AD 654936

AIR-TO-AIR COMBAT MODEL
PROGRAM & APPENDICES
TECHNICAL DETAILS

November 1967

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.

DEPUTY FOR DEVELOPMENT PLANNING
AERONAUTICAL SYSTEMS DIVISION
WRIGHT-PATTERSON AFB, OHIO

210

CAYWOOD-SCHILLER, ASSOCIATES

401 North Michigan Avenue

Chicago, Illinois

60611

ATAC-2: Single Search and Double Search

Terrell J. Harris
Barton J. Jacobs
Walter J. Strauss

November 1967

Contract Number AF33(615)-3707
Project Number 5101-57

prepared for

Deputy for ~~Systems~~ Engineering ~~Group~~ (ASD)

Air Force Systems Command

United States Air Force

Wright-Patterson Air Force Base, Ohio 45433

NOTE

This edition of the report on ATAC-2 is organized somewhat differently from the original edition. Volume I (Air-to-Air Combat Model, Description and Development, General Information) of the current edition consists of Volumes I and II of CSA Report Number 67-101 and of CSA Report Number 67-102. Volume II (Air-to-Air Combat Model, Program and Appendices, Technical Details) of the current edition consists of Volumes III and IV of CSA Report Number 67-101.

The Table of Contents and page numbers of the original edition are preserved here.

PREFACE

This report is published in four volumes. Volume I, Model Description, presents an overall view of the model and its two major submodels, the ENGAGEMENT Model and DATA PROCESSING Model. Volume II, Model Development, contains the rationale for the development and discussion of details, together with the derivations of all equations. Flow charts and program listings appear in Volume III, Program. Volume IV, Appendices, contains discussions of certain model concepts in detail.

The entire report is UNCLASSIFIED.

This report supersedes the original ATAC-2 document [Ref. 1]. The many changes and modifications made in the evolutionary development of the model, based on the analysis of many computer runs, have rendered the earlier version outdated. The program of the model as reported here was used for production runs in June, 1967.

Certain modifications which allow either aircraft to detect initially are reported separately in the document "Fighter Vs. Fighter Combat: ATAC-2 Model: Double Search," [Ref. 2].

ABSTRACT

ATAC-2 is a simulation model designed to help evaluate fighters in air-to-air combat. The model treats the one vs. one dogfight which arises from a random search situation. Both aircraft in the combat are (usually) aggressive. The two principal outputs from the model are the probability a given aircraft is killed in the fight and the expected number of enemy aircraft an aircraft kills over its useful life. Combat is restricted to a fixed altitude. The maneuvers are dynamic in that each aircraft responds to the situation at each moment in a duel depending on the information it has about an opponent's activities.

Inputs include, for each aircraft, search and tracking radar characteristics, passive radar sensors, optical capability, IFF, energy-maneuverability data, weapon loadings, weapon characteristics, and weapon kill probabilities.

The rationale for the model specifics are presented. Flow charts and program listings are included. The model has been run repeatedly on an IBM 7094.

ACKNOWLEDGMENT

It is a pleasure to acknowledge the technical contribution of the personnel of the Directorate of Operations Research with whom and for whom this model was developed. In particular, it is noted that many helpful comments, suggestions, and criticisms were made by Lawrence Boyd, Duane Dunlap, Walter Grady, Thomas Maney, Lt. Kenneth Pierscalla, and Ralph Schwab.

It is also a pleasure to acknowledge the many suggestions and ideas of our colleagues at CSA. In particular, it is worthy to single out Thomas Caywood, James Cerone, James Hall, Thomas Morton, James Rust, Donald Schiller, and Ronald Swanson.

The model was programmed by Jeannette Filsinger and Fay Young of the Deputy for ~~Systems~~ Engineering ~~Group~~, ASD, at Wright-Patterson Air Force Base. Their patience and response to the frequent changes and modifications made in the flow charts are appreciated.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

VOLUME IV - APPENDICES

LIST OF FIGURES

VOLUME I

VOLUME II

LIST OF FIGURES (CONT'D)

VOLUME IV

LIST OF FIGURES (CONT'D)

SECTION 7

DEFINITIONS

7.1 Introduction

Section 7 is intended primarily for reference. Section 7.2 defines all symbols used in the flow charts and the text, although the symbols are also defined where introduced in the text. The units associated with a variable are included, both for clarification and for use in setting input values.

An "I" after a definition means that this variable is an input to the model. A "C" means the variable is internally computed. This can be useful when reading the flow charts, as some variables which abstractly seem to be "inputs" are actually computed from other values. Variables used only in the text for model discussion have a 1) after their definition. This list is repeated in Volume III, Section 7.

Also included in this list is the FORTRAN symbol associated with each variable where appropriate. These symbols do not include the subscripts of subscripted FORTRAN symbols nor do they include the arguments of FORTRAN functions.

Section 7.3 discusses some input restrictions.

7.2 Definition List

Symbol	Definition	Fortran
A	An index with values of 0 and 1 indicating respectively that the bomber was unaware or aware during an engagement. Also used in text for the area within which the bomber flies during the search. (C)	IA
a	The area swept out by the fighter's detection pattern during its search.[1] (ft^2)	
a_i	The acceleration of aircraft i . (ft/sec^2) (C)	A
ACTIVE	A logical variable that takes values of YES and NO (or, equivalently, TRUE and FALSE) indicating whether or not an aircraft has active information from optical or detection radar. (C)	ACT
$a_{DEC}(i)$	The input deceleration of aircraft i . $a_{DEC}(i)$ must be input as a negative number. (ft/sec^2) (I)	ADEC
$A_M(i)$	The input parameter of the decreasing lag course function of aircraft i . This is the angle that aircraft i will try to lag by when its enemy is flying pure pursuit and $\lambda_i = 0$. (deg) (I)	AM
B	An index identifying the aircraft designated as "bomber," B always equals 2. (I)	IBMR
B_i	An index with values 0, 1 indicating the pilot'a sickness state. (C)	BINDEX
C	A flow chart symbol used to indicate the general maneuver of Circle.	
CL	A flow chart symbol denoting e general Circle Lost maneuver; i.e., lost information.	
D	The distance traveled by the bomber during the fighter's search. (ft) (I)	D
DIV	A temporary computation used in the Data Processing Model. (C)	DIV

1) Used in text only.

Symbol	Definition	Fortran
D(k, i)	An index indicating the general maneuver to be performed by aircraft i when in information state k ; 0 = evasive maneuver, 1 = aggressive maneuver. (I)	D
e	The base of the natural logarithm = 2.71828	EXP
EKB, EKT	The expected number of bombers killed by a fighter over its useful life. (C)	EKB
EKF	The expected number of fighters killed by a bomber over its useful life. (C)	EKF
ESB	A temporary calculation of the expected number of sorties completed by the fighter. (C)	ESB
E_C	A flow chart symbol used to indicate the general maneuver of Evade by Circle.	
E_L	A flow chart symbol used to indicate the general maneuver of Evade Linearly.	
E_S	Specific energy.1) (ft)	
$E_S^{(n)}$	The expected number of sorties completed in at most n attempts.1)	
ENV SW(MIS, i)	A switch which when ON, or TRUE, indicates that aircraft i has fired a weapon of type MIS; otherwise the weapon type has not been fired and the variable has a value of OFF, or FALSE. (C)	ENV SW
F	The segment Y* times the ratio of the bomber's velocity to the relative velocity; F is used in the computation of $P_D(\varepsilon)$. (ft) (C)	FSMALL
F	An index identifying the aircraft designated "fighter." F always equals 1. (C)	IFTR
f_1 , f_2	Oxygen flow leaving and returning to pilot's brain. Functional notation in text is made explicit in flow chart.1) (sec^{-1})	
g_1 , g_2	An arbitrary number of g's pulled.1)	
G_1 , G_2 , G_3	The three levels of target's total g's for which the weapon envelopes are input. (I)	GT
$G_B(i)$	An index with values of 0, 1 indicating respectively that the degree of oxygen debt of the pilot of aircraft i will not or will affect the maneuverability of his aircraft by limiting the g's of his aircraft. (I)	GB

Symbol	Definition	Fortran
G_i	The number of total g's being sustained by aircraft i . (C)	G
$G_i(V)$	The structural or aerodynamic limit of total g's for aircraft i as a function of its velocity. (I)	GBIG
$g_i(V)$	The total g function of velocity for aircraft i at which the specific power function is zero. Note if $g_i(V) > G_i(V)$ then for that V $g_i(V)$ is unattainable and $G_i(V)$ should be inputted. (I)	GMAXT
$g_{MIS}(i)$	The level of total g's for aircraft i above which weapon type MIS cannot be fired from aircraft i . (I)	GMIS
$G_p(i)$	The maximum number of total g's that the pilot of aircraft i is able to sustain. (I)	GP
h	The altitude of the simulated engagements; used as an identifier only. (ft) (I)	H
i, j	Indices that take on values of F and B and do not have the same value. These symbols always indicate an aircraft and nothing else. (C)	I, J
$IA(\ell)$	The aircraft that fired the ℓ^{th} weapon in an engagement. (C)	IAFIRI
ICAN(i)	An index with values 1, 0 indicating respectively that aircraft i's firing of any weapon is or is not being delayed in an engagement so as to get to a better position at the time of firing. (C)	ICAN
ID	Identifying titles of the combatants for printing purposes. (I)	ID
ISHIFT	An index with values 1, 2 and 3 indicating that the first, second or third value of $t_D(\cdot)$ is assigned to Δt . (C)	ISHIFT
ITEMP(i)	An index with values 1, 0 indicating respectively that aircraft i does or does not have IFF. (C)	ITEMP
k	An index with values 1, 2, ..., 11 indicating the information state of an aircraft. (Also used throughout as an arbitrary index with integer values; when used as such it is defined in context.) (C)	K

Symbol	Definition	Fortran
k_i	The k state of aircraft i . (C)	KPRT
K_F , K_B , K	Symbols used to describe the slope of the DEL Pursuit Course function.[1]	
k(MIS, i)	A counter of the number of weapons of type MIS fired from aircraft i . (C)	KOUNTR
L	One greater than the total number of weapons that may be fired by both aircraft; $L = 1 + \Sigma_i \; \Sigma_{MIS} \; N(MIS, i)$	LCAP
L	A flow chart symbol used to indicate a Linear course.	
ℓ	An index giving the order in which weapons were fired in an engagement. $\ell \leq L - 1$. (C)	LLITL
m	An index with values 1, 2 indicating whether firing by one or both aircraft is permitted. Used in P_j^m , PK_j^m , etc. (C)	M1
MI(ℓ)	The MIS identification number of the ℓ^{th} weapon fired in an engagement. (C)	MISTP1
MIS	The identification number assigned to a weapon type, MIS takes values of 1, 2, ..., $n_m(i)$. (C)	MIS
m_i	An index identifying the position, velocity and information (k) state of aircraft i . (C)	IMSTAT
N	The number of grid-points or points of initialization for each ϵ . (I)	N
n	An index with values 1, 2, ..., N indicating the grid-point number under consideration. Also used in text for other purposes but always so identified. (C)	IGRIDP
NUIND	An index with values of 0, 1, 2 indicating the mode of operation of the tracking radar of each aircraft. (I)	NUIND

Symbol	Definition	Fortran
$n_m(i)$	The number of distinct weapon types carried by aircraft i . (I)	NM
$N_u(\epsilon)$	The number of grid-points in which the bomber was unaware for a given ϵ . (C)	NU
N_ϵ	The total number of ϵ values used in an encounter. (C)	NEPS
n'	An index used to compute the variables $\bar{x}$ and $\bar{y}$. (C)	NT
n^*	The maximum number of sorties by each aircraft, for the calculation of EKB and EKF . (I)	YN
N(MIS, i)	The total number of weapons of type MIS on aircraft i . (C)	NUMIS
O_i	A variable indicating the amount of "oxygen debt" of the pilot of aircraft i incurred by pulling g's over a period of time. (C)	OXDEBT
P	A flow chart symbol used to indicate the general maneuver of Pursuit.	
P	Represents "pursuer" in text (often used as subscript).[1]	
PASSIVE	A logical variable that takes values of YES and NO or TRUE and FALSE indicating whether or not an aircraft has passive information. (C)	PASS
PKB	The probability for an encounter that the bomber is detected and killed. (C)	PKB
PKBGD	The probability that the bomber is killed given detection for the encounter. (C)	PKBGD
PKE	The probability that the bomber is detected and killed at or before the time it becomes aware for an encounter. (C)	PKE
PKF	The probability for an encounter that the fighter detects the bomber and the fighter is killed. (C)	PKF
PKFGD	The probability that the fighter is killed, given detection for the encounter. (C)	PKFGD

Symbol	Definition	Fortran
PKG	The probability that the bomber is killed after it becomes aware, given that it survived to the time of its awareness for an encounter. (C)	PKG
PKL	The probability that the bomber is killed after it becomes aware and it survived to tha time of its awareness for an encounter. (C)	PKL
P_D	The probability of detection of the bomber. (C)	PDD
P_S	Specific power.[1)] (ft/sec). Also probability of survival of an aircraft.[1)]	
P_{SB}	The probability for an encounter that the bomber survives. (C)	PSB
P_{SF}	Tha probability for an encounter that the fighter survives and detects the bomber and the bomber is aware of the fighter. (C)	PS
P_u	The probability that the bomber is unaware for an encounter. (C)	PUU
$P(\ell)$	The probability that the target is dead just after the ℓ^{th} weapon hits its target in some engagement. (C)	P
$p(\ell)$	The probability an aircraft is killed by the ℓ^{th} weapon only.[1)]	
PS(i)	An index indicating the capability of the passive receiver of aircraft i ; 0 implies no capability, 1 implies the ability to detect the presence of another aircraft but not the position, and 2 implies the capability of 1 with the ability to distinguish the hemisphere of the source. (I)	IPS
$P_1(j)$	An index that when set to zero will require aircraft j to turn its tracking radar on for one time pulse only when launching a weapon. (I)	P1
$P_2(i)$	An index with values 1 and 0 indicating respectively that aircraft i has or has not activated its tracking radar. (C)	P2
PV(x)	A function that gives the principal value of its angular argument. (rad) (C)	PV
P(x, y)	Probability distribution of bomber's (x, y) coordinates during search.[1)]	

Symbol	Definition	Fortran
$P_{U/D}$	The probability that the bomber is unaware, given that it is detected for an encounter. (C)	PUID
$P_D(\varepsilon)$	The probability of detection of the bomber by the fighter for some ε. (C)	PD
$P_U(\varepsilon)$	The probability that the bomber is unaware of the fighter for some value of ε. (C)	PU
$P_i(V, \dot{\beta})$	The specific power function of aircraft i at velocity V and turning rate $\dot{\beta}$. (ft/sec) (I)	PEEU
PK_B^*	The probability that the bomber is killed, given that the bomber is detected and unaware for an encounter. (C)	PK2B
PK_j^1	The encounter probability that aircraft j is killed, given that the bomber is detected and aware and that aircraft j does not fire. (C)	PKK
PK_j^2	The encounter probability that aircraft j is killed, given that the bomber is detected and aware. (C)	PKK
$P_B^*(\varepsilon)$	The probability for a given ε that the bomber is killed, given that it is detected and unaware.	PZ
$P_j^1(\varepsilon)$	The probability for a given ε that aircraft j is killed, given that the bomber is detected, aware and that aircraft j does not fire. (C)	PK
$P_j^2(\varepsilon)$	The probability for a given ε that aircraft j is killed, given that the bomber is detected and aware. (C)	PK
$P_C(\varepsilon)$	The probability that the bomber is killed at or before the time it becomes aware for some ε. (C)	PCCEPS
$P_B^*(\varepsilon, n)$	The probability for a given ε and grid-point n that the bomber is killed, given that it is detected and unaware. (C)	PPZ
$P_j^1(\varepsilon, n)$	The probability for a given ε and grid-point n that aircraft j is killed, given that the bomber is detected and aware and that j does not fire. (C)	PP
$P_j^2(\varepsilon, n)$	The probability for a given ε and grid-point n that aircraft j is killed, given the bomber is detected and aware. (C)	PP

Symbol	Definition	Fortran
$P_C(\epsilon, n)$	The probability that in an engagement the bomber is killed at or before the time it becomes aware. (C)	PCCEPN
$P'_k(MIS, i)$	The probability of kill of weapon type MIS on aircraft i . (I)	PKP
$P_k(MIS, i)$	The probability of kill of weapon type MIS on aircraft i ; set to the input value, $P'_k(MIS, i)$, or to zero. (C)	PK
Q_B	The quadrant of the point B on the fighter's detection pattern. (C)	QB
Q_C	The quadrant of the point C on the fighter's detection pattern. (C)	QC
$q(\ell)$	The probability both aircraft are alive at time $T(\ell)$.[1]	
$Q(X)$	A function that gives the quadrant of the angle X . (C)	Q
R	The range between the two aircraft. (ft) (C)	R
r	The range of the detection capability of the fighter; r is set to $R_{DET}(F)$ if this is not zero and to $R_{OPT}(F)$ otherwise. Also used in text for the Y* projection against a stationary target. (ft) (C)	RSMALL
R_1	A variable used to indicate whether the tracking radar is turned on. (ft) (C)	R1
R_1 , R_2	Distances used in describing steady state.[1] (ft)	
r_{in}	An override initial range that will act so as to shrink the detection range to r_{in} . (ft) (I)	RANGE
$R'(1, i)$, $R'(2, i)$	The first and second values respectively that will be assigned to $R^*(i)$, i.e., before and after the opponent becomes aware. (ft) (I)	RPRIME
RFLOOR(MIS, i)	A superimposed minimum boundary of weapon type MIS such that the weapon type may not be fired from aircraft i whenever the range is less than RFLOOR(MIS, i). (ft) (I)	RFLOOR

Symbol	Definition	Fortran
RNOW(i)	The range at which aircraft i may commence to fire at an unaware enemy; against an unaware target, firings by aircraft i are postponed until within a range of $R \leq RNOW(i)$. (ft) (C)	RNOW
RTEST(1) , RTEST(2)	The ranges at which Δt will change values from $t_D(1)$ to $t_D(2)$ and from $t_D(2)$ to $t_D(3)$ respectively. (ft) (I)	RTEST
$R_{PAS}(i)$	The range of the passive detection capability of aircraft i . (ft) (I)	RPAS
$R_{TRK}(i)$	The range of the tracking radar of aircraft i . (ft) (I)	RTRK
$R_{OPT}(i)$	The range of the optical capability of aircraft i . (ft) (I)	ROPT
$R_{DET}(i)$	The range of the detection radar of aircraft i . (ft) (I)	RDET
$R_{IFF}(i)$	The range of the IFF capability of aircraft i . (ft) (I)	RIFF
R_{min}	A minimum range which will terminate an engagement; $R < R_{min}$ causes termination of an engagement. (ft) (I)	RMIN
R'_{MIS}	The minimum range of some weapon type. (ft) (C)	RMISP
R_{MIS}	The maximum range of the weapon envelope of some weapon type. (ft) (C)	RMIS
$R_1(V_a, o_b, MIS, i)$	The outer weapon envelope of weapon type MIS on aircraft i associated with a velocity of V_a , an angle-off of o_b for a value of target g's of G_1 . (ft) (I)	RF1T
$R_2(V_a, o_b, MIS, i)$	The same as $R_1(V_a, o_b, MIS, i)$ but for a target g level of G_2 . (ft)(I)	RF2T
$R_3(V_a, o_b, MIS, i)$	The same as $R_1(V_a, o_b, MIS, i)$ but for a target g level of G_3 . (ft) (I)	RF3T
$R'_1(V_a, o_b, MIS, i)$	The inner envelope limit of weapon type MIS on aircraft i for an average velocity of V_a and angle-off of o_b for G(1) total target g's. (ft)(I)	RF1PT
$R'_2(V_a, o_b, MIS, i)$	The same as $R'_1(V_a, o_b, MIS, i)$ but for a target g level of G_2 . (ft) (I)	RF2PT
$R'_3(V_a, o_b, MIS, i)$	The same as $R'_1(V_a, o_b, MIS, i)$ but for a target g level of G_3 . (ft) (I)	RF3PT

Symbol	Definition	Fortran
$\dot{R}$	The rate of change of the range; (ft/sec) (C) $\dot{R} = dR/dt$.	RDOT
$R^*(i)$	The range which aircraft i will attempt to attain when in the rear of its enemy. (ft) (C)	RSTAR
$R_g(i)$	The range of the tail gun of aircraft i . (ft) (I)	RANGUN
$R(\phi_j)$	The maximum range at which aircraft j can fire any weapon when approaching an unaware target from the rear, based on the speeds of the aircraft and tracking angle of the pursuer. It assumes the target flies linearly. This is also used as the name of the routine that calculates $R(\phi_j)$. (ft) (C)	RPHIJ
S_i	The expected fraction of t_{MAX} that the pilot of aircraft i will spend in a "sick" condition, $0_i > 1$, for an encounter. Also used to represent the total amount of time that the pilot of aircraft i spends in a sick condition. (In latter case; sec)(C)	TIMSIC
$S_i(\varepsilon)$	The expected fraction of t_{MAX} that the pilot of aircraft i will spend in a "sick" condition for some value of ε . (C)	SICEPS
$S_i(\varepsilon, n)$	The fraction of t_{MAX} that the pilot of aircraft i spends in a "sick" condition during an engagement defined by ε and n . (C)	SICTIM
ST(i)	An index indicating the general maneuver of aircraft i ; (C) 0 ⟶ circle 1 ⟶ linear flight 2 ⟶ pursuit course 3 ⟶ circle, lost information 4 ⟶ evade 5 ⟶ evade, lost information	IST
S	An estimate of the amount of change in range when decelerating at a constant rate from some velocity to V_0 at a rate of $a_{DEC}(i)$. (ft) (C)	S
sgn(x)	The signature function of the argument x ; (C) $\mathrm{sgn}(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ -1, & \text{Otherwise.} \end{cases}$	SGN

Symbol	Definition	Fortran
$S_f(i)$	An index with values of 0, 1 indicating respectively that the degree of oxygen debt of the pilot of aircraft i will not or will effect the firing of weapons by retarding such firings. (1)	SF
S(MIS, i)	An index with values of 1 and 0 respectively indicating that aircraft i has or has not fired weapon type MIS. (C)	S
T	Represents "target" in text (often used as subscript).[1]	
t	The amount of time since initialization of an engagement. (sec) (C)	T
t_f	The time of flight of all weapons. (sec) (I)	TF
t'	An arbitrary time prior to T(ℓ).[1] Also used in text as a time prior to detection. (sec)	
t_{AWARE}	The time at which the bomber became aware of the presence of the fighter. (sec) (C)	TAWARE
t_{LAST}	The duration of time of an engagement. (sec) (C)	TLAST
t_{max}	The maximum amount of combat time allowed for a single engagement. (sec) (1)	TMAX
t_{min}	The amount of time required to elapse after which a loss of information by both combatants will terminate an engagement. (sec) (I)	TMIN
t_{PRT}	The amount of time until the next printout of each aircraft's relevant parameters. (sec) (C)	TPRT
$t_D(1)$, $t_D(2)$, $t_D(3)$	The first, second and third values assigned Δt at various transitions during an engagement. (sec) (I)	TDELTS
$t_C(i)$	The maximum amount of combat time allowed for aircraft i. (sec) (I)	TC
$t_{MIS}(i)$	The time at which aircraft i fired the first weapon of type MIS. (sec) (C)	TMIS
t(MIS, i)	The time of the last firing of a weapon of type MIS from aircraft i. (sec) (C)	TLASTF
t*	The amount of time between printouts of the combatants' relevant variables. (sec) (1)	TSTAR

Symbol	Definition	Fortran
$T(\ell)$	The time at which the ℓ^{th} firing took place during an engagement. (sec) (C)	TFIREI
V_a	Aircraft speed used in the input tables for launch envelopes and for energy-maneuverability. (ft/sec) (I)	VATAB VATAG
V_o	The speed for aircraft i which would make the range rate, $\dot{R}$, equal zero, subject to $V_o \geq V_C(i)$. (ft/sec) (C)	VZERO
V'	An arbitrary speed used in the distance function S.[1] (ft/sec)	
V^*	The speed of one aircraft relative to the other at the time of initialization. (ft/sec) (C)	VSTAR
V_i	The speed of aircraft i . (ft/sec) (C)	V
$V_o(i)$	The initial speed of aircraft i . (ft/sec) (I)	VZ
$V^*(i)$	The speed at which the sustainable turning rate of aircraft i is an absolute maximum. (ft/sec) (C)	VSTR
$V_C(i)$	The minimum sustainable speed of aircraft i . (ft/sec) (C)	VC
$V_{max}(i)$	The maximum sustainable speed of aircraft i . (ft/sec) (I)	VMAX
W_i	The weight of aircraft i . (lbs) (I)	W
X	One axis of the moving coordinate system used in the initiation phase.[1] (ft)	
XPHI	A multiplier with values 0 and 1 to change the computed angle ϕ^* to 0 or to leave it as is. (I)	XPHI
x_i	The x position of aircraft i in the (x, y) inertial coordinate system. (ft) (C)	X
x	A symbol used as the argument of various functions. Also used for one axis of the inertial coordinate system.[1] (ft in latter case)	
x	Used for temporary computations. (C)	CAPX

Symbol	Definition	Fortran
x_1	A temporary calculation of the maximum turning rate of an aircraft when pulling the minimum of $g_i(V_i)$ and $G_i(V_i)$ total g's. $(rad/sec)^2$)	
x_2	A temporary calculation of the maximum turning rate of an aircraft when pulling $G_i(V_i)$ total g's. $(rad/sec)^2$)	
$\bar{x}$	A temporary calculation of the probability that both aircraft are alive at the time of firing of tho ℓ^{th} weapon. (C)	X
x_i', y_i'	Temporary computations of the future position of aircraft i. (ft)2)	
Y	One axis of the moving coordinate system used in the initiation phase.1) (ft)	
y	One axis of the inertial coordinate system.1) (ft)	
Y_g	The relative Y coordinate of the bomber at the beginning of the engagement. (ft) (C)	YG
Y_{MAX}	The upper limit of the segment Y* in the (X, Y) coordinate system (used in initiating engagements). (ft) (C)	YMAX
Y_{MIN}	The lower limit of the segment Y* in the (X, Y) coordinate system (used in initiating engagements). (ft) (C)	YMIN
Y*	The normal projection of the fighter's detection pattern onto the Y axis in the (X, Y) coordinate system at the beginning of one engagement. (ft) (C)	YSTAR
y*	The steady state speed. (ft/sec) (C)	YNEW
$\bar{y}$, $y(\ell)$	A temporary calculation of the probability that aircraft j survived any weapons that hit between the firing and arrival of the ℓ^{th} weapon. (C)	Y
Y_B	A parameter used in determining the detection contour for some ε. (ft) (C)	B
Y_C	a parameter used in determining the detection contour for some ε. (ft) (C)	C

---

2) Used in flow chart only.

Symbol	Definition	Fortran
y_i	The y position of aircraft i in the (x, y) inertial coordinate system. (ft) (C)	Y
Z_K , Z_K' , Z_L , Z_L'	Temporary calculations used to compute the weapon launch envelopes. $(ft)^2$ 2)	
α_1 , α_2 , α_3 , α_4	Used to describe DEL course.1) (rad)	
$\dot{\alpha}_i$	The rate of change of α_i ; (rad/sec) (C) $\dot{\alpha}_i = d\alpha_i/dt$.	ALPDOT
$\alpha_g(i)$	The half-angle defining the tail gun capability of aircraft i measured from the tail of the aircraft. (deg) (I)	ALPGUN
$\alpha_{MAX}(i)$	The internally computed parameter of the DEL function of aircraft i . (rad) (C)	ALPMAX
$\alpha_{MIS}(i)$	The half-angle of the cone in which aircraft i must be tracking its enemy in order to fire weapon type MIS. (deg) (I)	ALPMIS
α_i	The tracking angle of aircraft i , measured from the inner line of sight to the velocity vector of the aircraft. (rad) (C)	ALPHA
$\alpha_{DET}(i)$	The half-angle of the detection radar cone of aircraft i . (deg) (I)	ALPDET
$\alpha_{IFF}(i)$	The half-angle of the IFF capability of aircraft i . (deg) (I)	ALPIFF
$\alpha_{OPT}(i)$	The half-angle of the optical capability of aircraft i . (deg) (I)	ALPOPT
$\alpha_{PAS}(i)$	The half-angle of the passive detection capability of aircraft i measured off the tail of the aircraft. (deg) (I)	ALPPAS
$\alpha_{TRK}(i)$	The half-angle of the tracking radar cone of aircraft i . (deg) (I)	ALPTRK
$\dot{\beta}_i$	The turning rate of aircraft i . (rad/sec) (C)	BETDOT
β_i	The angle of aircraft i's heading measured from the x axis. (rad) (C)	BETA
γ	An angle used in defining the initial positions of the aircraft. (rad) (C)	GAMMA

2) Used in flow chart only.

Symbol	Definition	Fortran
δ	The density of fighters assigned to an area for search purposes; used in computing the probability of detection, $P_D(\varepsilon)$. (ft^{-2}) (I)	DELTA
$\Delta\alpha$	The change in α in one time pulse.[1] (rad)	
Δt	The incremental time slice of simulation -- a time for which rate of change, $\dot{R}$, $\dot{\theta}$, etc., are assumed to be constant in the integration of the equations of motion. (sec) (C)	DELTAT
ΔY_g	The change in the value of Y_g for different grid-points. (ft) (C)	DELTYG
$\Delta\varepsilon$	The change between ε values for various sets of grid-points. If input to a value greater than 180° the program will execute only one value of ε . (deg) (I)	DELEPS
ε	The angle between the velocity vectors of the combatants measured from the fighter to the bomber at the time of initialization. Also used to indicate the index of a particular value of ε in summations. (In special cases this may be an input.) (deg) (C)	EPSLON
ε_1 , ε_2	The first and last values respectively of ε . (deg)(C)	EPSLON
η	The desired tracking angle of an aircraft as computed by the DEL course function. (rad) (C)	ETA
λ_i	The constant deviate angle flown by aircraft i when inside the ϕ^* cone of its enemy. (deg) (I)	LAMBDA
θ_B	The angle between the X-axis and the ray from (0, 0) to the point B of the fighter's detection pattern. (rad) (C)	THETB
θ_C	The angle between the X-axis and the ray from (0, 0) and the point C on the fighter's detection pattern. (rad) (C)	THETC
$\dot{\theta}$	The turning rate of the line of sight. (rad/sec) (C)	THEDOT
μ	The angle between the vector V^* and the heading of the fighter at initialization. (rad) (C)	EMU
π	3.14159 ... (C)	PI

Symbol	Definition	Fortran
ρ	The half-angle of the fighter's detection capability; ρ is set to $\alpha_{DET}(F)$ if this is not zero and to $\alpha_{OPT}(F)$ otherwise. (rad) (C)	RHO
σ_b	An angle indicating the angle-off times the sign of the direction of turn of the target. (rad) (I)	SIGB
σ(MIS, i)	A counter of the amount of time that aircraft i has spent in the envelope of weapon type MIS since the last firing of weapon type MIS. (sec) (C)	SIGMIS
τ_i	The approximate amount of time that aircraft i will need to fire a weapon after its tracking radar is activated; used to compute distance from $R(\phi_i)$ at which to activate its tracking radar. (sec) (I)	TAU
τ(MIS, i)	A counter of the total amount of time that aircraft i has spent in the envelope of weapon type MIS. (sec) (C)	TAUMIS
ϕ_i	The angle off the target of aircraft i ; measured from the outer line of sight to the heading of aircraft j , $j \neq i$. (rad) (C)	PHI
$\dot{\phi}_i$	The rate of change of ϕ_i ; (rad/sec) (C) $\dot{\phi}_i = d\phi_i/dt$.	PHIDOT
ϕ^*	The half-angle of the cone in the rear of a target aircraft within which pure pursuit is the navigation doctrine, if $\lambda_i = 0$. (rad) (C)	PHISTR

7.3 Input Considerations

While ATAC-2 was developed with a general user in mind, completely arbitrary inputs are, of course, impossible to handle. Limitations exist and considerations must be given to the values of inputs. These considerations are necessary, in some instances, due to the nature of air-to-air combat and in other instances due to the specifics of the program as it exists. Below are listed some of these considerations and limitations under various headings.

7.3.1 Parameter Guidance

(1) In the program the value of the internal variable R*(F) is initielly set to the input R'(1, F) . This is the range which the fighter tries to attain off the tail of an unaware bomber, before firing any weapons. If the fighter can maintain surprise R'(1, F) will be the range at which the bomber becomes aware by being fired upon. Two considerations, therefore, should be observed when selecting a value for R'(1, F) .

Firstly, this range should be such as to allow firings of the more lethal weapons of the fighter. Secondly, the range should be such as to allow the fighter to stay behind the bomber when the latter becomes aware and begins to maneuver. This last consideration is very difficult to quantify. In general, the ability of the fighter to stay behind a maneuvering bomber is a function of both velocities, both turning rates and both deceleration rates. However, the closer the fighter is to its steady state range and its associated velocity (see Appendix F), if it exists, the easier it will be to stay behind the bomber.

