NASA CONTRACTOR
REPORT

NASA CR-25

NASA CR-2583

AN ADAPTIVE MANEUVERING
LOGIC COMPUTER PROGRAM
FOR THE SIMULATION OF
ONE-ON-ONE AIR-TO-AIR COMBAT

Vol. II: Program Description

George H. Burgin and A. J. Owens

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • SEPTEMBER 1975

1. Report No. NASA CR-2583	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle AN ADAPTIVE MANEUVERING LOGIC COMPUTER PROGRAM FOR THE SIMULATION OF ONE-ON-ONE AIR-TO-AIR COMBAT - VOL. II: PROGRAM DESCRIPTION		5. Report Date SEPTEMBER 1975
		6. Performing Organization Code
7. Author(s) George H. Burgin and A. J. Owens		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address Decision Science Inc. 4508 Mission Bay Drive San Diego, California 92109		11. Contract or Grant No. NAS1-9115
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		13. Type of Report and Period Covered Contractor Report
		14. Sponsoring Agency Code

15. Supplementary Notes

FINAL REPORT

16. Abstract

A novel technique for computer simulation of air combat is described. Volume I describes the computer program and its development in general terms. Two versions of the program exist. Both incorporate a logic for selecting and executing air combat maneuvers with performance models of specific fighter aircraft. In the batch processing version the flight paths of two aircraft engaged in interactive aerial combat and controlled by the same logic are computed. The real-time version permits human pilots to fly air-to-air combat against the Adaptive Maneuvering Logic (AML) in Langley Differential Maneuvering Simulator (DMS). Volume II consists of a detailed description of the computer programs.

17. Key Words (Suggested by Author(s)) Air combat, differential games, simulation, Differential Maneuvering Simulator, control system	18. Distribution Statement Unclassified - Unlimited Subject Category 61		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 104	22. Price* $5.25

CONTENTS

AN ADAPTIVE MANEUVERING LOGIC COMPUTER PROGRAM

FOR THE SIMULATION OF ONE-ON-ONE AIR-TO-AIR COMBAT

Vol. II   Program Description

By George H. Burgin and A. J. Owens
Decision Science, Inc.

SUMMARY

This report supplements the information provided in the report: "An Adaptive Maneuvering Logic Computer Program for the Simulation of One-on-One Air-to-Air Combat. Volume I: General Description". It provides the user of the computer program sufficient detail to understand the mathematical and geometrical relationships as implemented in the computer program. It enables the use of the computer program as a research and design tool in air-to-air combat studies and provides the necessary information for program modifications, be it to represent different aircraft or weapons types or to modify the logic of the tactical decision process.

In a first section, the individual subroutines and their underlying mathematical relationships are described. A second section explains the required input data and the output provided by the program. A third section describes how the Adaptive Maneuvering Logic (AML) program interfaces with the program prepared by NASA personnel to drive the Differential Maneuvering Simulator (DMS) at the Langley Research Center (LRC).

INTRODUCTION

The purpose of Volume II of this report is to provide the user of the AML program enough details to completely understand the mathematical and geometrical details and to enable him to modify the program to accommodate new aircraft types or to make changes to the logic of the tactical decision process.

A prerequisite for understanding this report is familiarity with the concept of the adaptive maneuvering logic which may be found in Volume I of this report. Terms defined in Volume I will not be redefined here.

The off-line version of the AML program is described in detail. The DMS or real-time version is essentially the same program; the major difference is in the input-output sections. Aircraft initial conditions for individual runs are not read in from data cards on the DMS, but rather selected by the operator at the simulation control console. Also, during real-time operation on the DMS, no printout is prepared by the AML program. The DMS program records certain selected variables of the AML program.

To make it easier for the reader to understand the FORTRAN coding, in several places in this report FORTRAN variable names rather than symbols are used in the derivations.

It is suggested that the reader of this report consult the FORTRAN source program listing when reading the description of the individual subprograms. Numerous comments in this listing will facilitate the understanding of this report.

The off-line computer program contains many duplicate routines, one for the attacker, a second one for the target. All subroutines whose names end with a T (target) have an equivalent routine ending with an A for the attacker. In the DMS program the "target" subroutines are used. The subroutine descriptions in this volume are limited to the target routines with the understanding that an equivalent attacker routine exists in the off-

line version.

Write-ups of some simple routines and functions which are self-explanatory are not included. To facilitate referencing individual subroutines, their names are printed on each page in the upper right-hand corner.

Although some recent changes have been made in the program, this report describes the AML, in general, as it currently exists. It is to be expected that the program will undergo continual changes as required by the addition of new aircraft types and by increased understanding of the tactical decision process in air-to-air combat.

SYMBOLS

A	Matrix used in calculation of quaternion derivatives.
a_n	Normal acceleration of predicted turn.
$a_n = (a_{n1}, a_{n2}, a_{n3})^T$	Components of unit vector normal to maneuverplane in earth fixed coordinate system.
ATKR	Attacker.
a_{xe}, a_{ye}, a_{ze}	Acceleration of aircraft's c.g. along earth fixed x_e, y_e and z_e axes.
a_x, b_x, c_x a_y, b_y, c_y a_z, b_z, c_z	Coefficients for flight path extrapolation.
a_1, a_2, a_3, a_4	Quaternions.
b_1, b_2, b_3, b_4	Quaternions.
C_D	Coefficient of drag.
C_{D_D}	Coefficient of drag for fully deflected maneuvering flaps/slats.
C_{D_U}	Coefficient of drag for undeflected maneuvering flaps/slats.
C_L	Coefficient of lift.
C_{L_α}	Lift-curve slope.
c_s	Speed of sound
C_1, C_2, C_3, C_4	Initial values of quaternions.

4

D	Direction cosine matrix of body axes given.
$\overline{D}$	Direction cosine matrix of maneuverplane coordinate system.
$\tilde{D}$	Direction cosine matrix of wind axes system.
d_{ij}	Element of D.
$\overline{d}_{ij}$	Element of $\overline{D}$.
$\tilde{d}_{ij}$	Element of $\tilde{D}$.
dt	Integration step size.
F_{xw}, F_{yw}, F_{zw}	Components of aerodynamic and propulsive forces along x_w, y_w and z_w axes, respectively.
g	Acceleration due to earth gravity (32.17 ft/sec^2).
G_{mx}, G_{my}, G_{mz}	Components of gravity force vector along x_m, y_m and z_m axes, respectively.
H	Altitude. ($H = -z_e$)
I	Identity matrix.
K	Filter gain.
K_{BO}	Blackout function.
L	Lift vector or magnitude of lift.
LF_{mx}, LF_{my}, LF_{mz}	Components of lift vector along x_m, y_m and z_m axes, respectively.
M	Mach number.
m	Mass of aircraft.

5

M_c	Mach number at which speed brake deflection curve changes slope.
M_m	Mach number below which full speed brake deflection is allowed.
M_o	Mach number above which no speed brake deflection is allowed.
n_z	Normal acceleration
p, q, r	Body rotational rates about x_b, y_b and z_b axes, respectively.
$p_{max}, q_{max}, r_{max}$	Maximum permissible body rotational rates.
R	Radius of predicted turn.
R_T	Direction cosine matrix for consecutive rotation about z, y, x axes.
R_x	Direction cosine matrix for rotation about x-axis (roll).
R_y	Direction cosine matrix for rotation about y-axis (pitch).
R_z	Direction cosine matrix for rotation about z-axis (yaw).
T_p	Duration of predicted turn.
TRGT	Target.
t_o	Current time for extrapolation.
t_x', t_y', t_z'	Components of vector from own aircraft to opponent's extrapolated position in the own "unrotated" maneuverplane system.
t_x'', t_y'', t_z''	Components of vector from own aircraft to opponent's extrapolated position in the own maneuverplane coordinate system.

6

u, v, w	Velocity components in aircraft body axis system.
V_h	Component of velocity vector parallel to x_e-y_e plane.
x_b, y_b, z_b	Coordinates in body axes system: x_b along longitudinal axis, positive towards the nose, y_b pointing toward right wing tip, z_b pointing down.
x_e, y_e, z_e	Earth fixed coordinates: x_e positive north, y_e positive east, z_e positive down.
x_m, y_m, z_m	Coordinates in maneuverplane system: x_m aligned with velocity vector, $-z_m$ in maneuverplane towards concave side of flight path, y_m normal to maneuverplane.
x_m', y_m', z_m'	Coordinates in "unrotated" maneuverplane system: x_m' aligned with velocity vector, y_m' parallel to x_e-y_e plane, z_m' completing right handed coordinate system.
x_w, y_w, z_w	Coordinates in the wind axes system: x_w aligned with velocity vector, $-z_w$ direction of lift (for positive angle of attack), y_w completing right handed coordinate system.

$x_0,\ y_0,\ z_0$	Position at end of time interval used for extrapolation (current position).
$x_{-1},\ y_{-1},\ z_{-2}$	Position at time ΔT prior to current time.
$x_{-2},\ y_{-2},\ z_{-2}$	Position at time $2\Delta T$ prior to current time.
α	Angle of attack
Γ	Angle between the $-z_m$ axes of "intercept" maneuverplane and the vector for own aircraft to opponent's extrapolated position.
ΔT	Time increment for extrapolation.
$\Delta\rho$	Maneuverplane rotation angle increment.
η	Elevation of line of sight vector in observer's aircraft body axis system.
δ_s	Permissible speed brake deflection angle.
$\delta_{MF_{EFF}}$	Actual deflection angle of maneuvering flaps/slats.
$\delta_{MF_{MAX}}$	Maximum possible deflection angle of maneuvering flaps/slats.
λ	Line of sight angle.
ξ	Azimuth of line of sight vector in observer's aircraft body axis system.
ρ	Maneuverplane rotation angle. Air density.
ρ_s	Rotation angle of maneuverplane "closest" to opponent.
τ	Filter time constant.
ϕ^*	Euler roll angle between maneuverplane coordinate system and wind axes system.
$\psi,\ \theta,\ \phi$	Euler angles of body axes system (yaw, pitch, roll).

$\overline{\psi}, \overline{\theta}, \overline{\phi}$	Euler angles of maneuverplane coordinate system (yaw, pitch, roll).
$\widetilde{\psi}, \widetilde{\theta}, \widetilde{\phi}$	Euler angles of wind axes system (yaw, pitch, roll).
ψ', θ', ϕ'	Euler angles of body axes system in the "middle" of an integration step.
$\psi_{old}, \theta_{old}, \phi_{old}$	Euler angles of body axes system at the end of the previous integration cycle.
ψ_p, θ_p, ϕ_p	Euler angles at end of predicted turn.
Ω	Total angular change of velocity vector during predicted turn.
ω	Angular velocity during predicted turn.

COORDINATE SYSTEM DEFINITIONS

Name	Axes	Euler Angles w/Respect to Inertial	Direction Cosine Matrix(DCM)	Element of DCM	Remarks
Inertial	$x_e \ y_e \ z_e$	N/A	N/A	N/A	Pos. x North Pos. y East Pos. z Down
"Unrotated" Maneuverplane	$x_m' \ y_m' \ z_m'$	$\bar{\psi}, \ \bar{\theta}, \ 0$	Not Used	Not Used	x_m' aligned with the velocity vector, y_m' parallel to x_e-y_e plane.
Maneuverplane	$x_m \ y_m \ z_m$	$\bar{\psi}, \ \bar{\theta}, \ \bar{\phi}$ [1]	$\bar{D}$	$\bar{d}_{ij}$	x_m aligned with the velocity vector, negative z_m direction on the concave side of the flight path, y_m normal to the maneuverplane.
Wind Axes	$x_w \ y_w \ z_w$	$\hat{\psi}, \ \hat{\theta}, \ \hat{\phi}$	$\hat{D}$	$\hat{d}_{ij}$	
Body Axes	$x_b \ y_b \ z_b$	$\psi, \ \theta, \ \phi$	D [2]	d_{ij}	

1 At the time of a maneuverplane definition by a new command, $\bar{\phi} = \rho$. However, $\bar{\phi}$ varies in general during flight in a maneuverplane.

2 In the subroutine RELGN, C is the direction cosine matrix of the attackers body axes system, D of the target's body axes system.

DESCRIPTION OF THE INDIVIDUAL
SUBROUTINES OF THE AML PROGRAM

Program AMLOFL (Main Program)

This is the driver program for the off-line version. The aerodynamic tables are read (for details, see Section "Input Data Requirements"), it controls the flow of the calculations, the printout, the writing of the magnetic tape for subsequent plotting, and it performs the integration of the equations of motion. The flow of the main program is illustrated by a flowchart. (See Figure 1.)

Logic for aerodynamic data input.- The program accepts separate aircraft for the attacker and for the target. In case the two aircraft are the same (except for their weight), only one set of data cards is required.

Figure 1 shows in block diagram form the logic of the data read routine. The variable KEE serves as subscript, that is, KEE= 1 determines attacker data, KEE=2 determines target data. The variable LRD specifies how many sets of data cards are to be read in (one if both aircraft are the same, two if attacker and target are different aircraft). The variable LRDV is a flag used to determine LRD. The first card read after the ATMO data is an identifier. It contains two words at 4 characters each (FORMAT 2A4), the first word is a label, the second an identifier. The label may be: ATKR, TRGT or blank. ATKR specifies the following 156 cards to describe the attacker aircraft; if the label is TRGT, the following 156 data cards describe the target aircraft. If the label consists of four blank characters, the program assumes the following 156 cards to describe both the attacker and the target aircraft. In this case, the variable LRD is set to 1 (read only one set of data cards).

After a block of data cards have been read (block labeled 270 in Figure 1), a test is made whether an additional set of data is to be read. If that is the case, the program sets LRDV to 2 (to indicate that the next set of data is the last one to be read) and it sets the variable KEE to 3-KEE. That means, if the first set read was for the attacker (KEE=1), the next set must be for the target (KEE=2) and vice versa. The program now reads a new label and identifier and checks whether the label just read is TRGT (if the first set of data described the attacker) or if it is ATKR (if the first set of data described the target). If this is not the case, an error message is printed out.

If after reading a block of data, LRDV equals LRD, no more aircraft data has to be read. If now LRD equals two, two sets of aircraft data have been read and the input of aircraft data is terminated. If, however, LRD equals one, then only one set of aircraft data has been read and was stored as attacker data. In this case, the program copies all aircraft data from the attacker tables into the target tables.

Subroutine AERF4 (Aerodynamic Coefficients)

Purpose.- To obtain aerodynamic coefficients and thrust data from the corresponding tables by linear interpolation.

Input.- The input to the routine consists of three variables: XX, YY and IFLG. They are interpreted as follows: If IFLG is positive, the tables pertaining to the attacker aircraft are used; for negative IFLG, those pertaining to the target aircraft are used.

$$|IFLG| = 3:\quad XX = \text{Altitude}$$
$$YY = \text{Mach Number}$$

$$|IFLG| = 4:\quad XX = \text{Mach Number}$$
$$YY = \text{Coefficient of Lift}$$

Output.- With one exception, all the results of the AERF4 cal-
culations are returned through the named common block AEROUT. In
case the subroutine is called with |IFLAG| = 4 and the coefficient
of lift exceeds the maximum permissible coefficient of lift, the
value of YY by which AERF4 was called will be changed; that is, re-
duced to the maximum permissible lift coefficient.

Method.- When AERF4 is called with IFLAG having an absolute
value of 3, the following functions are obtained by linear interpo-
lation:

Variable	FORTRAN Name	Function Of
Maximum Permissible Load Factor	FLODMX	M, H
Sustained Load Factor	FLODSX	M, H
Dive Recovery Angle	RECANX	M, H
Maximum Permissible Mach NBR	FMAXX	H
Minimum Permissible Mach NBR	FMINX	H
Idle Thrust	TIDLEX	M, H
Military Thrust	TMILX	M, H
Afterburner Thrust	TABX	M, H

When AERF4 is called with |IFLAG| = 4, the following variables
are calculated:

Angle of Attack	ALPHAX	M, C_L
Coefficient of Drag	CDX	M, C_L

For the calculation of the coefficient of drag, a check is made
whether or not the aircraft has a throttle position of idle. If
that is the case, it is assumed that it also has its speedbrakes
deflected to their maximum permissible limit which is a function
of Mach and altitude. (This speedbrake deflection is calculated
as a fraction of the maximum possible speedbrake deflection in the
function subroutine SBDEF.) The fraction of the permissible speed-
brake deflection is named FRADEL. Next, an incremental drag coef-
ficient for a maximum speedbrake deflection is calculated. This

coefficient is tabulated as a function of Mach numbers. The effective incremental speedbrake drag coefficient, SBCDX, is finally obtained by multiplying the maximum incremental coefficient by FRADEL; SPCDX is returned in the named common AEROUT. It is also added to the basic drag coefficient.

The effect of maneuvering flap/slat deflections upon the coefficient of drag is handled in a similar manner. Slat and flap operations are synchronized and are assumed to occur automatically as a function of Mach number and altitude. The subroutine SLAPDF calculates the actual flap deflection angle as a fraction of the maximum possible deflection angle as a function of Mach number and altitude. Let $\delta_{MF_{MAX}}$ be the maximum possible flap/slat deflection angle and $\delta_{MF_{EFF}}$ the actual deflection angle. Further, let C_{D_U} be the drag coefficient associated with undeflected flaps and C_{D_D} the drag coefficient with fully deflected flaps. The effective drag coefficient is then calculated as:

$$C_D = C_{D_U} + \frac{\delta_{MF_{EFF}}}{\delta_{MF_{MAX}}} (C_{D_D} - C_{D_U})$$

The angle of attack is required by the AML program only in order to calculate the proper orientation of the aircraft's body axes. The calculation of the angle of attack is performed in exactly the same manner as the coefficient of drag.

The following table summarizes the tabular data from which the above-described variables are interpolated:

Array Name	Quantity	Function Of
TIDLE	Idle thrust of one engine	M, H
TMIL	Military thrust of one engine	M, H
TAB	Afterburner thrust of one engine	M, H
F4VG	Maximum allowable load factor	M, H
F4SG	Sustained load factor	M, H

Array Name	Quantity	Function Of
F4MMI	Minimum permissible Mach number	M, H
F4MMA	Maximum permissible Mach number	H
RECANG	Dive recovery angle	M, H
F4CLA	$C_{L\alpha}$ at $C_L \simeq 0$ (slats/flaps undeflected)	M
F4ALP	Angle of attack (slats/flaps undeflected)	C_L, M
CLMAX	Maximum permissible coefficient of lift	M
F5CLA	$C_{L\alpha}$ at $C_L \simeq 0$ with slats/flaps fully deflected	M
F5ALP	Angle of attack (slats/flaps fully deflected)	C_L, M
F4CDR	Coefficient of drag (slats/flaps undeflected)	C_L, M
SBCDR	Coefficient of drag increment speedbrake fully deflected	M
F5CDR	Coefficient of drag (slats/flaps fully deflected)	C_L, M

All the above referenced tables are input to the program with the exception of the two functions F4CLA and F5CLA. These two functions are calculated in the subroutine AMLOFL by the following formula:

$$C_{L\alpha} = \frac{2\Delta C_L}{\alpha(C_L = .3) - \alpha(C_L = .1)}$$

with $\Delta C_L = 0.2$.

This table of $C_{L\alpha}$ values is used to calculate the angle of attack for negative values of C_L which occur only during the maneuver of inverted straight flight.

It should be noted here that a major difference consists between the AML offline program and the version used on the DMS. While the offline program requires the function $\alpha = f(M, C_L)$, which permits the representation of a nonlinear lift slope curve, the DMS version obtains the angle of attack by assuming a linear relationship between C_L and α, by the relationship:

15

$$\alpha = \frac{C_L}{C_{L\alpha}}$$

with $C_{L\alpha}$ being a function of Mach and altitude. The off-line program assumes this linear relationship only for the small angles of attack associated with straight, inverted flight.

To accept with a minimum of program changes aircraft with no maneuvering slats/flaps, the entries in the table for the angle of attack for slaps fully deflected (F5ALP) have to be identical to those in the table F4ALP. The same holds true for the two tables F5CDR and F4CDR.

If the user selects to represent speedbrake effects by including the incremental speedbrake drag in the thrust table for idle thrust rather than specifying incremental speedbrake drag coefficients, the values in the table SBCDR should all be set to zero.

Maximum and minimum permissible Mach numbers are calculated in this subroutine; however, at the present time, the program does not use these two variables.

Subroutine CLOSS (Angle between two vectors)

Purpose.- To calculate the angle between two vectors.

Input.- Coordinates of two points determining one vector (x_a, y_a, z_a, x_t, y_t, z_t), three components of the second vector, x_d, y_d, z_d and the magnitude of the second vector V.

Output.- Angle λ between the two input vectors positive between zero and 180 degrees.

Method.- If $\vec{x}_1$ stands for first vector, $\vec{x}_2$ for second vector, then

$$\lambda = \text{arc cos} \frac{\vec{x}_1 \cdot \vec{x}_2}{|\vec{x}_1| \cdot |\vec{x}_2|}$$

16

Subroutine CMTRX (Direction Cosine Matrix)

Purpose.- This subroutine calculates the nine elements of the direction cosine matrix for given Euler angles.

Input.- Euler angles ψ, θ, ϕ.

Output.- The nine direction cosines.

Method.- The direction cosine matrix is calculated under the assumption that the standard sequence of rotations is performed (ref. 1, page II-7); first rotate about z-axis (yaw), then rotate about y-axis (pitch) and last rotate about x-axis (yaw).

$$z \text{ rotation } [R_z] = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$y \text{ rotation } [R_y] = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ +\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$x \text{ rotation } [R_x] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

Let $(x', y', z')^T$ be the new system after rotation in all three axes. Then

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = [R_x] \ [R_y] \ [R_z] \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

and the transformation matrix

$$[R_T] = [R_x][R_y][R_z]$$

$$[R_T] = \begin{bmatrix} \cos\theta \ \cos\psi & \cos\theta \ \sin\psi & -\sin\theta \\ \cos\psi \ \sin\theta \ \sin\phi \\ -\sin\psi \ \cos\phi & \cos\psi \ \cos\phi \\ +\sin\psi \ \sin\theta \ \sin\phi & \cos\theta \ \sin\phi \\ \cos\psi \ \sin\theta \ \cos\phi \\ +\sin\psi \ \sin\phi & \sin\psi \ \sin\theta \ \cos\phi \\ -\cos\psi \ \sin\phi & \cos\theta \ \cos\phi \end{bmatrix}$$

Note that a vector with components (a_1, a_2, a_3) in the original system has the components

$$[R_T] \cdot \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

in the rotated coordinate system.

