. If an input is required in integer form, then define the new parameter across an equal sign,

```
IP = RA(20)
```

Accurate records of assigned addresses should be kept to avoid conflicts.

8.2 Adding Variables to Trim Set

Program variables available for trim control or function definition are found in the table which drives the RETRVE/STORE routine. The table is a form of indirect addressing. The table can be easily extended. Currently, variables defined in blank COMMON and /INPUT/ labeled COMMON can be accommodated by extending the table and indicating in which COMMON the new variable resides. Also include the address within that COMMON. A study of Figure 12 will indicate the simple coding change required to add a new labeled COMMON to the table.

8.3 Adding Quantities to Output Set

The general mechanism by which the output signal set is defined has been discussed elsewhere in this volume as well as in Volume III. The key item to remember is that any program variable defined in blank COMMON can be chosen for the signal set.

Blank COMMON contains all the program variables considered output material as of this writing. A mechanism for extending the set has been provided, namely, the vector OUTD at the end of COMMON and the subprogram AUXOUT.

The auxillary output routine, AUXOUT, defines elements of the output vector, OUTD. To include a new output, define a heretofore undefined element of OUTD. Thus, its output identification is defined. It should be stressed that blank COMMON should not be redefined in any way. Any alteration of the order of variables in COMMON will invalidate current output identifications.

9. PROGRAM CROSS REFERENCE MAP

The collection of subprograms which constitute REXOR II contain many cross-referenced COMMON blocks and subroutines. Figure 30 is presented as an aid to developing the source and usage of any particular item. The vertical listing in this figure gives all the routines, and names or unnamed (one) COMMON blocks used in REXOR II. A second column, headed 'T', gives the use of the entry. The coding used is 'M' for main program, 'S' for subroutine, 'E' for entry of a subroutine, 'C' for common block. The list of names is in alphabetical order and numbered. The alphabetizing is by main to subroutines with a subalphabetizing of entry points under each subroutine. The COMMON blocks are listed last except for subroutines not included in the source deck. These subroutines are usually part of the computer operating (precompiled) package, and not particularly associated with REXOR II. The numbering is repeated horizontally, and corresponds to the vertical name list.

The vertical list on the left-hand side is the calling or active routine or element, and the horizontal line lists the routines called or referenced. Numbers at grid intersections show there is a reference and the level of reference. One indicates a direct reference. Two or three show there are one or two intermediate references, respectively. Note that a subroutine name will show all the references to all the entry points bounded by that name. By elimination, the references associated with the subroutine name up to the point of the first entry name can be determined.