(2) The range R*(i) , input as R'(2, i) , represents the range at which aircraft i would like to be off the tail of its enemy so as to be able to fire its shortest range weapons, usually guns, while the enemy is maneuvering. The selection of this parameter is influenced by the capability of this shortest range weapon. However, the maneuverability of both aircraft must be accounted for. In Appendix F it is shown that selection of R*(i) (or a steady state range) has implications on other parameters as well, namely the velocity and angle-off associated with R*(i) . Thus, if possible these considerations (the values of R*(i) that allow weapon firings and the values of R*(i) that allow steady state conditions to obtain) should be combined to arrive at a realizable R*(i) whenever possible.

It should be noted that R*(i) originally takes on an input value R'(1, i) which in the case of the fighter, is the range at which it may begin firing. Once this range is achieved, it then closes to this close range R'(2, i) .

(3) The length of the time pulse Δt influences the running time of the computer program. The running time is inversely proportional to Δt . However, as Δt increases some singularities occur. For example, the path of an aircraft may cross a launch envelope from one pulse to the next, without being in the launch envelope at the beginning of a pulse. A possible firing will be missed. Also, since the method of numerical integration in the model assumes that the time derivatives of the relative parameters are constant for a period of time Δt , the length of Δt should be kept fairly small. The error due to this assumption is inversely proportional to R^2 . The assumption is therefore worse for

smaller values of range; hence the capability in the program to decrease the length of Δt as the range gets smaller. Typical values for Δt are:

$$t_D(1) = 5.0 \text{ sec.} \qquad R_{TEST}(1) = 100{,}000$$

$$t_D(2) = 2.0 \text{ sec.} \qquad R_{TEST}(2) = 70{,}000$$

$$t_D(3) = .25 \text{ sec.}$$

7.3.2 Model Logic

The following are logical restrictions of the model, i.e., the violation of them will affect the logic of the model:

1. All calculations involving $a_{DEC}(i)$ assume an associated negative value; hence $a_{DEC}(i)$ must be inputted as less than or equal to zero.

2. IFF is necessary before firings can take place; $\alpha_{IFF}(i)$ and $R_{IFF}(i)$ must, therefore, be non-zero if aircraft i is to fire its weapons.

3. Each aircraft may obtain active information only from either its detection radar or optical system. An extension of tracking radar coverage over that of the detection coverage adds no capability. To avoid confusion and false interpretation the tracking pattern should be contained within the detection pattern: $\alpha_{TRK}(i) \leq \alpha_{DET}(i)$ and $R_{TRK}(i) \leq R_{DET}(i)$.

4. The speeds for which the specific power function is inputted should be restricted to allow the aircraft at least to maintain altitude at the assumed power setting and altitude. Both g functions of speed $g_1(V)$ and $G_1(V)$ must be greater than or equal to 1 for all possible speeds. The specific power function must be greater than or equal to zero at 1-total g (zero turning rate) for all possible speeds.

7.3.3 Flow Chart Restrictions

The following restrictions are implied by the flow charts:

1. There are three (3) g values for which the weapon launch envelopes are input, namely G_1, G_2, G_3.

2. In the tables for $g_1(V)$, $G_1(V)$, and weapon envelopes the minimum and maximum speeds should extend at least to the possible speed values of $V_C(i)$ and $V_{max}(i)$, respectively. Otherwise when a routine uses a table to find a functional value for a speed outside the tabular range, the routine linearly extrapolates to find the required functional value.

3. The time, t^*, between printings of the positional parameters of the aircraft should be a multiple of the time pulse Δt. This avoids round-off errors and ensures a constant time between printings.

4. The initial velocity, $V_o(i)$, of aircraft i must, of course, be within the velocity region $V_C(i)$ to $V_{max}(i)$.

5. Whenever $g_i(V)$ is unattainable due to the value of $G_i(V)$ being prohibitive, any value greater than or equal to $G_i(V)$ may be used for $g_i(V)$ at that V. The program will select the minimum of the two when appropriate.

6. The tail gun of aircraft i is assumed to be weapon number $n_m(i)$, the last weapon. Further $n_m(i)$ may not be zero. The capability and hence the effect of the tail gun may be negated by setting $R_g(i)$ and $\alpha_g(i)$ to zero, or setting $N(n_m(i), i)$ to zero.

7.3.4 Program Restrictions

The following are restrictions that exist in the current program. They arise from computer space allocation considerations. To change them, however, may require significant programming effort.

1. The number of weapon types must be no more than six: $n_m(i) \leq 6$.

2. The number of grid-points for one ε must be limited by twenty: $N \leq 20$.

3. The number of speed values for which the weapon launch envelopes are input must be either 2 or 3.

4. The number of angular values for which the weapon launch envelopes are input must be between two and fifteen inclusive.

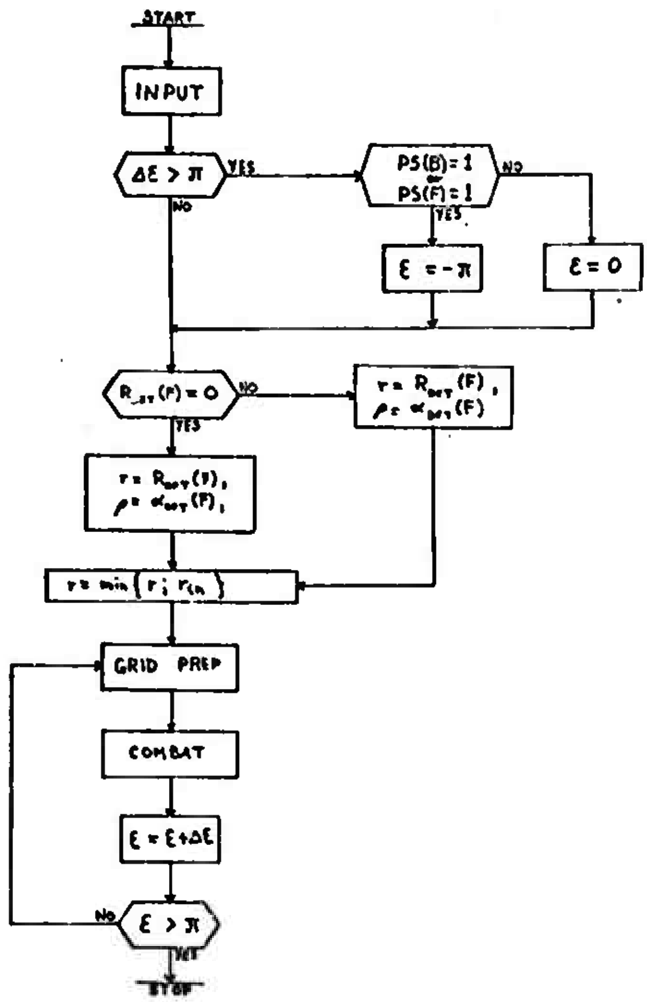
5. Since the number of ε values considered will be $180°/\Delta\varepsilon$ or $360°/\Delta\varepsilon$, $\Delta\varepsilon$ must be greater than or equal to 30° if PS (F or B) = 1 and greater than or equal to 15°, otherwise.

SECTION 8

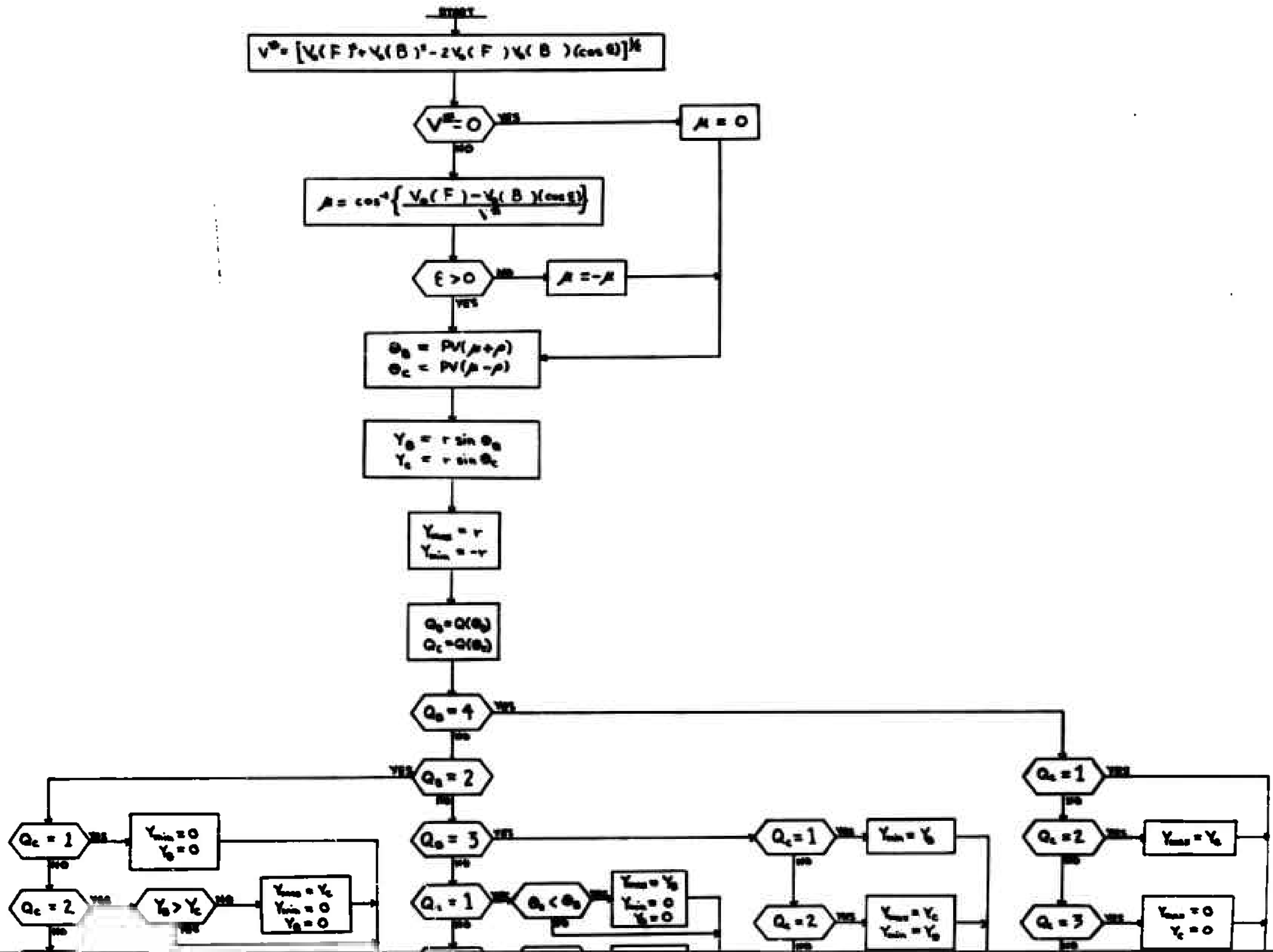
FLOW CHARTS

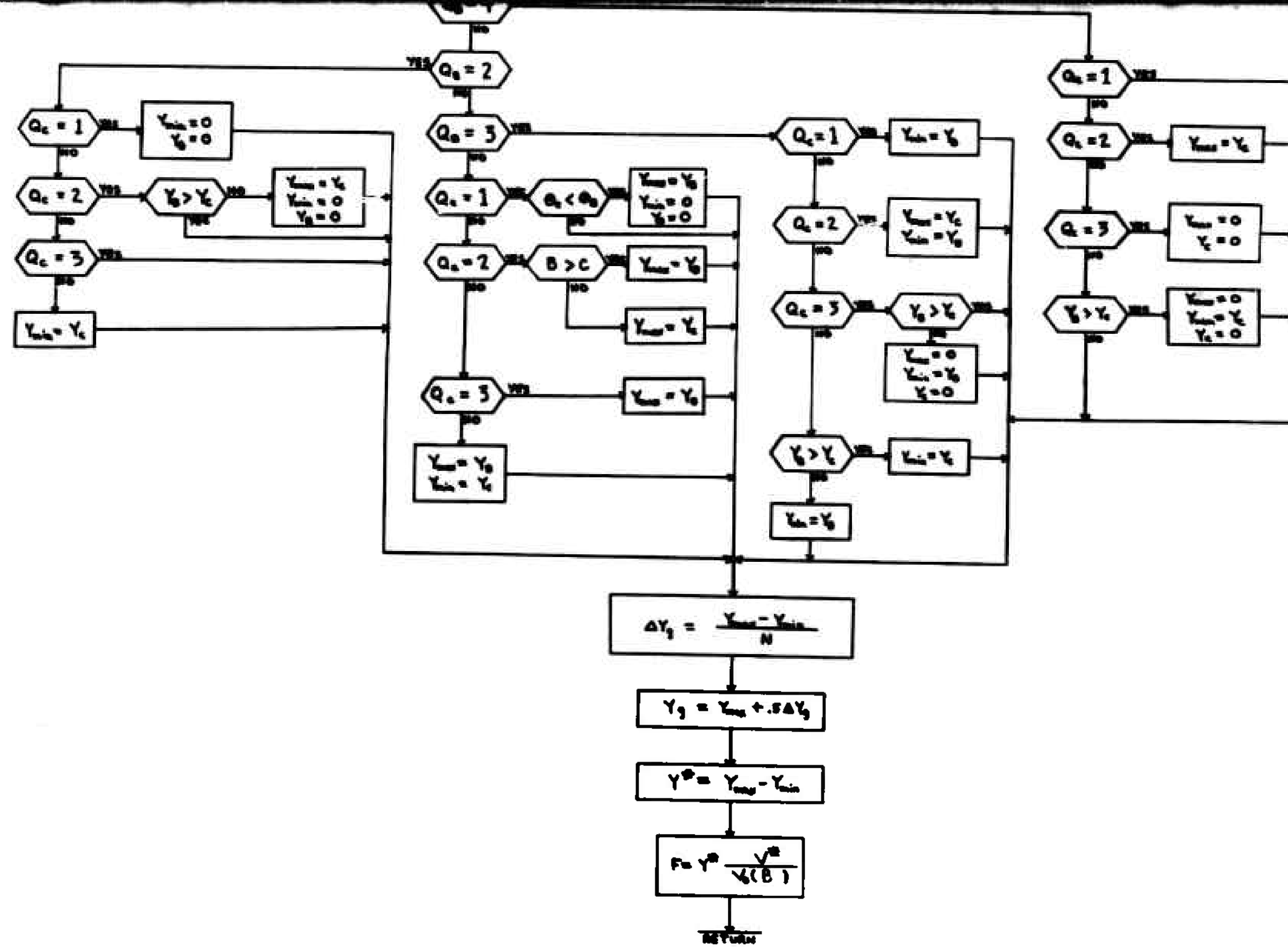
In this section the flow charts of the various ATAC-2 routines are presented. They are intended for use with the relevant discussion of Volume II. The EM flow charts are discussed in Section 5, Volume II. The discussion of the routines of the DATA PROCESSING Model is given in Section 6.5, Volume II.

THE EXECUTIVE ROUTINE-EM

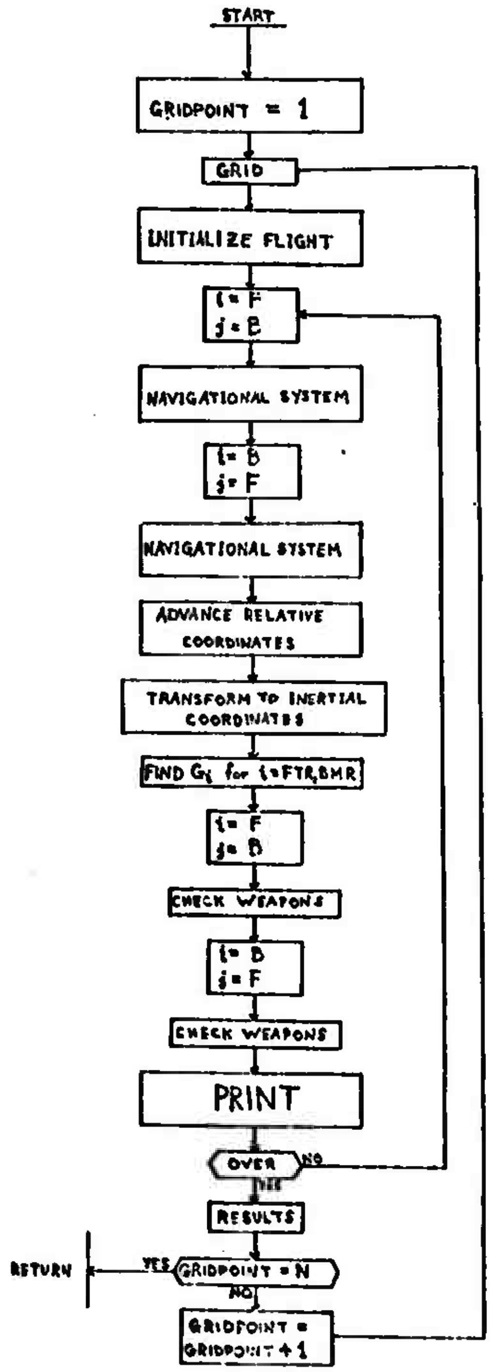


THE GRID PREPARATION ROUTINE

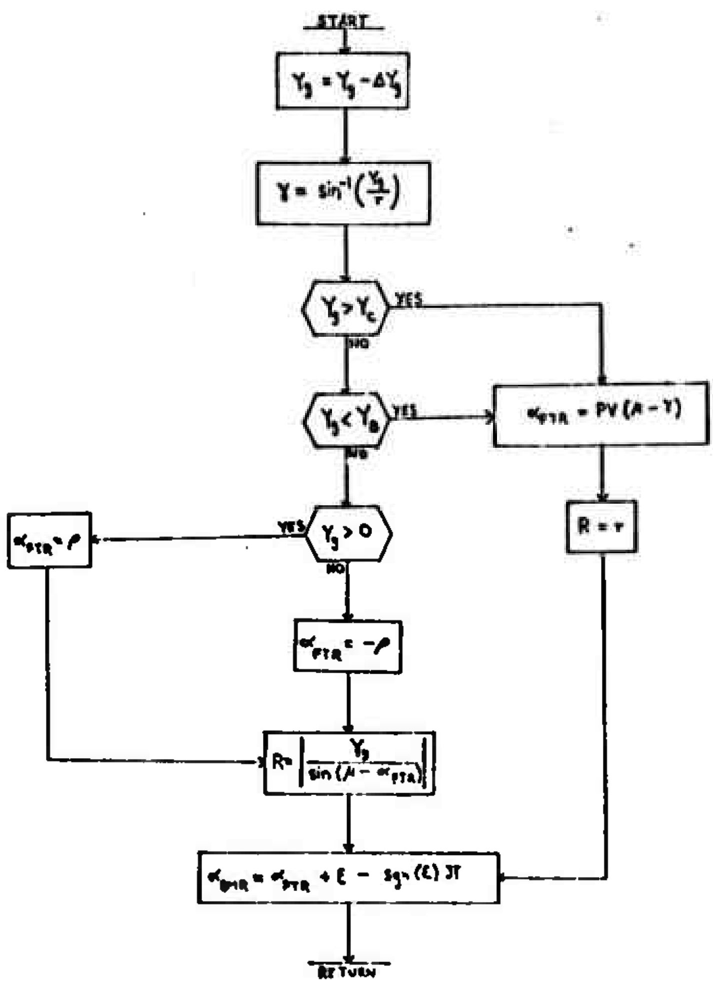



YES
NO
$\Delta Y_g = \frac{Y_{max} - Y_{min}}{N}$
$Y_g = Y_{max} + .5\Delta Y_g$
$Y^* = Y_{max} - Y_{min}$
RETURN

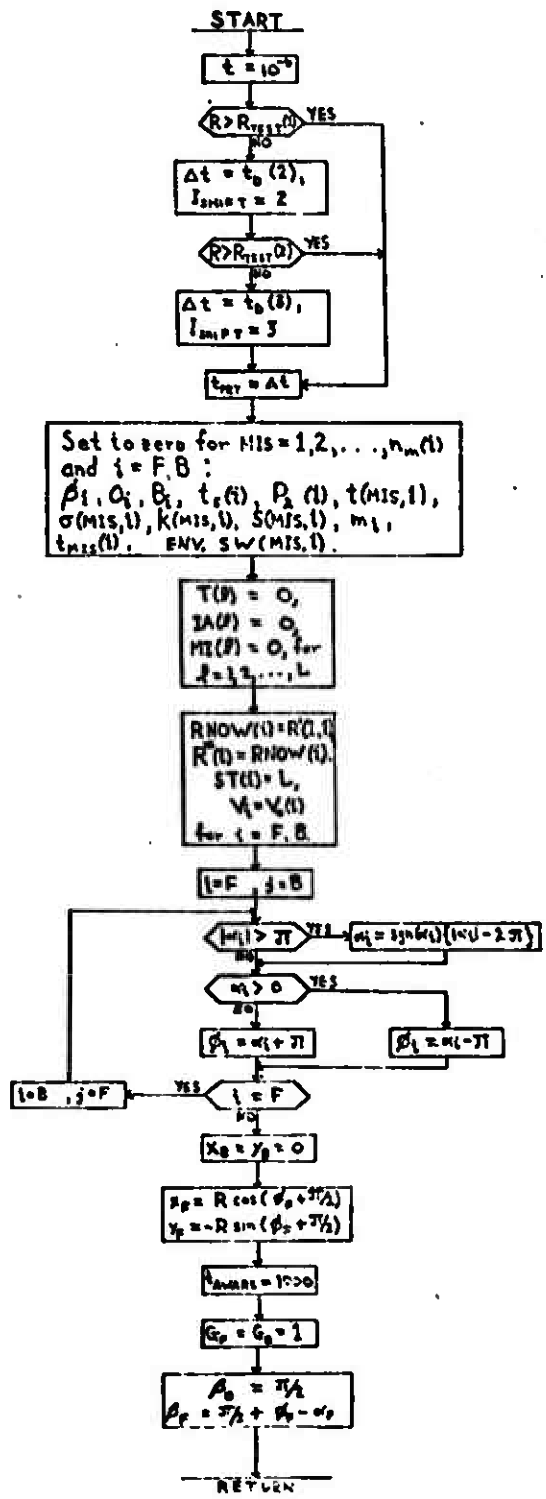
THE COMBAT ROUTINE



THE GRID ROUTINE

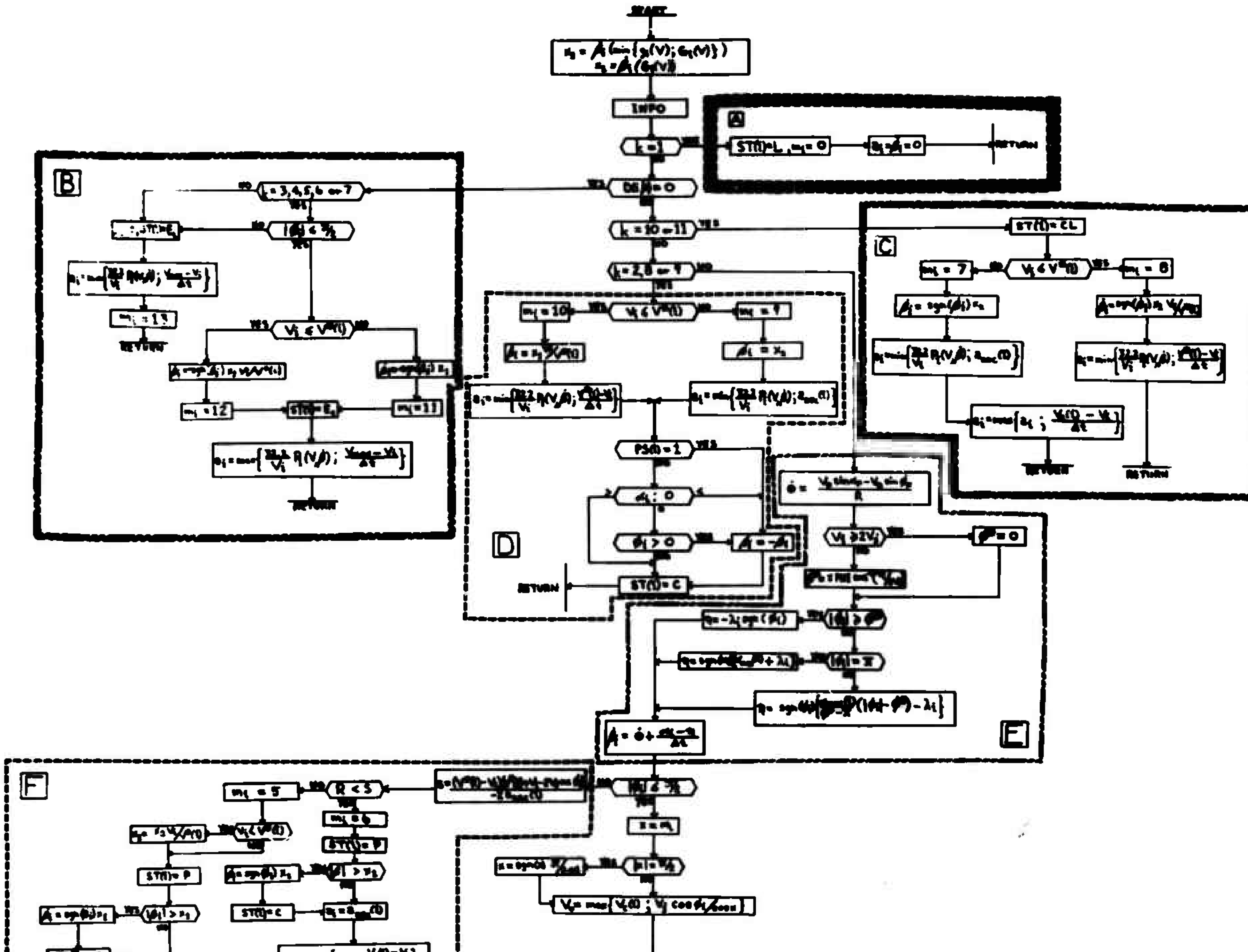


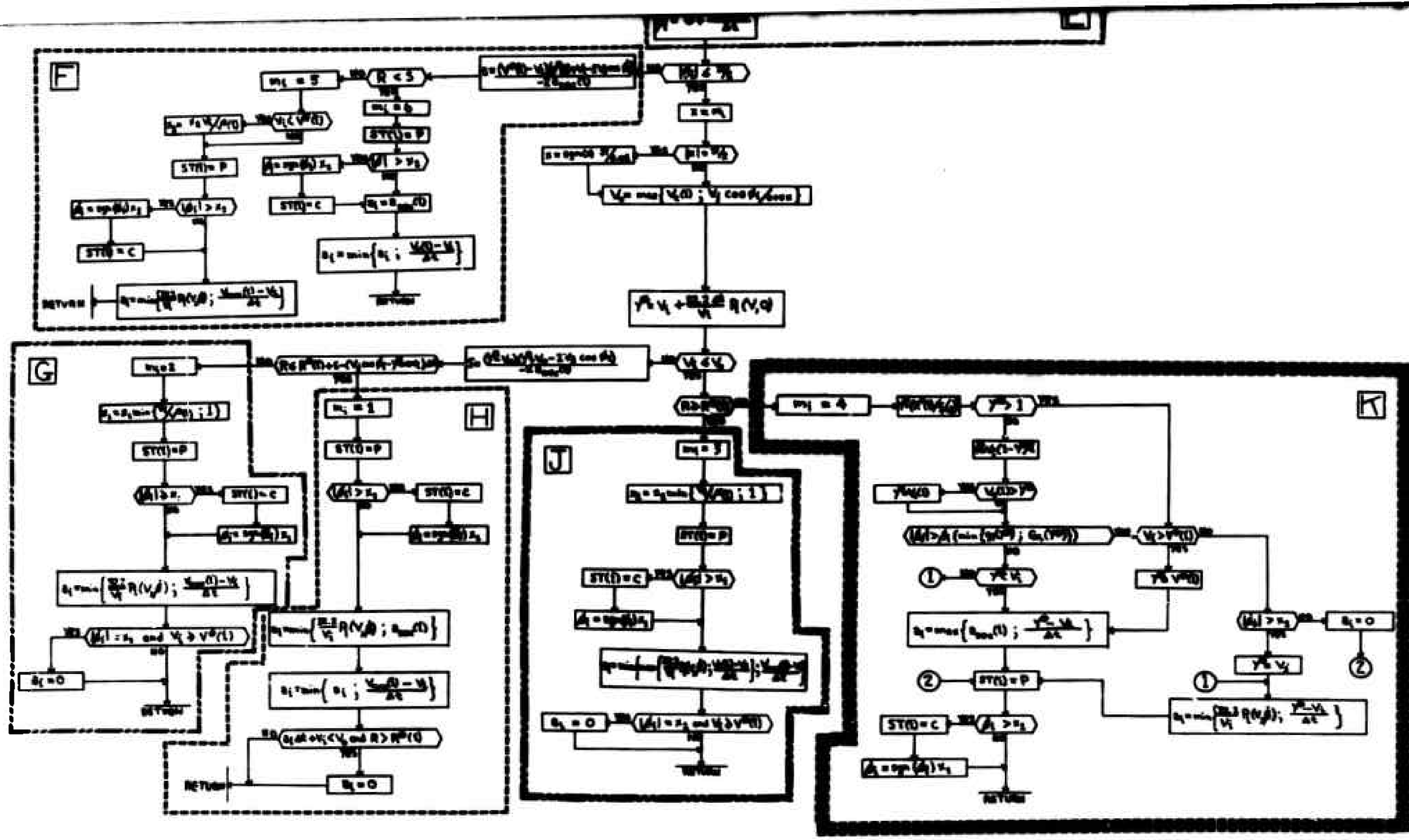
THE INITIALIZE FLIGHT ROUTINE



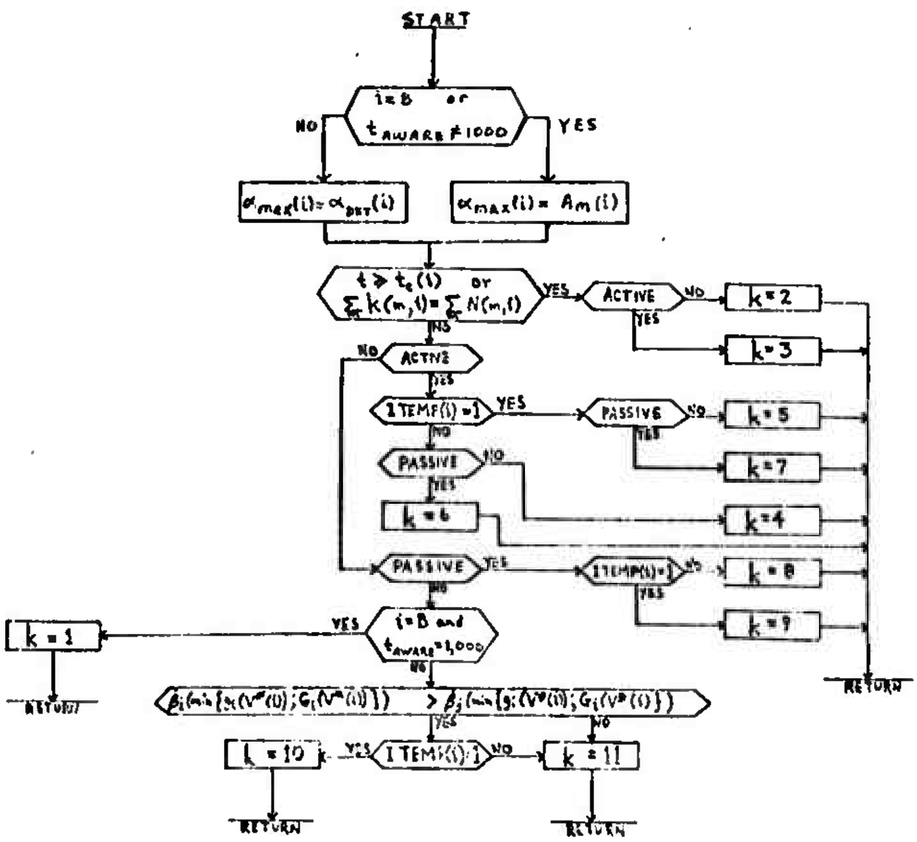
NAVIGATIONAL SYSTEMS

A

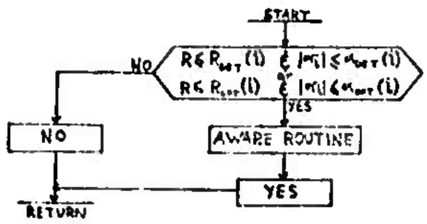



F
G
H
J
K
$m_i = 5$
$R < 5$
$m_i = 6$
$m_i = 2$
$m_i = 1$
$m_i = 3$
$m_i = 4$
STI(i) = P
STI(i) = C
$a_i = 0$
YES
NO
RETURN