The order of rotations from the original system to the rotated system is ψ, θ, ϕ. The rotation from the rotated system back to the original system has to be performed in the sequence ϕ, θ, ψ.

Subroutine CSRHO (Speed of Sound, Air Density)

Purpose.- To calculate the speed of sound, c_s, and air density, ρ, as a function of altitude.

Input.- Attitude.

Output.- Speed of sound (feet/sec). Air density (slugs/cubic feet).

Method.- Linear interpolation in table defining the standard atmosphere between sea level and 60,000 feet altitude in increments of 500 feet is used to obtain the two quantities.

Subroutine EQMOTT (Equations of Motion)

Purpose.- This subroutine calculates the forces (accelerations) acting on the center of gravity and the body axes rotational rates p, q and r about the three body axes, measured in the inertial system, which are required to execute the commanded maneuver.

Input.- Maneuver command consisting of maneuverplane rotation angle ρ, load factor ratio and throttle position, position and attitude of the target aircraft in inertial axis system at the conclusion of the previous integration cycle, the y coordinate of attacker in target aircraft's body axis system.

Output.- The three acceleration components of target aircraft in the inertial axis system (a_{xe}, a_{ye}, a_{ze}). The three velocity components in the body axis system (u, v, w). The three body rotational rates (p, q, r).

Method.- The subroutine first calculates aerodynamic quantities which are independent of the maneuvers being executed, i. e., idle, military and afterburner thrust; maximum available load factor; lift and drag corresponding to the commanded load factor.

Next, it is determined whether or not the maneuver command has changed since the last call to the routine. Consider first the case in which the command has changed. The required Euler bank angle ϕ is calculated for flight with the commanded load factor in the commanded maneuverplane.

If $\phi*$ is the Euler roll angle between the maneuverplane system and the wind axes system, then the three components of the lift in the maneuverplane system are:

(0, Lsin $\phi*$, - Lcos $\phi*$)

The three components of the gravity force in the maneuverplane system are:

$$(\overline{d}_{13}mg, \ \overline{d}_{23}mg, \ \overline{d}_{33}mg)$$

To keep the velocity vector in the maneuverplane, the total acceleration along the y_m axis must vanish, this determines the angle ϕ^* as:

$$L\sin\phi^* = -\overline{d}_{23}mg$$

$$\phi^* = -\arc \sin \left(\frac{\overline{d}_{23}mg}{L}\right)$$

The desired angle $\overset{\sim}{\phi}$ then is

$$\overset{\sim}{\phi} = \rho + \phi^*$$

and the approximately required change in the aircraft's bank angle:

$$\Delta\phi = \overset{\sim}{\phi} - \phi$$

By dividing $\Delta\phi$ by the maximum roll angle change per cycle, the number of cycles during which the aircraft will roll at maximum roll rate is calculated.

The direction of roll will always be so that the new bank angle is achieved in a roll of less than 180 degrees. For a bank angle change of exactly 180 degrees (for instance from straight upright to straight inverted flight) the direction of the roll is so that during the roll the cockpit will be pointed in the direction of the opponent and the opponent is visible during the roll maneuver. This concludes the discussion of the actions taken when the commanded flight maneuver has changed.

If no new maneuver command has occurred, the program distinguishes between two flight modes.

Transition Mode.- This mode is defined as the time during which, after a new maneuver command has occurred, the aircraft rolls to its new desired bank angle. During the transition mode, the magnitude of lift, angle of attack, drag and thrust are kept

equal to the values they had before the roll maneuver was initiated. The bank angle is incremented by constant increments each cycle until the roll is completed or until a new roll angle is commanded.

Maneuver Mode.- This is the normal mode of operation once the airplane starts flying in a specified maneuverplane. The calculations performed during this mode can be broken down into the seven steps explained individually below.

(1) Determination of the Euler angles of the maneuverplane coordinate system. ($\overline{\psi}$, $\overline{\theta}$, $\overline{\phi}$)

The angles $\overline{\psi}$ and $\overline{\theta}$ are fully determined by the target's velocity vector. (The subscript T is omitted in the following, since it is obvious that all variables pertain to the target aircraft.)

$$\overline{\psi} = \text{arc tan } (\dot{y}_e/\dot{x}_e)$$

$$\overline{\theta} = \text{arc tan } (-\dot{z}_e/V_h) \text{ where } V_h = (\dot{x}_e^2 + \dot{y}_e^2)^{1/2}$$

In order to obtain the roll angle $\overline{\phi}$ of the maneuverplane system, one has to bear in mind that by this rotation the maneuverplane y_m axis is being aligned with the maneuverplane normal, which is given by a vector $a_n = (a_{n1}, a_{n2}, a_{n3})^T$ in the inertial reference system. The transformation from the inertial system to the maneuverplane system is given by:

$$\begin{bmatrix} \hat{x}_m \\ \hat{y}_m \\ \hat{z}_m \end{bmatrix} = \begin{bmatrix} \cos\overline{\theta}\cos\overline{\psi} & \cos\overline{\theta}\sin\overline{\psi} & -\sin\overline{\theta} \\ \cos\overline{\psi}\sin\overline{\theta}\sin\overline{\phi} & \cos\overline{\psi}\cos\overline{\phi} & \cos\overline{\theta}\sin\overline{\phi} \\ -\sin\overline{\psi}\cos\overline{\phi} & +\sin\overline{\psi}\sin\overline{\theta}\sin\overline{\phi} & \\ \cos\overline{\psi}\sin\overline{\theta}\cos\overline{\phi} & \sin\overline{\psi}\sin\overline{\theta}\cos\overline{\phi} & \cos\overline{\theta}\cos\overline{\phi} \\ +\sin\overline{\psi}\sin\overline{\phi} & -\cos\overline{\psi}\sin\overline{\phi} & \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_e \\ \hat{y}_e \\ \hat{z}_e \end{bmatrix}$$

To align the y_m axis with the vector $\vec{a}_n$, it follows:

$$a_{n1} = \cos\overline{\psi} \, \sin\overline{\theta} \, \sin\overline{\phi} \; - \; \sin\overline{\psi} \, \cos\overline{\phi}$$

$$a_{n2} = \sin\overline{\psi} \, \sin\overline{\theta} \, \sin\overline{\phi} \; + \; \cos\overline{\psi} \, \cos\overline{\phi}$$

$$a_{n3} = \cos\overline{\theta} \, \sin\overline{\phi}$$

To determine $\overline{\phi}$ uniquely between $-180°$ and $+180°$ (the third equation alone would determine $\overline{\phi}$ only between $-90°$ and $+90°$), multiply the first equation by $-\sin\overline{\psi}$, the second equation by $\cos\overline{\psi}$ and adding the two equations yields:

$$\cos\overline{\phi} = a_{n2} \, \cos\overline{\psi} - a_{n1} \, \sin\overline{\psi}$$

This, together with the third equation above, uniquely determines $\overline{\phi}$ as

$$\overline{\phi} = \text{arc tan} \; \frac{a_{n3}}{\cos\overline{\theta} \, (a_{n2} \, \cos\overline{\psi} - a_{n1} \, \sin\overline{\psi})}$$

where the quadrant is determined by the signs of the numerator and denominator. It is worthwhile to note here that, in general, during a turning flight in a fixed maneuverplane, all three angles $\overline{\psi}$, $\overline{\theta}$ and $\overline{\phi}$ are changing, the vector $\vec{a}_n$ however remains constant.

(2) Determination of the Euler angles of the wind axes system. ($\tilde{\psi}$, $\tilde{\theta}$, $\tilde{\phi}$)

Note that the x-axis of the wind axis system is aligned with the x-axis of the maneuverplane system (which, of course, is aligned with the velocity vector); therefore, $\tilde{\psi}=\overline{\psi}$ and $\tilde{\theta}=\overline{\theta}$. The wind axes system is rolled from the maneuverplane system by an angle ϕ^*. The magnitude and direction of ϕ^* is determined by the fact that this roll is performed in order to compensate, by a component of the lift vector, the component of the gravity force acting along the y_m axis.

The gravity force component along the y_m axis follows from

$$\begin{bmatrix} G_{mx} \\ G_{my} \\ G_{mz} \end{bmatrix} = \begin{bmatrix} \overline{D} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}$$

so that

$$G_{my} = mg \, \sin\overline{\phi} \, \cos\overline{\theta}$$

If we assume a roll of ϕ^* between the maneuverplane coordinate system and the wind axis system, the lift component along the y_m axis is given by

$$\begin{bmatrix} LF_{mx} \\ LF_{my} \\ LF_{mz} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi^* & \sin\phi^* \\ 0 & -\sin\phi^* & \cos\phi^* \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -L \end{bmatrix}$$

so that

$$LF_{my} = -L \, \sin\phi^*$$

In order to obtain a lift component along the y_m axis of equal magnitude but of opposite direction than the gravity force component, it is required that

$$-L \, \sin\phi^* = mg \, \sin\overline{\phi} \, \cos\overline{\theta}$$

so that

$$\phi^* = -\text{arc sin} \, (\frac{mg}{L}) \, \sin\overline{\phi} \, \cos\overline{\theta}$$

Since the roll transformation is performed last in the sequence of the three rotations, it follows that

$$\overset{\sim}{\phi} = \overline{\phi} + \phi^*$$

(3) <u>Determination of the Euler angles of the body axes system.</u>
(ψ, θ, ϕ)

The body axes system is obtained by a rotation of the angle of attack α about the wind y_w axis. This is the final body axis system since a basic assumption is that there are no side forces or side slip. The total transformation from the inertial axes system to the body axes system is:

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} \tilde{D} \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix}$$

$$= \begin{bmatrix} \tilde{d}_{11}\cos\alpha - \tilde{d}_{31}\sin\alpha & \tilde{d}_{12}\cos\alpha - \tilde{d}_{32}\sin\alpha & \tilde{d}_{13}\cos\alpha - \tilde{d}_{33}\sin\alpha \\ \tilde{d}_{21} & \tilde{d}_{22} & \tilde{d}_{23} \\ \tilde{d}_{11}\sin\alpha + \tilde{d}_{31}\cos\alpha & \tilde{d}_{12}\sin\alpha + \tilde{d}_{32}\cos\alpha & \tilde{d}_{13}\sin\alpha + \tilde{d}_{33}\cos\alpha \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix}$$

The Euler angles of the body axes system therefore are:

$$\theta = -\arc \sin\alpha(\tilde{d}_{13} \cos\alpha - \tilde{d}_{33} \sin\alpha)$$

$$\psi = \arc \tan \frac{\tilde{d}_{12} \cos\alpha - \tilde{d}_{32} \sin\alpha}{\tilde{d}_{11} \cos - \tilde{d}_{31} \sin\alpha}$$

$$\phi = \arc \tan \frac{\tilde{d}_{33}}{\tilde{d}_{13} \sin\alpha + \tilde{d}_{33} \cos\alpha}$$

(4) <u>Calculation of the aerodynamic and propulsive forces in the wind axes system.</u> (F_{xw}, F_{yw}, F_{zw})

Assuming no side forces, the three forces in the wind axes system are:

24

$$F_{xw} = T \cos\alpha - D$$

$$F_{yw} = 0$$

$$F_{zw} = -L - T \sin\alpha$$

(5) Calculation of the linear accelerations in the inertial system. (a_{xe}, a_{ye}, a_{ze})

These accelerations include now the gravity force and are:

$$\begin{bmatrix} a_{xe} \\ a_{ye} \\ a_{ze} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} D \end{bmatrix}^{-1} \begin{bmatrix} F_{xw} \\ F_{yw} \\ F_{zw} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}$$

(6) Calculation of the required changes in the three Euler angles of the body axes system. ($\Delta\psi$, $\Delta\theta$, $\Delta\phi$)

The Euler angles calculated in step (3) indicate the required attitude of the aircraft in order to fly in the specified maneuver-plane. The differences between the aircraft's Euler angles at the end of the previous integration step and those above calculated angles are the desired changes.

$$\Delta\psi = \psi - \psi_{old}$$

$$\Delta\theta = \theta - \theta_{old}$$

$$\Delta\phi = \phi - \phi_{old}$$

where the subscript old indicates the values at the end of the previous integration cycle.

(7) Calculation of body rotational rates. (p, q, r)

Approximate Euler angle rates are obtained by dividing the above differences by the integration stepsize dt:

$$\dot{\psi} = \Delta\psi/dt \qquad \dot{\theta} = \Delta\theta/dt \qquad \dot{\phi} = \Delta\phi/dt$$

using the conventional relationships between body rotational rates

25

and Euler angles and Euler angle rates (see reference 1, page II-10) the required rates are:

$$p_{req} = \dot{\phi} - \dot{\psi} \sin\theta'$$

$$q_{req} = \dot{\theta} \cos\phi' + \dot{\psi} \sin\phi' \cos\theta'$$

$$r_{req} = \dot{\psi} \cos\phi' \cos\theta' - \dot{\theta} \sin\phi' \qquad \text{where}$$

(It would be inaccurate to substitute either the old Euler angles or the new desired ones in the above three equations; the best approximation to use is)

$$\psi' = \psi_{old} + 1/2 (\psi - \psi_{old})$$

$$\theta' = \theta_{old} + 1/2 (\theta - \theta_{old})$$

$$\phi' = \phi_{old} + 1/2 (\phi - \phi_{old}).$$

To smooth the motion of the displayed aircraft in the DMS, the above calculated rotational rates (p_{req}, q_{req} and r_{req}) are passed through a digital filter before being transferred to the DMS program (see write-up of Subroutine FILTRT).

(8) <u>Calculation of the velocity components in the body axis system.</u> (u, v, w)

These three velocities are obtained by multiplying the present direction cosine matrix of the body axes system with the inertial velocity vector.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} D \end{bmatrix} \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{z}_e \end{bmatrix}$$

Subroutine EXTRT (Extrapolation)

Purpose.- To extrapolate the attacker's position, velocity and attitude.

Input.- The attacker's present position, attacker's position Δt and $2\Delta t$ seconds ago.

Output.- The attacker's extrapolated position, velocity and attitude at the end of the prediction time.

Method.- Define the attacker's positions at the three last observed points as follows:

At time t: $\quad\quad x_0, \; y_0, \; z_0.$

At time t - Δt: $\quad x_{-1}, \; y_{-1}, \; z_{-1}.$

At time t - 2Δt: $\quad x_{-2}, \; y_{-2}, \; z_{-2}.$

Define two vectors u and v, u connecting the points at time t - 2Δt and t, v connecting the points at time t - ΔT and t.

If the cross product between u and v vanishes, the three points lie in a straight line, in which case the extrapolation of the attacker's attitude requires some special attention.

Assuming the equations of the flight path to be of the form:

$$x(t) = a_x t^2 + b_x t + c_x$$
$$y(t) = a_y t^2 + b_y t + c_y$$
$$z(t) = a_z t^2 + b_z t + c_z.$$

The unknown coefficients a_x, b_x and so forth can be obtained by postulating that at time t_0, the flight path must go through (x_0, y_0, z_0); at time $t_0 - \Delta T$, it must pass through (x_{-1}, y_{-1}, z_{-1}) and at time $t_0 - 2\Delta T$ through (x_{-2}, y_{-2}, z_{-2}). Without loss of generality, t_0 may be set to zero, so that the equations for the x-coordinate will be

27

$$c_x = x_o$$

$$a_x (\Delta t)^2 - b_x (\Delta T) = x_{-1} - x_o$$

$$4a_x (\Delta T)^2 - 2b_x (\Delta T) = x_{-2} - x_o$$

from which follows:

$$a_x = \frac{1}{2(\Delta T)^2} (x_o - 2 x_{-1} + x_{-2})$$

$$b_x = \frac{1}{2(\Delta T)} (3x_o - 4x_{-1} + x_{-2})$$

$$c_x = x_o$$

and equivalent expressions for the remaining coefficients result. It can be easily proven that the points of the flight path extrapolated by the above formula lie in the plane defined by the three input points.

The extrapolated velocity at time $t_o + \Delta T$ is:

$$V = \left((2 a_x\Delta T + b_x)^2 + (2 a_y\Delta T + b_y)^2 + (2 a_z \Delta T + b_z)^2\right)^{1/2}$$

If the three input points were not colinear, the extrapolated attitude is assumed such that the aircraft x-body axis is aligned with the extraplated velocity vector and that the wing axis is perpendicular to the plane of the extrapolated flight path, such that the cockpit points toward the concave side of the flight path.

Subroutine FILTRT (Body Rotational Rate Filter)

Purpose.- To smooth the motion of the displayed aircraft on the DMS by first filtering the body rotational rates.

Input.- Requested rates for p or q or r. (p_{req}, q_{req}, r_{req})

28

Output.- Actual rates for p or q or r. $(p_{act}, q_{act}, r_{act})$

Method.- The circuit as depicted in Figure 2 is simulated. The equation, for a first order approximation to the actual rates are (given for roll rate p, corresponding equations hold for q and r):

$$p_{act}(t + dt) = p_{act}(t) + \frac{dt}{\tau}\left(p_{in}(t) - p_{act}(t)\right)$$

where p_{in} is equal to p_{req} unless p_{req} is greater than p_{max}, in which case p_{in} is equal to p_{max}, or if p_{req} is less than $-p_{max}$, in which case p_{in} is equal to $-p_{max}$.

Subroutine FIRCON (Fire Cone Determination)

Purpose.- To determine whether the subject aircraft is within range and angles to fire one of three kinds of weapons at the opponent.

Input.- Flag indicating subject aircraft (KAORT = 1 for attacker; KAORT = 2 for target) range, deviation angle, angle off.

Output.- Flag indicating whether or not opponent is within any one of the weapons delivery envelopes

    FIRYES = 0 if opponent outside weapons envelope
    FIRYES = 1 if opponent within weapons envelope

Method.- Upper and lower limits of range and upper limits of deviation angle and angle off are specified in data statements for three different weapons separately for the attacker and the target aircraft. The flag FIRYES is set to one if any one of the three weapons envelopes is satisfied.

Subroutine GETRXN (Get Closest Maneuverplane)

Purpose.- To find the number of maneuverplane rotation incre-
ments which will rotate a maneuverplane closest to some desired
rotation angle.

Input.- Desired maneuverplane rotation angle (ρ_{des}), maneuver-
plane rotation angle increment ($\Delta\rho$).

Output.- Rotation angle of discrete maneuverplane closest to
the desired maneuverplane (ρ). The number of rotation increments
from the unrotated maneuverplane to the discrete maneuverplane (N).

Method.-

$$N = \left[\frac{|\rho_{des}|}{\Delta\rho} \right] \qquad \text{where [ ] indicates "integer portion of"}$$

$$\rho_0 = N \cdot \Delta\rho$$

If $\rho_0 + \frac{\Delta\rho}{2} \leq |\rho_{des}|$ Then $\rho_0 = \rho_0 + \Delta\rho$

$$N = N + 1$$

Finally, a test is made whether ρ_{des} was negative.

For positive ρ_{des}: $\rho = \rho_0$; $N = N$

For negative ρ_{des}: $\rho = -\rho_0$; $N = -N$

The following FORTRAN variables represent the above variables:
ROTS = ρ_{des}; ROTN = $\Delta\rho$; ROTX = ρ; ICOMD = N

Subroutine HUBLO (Human Pilot Blackout)

Purpose.- To reduce the loadfactor if the blackout function
exceeds a certain limit when the option of having the aircraft
flown subject to pilot blackout has been selected.

Input.- A flag to indicate initialization (NDX), a flag indicating which of the two aircraft is analyzed (LABLAT), the integration step size for the integration of the blackout function (TINC), the presently flown loadfactor (GLEVEL).

Output.- Loadfactor (GLEVEL). If the blackout function exceeds a specified limit, a reduced loadfactor is returned; otherwise, the input value of the loadfactor is returned.

Method.- The blackout model assumes that a pilot will tend to grayout if exposed to greater than 5 g's and will tend to recover if operating below 5 g's and that a direct relationship between the logarithm of the loadfactor and the logarithm of the time to blackout is assumed. The blackout function, K_{BO}, is defined as:

$$K_{BO} = \int_0^t \dot{K}_{BO} \, dt \qquad 0 \le K_{BO}$$

with $\dot{K}_{BO}$ being defined as:

$$\dot{K}_{BO} = (1/3)(0.2n_z)^{5.7234} \qquad \text{if } n_z \ge 5$$
$$= -(100/n_z) \qquad \text{if } 5 < n_z < 1$$
$$= -100 \qquad \text{if } n_z \le 1$$

Under this condition, the blackout function increased from zero to 100 in 300 seconds for a 5 g load; for a 9 g load, this time is 10 seconds.

The subroutine reduces the loadfactor to 4.99 g if the blackout function exceeds a value of 80.

Subroutine INRD (Input Data Reading)

This subroutine reads the input data for an individual run, prints the same data out, initializes quaternions, and prints the definitions of the questions defining the state.

Detailed information about the input data read by this routine may be found in the section "INPUT DATA DESCRIPTION".

Subroutine MEINTL (Accumulation of Weapons Time)

Purpose.- To accumulate offensive time and weapons delivery time for both aircraft.

Input.- Range, relative geometry for both aircraft.

Output.- Accumulated offensive time and weapons delivery time.

Method.- Offensive time for a given aircraft is defined as time during which the absolute values of the deviation angle and the angle off are both less than 60°. Weapons delivery conditions are obtained by calls to subroutine FIRCON.

Subroutine NORPLN (Normal to Maneuverplane)

Purpose.- To calculate the components of a unit vector normal to a specified maneuverplane.

Input.- The three components of the velocity vector through which the maneuverplane has to pass and the rotation angle ρ of the maneuverplane.

Output.- The three components a_{n1}, a_{n2}, and a_{n3} of a unit vector normal to the maneuverplane in the inertial coordinate system.