NO	ROUTINE	T	1 12345 67890	11111 11112 12345 67890	22222 22223 12345 67890	33333 33334 12345 67890	44444 44445 12345 67890	55555 55556 12345 67890	66666 66667 12345 67890	77777 77778 12345 67890	88888 88889 12345 67890	99999 9999A 12345 67890	AAAAA AAAAB 12345 67890	BBBBB B 12345 6
1	MAIN	M	2222 33222	22332 21312	22122 21132	22222 31322	23122 22221	23322 33323	33233 33211	22232 22222	22222 22221	11111 12222	22222 22222	22222 2
2	ABLXCH	S												
3	ACCEL	S	22 12	2 2 2 2	1 1 22	2 2 2	22 2	22 222 2	2 222 2	22 22222	22222 2222	11222 2222	222 2	2
4	AWAVEC	S												
5	AUXOUT	S										1 1 11		
6	BLADSP	S										111 1 1	1	
7	BLADSH	S		1								11 1		
8	BSCALE	S												
9	CHOSEY	S												
10	CLCOCH	S										1		
11	CMPRES	S												
12	CNTROL	S						1				1 1		
13	CONNFS	S												
14	COORD	S		1								111	1	
15	COULOM	S												
16	CYCLIC	S										1 1 1	1	
17	DATAP	S	1					2				2		
18	DMSTOR	S												
19	DTPREP	S										11	11	
20	DWASH	S										1 1 1	1	
21	ETORQ	S										1 1	1	
22	EULERY	S												
23	FLY	S	121 33 23	31313 1 3 1	12 2 2 33	1112 3 313	33 11 1	3322 33333	31333 3	11123 33333	33333 3333	11333 2133	3321 32	2
24	FORIT	S												
25	GFORCE	S	1 2	2	1	2 2	2	1 1 1	21 1	22 22222	22222 222	112 1 2222	222 2	2
26	GMASS	S	2	2 2 2	2	1 2	21 1	2 2 2	2 2		2	2111 2 2	2 2	
27	GRAPH	S	2				1					1111 2	11 1 2	22
28	HARM	S			1		1					11 1 1		
29	HUMP	S				1	1					111	1 1	
30	HUBM	S				1	1					111 1	1 1	
31	INIT	S		1	1							111 11	1 1	
32	INTGAX	S												
33	INTGBC	S											1	
34	JETTN	S										111	11	
35	LINVEC	S												
36	LOADS	S										1 1 1	1 1	
37	MASBEM	S	12 33 23	31313 1 3	2 1 2 33	2 3 313	33 31	3322 33323	3 333 3	1 123 33333	33333 3333	11133 2133	332 3	2
38	MASSSP	S	1	2 2 1	1	2	2	1 2 1	1 1		1	212 2 2	2 2	
39	MOTION	S	1			1 1	1					1 1 1	1	
40	MTMULP	S												
41	MXMULP	S												
42	MXOUT	S												
43	OUTPUT	S					1					11 1		
44	PAGE	S												
45	PDERIV	S					1					111 1 1	1 1	
46	PHORN	S										111		
47	PRINTD	S					1					1	1	
48	PRINTC	S					1					1	1	
49	RCPLOT	S	1									11 1	1 1 1	11
50	READIN	S					2 11					1 1	2	1
51	REDATB	S												
52	REFR	S				1 1	1					1 1 21	1 2	
53	REFH	S		1		1	1					1 1	1	
54	RETRVE	S										1 1		
55	STORE	S										1 1		
56	RMSTOR	S												
57	ROTORF	S										111 1	1	
58	ROTORM	S		1								111 1	1	
59	SAS	S												
60	SHAFTF	S				1	1					11 1	1	

Figure 30. - Common subroutine directory.

ORIGINAL PAGE IS OF POOR QUALITY

61 SHAPTN	S		1		1	1					1 1		
62 SIGGEN	S										1 1		
63 STALL	S	1							1		1 2		1
64 SWASHP	S		1								111 1	1	
65 SWASHN	S				1	1					111 1 1	1 1	
66 SWEEP1	S	1	1			1		1	21 11111	11111 111	111 1 2111	11 1	2
67 SWEEP	S	1	1			1		1	21 11111	11111 111	111 1 2111	11 1	2
68 TABFIX	S												
69 TARGEN	S	1									2		1
70 TAROUT	S							1			1		
71 TFORM	S			1		1					11	1 1	
72 TIMS	S										111	1	
73 TRIM	S						11				111 11		
74 XTRP1	S												
75 XTERMO	S												
76 XTERM1	E												
77 XTERM2	E												
78 XTERM3	E												
79 XTERM4	E												
80 XTERM5	E												
81 XTERM6	E												
82 XTERM7	E												
83 XTERM8	E												
84 XTERM9	E												
85 XTERMA	E												
86 XTERMC	E												
87 XTERMD	E												
88 XTERMF	E												
89 ZERORM	S				1	1					11 1	1 1	
90 PDATE	S												1 1
91	C												
92 DATAS1	C												
93 INPUT	C												
94 DATEC	C												
95 ANSWLO	C												
96 LABELS	C												
97 TAB	C												
98 DATAS3	C												
99 CONSTS	C												
100 PARTLS	C												
101 DATAS2	C												
102 DATAS4	C												
103 DATAS6	C												
104 PLOTC	C												
105 PLOTS	S												
106 PLOT	S												
107 TFORMS	C												
108 BACKV	C												
109 NAMES	C												
110 SYMBOL	S												
111 NUMBER	S												
112 LINE	S												
113 RATNIT	C												
114 CLACLO	C												
115 TDATE	S												
116 MOVE	S												

Figure 30. - Concluded.

10. PROGRAM LISTING

Due to the large number of pages in the REXOR II program source listing, this material is handled under a separate binder. Copies may be obtained from the distributing agency, NASA, Langley Research Center, Hampton, Virginia.

11. REFERENCE

1. A. Ralston, H. Wilf, Mathematical Methods for Digital Computers, John Wiley and Sons, New York, 1960, Chapter 24.