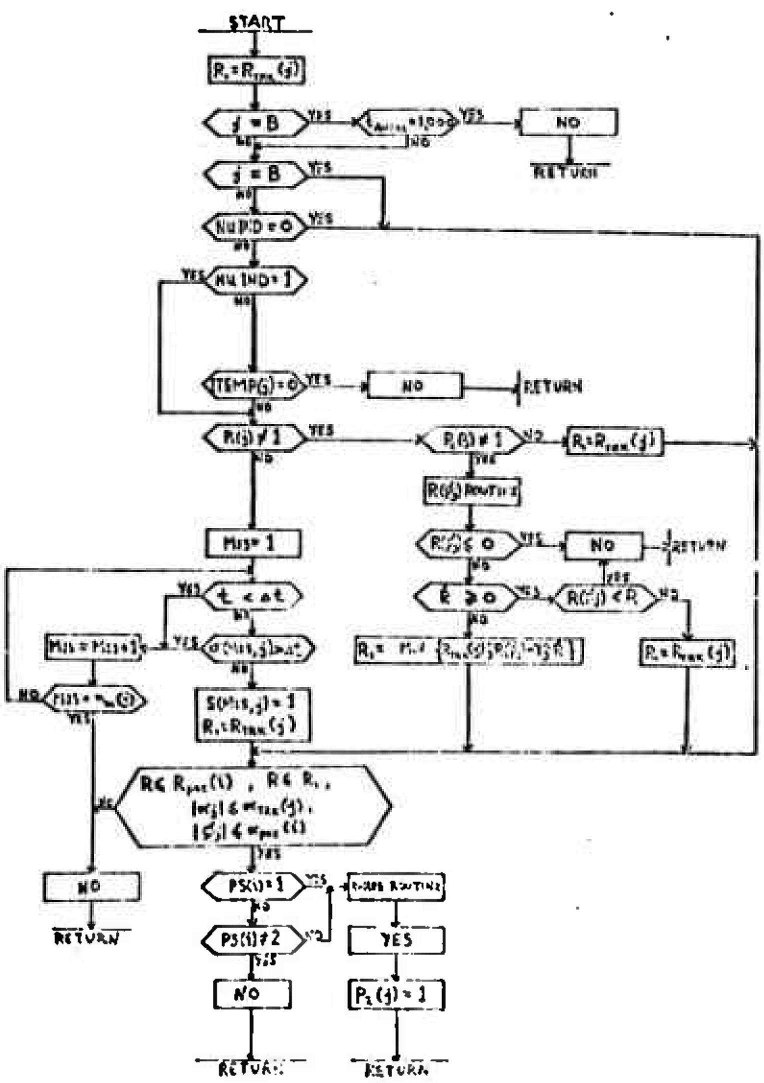
THE INFO ROUTINE



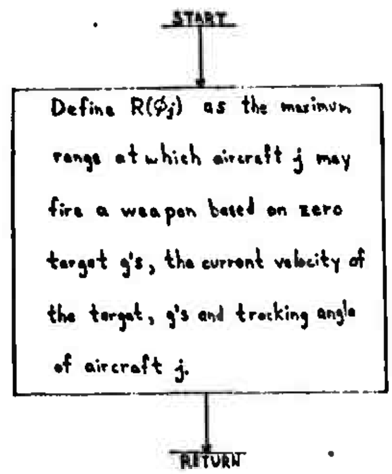
THE ACTIVE ROUTINE



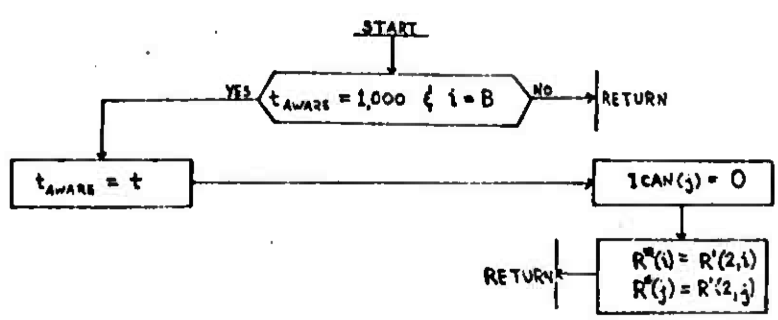
THE PASSIVE ROUTINE



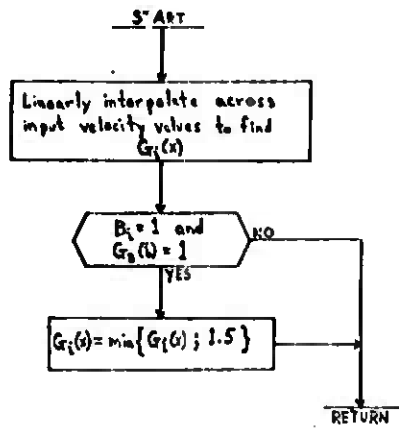
THE R(ϕ_j) FUNCTION



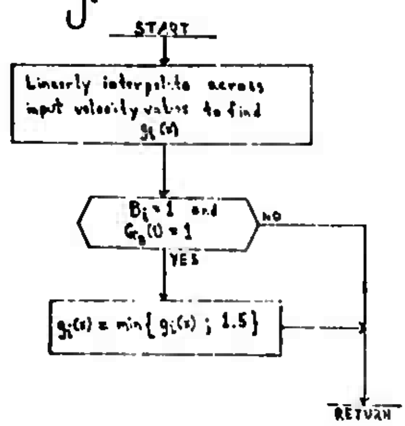
THE AWARE ROUTINE



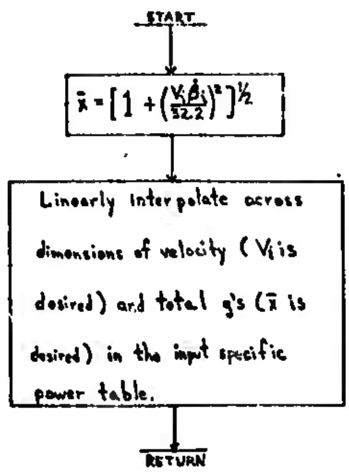
THE $G_i(X)$ ROUTINE



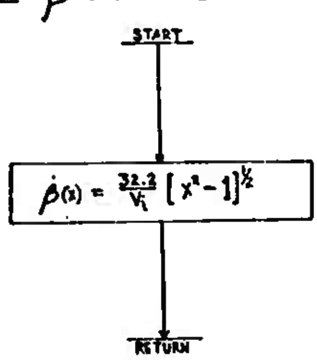
THE $g_i(X)$ ROUTINE



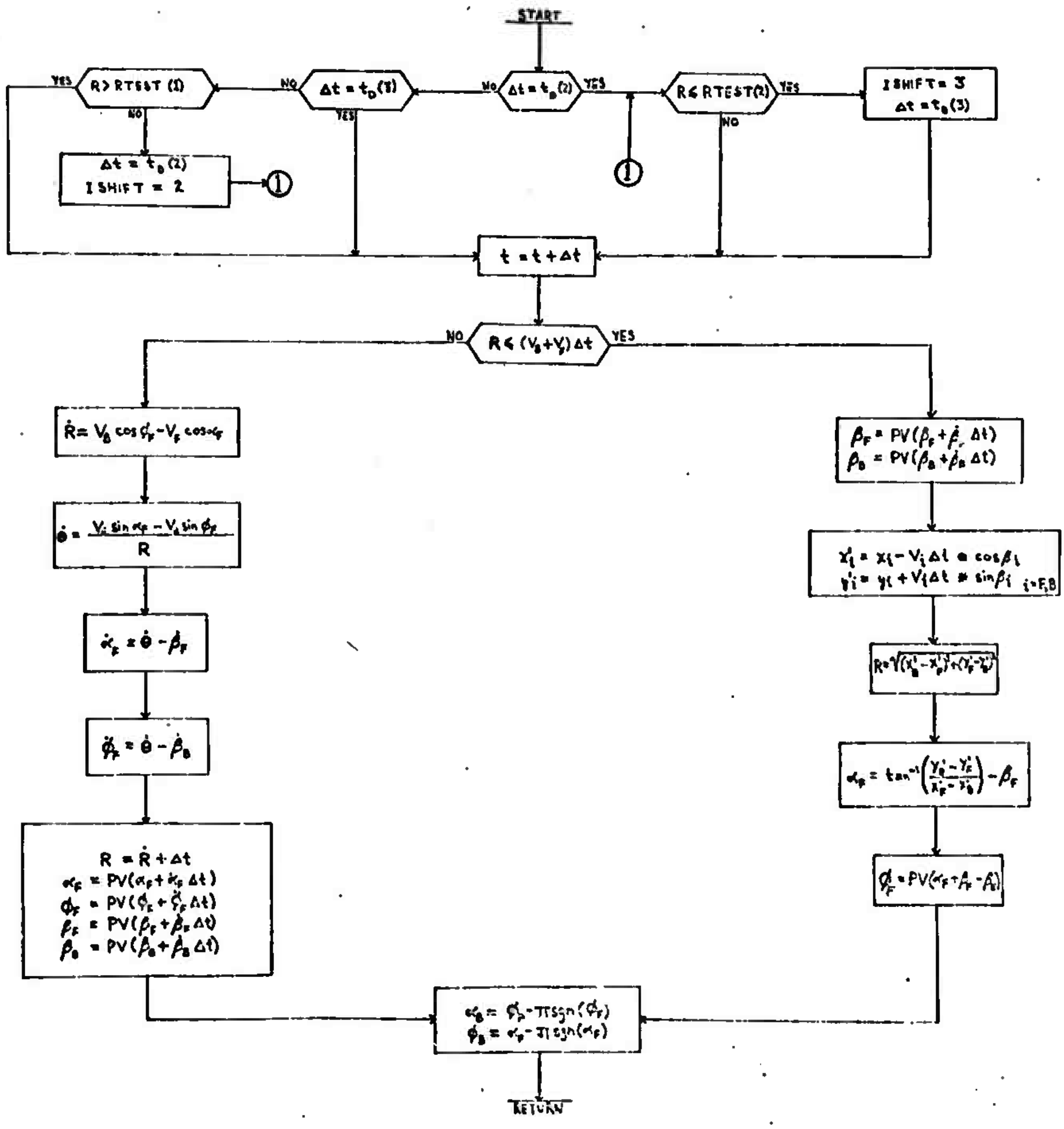
THE $P_i(V_i, \dot{\beta}_i)$ FUNCTION



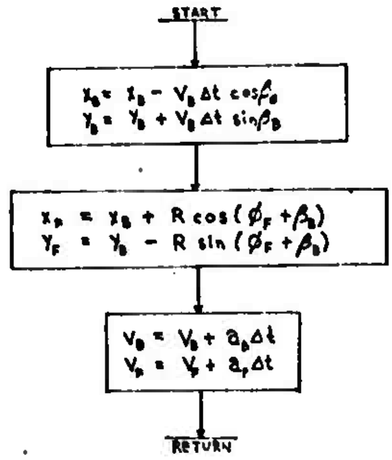
THE $\dot{\beta}(x)$ FUNCTION



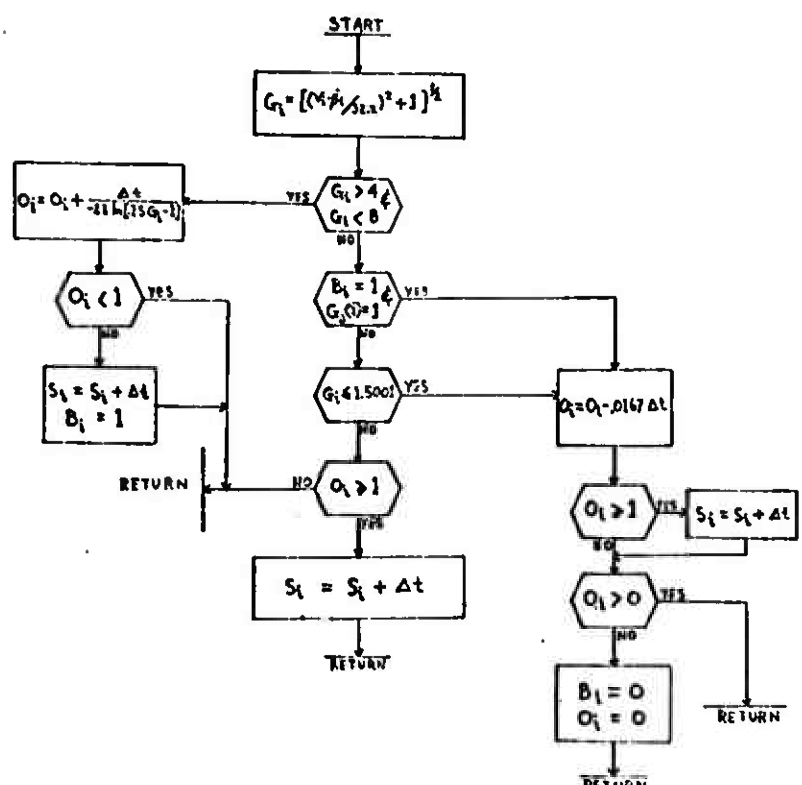
ADVANCE RELATIVE COORDINATES



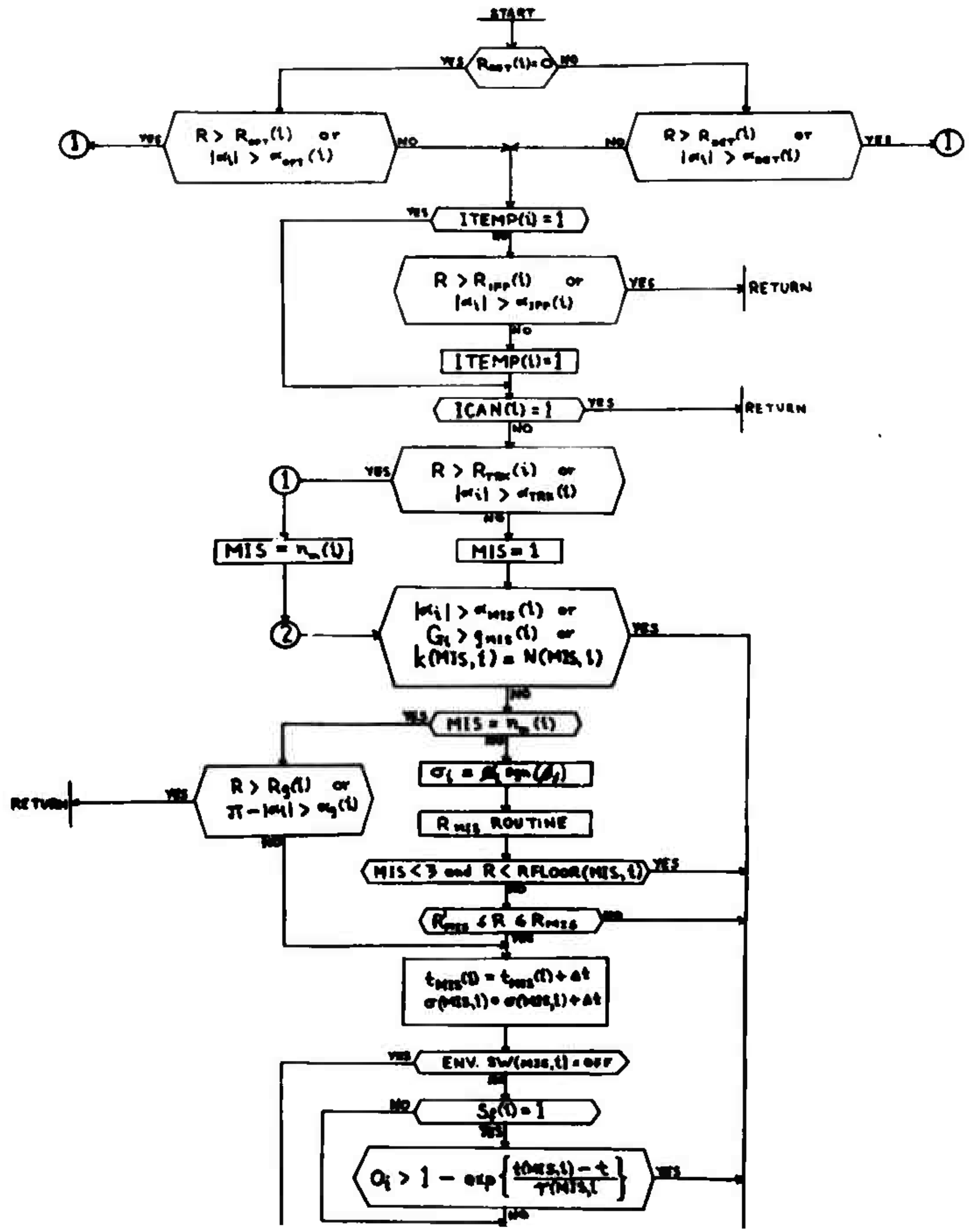
TRANSFORM TO INERTIAL COORDINATES ROUTINE