Method.- A straightforward way to calculate this normal would be to compute the direction cosines $\bar{d}_{21}$, $\bar{d}_{22}$, and $\bar{d}_{23}$ of the

maneuverplane coordinate system which correspond, of course, to the three components a_{n1}, a_{n2}, and a_{n3}, which are:

$$a_{n1} = \cos\overline{\psi} \ \sin\overline{\theta} \ \sin\rho \ - \ \sin\overline{\psi} \ \cos\rho$$

$$a_{n2} = \cos\overline{\psi} \ \cos\rho + \sin\overline{\psi} \ \sin\overline{\theta} \ \sin\rho$$

$$a_{n3} = \cos\overline{\theta} \ \sin\rho$$

A computationally more efficient way is used in the program. Let x, y and z be components of a unit vector in the inertial system along the velocity vector. Write V_h for $\sqrt{x^2 + y^2}$ and remember that $\sqrt{x^2 + y^2 + z^2} = 1$.

Then

$$\sin\overline{\psi} = y/V_h \qquad\qquad \cos\overline{\psi} = x/V_h$$

$$\sin\overline{\theta} = -z \qquad\qquad \cos\overline{\theta} = \sqrt{x^2 + y^2} = V_h$$

Therefore:

$$a_{n1} = (x/V_h) \ (-z) \ (\sin\rho \ ) \ -(y/V_h) \ (\cos\rho)$$

$$a_{n2} = (x/V_h) \ (\cos\rho) + y/V_h \ (-z) \ (\sin\rho)$$

$$a_{n3} = V_h \ (\sin\rho)$$

Substituting F_k for V_h and F_{kk} for V_h, the three components, in the order as calculated in the program are:

$$a_{n3} = F_{kk}$$

$$a_{n2} = (F_k \ x - a_{n3} \ z \ y)/ \ (x^2 + y^2)$$

$$a_{n1} = (-a_{n3} \ x \ z - F_k \ y) \ /(x^2 + y^2)$$

33

Subroutine PCSRO
(Print Speed of Sound and Air Density)

Purpose.- To print speed of sound and air density tables. It is called at the beginning of a run if the option to print aerodynamic data is selected.

Subroutine PRCELL (Print Tactical Situation)

Purpose.- To print the tactical situation for both aircraft. Interaircraft data, such as range, range rate, line of sight angle, are printed. The subroutine also prints accumulated offensive time and accumulated weapons time. Further, the cell-state for the two aircraft is listed.

Subroutine PRESR (Predict Straight Flight)

Purpose.- To predict position, velocity, and attitude for trial maneuvers with straight flight.

Input.- Prediction time (TIMPR), flag for upright or inverted attitude (ISOM), present velocity (V_2), present position (XX, YY, ZZ), present yaw and pitch angle of maneuverplane coordinate system in form of $\sin\overline{\theta}$, $\cos\overline{\theta}$, $\sin\overline{\Psi}$, $\cos\overline{\Psi}$.

Output.- The predicted direction cosine matrix (CNEW), the predicted velocity vector (VXNEW, VYNEW, YZNEW), the predicted position (XXNEW, YYNEW, ZZNEW).

Method.- It is assumed that straight flight will be in the direction of the present velocity vector with $\phi=0°$ for upright flight $\phi=180°$ for inverted flight.

34

Subroutine PRETNW (Predict Turning Flight)

Purpose.- To predict position, velocity and attitude for turning flight in a specified maneuverplane.

Input.- Unit vector normal to maneuverplane along the positive y_m axis; direction cosine matrix of maneuverplane coordinate system ($\overline{C}_{TR}$); normal acceleration (a_{ng}); prediction time (T_p); present velocity vector ($\dot{x}_e$, $\dot{y}_e$, $\dot{z}_e$); present position (x_e, y_e, z_e); magnitude of present velocity vector (VELT).

Output.- Predicted position (XXNEW, YYNEW, ZZNEW): predicted velocity vector (VXNEW, VYNEW, YZNEW); predicted direction cosine matrix of body axis system (CNEW).

Method: It is assumed that the flight path is a circular arc in the maneuverplane with the concave side towards the $-z_m$ axis and with the present velocity vector being tangential to the flight path. The magnitude of the velocity vector is assumed to be constant along the predicted flight path. Figure 3 illustrates the flight path prediction in the maneuverplane system. For a flight path with constant speed and constant normal acceleration, the following relations hold:

$$R = \frac{V^2}{a_n}$$

$$\omega = \frac{V}{R}$$

$$\Omega = \omega T_p$$

R = turn radius

V = speed

a_n = normal acceleration

ω = angular velocity

Ω = total angular change of velocity vector during prediction interval

The following equations describe the flight path in the maneuverplane coordinate system during the prediction interval:

35

$$x_m(t) = R \sin \omega t$$

$$y_m(t) = 0$$

$$z_m(t) = -R (1 - \cos \omega t)$$

The coordinate changes between the present position and the position at the end of the prediction interval in the inertial coordinate system are:

$$
\begin{bmatrix} \Delta x_e \\ \Delta y_e \\ \Delta z_e \end{bmatrix} = \begin{bmatrix} \bar{D}^T \end{bmatrix} \begin{bmatrix} R\sin\Omega \\ 0 \\ -R(1-\cos\Omega) \end{bmatrix}
$$

The velocity vector at the end of the prediction interval has the following components in the inertial coordinate system:

$$
\begin{bmatrix} VXNEW \\ VYNEW \\ VZNEW \end{bmatrix} = \begin{bmatrix} \bar{D}^T \end{bmatrix} \begin{bmatrix} V\cos\Omega \\ 0 \\ -V\sin\Omega \end{bmatrix}
$$

The Euler angles at the end of the prediction interval are:

$$\psi_p = \text{atan } (VYNEW/VXNEW)$$

$$\theta_p = \text{atan } (-VZNEW/(VXNEW^2 + VYNEW^2)^{1/2})$$

$$\phi_p = \text{atan } \frac{a_{n3}}{\cos\theta_p} / (a_{n2} \cos\psi_p - a_{n1} \sin\psi_p)$$

These three Euler angles assume the aircraft's longitudinal axis to be aligned with the predicted velocity vector and the aircraft's wings to be perpendicular to the maneuverplane used for the flight path extrapolation.

Subroutine Oiler (Euler Angles)

Purpose.- To obtain the Euler angles from a given direction cosine matrix.

Input.- Direction cosine matrix (c_{ij}).

Output.- The three Euler angles.

Method.- $\theta = -\text{arc sin } c_{13}$

$\psi = \text{arc tan } (c_{12}/c_{11})$

$\phi = \text{arc tan } (c_{23}/c_{33})$

Subroutine PRNTAC (Print Aircraft Data)

Purpose.- This subroutine prints positions, velocities, some aerodynamic and propulsive data for both aircraft at intervals controlled by an input parameter.

Subroutine PRTF4 (Print Aerodynamic Data)

This subroutine prints the aerodynamic data defining the two aircraft. This printing is optional and is controlled by an input parameter. If the input data defines both aircraft as being the same, only one set of aerodynamic data is printed. In addition to the wing area, the following tables are printed:

- maximum load factor (f (M,H))
- sustained load factor (f (M,H))
- idle thrust (f (M,H))
- military thrust (f (M,H))
- afterburner thrust (f (M,H))
- angle of attack with slats/flaps retracted (f (M,C_L))
- C_L (f (M)) for negative values of C_L
- angle of attack with slats/flaps extended (f (M,C_L))

37

- coefficient of drag with slats/flaps retracted ($f(M,C_L)$)
- coefficient of drag with slats/flaps extended ($f(M,C_L)$)
- $C_{L\ max}$ as function of Mach number
- minimum and maximum Mach number as function of altitude
- dive recovery angle ($f(M,H)$)
- incremental coefficient of drag for speed brakes fully extended ($f(M)$)

Subroutine QUAT (Quaternion Integration)

Purpose.- To perform one integration step for quaternions.

Input.- Present values of quaternions, body axes rotational rates p, q and r and integration step size.

Output.- Values of the quaternions at the end of the integration step.

Method.- Background information about the use of quaternions in the simulation of rotating bodies may be found in reference 2. The same set of quaternions as defined in this reference is used in the DMS and AML program to represent the body axis rotations. Letting

$$a(t) = (a_1(t),\ a_2(t),\ a_3(t),\ a_4(t))^T$$

and

$$\dot{a}(t) = (\dot{a}_1(t),\ \dot{a}_2(t),\ \dot{a}_3(t),\ \dot{a}_4(t))^T$$

and considering the body rotational rates as constants P, Q, and R the differential equations governing the quaternions can be written as:

$$\dot{a}(t) = Aa(t) \qquad \text{with}$$

$$A = \begin{bmatrix} 0 & -R/2 & -Q/2 & -P/2 \\ R/2 & 0 & -P/2 & Q/2 \\ Q/2 & P/2 & 0 & -R/2 \\ P/2 & -Q/2 & R/2 & 0 \end{bmatrix}$$

The solution to these equations is given by

$$a(t) = e^{At} C \text{ with } C = (a_1(o), a_2(o), a_3(o), a_r(o))^T$$

and

$$e^{At} = I + At + \frac{A^2 t^2}{2!} + \frac{A^3 t^3}{3!} + \dots$$

where I is the identify matrix.

Because of the skew-symmetric form of the matrix A, it can be shown that this solution can be written in the form:

$$a_1(t) = C_1 \cos\left(\frac{at}{2}\right) - \left(\frac{RC_2 + QC_3 + PC_4}{a}\right) \sin\left(\frac{at}{2}\right)$$

$$a_2(t) = C_2 \cos\left(\frac{at}{2}\right) + \left(\frac{RC_1 - PC_3 + QC_4}{a}\right) \sin\left(\frac{at}{2}\right)$$

$$a_3(t) = C_3 \cos\left(\frac{at}{2}\right) + \left(\frac{QC_1 + PC_2 - RC_4}{a}\right) \sin\left(\frac{at}{2}\right)$$

$$a_4(t) = C_4 \cos\left(\frac{at}{2}\right) - \left(\frac{PC_1 - QC_2 + RC_3}{a}\right) \sin\left(\frac{at}{2}\right)$$

where $a = \sqrt{P^2 + Q^2 + R^2}$

If one replaces the trigonometric functions in the above formula by their first order approximation, and replaces t by dt, the integration stepsize, the following formulae result:

$$a_1(t+dt) = a_1(t) - 0.5 \cdot (Ra_2(t) + Qa_3(t) + Pa_4(t)) \, dt$$

$$a_2(t+dt) = a_2(t) + 0.5 \cdot (Ra_1(t) - Pa_3(t) + Qa_4(t)) \, dt$$

$$a_3(t+dt) = a_3(t) + 0.5 \cdot (Qa_1(t) + Pa_2(t) - Ra_4(t)) \, dt$$

$$a_4(t+dt) = a_4(t) + 0.5 \cdot (Pa_1(t) - Qa_2(t) + Ra_3(t)) \, dt$$

These four relationships are obviously identical to the integration of the quaternion derivatives by Euler's method.

The DMS program integrates the quaternions using the form with the trigonometric functions while the off-line program uses the ones which correspond to Euler's method.

After the integration is performed, the quaternions are nor-normalized so that $\sqrt{a_1{}^2 + a_2{}^2 + a_3{}^2 + a_4{}^2} = 1$.

Subroutine Quatex (Quaternions to Direction Cosines)

Purpose.- To transform the quaternions into the nine direction cosines.

Input.- Quaternions b_1, b_2, b_3, b_4

Output.- Direction cosines d_{ij} $i=1...3$ $j=1...3$

Method.- The following relations between quaternions and direction cosines exist:

$$d_{11} = b_1{}^2 - b_2{}^2 - b_3{}^2 + b_4{}^2$$

$$d_{21} = 2 (b_3 b_4 - b_1 b_2)$$

$$d_{31} = 2 (b_1 b_3 + b_2 b_4)$$

$$d_{12} = 2 (b_3 b_4 + b_1 b_2)$$

$$d_{22} = b_1{}^2 - b_2{}^2 + b_3{}^2 - b_4{}^2$$

$$d_{32} = 2 (b_2 b_3 - b_1 b_4)$$

$$d_{13} = 2 (b_2 b_4 - b_1 b_3)$$

$$d_{23} = 2 (b_2 b_3 + b_1 b_4)$$

$$d_{33} = b_1{}^2 + b_2{}^2 - b_3{}^2 - b_4{}^2$$

Subroutine QUATIN (Euler Angles to Quaternions)

Purpose.- To calculate the Euler angles corresponding to given quaternions.

Input.- Euler angles ψ, θ, ϕ

Output.- Quaternions b_1, b_2, b_3 and b_4.

Method.- The following relationships are programmed:

$$b_1 = \cos\frac{\psi}{2} \cos\frac{\theta}{2} \cos\frac{\phi}{2} + \sin\frac{\psi}{2} \sin\frac{\theta}{2} \sin\frac{\phi}{2}$$

$$b_2 = \cos\frac{\psi}{2} \sin\frac{\theta}{2} \sin\frac{\phi}{2} + \sin\frac{\psi}{2} \cos\frac{\theta}{2} \cos\frac{\phi}{2}$$

$$b_3 = \cos\frac{\psi}{2} \sin\frac{\theta}{2} \cos\frac{\phi}{2} + \sin\frac{\psi}{2} \cos\frac{\theta}{2} \sin\frac{\phi}{2}$$

$$b_4 = \cos\frac{\psi}{2} \cos\frac{\theta}{2} \sin\frac{\phi}{2} - \sin\frac{\psi}{2} \sin\frac{\theta}{2} \cos\frac{\phi}{2}$$

Function RADIFY (Random Perturbation)

Purpose.- To introduce randomness in the decision process by randomly perturbing selected variables about their nominal value.

Input.- Code word, nominal value, upper limit for randomized value, lower limit for randomized value.

Output.- The output depends on whether the code word by which this function was called matches one of the names specified in this function. If the code word matches, output is a random variable uniformly distributed between the lower and the upper limit by which function was called. If the code word does not match one of the names in the function, the nominal value is returned.

Method.- The subroutine checks first whether the code word in the argument list matches one of the 12 names acceptable for random change. Table I lists the code words causing randomization and also lists the variables which will be randomized. There are 20 calls to this function in the program, 10 for the target, 10 for the attacker. REACTT calls RADIFY 8 times, TRYNXT once. The variables randomized and the code word in the argument list are summarized in Table

If the user wants to eliminate certain variables from being randomized, he has to delete the corresponding names in the function RADIFY. If the user wants no randomization at all, function RADIFY has to be modified so that it always returns the nominal value.

Subroutine REACTT (Reaction Routine)

Purpose.- To determine the most promising tactical maneuver and to set up parameters required by the equation of motion routine.(EQMOTT) to execute it.

Input.- Target's geometry, attacker's position.

Output.- Maneuver command consisting of maneuverplane rotation angle, load factor, and throttle setting.

Method.- Subroutine REACTT performs the following major functions:

- Saving the attackers position for extrapolation

-       Extrapolating the attacker's position and attitude by calling subroutine EXTRT

-       Defining the trial maneuvers by calling subroutine TRYNXT

-       Calculating thrust, drag and specific energy rate for the trial maneuvers

-       Predicting the situation resulting from each trial maneuver by calling PRESR or PRETNEW and RELGN and STATET

-       Selecting the trial maneuver with the highest expected value.

-       (Optionally) printing information about trial maneuvers and the selected maneuver.

The requirement for processor time by the AML program is not constant for all integration cycles. Between tactical decisions, only the equations of motion routine is exercised which requires very little computational effort. When, however, the AML program reaches a tactical decision point, a substantial amount of computation has to be performed in setting up and in evaluating the expected outcomes of the trial maneuvers. To alleviate the problem of excessive computational activity during the decision-making process, a scheme was devised to break the decision process up and to spread it throughout several integration cycles. During the first cycle, the extrapolation of the target's position and attitude is performed. In the second cycle, the trial maneuvers are set up and their expected outcome is calculated. During the third cycle, the most promising trial maneuver is selected and the command parameters associated with this maneuver are set up.

Breaking the decision process up in this way proved sufficient to exercise the AML program in real time with the central processor being a CDC 6600 computer. For computers with a slower cycle time, the second cycle could be broken down into two portions; setting up the trial maneuvers in one cycle and evaluating them in a second cycle.

It is, of course, desirable to perform the decision process in as few integration cycles as possible; each additional cycle delays the execution of the selected best maneuver. The present delay of about three cycles, with an integration stepsize of approximately 40 milliseconds, is certainly of the order of magnitude as the decision time of a human pilot.

Subroutine RELGN (Relative Geometry)

Purpose.- To compute the variables defining the relative geometry between two aircraft.

Input.- The following variables are transferred through the argument list:

Position of aircraft 1 in inertial frame: x_1, y_1, z_1

Velocity of aircraft 1 in inertial frame: $\dot{x}_1$, $\dot{y}_1$, $\dot{z}_1$

Position of aircraft 2 in inertial frame: x_2, y_2, z_2

Velocity of aircraft 2 in inertial frame: $\dot{x}_2$, $\dot{y}_2$, $\dot{z}_2$

Line of sight angle from aircraft 1
to aircraft 2 at time $t - \Delta T$ $\lambda_{1,-1}$

Line of sight angle from aircraft 2
to aircraft 1 at time $t - \Delta T$ $\lambda_{2,-1}$

Time difference for line of sight angle
observations ΔT

Magnitude of velocity vector for
aircraft 1 V_1

Magnitude of velocity vector for
aircraft 2 V_2

The following variables are transferred through common

44

Direction cosine matrix from inertial

frame to aircraft 1   CRGR

Direction cosine matrix from inertial

frame to aircraft 2   DRGR

<u>Output</u>.- All output variables are transferred through common

Range	R
Range Rate	$\dot{R}$
Coordinates of aircraft 2 in the body frame of aircraft 1	x_{1-2} y_{1-2} z_{1-2}
Coordinates of aircraft 1 in the body frame of aircraft 2	x_{2-1} y_{2-1} z_{2-1}
Elevation of the vector from aircraft 1 to aircraft 2 in the body frame of aircraft 1	η_{1-2}
Azimuth of the vector from aircraft 1 to aircraft 2 in the body frame of aircraft 1	ξ_{1-2}
Elevation of the vector from aircraft 2 to aircraft 1 in the body frame of aircraft 2	η_{2-1}
Azimuth of the vector from aircraft 2 to aircraft 1 in the body frame of aircraft 2	ξ_{2-1}
Angle between the velocity vector of aircraft 1 and the vector from aircraft 1 to aircraft 2	λ_1

Angle between the velocity vector of aircraft 1 and the vector from aircraft 1 to aircraft 2 λ_2

Rate of change of the angle λ_1 $\dot{\lambda}_1$

Rate of change of the angle λ_2 $\dot{\lambda}_2$

Method.- The following table exhibits the FORTRAN variable names used for the input and output variables of subroutine RELGN.

Variable	FORTRAN Name	Variable	FORTRAN Name
x_1, y_1, z_1	XE1, YE1, ZE1	CRGR	CRGR
$\dot{x}_1, \dot{y}_1, \dot{z}_1$	XEDOT1, YEDOT1, ZEDOT1	DRGR	DRGR
x_2, y_2, z_2	XE2, YE2, ZE2	R	RANGE
x_2, y_2, z_2	XEDOT2, YEDOT2, ZEDOT2	$\dot{R}$	RRATE
$\lambda_{1,-1}$	FLOSM1	x_{1-2}	XTINA
$\lambda_{2,-1}$	FLOSM2	y_{1-2}	YTINA
ΔT	TM	z_{1-2}	ZTINA
V_1	VVEL1	x_{2-1}	XAINT
V_2	VVEL2	y_{2-1}	YAINT
		z_{2-1}	ZAINT
η_{1-2}	ANGETA	λ_1	FLOSSA
ξ_{1-2}	ANXIA	λ_2	FLOSST
η_{2-1}	ANGETT	$\dot{\lambda}_1$	DLOSSA
ξ_{2-1}	ANGXIT	$\dot{\lambda}_2$	DLOSST

The output variables are calculated as follows:

$$R = ((x_2-x_1)^2 + (y_2-y_1)^2 + (z_2-z_1)^2)^{1/2}$$

$$\dot{R} = ((x_2-x_1)(\dot{x}_2-\dot{x}_1) + (y_2-y_1)(\dot{y}_2-\dot{y}_1) + (z_2-z_1)(\dot{z}_1-\dot{z}_1))/R$$

$$\begin{bmatrix} x_{1-2} \\ y_{1-2} \\ z_{1-2} \end{bmatrix} = \begin{bmatrix} CRGR \end{bmatrix} \begin{bmatrix} x_2-x_1 \\ y_2-y_1 \\ z_2-z_1 \end{bmatrix}$$

$$\begin{bmatrix} x_{2-1} \\ y_{2-1} \\ z_{2-1} \end{bmatrix} = - \begin{bmatrix} DRGR \end{bmatrix} \begin{bmatrix} x_2-x_1 \\ y_2-y_1 \\ z_2-z_1 \end{bmatrix}$$

$$\eta_1 = \arcsin \frac{-z_{1-2}}{R} \qquad \eta_2 = \arcsin \frac{-z_{2-1}}{R}$$

$$\xi_1 = \arctan \frac{y_{1-2}}{x_{1-2}} \qquad \xi_2 = \arctan \frac{y_{2-1}}{x_{2-1}}$$

$$\dot{\lambda}_1 = \frac{\lambda_1 - \lambda_{1,-1}}{\Delta T} \qquad \dot{\lambda}_2 = \frac{\lambda_2 - \lambda_{2,-1}}{\Delta T}$$

Function SBDEF (Maximum Speed Brake Deflection)

Purpose.- To calculate maximum permissible speed brake deflections as function of Mach number and altitude.

Input.- Altitude, Mach number, flag for attacker or target aircraft.

Output.- Ratio of permissible speed brake deflection to maximum possible speed brake deflection.

Method.- The curves defining maximum allowable speed brake deflections for a variable geometry aircraft were approximated by

47

straight line segments. (See Figure 4.) The actual permissible speed brake deflection is then obtained by linear interpolation as follows:

For a given altitude, there is a Mach number M_0 above which no speed brake deflection is allowed.

$$M_0 = 1.95 + 4.00 \cdot 10^{-5} \ (H - 30000) \text{ for } H > 30000$$

$$M_0 = 1.95 + 1.70 \cdot 10^{-5} \ (H - 30000) \text{ for } H < 30000$$

For any given altitude, there is a Mach number M_m at and below which maximum speed brake deflection is permissible:

$$M_m = 1.18 + 4.35 \cdot 10^{-5} \ (H - 30000) \text{ for } H > 30000$$

$$M_m = 1.18 + 1.60 \cdot 10^{-5} \ (H - 30000) \text{ for } H < 30000$$

Finally, call the Mach number at which the slope of the deflection angle changes, M_c.

$$M_c = 1.27 + 4.25 \cdot 10^{-5} \ (H - 30000) \text{ for } H > 30000$$

$$M_c = 1.27 + 1.30 \cdot 10^{-5} \ (H - 30000) \text{ for } H < 30000$$

The computational procedure of this subroutine is illustrated in Figure 5. This subroutine must be reprogrammed for different aircraft.

Function SLAPDF (Slat/Flap Deflection Limits)

Purpose.- To calculate flap/slat deflections as a function of Mach number and altitude.

Input.- Altitude, Mach number, aircraft flag (attacker or target).

Output.- The ratio of actual flap/slat deflection to full deflection.

Method.- A linear interpolation between the zero degree limit and the full deflection limit in the Mach altitude plane is used (see Figure 6). This function requires reprogramming for different aircraft types.

Subroutine STATET (State Evaluation)

Purpose.- To determine the cell in the situation matrix for a given relative situation between target and attacker aircraft.

Input.- Relative geometry variables; specific energy rate for target, specific energy for target and for attacker, weight factors.

Output.- State vector (zeros or ones for the currently used 15 questions), cell number, cell value.

Method.- Fifteen questions, as explained in the section "Value Assignment to Individual Cells" in Volume I of this report, are evaluated. If the condition is true, a one is put into the corresponding element of the state-vector, otherwise, this element is set to zero.

The cell value is obtained by the summation of the products of each state-vector element and its corresponding weight factor. To obtain the cell number, the state-vector (consisting of ones and zeros) is interpreted as representing a binary number, with question 1 representing the least significant bit.

Though the cell numbers have no particular meaning, they provide a quick way to compare two state vectors, if their cell numbers are the same, they agree in all the individual elements. This proves useful in analyzing the outcome of different trial maneuvers; several trial maneuvers may produce the same cell value but not necessarily the same state vector.

Subroutine THRTLT (Thrust Control)

Purpose.- To determine desirable throttle position. (In the current version afterburner or idle.)

Input.- Target variables and relative geometry information

Output.- Throttle setting: TPOST = 0 for idle thrust

TPOST = 1 for military thrust

TPOST = 2 for afterburner thrust

Method.- The subroutine THRTLT is called as the last routine during every integration cycle. It will set the thrust to afterburner thrust unless one of the following two conditions prevails:

1. $X_{AT} > 0$ and

absolute value of target's angle off < 60° and

absolute value of target's deviation angle < 60° and

range < 5000 feet and

range > 3000 feet and

closure rate > 300 feet/sec and

$\bar{\theta}$ < 60°

2. $X_{AT} < 0$ and

absolute value of attacker's deviation angle > 10° and

absolute value of attacker's deviation angle < 60° and

range < 1500 feet and

closure rate > 300 feet/sec

where X_{AT} is the attacker's x coordinate in the target's body axes system.

Subroutine TRYNXT (Try Next Maneuver)

Purpose.- To set up the trial maneuvers to be evaluated during the current decision process.

Input.- Target variables, predicted attacker position previous maneuver, and maneuverplane rotation angle increment.

Output.- Total number of trial maneuvers (NTRYT). For each trial maneuver, the following quantities are determined:

- maneuverplane normal (a_{n1}, a_{n2}, a_{n3})
- maneuverplane rotation angle
- command type (straight upright (ICTRYT=1), straight inverted (ICTRYT=-1), turn (ICTRYT=0).
- load factor ratio (maneuver load factor/maximum permissible load factor)

Method.- Trial maneuvers are set up in the following order of priorities:

1. Dive recovery
2. Low speed recovery
3. Standard trial maneuver selection

As the trial maneuvers are described in some detail in volume I of this report, only additional details are listed here.

The rotation angle, ρ_s, of a maneuverplane passing through the attacker's extrapolated position, is obtained as follows:

The vector from the target's present position to the attacker's extrapolated position has, in the inertial coordinate system, the components t_x, t_y, t_z:

$$t_x = x_{ex,A} - x_{e,T}$$

$$t_y = y_{ex,A} - y_{e,T}$$

$$t_z = z_{ex,A} - z_{e,T}$$

where $x_{ex,A}$ stands for the attacker's extrapolated x-coordinate in the inertial system and $x_{e,T}$ the targets present x-coordinate in the inertial system (same meaning for y and z coordinates). This vector will be called range vector in the following derivation.

51

In the "unrotated" maneuverplane system, this range vector has components t_x', t_y' and t_z':

$$\begin{bmatrix} t_x' \\ t_y' \\ t_z' \end{bmatrix} = \begin{bmatrix} \cos\bar\theta\,\cos\bar\psi & \cos\bar\theta\,\sin\bar\psi & -\sin\bar\theta \\ -\sin\bar\psi & \cos\bar\psi & 0 \\ \cos\bar\psi\,\sin\bar\theta & \sin\bar\psi\,\sin\bar\theta & \cos\bar\theta \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

To determine ρ_s, the "unrotated" maneuverplane system has to be rotated by an angle ρ_s about its x-axis such that the range vector will lie in the x-z plane, that is, the component along the y-axis of this rotated maneuverplane will be zero. The following transformation expresses the range vector in this rotated maneuverplane system.

$$\begin{bmatrix} t_x'' \\ t_y'' \\ t_z'' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\rho_s & \sin\rho_s \\ 0 & -\sin\rho_s & \cos\rho_s \end{bmatrix} \begin{bmatrix} t_x' \\ t_y' \\ t_z' \end{bmatrix}$$

Specifically, the t_y'' component (the one which has to vanish), is:

$$t_y'' = t_y'\,\cos\rho_s + t_z'\,\sin\rho_s$$

$$= \cos\rho_s\,(- t_x\,\sin\bar\psi + t_y\,\cos\bar\psi)$$

$$+ \sin\rho_s\,(t_x\,\cos\bar\psi\,\sin\bar\theta + t_y\,\sin\bar\psi\,\sin\bar\theta + t_z\,\cos\bar\theta)$$

The cosines and sines of the angles $\bar\psi$ and $\bar\theta$ are known and are:

$$\sin\bar\psi = \frac{\dot y_{e,T}}{V_{h,T}} \qquad\qquad \cos\bar\psi = \frac{\dot x_{e,T}}{V_{h,T}}$$

$$\sin\bar\theta = \frac{-\dot z_{e,T}}{V_T} \qquad\qquad \cos\bar\theta = \frac{V_{h,T}}{V_T}$$

with $V_{h,T} = (\dot{x}_{e,T}^2 + \dot{y}_{e,T}^2)^{1/2}$

and $V_T = (\dot{x}_{e,T}^2 + \dot{y}_{e,T}^2 + \dot{z}_{e,T}^2)^{1/2}$

Setting t_y'' equal to zero and substituting the above defined quantities yields:

$$\rho_s = \text{arc tan} \frac{-\dot{y}_{e,T} t_x + \dot{x}_{e,T} t_y}{(\dot{x}_{e,T} \dot{z}_{e,T} t_x + \dot{y}_{e,T} \dot{z}_{e,T} t_y - V_{h,T}^2 t_z)/V_T}$$

If the target's deviation angle is less than 90°, a trial maneuver is added consisting of a turn in the maneuverplane closest to the opponent (the discrete equivalent of the above calculated maneuverplane) so that the flight path would approximately intersect the attacker's extrapolated position. This maneuver is only added if the required load factor is less than the maximum permissible load factor.

The calculation of the required load factor is performed in a maneuverplane coordinate system defined by the three Euler angles $\bar{\psi}$, $\bar{\theta}$, and ρ_s and with origin at the target's present position. This is the maneuverplane containing the attacker's predicted position. The direction cosine matrix between this coordinate system and the inertial system is $\bar{D}$. Figure 7 depicts this situation in the plane defined by the $-z_m$ and x_m axes.

$$\cos\Gamma = \frac{z_{m_A}}{D} \qquad \cos\Gamma = \frac{D/2}{R}$$

$$R = \frac{D^2}{2 z_{m_A}}$$

Assuming constant speed along the "intercept" flight path and constant normal acceleration, the required normal acceleration a_n is:

$$a_n = \frac{V^2}{R}$$

where V = predicted constant target velocity

and   R = radius of curvature of the intercept flight path.

Substituting the above value for R, the required normal ac-
cleration for the intercept path in the $-z_m$ $-x_m$ plane is

$$a_n = 2\frac{V^2}{D^2} z_{m_A}$$

At the origin of the intercept path, the components of the gravity
force in the maneuverplane coordinate system are mg $\overline{d}_{13}$, mg $\overline{d}_{23}$,
and mg $\overline{d}_{33}$. The first component results in a tangential acceler-
ation or deceleration, the second component is directed perpendic-
ular to the maneuverplane and has therefore to be compensated by
the lift, while the third component is in the direction of or op-
posite to the required normal acceleration. Assuming this third
component to remain constant during the entire intercept turn, the
following normal acceleration has to be provided by the lift:

$$a_n^I = 2\frac{V^2}{D^2} z_{m_A} + g \overline{d}_{33}$$

In addition to the component in the maneuverplane, a force
component has to be generated by the lift perpendicular to the ma-
neuverplane (to compensate for the gravity component out of the
maneuverplane). The magnitude of this force is:

$$|a_n^{II}| = |\overline{d}_{23}| \; mg \; |$$

The total acceleration due to lift is the vector sum of a_n^I and a_n^{II},
which, when divided by g is the required load factor.

In an attempt to provide a load factor for this maneuver such
that the AML aircraft would turn "inside" the opponent's turn, a
slightly higher load factor than the one calculated above is used

for this trial maneuver. If we call the above calculated normal acceleration and that actually used $a_{n\ act}$ then:

$$a_{n\ act} = 1.1\ a_{n\ des} + 0.1\ g.$$

At the end of the subroutine TRYNXT, the selected load factors are tested to see whether they would cause pilot blackout during the next decision interval. If they will, the load factor for those trial maneuvers are reduced to a value less than 5 g's.

A final check in TRYNXT concerns the feasibility of the trial maneuverplanes. Maneuverplanes are considered feasible if the magnitude of the lift vector is sufficient to compensate the gravity force along the y_m axis and to produce a normal acceleration which is directed along the negative z_m axis.

For flight in a maneuverplane wherein $\rho \neq 0$, the lift vector must have a magnitude of at least $W \cos\bar{\theta}$. This follows because to fly in the maneuverplane, the magnitude of the lift component along the y_m axis must be $|W \cos\bar{\theta} \sin\rho|$ and in order that the concave side of the flight path be toward the negative z_m axis the magnitude of the lift component along the $-z_m$ axis must be $\geq |W \cos\bar{\theta} \cos\rho|$. Hence,

$$L \geq \sqrt{W^2 \cos^2\bar{\theta} \sin^2\rho + W^2 \cos^2\bar{\theta} \cos^2\rho} \qquad \text{or}$$

$$L \geq |W \cos\theta|$$

Figure 8 illustrates these relationships.

Information Flow Between Main
Program and Subroutines

Figure 9 shows in slightly simplified form the sequence in which routines are called by the main program AMLOFL. It also shows which other subroutines are called by these routines.

Any line connecting the name of two subroutines indicates that the routine on the left hand side calls the subroutine on the right hand side. Broken and solid lines are used simply for clarity of the drawing.

The subroutines are grouped in three levels. In the left column (level 1) are those subroutines which are called by the main program. They are listed in the sequence in which they are called.

Level two routines are listed in the center column. Subroutines may be listed more than once and in more than one column, if they are called at different levels. For instance, subroutine CLOSS is called directly from the main program, but also from level 1 and from level 2 subroutines.

In order to keep the figure uncluttered, the subroutine CMTRX is omitted. It would appear in level 2, being called by EQMOTA, EQMOTT, REACTA and REACTT, and in level 3 being called by TRYNXA, TRYNXT and RRETNW.

With a few exceptions, all the subroutines listed in Figure 9 are described in the preceding sections. Routines not described elsewhere in this report are:

GG, GGG and GGGG: These are function subroutines used for interpolation.

RANDU: A subroutine to generate uniformly distributed pseudo random numbers.

TRAPL: Routine to generate CALCOMP plots.

PROGRAM INPUT AND OUTPUT

Input Data Requirements

The input data required to run the off-line program can be classified into three groups. The first group describes the atmospheric model used in the simulation. A second group of data defines the two fighter aircraft engaged in the simulation. These first two sets of data remain unchanged from one engagement to the next during a computer job. For each engagement simulated, a set of cards of the third group is needed. These data define the geometrical initial conditions, the weight factors for the calculation of cell values and such run control parameters as duration of engagement, and requests for printing and plotting. Several engagements may be simulated in one job, each engagement requires an additional set of input variables of this third group.

Input data cards are grouped into eight individual groups. The data cards for each group must be preceded with a card which contains a code word. This code word consists of four characters punched in columns 1 through 4.

Atmospheric Model Definition.- Code Word: ATMO

(a) The speed of sound c_s (FORTRAN name CSO) in ft/sec is tabulated as a function of altitude at 500 feet intervals from sea level to 60,000 feet.

     121 values   13 cards   FORMAT (10F8.7)

(b) Air density ρ (FORTRAN name RHO) in lbs sec^2/ft is tabulated as a function of altitude at 500 feet intervals from sea level to 60,000 feet.

     121 values   13 cards   FORMAT (10F8.7)

Aircraft Identification and Wing Area.- Code Word: ATKR or TRGT.

If both aircraft are to be identical, this code word is left blank. In this case, only one set of data cards describing the aircraft will be read. If the code word is ATKR, a second set of data cards describing the target aircraft has to follow the set for the attacker aircraft and vice versa.

The aircraft identification card may contain a second four-character word, punched in columns 5 through 8. This label is used to identify the printout of the aerodynamic data.

The aircraft identification card is followed by one card containing the wing reference area (FORTRAN name SURF) in feet^2.

    1 value   1 card   FORMAT (10F8.7)

Load Factor Information.- Code Word: LDFT

(a) The maximum permissible load factor (FORTRAN name F4VG) in g's is tabulated as a function of altitude and Mach number. One card each for the following altitudes (sea level, 15,000, 30,000, 45,000 and 55,000 feet) contains maximum load factors for the following 12 Mach numbers (0.2, 0.5, 0.8, 0.9, 1.0, 1.1, 1.2, 1.5, 1.8, 2.0, 2.0, 2.2 and 2.4)

    60 values   5 cards   FORMAT (15F5.2)

(b) The sustained load factor (FORTRAN name F4SG) in g's is tabulated as a function of altitude and Mach number for the same altitudes and Mach numbers as the maximum load factor.

    60 values   5 cards   FORMAT (15F5.2)

Engine Performance Data.- Code Word: ENGN

All thrust data is given for one engine, the program multiplies thrust by two, assuming two engine aircraft.

(a) Idle thrust (FORTRAN name TIDLE) in lbs. is tabulated as a function of altitude and Mach number. There are two cards for each of the following altitudes (sea level, 10,000, 20,000, 30,000,

40,000, 50,000 and 60,000 feet) containing thrust values for the 14 Mach numbers (0.2, 0.4, 0.6, 0.8, 0.9, 1.0, 1.1, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2 and 2.4)

98 values   14 cards   FORMAT (7F10.0)

(b) Military thrust (FORTRAN name TMIL) in lbs is tabulated as a function of altitude and Mach number. The same seven altitudes and 14 Mach numbers as for the idle thrust table are used.

98 values   14 cards   FORMAT (7F10.0)

(c) Afterburner thrust (FORTRAN name TAB) in lbs. is tabulated as a function of altitude and Mach number. The same seven altitudes and 14 Mach numbers as for the idle thrust table are used.

98 values   14 cards   FORMAT (7F10.0)

Aerodynamic Data.- Code Word: ARDT

(a) Angle of attack α with slats/flaps fully retracted (FORTRAN name F4ALP) in degrees is tabulated as a function of lift coefficient (C_L) and Mach number. Each card contains the angle of attack for one value of lift coefficient and 10 values of Mach numbers. The lift coefficient ranges from 0 to 2.0 in equal intervals of 0.1. The following 10 points of Mach numbers are used (0.2, 0.5, 0.8, 0.9, 1.0, 1.1, 1.2, 1.5, 1.8 and 2.0).

210 values   21 cards   FORMAT (10F8.7)

(b) Angle of attack α with slats/flaps fully extended (FORTRAN name F5ALP) in degrees is tabulated as a function of lift coefficient and Mach number for the same values of coefficient of lift and Mach numbers as under (a) above.

210 values   21 cards   FORMAT (10F8.7)

(c) Coefficient of drag C_{D_U} with slats/flaps fully retracted (FORTRAN name F4 CDR) is tabulated as a function of lift coefficient and Mach number for the same values of coefficient of lift and Mach numbers as under (a) above

210 values   21 cards   FORMAT (10F8.7)

(d) Coefficient of drag C_{D_D} with slats/flaps fully extended (FOR-
TRAN name F5CDR) is tabulated as a function of lift coefficient and
Mach number in the same arrangement as given under (a) above.

210 values   21 cards   FORMAT (10F8.7)

Limits.- Code Word: LMTS

(a) Maximum lift coefficient $C_{L_{max}}$ (FORTRAN name CLMAX) is tabu-
lated as a function of Mach number for 14 Mach numbers (0.2, 0.4,
0.6, 0.8, 0.9, 1.0, 1.1, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4)

14 values   1 card   FORMAT (15F5.2)

(b) Minimum Mach number (FORTRAN name F4MMI) is tabulated as a
function of altitude for 13 values (sea level to 60,000 feet in in-
crements of 5,000 feet)

13 values   1 card   FORMAT (15F5.2)

(c) Maximum Mach number (FORTRAN name F4MMA) is tabulated as a
function of altitude for the same 13 altitudes as in (b) above.

13 values   1 card   FORMAT (15F5.2)

(d) Dive recovery angle (FORTRAN name RECANG) in degrees is tabu-
lated as a function of altitude and Mach number. There are ten
cards for altitude intervals of 5,000 feet from sea level to
45,000 feet. (Dive recovery has no limit above 45,000 feet.)
Each card contains values for 12 Mach numbers (0.2, 0.5, 0.8, 0.9,
1.0, 1.1, 1.2, 1.5, 1.8, 2.0, 2.2, 2.4)

120 values   10 cards   FORMAT (15F5.2)

Speed Brake Drag Coefficient.- Code Word SPBR
(a) The drag coefficient increment (FORTRAN name SBCDR) for fully
deflected speed brakes is tabulated as a function of the 14 Mach
numbers (0.2, 0.4, 0.6, 0.8, 0.9, 1.0, 1.1, 1.2, 1.4, 1.6, 1.8,
2.0, 2.2 and 2.4)

14 values   1 card   FORMAT (15F5.2)

This concludes the input defining one aircraft. When data is to be read for a second aircraft (attacker and target different aircraft types), the preceding sequence, starting with aircraft identification and wing area, is repeated.

The variables F4ALF, F5ALF and RECANG are converted internally to radians.

The final card of the input defining the aircraft is a control card for printing of all the data read so far. If this card contains the code word PRNT in the first four columns, atmospheric data and aircraft data will be printed. If the first four columns contain anything else than the code word PRNT, printing of these tables will be suppressed.

Input data for Each Individual Engagement.-

Each individual engagement requires 10 input cards. These are:

(1) Title card.- The title card is used to identify the run. It is printed out at the beginning of a run. The title card also serves to terminate a job; if the first three columns contain the word END and column 4 is blank, the computer terminates the present job.

      1 card   FORMAT (20A4)

(2) Attacker weight.- (FORTRAN name WA); weight of attacker aircraft in lbs.

      1 Value   1 Card   FORMAT (8F10.4)

(3) Attacker geometrical initial conditions.- This card contains the following information:

Position in inertial frame $x_{e,A}$, $y_{e,A}$, $z_{e,A}$ (feet)

Magnite of velocity vector V_A (ft/sec)

Euler angles of wind axis system ψ, θ, ϕ (deg)

      7 Values   1 Card   FORMAT (8F10.4)

(4) <u>Parameters for attackers tactical decision process</u>.- Initial maneuver index (1.0 for straight flight upright, -1.0 for straight flight inverted, 0.0 for turn in a maneuverplane)

Load factor ratio (actual initial load factor divided by maximum allowable load factor). This input has meaning only if the initial maneuver selected is a turn in a maneuverplane; for straight flight the program will determine the appropriate load factor independent of this input.

Initial throttle position (0.0 for idle, 1.0 for military power, 2.0 for afterburner).

Time between tactical decisions (sec).

Maneuverplane rotation angle resolution N (where $\Delta\rho = 90°/(N+1)$)

> 5 values   1 card   FORMAT (8F10.4

(5) <u>Weight factors</u>.- The weight factors for the cell value determination for the 15 questions for the attacker's state vector.

> 15 values   1 card   FORMAT (8F10.4)

(6) to (9) <u>Target data</u>.- These four cards are equivalent to cards 2 to 5, but with data pertaining to the target aircraft.

(10) <u>Run control parameters</u>.- Each engagement is controlled by the following parameters:

> Starting time in seconds
> End time in seconds
> Integration step size in seconds
> Number of integration steps between printout
>   of attacker and target data
> Flag for suppression of printout of trial maneuver
>   information (0.0 if printout should occur, 1.0
>   to suppress printout)
> Number of integration steps between recording of
>   data on tape for plotting. (If this variable
>   is set to 0.0, no plot-tape is prepared.)

>   6 values   1 card   FORMAT (8F10.4)

These 10 cards are repeated for each engagement to be simulated in one job. A title card with END in columns 1 through 3 and column 4 blank terminates a job.

Figure 10 illustrates the input data deck setup for a job which simulates two engagements in which attacker and target aircraft are different aircraft types.

Program Output

In contrast to the real time version of the AML program where the output consists simply of a few variables such as u, v, w and p, q and r which are transferred to the DMS program, the off-line program has options to select extensive printout reflecting the progress of an air-to-air engagement. In addition to this printed output, information about the engagement may be recorded on magnetic tape and later processed to produce several plots to visually display the situation as it occured during the flight.

The printed output of the off-line program can be divided into four groups as follows:

<u>Atmospheric and aircraft definition printout</u>.- This printout occurs if the print control option on the final card for the aircraft definition input was selected (code word PRNT). The speed of sound and air density tables are printed and all the data which were read in to define the aircraft performance data.