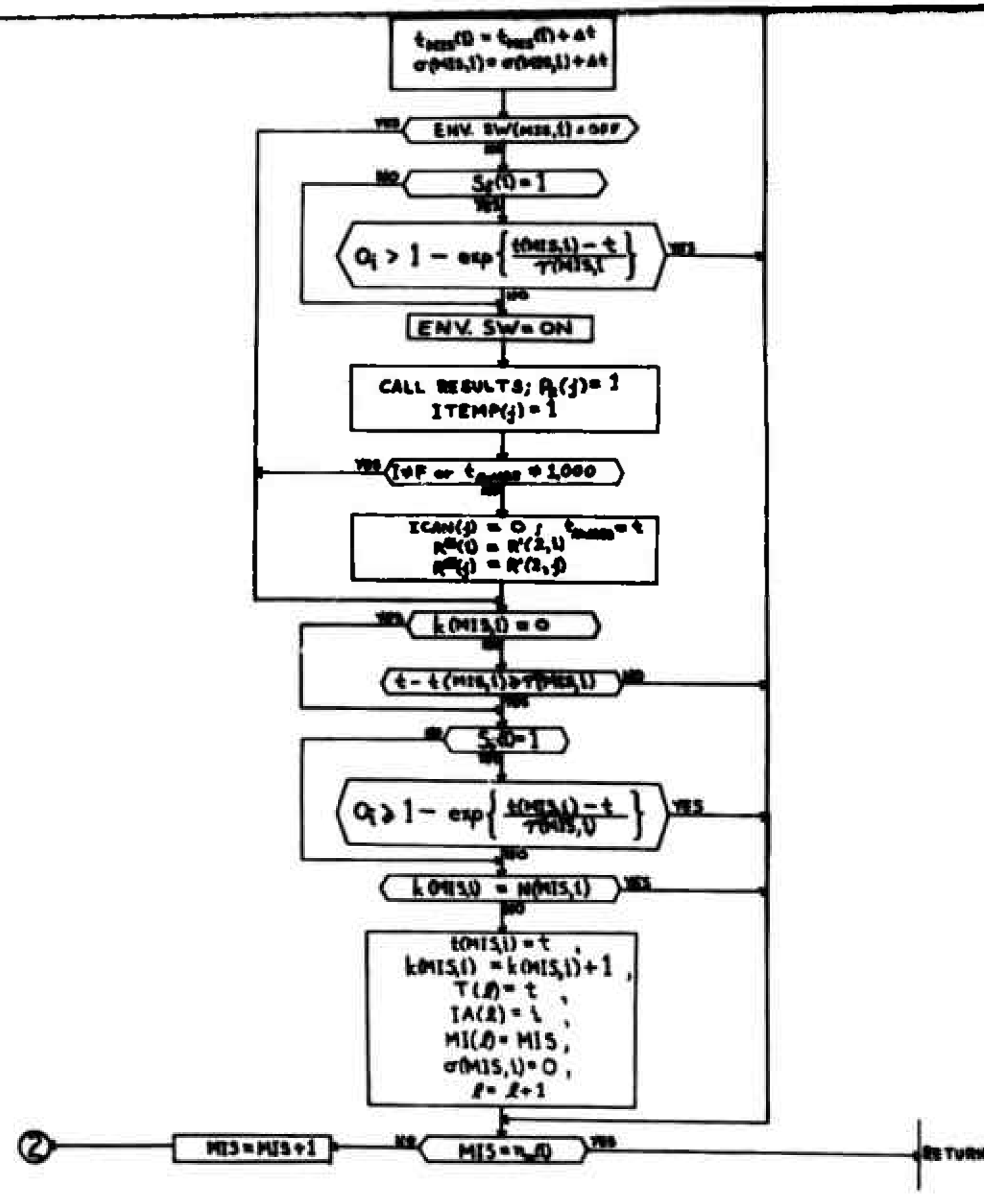
THE FIND G_i ROUTINE



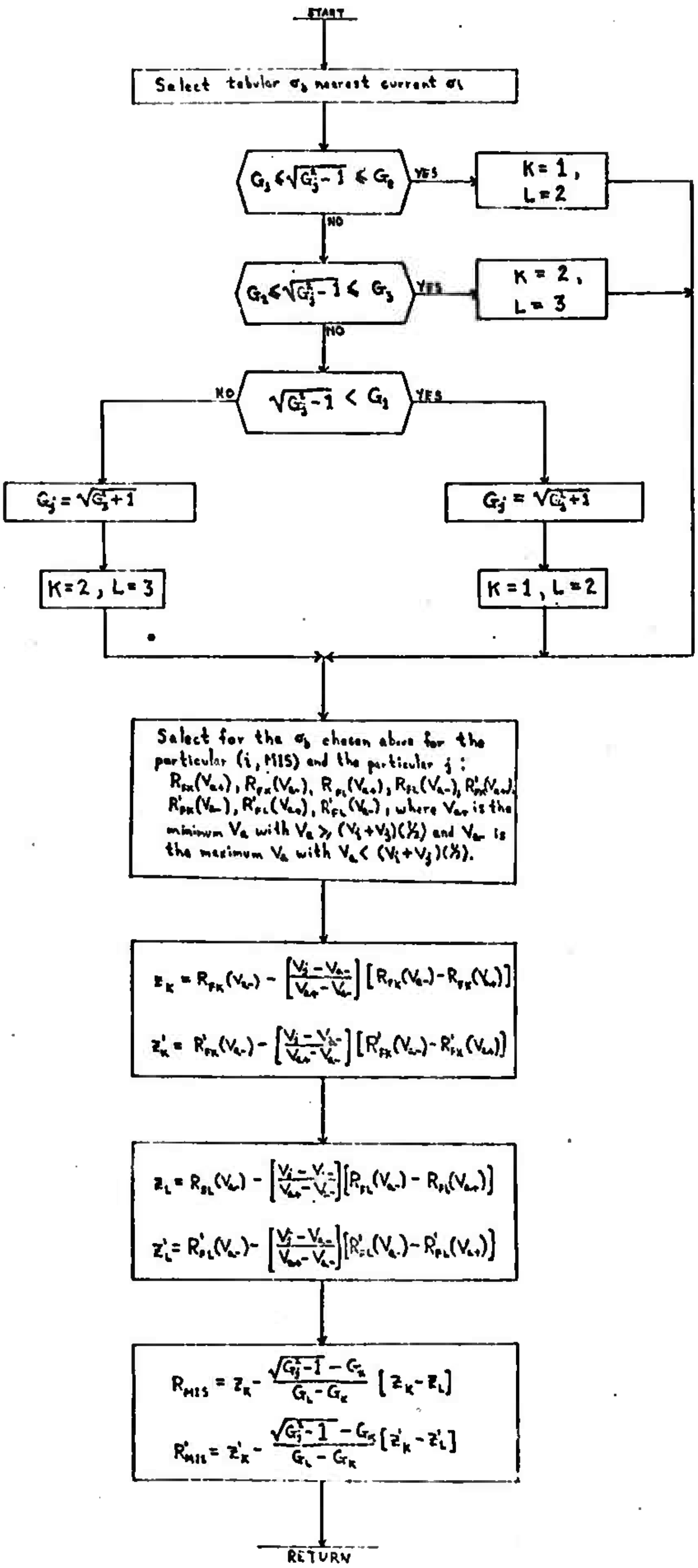
CHECK WEAPONS ROUTINE



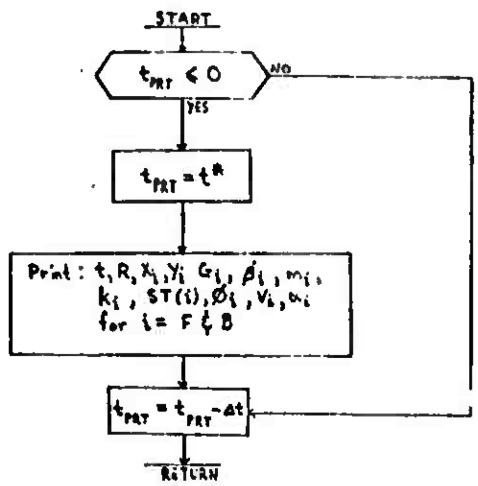
A


ENV. SW = ON
CALL RESULTS
ITEMP(j) = 1
YES
NO
RETURN
MIS = MIS + 1

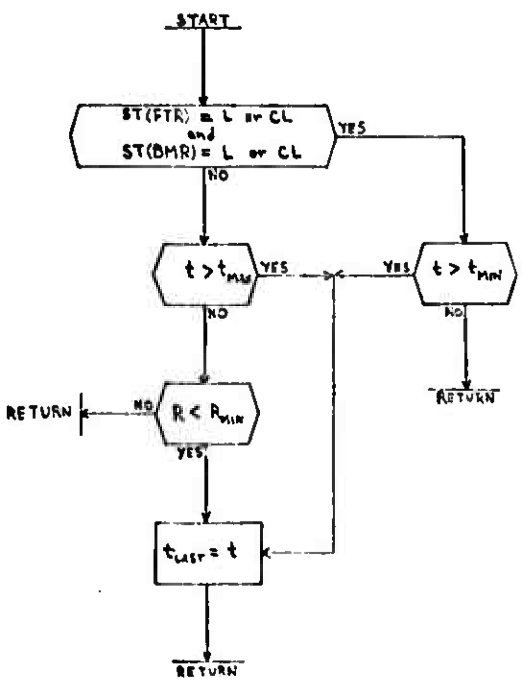
THE R_{MIS}, R'_{MIS} ROUTINE



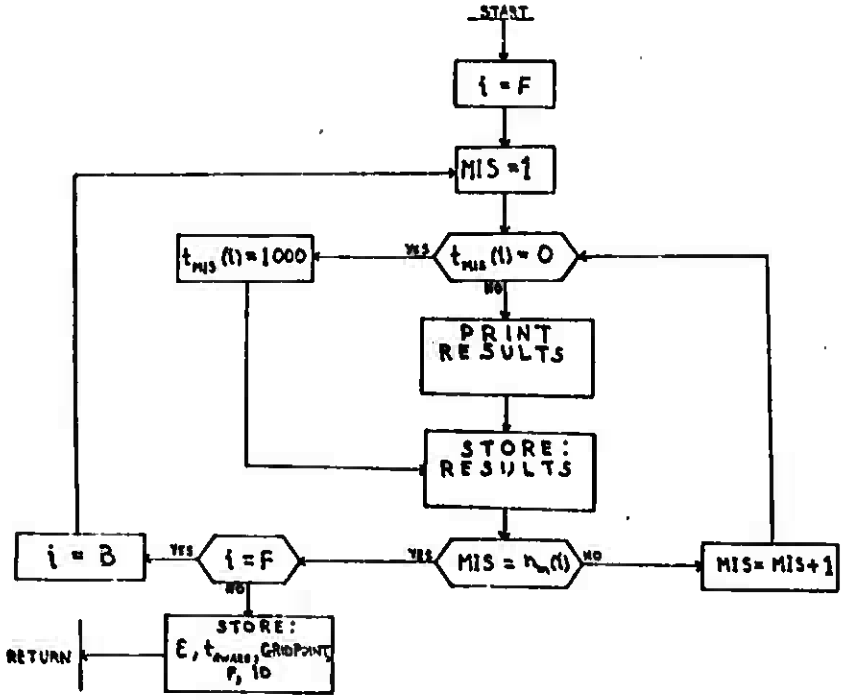
THE PRINT ROUTINE



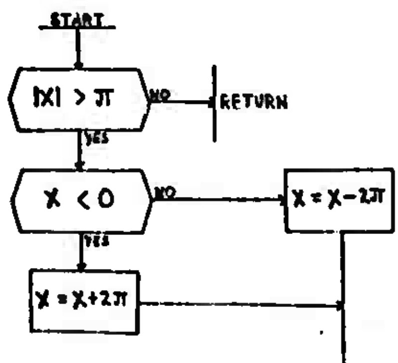
THE OVER ROUTINE



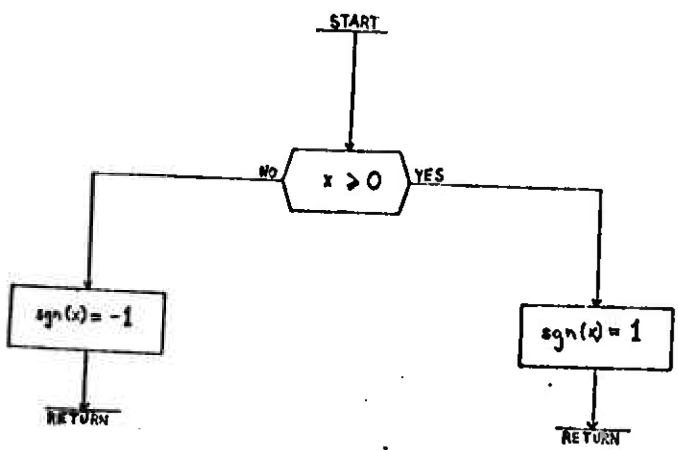
THE RESULTS ROUTINE



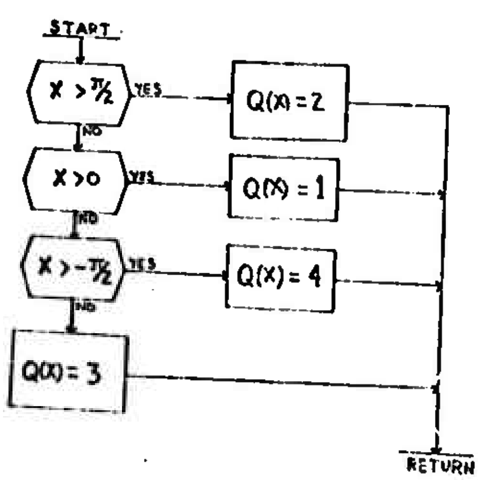
THE PV(X) FUNCTION



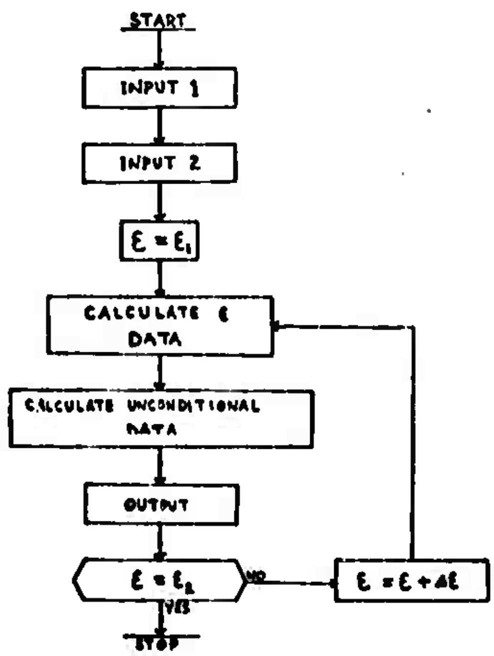
THE SGN(X) FUNCTION



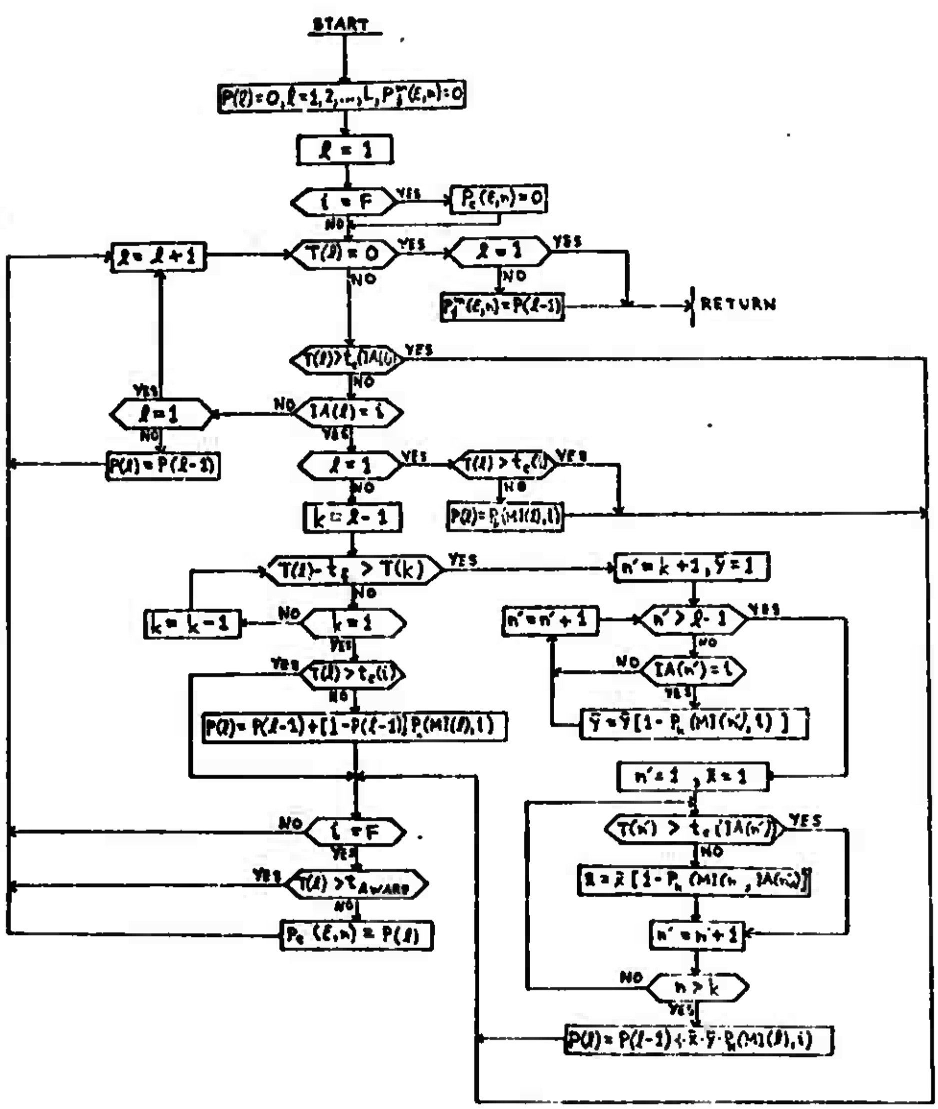
THE Q(X) FUNCTION



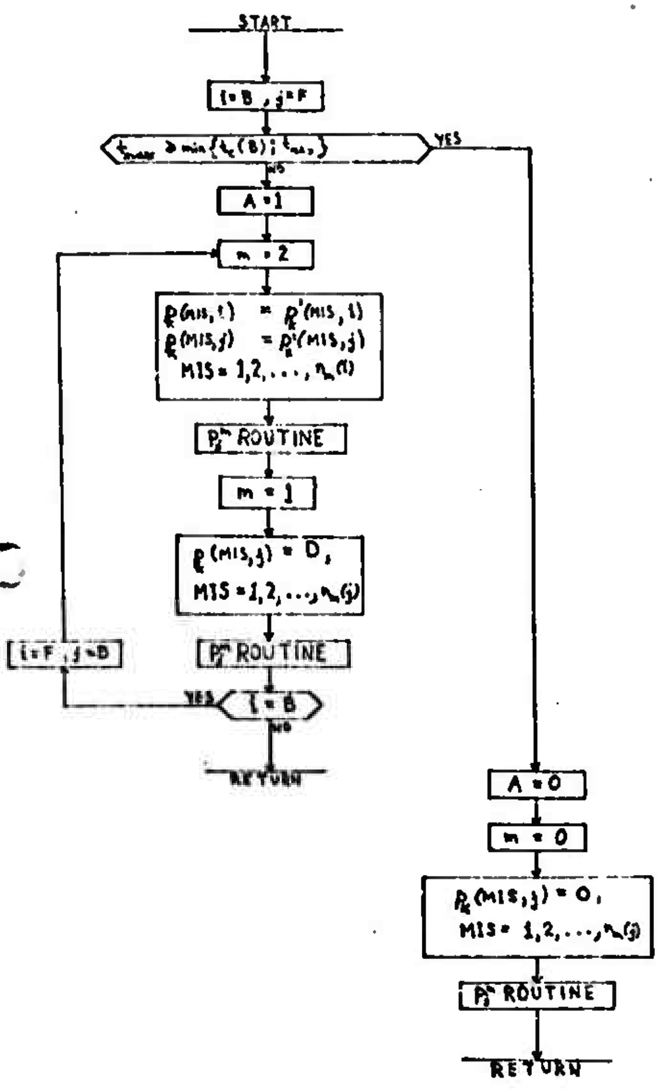
THE EXECUTIVE ROUTINE-DPM



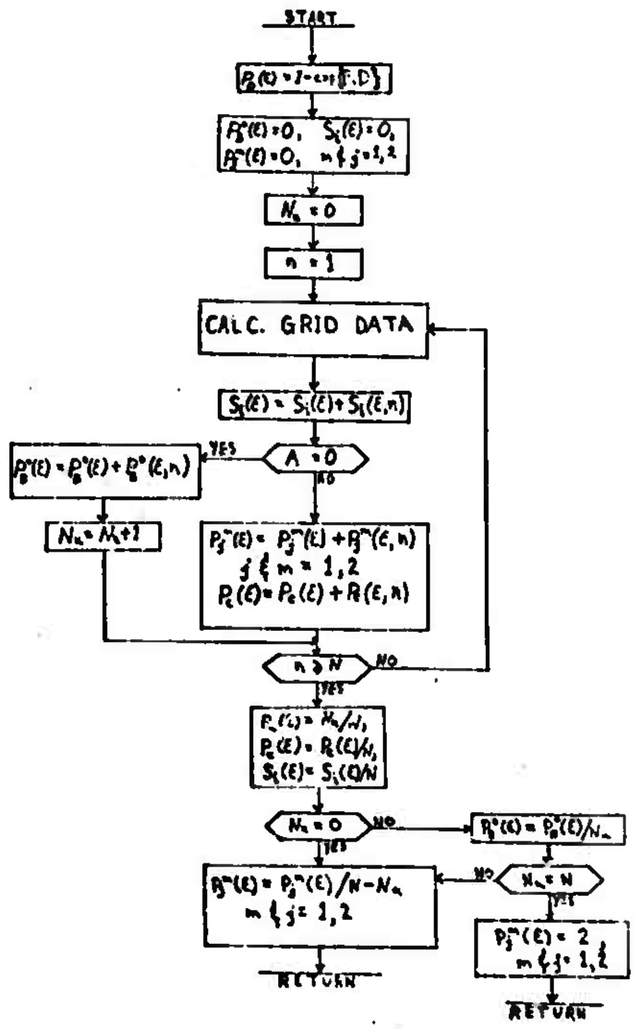
THE P_j^m ROUTINE



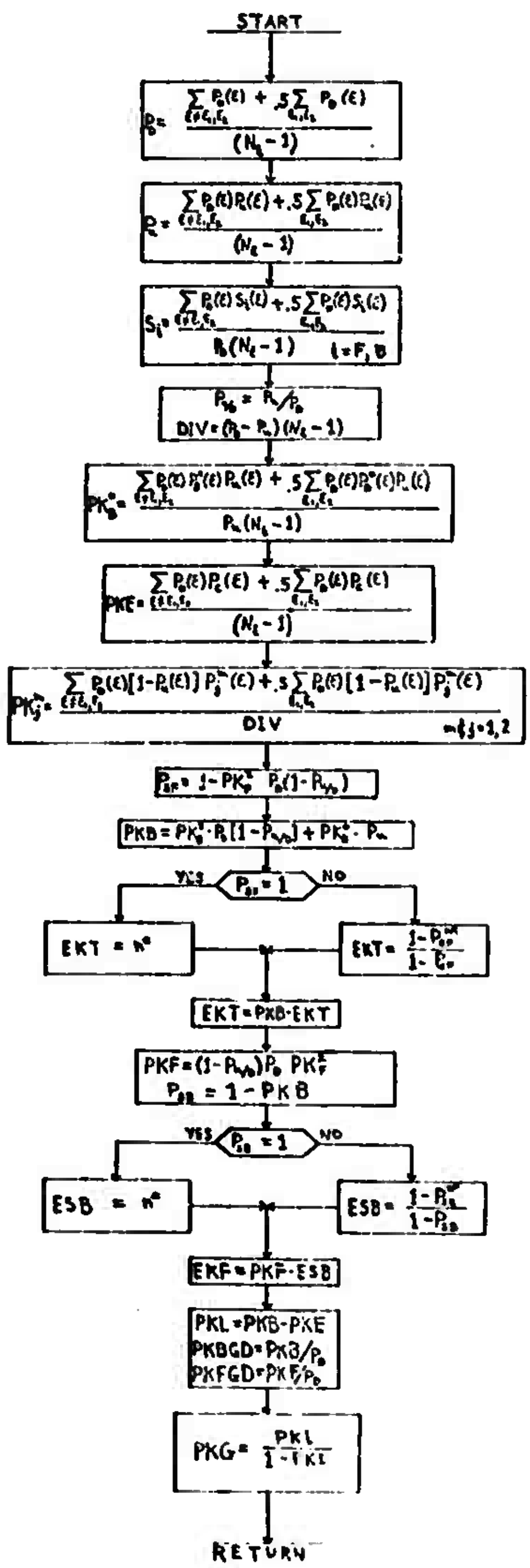
THE CALC. GRID DATA ROUTINE



THE CALC. ε DATA ROUTINE



THE CALC. UNCOND. DATA ROUTINE



SECTION 9

PROGRAM LISTING

In this section the FORTRAN IV program of ATAC-2 is presented. For the EM and DPM each, the program consists of a main program (corresponding to the EXECUTIVE Routine) and various FORTRAN subroutines all of which correspond to the previously presented routines.

```
$IBFTC SUB6    XR7,DECK
      SUBROUTINE NAVSYS
      COMMON/MAXALP/AM(2)
      COMMON/EXEC/RSMALL(2),RHO(4),IFTR,IBMR,FM(2),BSMALL(2),
      COMMON/PURSUE/A(2),BETMAX(2),THEDOT,PHISTR,ETA,ALPMAX(2),XLAMDA(2)
     *,XPHI
      COMMON/BETFIX/BETTAB(27,2),VSTR(2)
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NM(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON/MANUV/IMSTAT(2)
      COMMON/GRIDSO/GAMMA,ALPHA(2),R
      COMMON/INIFLT/T,TPRT,BETDOT(2),ENVSW(6,2),
     1IST(2),V(2),II,JJ,TM,S(5,2),PHI(2),Y(2),X(2),
     2DX(2),DY(2),TAWARE,G(2),LEFTSW
      COMMON/SLOWUP/DECLE
      COMMON/NAVSY/GLAT(2),VZERO,S,ASMALL,RC1,
     1FLAMDA,M,XC(2),YC(2)
      COMMON/RVARY/KCAPP,KLITLP,TATRPR(6,2),TTHERE(2),MISRPR(6,2),TOLER,
     *RPRIME(6,2)
      COMMON/COMBA/I,J
      COMMON /COMP/ ICOMP
      COMMON/COWARD/VPRIME(2), K  ,D(11,2),GCAP(2)
      COMMON/NUWRIT/KPRT(2)
      LOGICAL ACT,LEFTSW,PASS
      DIMENSION GMAX(2)
      PI=3.14159265
      GCAP(I)=GCAPF(V(I),I)
```

```
      GMAX(I)=GMAXF(V(I),I)
      BETMAX(I)=0.
      CALL INFO
      ALPMAX(I)=AM(I)
      IF(I.NE.IFTR) GOTO 52
      IF(TAWARE.NE.1000.) GOTO 52
      ALPMAX(I)=ALPDET(I)
52    CONTINUE
      KPRT(I)=K
      IF (K.EQ.1) GO TO 700
      IF (D(K,I).NE.0.) GO TO 100
      IF ((K.EQ.2).OR.(K.GE.8)) GOTO 12
      IF(ABS(PHI(I)).GT.PI/2.) GOTO 12
      IST(I)=5
      IF (V(I).GE.VSTR(I)) GOTO 10
      CAPX=AMIN1(GCAP(I),GMAX(I))
      BETDOT(I)=-SGN(BETDOT(J))*BFUNCT(CAPX,V(I),I)
     **(V(I)/VSTR(I))
      IMSTAT(I)=12
      GOTO 13
   10 CONTINUE
      CAPX=AMIN1(GCAP(I),GMAX(I))
      BETDOT(I)=-SGN(BETDOT(J))*BFUNCT(CAPX,V(I),I)
      A(I)=32.2/V(I)*PEEU(V(I),BETDOT(I),I)
      A(I)=AMIN1(AMAX1(A(I),(VC(I)-V(I))/DELTAT),(VMAX(I)-V(I))/DELTAT)
      IMSTAT(I)=11
      RETURN
   12 BETDOT(I)=0.
      IMSTAT(I)=13
      IST(I)=4
13    A(I)=32.2/V(I)*PEEU(V(I),BETDOT(I),I)
      A(I)=AMIN1(A(I),(VMAX(I)-V(I))/DELTAT)
      RETURN
100   IF((K.NE.10 ).AND.(K.NE.11)) GOTO 125
      IST(I)=3
      IF (V(I).LE.VSTR(I)) GOTO 106
      BETDOT(I)=SGN(BETDOT(I))* BFUNCT(GCAP(I),V(I),I)
```

```
      A(I)=AMIN1(32.2/V(I)*PEEU(V(I),BETDOT(I),I),ADEC(I))
      A(I)=AMAX1(A(I),(VC  (I)-V(I))/DELTAT)
      IMSTAT(I)=7
      RETURN
  106 CONTINUE
      BETDOT(I)=SGN(BETDOT(I))*BFUNCT(AMIN1(GCAP(I),GMAX(I)),V(I),I)
     **(V(I)/VSTR(I))
      A(I)=AMIN1(32.2/V(I)*PEEU(V(I),BETDOT(I),I),(VSTR(I)-V(I))/DELTAT)
      IMSTAT(I)=8
      RETURN
  125 IF ((K.EQ.2).OR.(K.EQ.8).OR.(K.EQ.9)) GOTO 600
      THEDOT=(V(IFTR)*SIN(ALPHA(IFTR))-V(IBMR)*SIN(PHI(IFTR)))/R
      PHISTR=0.
      IF(V(I).LT.2.*V(J)) PHISTR=ARCOS(V(I)/(2.*V(J)))
      PHISTR=XPHI*PHISTR
      IF (ABS(PHI(I)).GE.PHISTR) GO TO 2
      ETA=-XLAMDA(I)*SGN(PHI(I))
      GO TO 3
    2 ETA=SGN(PHI(I))*(ALPMAX(I)/(PHISTR-PI)*(ABS(PHI(I))-PHISTR)-
     *XLAMDA(I))
      IF (ABS(PHI(I)).EQ.PI) ETA=SGN(ALPHA(I))*(ALPMAX(I)+XLAMDA(I))
    3 CONTINUE
      BETDOT(I)=THEDOT+(ALPHA(I)-ETA)/DELTAT
      IF (ABS(PHI(I)).LE.PI/2.) GO TO 200
      CAPX=AMIN1(GCAP(I),GMAX(I))
      CAPX=BFUNCT(CAPX,V(I),I)
      S=((V(I)-VSTR(I))*(V(I)+VSTR(I)-2.*V(J)*COS(PHI(I)))))/
     *(-2.*ADEC(I))
      IF (R.LT.S) GO TO 126
      IMSTAT(I)=5
      IF (V(I).LT.VSTR(I)) CAPX=CAPX*V(I)/VSTR(I)
      A(I)=32.2/V(I)*PEEU(V(I),BETDOT(I),I)
      GOTO 127
  126 CONTINUE
      A(I)=ADEC(I)
      IMSTAT(I)=6
  127 IF(ABS(BETDOT(I)).LT.CAPX) GOTO 128
```

```
      BETDOT(I)=SGN(BETDOT(I))*CAPX
      IF (IMSTAT(I).EQ.5) A(I)=32.2/V(I)*PEEU(V(I),BETDOT(I),I)
      IST(I)=0
      GOTO 129
  128 IST(I)=2
  129 CONTINUE
      A(I)=AMIN1(AMAX1(A(I),(VC(I)-V(I))/DELTAT),(VMAX(I)-V(I))/DELTAT)
      RETURN
  200 CONTINUE
      XX=ALPHA(I)
      IF(ABS(XX).EQ.PI/2.) XX=SGN(XX)*PI/2.01
      VZERO=AMAX1(VC(I),V(J)*COS(PHI(I))/COS(XX))
      IF (V(I).LT.VZERO) GO TO 800
      YNEW=V(I)+((32.2/V(I))*PEEU(V(I),0.,I)*DELTAT)
      S=((YNEW-VZERO)*(YNEW+VZERO-2.*V(J)*COS(PHI(I))))/(-2.*ADEC(I))
      IF (R.GT.RSTAR(I)+S-(V(J)*COS(PHI(I))-YNEW*COS(ALPHA(I)))
     **DELTAT) GO TO 300
      IMSTAT(I)=1
  250 CONTINUE
      IST(I)=2
      IF(ABS(BETDOT(I)).LT.BFUNCT(GCAP(I),V(I),I))GOTO 225
      BETDOT(I)=SGN(BETDOT(I))*BFUNCT(GCAP(I),V(I),I)
      IST(I)=0
  225 A(I)=AMIN1(ADEC(I),32.2/V(I)*PEEU(V(I),BETDOT(I),I))
      A(I)=AMIN1(A(I),(VMAX(I)-V(I))/DELTAT)
      IF (((V(I)+A(I)*DELTAT).LT.VZERO).AND.(R.GT.RSTAR(I))) A(I)=0.
      RETURN
  300 CONTINUE
      CAPX=BFUNCT(AMIN1(GCAP(I),GMAX(I)),V(I),I)* AMIN1(V(I)/VSTR(I),1.)
      IMSTAT(I)=2
      IST(I)=2
      IF(ABS(BETDOT(I)).LT.CAPX) GOTO 310
      BETDOT(I)=SGN(BETDOT(I))*CAPX
      IST(I)=0
  310 CONTINUE
      A(I)=(32.2/V(I))*PEEU(V(I),BETDOT(I),I)
      A(I)=AMIN1(A(I),(VMAX(I)-V(I))/DELTAT)
```

```
      IF ((ABS(BETDOT(I)).EQ.CAPX).AND.(V(I).GE.VSTR(I))) A(I)=0.
      RETURN
  600 CONTINUE
      BETDOT(I)=BFUNCT(AMIN1(GCAP(I),GMAX(I)),V(I),I)*(V(I)/VSTR(I))
      IMSTAT(I)=10
      A(I)=AMIN1(32.2/V(I)*PEEU(V(I),BETDOT(I),I),(VSTR(I)-V(I))/DELTAT)
      IF (V(I).GT.VSTR(I)) BETDOT(I)=BFUNCT(GCAP(I),V(I),I)
      IF (V(I).LE.VSTR(I)) GO TO 604
      A(I)=AMIN1(32.2/V(I)*PEEU(V(I),BETDOT(I),I),ADEC(I))
      IMSTAT(I)=9
  604 CONTINUE
      IF (IPS(I).EQ.1) GOTO 602
      IF(ALPHA(I)) 602,601,603
 601  IF(PHI(I).LE.0.) GOTO 603
 602  BETDOT(I)=-BETDOT(I)
 603  IST(I)=0
      RETURN
 700  CONTINUE
      A(I)=0.
      IST(I)=1
      BETDOT(I)=0.
      IMSTAT(I)=0
      RETURN
  800 CAPX=BFUNCT(AMIN1(GCAP(I),GMAX(I)),V(I),I)* AMIN1(V(I)/VSTR(I),1.)
      IMSTAT(I)=3
      IF (R.GT.RSTAR(I)) GO TO 802
      CAPX=BFUNCT(AMIN1(GCAP(I),GMAX(I)),V(I),I)
      IMSTAT(I)=4
      YNEW=((RSTAR(I)*BETDOT(J))/V(J))**2
      IF (YNEW.GT.1.) GO TO 850
      YNEW=V(J)*SQRT(1.-YNEW)
      IF (YNEW.LT.VC(I)) YNEW=VC(I)
      IF(ABS(BETDOT(J)).GT.BFUNCT(AMIN1(GCAP(I),GMAX(I)),YNEW,I))
     *GO TO 850
      IF (YNEW.LT.V(I)) GO TO 825
      A(I)=AMIN1(32.2/V(I)*PEEU(V(I),CAPX,I),(YNEW-V(I))/DELTAT)
      GO TO 802
```

```
825 A(I)=AMAX1(ADEC(I),(YNEW-V(I))/DELTAT)
    GO TO 802
850 IF(V(I).GT.VSTR(I)) GO TO 852
    IF(ABS(BETDOT(J)).GT.CAPX) GOTO 851
    A(I)=0.
    GO TO 802
851 YNEW=V(J)
    A(I)=AMIN1(32.2/V(I)*PEEU(V(I),CAPX,I),(YNEW-V(I))/DELTAT)
    GO TO 802
852 YNEW=VSTR(I)
    GO TO 825
802 CONTINUE
    IST(I)=2
    IF (ABS(BETDOT(I)).LE.CAPX) GO TO 804
    BETDOT(I)=SGN(BETDOT(I))*CAPX
    IST(I)=0
804 CONTINUE
    IF (IMSTAT(I).NE.3) GO TO 805
    A(I)=(32.2/V(I))*PEEU(V(I),BETDOT(I),I)
    A(I)=AMIN1(AMAX1(A(I),(VC(I)-V(I))/DFLTAT),(VMAX(I)-V(I))/DELTAT)
    IF ((ABS(BETDOT(I)).EQ.CAPX).AND.(V(I).GE.VSTR(I))) A(I)=0.
805 CONTINUE
    RETURN
    END
```

```
$IBFTC TRK      XR7,DECK
      SUBROUTINE RPHI
      COMMON /RADAR/ P1(2),TAU(2),P2(2),S(6,2),RI,RPHIJ,NUIND
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NV(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADFC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON/DBLLNC / SIGMIS(6,2),KOUNTR(6,2),NUMIS(6,2),TAUMIS(6,2)
     *,TFIREI(25),MISTPI(25),IAFIRI(25),TFIREP(25,10),MISTPP(25,10),IAFI
     *RP(25,10),LCAP,LLITL
      COMMON/INIFLT/T,TPRT,BETDOT(2),FNVSW(6,2),
     1IST(2),V(2),II,JJ,TMIS(6,2),PHI(2),Y(2),X(2),
     2DX(2),DY(2),TAWARF,G(2),LEFTSW
      COMMON/GRIDSO/GAMMA,ALPHA(2),R
      COMMON/COMBA/I,J
      COMMON/NOWFIR/ICAN(2),RNOW(2)
C     NEED INPUT COMMON,INIFLT,RADAR
      PI=3.14159265
      IF((ABS(ALPHA(J)).LT.PI/2.).AND.(ABS(PHI(J)).LT.PI/2.)) GOTO 50
 5    CONTINUE
      IPHI=1
      DIFF=ABS(PHI(J)-SIGTAB(1))
      DO 10 LIST=2,NSIGMA
      IF (ABS(PHI(J)-SIGTAB(LIST)).GE.DIFF) GO TO 10
      IPHI=LIST
      DIFF=ABS(PHI(J)-SIGTAB(LIST))
   10 CONTINUE
      VTEST=(V(I)+V(J))/2.
      DO 20 LIST=2,NVA
```

```
      IF (VATAB(LIST).LT.VTEST) GO TO 20
      IVAP=LIST
      IVAM=LIST-1
      GO TO 21
   20 CONTINUE
      IVAP=NVA
      IVAM=NVA-1
      DIMENSION TEST(6)
   21 NMIS=NM(J)-1
      DO 25 K=1,NMIS
      ARGM=RFIT(IVAM,IPHI,K,J)
      ARGP=RFIT(IVAP,IPHI,K,J)
   25 TEST(K)=ARGM+((ARGP-ARGM)*(VTEST-VATAB(IVAM)))/
     *(VATAB(IVAP)-VATAB(IVAM))
      IF (NMIS.EQ.1) GO TO 40
      RPHIJ=0.
      DO 28 K=1,NMIS
      IF (NUMIS(K,J).EQ.0) GO TO 28
      IF (RPHIJ.GE.TEST(K)) GO TO 28
      IJ=K
      RPHIJ=TEST(K)
   28 CONTINUE
      IF (ABS(ALPHA(J)).GE.ALPMIS(IJ,J)) RPHIJ=0.
      IF (G(J).GE.GMIS(IJ,J)) RPHIJ=0.
      IF(RPHIJ.NE.0.) ICAN(J)=0
      RETURN
   40 RPHIJ=TEST(1)
      IF (ABS(ALPHA(J)).GE.ALPMIS( 1,J)) RPHIJ=0.
      IF (G(J).GE.GMIS(1 ,J)) RPHIJ=0.
      IF(RPHIJ.NE.0.) ICAN(J)=0
      RETURN
   50 CONTINUE
      IF (R.LE.RNOW(J)) GO TO 5
      RPHIJ=0.
      RETURN
      END
```

```
$IBFTC PASSI.  XR7,DECK
      SUBROUTINE PASSIV(PASS)
      COMMON/DBLLNC / SIGMIS(6,2),KOUNTR(6,2),NUMIS(6,2),TAUMIS(6,2)
     *,TFIREI(25),MISTPI(25),IAFIRI(25),TFIREP(25,10),MISTPP(25,10),IAFI
     *RP(25,10),LCAP,LLITL
      COMMON/RIFFCK/ITEMP(2)
      COMMON/ADRECO/RDOT,PHIDOT(2),ALPDOT(2),GPUR
      COMMON /RADAR/ P1(2),TAU(2),P2(2),S(6,2),R1,RPHIJ,NUIND
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NM(2),IPS(2),NID,
     IVI(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON/GRIDSO/GAMMA,ALPHA(2),R
      COMMON/INIFLT/T,TPRT,BETDOT(2),ENVSW(6,2),
     1IST(2),V(2),II,JJ,TMIS(6,2),PHI(2),Y(2),X(2),
     2DX(2),DY(2),TAWARE,G(2),LEFTSW
      COMMON/COMBA/I,J
      COMMON/EXEC/RSMALL(2),RHO(4),IFTR,IBMR,FM(2),BSMALL(2),
      LOGICAL LEFTSW,PASS
      LOGICAL ENVSW
      R1=RTRK(J)
      IF ((J.EQ.IBMR).AND.(TAWARE.EQ.1000.)) GO TO 35
      IF (J.EQ.IBMR) GO TO 40
      IF (NUIND.EQ.0) GO TO 40
      IF (NUIND.EQ.1) GO TO 5
      RI=0.
      IF (ITEMP(J).EQ.0) GO TO 9
    5 CONTINUE
      IF(P1(J).NE.I.) GO TO 30
```

```
      NMIS=NM(J)
      DO 25 L=1,NMIS
      IF (T.LT.DELTAT) GO TO 25
      IF (SIGMIS(L,J).GT.DELTAT) GO TO 25
      S(L,J)=1.
      R1=RTRK(J)
      GO TO 40
   25 CONTINUE
      PASS=.FALSE.
      RETURN
   30 IF(P2(J).NE.I.) GO TO 31
      RI=RTRK(J)
      GO TO 40
   31 CALL RPHI
      IF(RPHIJ.LE.0.) GO TO 35
      IF (RDOT.GE.0.) GO TO 36
      R1= AMIN1(RTRK(J),RPHIJ-TAU(J)*RDOT)
      GO TO 40
   35 PASS=.FALSE.
      RETURN
   36 IF (RPHIJ.LT.R) GO TO 35
      R1=RTRK(J)
   40 IF((R.LE.RPAS(I)).AND.(R.LE.R1).AND.
     1(ABS(ALPHA(J)).LE.ALPTRK(J)).AND.
     2(ABS(PHI(J)).LE.ALPPAS(I))) GO TO 10
9     PASS=.FALSE.
      RETURN
10    IF(IPS(I).EQ.1)GO TO 15
      IF(IPS(I).NE.2)GO TO 9
13    CALL AWARE
      PASS=.TRUE.
      P2(J)=1.
      RETURN
15    LEFTSW=.TRUE.
      GO TO 13
      END
      GO TO 10
```

```
19 IF ((PI-ABS(ALPHA(I)).LE.ALPGUN(I)).AND .(R.LE.RANGUN(I))) GO TO21
   RETURN
21 ITEMP(I)=1
   GO TO 12
11 MIS=NM(I)
   GO TO 10
   END
```

```
$IBFTC MAIN    XR7,DECK
      COMMON/NOWFIR/ICAN(2),RNOW(2)
      COMMON /PLTIME/
     *      EPSPLT(5),IGRPLT(5),PLOTIM,NOWPLT,KOUNT,XPLT(300,4),YPLT(300
     *,4),XN,XX,YN,YX,XLL,XRL,YLL,YUL,NP(4),IPLOT,TIMPLT,IERR
      COMMON/COWARD/VPRIME(2), K   ,D(11,2),GCAP(2)
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,RANGE,GT(3),RMIN,H,
     *N,NM(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON/EXEC/RSMALL,RHO,IFTR,IBMR,FM(2),BSMALL(2)
      COMMON /COMP/ ICOMP
      COMMON/PURSUE/A(2),BETMAX(2),THEDOT,PHISTR,ETA,ALPMAX(2),XLAMDA(2)
     *,XPHI
      COMMON/DBLLNC / SIGMIS(6,2),KOUNTR(6,2),NUMIS(6,2),TAUMIS(6,2)
     *,TFIREI(25),MISTPI(25),IAFIRI(25),TFIREP(25,10),MISTPP(25,10),IAFI
     *RP(25,10),LCAP,LLITL
      COMMON/SLOWUP/DECLF
      COMMON /RADAR/ P1(2),TAU(2),P2(2),S(6,2),R1,RPHIJ,NUIND
      COMMON/REARGN/RANGUN(2),ALPGUN(2)
      COMMON/BETFIX/BETTAB(27,2),VSTR(2)
      COMMON/CORPUS/OXDEBT(2),BINDEX(2),GB(2),SICKF(2),OX(2)
      COMMON/TOTIME/TDELTS(3),RTEST(2)
      COMMON/RVARY/KCAPP,KLITLP,TATRPR(6,2),TTHERE(2),MISRPR(6,2),TOLER,
     *RPRIME(6,2)
      COMMON/MAXALP/AM(2)
      NAME LIST/SLODWN/DECLE
      NAME LIST/PURS/ALPMAX,XLAMDA,XPHI
      NAME LIST/FIRINE/TAUMIS,NUMIS,LCAP
```

```
      NAMELIST/STAPUT/
     *KCAPP,TATRPR,RPRIME,MISRPR,TOLER
      NAME LIST/WRITAL/ALPMAX,XLAMDA,P1,TAU,TAUMIS,NUMIS,LCAP
     *,VSTR,VPRIME,RANGUN,ALPGUN,RTEST,TDELTS,D,NUIND,GB,SICKF
     *,KCAPP,TATRPR,RPRIME,MISRPR,TOLER
      NAME LIST/TAIL/RANGUN,ALPGUN
      NAME LIST/IMSICK/GB,SICKF
      DATA PIE/3.1415927/,RADIAN/.01745329/,RTD/57.2957795/
      NRUN=1
      IETR=1
      IRMR=2
      WRITE(6,30)
   30 FORMAT(1H1)
      CALL INPUT
      DO 3 I1=1,2
      MISTOL=NM(I1)
      DO 3 J1=1,MISTOL
    3 ALPMIS(J1,I1)=ALPMIS(J1,I1)*RADIAN
      DO 4 I1=1,NSIGMA
    4 SIGTAB(I1)=SIGTAB(I1)*RADIAN
    1 DO 2 I1=1,2
      ALPDET(I1)=ALPDET(I1)*RADIAN
      ALPIFF(I1)=ALPIFF(I1)*RADIAN
      ALPOPT(I1)=ALPOPT(I1)*RADIAN
      ALPPAS(I1)=ALPPAS(I1)*RADIAN
    2 ALPTRK(I1)=ALPTRK(I1)*RADIAN
      WRITE(6,7)
    7 FORMAT(58X,16HTHE ATAC-2 MODEL)
      WRITE(6,6)(XID(ID),ID=1,NID)
    6 FORMAT(1H0/(30X,12A6))
      READ (5,101)IPLOT
      IF (IPLOT.EQ.0) GO TO 9
      READ (5,102) (ERSPLT(I1),I1=1,IPLOT)
      READ (5,103) (IGRPLT(I1),I1=1,IPLOT)
      DO 8 I1=1,IPLOT
    8 ERSPLT(I1)=ERSPLT(I1)*RADIAN
      READ (5,102) PLOTIM
```

```
9     CONTINUE
101   FORMAT(19X,I10)
102   FORMAT(19X,5F10.0)
103   FORMAT(19X,5I10)
      READ (5,PURS)
      READ (5,SLODWN)
      READ (5,FIRINF)
      READ (5,TAIL)
      READ (5,IMSICK)
      READ(5,STAPUT)
      ALPMAX(1)=ALPMAX(1)*RADIAN
      ALPMAX(2)=ALPMAX(2)*RADIAN
      XLAMDA(1)=XLAMDA(1)*RADIAN
      XLAMDA(2)=XLAMDA(2)*RADIAN
      ALPGUN(1)=ALPGUN(1)*RADIAN
      ALPGUN(2)=ALPGUN(2)*RADIAN
      WRITE (6,WRITAL)
      SDELEP=DELEPS
      SEPSLO=EPSLON
      DELEPS=DELEPS*RADIAN
      SIGB=SIGB*RADIAN
      EPSLON=EPSLON*RADIAN
      IEPS=0.
      IF(DELEPS.GT.PIE)GO TO 10
      IF((IPS(IBMR).EQ.1).OR.(IPS(IFTR).EQ.1))GO TO 20
      EPSLON=0.
      GO TO 10
20    EPSLON=-PIE
10    CONTINUE
      NTEPS=(PIE-EPSLON)/DELEPS+1.
   50 FORMAT(1H0,5I5)
      WRITE (11,50) NRUN,NTEPS,N,NM(1),NM(2)
      WRITE (11,40) NID,(XID(ID),ID=1,NID)
      WRITE (11,50) ICOMP
   40 FORMAT(14X,I2/(12A6))
      IF(RDET(IFTR).EQ.0.)GO TO 11
      RSMALL=RDET(IFTR)
```

```
      RHO=ALPDET(IFTR)
      GO TO 15
 11   RSMALL=ROPT(IFTR)
      RHO=ALPOPT(IFTR)
   15 CONTINUE
      RSMALL=AMIN1(RSMALL,RANGE)
      IEPS=IEPS+1
      AM(1)=ALPMAX(1)
      AM(2)=ALPMAX(2)
  201 CONTINUE
      CALL GRIDPR
      CALL COMBAT(IEPS)
      EPSLON=EPSLON+DELEPS
      IF(EPSLON.LE.PIE)GO TO 201
      DO 25 I1=1,2
      ALPDET(I1)=ALPDET(I1)*RTD
      ALPIFF(I1)=ALPIFF(I1)*RTD
      ALPOPT(I1)=ALPOPT(I1)*RTD
      ALPPAS(I1)=ALPPAS(I1)*RTD
 25   ALPTRK(I1)=ALPTRK(I1)*RTD
      ALPGUN(2)=ALPGUN(2)*RTD
      ALPGUN(1)=ALPGUN(1)*RTD
      DELEPS=DELEPS*RTD
      EPSLON=EPSLON*RTD
      SIGR=SIGR*RTD
      ALPMAX(1)=ALPMAX(1)*RTD
      ALPMAX(2)=ALPMAX(2)*RTD
      XLAMDA(1)=XLAMDA(1)*RTD
      XLAMDA(2)=XLAMDA(2)*RTD
      DELEPS=SDELEP
      EPSLON=SEPSLO
      WRITE(6,30)
      IF (ICOMP.EQ.0) GO TO 250
      RDET(IFTR)=RDET(IFTR)/2.
      ROPT(IFTR)=ROPT(IFTR)/2.
      RSMALL=RSMALL*2.
  250 CONTINUE
```

```
CALL INPUT2
NRUN=NRUN+1
GO TO 1
END
```

```
$IBFTC COMBA.  XR7,DECK
      SUBROUTINE COMBAT(IEPS)
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NM(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON/SICLV/TIMSIC(2),SICTIM(2,25)
      COMMON /PLTINF/
     *       EPSPLT(5),IGRPLT(5),PLOTIM,NOWPLT,KOUNT,XPLT(300,4),YPLT(300
     *,4),XN,XX,YN,YX,XLL,XRL,YLL,YUL,NP(4),IPLOT,TIMPLT,IERR
      COMMON/EXEC/RSMALL,RHO,IFTR,IBMR,FM(2),BSMALL(2)
      COMMON/RIFFCK/ITEMP(2)
      COMMON/COMBA/I,J
      COMMON/RES/EPS(73),TLIT(25),AWARET(25),FPR(25),TMISP(6,2,25)
      COMMON/PRNT1/IPRINT
      COMMON/TOTIME/TDELTS(3),RTEST(2)
      COMMON/TIME/TTMIS(6,2,25)
      COMMON/DBLLNC / SIGMIS(6,2),KOUNTR(6,2),NUMIS(6,2),TAUMIS(6,2)
     *,TFIREI(25),MISTPI(25),IAFIRI(25),TFIREP(25,10),MISTPP(25,10),IAFI
     *RP(25,10),LCAP,LLITL
      COMMON/NAVSY/GLAT(2),VZERO,S,ASMALL,RC1,
     1FLAMDA,M,XC(2),YC(2)
      COMMON/DIANOS/VZERRT(2),SPRT(2)
      COMMON/TOM/ISHIFT
      COMMON/PURSUE/A(2),BETMAX(2),THEDOT,PHISTR,ETA,ALPMAX(2),XLAMDA(2)
     *,XPHI
      LOGICAL AGAIN
      IGRIDP=1
      NM1=NM(1)
```

```
      NM2=NM(2)
   10 CALL SOGRID
      DO 15 IADD=1,25
      TFIREP(IADD,IGRIDP)=0.
      MISTPP(IADD,IGRIDP)=0
      IAFIRP(IADD,IGRIDP)=0
   15 CONTINUE
      ITEMP(1)=0
      ITEMP(2)=0
      IPRINT=0
      NOWPLT=0
      TIMPLT=0.
      KOUNT=0
      IF (IPLOT.EQ.0) GO TO 13
      DO 11 I1=1,IPLOT
      IF (EPSLON.NE.EPSPLT(I1)) GO TO 11
      IF (IGRIDP.NE.IGRPLT(I1)) GO TO 13
      NOWPLT=1
      GO TO 13
   11 CONTINUE
   13 CONTINUE
      ISHIFT=1
      DELTAT=TDELTS(1)
      CALL INITFL
   20 I=IFTR
      J=IBMR
      CALL NAVSYS
      VZEPRT(I)=VZERO
      SPRT(I)=S
      I=IBMR
      J=IFTR
      CALL NAVSYS
      VZEPRT(I)=VZERO
      SPRT(I)=S
      I=IFTR
      J=IBMR
      CALL CKWEAR
```

```
      I=IBMR
      J=IFTR
      CALL CKWFAP
      CALL PRIFNE
      CALL ADRELC
      CALL TINERC
      I=IFTR
      CALL FINDG
      I=IBMP
      CALL FINDG
      CALL OVER(AGAIN)
      IF(.NOT.AGAIN)GO TO 20
      CALL RESULT(IGRIDP,IEPS)
      IF (IGRIDP.EQ.N) GO TO 12
      IGRIDP=IGRIDP+1
      GO TO 10
   12 WRITE(11,100)EPS(IEPS),(TLIT(IP),AWARET(IP),FPR(IP),IP=1,N),
     *((TMISP(MISS,1,NGP),MISS=1,NM1),NGP=1,N),
     *((TMISP(MISS,2,NGP),MISS=1,NM2),NGP=1,N)
      WRITE (11,100) ((TTMIS(MISS,1,NGP),MISS=1,NM1),NGP=1,N),
     *((TTMIS(MISS,2,NGP),MISS=1,NM2),NGP=1,N)
      DO 50 NGP=1,N
      IFZERO=0
      DO 45 IWRIT=1,LCAP
      IF(TFIREP(IWRIT,NGP).GT.0.) GO TO 44
      IF(IFZERO.NE.0) GO TO 46
      IFZERO=1
   44 WRITE (11,200)NGP,TFIREP(IWRIT,NGP),MISTPP(IWRIT,NGP),
     *IAFIRP(IWRIT,NGP),IFZERO
   45 CONTINUE
   46 CONTINUE
   50 CONTINUE
      WRITE (11,100) (SICTIM(1,NGP),SICTIM(2,NGP),NGP=1,N)
      WRITE (11,300) (IE(NGP),NGP=1,N)
  300 FORMAT(1H ,10I5)
  200 FORMAT(1H ,I5,E15.7,3I10)
  100 FORMAT(1H0,4E16.8/(17X,3E16.8))
      RETURN
      END
```

```
I=IBMR
J=IFTR
CALL CKWEAR
CALL PRIFNE
CALL ADRELC
CALL TINERC
I=IFTR
CALL FINDG
I=IBMR
CALL FINDG
```

```
$IBFTC INITL    XR7,DECK
      SUBROUTINE INITFL
      COMMON/MULFIR/TLASTF(6,2)
      COMMON/EXEC/RSMALL,RHO,IFTR,IBMR,FM(2),BSMALL(2)
      COMMON/CORPUS/OXDEBT(2),BINDEX(2)
      COMMON /RADAR/ P1(2),TAU(2),P2(2),S(6,2),R1,RPHIJ,NUIND
      COMMON/PRNT1/IPRINT
      COMMON/TINCOO/XP(2),YP(2),BETA(2)
      COMMON/GRAPHO/XTOP,XBOT,YTOP,YBOT
      COMMON/CKWEA/SIG(2),ANS(24,2)              ,MIS
      COMMON/SICLV/TIMSIC(2),SICTIM(2,25)
      COMMON/NUWRIT/KPRT(2)
      COMMON/MANUV/IMSTAT(2)
      COMMON/FIRWRT/RANFIR(25),ALPFIR(25),PHIFIR(25)
      COMMON/RVARY/KCAPP,KLITLP,TATRPR(6,2),TTHERE(2),MISRPR(6,2),TOLER,
     *RPRIME(6,2)
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NM(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON/INIFLT/T,TPRT,BETDOT(2),ENVSW(6,2),
     1IST(2),V(2),II,JJ,TMIS(6,2),PHI(2),Y(2),X(2),
     2DX(2),DY(2),TAWARE,G(2),LEFTSW