<u>Run control parameters and initial conditions</u>.- This information is printed out at the beginning of each engagement. Initial conditions, run control parameters, the 15 questions defining the attacker's state vector, the 15 questions defining the target's state vector and information about whether or not the aircraft maneuvers are constraint by the possibility of pilot blackout.

<u>Target and attacker data</u>.- This printout occurs during the simulation of an engagement. The time interval between printouts is determined by a run control parameter (the fourth parameter on the tenth card of the input set for an individual engagement). This printout is illustrated in Figures 11 and 12. The symbols used in this printout are self-explanatory. Note that the line of sight angle for one aircraft is equal to 180 minus the angle off of the other aircraft.

<u>Information about tactical desision process</u>.- This printout occurs if the fifth parameter of the run control parameters (card 10) was set to zero. Every time the attacker or the target makes

64

a tactical decision, the information as illustrated in Figure 13 is printed. The symbols used in this printout are explained in Table II.

Error Messages

The off-line program has a number of error conditions which cause the program to terminate. When one of these conditions occurs, the message

EXIT NUMBER N

is printed. The number N=1 indicates a normal exit, while N=4 to 11 indicates the following erroneous conditions:

N	Subroutine Called From	Error Condition
4	AERF4	Negative Mach number or altitude
5	AMLOFL	Code word ATMO missing
6	AMLOFL	Code word LDFT missing
7	AMLOFL	Code word ENGN missing
8	AMLOFL	Code word ARDT missing
9	AMLOFL	Code word LMTS missing
10	AMLOFL	Code word SPBR missing
11	AMLOFL	Second data set for same aircraft as first data set

INTERACTION BETWEEN THE DMS AND AML PROGRAM

General Considerations

The objective of this section is to outline how the real-time version of the AML program interacts with the DMS program, which was written by LRC personnel. The underlying philosophy in designing this program interface was to concentrate all the interaction of the two programs in as few subroutines as possible and to keep as many of the AML subprograms identical in both the off-line and the real-time version. This design philosophy permits improvements in the AML off-line program to be transferred to the AML-DMS program with a minimum of effort. On the other hand, changes in the DMS program should not affect the operation of the real-time version of the AML program.

The interaction between the DMS program and the AML program occurs in the DMS program RTMAIN, which is the DMS main program for the real-time operation and the AML subroutine AMLVS2, which is the driver program for all the AML subroutines.

The program AMLVS2 is called by RTMAIN in three different ways. A "CALL AMLVS2(1)" is used only once during an entire simulation. This call affects initialization of certain constants and reading of the input data defining the AML aircraft. In the following discussion, it is assumed that the AML program represents aircraft 2. A "CALL AMLVS2(2)" is used during resetting to establish new initial conditions. Finally, a "CALL AMLVS2(3)" is used during the real-time OPERATE loop; once every integration cycle this call is made. Here, the AML program performs the calculations for the equations of motion and at the appropriate times, a new tactical decision is performed.

Figure 14 represents a simplified block diagram of the program RTMAIN emphasizing those portions of the program where variables between the DMS program and the AML program are exchanged. Most of this exchange occurs actually in the subprogram AMLVS2,

and Figure 15 further illustrates the interchange of variables. The blocks exhibiting variable interchanges in Figure 14 are drawn with double frames. Before discussing in some detail how the flow of information back and forth occurs in the two programs, it should be noted that most AML variables are unsubscripted variables ending with the character "T", while the DMS variables are subscripted such that a 1 indicates attacker variables (the DMS driven aircraft), a subscript 2 indicates target variables (the AML driven aircraft). All the information transfer between the two programs is of the form:

           AML VARIABLE = DMS VARIABLE
or
           DMS VARIABLE = AML VARIABLE

While in some instances, equivalence statements or other means could be used for information exchange between the two variables. It was found that this method provides the most independence between the two programs.

DMS Program RTMAIN

The simplified block diagram of RTMAIN as represented in Figure 14 not only indicates where variables between the DMS and AML program are exchanged, but it also indicates the sequence in which these variables are calculated and where the integration of the pertinent state variables occurs.

The numbers to the left above the blocks in Figure 14 represent statement numbers in the program RTMAIN. Numbers in parenthesis indicate that additional comments are given in the explanations below.

(1) The program will come to this point at the beginning of a reset cycle. This can occur either after a program start up or from block (11) if the operator presses the RESET button on the console.

(2) The flag MODEAML is set to 2 for initialization of the AML routines. Initial Euler angles and body rotational rates are read in from the console.

(3) The subroutine IC establishes the initial conditions of the positions and attitudes of the subject aircraft. Note that SX(1) and SY(1) are always set to zero, thus defining the attacker to start at the origin of the x-y axes system. $\dot{p}$, $\dot{q}$, and $\dot{r}$ are set to zero for both aircraft.

The subroutine QIC essentially establishes the initial conditions of the quaternions. The call to QIC for aircraft 2 also calculates the initial inertial coordinates for aircraft 2, that is, SX(2) and SY(2). Note that altitude is input for both aircraft in the inertial system.

(4) The array DERINT (1,J) contains all the variables which are integrated by the standard DMS integration routine (a third order Adams-Bashforth predictor method). The variable DERINT (2,J) is the derivative of the variable DERINT (1,J). The inertial linear accelerations of aircraft 2 are calculated in the AML program and are integrated by the DMS program. The result of this integration is XEDOTT, YEDOTT, ZEDOTT. In this block, the initial values of XEDOTT, YEDOTT and ZEDOTT are transferred into the corresponding positions in the array DERINT.

(5) This is the reentry point into the operate loop after the DMS program has completed one integration cycle. As long as the real-time OPERATE remains selected, block (18) will cause the program to return to block (5).

(6) This is the entry point for the HOLD condition. If HOLD is selected, the return from block (11) will be to block (6).

(7) The DMS program calculates some geometrical variables pertaining to aircraft 1. These are attacker variables which are used by the AML program, specifically, velocities in inertial frame and the direction cosine matrix of the attacker.

68

These same variables are calculated for the target aircraft, specifically, the direction cosine matrix of the target body axes system as a function of the target's quaternions that is

$$CD11(2) = AB1(2)^2 - AB2(2)^2 - AB3(2)^2 + AB4(2)^2$$

$$\vdots$$

$$CD33(2) = (AB1(2), AB2(2), AB3(2), AB4(2))$$

The target's Euler angles as functions of these direction cosine matrix are calculated

$$PSI(2) = ATAN2 (CD12(2), CD11(2))$$
$$PHI(2) = ATAN2 (CD23(2), CD33(2))$$
$$THETA(2) = -ASIN (CD13(2))$$

The DMS subroutine VELOC calculates the target's velocity components in the inertial system:

$$SXDOT(2) = CD11(2) * U(2) + CD21(2) * V(2) + CD31(2) W(2)$$

and similar expressions for

$$SYDOT(2) \quad and \quad HDOT(2)$$

(8) This is the block indicating the most important interaction between the DMS program and the AML subprogram. Note that the variable MODEAML in this call has either the value 2 (during RESET) or 3 (during OPERATE). A detailed list of variables interchanged between the two programs during this call is given in Figure 15.

(9) This entire block is strictly associated with DMS calculations of forces, moments and so forth.

(10) Both DMS and AML variables which are recorded during an engagement and which are displayed on the scope as well as those variables driving the projection equipment are output at this time.

(11) This block represents a call to RTMODE and depending upon what mode is selected on the simulation console by the operator, the program will return to block (1) or (5) or (6) or (12).

(12) This is the entry point to operate. Like operating an analog computer, the mode OPERATE essentially calls for the integration of the derivatives of all the state variables. This is done in the DMS program in blocks (15) and (17).

(13) The variable MODEAML is set to 3, indicating to the AMLVS2 subprogram that it should perform the normal calculations of one integration cycle. MODEAML remains 3 until the operator selects the mode RESET.

(14) The accelerations as calculated in EQMOTT of the AML program are transferred into the DERINT array for integration by the DMS program.

(15) Execution of the subroutine IGRATE1 updates the state variables in the array DERINT

$$DERINT(1,J) = DERINT(1,J) + \int_{t}^{t+dt} DERINT(2,J)dt$$

(16) The AML variables for the inertial velocities are set equal to the ones obtained in block (15).

(17) The DMS program uses essentially three different methods of integration. While most of the state variables are integrated in block (15), a refined integration method is used for the quaternions and for the linear velocities. A call to QUAT integrates the quaternions for the subject aircraft (subscript I)

$$AB1(I) = AB1(I) + \int_{t}^{t+dt} AB1DOT(I)dt$$

and the same for AB2, AB3 and AB4.

The subroutine LVELOC integrates the inertial velocities to inertial coordinates

$$SX(I) = SX(I) + \int_{t}^{t+dt} SXDOT(I)dt$$

SXDOT(I) is calculated as a function of U(I), V(I) and W(I) and the direction cosines of the subject aircraft. The subroutine LVELOC uses a Taylor series expansion of the variables SX(I), SY(I), H(I), truncated after the quadratic term.

(18)   This is the end of the normal calculation cycle, which is now repeated, starting at block (5).

Information Transfer

Table III summarizes the information transfer in terms of FORTRAN variable names. Note that each AML variable is either equated to a variable representing the same quantity in the DMS program (and vice versa) or is computed in an AML subroutine. Also note that whenever a variable of one program is quoted to the corresponding variable of the other program, this latter variable must appear somewhere in this table and it must be indicated where this variable is calculated. To illustrate, U(2) is equated to UT which in turn is calculated in the AML subroutine EQMOTT. Only variables transferring information back and forth associated with the target aircraft are listed in the table, transfer of attacker information to the AML program is straight forward.

It is worthwhile to note that the DMS program, by calling subroutine DIRCOS with an argument 2 effectively calculates the direction cosine matrix of the target body axes system at the end of an integration cycle; this information, however, is not transferred to the AML program which recalculates this direction cosine matrix from the Euler angles PHIO, THETO and PSIO.

CONCLUDING REMARKS

The program description provided in this report reflects the status of the AML program in early Spring, 1975. Most of the runs at this time have been made between equal or very similar aircraft. As more experience is gained in simulation of combat between dissimilar aircraft, it is to be expected that some modifications and extensions will be made to the program.

Decision Science, Inc. is currently adding to the program a capability for missile trajectory calculation to perform a weapon systems effectiveness study. This addition will permit deciding more accurately whether or not at any given instant the aircraft is within weapons firing envelope.

The AML technique developed for one-on-one air combat simulation offers capability for expansion to two-on-one and two-on-two engagements.

Decision Science, Inc.
San Diego, California, April 23, 1975.

REFERENCES

1. Anonymous: Dynamics of the Airframe. Report AE-61-411, Northrop Corporation, Norair Division, September 1952.

2. Mitchell, Ewart E. L. and Rogers, Alan E.: Quaternion Parameters in the Simulation of a Spinning Rigid Body in Simulation, the Modeling of Ideas and Systems with Computers. Edited by John McLeod, McGraw Hill Book Co., Inc. 1968.

TABLE I.- LIMITS FOR RANDOMIZATION OF VARIABLES

CODE WORD	RANDOMIZED VARIABLE	RANGE
DTPRA	Attacker decision time	$0.8DTPRA^1$, $1.2DTPRA$
TIMEPA	Attacker prediction time increment	$0.8TIMEPA^1$, $1.2TIMEPA$
GLEVLA	Loadfactor for first trial maneuver	See Subroutine TRYNXA[3]
RANGA	Range and range rate determined for each trial maneuver	$0.9RANGE^2$, $1.1RANGE$ $0.9RRATE^2$, $1.1RRATE$
LOSA	LOS angle to target and its rate as determined for each trial maneuver	$0.9FLOSSA^2$, $1.1FLOSSA$ $0.9DLOSSA^2$, $1.1DLOSSA$
ENOPA	Specific energy difference between target and attacker as determined for trial maneuvers	$0.9(SPENYT-SPENYA)$ $1.1(SPENYT-SPENYA)$
DTPRT	Target decision time	$0.8DTPRT^1$, $1.2DTPRT$
TIMEPT	Target prediction time increment	$0.8TIMEPT^1$, $1.2TIMEPT$
GLEVLT	Loadfactor for first trial maneuver	See Subroutine TRYNXT[3]
RANGT	Range and range rate determined for each trial maneuver	$0.9RANGE^2$, $1.1RANGE$ $0.9RRATE^2$, $1.1RRATE$
LOST	LOS angle to attacker and its rate as determined for each trial maneuver	$0.9FLOSST^2$, $1.1RANGE$ $0.9DLOSST^2$, $1.1DLOSST$
ENOPT	Specific energy difference between attacker and target as determined for trial maneuvers	$0.9(SPENYA-SPENYT)$ $1.1(SPENYA-SPENYT)$

1. Nominal value as specified in input.

2. Values as calculated in Subroutine RELGN for Subroutines TRYNXA or TRYNXT.

3. The range is between 0.5 to 1 of the nominal value which is set to 25% of the maximum allowable loadfactor.

TABLE II.- SYMBOLS AND UNITS IN TARGET'S TACTICAL DECISION PROCESS PRINTOUT (see Figure 13).

Symbol	Unit	Explanation
KROTST		Index for maneuver with highest priority if several maneuvers have same highest value.
NTRYT		Number of trial maneuvers analyzed.
ROTST	Deg	Rotation angle of the target's maneuverplane passing through the attacker's extrapolated position.
I		Trial maneuver number.

ISTRYT		Maneuver type flag
		  1 = straight flight upright
		  -1 = straight flight inverted
		  0 = turn in maneuverplane

ICTRYT		Turn command: Number of $\Delta\rho$ increments between "unrotated" maneuverplane and trial maneuverplane. Note: Trial maneuver 5 and 6 have ICTRYT= 8, which corresponds to ρ=80 deg., which is the maneuverplane "closest" to the opponent (since ROTST = 83.274 deg.).
G's	g	Trial maneuver load factor.
ROTRYT	Deg	Rotation angle of trial maneuverplane.
AN1TRT AN2TRT AN3TRT		Components of unit vector normal to trial maneuverplane in earth fixed coordinate system.
P SUB S	Feet/Sec	Predicted range rate at the end of trial maneuver.

75

TABLE II.- Concluded

Symbol	Unit	Explanation
COMMAND		The command consists of two items: ISTRYT and ICTRYT as explained above.
RANGE	Feet	Predicted range at end of trial maneuver.
RATE	Feet/Sec	Predicted range rate at end of trial maneuver.
CELL-STATE		Cell information for each trial maneuver. Answers to questions 1 to 15 in that order (1=YES; 0=NO).
STATE NO.		Cell number of cell-state for each trial maneuver.
STATE VALUE		The weighted sum of the answers to the 15 questions. Note that in the example in Figure 13 all weight factors are equal to one.

VARIABLE NAME	PRIMARY USE IN DMS/AML	COMMON BLOCK	EQUATED TO	IN SUB-ROUTINE	CALCULATED IN	MEANING (ALL VARIABLES ASSUMED SUBSCRIPTED WITH T)
XET	AML	BABLT	SX(2)	AMLVS2		x_e
YET	AML	VARBLT	SY(2)	AMLVS2		y_e
ZET	AML	VARBLT	HAB(2)	AMLVS2		z_e
SX(2)	DMS	BK28			DMS/LVELOC	x_e
SY(2)	DMS	BK28			DMS/LVELOC	y_e
HAB(2)	DMS	BK28			DMS/LVELOC	z_e
XEDOTT	AML	VARBLT	DERINT(1,21)	RTMAIN		$\dot{x}_e$
YEDOTT	AML	VARBLT	DERINT(1,22)	RTMAIN		$\dot{y}_e$
ZEDOTT	AML	VARBLT	DERINT(1,23)	RTMAIN		$\dot{z}_e$
ACCXET	AML	VARBLT			EQMOTT	$\ddot{x}_e$
ACCYET	AML	VARBLT			EQMOTT	$\ddot{y}_e$
ACCZET	AML	VARBLT			EQMOTT	$\ddot{z}_e$
SXDOT(2)	DMS	BK28			DMS/VELOC	$\dot{x}_e$
SYDOT(2)	DMS	BK28			DMS/VELOC	$\dot{y}_e$
HDOT(2)	DMS	BK28			DMS/VELOC	$\dot{z}_e$

TABLE III.- Continued

VARIABLE NAME	PRIMARY USE IN DMS/AML	COMMON BLOCK	EQUATED TO	IN SUB-ROUTINE	CALCULATED IN	MEANING (ALL VARIABLES ASSUMED SUBSCRIPTED WITH T)
DERINT(2,21)	DMS	INTCOMM	ACCXET	RTMAIN		$\ddot{x}_e$
DERINT(2,22)	DMS	INTCOMM	ACCYET	RTMAIN		$\ddot{y}_e$
DERINT(2,23)	DMS	INTCOMM	ACCZET	RTMAIN		$\ddot{z}_e$
DERINT(1,21)	DMS	INTCOMM			DMS/IGRATE1	$\dot{x}_e$
DERINT(1,22)	DMS	INTCOMM			DMS/IGRATE1	$\dot{y}_e$
DERINT(1,23)	DMS	INTCOMM			DMS/IGRATE1	$\dot{z}_e$
UT	AML	VARBLT			EQMOTT	u
VT	AML	VARBLT			EQMOTT	v
WT	AML	VARBLT			EQMOTT	w
U(2)	DMS	BK32	UT	AMLVS2		u
V(2)	DMS	BK32	VT	AMLVS2		v
W(2)	DMS	BK32	WT	AMLVS2		w
PT	AML	VARBLT			EQMOTT	p
QT	AML	VARBLT			EQMOTT	q
RT	AML	VARBLT			EQMOTT	r

TABLE III.- Concluded

VARIABLE NAME	PRIMARY USE IN DMS/AML	COMMON BLOCK	EQUATED TO	IN SUB-ROUTINE	CALCULATED IN	MEANING (ALL VARIABLES ASSUMED SUBSCRIPTED WITH T)
P(2)	DMS	BK05	PT	AMLVS2		p
Q(2)	DMS	BK05	QT	AMLVS2		q
R(2)	DMS	BK05	RT	AMLVS2		r
PHIϕ	AML	NOT IN C	PHI(2)	AMLVS2		ϕ
THETϕ	AML	NOT IN C	THETA(2)	AMLVS2		θ
PSIϕ	AML	NOT IN C	PSI(2)	AMLVS2		ψ
PHI(2)	DMS	BK21			DMS/DIRCOS	ϕ
THETA(2)	DMS	BK21			DMS/DIRCOS	θ
PSI(2)	DMS	BK21			DMS/DIRCOS	ψ
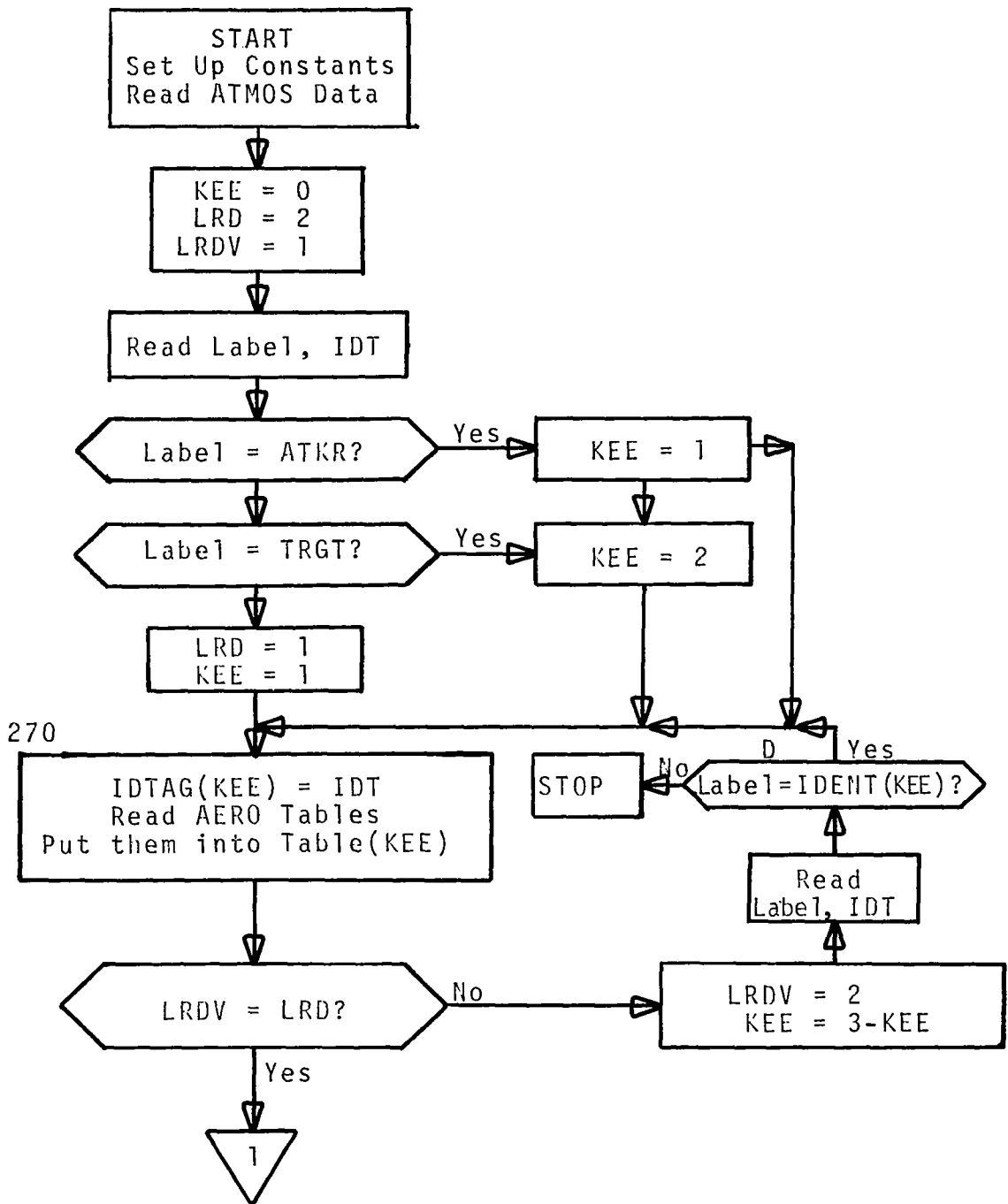
Figure 1.- Flowchart of Program
AMLOFL (Main Program).

The portion shown on this page is
described in the section "Logic for
aerodynamic data input".
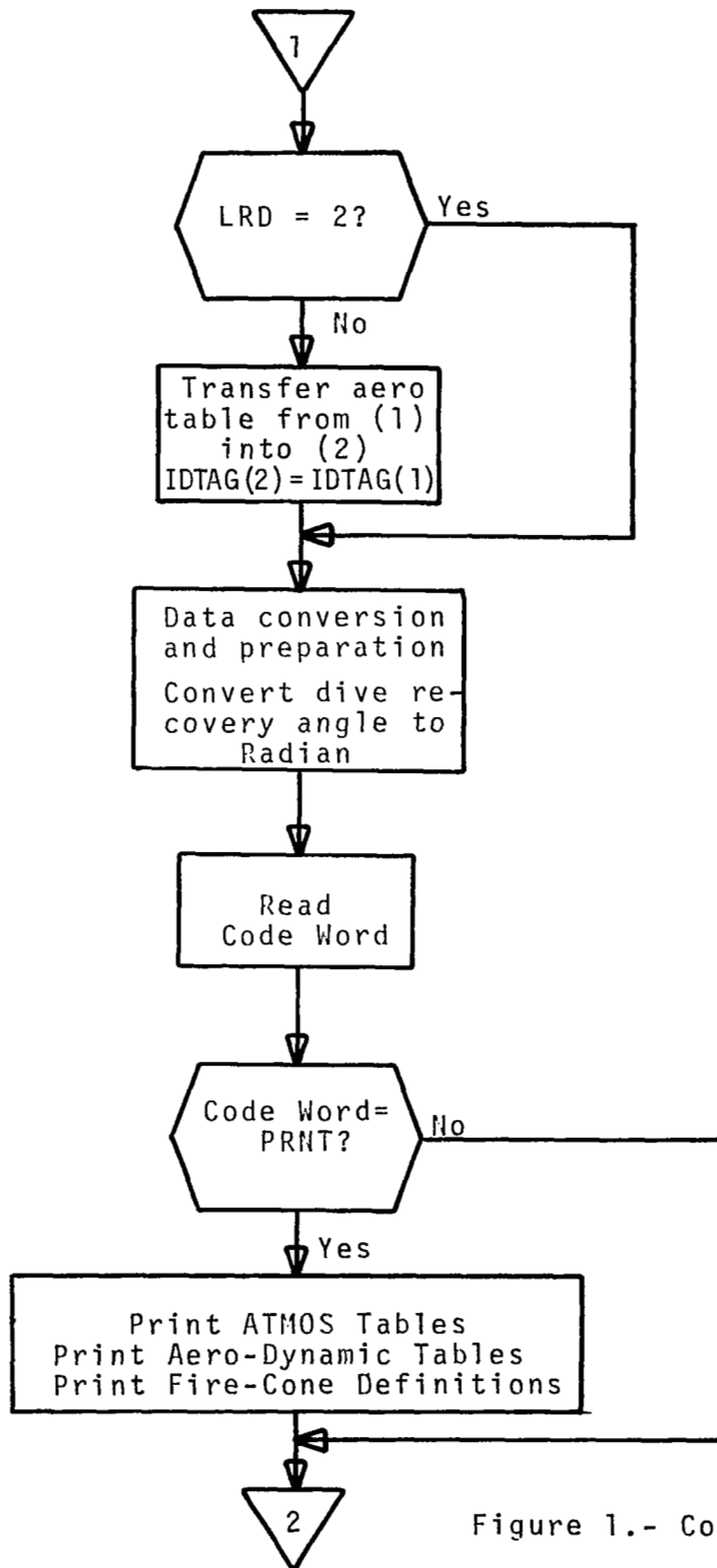
Figure 1.- Continued
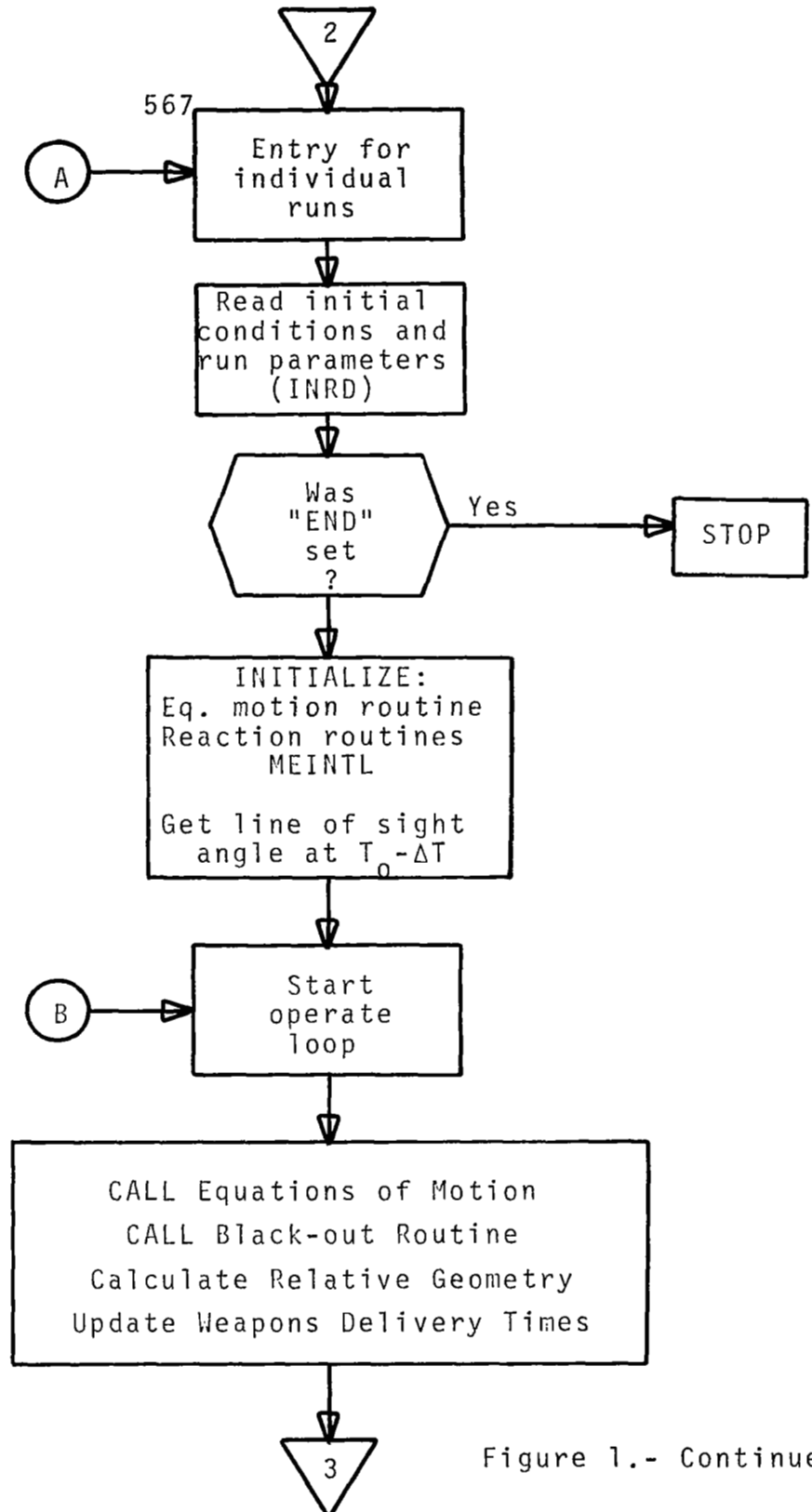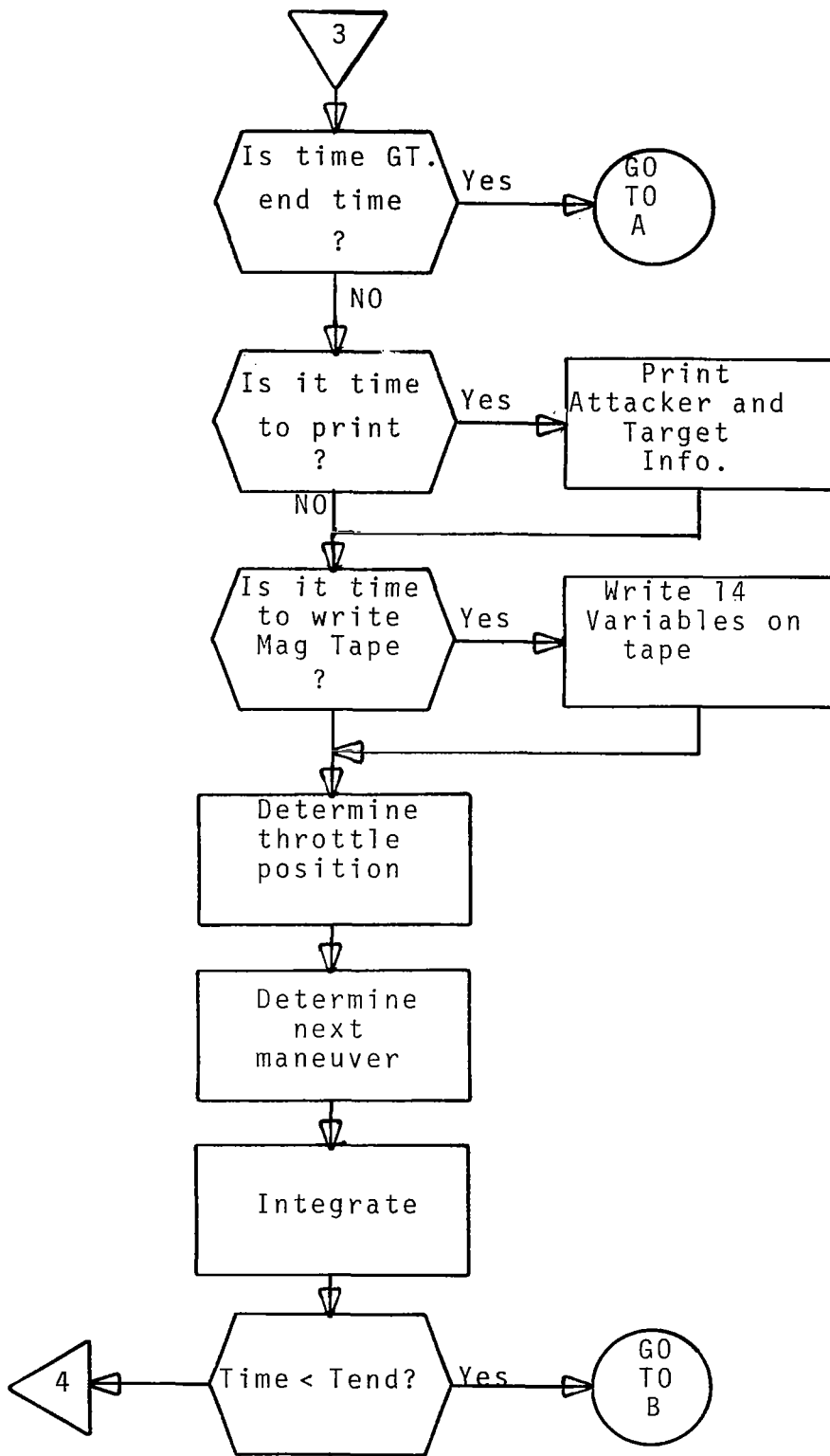
Figure 1.- Continued

Figure 1.- Continued
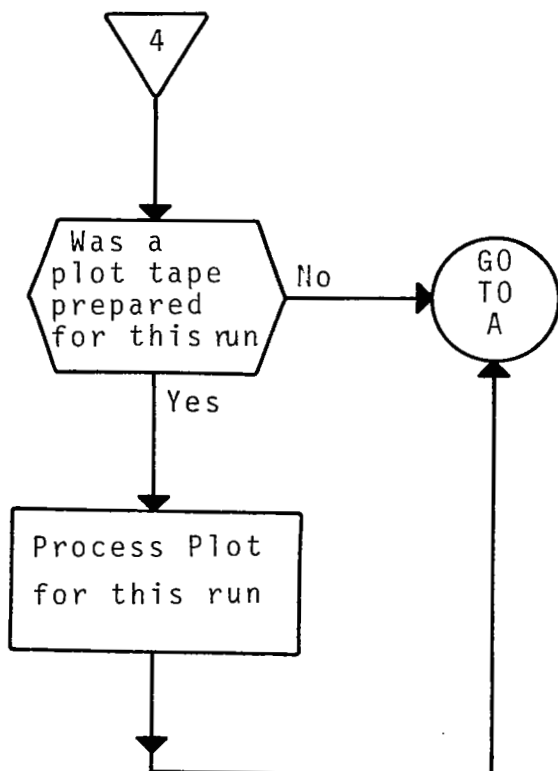
Figure 1.- Concluded
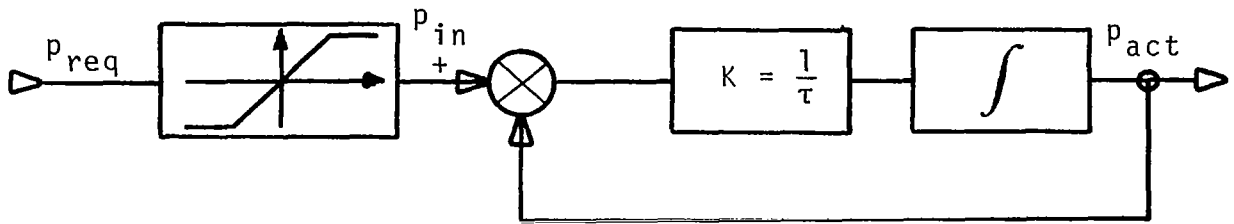
Figure 2.- Block diagram of filter for rotational rates.

p_{req} = Requested roll rate (filter input)

p_{act} = Actual roll rate (filter output)

p_{in} = p_{req} unless:

$\quad\quad p_{req} > p_{max}$: then p_{in} = p_{max}

$\quad\quad p_{req} < -p_{max}$: then p_{in} = $-p_{max}$

$p_{act}(n+1) = p_{act}(n) + \frac{dt}{\tau}(p_{in}(n) - p_{act}(n))$
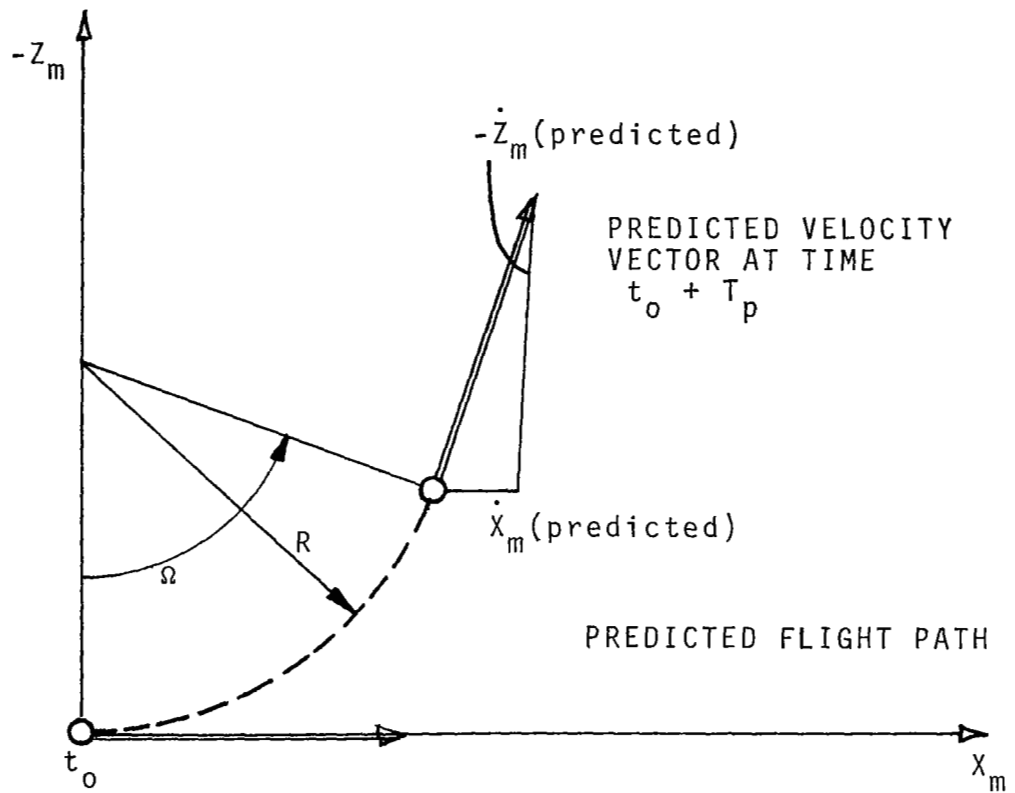
Figure 3.- Prediction of turns in maneuverplane system.