      COMMON/COMBA/I,J
      COMMON/DBLLNC / SIGMIS(6,2),KOUNTR(6,2),NUMIS(6,2),TAUMIS(6,2)
     *,TFIREI(25),MISTPI(25),IAFIRI(25),TFIREP(25,10),MISTPP(25,10),IAFI
     *RP(25,10),LCAP,LLITL
      COMMON/GRIDSQ/GAMMA,ALPHA(2),R
      COMMON/NOWFIR/ICAN(2),RNOW(2)
```

```
      COMMON/TOTIME/TDELTS(3),RTEST(2)
      COMMON/TOM/ISHIFT
      LOGICAL LEFTSW,ENVSW
      DATA PI/3.14159265/
  200 FORMAT(1H0,3X,1HT,5X,1HR,7X,2HXB,8X,2HYB,6X,2HVB,3X,2HMB,
     =3X,2HGB,4X,2HBB,2X,4HALPB,2X,1HI,2X,4HPHIB,2X,2HKB,5X,
     =2HXF,6X,2HYF,4X,2HVF,3X,2HMF,3X,2HGF,4X,2HBF,4X,4HALPF,1X,1HI,
     *2X,4HPHIF,2X,2HKF)
      WRITE (6,200)
      T=.000001
      IF (R.GT.RTEST(1)) GO TO 6
      DELTAT=TDELTS(2)
      ISHIFT=2
    5 CONTINUE
      IF(R.LE.RTEST(2)) ISHIFT=3
      IF (R.LE.RTEST(2)) DELTAT=TDELTS(3)
    6 CONTINUE
      TPRT=DELTAT
      BETDOT(IFTR)=0.
      BETDOT(IBMR)=0.
      OXDEPT(1)=0.
      OXDEPT(2)=0
      BINDEX(1)=0.
      BINDEX(2)=0.
      ICAN(1)=1
      ICAN(2)=0
      TIMSIC(1)=0.
      TIMSIC(2)=0.
      LEFTSW=.FALSE.
      KLITLP=1
      DO 70 IK=1,2
      RNOW(IK)=RPRIME(1,IK)
      TTHERE(IK)=0.
      RSTAR(IK)=RPRIME(1,IK)
      L=NX(IK)
      DO 70 IMIS=1,L
      TLASTF(IMIS,IK)=0.
```

```
      SIGMIS(IMIS,IK)=0.
      KOUNTR(IMIS,IK)=0
70    FNVSW(IMIS,IK)=.FALSE.
      LLITL=1
      DO 55 K=1,2
      NMIS=NM(K)
      P2(K)=0.
      DO 55 L=1,NMIS
   55 S(L,K)=0.
      DO 80 IK=1,25
      TFIREI(IK)=0.
      MISTPI(IK)=0
      IAFIRI(IK)=0
      RANFIR(IK)=0.
      ALPFIR(IK)=0.
      PHIFIR(IK)=0.
   80 CONTINUE
      IST(IFTR)=1
      IST(IBMR)=1
      V(IFTR)=VZ(IFTR)
      V(IBMR)=VZ(IBMR)
      II=IBMR
      JJ=IFTR
      DO 71 IK=1,2
      KPPT(IK)=0
      IMSTAT(IK)=0
      G(IK)=0.
      DO 60 KI=1,24
   60 ANS(KI,IK)=0.
      DO 61 KI=1,24,4
   61 ANS(KI,IK)=2000.
      L=NM(IK)
      DO 71 IMIS=1,L
71    TMIS(IMIS,IK)=0.
      DO 100 I1=1,2
      J1=2
      IF(I1.EQ.2) J1=1
```

```
      IF(ABS(ALPHA(I1)).GT.PI) ALPHA(I1) =SGN(ALPHA(I1))*(ABS(ALPHA(I1))
     1-2.*PI)
 90   IF(ALPHA(I1).GT.0.) GO TO 96
      PHI(J1)=ALPHA(I1)+PI
      GO TO 100
 96   PHI(J1)=ALPHA(I1)-PI
 100  CONTINUE
      X(IBMR)=0.
      Y(IBMR)=0.
      X(IFTR)=R*COS(PHI(IFTR)+PI/2.)
      Y(IFTR)=-R*SIN(PHI(IFTR)+PI/2.)
      IPRINT=1
      DX(IFTR)=0.
      DY(IFTR)=V(IFTR)*DELTAT
      DX(IBMR)=-V(IBMR)*DELTAT*XSIN(EPSLON)
      DY(IBMR)=V(IBMR)*DELTAT*XCOS(EPSLON)
      TAKARE=1000.
      G(IFTR)=1.
      G(IBMR)=1.
      BETA(IBMR)=PI/2.
      BETA(IFTR)=PI/2.-ALPHA(IFTR)+PHI(IFTR)
      XTOP=AMAX1(X(IBMR),X(IFTR))
      XBOT=AMIN1(X(IBMR),X(IFTR))
      YTOP=AMAX1(Y(IBMR),Y(IFTR))
      YBOT=AMIN1(Y(IBMR),Y(IFTR))
      RETURN
      END
```

```
$IBFTC PRINT    XR7,DECK
      SUBROUTINE PRIFNE
      COMMON/PURSUE/A(2),BETMAX(2),THEDOT,PHISTR,ETA,ALPMAX(2),XLAMDA(2)
     *,XPHI
      COMMON /PLTINF/
     *        EPSPLT(5),IGRPLT(5),PLOTIM,NOWPLT,KOUNT,XPLT(300,4),YPLT(300
     *,4),XN,XX,YN,YX,XLL,XRL,YLL,YUL,NP(4),IPLOT,TIMPLT,IERR
      COMMON/INIFLT/T,TPRT,BETDOT(2),ENVSW(6,2),
     1IST(2),V(2),II,JJ,TMIS(6,2),PHI(2),Y(2),X(2),
     2DX(2),DY(2),TANARF,G(2),LEFTSW
      COMMON/ADRECO/RDOT,PHIDOT(2),ALPDOT(2),GPUR
      COMMON/INPUTV/TMAX,TMIN,TSTAR,DENS,
     1DELEPS,DELTAT,EPSLON,SIGR,GT(3),RMIN,H,
     *N,NV(2),IPS(2),NID,
     1TA(2),TB(2),SF(2),SB(2),W(2),WGT(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPVIS(6,2),GMAXT(27,2),SIGTAR(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUMRC(2)
      COMMON/DIANOS/VZPRRT(2),SPRT(2)
      COMMON/MANUV/IMSTAT(2)
      COMMON/CORPUS/OXDEBT(2),BINDEX(2),GB(2),SICKF(2),OX(2)
      COMMON/EXEC/PSMALL,RHO,IFTR,IBMR,FM(2),BSMALL(2)
      COMMON/GRIDSQ/GAMMA,ALPHA(2),R
      COMMON/GRAPHO/XTOP,XBOT,YTOP,YBOT
      COMMON/NUMRIT/KPRT(2)
      COMMON/TOW/ISHIFT
      COMMON/RIFFCK/ITEMP(2)
      COMMON/NOWFIR/ICAN(2),RNOW(2)
      DATA PIE/3.1415927/,RADIAN/.01745329/,RTD/57.2957795/
      DIMENSION CL(2)
      FTNM=6076.10333
      IF (NOWPLT.EQ.0) GO TO 20
```

```
      IF (ABS(TIMPLT).GT..001) GO TO 15
      TIMPLT=PLOTIM
      IF (KOUNT.GE.300) GO TO 20
      KOUNT=KOUNT+1
      XPLT(KOUNT,1)=X(IBMR)/FTNM
      XPLT(KOUNT,2)=X(IFTR)/FTNM
      YPLT(KOUNT,1)=Y(IBMR)/FTNM
      YPLT(KOUNT,2)=Y(IFTR)/FTNM
      GO TO 20
   15 TIMPLT=TIMPLT-DELTAT
   20 CONTINUE
      IF(TSTAR.EQ.0.)RETURN
      TPRT=TPRT-DELTAT
      IF (TPRT.GT..001) GO TO 10
      TPRT=TSTAR
      XMACHB=V(IBMR)/W(2)
      XMACHF=V(IFTR)/W(1)
      IMB=IMSTAT(IBMR)
      IMF=IMSTAT(IFTR)
      CL(IFTR)=WGT(IFTR)/SF(IFTR)*(2./(DENS*V(IFTR)**2))*G(IFTR)
      CL(IBMR)=WGT(IBMR)/SF(IBMR)*(2./(DENS*V(IBMR)**2))*G(IBMR)
      ALPB=ALPHA(IBMR)*RTD
      ALPF=ALPHA(IFTR)*RTD
      PHIB =PHI(IBMR)*RTD
      PHIF=PHI(IFTR)*RTD
      WRITE (6,100) T,ISHIFT
     $                  ,R,X(IBMR),Y(IBMR),V(IBMR),IMB    ,G(IBMR),
     *BETDOT(IBMR),ALPB     ,IST(IBMR),PHIB          ,KPRT(IBMR),
     *X(IFTR),Y(IFTR),V(IFTR),IMF    ,G(IFTR),
     *BETDOT(IFTR),ALPF     ,IST(IFTR) ,PHIF          ,KPRT(IFTR)
  100 FORMAT(1H0,F5.1,I1,1X,
     $                      F7.0,1X,F8.0,1X,F8.0,1X,F5.0,1X,I3,1X,
     *F4.2,1X,F5.2,1X,F6.1,1X,I1,1X,F6.1, 1X,I2,1X,
     *F8.0,1X,F8.0,1X,F5.0,1X,I4   ,1X,F4.2,1X,F5.2,1X,F6.1,1X,I1,1XF6.1,
     *1X,I2)
      ETADEG=ETA*RTD
      TETADG=THEDOT*RTD
```

```
      ALFADG=ALPMAX(IFTR)*RTD
      XTOP=AMAX1(XTOP,X(IBMR),X(IFTR))
      XBOT=AMIN1(XBOT,X(IBMR),X(IFTR))
      YTOP=AMAX1(YTOP,Y(IBMR),Y(IFTR))
      YBOT=AMIN1(YBOT,Y(IBMR),Y(IFTR))
   10 RETURN
      END
```

```
$IBFTC CKWEP    XR7,DECK
      SUBROUTINE CKWEAP
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NM(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GRIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON/DBLLNC / SIGMIS(6,2),KOUNTR(6,2),NUMIS(6,2),TAUMIS(6,2)
     *,TFIREI(25),MISTPI(25),IAFIRI(25),TFIREP(25,10),MISTPP(25,10),IAFI
     *RP(25,10),LCAP,LLITL
      COMMON/GRIDSO/GAMMA,ALPHA(2),R
      COMMON/EXEC/RSMALL,RHO,IFTR,IBMR,FM(2),BSMALL(2)
      COMMON/INIFLT/T,TPRT,BETDOT(2),ENVSW(6,2),
     1IST(2),V(2),II,JJ,TMIS(6,2),PHI(2),Y(2),X(2),
     2DX(2),DY(2),TAWARE,G(2),LEFTSW
      COMMON/COMBA/I,J
      COMMON /RMISR/ RMIS,RMISP
      COMMON /CKWEA/ SIG(2),ANS(24,2)
     *,MIS
      COMMON/RIFFCK/ITEMP(2)
      COMMON/REARGN/RANGUN(2),ALPGUN(2)
      COMMON/CORPUS/OXDERT(2),BINDEX(2),GB(2),SICKF(2),OX(2)
      COMMON /RADAR/ P1(2),TAU(2),P2(2),S(6,2),R1,RPHIJ,NUIND
      COMMON/MULFIR/TLASTF(6,2)
      COMMON/FIRWRT/RANFIR(25),ALPFIR(25),PHIFIR(25)
      COMMON/NOWFIR/ICAN(2),RNOW(2)
      COMMON/RVARY/KCAPP,KLITLP,IATRPR(6,2),TTHERE(2),MISRPR(6,2),TOLER,
     *RPRIME(6,2)
      DIMENSION RFLOOR(6)
      LOGICAL ENVSW
```

```
      DATA PI/3.1415927/
      RTD=57.2957795
      RFLOOR(1)=3000.
      RFLOOR(2)=4000.
      IF (RDET(I).NE.0.) GO TO 5
      IF (R.GE.ROPT(I)) GO TO 11
      IF (ABS(ALPHA(I)).GE.ALPOPT(I)) GO TO 11
      GO TO 6
    5 CONTINUE
      IF (R.GT.RDET(I)) GO TO 11
      IF (ABS(ALPHA(I)).GT.ALPDET(I)) GO TO 11
    6 CONTINUE
      IF (ITEMP(I).EQ.1) GO TO 40
      IF (RIFF(I).LE.R) RETURN
      IF(ALPIFF(I).LE.ABS(ALPHA(I))) RETURN
      ITEMP(I)=1
   40 CONTINUE
      IF(ICAN(I).EQ.1) RETURN
      IF ((R.GT.RTRK(I)).OR.(ABS(ALPHA(I)).GT.ALPTRK(I))) GO TO 11
      MIS=1
   10 IF(ALPMIS(MIS,I).LE.ABS(ALPHA(I))) GO TO 20
      IF(GMIS(MIS,I).LE.G(I)) GO TO 20
      IF(KOUNTR(MIS,I).GE.NUMIS(MIS,I)) GO TO 20
      IF (MIS.EQ.NM(I)) GO TO 19
      SIG(I)=SGN(RETDOT(J))*PHI(I)
      CALL RMISRP
      IF ((MIS.LT.3).AND.(R.LE.RFLOOR(MIS))) GO TO 20
      IF(R.GE.RMIS) GO TO 20
      IF (R.LE.RMISP) GO TO 20
   12 CONTINUE
      TMIS(MIS,I)=TMIS(MIS,I)+DELTAT
      OX(I)=1.-OXDERT(I)
      SIGMIS(MIS,I)=SIGMIS(MIS,I)+DELTAT
      IF(ENVSW(MIS,I)) GO TO 14
      IF (SICKF(I).NE.1.) GO TO 13
      IF (OX(I).LT.EXP((TLASTF(MIS,I)-T)/TAUMIS(MIS,I))) GO TO 20
   13 CONTINUE
```

```
      ENVSW(MIS,I)=.TRUE.
      IMIS=4*MIS-3
      ANS(IMIS,I)=T
      ANS(IMIS+1,I)=R
      ANS(IMIS+2,I)=ALPHA(I)*RTD
      ANS(IMIS+3,I)=PHI(I)*RTD
      P2(J)=1.
      ITEMP(J)=1
      IF(I.NE.IFTR) GOTO 14
      IF(TAWARF.NE.1000.) GOTO 14
      TAWARF=T
      ICAN(J)=0
      RSTAR(1)=RPRIME(2,1)
      RSTAR(2)=RPRIME(2,2)
14    CONTINUE
      TOP=TLASTF(MIS,I)-T
      IF(KOUNTR(MIS,I).EQ.0) GO TO 15
      IF ((T-TLASTF(MIS,I)).LT.TAUMIS(MIS,I)) GO TO 20
15    CONTINUE
      IF (SICKF(I).NE.1.) GO TO 16
      IF (OX(I).LT.EXP(TOP/TAUMIS(MIS,I)))GO TO 20
 16   CONTINUE
      IF(KOUNTR(MIS,I).GE.NUMIS(MIS,I)) GO TO 20
      TLASTF(MIS,I)=T
      KOUNTR(MIS,I)=KOUNTR(MIS,I)+1
      TFIREI(LLITL)=T
      MISTPI(LLITL)=MIS
      IAFIRI(LLITL)=I
      RANFIR(LLITL)=R
      ALPFIR(LLITL)=ALPHA(I)*RTD
      PHIFIR(LLITL)=PHI(I)*RTD
      SIGMIS(MIS,I)=0
      LLITL=LLITL+1
20    IF(MIS.EQ.NM(I)) RETURN
      MIS=MIS+1
      CALL OVER(AGAIN)
      IF(.NOT.AGAIN)GO TO 20
```

```
      CALL RESULT(IGRIDP,IEPS)
      IF (NOWPLT.EQ.0) GO TO 14
      DIMENSION MODE(4),ISYM(4)
      DATA MODE(1),MODE(2),MODE(3),MODE(4),ISYM(1),ISYM(2),ISYM(3),ISYM(
     14)/3,3,4,4,0,0,1,2/
      CALL PREPPT
      IF (IERR.GT.0) GO TO 14
      CALL PLOT(XPLT,YPLT,  2,4,NP,MODE,ISYM  ,XN,XX,YN,YX,1,XLL,XRL,YLL,
     *YUL,1,1,1,12)
 14   CONTINUE
      IF (IGRIDP.EQ.N) GO TO 12
      IGRIDP=IGRIDP+1
      GO TO 10
   12 WRITE(11,100)EPS(IEPS),(TLIT(IP),AWARET(IP),FPR(IP),IP=1,N),
     *((TMISP(MISS,1,NGP),MISS=1,NM1),NGP=1,N),
     *((TMISP(MISS,2,NGP),MISS=1,NM2),NGP=1,N)
      WRITE (11,100) ((TTMIS(MISS,1,NGP),MISS=1,NM1),NGP=1,N),
     *((TTMIS(MISS,2,NGP),MISS=1,NM2),NGP=1,N)
      DO 50 NGP=1,N
      IFZERO=0
      DO 45 IWRIT=1,LCAP
      IF(TFIREP(IWRIT,NGP).GT.0.) GO TO 44
      IF(IFZERO.NE.0) GO TO 46
      IFZERO=1
   44 WRITE (11,200)NGP,TFIREP(IWRIT,NGP),MISTPP(IWRIT,NGP),
     *IAFIRP(IWRIT,NGP),IFZERO
   45 CONTINUE
   46 CONTINUE
   50 CONTINUE
      WRITE (11,100) (SICTIM(1,NGP),SICTIM(2,NGP),NGP=1,N)
  200 FORMAT(1H ,I5,F15.7,3I10)
 100  FORMAT(1H0,4F16.8/(17X,3F16.8))
      RETURN
      END
```

```
$IBFTC RMIS    XR7,DECK
      SUBROUTINE RMISRP
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NM(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON/INIFLT/T,TPRT,BETDOT(2),ENVSW(6,2),
     1IST(2),V(2),II,JJ,TMIS(6,2),PHI(2),Y(2),X(2),
     2DX(2),DY(2),TAWARE,G(2),LEFTSW
      COMMON/COMBA/I,J
      COMMON /RMISP/ RMIS,RMISP
      COMMON /CKWEA/ SIG(2),ANS(24,2)
     *,MIS
      ISIG=1
      DIFF=ABS(SIG(I)-SIGTAB(1))
      DO 10 LIST=2,NSIGMA
      IF(ABS(SIG(I)-SIGTAB(LIST)).GE.DIFF) GO TO 10
      ISIG=LIST
      DIFF=ABS(SIG(I)-SIGTAB(LIST))
10    CONTINUE
      VTEST=(V(I)+V(J))/2.
      DO 20 LIST=2,NVA
      IF (VATAB(LIST).LT.VTEST) GO TO 20
      IVAP=LIST
      IVAM=LIST-1
      GO TO 21
20    CONTINUE
      IVAP=NVA
      IVAM=NVA-1
```

```
21    TEST=SQRT(G(J)**2-1.)
      IF(ABS(TEST-GT(1)).LE..001) GOTO 50
      IF (TEST.LT.GT(1)) GO TO 25
      IF(TEST.LE.GT(2)) GO TO 31
      IF(TEST.LE.GT(3)) GO TO 26
25    IF(TEST.LT.GT(1))GO TO 30
      TEST=GT(3)
26    GL=GT(3)
      GK=GT(2)
      RFKVAP=RF2T(IVAP,ISIG,MIS,I)
      RFKVAM=RF2T(IVAM,ISIG,MIS,I)
      RFLVAP=RF3T(IVAP,ISIG,MIS,I)
      RFLVAM=RF3T(IVAM,ISIG,MIS,I)
      RFKPVP=RF2PT(IVAP,ISIG,MIS,I)
      RFKPVM=RF2PT(IVAM,ISIG,MIS,I)
      RFLPVP=RF3PT(IVAP,ISIG,MIS,I)
      RFLPVM=RF3PT(IVAM,ISIG,MIS,I)
      GO TO 40
30    TEST=GT(1)
31    GK=GT(1)
      GL=GT(2)
      RFKVAP=RF1T(IVAP,ISIG,MIS,I)
      RFKVAM=RF1T(IVAM,ISIG,MIS,I)
      RFLVAP=RF2T(IVAP,ISIG,MIS,I)
      RFLVAM=RF2T(IVAM,ISIG,MIS,I)
      RFKPVP=RF1PT(IVAP,ISIG,MIS,I)
      RFKPVM=RF1PT(IVAM,ISIG,MIS,I)
      RFLPVP=RF2PT(IVAP,ISIG,MIS,I)
      RFLPVM=RF2PT(IVAM,ISIG,MIS,I)
40    DIV=(VTEST-VATAB(IVAM))/(VATAB(IVAP)-VATAB(IVAM))
      ZK=RFKVAM-DIV*(RFKVAM-RFKVAP)
      ZKP=RFKPVM-DIV*(RFKPVM-RFKPVP)
      ZL=RFLVAM-DIV*(RFLVAM-RFLVAP)
      ZLP=RFLPVM-DIV*(RFLPVM-RFLPVP)
      RMIS=ZK-((TEST-GK)/(GL-GK))*(ZK-ZL)
      RMISP=ZKP-((TEST-GK)/(GL-GK))*(ZKP-ZLP)
      RETURN
```

```
50    CONTINUE
      ARGP=RFIT(IVAP,ISIG,MIS,I)
      ARGM=RFIT(IVAM,ISIG,MIS,I)
      DIV=(VTEST-VATAB(IVAM))/(VATAB(IVAP)-VATAB(IVAM))
      RMIS=ARGM+(ARGP-ARGM)*DIV
      ARGM=RFIPT(IVAM,ISIG,MIS,I)
      ARGP=RFIPT(IVAP,ISIG,MIS,I)
      RMISP=ARGM-(ARGP-ARGM)*DIV
      RETURN
      END
```

```
$IBFTC ADVCOR  XR7,DECK
      SUBROUTINE ADRELC
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NM(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON/PURSUE/A(2),BETMAX(2),THEDOT,PHISTR,ETA,ALPMAX(2),XLAMDA(2)
     *,XPHI
      COMMON/ADRECO/RDOT,PHIDOT(2),ALPDOT(2),GPUR
      COMMON/GRIDSO/GAMMA,ALPHA(2),R
      COMMON/INIFLT/T,TPRT,BETDOT(2),ENVSW(6,2),
     1IST(2),V(2),II,JJ,TMIS(6,2),PHI(2),Y(2),X(2),
     2DX(2),DY(2),TAWARE,G(2),LEFTSW
      COMMON/EXEC/RSMALL,RHO,IFTR,IBMR,FM(2),BSMALL(2)
      COMMON/TINCOO/XP(2),YP(2),BETA(2)
      COMMON/TOTIME/TDELTS(3),RTEST(2)
      COMMON/TOV/ISHIFT
      DATA PI/3.14159265/
      IF (DELTAT.EQ.TDELTS(2)) GO TO 5
      IF (DELTAT.EQ.TDELTS(3)) GO TO 6
      IF (R.GT.RTEST(1)) GO TO 6
      DELTAT=TDELTS(2)
      ISHIFT=2
    5 CONTINUE
      IF(R.LE.RTEST(2)) ISHIFT=3
      IF (R.LE.RTEST(2)) DELTAT=TDELTS(3)
    6 CONTINUE
      T=T+DELTAT
      IF (R.LE.(V(IBMR)+V(IFTR))*DELTAT) GO TO 100
```

```
      RDOT=V(IBMR)*COS(PHI(IFTR))-V(IFTR)*COS(ALPHA(IFTR))
      THEDOT=(V(IFTR)*SIN(ALPHA(IFTR))-V(IBMR)*SIN(PHI(IFTR)))/R
      ALPDOT(IFTR)=THEDOT-BETDOT(IFTR)
      PHIDOT(IFTR)=THEDOT-BETDOT(IBMR)
      R=R+RDOT*DELTAT
      ALPHA(IFTR)=PV(ALPHA(IFTR)+ALPDOT(IFTR)*DELTAT)
      PHI(IFTR)=PV(PHI(IFTR)+PHIDOT(IFTR)*DELTAT)
      BETA(IFTR)=PV(BETA(IFTR)+BETDOT(IFTR)*DELTAT)
      BETA(IBMR)=PV(BETA(IBMR)+BETDOT(IBMR)*DELTAT)
      PHI(IBMR)=ALPHA(IFTR)-PI*SGN(ALPHA(IFTR))
      ALPHA(IBMR)=PHI(IFTR)-PI*SGN(PHI(IFTR))
      RETURN
100   CONTINUE
      BETA(IFTR)=PV(BETA(IFTR)+BETDOT(IFTR)*DELTAT)
      BETA(IBMR)=PV(BETA(IBMR)+BETDOT(IBMR)*DELTAT)
      R=SQRT((X(IBMR)-X(IFTR)+V(IFTR)*DELTAT*COS(BETA(IFTR))-V(IBMR)*
     *DELTAT*COS(BETA(IBMR)))**2+(Y(IBMR)-Y(IFTR)+V(IBMR)*DELTAT*SIN(
     *BETA(IBMR))-V(IFTR)*DELTAT*SIN(BETA(IFTR)))**2)
      XX=X(IFTR)-X(IBMR)-V(IFTR)*DELTAT*COS(BETA(IFTR))+V(IBMR)*DELTAT*
     *COS(BETA(IBMR))
      YY=Y(IBMR)-Y(IFTR)+V(IBMR)*DELTAT*SIN(BETA(IBMR))-V(IFTR)*DELTAT*
     *SIN(BETA(IFTR))
      ALPHA(IFTR)=ATAN2(YY,XX) -BETA(IFTR)
      ALPHA(IFTR)=PV(ALPHA(IFTR))
      PHI(IFTR)   = PV( ALPHA(IFTR) + BETA(IFTR) - BETA(IBMR))
      PHI(IBMR)   = ALPHA(IFTR) - PI*SGN(ALPHA(IFTR))
      ALPHA(IBMR)   = PHI(IFTR) - PI*SGN(PHI(IFTR))
      RETURN
      END
```

```
$IBFTC ACTIV    XR7,DECK
      SUBROUTINE ACTIVE(ACT)
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NW(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON/GRIDSO/GAMMA,ALPHA(2),R
      COMMON/COMBA/I,J
      LOGICAL ACT
      IF((R.LE.RDET(I)).AND.(ABS(ALPHA(I)).LE.ALPDET(I)))GO TO 10
      IF((R.LE.ROPT(I)).AND.(ABS(ALPHA(I)).LE.ALPOPT(I)))GO TO 10
      ACT=.FALSE.
      RETURN
   10 CALL AWARE
      ACT=.TRUE.
      RETURN
      END
```

```
$IBFTC AWARE.  XR7,DECK
      SUBROUTINE AWARE
      COMMON/RVARY/KCAPP,KLITLP,TATRPR(6,2),TTHERE(2),MISRPR(6,2),TOLER,
     *RPPIME(6,2)
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NM(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON/NOWFIR/ICAN(2),RNOW(2)
      COMMON/EXEC/RSMALL,RHO,IFTR,IBMR,FM(2),BSMALL(2)
      COMMON/INIFLT/T,TPRT,BETDOT(2),ENVSW(6,2),
     1IST(2),V(2),II,JJ,TMIS(6,2),PHI(2),Y(2),X(2),
     2DX(2),DY(2),TAWARE,G(2),LEFTSW
      COMMON/COMBA/I,J
      IF(I.NE.IBMR) RETURN
      IF(TAWARE.NE.1000.)RETURN
      TAWARE=T
      RSTAR(1)=RPPIME(2,1)
      RSTAR(2)=RPPIME(2,2)
      ICAN(J)=0
      RETURN
      END
```

```
$IBFTC INFORM  DECK,XR7
      SUBROUTINE INFO
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NY(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON/INIFLT/T,TPRT,BETDOT(2),ENVSW(6,2),
     1IST(2),V(2),II,JJ,TMIS(6,2),PHI(2),Y(2),X(2),
     2DX(2),DY(2),TAWARE,G(2),LEFTSW
      COMMON/EXEC/RSMALL,RHO,IFTR,IPMR,FM(2),BSMALL(2)
      COMMON/COMBA/I,J
      COMMON/DBLLNC / SIGMIS(6,2),KOUNTR(6,2),NUMIS(6,2),TAUMIS(6,2)
     *,TFIREI(25),MISTPI(25),IAFIRI(25),TFIREP(25,10),MISTPP(25,10),IAFI
     *PP(25,10),LCAP,LLITL
      COMMON/BETFIX/BETTAB(27,2),VSTR(2)
      COMMON/BIFFCK/ITEMP(2)
      COMMON/COWARD/VPRIME(2), K   ,D(11,2),GCAP(2)
      ISUMK=0
      ISUMNU=0
      NUMDO=NM(I)
      DO 1 LEAP=1,NUMDO
      ISUMK=ISUMK+KOUNTR(LEAP,I)
      ISUMNU=ISUMNU+NUMIS(LEAP,I)
    1 CONTINUE
      LOGICAL ACT,PASS
      IF ((T.GE.TC(I)).OR.(ISUMK.EQ.ISUMNU)) GOTO 2
      CALL ACTIVE(ACT)
      IF (.NOT.ACT) GO TO 10
      IF(ITEMP(I).EQ.1) GOTO 3
```

```
      CALL PASSIV(PASS)
      K=6
      IF(PASS)  RETURN
      K=4
      RETURN
2     CALL ACTIVE(ACT)
      K=3
      IF(ACT) RETURN
      K=2
      RETURN
3     CALL PASSIV(PASS)
      K=7
      IF (PASS) RETURN
      K=5
      RETURN
10    CALL PASSIV(PASS)
      IF(.NOT.PASS) GOTO 11
      K=9
      IF(ITEMP(I).EQ.1) RETURN
      K=8
      RETURN
11    K=1
      IF((I.EQ.IBMR).AND.(TAWARE.EQ.1000.)) RETURN
      K=10
      BETAMI=(32.2*SQRT(GMAXF(VSTR(I),I)**2-1.))/VSTR(I)
      BETAMJ=(32.2*SQRT(GMAXF(VSTR(J),J)**2-1.))/VSTR(J)
      IF((BETAMI.GT.BETAMJ).AND.(ITEMP(I).NE.1)) RETURN
      K=11
      RETURN
      END
```

```
$IBFTC BDOTFN  DECK,XR7
      FUNCTION BFUNCT(GARG,VARG,INDEX)
      BFUNCT=0.
      IF (GARG.LE.1.) RETURN
      BFUNCT=32.2/VARG*SQRT(GARG**2-1.)
      RETURN
      END
```

```
$IBFTC OVER1   XR7,DECK
      SUBROUTINE OVER(AGAIN)
      COMMON/GRIDSO/GAMMA,ALPHA(2),R
      COMMON /OVERR/TLAST
      COMMON/INIFLT/T,TPRT,BETDOT(2),ENVSW(6,2),
     1IST(2),V(2),II,JJ,TMIS(6,2),PHI(2),Y(2),X(2),
     2DX(2),DY(2),TAWARE,G(2),LEFTSW
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NM(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON/EXEC/RSMALL,RHO,IFTR,IBMR,FM(2),BSMALL(2)
      COMMON/GRAPHO/XTOP,XBOT,YTOP,YBOT
      LOGICAL AGAIN
      IF (MOD(IST(IFTR),2).EQ.0) GO TO 5
      IF (MOD(IST(IBMR),2).NE.0) GO TO 20
    5 CONTINUE
      IF(T.GT.TMAX) GO TO 10
   11 CONTINUE
      IF (R.LT.RMIN) GO TO 10
      AGAIN=.FALSE.
      RETURN
 10   TLAST=T
      AGAIN=.TRUE.
      WRITE (6,100) IST(IFTR),IST(IBMR),R
 100  FORMAT(1H0,4HFTR=,I2,10X,4HBMR=,I2,10X,2HR=, E15.7)
      WRITE (6,200) XBOT,XTOP,YBOT,YTOP
 200  FORMAT (1H0,8HX RANGE(,F8.0,1H,,F8.0,11H)  Y RANGE(,
     *F8.0,1H,,F8.0,1H))
```

```
      RETURN
20    IF(T.GT.TMIN) GO TO 10
      GO TO 11
      END
```

```
$IBFTC GLARGE  DECK,XR7
      FUNCTION GCAPF(V,I)
      COMMON/CORPUS/OXDEBT(2),BINDEX(2),GB(2),SICKF(2),OX(2)
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NM(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      NVADO=NVAG(I)
      DO 1 I1=1,NVADO
      J1=I1
      IF(VATAG(I1,I)-V) 1,6,2
1     CONTINUE
      GCAPF=GBIG(NVADO,I)
      GO TO 10
6     GCAPF=GBIG(J1,I)
      GOTO 10
2     IF (J1.GT.1) GOTO 4
      GCAPF=GBIG(1,I)
      GOTO 10
4     GCAPF=GBIG(J1-1,I)+(GBIG(J1,I)-GBIG(J1-1,I))*
     1(V-VATAG(J1-1,I))/(VATAG(J1,I)-VATAG(J1-1,I))
10    GCAPF=AMIN1(GCAPF,GP(I))
      IF (BINDEX(I).NE.1.) RETURN
      IF (GB(I).NE.1.) RETURN
      GCAPF=AMIN1(GCAPF,1.5)
      RETURN
      END
```

```
$IBFTC GFIND   XR7,DECK
      SUBROUTINE FINDG
      COMMON/PURSUE/A(2),BETMAX(2),THEDOT,PHISTR,ETA,ALPMAX(2),XLAMDA(2)
     *,XPHI
      COMMON/CORPUS/OXDEBT(2),BINDEX(2),GB(2),SICKF(2),OX(2)
      COMMON/COMBA/I,J
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NM(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON/ADRECO/RDOT,PHIDOT(2),ALPDOT(2),GPUR
      COMMON/INIFLT/T,TPRT,BETDOT(2),ENVSW(6,2),
     1IST(2),V(2),II,JJ,TMIS(6,2),PHI(2),Y(2),X(2),
     2DX(2),DY(2),TAWARE,G(2),LEFTSW
      COMMON/SIGLV/TIMSIC(2),SICTIM(2,25)
      G(I)=SQRT(((V(I)*BETDOT(I))/32.2)**2+1.)
      IF ((G(I).GT.4.).AND.(G(I).LT.8.))GO TO 10
      IF ((BINDEX(I).EQ.1.).AND.(GB(I).EQ.1.)) GO TO 5
      IF (G(I).LE.1.5001) GO TO 5
      IF (OXDEBT(I).GE.1.) TIMSIC(I)=TIMSIC(I)+DELTAT
      RETURN
 5    OXDEBT(I)=OXDEBT(I)-.0167*DELTAT