$$\omega = \frac{V}{R} \qquad \qquad \Omega = \omega T_p$$
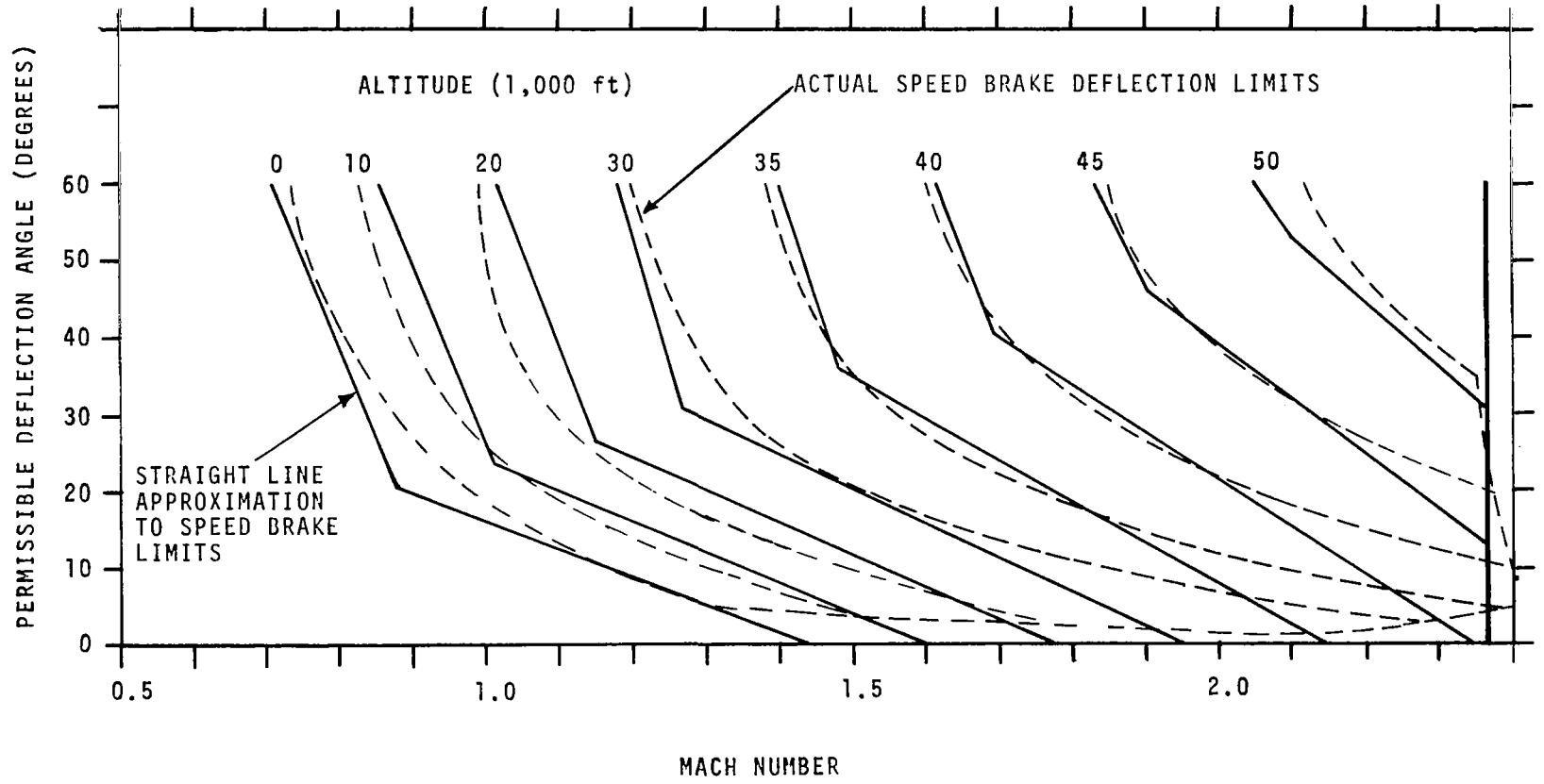
Figure 4.- Simulation of maximum permissible speed brake deflection
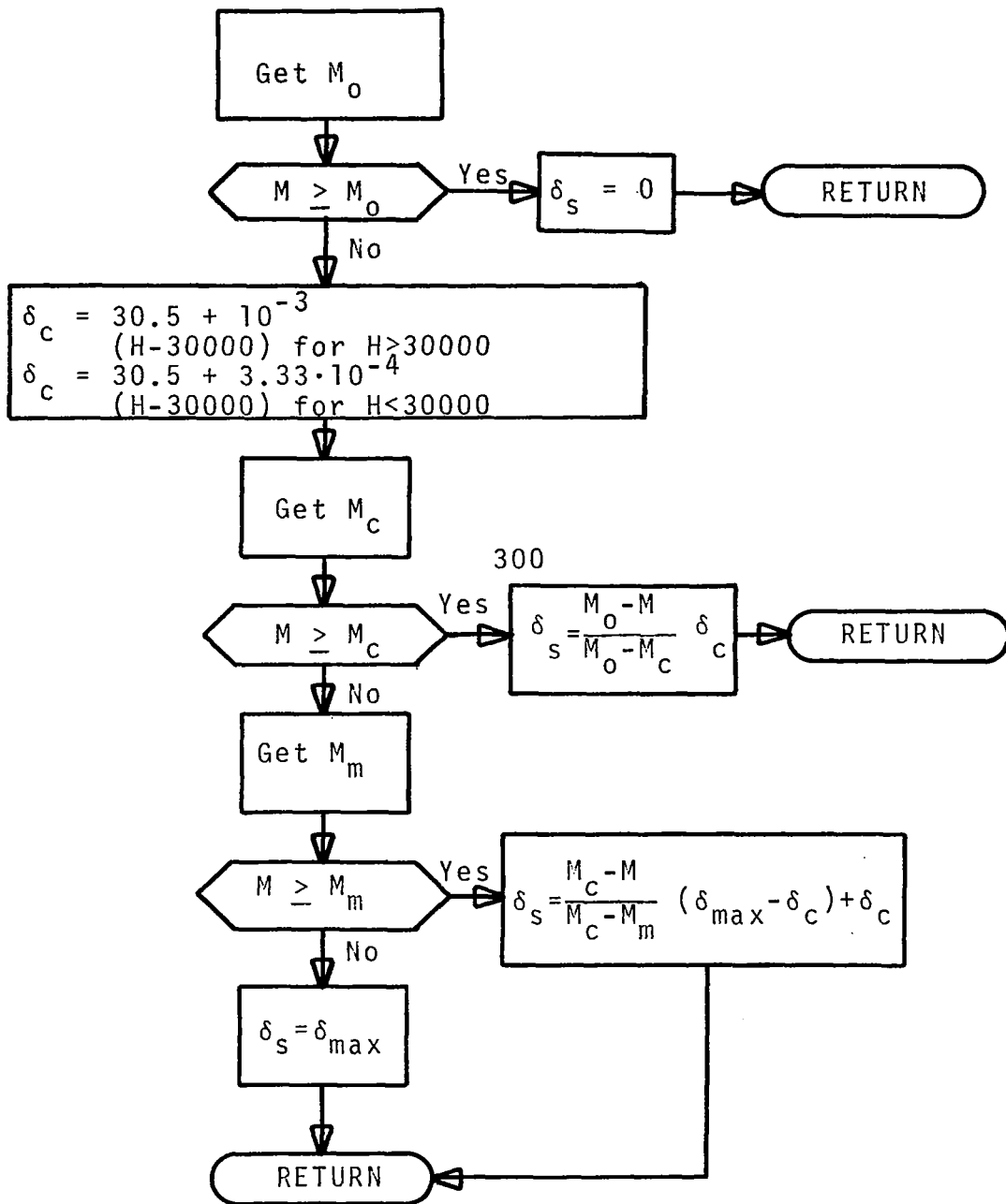
Figure 5.- Flow chart for speed brake deflection limits calculation
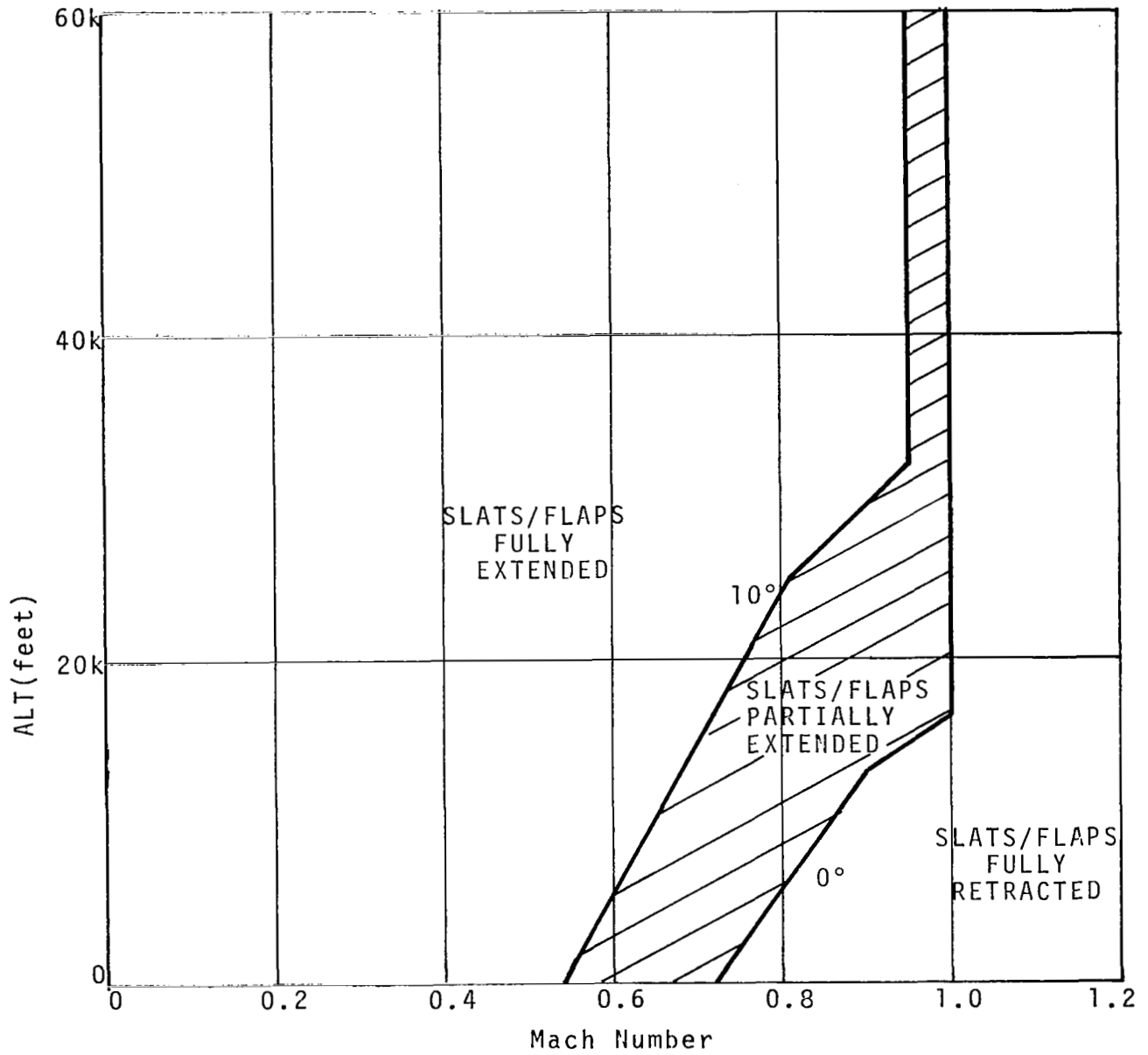
Figure 6.- Definition of slat/flap deflection angles.
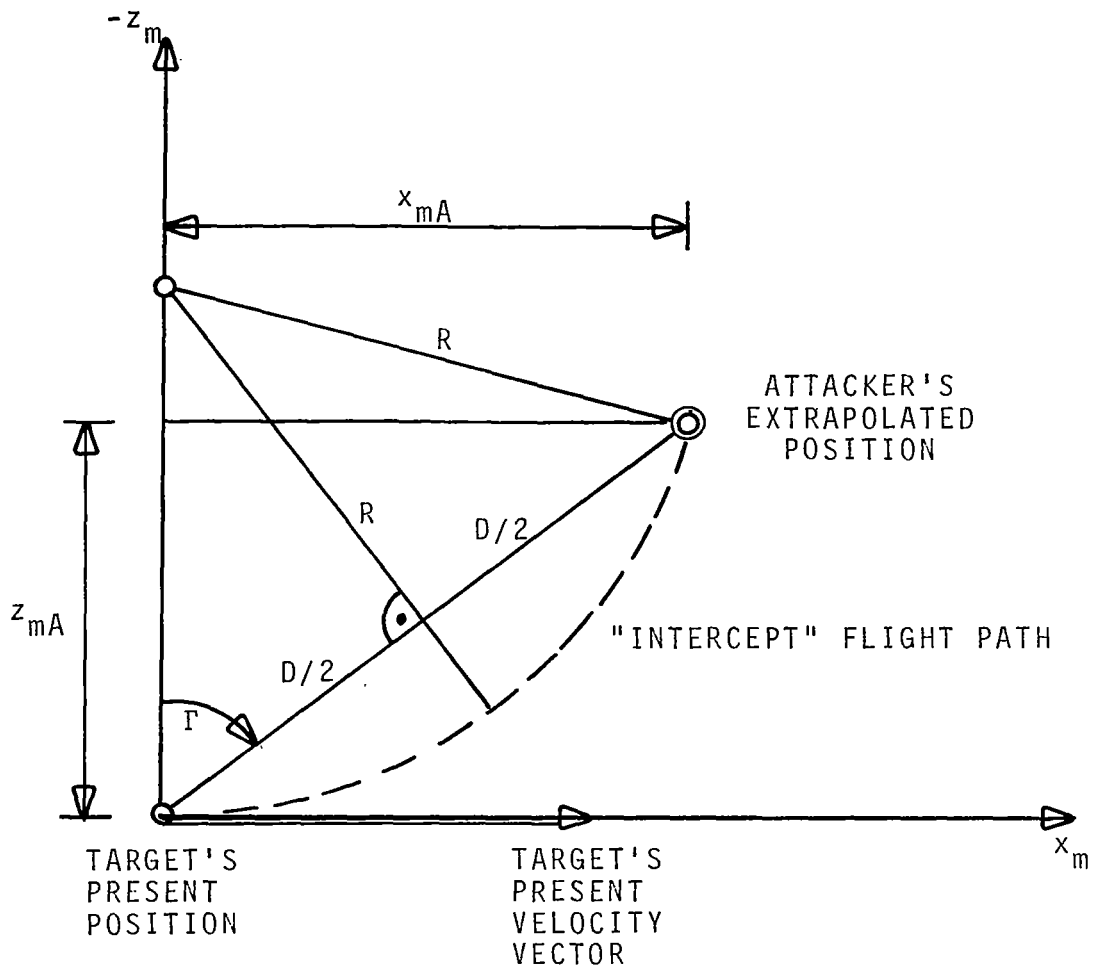
Figure 7.- Calculation of g-level required
for "intercept" maneuvers.
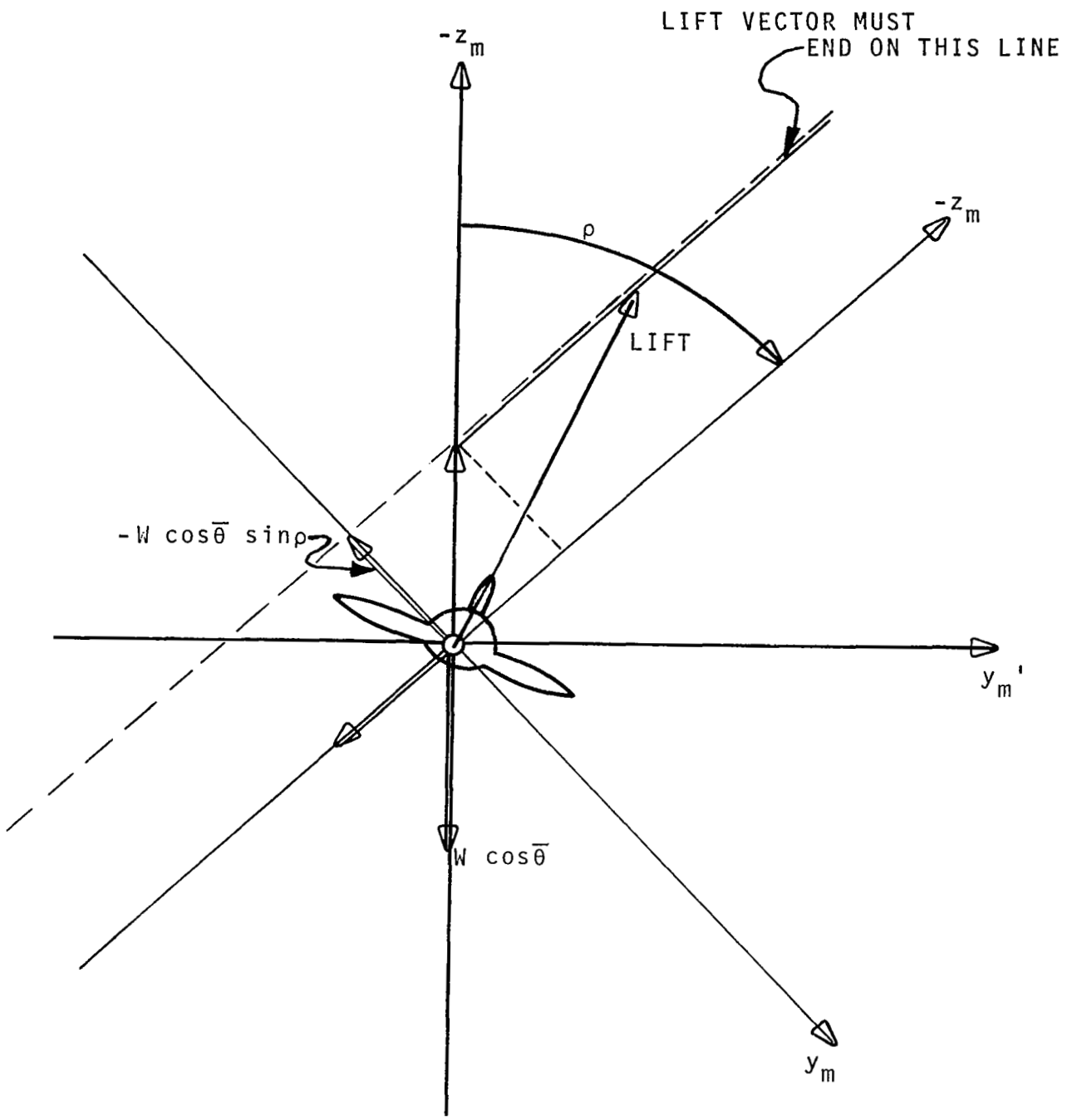
Figure 8.- Test for maneuverplane feasibility.

	LEVEL 1	LEVEL 2	LEVEL 3
JOB INITIAL-IZATION	TRAFER PCSRO PRTF4 FIRCON		
INITIALIZATION OF ONE ENGAGEMENT	INRD HUBLO EQMOTA EQMOTT REACTA REACTT MEINTL CLOSS	QUATIN NORPLN FILTRA CSRO AERF4 FILTRT FIRCON	GG GGG GGGG
OPERATE LOOP FOR ONE ENGAGEMENT	EQMOTA EQMOTT HUBLO RELGN MEINTL STATEA STATET PAIRCR PCELL THRTLA THRTLT REACTA REACTT	NORPLN CSRO AERF4 OILER FILTRA FILTRT QUATEX QUAT CLOSS FIRCON RELGN PRESR PRETNW TRYNXA TRYNXT RADIFY STATEA STATET EXTRA EXTRT	GG GGG GGGG SBDEF SLAPDF CLOSS HUBLO CLADES GETRXN NORPLN RANDU
JOB TERMIN-ATION	TRAPL		

Figure 9.- Information flow between
main program and subroutines
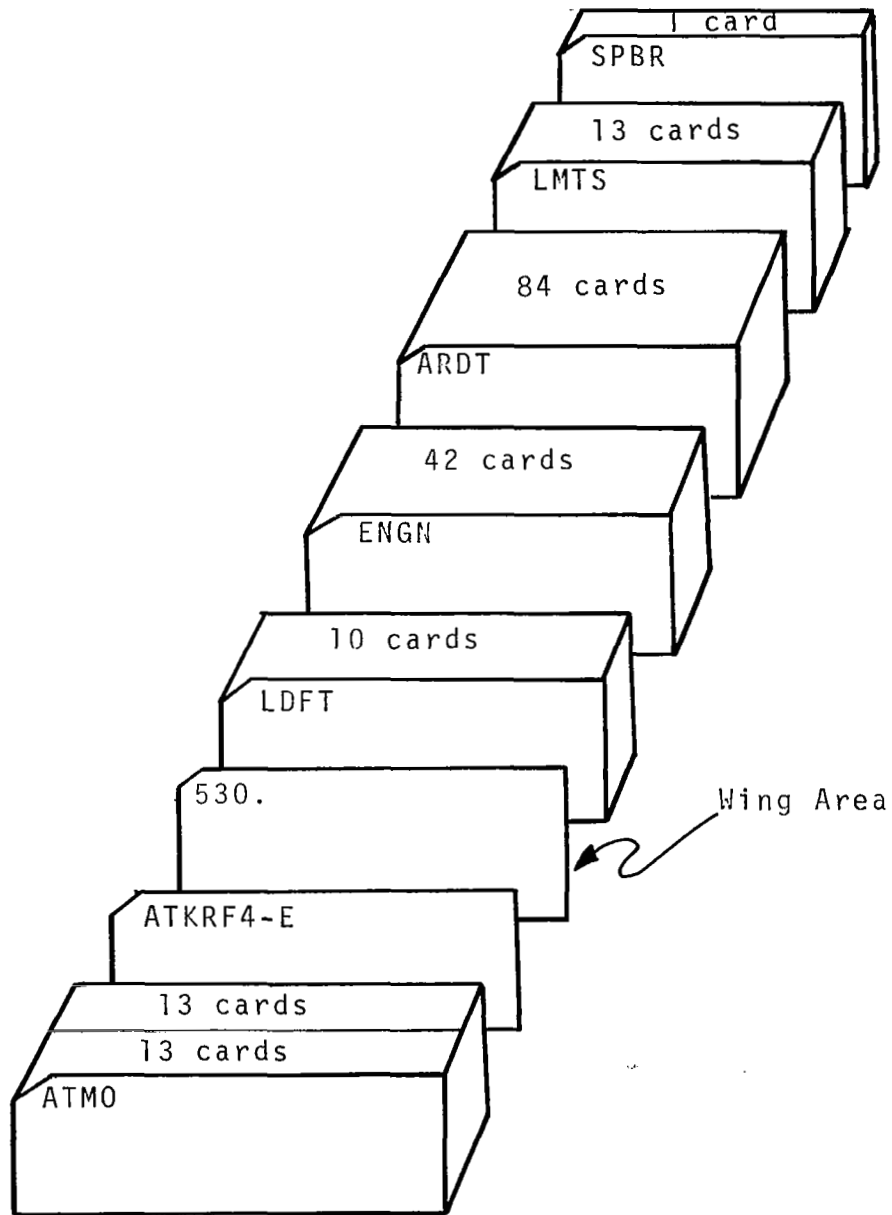
Figure 10.- Input data deck setup.
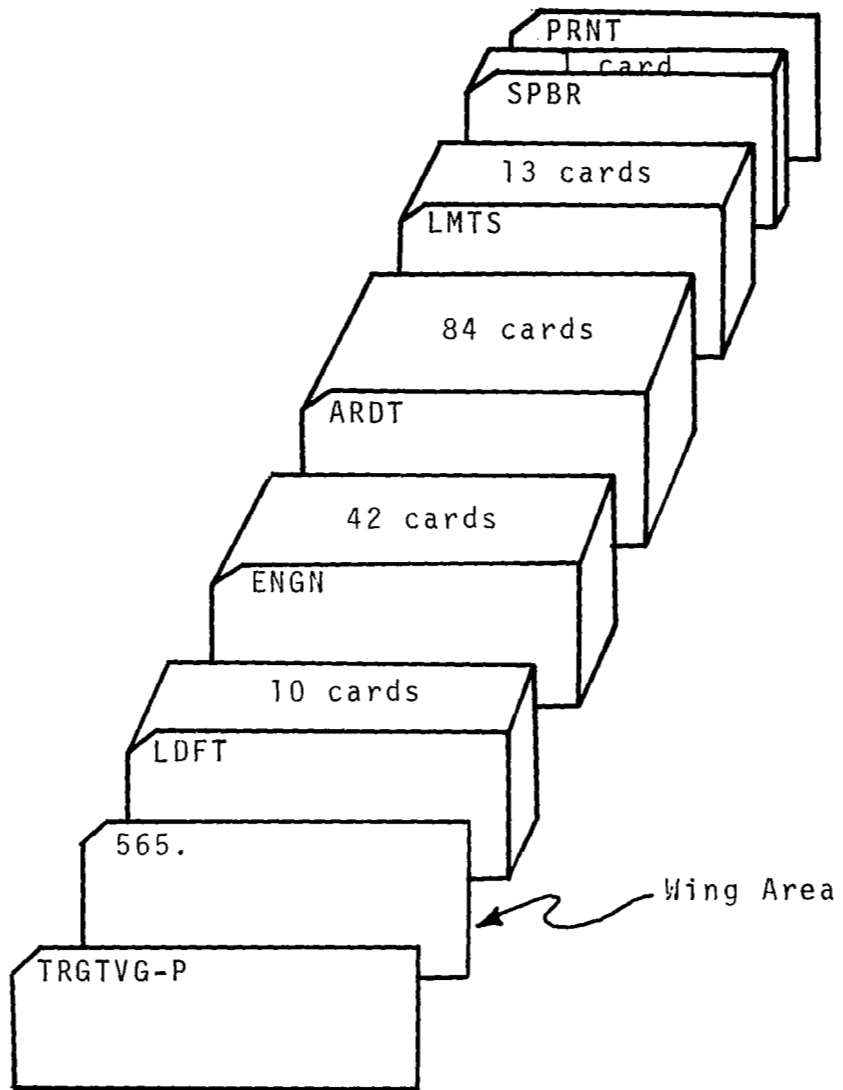
Figure 10.- Continued
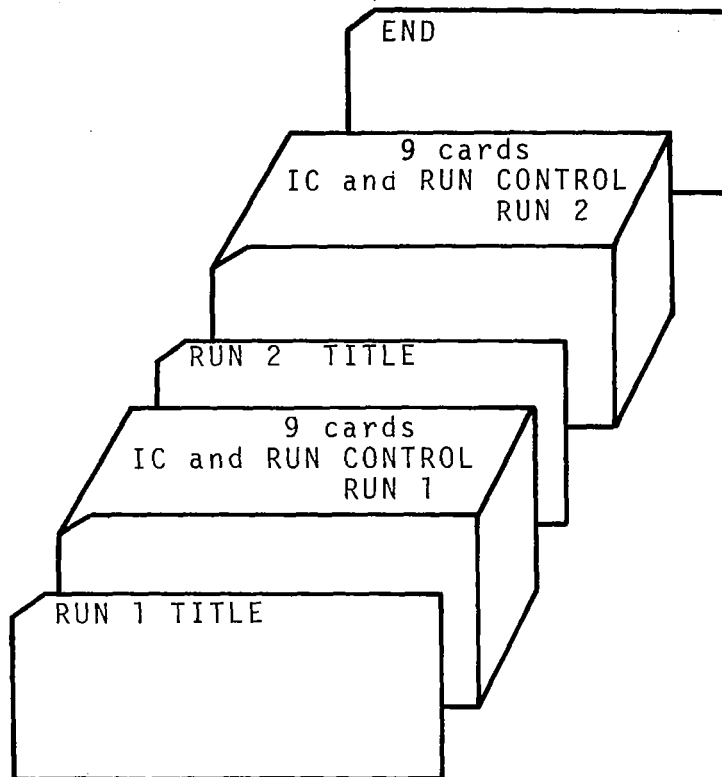
Figure 10.- Concluded

AIRCRAFT DATA AT TIME = 58.0000 SECONDS

		ATTACKER	TARGET
AIRCRAFT ID.		F-4E	F-4E
MACH		0.80	0.63
TRUE AIR SPEED	KNOTS	496.45	385.81
INDICATED AIR SPEED	KNOTS	374.29	282.08
ANGLE OF ATTACK	DEG	14.95	18.67
SIDESLIP ANGLE	DEG	-0.60	0.00
LOAD FACTOR	G	3.08	3.81
THETA (PITCH)	DEG	-52.10	-50.65
PHI (ROLL)	DEG	111.84	-63.01
PSI (YAW)	DEG	128.39	106.64
XE	FEET	-4838.9	-4453.7
YE	FEET	4919.7	3817.3
ALTITUDE	FEET	18096.4	19922.9
XEDOT	FEET/SEC	-190.9	-269.0
YEDOT	FEET/SEC	564.1	251.6
ALT DOT	FEET/SEC	-589.4	-537.0
VELOCITY	FEET/SEC	837.9	651.2
P	DEG/SEC	90.00	0.58
Q	DEG/SEC	25.00	10.98
R	DEG/SEC	15.00	-1.51
U	FEET/SEC	823.0	616.9
V	FEET/SEC	-8.8	0.0
W	FEET/SEC	157.1	208.4
MAX PERMISS. LDFCT.	G	5.46	4.00
SUSTAINED LOAD FCT.	G	3.64	2.49
THROTTLE POSITION	N/A	2.00	0.00
THRUST	POUNDS	23474.6	-617.8
LIFT	POUNDS	222375.1	155575.0
DRAG	POUNDS	60246.8	47119.5
C SUB L	N/A	0.50	1.09
SPEC. ENERGY/1000	FEET	29.01	26.51
SPEC. ENERGY RATE	FEET/SEC	-753.85	-760.54

Figure 11.- Printout of aircraft data.