      IF (OXDEBT(I).GE.1.) TIMSIC(I)=TIMSIC(I)+DELTAT
      IF (OXDEBT(I).GT.0.) RETURN
      BINDEX(I)=0.
      OXDEBT(I)=0.
      RETURN
 10   OXDEBT(I)=OXDEBT(I)+DELTAT/(-23.*ALOG(.25*G(I)-1.))
      IF (OXDEBT(I).LT.1.) RETURN
```

```
TIMSIC(I)=TIMSIC(I)+DELTAT
BINDEX(I)=1.
RETURN
END
```

```
*IPETC GMAX1    XR7,DECK
      FUNCTION GMAXF(V,I)
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NM(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON/CORPUS/OXDEBT(2),BINDEX(2),GB(2),SICKF(2),OX(2)
      NVADO=NVAG(I)
      DO 1 I1=1,NVADO
      J1=I1
      IF(VATAG(I1,I)-V)1,6,2
 1    CONTINUE
      GMAXF=GMAXT(NVADO,I)
      GO TO 10
 6    GMAXF=GMAXT(J1,I)
      GO TO 10
 2    IF(J1.GT.1)GO TO 4
      GMAXF=GMAXT(1,I)
      GO TO 10
 4    GMAXF=GMAXT(J1-1,I)+(GMAXT(J1,I)-GMAXT(J1-1,I))*
     1(V-VATAG(J1-1,I))/(VATAG(J1,I)-VATAG(J1-1,I))
 10   GMAXF=AMIN1(GMAXF,GP(I))
      IF (BINDEX(I).NE.1.) RETURN
      IF (GB(I).NE.1.) RETURN
      GMAXF=AMIN1(GMAXF,1.5)
      RETURN
      END
```

```
$IBFTC INERCO   XR7,DECK
       SUBROUTINE TINERC
       COMMON/EXEC/RSMALL,RHO,IFTR,IBMR,FM(2),BSMALL(2)
       COMMON/PURSUE/A(2),BETMAX(2),THEDOT,PHISTR,ETA,ALPMAX(2),XLAMDA(2)
      *,XPHI
       COMMON/TINCOO/XP(2),YP(2),BETA(2)
       COMMON/COMBA/I,J
       COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
      1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
      *N,NM(2),IPS(2),NID,
      1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
      2ADEC(2),RDET(2),RIFF(2),POPT(2),
      3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
      4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
      5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
      6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
      7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
      8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
      9NUGRC(2)
       COMMON/GRIDSO/GAMMA,ALPHA(2),R
       COMMON/INIFLT/T,TPRT,BETDOT(2),ENVSW(6,2),
      1IST(2),V(2),II,JJ,TMIS(6,2),PHI(2),Y(2),X(2),
      2DX(2),DY(2),TAWARE,G(2),LEFTSW
       COMMON/GRIDP/VSTAR,FMU,THETB,THETC,B,C,
      1FMAX,FMIN,QB,QC,DELTF,FSMALL,F
       COMMON/NAVSY/GLAT(2),VZERO,S,ASMALL,RC1,
      1FLAMDA,M,XC(2),YC(2)
       DATA PI/3.14159265/
       X(IBMR)=X(IBMR)-V(IBMR)*DELTAT*COS(BETA(IBMR))
       Y(IBMR)=Y(IBMR)+V(IBMR)*DELTAT*SIN(BETA(IBMR))
       X(IFTR)=X(IBMR)+R*COS(PHI(IFTR)+BETA(IBMR))
       Y(IFTR)=Y(IBMR)-R*SIN(PHI(IFTR)+BETA(IBMR))
       V(1)=V(1)+A(1)*DELTAT
       V(2)=V(2)+A(2)*DELTAT
       RETURN
       END
```

```
$IBFTC IQUIT    DECK,XR7
      FUNCTION REFU(VARG,BETARG,I)
      COMMON/COWARD/VPRIME(2), K   ,D(11,2),GCAP(2)
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NM(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      GARG=SQRT(1.+((VARG*BETARG)/32.2)**2)
      NVADO=NVAG(I)
      NUMGDO=NUGRC(I)
      DO 2 LIST=2,NVADO
      IF(VATAG(LIST,I).LT.VARG) GOTO 2
      IVAPL=LIST
      IVAMI=LIST-1
      GOTO 3
 2    CONTINUE
      IVAPL=NVADO
      IVAMI=NVADO-1
 3    DO 4 LIST=2,NUMGDO
      IF(GOFRC(LIST,I).LT.GARG) GOTO 4
      IGPL=LIST
      IGMI=LIST-1
      GOTO 5
 4    CONTINUE
      IGPL=NUMGDO
      IGMI=NUMGDO-1
 5    TEMPP=RC(IVAMI,IGMI,I)+(RC(IVAMI,IGPL,I)-RC(IVAMI,IGMI,I))
     **(GARG-GOFRC(IGMI,I))/(GOFRC(IGPL,I)-GOFRC(IGMI,I))
      TEMPPP=RC(IVAPL,IGMI,I)+(RC(IVAPL,IGPL,I)-RC(IVAPL,IGMI,I))*
```

```
     *(GARG-GOFRC(IGMI,I))/(GOFRC(IGPL,I)-GOFRC(IGMI,I))
      PEEU=TEMPP+(TEMPPP-TEMPP)*(VARG-VATAG(IVAMI,I))/
     *(VATAG(IVAPL,I)-VATAG(IVAMI,I))
      RETURN
      END
```

```
$IBFTC BLK      XR7,DECK
       BLOCK DATA
       COMMON/EXEC/RSMALL,RHO,IFTR,IBMR,FM(2),BSMALL(2)
       DATA RSMALL,RHO/0.,0./
       COMMON/INIFLT/T,TPRT,BETDOT(2),ENVSW(6,2),
      1IST(2),V(2),II,JJ,TMIS(6,2),PHI(2),Y(2),X(2),
      2DX(2),DY(2),TAWARE,G(2),LEFTSW
       DATA T,TPRT,BETDOT(1),BETDOT(2),V(1),V(2),PHI(1),PHI(2),Y(1),Y(2),
      *X(1),X(2),DX(1),DX(2),DY(1),DY(2)/16*0./
       COMMON /ADRECO/RDOT,PHIDOT(2),ALPDOT(2),GPUR
       DATA RDOT,PHIDOT(1),PHIDOT(2),ALPDOT(2),GPUR/5*0./
       COMMON /TINCOO/XP(2),YP(2),BETA(2)
       DATA XP(1),XP(2),YP(1),YP(2),BETA(1),BETA(2)/6*0./
       COMMON /GPIDSQ/GAMMA,ALPHA(2),R
       DATA GAMMA,ALPHA(1),ALPHA(2),R/4*0./
       COMMON /NAVSY/GLAT(2),VZERO,S,ASMALL,RC,FLAMDA,M,XC(2),YC(2)
       DATA GLAT(1),GLAT(2),VZERO,S,ASMALL,RC,FLAMDA,XC(1),XC(2),YC(1),YC
      1(2)
      */11*0./
       END
```

```
$IBFTC GRID    XR7,DECK
      SUBROUTINE GRIDPR
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NM(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON/GRIDP/VSTAR,FMU,THETB,THETC,B,C,
     1FMAX,FMIN,QB,QC,DELTF,FSMALL,F
      COMMON/EXEC/RSMALL,RHO,IFTR,IBMR,FM(2),BSMALL(2)
      COMMON /COMP/ ICOMP
      COMMON/GRIDSO/GAMMA,ALPHA(2),P
      IF  (ICOMP.EQ.0) GO TO 1
      XN=N
      DELTF=(2.*RHO)/ XN
      GAMMA=RHO+.5*DELTF
      RETURN
    1 CONTINUE
      VSTAR=SQRT(VZ(IFTR)**2+VZ(IBMR)**2-2.*VZ(IFTR)*
     *VZ(IBMR)*COS(EPSLON))
      IF (VSTAR.EQ.0.) GO TO 10
      FMU=ACOS((VZ(IFTR)-VZ(IBMR)*COS(EPSLON))/VSTAR)
      GO TO 11
   10 FMU=0.
   11 CONTINUE
      IF(EPSLON.LE.0.) FMU=-FMU
      THETB=PV(FMU+RHO)
      THETC=PV(FMU-RHO)
      B=RSMALL*XSIN(THETB)
      C=RSMALL*XSIN(THETC)
```

```
      FMAX=RSMALL
      FMIN=-FMAX
      QB=Q(THETB)
      QC=Q(THETC)
      IF(QB.EQ.4.) GO TO 250
      IF(QB.EQ.3.) GO TO 240
      IF(QB.EQ.2.) GO TO 230
      IF(QC.NE.1.) GO TO 200
      IF(THETC.GE.THETB) GO TO 260
      FMAX=B
      FMIN=0.
      B=0.
      GO TO 260
200   IF(QC.NE.2.) GO TO 205
      IF(B.GT.C) GO TO 201
      FMAX=C
      GO TO 260
201   FMAX=B
      GO TO 260
205   IF(QC.EQ.3.) GO TO 201
      FMAX=B
      FMIN=C
      GO TO 260
230   IF(QC.EQ.1.) GO TO 231
      IF(QC.EQ.2.) GO TO 232
      IF(QC.EQ.3.) GO TO 260
      FMIN=C
      GO TO 260
231   FMIN=0.
      B=0.
      GO TO 260
232   IF (B.GT.C) GO TO 260
      FMAX=C
      FMIN=0.
      B=0.
      GO TO 260
240   IF(QC.EQ.1.) GO TO 241
```

```
      IF(QC.EQ.2.) GO TC 242
      IF(QC.EQ.3.) GO TO 243
      IF(B.GT.C) GO TO 244
      FMIN=B
      GO TO 260
241   FMIN=B
      GO TO 260
242   FMAX=C
      FMIN=B
      GO TO 260
243   IF(B.GT.C) GO TO 260
      FMAX=0.
      FMIN=B
      C=0.
      GO TO 260
244   FMIN=C
      GO TO 260
250   IF(QC.EQ.1.) GO TO 260
      IF(QC.EQ.2.) GO TO 251
      IF(QC.EQ.3.) GO TO 252
      IF(B.LE.C) GO TO 260
      FMAX=0.
      FMIN=C
      C=0.
      GO TO 260
251   FMAX=C
      GO TO 260
252   FMAX=0.
      C=0.
260   DFLTF=(FMAX-FMIN)/FLOAT(N)
      FSMALL=FMAX+.5*DFLTF
      F=(FMAX-FMIN)*VSTAR/VZ(IBMR)
      RETURN
      END
```

```
$IBFTC PLOPRP  XR7,DECK
      SUBROUTINE PREPPT
      COMMON /PLTINF/
     *      EPSPLT(5),IGRPLT(5),PLOTIM,NOWPLT,KOUNT,XPLT(300,4),YPLT(300
     *,4),XN,XX,YN,YX,XLL,XRL,YLL,YUL,NP(4),IPLOT,TIMPLT,IERR
      XMAX=XPLT(1,1)
      YMAX=YPLT(1,1)
      DO 4 I=1,KOUNT
      DO 4 J=1,2
      IF (XMAX.GE.XPLT(I,J)) GO TO 3
      XMAX=XPLT(I,J)
    3 IF (YMAX.GE.YPLT(I,J)) GO TO 4
      YMAX=YPLT(I,J)
    4 CONTINUE
      IF (XMIN.LT.0.) GO TO 5
      XMIN=0.
      LXMIN=0
      GO TO 6
    5 LXMIN=XMIN-1.
      XMIN=LXMIN
    6 IF (YMIN.LT.0.) GO TO 7
      YMIN=0.
      LYMIN=0
      GO TO 8
    7 LYMIN=YMIN-1.
      YMIN=LYMIN
    8 LXMAX=XMAX+1.
      XMAX=LXMAX
      LYMAX=YMAX+1.
      YMAX=LYMAX
      XRANGE=XMAX-XMIN
      YRANGE=YMAX-YMIN
      IF (XRANGE.LE.9.) GO TO 50
      IF (XRANGE.LE.18.) GO TO 100
      IF (XRANGE.LE.36.) GO TO 200
      IF (XRANGE.GT.180.) GO TO 2000
   40 IF ((LXMIN/10)*10.EQ.LXMIN) GO TO 41
```

```
      LXMIN=LXMIN-1
      GO TO 40
41    IF ((LXMAX/10)*10.EQ.LXMAX) GO TO 42
      LXMAX=LXMAX+1
      GO TO 41
42    IF ((LYMIN/10)*10 .EQ.LYMIN) GO TO 43
      LYMIN=LYMIN-1
      GO TO 42
43    IF ((LYMAX/10)*10 .EQ.LYMAX) GO TO 44
      LYMAX=LYMAX+1
      GO TO 43
44    XMAX=LXMAX
      XMIN=LXMIN
      YMAX=LYMAX
      YMIN=LYMIN
      IF ((YMAX-YMIN).GT.250.) GO TO 2000
      IF ((XMAX-XMIN).GT.180.) GO TO 2000
      LXMAX=LXMAX+180-(LXMAX-LXMIN)
      LYMAX=LYMAX+250-(LYMAX-LYMIN)
      XMAX=LXMAX
      YMAX=LYMAX
      GO TO 500
50    IF (YRANGF.GT.12.) GO TO 100
C     SCALE=.5NM/CM
      LXMAX=LXMAX+9-(LXMAX-LXMIN)
      LYMAX=LYMAX+12-(LYMAX-LYMIN)
      XMAX=LXMAX
      YMAX=LYMAX
      YMAX=YMAX+.5
      GO TO 500
100   IF (YRANGF.GT.25.) GO TO 200
C     SCALE=1NM/CM
      LXMAX=LXMAX+18-(LXMAX-LXMIN)
      LYMAX=LYMAX+25-(LYMAX-LYMIN)
      XMAX=LXMAX
      YMAX=LYMAX
      GO TO 500
```

```
200   IF (YRANGE.GT.50.) GO TO40
      IF ((LXMIN/2)*2.NE.LXMIN) LXMIN=LXMIN-1
      IF ((LXMAX/2)*2.NE.LXMAX) LXMAX=LXMAX+1
      XMAX=LXMAX
      XMIN=LXMIN
      IF ((XMAX-XMIN).GT.36.) GO TO 40
      IF ((LYMIN/2)*2.NE.LYMIN) LYMIN=LYMIN-1
      IF ((LYMAX/2)*2.NE.LYMAX) LYMAX=LYMAX+1
      YMAX=LYMAX
      YMIN=LYMIN
      IF ((YMAX-YMIN).GT.50.) GO TO 40
      LXMAX=LXMAX+36-(LXMAX-LXMIN)
      LYMAX=LYMAX+50-(LYMAX-LYMIN)
      XMAX=LXMAX
      YMAX=LYMAX
500   WRITE (6,150) XMAX,XMIN,YMAX,YMIN
C     DRAW AXIS
      DIMENSION XAXIS(2,2),YAXIS(2,2),NPAX(2),MODAX(2),ISYMAX(2)
      IERR=0
      XAXIS(1,1)=0.
      XAXIS(2,1)=0.
      YAXIS(1,1)=YMIN
      YAXIS(2,1)=YMAX
      XAXIS(1,2)=XMIN
      XAXIS(2,2)=XMAX
      YAXIS(1,2)=0.
      YAXIS(2,2)=0.
      DATA NPAX(1),NPAX(2),MODAX(1),MODAX(2),ISYMAX(1),ISYMAX(2)
     */2,2,3,3,1,1/
      CALL PLOT(XAXIS,YAXIS,2,2,NPAX,MODAX,ISYMAX,XMIN,XMAX,
     *YMIN,YMAX,1.0,18.,0.,25.,1,0,0,12)
      J=0
      DO 180 I1=1,KOUNT,10
      J=J+1
      XPLT(J,3)=XPLT(I1,1)
      YPLT(J,4)=XPLT(I1,2)
      YPLT(J,3)=YPLT(I1,1)
```

```
      YPLT(J,4)=YPLT(I1,2)
180   CONTINUE
      NP(3)=J
      NP(4)=J
      YN=YMIN
      YX=YMAX
      XN=XMIN
      XX=XMAX
      XLL=0.
      XRL=18.
      YLL=0.
      YUL=25.
      RETURN
2000  IERR=1
      WRITE (6,182)
182   FORMAT(1H0,10X,37HPLOT INFORMATION OUT OF PREDETERMINED,
     *6H SCALE)
 150  FORMAT(1H0,20X,16HPLOT INFORMATION/21X,6HMAX X=,E15.7,5X,6HMIN X=,
     *E15.7,5X,6HMAX Y=,E15.7,5X,6HMIN Y=,E15.7)
      WRITE (6,151) XMAX,XMIN,YMAX,YMIN
 151  FORMAT(1H ,20X,6HMAX X=,E15.7,5X,6HMIN X=,E15.7,5X,6HMAX Y=,E15.7
     *,6HMIN Y=,E15.7)
      RETURN
      END
```

```
$IBFTC INPUT1  XR7,DECK
      SUBROUTINE INPUT
      COMMON/TOTIME/TDELTS(3),RTEST(2)
      COMMON/PURSUE/A(2),BETMAX(2),THEDOT,PHISTR,ETA,ALPMAX(2),XLAMDA(2)
      COMMON/INPUTV/TMAX,TMIN,TSTAR,DENS,
     1DELEPS,DELTAT,EPSLON,RANGE,GT(3),RMIN,H,
     *N,NM(2),IPS(2),NIP,
     1TA(2),TB(2),SF(2),SB(2),W(2),WGT(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON /COMP/ ICOMP
      COMMON/BETFIX/BETTAB(27,2),VSTR(2)
      COMMON/COWARD/VPRIME(2), K  ,D(11,2),GCAP(2)
      COMMON /RADAR/ P1(2),TAU(2),P2(2),S(6,2),R1,RPHIJ,NUIND
      COMMON/EXEC/RSMALL,RHO,IFTR,IPMP,FM(2),BSMALL(2)
      DIMENSION VAR(13),IVAR(6),TAB2(50),FNAME(13)
     1,XINAME(5),TANAME(50),TYNAME(3)
     *,LATG(2),INTS(6)
      EQUIVALENCE(TMAX,VAR),(N,IVAR),(TA,TAB2)
      DATA(FNAME(I),I=1,13)/6HTMAX
     *,6HTMIN  ,6HTSTAR ,6HDENS  ,6HDELEPS,6HDELTAT,
     *6HEPSLON,6HRANGE ,6HG1    ,6HG2    ,6HG3    ,6HRMIN  ,6HH     /,
     *(XINAME(I),I=1,5,2)/6HN     ,6HNM    ,6HIPS   /,
     *(TANAME(I),I=1,50,2)/6HTA    ,6HTB    ,6HSF    ,
     *6HSB    ,6HW     ,6HWGT   ,6HADEC  ,6HRDET  ,6HRIFF  ,
     *6HROPT  ,6HRPAS  ,6HRTRK  ,6HVC    ,6HVZ    ,
     *6HRSTAR ,6HALPDET,6HALPIFF,6HALPOPT,6HALPPAS,6HALPTRK,
     *6HVMAX  ,6HGP    ,6HACC   ,6HD2    ,6HD3    /
      DATA (INTS(I),I=1,6)/1,2,3,4,5,6/
      NAME LIST/RADON/TAU,P1,NUIND
      NAME LIST/NUTAB/MORE
```

```
      NAME LIST/GTAB/GMAXT,LATG,VATAG,NVAG,RC,TC,GBIG,
     *GOFRC,NUGRC,D
C     READ 2,3 DIMENSION TABLES
      DATA (TYNAME(I),I=1,3)/6HFLOPT ,6HINVAR ,6HTAB1  /
      READ (5,117) NM(1),NM(2)
  116 FORMAT(19X,I5/(19X,5F10.0))
      DO 180 I=1,2
      J1=NM(I)
  180 READ (5,171) (ALPMIS(J,I),J=1,J1)
      DO 181 I=1,2
      J1=NM(I)
  181 READ (5,171) (GMIS(J,I),J=1,J1)
C     READ TABLES
      READ (5,117) NVA,NSIGMA
      READ (5,171) (VATAB(I),I=1,NVA)
      READ (5,171) (SIGTAB(I),I=1,NSIGMA    )
      READ (5,117)IBMIS
      DO 1544 I=1,2
      J1=NM(I)-1
      DO 1544 J=1,J1
      DO 1544 K=1,NSIGMA
      IF(I.EQ.1) GO TO 154
      IF (IBMIS.EQ.0) GO TO 154
      DO 1541 L=1,NVA
 1541 RFIT(L,K,J,2)=RFIT(L,K,J,1)
      GO TO 1544
  154 READ(5,171)(RFIT(L,K,J,I),L=1,NVA)
 1544 CONTINUE
  171 FORMAT(19X,5F10.0)
      DO 1554 I=1,2
      J1=NM(I)-1
      DO 1554 J=1,J1
      DO 1554 K=1,NSIGMA
      IF(I.EQ.1) GO TO 155
      IF (IBMIS.EQ.0) GO TO 155
      DO 1551 L=1,NVA
 1551 RF2T(L,K,J,2)=RF2T(L,K,J,1)
```

```
      GO TO 1554
155   READ(5,171)(RF2T(L,K,J,I),L=1,NVA)
1554  CONTINUE
      DO 1564 I=1,2
      J1=NM(I)-1
      DO 1564 J=1,J1
      DO 1564 K=1,NSIGMA
      IF (I.EQ.1) GO TO 156
      IF (IBMIS.EQ.0) GO TO 156
      DO 1561 L=1,NVA
1561  RF3T(L,K,J,2)=RF3T(L,K,J,1)
      GO TO 1564
156   READ(5,171)(RF3T(L,K,J,I),L=1,NVA)
1564  CONTINUE
      DO 1574 I=1,2
      J1=NM(I)-1
      DO 1574 J=1,J1
      DO 1574 K=1,NSIGMA
      IF (I.EQ.1) GO TO 157
      IF (IBMIS.EQ.0) GO TO 157
      DO 1571 L=1,NVA
1571  RF1PT(L,K,J,2)=RF1PT(L,K,J,1)
      GO TO 1574
157   READ(5,171)(RF1PT(L,K,J,I),L=1,NVA)
1574  CONTINUE
      DO 1584 I=1,2
      J1=NM(I)-1
      DO 1584 J=1,J1
      DO 1584 K=1,NSIGMA
      IF (I.EQ.1) GO TO 158
      IF (IBMIS.EQ.0) GO TO 158
      DO 1581 L=1,NVA
1581  RF2PT(L,K,J,2)=RF2PT(L,K,J,1)
      GO TO 1584
158   READ(5,171)(RF2PT(L,K,J,I),L=1,NVA)
1584  CONTINUE
      DO 1594 I=1,2
```

```
      J1=NM(I)-1
      DO 1594 J=1,J1
      DO 1594 K=1,NSIGMA
      IF (I.EQ.1) GO TO 159
      IF (IBMIS.EQ.0) GO TO 159
      DO 1591 L=1,NVA
 1591 RE3PT(L,K,J,2)=RE3PT(L,K,J,1)
      GO TO 1594
  159 READ(5,171)(RE3PT(L,K,J,I),L=1,NVA)
 1594 CONTINUE
  117 FORMAT(19X,2I10)
      READ (5,171) RTEST(1),RTEST(2),(TDELTS(M),M=1,3)
      ENTRY INPUT2
      READ (5,115) NID,(XID(I),I=1,NID)
  115 FORMAT(19X,I5/(12A6))
C     SECOND ENTRY READ ID
      READ (5,110) NUTYP
      IF(NUTYP.EQ.0) GO TO 301
      DO 300 M=1,NUTYP
      READ (5,111) TYPE,NUM
  110 FORMAT(19X,I10)
  111 FORMAT(A6,13X,I10)
      DO 200 L=1,3
      IF(TYPE.NE.TYNAME(L))GO TO 200
      NGO=L
      GO TO 201
  200 CONTINUE
      WRITE(6,202)TYPE
  202 FORMAT(1H0,10X,12HNUMBER TYPE ,A6,
     146H IS NOT INCLUDED IN THE ATAC-2 INPUT ROUTINE. ,
     221HPROGRAM DISCONTINUED.)
      STOP
  201 GO TO (205,225,250),NGO
C205  FLOATING PT. VARIABLES
  205 DO 220 L=1,NUM
      READ (5,112) VARNAM,VALUE
  112 FORMAT(A6,13X,E10.0)
```

```
      DO 215 K=1,13
      IF(VARNAM.NE.FNAME(K)) GO TO 215
      VAR(K)=VALUE
      GO TO 220
  215 CONTINUE
      WRITE(6,203)VARNAM,VALUE
  203 FORMAT(1HO,10X,29HTHE FOLLOWING FLOATING POINT ,
     145HVARIABLE IS NOT INCLUDED IN THE ATAC-2 INPUT ,
     28HROUTINE./20X,A6,13X,F8.2/20X,21HPROGRAM DISCONTINUED.)
      STOP
  220 CONTINUE
      GO TO 300
C225  INTEGER VALUES
  225 DO 240 L=1,NUM
      READ (5,113) VARNAM,IVAL1,IVAL2
  113 FORMAT(A6,13X,2I10)
      IF(VARNAM.NE.XINAME(1)) GO TO 227
      IVAR(1)=IVAL1
      GO TO 240
  227 IF(VARNAM.NE.XINAME(2)) GO TO 228
      IVAR(2)=IVAL1
      IVAR(3)=IVAL2
      GO TO 240
  228 IVAR(4)=IVAL1
      IVAR(5)=IVAL2
  240 CONTINUE
      GO TO 300
C250  ONE DIMENSIONAL TABLES
  250 DO 275 L=1,NUM
  114 FORMAT(A6,13X,2E10.0)
      READ (5,114) VARNAM,VAL1,VAL2
      DO 260 K=1,50,2
      IF(VARNAM.NE.TANAME(K)) GO TO 260
      TAB2(K)=VAL1
      TAB2(K+1)=VAL2
      GO TO 275
  260 CONTINUE
```

```
      WRITE(6,204)VARNAM,VAL1,VAL2
  204 FORMAT(1H0,10X,36HTHE FOLLOWING FLT. PT. TABLE IS NOT ,
     137HINCLUDED IN THE ATAC-2 INPUT ROUTINE./20X,
     2A6,13X,2F10.2/20X,21HPROGRAM DISCONTINUED.)
      STOP
  275 CONTINUE
  300 CONTINUE
  301 CONTINUE
      READ (5,110)   ICOMP
C     WRITE OUT INPUT DATA
      WRITE(6,130)
  130 FORMAT(54X,26HINPUT FOR THE ATAC-2 MODEL)
      WRITE (6,119) (XID(I),I=1,NID)
  119 FORMAT(1H0/(30X,12A6))
      WRITE (6,122) XINAME(1),IVAR(1),XINAME(3),IVAR(2),
     *IVAR(3),XINAME(5),IVAR(4),IVAR(5)
  122 FORMAT(1H0,48X,A6,3X,I12/(49X,A6,3X,2I12))
      WRITE(6,120)(FNAME(I),VAR(I),I=1,13)
  120 FORMAT(1H0,48X,A6,3H = ,F12.3/(49X,A6,3H = ,F12.3))
      WRITE (6,121)(TANAME(I),TAB2(I),TAB2(I+1),I=1,46,2)
  121 FORMAT(1H0,48X,A6,3X,2F12.3)
C     WRITE TABLES
      DO 160 I=1,2
      J1=NM(I)-1
      DO 160 J=1,J1
      WRITE (6,123)I,J,(SIGTAB(K),K=1,NSIGMA)
      WRITE(6,80)
   80 FORMAT(3H VA)
      DO 160 L=1,NVA
  160 WRITE (6,124)VATAB(L),(RFIT(L,K,J,I),K=1,NSIGMA)
  123 FORMAT(1H0,58X,13HRF1(VA,SIGMA)/52X,8HA/C NO. I1,5X,
     111HWEAPON NO. ,I1/9H   SIGMAS ,15F8.2)
  124 FORMAT(1H ,16F8.0)
  125 FORMAT(1H0,58X,13HRF2(VA,SIGMA)/52X,8HA/C NO. I1,5X,
     111HWEAPON NO. ,I1/9H   SIGMAS ,15F8.2)
  126 FORMAT(1H0,58X,13HRF3(VA,SIGMA)/52X,8HA/C NO. I1,5X,
     111HWEAPON NO. ,I1/9H   SIGMAS ,15F8.2)
```

```
127   FORMAT(1H0,58X,14HRF1-(VA,SIGMA)/52X,8HA/C NO. I1,5X,
     111HWEAPON NO. ,I1/9H    SIGMAS ,15F8.2)
128   FORMAT(1H0,58X,14HRF2-(VA,SIGMA)/52X,8HA/C NO. I1,5X,
     111HWEAPON NO. ,I1/9H    SIGMAS ,15F8.2)
129   FORMAT(1H0,58X,14HRF3-(VA,SIGMA)/52X,8HA/C NO. I1,5X,
     111HWEAPON NO. ,I1/9H    SIGMAS ,15F8.2)
      DO 161 I=1,2
      J1=NM(I)-1
      DO 161 J=1,J1
      WRITE (6,125) I,J,(SIGTAB(K),K=1,NSIGMA)
      WRITE(6,80)
      DO 161 L=1,NVA
161   WRITE (6,124) VATAB(L),(RF2T(L,K,J,I),K=1,NSIGMA)
      DO 162 I=1,2
      J1=NM(I)-1
      DO 162 J=1,J1
      WRITE (6,126) I,J,(SIGTAB(K),K=1,NSIGMA)
      WRITE(6,80)
      DO 162 L=1,NVA
162   WRITE (6,124) VATAB(L),(RF3T(L,K,J,I),K=1,NSIGMA)
      DO 163 I=1,2
      J1=NM(I)-1
      DO 163 J=1,J1
      WRITE (6,127) I,J,(SIGTAB(K),K=1,NSIGMA)
      WRITE(6,80)
      DO 163 L=1,NVA
163   WRITE (6,124) VATAB(L),(RF1PT(L,K,J,I),K=1,NSIGMA)
      DO 164 I=1,2
      J1=NM(I)-1
      DO 164 J=1,J1
      WRITE (6,128) I,J,(SIGTAB(K),K=1,NSIGMA)
      WRITE(6,80)
      DO 164 L=1,NVA
164   WRITE (6,124) VATAB(L),(RF2PT(L,K,J,I),K=1,NSIGMA)
      DO 165 I=1,2
      J1=NM(I)-1
      DO 165 J=1,J1
```

```
      WRITE (6,129) I,J,(SIGTAB(K),K=1,NSIGMA)
      WRITE(6,80)
      DO 165 L=1,NVA
165   WRITE (6,124) VATAB(L),(RF3PT(L,K,J,I),K=1,NSIGMA)
      READ (5,RADON)
      READ (5,NUTAB)
      IF(MORE.EQ.0) GO TO 350
      READ (5,GTAB)
      DO 500 I=1,2
      NVADO=NVAG(I)
      IF(LATS(I).EQ.0) GO TO 450
      DO 400 J=1,NVADO
400   GMAXT(J,I)=SQRT(GMAXT(J,I)**2+1.)
450   DO 470 J=1,NVADO
      IF (GMAXT(J,I).GT.8.) GMAXT(J,I)=8.
      BETTAB(J,I)=0.
      TEMP=AMIN1(GMAXT(J,I),GBIG(J,I))
      IF (TEMP.LE.1.) GO TO 470
      BETTAB(J,I)=(32.2*SQRT(TEMP**2-1.))/(VATAG(J,I)*W(I))
470   VATAG(J,I)=W(I)*VATAG(J,I)
      VSTR(I)=VATAG(1,I)
      TABMAX=BETTAB(1,I)
      DO 480 J=2,NVADO
      IF (BETTAB(J,I).LE.TABMAX) GO TO 480
      VSTR(I)=VATAG(J,I)
      TABMAX=BETTAB(J,I)
480   CONTINUE
      VPRIME(I)=0.
      TABMAX=0.
      DO 490 J=1,NVADO
      BETG=0.
      IF (GBIG(J,I).GT.1.) BETG=(32.2*SQRT(GBIG(J,I)**2-1.))/VATAG(J,I)
      IF (BETG.LE.TABMAX) GO TO 490
      TABMAX=BETG
      VPRIME(I)=VATAG(J,I)
490   CONTINUE
500   CONTINUE
```

```
350   CONTINUE
 101  FORMAT(1H0,40X,36HENERGY MANUVERABILITY TABLES FOR AC=I2/
     *56X,17HG-S FOR RC TABLES/41X,8F10.3)
 102  FORMAT(1H ,12F10.3)
 103  FORMAT(1H0,3X,2HVA,8X,5HGMAXT,3X,6HBETDOT,5X,4HGBIG,18X,4HRC-S)
      DO 550 I=1,2
      NVADO=NVAG(I)
      NGDO=NUGRC(I)
      WRITE (6,101) I,(GOFRC(J1,I),J1=1,NGDO)
      WRITE (6,103)
      DO 550 I1=1,NVADO
      WRITE (6,102) VATAG(I1,I),GMAXT(I1,I),BETTAB(I1,I),
     *GBIG(I1,I),(RC(I1,J1,I),J1=1,NGDO)
550   CONTINUE
      WRITE (6,574)
      DO 575 I=1,2
      J1=NM(I)
575   WRITE (6,109)I,(ALPMIS(J,I),J=1,J1)
574   FORMAT (1H0,57X,19HALPMIS(A/C,MISSILE))
109   FORMAT(2X,I1,5X,10F12.2)
      WRITE (6,182)
182   FORMAT(1H0,57X,17HGMIS(A/C,MISSILE))
108   FORMAT(2X,I1,6X,6F11.2)
      J1=NM(1)
      IF(NM(1).LT.NM(2))J1=NM(2)
      WRITE(6,106)(INTS(II),II=1,J1)
106   FORMAT(4X,7HMISSILE,6X,6(I1,10X))
      WRITE(6,107)
107   FORMAT(4H A/C)
      DO 10 I=1,2
      J1=NM(I)
10    WRITE(6,108) I,(GMIS(J,I),J=1,J1)
      WRITE (6,105)
105   FORMAT(1H1)
      RETURN
      END
```

```
$IBFTC SEGRID  XR7,DECK
      SUBROUTINE SOGRID
      COMMON/EXEC/RSMALL,RHO,IFTR,IBIR,FM(2),BSMALL(2)
      COMMON/GRIDSO/GAMMA,ALPHA(2),R
      COMMON/GRIDP/VSTAR,FMU,THETB,THETC,B,C,
     1FMAX,FMIN,QB,QC,DELTF,FSMALL,F
      COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
     1DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H,
     *N,NM(2),IPS(2),NID,
     1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
     2ADEC(2),RDET(2),RIFF(2),ROPT(2),
     3RPAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
     4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
     5RFIT(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
     6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
     7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
     8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
     9NUGRC(2)
      COMMON /COMP/ ICOMP
      DATA PI/3.14159265/
      DATA TOL/.000001/
      IF (ICOMP.EQ.0)  GO TO 1
      GAMMA =GAMMA-DELTF
      ALPHA(IFTR)=GAMMA
      R=RSMALL
      GO TO 102
    1 CONTINUE
      FSMALL=FSMALL-DELTF
      GAMMA=ASIN(FSMALL/RSMALL)
      IF((FSMALL-C).GT.TOL) GO TO 100
      IF(FSMALL-B+TOL.LT.0.) GO TO 100
      IF(FSMALL.GT.0.) GO TO 101
      ALPHA(IFTR)=-RHO
      GO TO 102
  100 ALPHA(IFTR)=PV(FMU-GAMMA)
      R=RSMALL
      GO TO 103
```

```
101   ALPHA(IFTR)=RHO
102   R=ABS(FSMALL/XSIN(FMU-ALPHA(IFTR)))
103   ALPHA(IBMR)=ALPHA(IFTR)+EPSLON-SGN(EPSLON)*PI
      RETURN
      END
```

```
$IBFTC SGNFN    XR7,DECK
      FUNCTION SGN(X)
      SGN=1.
      IF(X.LT.0.) SGN=-1.
      RETURN
      END
```

```
$IBFTC PV1      XR7,DECK
      FUNCTION PV(THETA)
      PI=3.14159265
      PV=THETA
      IF(ABS(THETA).GT.PI) GO TO 1
      RETURN
 1    IF(THETA.GT.0.) GO TO 2
      PV=THETA+2.*PI
      RETURN
    2 PV=THETA-2.*PI
      RETURN
      END
```

```
$IBFTC MYSIN   XR7,DECK
      FUNCTION XSIN(X)
      DATA TOL/.00001/
      XSIN=SIN(X)
      IF(ABS(XSIN).GT.TOL) GO TO 1
      XSIN=0.
      RETURN
 1    IF(ABS(ABS(XSIN)-1.).GT.TOL) GO TO 2
      XSIN=SIGN(1.,XSIN)
 2    RETURN
      END
```

```
$IBFTC MYCOS    XR7,DECK
      FUNCTION XCOS(X)
      DATA TOL/.00001/
      XCOS=COS(X)
      IF (ABS(XCOS).GT.TOL) GO TO 1
      XCOS=0.
      RETURN
    1 IF (ABS(ABS(XCOS)-1.).GT.TOL) GO TO 2
      XCOS=SIGN(1.,XCOS)
    2 RETURN
      END
```

```
$IBFTC QI       XR7,DECK
      FUNCTION Q(THETA)
      PI2=1.5707962
      IF (THETA.GT.PI2) GO TO 2
      IF (THETA.GT.0.) GO TO 1
      IF (THETA.GT.-PI2) GO TO 4
      Q=3.
      RETURN
 1    Q=1.
      RETURN
 2    Q=2.
      RETURN
 4    Q=4.
      RETURN
      END
```

```
$IBFTC EPSCA.  XR7,DECK
      SUBROUTINE EPSCAL
      COMMON/EPSCA/PD(73),PU(73),P(2,2,73),PZ(2,73),IGRIDP
      COMMON/EXEC/NEPS,IFTR,IBMR,TMAXBF
     *,TTMAX
      COMMON/COMP/ICOMP
      COMMON/INPUTC/D,RHO,TC(2),M(2),IWK(6,2),
     1PKP(6,2),N,NRUNC,NDPMID,DPMID(72)
      COMMON/PKGCAL/PKL,PKG,PKE,PCCEPN(73,25),PCCEPS(73)
      COMMON/INPUTT/NRUN,NID,XID(72),NTEPS,EPSLON(73),
     1NN,TSMALL(25),TAWARE(25),F(25),T(6,2,25)
      COMMON/GRIDD/I,J,LIMA,LIMB,LIMC,IU,A,PK(6,2)
      COMMON/PJUC/PP(2,2,73,25), PPZ(2,73,25)
      COMMON/SICLV/TIMSIC(2),SICTIM(2,25),SICEPS(2,5)
      DO 1 I2=1,2
      SICEPS(I2,NEPS)=0.
      PZ(I2,NEPS)=0.
      DO 1 J=1,2
    1 P(I2,J,NEPS)=0.
      NU=0
      DO 11 IJ=1,N
      IGRIDP=IJ
      PD(NEPS)=1.
      IF (ICOMP.EQ.0) PD(NEPS)=1.-EXP(-F(IGRIDP)*RHO*D)
      CALL GRIDDC
      SICEPS(1,NEPS)=SICEPS(1,NEPS)+SICTIM(1,NEPS)/TSMALL(NEPS)
      SICEPS(2,NEPS)=SICEPS(2,NEPS)+SICTIM(2,NEPS)/TSMALL(NEPS)
      IF(A.EQ.0.)GO TO 10
      P(2,IBMR,NEPS)=P(2,IBMR,NEPS)+PP(2,IBMR,NEPS,IGRIDP)
      P(1,IBMR,NEPS)=P(1,IBMR,NEPS)+PP(1,IBMR,NEPS,IGRIDP)
      P(1,IFTR,NEPS)=P(1,IFTR,NEPS)+PP(1,IFTR,NEPS,IGRIDP)
      P(2,IFTR,NEPS)=P(2,IFTR,NEPS)+PP(2,IFTR,NEPS,IGRIDP)
      PCCEPS(NEPS)=PCCEPS(NEPS)+PCCEPN(NEPS,IGRIDP)
      GO TO 11
   10 NU=NU+1
      PZ(IBMR,NEPS)=PZ(IBMR,NEPS)+PPZ(IBMR,NEPS,IGRIDP)
   11 CONTINUE
```

```
      PU(NEPS)=FLOAT(NU)/FLOAT(N)
      XN=N
      PCCEPS(NEPS)=PCCEPS(NEPS)/XN
      SICEPS(1,NEPS)=SICEPS(1,NEPS)/XN
      SICEPS(2,NEPS)=SICEPS(2,NEPS)/XN
      IF(NU.EQ.0)GO TO 14
      PZ(IBMR,NEPS)=PZ(IBMR,NEPS)/FLOAT(NU)
      IF(NU.NE.N)GO TO 15
      P(1,IBMR,NEPS)=2.
      P(2,IBMR,NEPS)=2.
      P(1,IFTR,NEPS)=2.
      P(2,IFTR,NEPS)=2.
      RETURN
   14 PZ(IBMR,NEPS)=2.
   15 FNNU=N-NU
      P(1,IBMR,NEPS)=P(1,IBMR,NEPS)/FNNU
      P(2,IBMR,NEPS)=P(2,IBMR,NEPS)/FNNU
      P(1,IFTR,NEPS)=P(1,IFTR,NEPS)/FNNU
      P(2,IFTR,NEPS)=P(2,IFTR,NEPS)/FNNU
      RETURN
      END
```

```
$IBFTC UNCON.   XR7,DECK
      SUBROUTINE UNCOND
      COMMON/EKT1/EKT,YN,EKF
      COMMON/EXEC/NEPS,IFTR,IBMR,TMAXBF
     *,TTMAX
      COMMON/PKGCAL/PKL,PKG,PKE,PCCEPN(73,25),PCCEPS(73)
      COMMON/INPUTT/NRUN,NID,XID(72),NTEPS,EPSLON(73),
     1NN,TSMALL(25),TAWARE(25),F(25),T(6,2,25)
      COMMON/EPSCA/PD(73),PU(73),P(2,2,73),PZ(2,73),IGRIDP
      COMMON/UNCON/PUPD,PDD,PKZB,PKK(2,2)
     1,PKBU,PKD1FB,PKD2FB,PKJIA(2,2)
      COMMON/SICLV/TIMSIC(2),SICTIM(2,25),SICEPS(2,5)
      COMMON /NUOUT/PKB
      COMMON/NUPROB/PKBGDE(25),PKFGDE(25),PKBGD,PKFGD
      NEPSM1=NTEPS-1
      XEPS=NEPSM1
      PUU=.5*(PD(1)*PU(1)+PD(NTEPS)*PU(NTEPS))
      PDD=.5*(PD(1)+PD(NTEPS))
      PKZB=.5*(PD(1)*PU(1)*PZ(IBMR,1)+PD(NTEPS)*PU(NTEPS)
     1*PZ(IBMR,NTEPS))
      PKE=.5*(PD(1)*PCCEPS(1)+PD(NTEPS)*PCCEPS(NTEPS))
      TIMSIC(1)=.5*(PD(1)*SICEPS(1,1)+PD(NTEPS)*SICEPS(1,NTEPS))
      TIMSIC(2)=.5*(PD(1)*SICEPS(2,1)+PD(NTEPS)*SICEPS(2,NTEPS))
      DO 5 I1=1,2
      DO 5 J=1,2
    5 PKK(I1,J)=0.
      DO 10 IEPS=2,NEPSM1
      TIMSIC(1)=TIMSIC(1)+PD(IEPS)*SICEPS(1,IEPS)
      TIMSIC(2)=TIMSIC(2)+PD(IEPS)*SICEPS(2,IEPS)
      PUU=PUU+PD(IEPS)*PU(IEPS)
      PDD=PDD+PD(IEPS)
      PKZB=PKZB+PD(IEPS)*PU(IEPS)*PZ(IBMR,IEPS)
      PKE=PKE+PD(IEPS)*PCCEPS(IEPS)
   10 CONTINUE
      PDD=PDD/XEPS
      PUU=PUU/XEPS
      TIMSIC(1)=TIMSIC(1)/(PDD*XEPS)
```

```
      TIMSIC(2)=TIMSIC(2)/(PDD*XEPS)
      PUPD=PUU/PDD
      PKZB=PKZB/(XEPS*PUU)
      PKE=PKE/XEPS
      DIV=XEPS*(PDD-PUU)
      DO 20 I1=1,2
      DO 20 J=1,2
      PKK(J,I1)=.5*PD(1)*(1.-PU(1))*P(J,I1,1)
      DO 15 K=2,NEPSM1
      PKK(J,I1)=PKK(J,I1)+PD(K)*(1.-PU(K))*P(J,I1,K)
   15 CONTINUE
   20 PKK(J,I1)=(PKK(J,I1)+.5*PD(NTEPS)*(1.-PU(NTEPS)) *
     *P(J,I1,NTEPS))/DIV
      PS=1.-PKK(2,IFTR)*PDD*(1.-PUPD)
      PKB=PKK(2,IBMR)*PDD*(1.-PUPD)+PKZB*PUU
      IF (PS.EQ.1.) GO TO 30
      EKT=(1.-PS**YN)/(1.-PS)
      GO TO 31
   30 EKT=YN
   31 EKT=PKB*EKT
      PKF=(1.-PUPD)*PDD*PKK(2,IFTR)
      PSB=1.-PKF
      IF (PSB.EQ.1.) GO TO 40
      ESB=(1.-PSB**YN)/(1.-PSB)
      GO TO 41
   40 ESB=YN
   41 EKF=PKF*ESB
      PKL=PKB-PKE
      PKG=PKL/(1.-PKE)
      PKBGD=PKB/PDD
      PKFGD=PKF/PDD
      DO 25 K=1,NTEPS
      PKBGDE(K)=P(2,IBMR,K)*(1.-PU(K))+PZ(IBMR,K)*PU(K)
      PKFGDE(K)=P(2,IFTR,K)*(1.-PU(K))
   25 CONTINUE
      RETURN
      END
```

```
$IBFTC PJU.    XR7,DECK
      SUBROUTINE PJU
      COMMON/EXEC/NEPS,IFTR,IBMR,TMAXBF
     *,TTMAX
      COMMON/INPUTT/NRUN,NID,XID(72),NTEPS,EPSLON(73),
     1NN,TSMALL(25),TAWARE(25),F(25),T(5,2,25)
      COMMON/INPUTC/D,RHO,TC(2),M(2),IWK(6,2),
     1PKP(6,2),N,NRUNC,NDPMID,DPMID(72)
      COMMON/EPSCA/PD(73),PU(73),P9(2,2,73),PZ(2,73),IGRIDP
      COMMON/GRIDD/I,J,LIMA,LIMB,LIMC,IU,A,PK(6,2)
      COMMON/ORDERR/TT(6,2,7),IWAB(6,2)
      COMMON/PJUC/PP(2,2,73,25), PPZ(2,73,25)
      COMMON/PKGCAL/PKL,PKG,PKE,PCCEPN(73,25),PCCEPS(73)
      COMMON/MULTPL/TFIREP(25,10),MISTPP(25,10),IAFIRP(25,10)
      COMMON/SIMULT/TF
      M1=M(1)
      M2=M(2)
      DO 1 I1=1,M1
      KW1=IWK(I1,1)
    1 PK(KW1,1)=PKP(KW1,1)
      DO 4 I2=1,M2
      KW2=IWK(I2,2)
    4 PK(KW2,2)=PKP(KW2,2)
      ENTRY PJU1
      LLITL=1
      P=0.
      IF (I.EQ.IFTR) PCCEPN(NEPS,IGRIDP)=0.
    5 IF (TFIREP(LLITL,IGRIDP).EQ.0.) GO TO 35
      IAL=IAFIRP(LLITL,IGRIDP)
      IF (TFIREP(LLITL,IGRIDP).GT.TC(IAL)) GO TO 30
      IF (IAFIRP(LLITL,IGRIDP).EQ.I) GO TO 10
    7 LLITL=LLITL+1
      GO TO 5
   10 IF (LLITL.GT.1) GO TO 12
      IF (TFIREP(LLITL,IGRIDP).GT.TC(I)) GO TO 30
      K=MISTPP(1,IGRIDP)
      P=PK(K,I)
```

```
      GO TO 30
   12 KTINY=LLITL-1
   14 IF(TFIREP(LLITL,IGRIDP)-TF.GT.TFIREP(KTINY,IGRIDP))GO TO 20
      IF(KTINY.EQ.1) GO TO 16
      KTINY=KTINY-1
      GO TO 14
   16 IF(TFIREP(LLITL,IGRIDP).GT.TC(I)) GO TO 30
      K=MISTPP(LLITL,IGRIDP)
      P=P+(1.-P)*PK(K,I)
      GO TO 30
   20 CONTINUE
      NT=KTINY+1
      Y=1.
   21 IF (NT.GT.LLITL-1) GO TO 25
      K=MISTPP(NT,IGRIDP)
      IF(IAFIRP(NT,IGRIDP).EQ.I) Y=Y*(1.-PK(K,I))
      NT=NT+1
      GO TO 21
   25 NT=1
      X=1
   26 CONTINUE
      MINT=MISTPP(NT,IGRIDP)
      IANT=IAFIRP(NT,IGRIDP)
      IF (TFIREP(NT,IGRIDP).LE.TC(IANT)) X=X*(1.-PK(MINT,IANT))
      NT=NT+1
      IF(NT.LE.KTINY) GO TO 26
      K=MISTPP(LLITL,IGRIDP)
      P=P+X*Y*PK(K,I)
   30 CONTINUE
      IF (I.NE.IFTR) GO TO 7
      IF (ABS(TFIREP(LLITL,IGRIDP)-TAWARE(IGRIDP)).GT..001) GO TO 7
      PCCEPN(NEPS,IGRIDP)=P
      GO TO 7
   35 CONTINUE
      IF (IU.EQ.0) GO TO 37
      PP(IU,J,NEPS,IGRIDP)=P
      RETURN
```

```
37    PPZ(J,NEPS,IGRIDP)=P
      RETURN
      END
```

```
$IBFTC OUTP.    XR7,DECK
      SUBROUTINE OUTPUT
      COMMON/EXEC/NEPS,IFTR,IBMR,TMAXBF
     *,TTMAX
      COMMON/PKGCAL/PKL,PKG,PKE,PCCEPN(73,25),PCCEPS(73)
      COMMON/FKT1/FKT,YN,EKF
      COMMON/EPSCA/PD(73),PU(73),P(2,2,73),PZ(2,73),IGRIDP
      COMMON/PJUC/PP(2,2,73,25), PPZ(2,73,25)
      COMMON/INPUTC/D,RHO,TC(2),M(2),IWK(6,2),
     1PKP(6,2),N,NPUNC,NDPMID,DPMID(72)
      COMMON/UNCON/PUPD,PDD,PKZB,PKK(2,2)
     1,PKBU,PKD1FB,PKD2FB,PKJIA(2,2)
      COMMON/INPUTT/NRUN,NID,XID(72),NTEPS,EPSLON(73),
     1NN,TSMALL(25),TAWARE(25),F(25),T(6,2,25)
      COMMON /NUOUT/PKB
      COMMON/SICLV/TIMSIC(2),SICTIM(2,25),SICEPS(2,5)
      COMMON/NUPROB/PKBGDE(25),PKFGDE(25),PKBGD,PKFGD
      DIMENSION ICOUNT(25),LIMUP(3)
      DATABP7B,BP1B,BP2F,BP1F,BP2F/6HP2B    ,6HP1B    ,6HP2B    ,
     16HP1F    ,6HP2F    /,(ICOUNT(I),I=1,10)/1,2,3,4,5,6,7,8,9,10/
      DATA (ICOUNT(I),I=11,25)/11,12,13,14,15,16,17,18,19,20,21,22,23,24
     1,25/
      WRITE(6,2)
 2    FORMAT(1H1,47X,28HATAC-2 DATA PROCESSING MODEL)
      WRITE (6,1) (XID(IJ),IJ=1,NID)
 1    FORMAT(1H0,35X,12A6)
      WRITE(6,11)(DPMID(IJ),IJ=1,NDPMID)
 11   FORMAT(36X,12A6)
      IF (N.GT.10) GO TO 100
      KOUNT=1
      LIMUP(1)=N
      GO TO 120
  100 IF (N.GT.20) GO TO 105
      KOUNT=2
      LIMUP(1)=10
      LIMUP(2)=N
      GO TO 120
```

```
105 KOUNT=3
    LIMUP(1)=10
    LIMUP(2)=20
    LIMUP(3)=N
120 CONTINUE
    WRITE(6,3)NEPS,N
3   FORMAT(1H0,32X,24HNO. OF INITIAL ANGLES = I2,10X,
   130HNO. OF GRIDPOINTS PER ANGLE = I2)
    DO 20 I=1,NTEPS
20  EPSLON(I)=EPSLON(I)*57.2957795
    DO 130 L=1,KOUNT
    N1=L*10-9
    N2=LIMUP(L)
    WRITE (6,4)BPZB,(ICOUNT(IJ),IJ=N1,N2)
  4 FORMAT(1HA,63X,A6/11X,4HGRID/11X,6HPOINTS,10(8X,I2))
    WRITE(6,26)
26  FORMAT(5X,6HEPSLON)
    DO 21 I=1,NTEPS
 21 WRITE (6,5) EPSLON(I),(PPZ(IBMR,I,J),J=N1,N2)
130 CONTINUE
5   FORMAT(4X,F8.3,11X,10(F7.5,3X))
    DO 140 L=1,KOUNT
    N1=L*10-9
    N2=LIMUP(L)
    WRITE (6,4) BP1B,(ICOUNT(IJ),IJ=N1,N2)
    WRITE(6,26)
    DO 22 I=1,NTEPS
 22 WRITE (6,5) EPSLON(I),(PP(1,IBMR,I,J),J=N1,N2)
140 CONTINUE
    DO 150 L=1,KOUNT
    N1=L*10-9
    N2=LIMUP(L)
    WRITE (6,4) BP2B,(ICOUNT(IJ),IJ=N1,N2)
    WRITE(6,26)
    DO 23 I=1,NTEPS
 23 WRITE (6,5) EPSLON(I),(PP(2,IBMR,I,J),J=N1,N2)
150 CONTINUE
```

```
      DO 160 L=1,KOUNT
      N1=L*10-9
      N2=LIMUP(L)
      WRITE (6,4) BP1F,(ICOUNT(IJ),IJ=N1,N2)
      WRITE(6,26)
      DO 24 I=1,NTEPS
   24 WRITE (6,5) EPSLON(I),(PP(1,IFTR,I,J),J=N1,N2)
  160 CONTINUE
      DO 170 L=1,KOUNT
      N1=L*10-9
      N2=LIMUP(L)
      WRITE (6,4) BP2F,(ICOUNT(IJ),IJ=N1,N2)
      WRITE(6,26)
      DO 25 I=1,NTEPS
   25 WRITE (6,5) EPSLON(I),(PP(2,IFTR,I,J),J=N1,N2)
  170 CONTINUE
      WRITE(6,8)
    8 FORMAT(1H0,19X,2HPU,11X,2HPD,11X,3HPZB,10X,3HP1B,10X,3HP2B,10X,3HP
     *1F,10X,3HP2F,8X,5HPKBGD,8X,5HPKFGD/5X,6HEPSLON)
      DO 10 I=1,NTEPS
10    WRITE(6,9)EPSLON(I),PU(I),PD(I),PZ(IBMR,I),P(1,IBMR,I),
     1P(2,IBMR,I),P(1,IFTR,I),P(2,IFTR,I)
     2,PKBGDE(I),PKFGDE(I)
    9 FORMAT(4X,F8.3,9(6X,F7.5))
      WRITE(6,6)PUPD,PDD,PKZB,PKK(1,IBMR),PKK(2,IBMR),
     1PKK(1,IFTR),PKK(2,IFTR)
6     FORMAT(1HA,4X,7HPU/D = F7.5,5X,5HPD = F7.5,3X,7HPKZB = F7.5,3X,
     17HPK1B = F7.5,3X,7HPK2B = F7.5,3X,7HPK1F = F7.5,3X,
     27HPK2F = F7.5)
      WRITE (6,200) EKT,EKF,PKBGD,PKFGD
  200 FORMAT(1H0,4X,4HEKB=F9.5,4X,4HEKF=F9.5,4X,6HPKBGD=F9.5,4X,6HPKFGD=
     1F9.5)
      WRITE(6,500) PKB
500   FORMAT(1H0,40X,4HPKB=,F9.5)
      WRITE (6,300) (ICOUNT(IJ),IJ=1,N)
300   FORMAT(1HA,63X,9HPC(EPS,N)/11X,4HGRID/11X,6HPOINTS,10(8X,I2))
      WRITE (6,26)
```

```
      DO 350 I=1,NTEPS
  350 WRITE (6,5) EPSLON(I),(PCCEPN(I,J),J=1,N)
      WRITE (6,301)
  301 FORMAT(1H0,21X,8HPC(IEPS)/5X,6HEPSLON)
      DO 351 I=1,NTEPS
  351 WRITE (6,9) EPSLON(I),PCCEPS(I)
      WRITE (6,302) PKL,PKG,PKE
  302 FORMAT(1H0,10X,4HPKL=F7.5,5X,4HPKG=F7.5,5X,4HPKE=F7.5)
      WRITE (6,303) TIMSIC(1),TIMSIC(2)
  303 FORMAT(1H0,10X,5HS(1)=,F5.3,20X,5HS(2)=,F5.3)
      RETURN
      END
```

```
$IBFTC GRIDC.   XR7,DECK
      SUBROUTINE GRIDDC
      COMMON/EXEC/NRPS,IFTR,IBMR,TMAXBF
     *,TTMAX
      COMMON/INPUTC/D,RHO,TC(2),M(2),IWK(6,2),
     1PKP(6,2),N,NRUNC,NDPMID,DPMID(72)
      COMMON/INPUTT/NRUN,NID,XID(72),NTEPS,EPSLON(73),
     1NN,TSMALL(25),TAWARE(25),F(25),T(6,2,25)
      COMMON/GRIDD/I,J,LIMA,LIMB,LIMC,IU,A,PK(6,2)
      COMMON/EPSCA/PD(73),PU(73),P(2,2,73),PZ(2,73),IGRIDP
      COMMON/MULTPL/TFIREP(25,10),MISTPP(25,10),IAFIRP(25,10)
      I=IBMR
      J=IFTR
      LIMA=M(I)
      IM1=IWK(1,IFTR)
      TTMAX=T(IM1,1,IGRIDP)
      IF (LIMA.EQ.1) GO TO 6
      DO 5 IM=2,LIMA
      IM1=IWK(IM,IFTR)
      IF (T(IM1,1,IGRIDP).EQ.1000.) GO TO 5
      IF (TTMAX.LT.T(IM1,1,IGRIDP) ) TTMAX=T(IM1,1,IGRIDP)
    5 CONTINUE
    6 CONTINUE
      IF (TAWARE(IGRIDP).GE.AMIN1(TC(1),TTMAX)) GO TO 20
      A=1.
      I=IBMR
      J=IFTR
      IU=2
      CALL PJU
      IU=1
      I2=M(J)
      DO 10 I1=1,I2
      KW=IWK(I1,J)
   10 PK(KW,J)=0.
      CALL PJU1
      I=IFTR
      J=IBMR
```

```
      IU=2
      CALL PJU
      IU=1
      I2=M(J)
      DO 11 I1=1,I2
      KW=IWK(I1,J)
11    PK(KW,J)=0.
      CALL PJU1
      RETURN
20    A=0.
      IU=0
      I2=M(J)
      DO 12 I1=1,I2
      KW=IWK(I1,J)
12    PK(KW,J)=0.
      M1=M(1)
      DO 1 I1=1,M1
      KW1=IWK(I1,1)
1     PK(KW1,1)=PKP(KW1,1)
      CALL PJU1
      RETURN
      END
```

```
$IBFTC MAIN.    XR7,DECK
      COMMON/PKGCAL/PKL,PKG,PKE,PCCEPN(73,25),PCCEPS(73)
      COMMON/EXEC/NEPS,IETR,IBMR,TMAXBF
     *,TTMAX
      COMMON/INPUTC/D,RHO,TC(2),M(2),IWK(6,2),
     1PKP(6,2),N,NFUNC,NDPMID,DPMID(72)
      COMMON/INPUTT/NRUN,NID,XID(72),NTEPS,EPSLON(73),
     1NN,TSMALL(25),TAWARE(25),F(25),T(6,2,25)
      COMMON/PJUC/PP(2,2,73,25), PPZ(2,73,25)
      REWIND 11
 1    IETR=1
      IBMR=2
      DO 2 K1=1,2
      DO 2 K2=1,73
      DO 2 K3=1,25
      PPZ(K1,K2,K3)=0.
      DO 2 K4=1,2
 2    PP(K1,K4,K2,K3)=0.
      WRITE(6,11)
 11   FORMAT(1H1)
      CALL INPUT1
      TMAXBF=AMAX1(TC(1),TC(2))
      DO 10 I1=1,NTEPS
      NEPS=I1
      PCCEPS(NEPS)=0.
      CALL INPUT2
      CALL EPSCAL
 10   CONTINUE
      CALL UNCOND
      CALL OUTPUT
      REWIND 11
      GO TO 1
      END
```

```
$IBFTC INPUT.   XR7,DECK
      SUBROUTINE INPUT1
      COMMON/COMP/ICOMP
      COMMON/EKT1/EKT,YN,EKF
      COMMON/EXEC/NEPS,IFTR,IBMR,TMAXBF
     *,TTMAX
      COMMON/INPUTC/D,RHO,TC(2),M(2),IWK(6,2),
     1PKP(6,2),N,NRUNC,NDPMID,DPMID(72)
      COMMON/INPUTT/NRUN,NID,XID(72),NTEPS,EPSLON(73),
     1NN,TSMALL(25),TAWARE(25),F(25),T(6,2,25)
      COMMON/TIME/TTMIS(6,2,25)
      COMMON/MULTPL/TFIREP(25,10),MISTPP(25,10),IAFIRP(25,10)
      COMMON/SICLV/TIMSIC(2),SICTIM(2,25),SICEPS(2,5)
      COMMON/SIMULT/TF
      DIMENSION TOTIME(6,2)
      DIMENSION NTOT(2)
      READ(5,1)NRUNC,N,(M(IJ),IJ=1,2)
 1    FORMAT(14X,I3/14X,I2/14X,2I5)
      M1=M(1)
      M2=M(2)
      READ(5,2)(IWK(K,1),K=1,M1),(IWK(L,2),L=1,M2)
 2    FORMAT(14X,11I5)
      READ(5,40) NDPMID,(DPMID(I),I=1,NDPMID)
 40   FORMAT(14X,I2/(12A6))
      READ(5,3)D,RHO,TC(1),TC(2)
 3    FORMAT(14X,E15.0/14X,E15.0/14X,2E15.0)
      DO 15 I1=1,M1
      KW1=IWK(I1,1)
      READ(5,16)PKP(KW1,1)
 15   CONTINUE
 16   FORMAT(14X,E15.0)
      DO 17 I2=1,M2
      KW2=IWK(I2,2)
 17   READ(5,16)PKP(KW2,2)
      READ (5,16) YN
      READ (5,2) LCAP
      READ (5,16) TF
```

```
    5 READ (11,50) NRUN,NTEPS,NN,NM1,NM2
   50 FORMAT(1H0,5I5)
      READ (11,40) NID,(XID(ID),ID=1,NID)
      READ (11,50)   ICOMP
      IF(NRUNC.EQ.NRUN)GO TO 4
      DO 10 II=1,NTEPS
      READ(11,52)EPSLON(II),(TSMALL(IJ),TAWARE(IJ),F(IJ),
     1IJ=1,NN),((T(MIS1,1,NGP1),MIS1=1,NM1),NGP1=1,NN),
     2((T(MIS2,2,NGP2),MIS2=1,NM2),NGP2=1,NN)
      READ (11,52) ((TTMIS(MIS1,1,NGP1),MIS1=1,NM1),NGP1=1,NN),
     *((TTMIS(MIS2,2,NGP2),MIS2=1,NM2),NGP2=1,NN)
      DO 1002 IDISC=1,NN
      DO 1000 IMNUTS=1,LCAP
      READ (11,200) NGP,TFIREP(IMNUTS,NGP),MISTPP(IMNUTS,NGP),
     *IAFIRP(IMNUTS,NGP),IFZERO
      IF(IFZERO.NE.0) GO TO 1001
 1000 CONTINUE
 1001 CONTINUE
 1002 CONTINUE
      READ(11,52) (SICTIM(1,NGP1),SICTIM(2,NGP1),NGP1=1,NN)
 10   CONTINUE
      GO TO 5
      ENTRY INPUT2
      READ(11,52)EPSLON(NEPS),(TSMALL(IJ),TAWARE(IJ),F(IJ),
     1IJ=1,NN),((T(MIS1,1,NGP1),MIS1=1,NM1),NGP1=1,NN),
     2((T(MIS2,2,NGP2),MIS2=1,NM2),NGP2=1,NN)
      READ (11,52) ((TTMIS(MIS1,1,NGP1),MIS1=1,NM1),NGP1=1,NN),
     *((TTMIS(MIS2,2,NGP2),MIS2=1,NM2),NGP2=1,NN)
 52   FORMAT(1H0,4E16.8/(17X,3E16.8))
      EPSLD=EPSLON(NEPS)*57.2957795
      WRITE(6,30)EPSLD
 30   FORMAT(1H0,57X,9HEPSLON = ,F7.3)
      WRITE(6,31)
 31   FORMAT(1H0,22X,9HGRIDPOINT,20X,1HT,21X,6HTAWARE,22X,1HF)
      DO 32 IJ=1,N
 32   WRITE(6,33)IJ,TSMALL(IJ),TAWARE(IJ),F(IJ)
 33   FORMAT(26X,I2,4X,3(15X,F10.3))
```

```
      WRITE(6,34)
34    FORMAT(1H0,37X,7HMISSILE,7X,3HA/C,6X,9HGRIDPOINT,3X,
     120HT(MIS,A/C,GRIDPOINT),6X,12HTIME IN ENV.)
      NTOT(1)=NM1
      NTOT(2)=NM2
      DO 35 I1=1,2
      MIS=NTOT(I1)
      DO 35 I2=1,MIS
      DO 35 I3=1,N
   35 WRITE (6,36) I2,I1,I3,T(I2,I1,I3),TTMIS(I2,I1,I3)
36    FORMAT(41X,I1,10X,I2,10X,I2,13X,F8.3,13X,F8.3)
      DO 100 I1=1,2
      MIS=NTOT(I1)
      DO 100 I2=1,MIS
      TOTIME(I2,I1)=0.
      DO 100 I3=1,N
      TOTIME(I2,I1)=TOTIME(I2,I1)+TTMIS(I2,I1,I3)
  100 CONTINUE
      WRITE (6,37) (TOTIME(I1,1),I1=1,NM1)
      WRITE (6,38) (TOTIME(I2,2),I2=1,NM2)
   37 FORMAT(1H0,31X,13HTOTAL(K,1) = 6(F8.3,5X))
   38 FORMAT(1H0,31X,13HTOTAL(K,2) = 6(F8.3,5X))
      WRITE (6,202)
  202 FORMAT(1H0,3X,10HGRID POINT,6X,8HAIRCRAFT,7X,7HMISSILE,
     *4X,10HTIME FIRED)
      DO 2002 ISAVE=1,NN
      DO 2000 IMNUTS=1,LCAP
      READ (11,200) NGP,TFIREP(IMNUTS,NGP),MISTPP(IMNUTS,NGP),
     *IAFIRP(IMNUTS,NGP),IFZERO
      WRITE (6,201) NGP,IAFIRP(IMNUTS,NGP),MISTPP(IMNUTS,NGP),
     *TFIREP(IMNUTS,NGP)
      IF (IFZERO.NE.0) GO TO 2001
 2000 CONTINUE
 2001 CONTINUE
 2002 CONTINUE
      READ(11,52) (SICTIM(1,NGP1),SICTIM(2,NGP1),NGP1=1,NN)
  200 FORMAT(1H ,I5,F15.7,3I10)
```

```
201 FORMAT(1H ,3I15,F10.3)
    RETURN
4   WRITE(6,7)
7   FORMAT(41X,42HINPUT FOR THE ATAC-2 DATA PROCESSING MODEL)
    WRITE(6,6)(XID(IJ),IJ=1,NID)
6   FORMAT(1H0,29X,12A6)
    WRITE(6,20)NRUNC,N,M(1),M(2),NTEPS
20  FORMAT(1H0,31X,11HRUN NUMBER ,I3,5X,4HN = ,I3,5X,
   17HM(1) = ,I3,5X,7HM(2) = ,I3,5X,8HNTEPS = ,I3)
    WRITE(6,21)(IWK(K,1),K=1,M1)
21  FORMAT(1H0,31X,9HW(K,1) = ,6(I3,5X))
    WRITE(6,22)(IWK(K,2),K=1,M2)
22  FORMAT(1H0,31X,9HW(K,2) = ,6(I3,5X))
    WRITE(6,23)D,RHO,TC(1),TC(2)
23  FORMAT(1H0,29X,4HD = F8.3,5X,6HRHO = F8.3,5X,
   18HTC(1) = F8.3,5X,8HTC(2) = F8.3//)
    DO 25 I1=1,M1
    KW1=IWK(I1,1)
25  WRITE(6,26)KW1,PKP(KW1,1)
26  FORMAT(55X,3HPK(,I2,6H,1) = F8.3)
    DO 27 I2=1,M2
    KW2=IWK(I2,2)
27  WRITE(6,28)KW2,PKP(KW2,2)
28  FORMAT(55X,3HPK(,I2,6H,2) = F8.3)
    WRITE (6,60) TF
60  FORMAT(1H0,54X,3HTF=F8.3)
    RETURN
    END
```