TACTICAL SITUATION AT TIME = 58.0000 SECONDS

		ATTACKER F-4E	TARGET F-4E
AIRCRAFT ID.			
RANGE	FEET		2167.9
RANGE RATE	FEET/SEC		189.2
LINE OF SIGHT ANGLE (LOS)	DEG	169.25	6.93
AZIMUTH OF LOS (IN BODY AXES)	DEG	-169.93	-5.41
ELEVATION OF LOS (IN BODY AXES)	DEG	3.78	-4.34
DEVIATION ANGLE	DEG	167.29	15.27
DEVIATION ANGLE RATE	DEG/SEC	14.28	-5.00
ANGLE OFF	DEG	173.07	10.75
ACCUMULATED OFFENSIVE TIME	SEC	0.0000	4.1875
ACCUMULATED TIME FOR WEAPON 1	SEC	1.0625	22.1250
ACCUMULATED TIME FOR WEAPON 2	SEC	0.0000	0.0000
ACCUMULATED TIME FOR WEAPON 3	SEC	0.0000	0.0000

01 IS OPPONENT IN FRONT OF ME	0	1
02 AM I BEHIND OPPONENT	0	1
03 CAN I SEE OPPONENT	0	1
04 CAN OPPONENT NOT SEE ME	0	1
05 CAN I FIRE AT OPPONENT	0	1
06 CAN OPPONENT NOT FIRE AT ME	1	1
07 IS OPPONENT WITHIN CERTAIN CONE OF ME	0	1
08 AM I OUTSIDE CERTAIN CONE OF OPPONENT	1	1
09 IS CLOSURE RATE FAVORABLE	0	1
10 CAN I STAY IN WEAPONS DEL. ENVELOPE	0	1
11 NOT DEFENSIVE OR OPPONENT CAN'T GET	1	1
12 NOT DEFENSIVE OR OPPONENT CAN'T STAY	1	1
13 LINE OF SIGHT ANGLE WITHIN LIMITS	0	1
14 RATE OF LOS WITHIN LIMITS	0	1
15 WILL I HAVE AN ENERGY ADVANTAGE	0	0
CELL NUMBER	3232	16383
CELL VALUE	4	14

Figure 12.- Printout of tactical situation.

TRYNXT RETURNS   KROTST, NTRYT, ROTST =   6   6   83.274

I	ISTRYT	ICTRYT	G S	ROTRYT	AN1TRT	AN2TRT	AN3TRT	P SUB S
1	0	10	2.6036	104.765	-0.193	0.197	0.961	277.765
2	0	10	6.0284	100.000	-0.117	0.167	0.979	-882.046
3	0	11	6.0284	110.000	-0.274	0.229	0.934	-882.046
4	1	0	0.9941	0.000	0.919	-0.395	0.000	395.258
5	0	8	6.0284	80.000	0.202	0.030	0.979	-882.046
6	0	8	2.7306	80.000	0.202	0.030	0.979	262.843

TARGET TACTICS PREDICTS IN   3.00 SECONDS FROM CURRENT TIME   4.0000 SECONDS

COMAND	RANGE FEET	RATE FPS	CELL STATE																STATE NO	STATE VALUE
0 10	15754	-593	1	1	1	0	0	1	1	1	0	0	1	1	1	1	1	31975	11
0 10	14159	-582	1	1	1	0	0	1	1	1	0	0	1	1	1	0	0	7399	9
0 11	14183	-488	1	1	1	0	0	1	1	1	0	0	1	1	1	1	0	15591	10
1 0	14190	-459	1	1	1	0	0	1	1	1	0	0	1	1	1	0	1	23783	10
0 8	15568	-632	1	1	1	0	0	1	1	1	0	0	1	1	1	0	0	7399	9
0 8	13677	-575	1	1	1	0	0	1	1	1	0	0	1	1	1	1	1	31975	11

TARGET SELECTS TRIAL MANEUVER NUMBER   6

Figure 13.- Printout of target's tactical
decision process.

```
90003 ┌─────────────┐ (1)
      │   RESET     │
      │   ENTRY     │
      └─────────────┘
             │
             ▽
┌─────────────────────────────────┐ (2)
│                                 │
│          MODEAML = 2            │
│   GET INITIAL CONDITIONS FROM   │
│       SIMULATION CONSOLE        │
│   SET ψ₀(2), θ₀(2), φ₀(2)       │
│       P₀(2), Q₀(2), R₀(2)       │
└─────────────────────────────────┘
             │
             ▽
      ┌─────────────┐ (3)
      │ CALL  IC(1) │
      │ CALL  IC(2) │
      │ CALL QIC(1) │
      │ CALL AIC(2) │
      └─────────────┘
             │
             ▽
  ┌─────────────────────────┐ (4)
  │ ┌─────────────────────┐ │
  │ │DERINT(1,21) = XEDOTT│ │
  │ │DERINT(1,22) = YEDOTT│ │
  │ │DERINT(1,23) = ZEDOTT│ │
  │ └─────────────────────┘ │
  └─────────────────────────┘
             │
             ▽
90006 ┌─────────────┐ (5)
      │   RECYCLE   │
      │   ENTRY     │
      └─────────────┘
             │
             ▽
90002 ┌─────────────┐ (6)
      │    HOLD     │
      │   ENTRY     │
      └─────────────┘
             │
             ▽
            ╱ ╲
           ╱ 1 ╲
           ‾‾‾‾‾
```
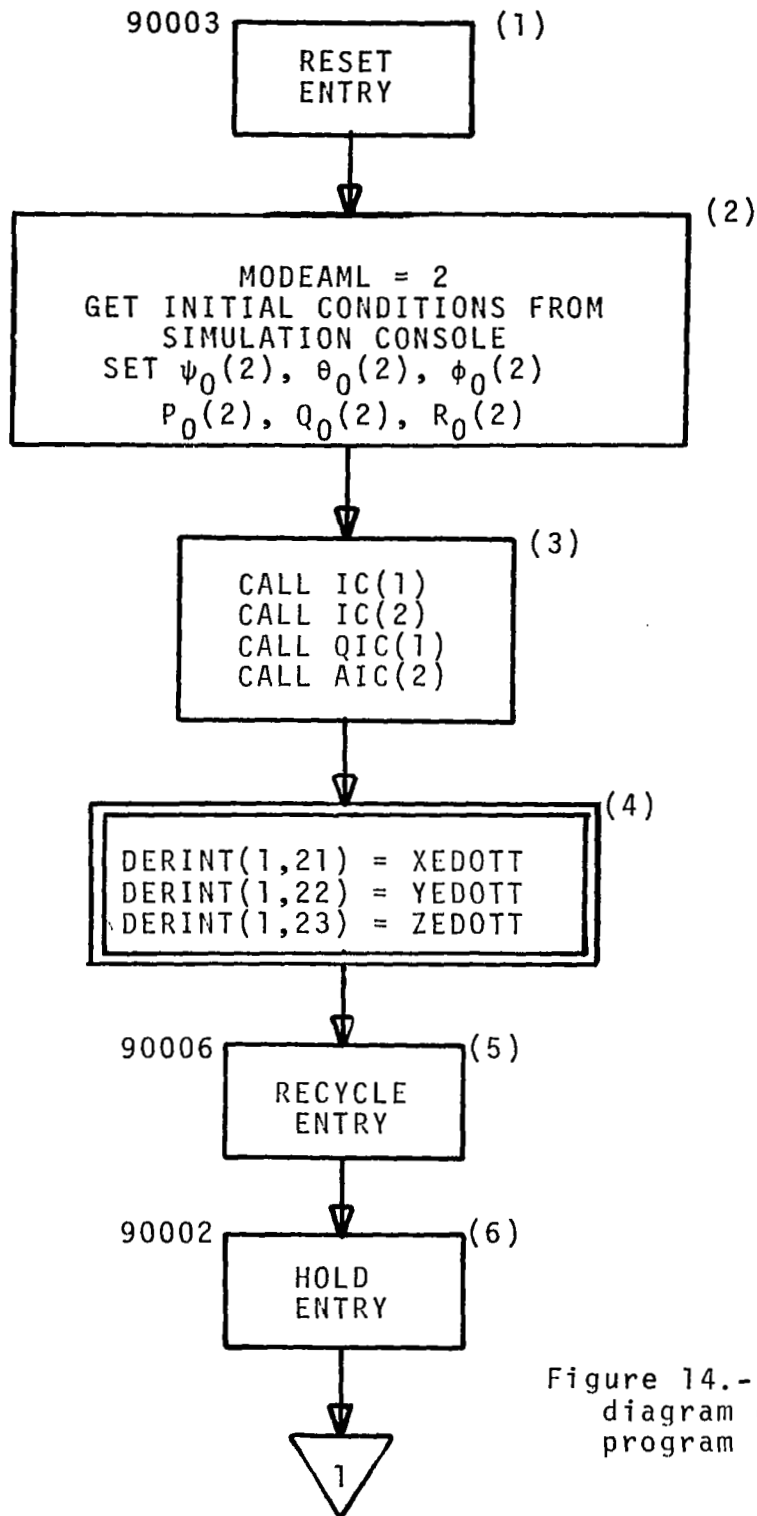
Figure 14.- Block
diagram of DMS
program RTMAIN

The blocks contain:

(1) 90003 RESET ENTRY

(2) MODEAML = 2
GET INITIAL CONDITIONS FROM SIMULATION CONSOLE
SET $\psi_0(2)$, $\theta_0(2)$, $\phi_0(2)$
$P_0(2)$, $Q_0(2)$, $R_0(2)$

(3) CALL IC(1)
CALL IC(2)
CALL QIC(1)
CALL AIC(2)

(4) DERINT(1,21) = XEDOTT
DERINT(1,22) = YEDOTT
DERINT(1,23) = ZEDOTT

(5) 90006 RECYCLE ENTRY

(6) 90002 HOLD ENTRY

99

```
                          ▽
                          1
                          │
                          ▽
        ┌──────────────────────────┐ (7)
        │ CALL DIRCOS(1)           │
        │ CALL VELOC(1)            │
        │ CALL MACHRHO             │
        │ CALL DYPRES              │
        │ CALL DIRCOS(2)           │
        │ CALL VELOC(2)            │
        └──────────────────────────┘
                          │
                          ▽
     ┌──────────────────────────────┐ (8)
     │ ┌──────────────────────────┐ │
     │ │  CALL AMLVS2 (MODEAML)   │ │
     │ └──────────────────────────┘ │
     └──────────────────────────────┘
                          │
                          ▽
  ┌─────────────────────────────────────────────────┐ (9)
  │ CALL FORCE      CALL TRANSEQ      CALL ROTEQ     │
  │ CALL MOMENT     CALL BLKOUT                      │
  └─────────────────────────────────────────────────┘
                          │
                          ▽
        ┌──────────────────────────┐ (10)
        │        REAL TIME          │
        │   OUTPUT AND DISPLAY      │
        └──────────────────────────┘
                          │
                          ▽
     ╱────────────────────────────────╲ (11)
    ╱  GO TO (DEPENDING ON             ╲
    │     CONSOLE SWITCHES)            │
    ╲ RESET, HOLD, OPERATE,            ╱
     ╲       TERMINATE               ╱
      ╲────────────────────────────╱
                          │
                          ▽
        ┌──────────────────────────┐ (12)
        │         OPERATE           │
        │          ENTRY            │
        └──────────────────────────┘
                          │
                          ▽
        ┌──────────────────────────┐ (13)
        │  MODEAML = 3              │
        └──────────────────────────┘
                          │
                          ▽
                          2
```

Figure 14.- Continued

Figure 14.- Concluded

```
        XET  = SX(2)
        YET  = SY(2)
        ZET  = -HAB(2)
        PHI  = PHI(2)
        THET = THETA(2)
        PSI  = PSI(2)
```

```
        XEA    = SX(1)
        YEA    = SY(1)
        ZEA    = -HAB(1)
        XEDOTA = SXDOT(1)
        YEDOTA = SYDOT(1)
        ZEDOTA = -HDOT(1)
        C_{ij}  = CD_{ij}(1)
```

AML PROGRAM PERFORMS
CALCULATION OF EQUATIONS
OF MOTION FOR A/C (2) AND
TACTICS

```
U(2) = UT, V(2) = VT, W(2) = WT
P(2) = PT, Q(2) = QT, R(2) = RT
```

```
DERINT(2,21) = ACCXET
DERINT(2,22) = ACCYET
DERINT(2,23) = ACCZET
```

REAL TIME INTEGRATION

```
XEDOTT = DERINT(1,21)
YEDOTT = DERINT(1,22)
ZEDOTT = DERINT(1,23)
```
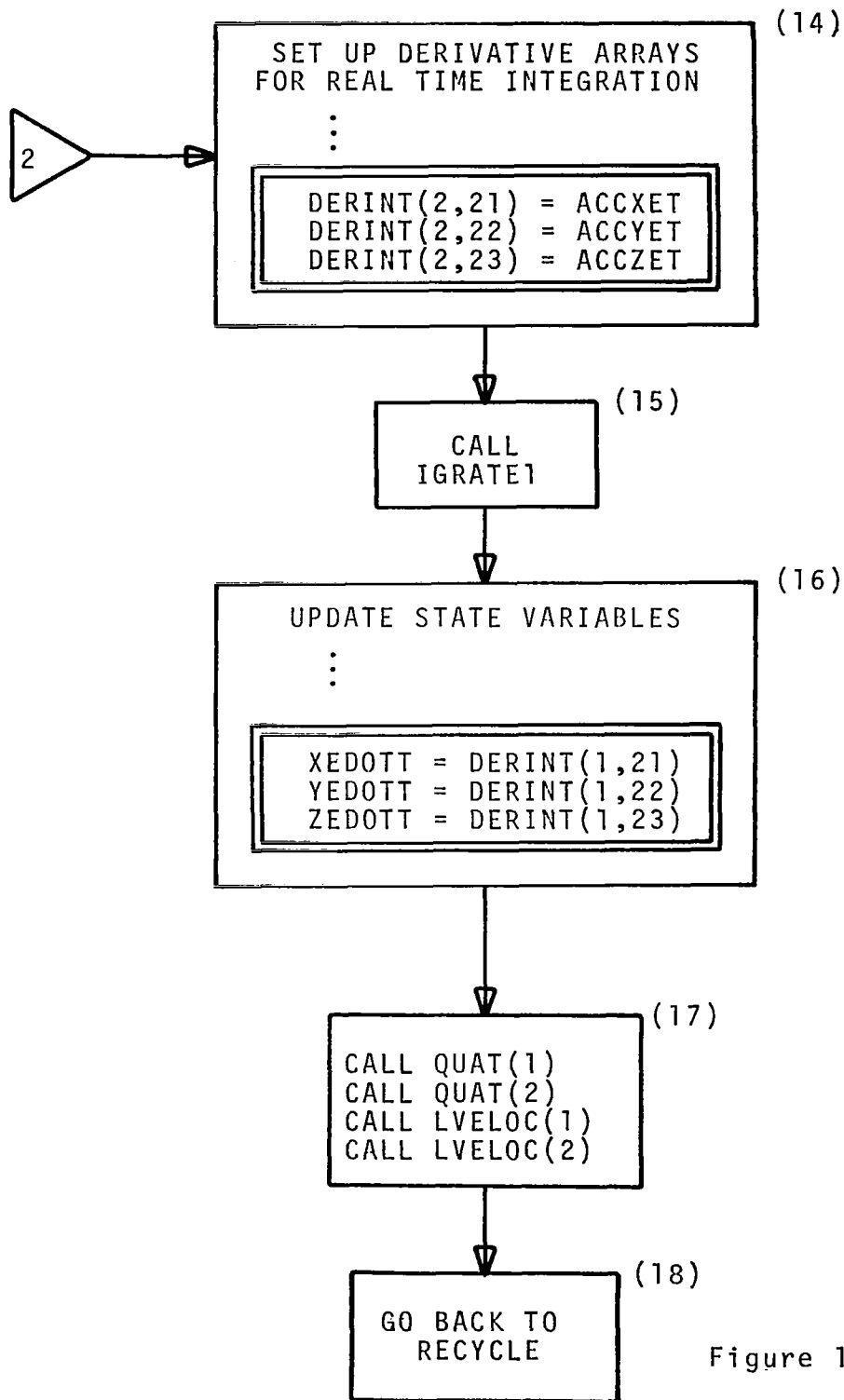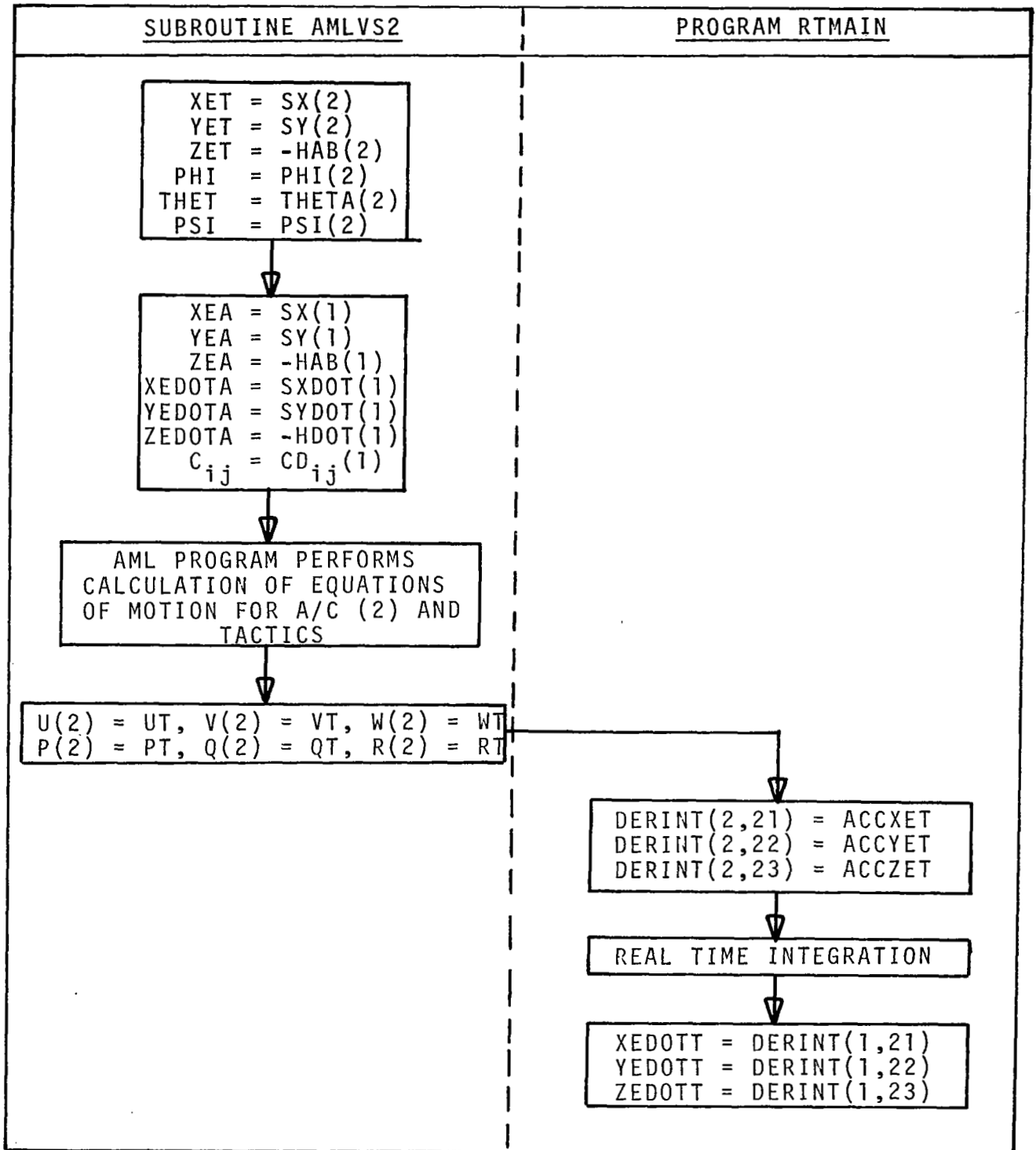
Figure 15.- Exchange of variables between
DMS and AML program during
operate cycle