SECTION 10

AN EXAMPLE OF INPUT AND OUTPUT

In this section an exemple of the computer printout for the simulation of one engagement is presented and discussed. The inputs and the printout for this run are shown on pages 144 through 150.

10.1 Inputs

On page 144 and the top of page 145 are shown some of the inputs of the program. The combatants are coded on top of page 144 as aircraft 10 (fighter) and 11 (bomber) for identification purposes. The range of the fighter's IFF is given as 12,000 feet. The altitude associated with the performance data of each aircraft is 10,000 feet. Then data are listed which in most cases are self-explanatory. For example, t_{max}, the limit on the time of each engagement is 300 seconds. Moving down to the double columns of data (aircraft 10 is on the left, aircraft 11 on the right) it is seen that the avionics data for the aircraft are identical. Each has a detection radar with range of 140,000 ft. and half-angle 60°, etc.

On pages 145 and 146 soma of the weapon envelopes are shown. The entries of the big blocks of data on pages 145 and 146, give the ranges of a weapon envelope as a function of the angle-off and target velocity. The first weapon number, MIS, is 1 and it is on aircraft 1 (aircraft 10).[1)] The first tables are for RF1, the outer range of the envelopes when the target pulls G1 lateral g's. G1, G2, G3 in the input list correspond to the FORTRAN names GT(1), GT(2), GT(3), respectively. Page 144 shows G1 = 0, which means the target is non-maneuvering.

1) I.e., in this particular example, aircraft number 1 is aircraft design number 10, and aircraft number 2 is design number 11.

The last block of data shown at the bottom of page 146 gives the inner range of MIS number 5 on aircraft 2 (aircraft 11) associated with G_3 (= 4.9) lateral g's.

On page 147 the all important specific power and g_i functions of each aircraft are shown. At the extreme left the speeds associated with all the other entries are listed. The second column headed GMAXT gives the $g_i(V)$ function of aircraft i . For example, at 1185.8 feet per second, aircraft 1 is able to sustain 4.3 total g's without having to decelerate. The third column, which is not an input, but is computed by the program gives the turning rate $\dot{\beta}_i$, at $g_i(V)$ total g's. The fourth column, headed GBIG, contains the $G_i(V)$ function of aircraft 1. This is the upper bound to the total g's of the aircraft. For example, at 754.6 feet per second aircraft 1 may sustain no more than 6.2 total g's. Finally, columns 5 through 11 describe the specific power as a function of speed (by row) and total g's (by column). At, say, a speed of 1401.4 feet per second and 3 total g's this aircraft has a specific power of 220.5 feet per second. Also at a speed 1,078 feet per second and, say, 6.0 total g's the aircraft must decelerate, for its specific power is - 99.3 feet per second. Finally, the last two blocks of data on page 147 give the $\alpha_{MIS}(i)$ and $g_{MIS}(i)$ limits, respectively.

On page 148 some more input date are shown. At the top of the page is shown $\alpha_{max}(i)$ for both aircraft as .26 radians. The angle is actually inputted as 15°, but is converted to radians by the computer prior to printing out. Further down, opposite the word NUMIS, is shown the number of weapons N(MIS, i) of type MIS carried by aircraft i . Here it is seen that each aircraft carries two of weapon types 1 and 2, and eight of weapon type 5. VSTR corresponds to V_i^* and is computed by the program.

Finally, pages 149 and 150 show the printout (results) of one engagement. The whole table is ordered on time, shown in seconds in the first column under T . The second column gives the range, R , between the aircraft. Then the variables peculiar to the bomber are given. They are from left to right:

XB	the x coordinate of the bomber,
YB	the y coordinate of the bomber,
VB	the speed of the bomber,
MB	tha m-state of the bomber,
GB	the total g's of the bomber,
BB	the turning rate of the bomber,
ALPB	the tracking angle of the bomber,
I	the ST(i) indicator of the bomber,
PHIB	the bomber's angle off the fighter,
KB	the information state of the bomber.

The right side of the page gives these same variables for the fighter.

At the bottom of page 150 the summary of firings of each aircraft is given. The first block of data gives firings of all weapon types. The entry of 1,000.0 for the time of firing indicates that the weapon was not fired. The THIS column gives the time spent in that weapon's envelope, and the remaining columns give the relative parameters at the time of firing: the range, tracking angle and angle-off of the firing aircraft. The last block of data also lists the variables $IA(\ell)$, $MI(\ell)$, $T(\ell)$, which are inputs to the DPM. For example, at 83.5 seconds into the engagement aircraft 1 fired two missiles at aircraft 2, weapons number 1 and 2. Then aircraft 2 fired missile type 1 at 163 seconds followed by another at 190.5 seconds. One observes that the first firing of aircraft 1 alerted its enemy, as evidenced by t_{AWARE} = 83.5 seconds. Finally, the last line gives the sick time of each aircraft. This is the amount of time that each aircraft spent in a condition in which $O_1 \geq 1$.

It may be interesting and instructivs to observe these data in greater detail and thereby better understand some of the notions previously developed. The reader is cautioned to consider this as only one example and as such it cannot present a general view of the model.

Firstly, for 83.5 seconds the bomber is totally unaware of the fighter; it is moving up the y-axis in linear flight at a constant velocity of 916 feet per second. During this time the fighter is afforded the opportunity to gain a very favorable position. The fighter initially has active information (KF = 4). It also notices that the bomber is unaware and hence the fighter sets its lag angle, $\alpha_{max}(F)$, to 60° and thereby tries to get behind the bomber. Also, in the initial 50 seconds the fighter accelerates to press the attack. Then as the fighter acquires the tail of its enemy it lags less and less till it reaches ϕ^* , and thereafter flies a pure pursuit course, ALPF = α_F = 0 . As the data indicate, ϕ^* is a little less than 57.8°. It is $\cos^{-1} (V_F/2V_B)$, or about $\cos^{-1}$ (1350/2 · 916) = 57°. From this point until the awareness of the bomber, the fighter is in the ϕ^* cone of the bomber and flies a pure pursuit course. Another point of interest in this portion of the engagement is seen in the m-state of the bomber. Between 40 and 50 seconds this index changes from 2 to 1. As previously noted this signals that the fighter goes from a state of acceleration, m = 2 , to a state of deceleration, m = 1 . Of course, this is clearly indicated by the speed printout as well. In this portion of the engagement the fighter gains IFF. The range of the

IFF gear of the fighter is 12,000 feet (see inputs). And this range is attained somewhere between 50 and 60 seconds of combat. Verification of the acquisition of IFF by the fighter is seen in its information state KF. This index changes from 4 (active information) to 5 (active with IFF). These are some of the notable events in the early portions of the combat.

All during the early pursuit the fighter holds fire so as to get to a good range off the tail of the enemy, R^*. Against an unaware enemy $R^*(F)$ is set to $R'(1, F)$ which in this case is 4,100 feet. This range is attained shortly after 80 seconds, for at 80 seconds the range is 4,265 feet. Of course, due to the time slice nature of the simulation 4,100 feet will not be hit exactly. However, the fighter attains this range within 5 feet, for it is seen in the firing summary that the fighter launched its first 2 missiles at a range of 4,095 feet. It then alerted the bomber by these firings. And from 83.5 seconds on, both aircraft are aware of each other.

At 90 seconds the bomber is in information state 9: Passive and IFF. This is because the fighter is illuminating the bomber with its tracking radar. It has IFF because it was fired upon. In this state the bomber turns as hard as possible to acquire its enemy. In so doing it loses the passive information and goes into a lost information state, KB = 11. Therein (100 to 150 seconds) it continues to turn hard and decelerates to V^* = 755 feet per second. This is the speed at which its sustainable turning rate is best. In so doing the bomber finally, at about 150 seconds, acquires the fighter and gets active information, KB = 5. The maneuver also causes the fighter to lose information, KF = 11.

Through the rest of the engagement each aircraft turns hard to acquire the other. However, it is seen that the bomber, in fact, does a better job of this than the fighter, for from 150 seconds until the end, the bomber has active information most of the time, while the fighter is in a lost or passive information state. Further, the bomber is able to launch two weapons at the maneuvering fighter, one at 163 seconds and the other at 190 seconds. This superiority of the bomber over the fighter is, in part, clear from the inputs. From the $\dot{\beta}$ functions for each aircraft, (see page 147) it is seen that V^* for both is 754.6 feet per second. But at this speed the bomber can out-turn the fighter since $\dot{\beta}_B = .274$ radians per second, while $\dot{\beta}_F = .261$. Thus by the arguments presented in Appendix D, superiority of the bomber over the fighter in the latter half of the engagement is expected.

INPUT FOR THE ATAC-2 MODEL

```
AIRCRAFT 10
AIRCRAFT 11
 RIFF=12000, ALT=10.000, -RC DATA

N                  10
NM                  6            6
IPS                 2            2

TMAX    =       300.000
TMIN    =       300.000
TSTAR   =        10.000
OENS    =         0.002
OELEPS  =        45.000
OELTAT  =         0.200
EPSLON  =         0.
RANGE   =    600000.000
G1      =         0.
G2      =         2.830
G3      =         4.900
RMIN    =         1.000
H       =     10000.000
```

ADEC	-16.100	-16.100
RDET	140000.000	140000.000
RIFF	12000.000	12000.000
ROPT	36000.000	36000.000
RPA	330000.000	330000.000
RTRK	140000.000	140000.000
VC	323.000	302.000
VZ	916.000	916.000
RSTAR	600.000	600.000
ALPDET	60.000	60.000
ALPIFF	60.000	60.000
ALPOPT	90.000	90.000
ALPPAS	180.000	180.000

ALPTRK	20.C00	20.000
VMAX	1563.C00	1563.000
GP	8.CC0	8.000

RF1(VA,SIGMA1
A/C NO. 1 WEAPON NC. 1

SIGMAS VA	-180.C0	-150.CC	-120.00	-9C.CC	-60.00	-3C.00	0.	30.00	60.00	90.00	120.00	150.00	18C.00
647.	1800C.	180CC.	18CC0.	14CCC.	120CC.	1000C.	10000.	10000.	12000.	14000.	18000.	18CCC.	18C00.
97C.	18C0C.	180CC.	180C0.	14CCC.	11CCC.	9CC0.	7000.	9000.	11000.	14000.	18000.	18C00.	18C00.
1294.	3900C.	35CCC.	160C0.	0.	7CCC.	6000.	5000.	6000.	7000.	0.	16000.	35CC0.	39C00.
1617.	4600C.	30CCC.	0.	C.	C.	C.	4C00.	C.	0.	0.	0.	30C0C.	46C00.

RF1(VA,SIGMA1
A/C NO. 1 WEAPON NC. 2

SIGMAS VA	-18C.CC	-150.CC	-120.00	-9C.CC	-60.C0	-30.C0	0.	30.00	60.00	90.C0	120.00	15C.00	18C.0C
647.	5C00C.	42CCC.	26000.	20C0C.	16C0C.	14C00.	14000.	14000.	16000.	20000.	26000.	42C00.	50C00.
970.	8C00C.	58CCC.	26CC0.	22CCC.	16CCC.	1400C.	14000.	14000.	16000.	22000.	26000.	58CC0.	8CC00.
1294.	8CC0C.	58CCC.	280C0.	18CCC.	15CCC.	120CC.	11000.	12000.	15000.	18000.	28000.	58C00.	80C00.
1617.	8C00C.	56CCC.	22000.	C.	120CC.	10C00.	8000.	10000.	12000.	0.	22000.	56C00.	8CC00.

RF1(VA,SIGMA1
A/C NO. 1 WEAPON NC. 3

SIGMAS VA	-180.0C	-150.CC	-120.00	-9C.CC	-60.C0	-30.00	0.	30.00	60.00	90.00	120.00	15C.00	18C.00
647.	5C00C.	4200C.	260C0.	2CCCC.	16CCC.	14C0C.	14C00.	14000.	160C0.	20000.	26000.	42C00.	50C00.
970.	8C000.	58CCC.	260C0.	22CCC.	16CCC.	14C00.	14000.	14000.	16000.	22000.	26000.	58CC0.	8CC00.
1294.	8C00C.	580CC.	280C0.	18CCC.	15CCC.	120CC.	11C00.	12000.	15000.	15000.	28000.	58CCC.	8CC00.
1617.	8C00C.	560CC.	220CC.	C.	12CCC.	10CCC.	8C00.	1C000.	12000.	0.	22000.	56CCC.	8CC00.

RF1(VA,SIGMA1
A/C NO. 1 WEAPON NC. 4

SIGMAS VA	-180.C0	-150.CC	-120.00	-9C.CC	-6C.C0	-3C.C0	0.	30.00	60.00	90.00	120.00	15C.C0	18C.0C
647.	C.	C.	0.	C.	8CC0.	8CCC.	10C00.	8000.	8000.	0.	0.	0.	0.
970.	0.	C.	0.	C.	C.	6C0C.	6000.	6000.	0.	0.	0.	C.	0.
1294.	C.	C.	0.	C.	C.	C.	4000.	0.	0.	0.	0.	C.	0.
1617.	C.	C.	0.	C.	C.	C.	4000.	0.	0.	0.	0.	0.	0.

RF1(VA,SIGMA1
A/C NO. 1 WEAPON NC. 5

SIGMAS VA	-180.00	-15C.CC	-120.00	-9C.CC	-6C.C0	-3C.C0	0.	30.00	60.00	90.00	120.00	150.00	18C.0C
647.	C.	C.	0.	C.	C.	3C0C.	3C00.	3C00.	0.	0.	0.	C.	0.
97C.	C.	C.	0.	C.	C.	3C0C.	3000.	300C.	0.	0.	0.	0.	0.
1294.	C.	C.	0.	C.	C.	3C0C.	3000.	3000.	0.	0.	0.	0.	0.
1617.	C.	C.	0.	C.	C.	300C.	3C00.	3000.	0.	0.	0.	C.	0.

RF1(VA,SIGMA1
A/C NO. 2 WEAPON NC. 1

SIGMAS VA	-180.C0	-150.C0	-120.00	-9C.CC	-60.C0	-3C.00	0.	30.00	60.00	90.00	120.00	15C.00	18C.0C
647.	1800C.	18CCC.	18CCC.	14CCC.	12CCC.	10C0C.	10000.	10000.	12000.	14000.	180C0.	18CC0.	18C00.
97C.	1800C.	180CC.	180C0.	14CCC.	11CCC.	9C0C.	7000.	9000.	11000.	14000.	18000.	18CC0.	18C00.
1294.	3900C.	35CCC.	160C0.	C.	7CCC.	6C0C.	5000.	6000.	7000.	0.	16000.	35CCC.	39C00.

SIGMAS	-180.00	-150.00	-120.00	-90.00	-60.00	-30.00	0.	30.00	60.00	90.00	120.00	150.00	180.00
VA													
647.	0.	0.	0.	0.	0.	4000.	4000.	3000.	6000.	6000.	0.	0.	0.
970.	0.	0.	0.	0.	0.	0.	4000.	4000.	0.	0.	0.	0.	0.
1294.	0.	0.	0.	0.	0.	0.	3000.	0.	0.	0.	0.	0.	0.
1617.	0.	0.	0.	0.	0.	0.	3000.	0.	0.	0.	0.	0.	0.

RF3'(VA,SIGMA)
A/C NO. 1 WEAPON NO. 5

SIGMAS	-180.00	-150.00	-120.00	-90.00	-60.00	-30.00	0.	30.00	60.00	90.00	120.00	150.00	180.00
VA													
647.	0.	0.	0.	0.	0.	500.	500.	500.	0.	0.	0.	0.	0.
970.	0.	0.	0.	0.	0.	500.	500.	500.	0.	0.	0.	0.	0.
1294.	0.	0.	0.	0.	0.	500.	500.	500.	0.	0.	0.	0.	0.
1617.	0.	0.	0.	0.	0.	500.	500.	500.	0.	0.	0.	0.	0.

RF3'(VA,SIGMA)
A/C NO. 2 WEAPON NO. 1

SIGMAS	-180.00	-150.00	-120.00	-90.00	-60.00	-30.00	0.	30.00	60.00	90.00	120.00	150.00	100.00
VA													
647.	10000.	8000.	6000.	6000.	6000.	6000.	6000.	4000.	4000.	6000.	8000.	10000.	10000.
970.	10000.	7000.	6000.	6000.	6000.	6000.	5000.	3000.	3000.	5000.	7000.	10000.	10000.
1294.	10000.	7000.	6000.	7000.	7000.	7000.	5000.	3000.	3000.	6000.	9000.	11000.	10000.
1617.	14000.	10000.	12000.	7000.	7000.	7000.	5000.	2000.	3000.	6000.	9000.	11000.	14000.

RF3'(VA,SIGMA)
A/C NO. 2 WEAPON NO. 2

SIGMAS	-180.00	-150.00	-120.00	-90.00	-60.00	-30.00	0.	30.00	60.00	90.00	120.00	150.00	180.00
VA													
647.	12000.	14000.	17000.	10000.	8000.	6000.	8000.	8000.	8000.	8000.	8000.	8000.	12000.
970.	16000.	18000.	16000.	10000.	8000.	6000.	0000.	8000.	8000.	8000.	10000.	12000.	16000.
1294.	16000.	22000.	16000.	12000.	10000.	6000.	6000.	6000.	6000.	8000.	10000.	16000.	16000.
1617.	16000.	24000.	18000.	14000.	10000.	6000.	6000.	6000.	6000.	10000.	12000.	16000.	16000.

RF3'(VA,SIGMA)
A/C NO. 2 WEAPON NO. 3

SIGMAS	-180.00	-150.00	-120.00	-90.00	-60.00	-30.00	0.	30.00	60.00	90.00	120.00	150.00	180.00
VA													
647.	17000.	14000.	12000.	10000.	8000.	6000.	8000.	8000.	0.	8000.	0000.	8000.	17000.
970.	16000.	18000.	16000.	10000.	8000.	6000.	0.	0.	8000.	0000.	10000.	17000.	16000.
1294.	0.	22000.	16000.	12000.	10000.	0.	6000.	6000.	6000.	8000.	10000.	16000.	0.
1617.	0.	24000.	18000.	14000.	0.	0.	6000.	0000.	6000.	10000.	12000.	0.	0.

RF3'(VA,SIGMA)
A/C NO. 2 WEAPON NO. 4

SIGMAS	-180.00	-150.00	-120.00	-90.00	-60.00	-30.00	0.	30.00	60.00	90.00	120.00	150.00	180.00
VA													
647.	0.	0.	0.	0.	0.	4000.	4000.	3000.	6000.	6000.	0.	0.	0.
970.	0.	0.	0.	0.	0.	0.	4000.	4000.	0.	0.	0.	0.	0.
1294.	0.	0.	0.	0.	0.	0.	3000.	0.	0.	0.	0.	0.	0.
1617.	0.	0.	0.	0.	0.	0.	3000.	0.	0.	0.	0.	0.	0.

RF3'(VA,SIGMA)
A/C NO. 2 WEAPON NO. 5

SIGMAS	-180.00	-150.00	-120.00	-90.00	-60.00	-30.00	0.	30.00	60.00	90.00	120.00	150.00	100.00
VA													
647.	0.	0.	0.	0.	0.	500.	500.	500.	0.	0.	0.	0.	0.
970.	0.	0.	0.	0.	0.	500.	500.	500.	0.	0.	0.	0.	0.
1294.	0.	0.	0.	0.	0.	500.	500.	500.	0.	0.	0.	0.	0.
1617.	0.	0.	0.	0.	0.	500.	500.	500.	0.	0.	0.	0.	0.

G'S FOR RC TABLES

VA	GMAXT	EETCOT	GBIG	RC'S 1.000	2.000	3.000	4.000	5.000	6.000	7.000
323.400	1.000	0.	1.000	133.400	-1000.000	-1000.000	-1000.000	-1000.000	-1000.000	-1000.000
431.200	2.000	0.129	2.000	197.600	146.800	-1000.000	-1000.000	-1000.000	-1000.000	-1000.000
539.000	3.100	0.175	3.100	260.200	229.200	162.000	-1458.000	-1000.000	-1000.000	-1000.000
646.800	4.500	0.218	4.500	322.100	296.900	253.700	182.600	-180.000	-1000.000	-1000.000
754.600	6.200	0.261	6.200	389.500	365.500	326.600	271.600	199.400	72.100	-360.000
862.400	7.000	0.259	7.000	459.400	426.500	368.300	293.900	222.600	131.300	-1.000
970.200	6.500	0.213	7.000	522.800	481.200	426.000	349.500	250.900	109.900	-112.600
1078.000	5.000	0.146	7.000	430.400	359.700	242.000	88.900	3.440	-99.300	-220.000
1185.800	4.300	0.114	7.000	429.200	358.900	246.000	76.800	-165.700	-461.500	-691.900
1293.600	4.500	0.109	7.000	405.500	339.800	239.500	83.000	-114.100	-345.700	-627.700
1401.400	4.400	0.098	7.000	393.000	331.800	220.500	66.000	-118.200	-368.700	-641.200
1509.200	4.200	0.087	7.000	363.500	317.800	188.000	24.500	-159.800	-380.000	-681.800
1563.100	4.100	0.082	7.000	336.100	289.200	160.600	20.800	-188.700	-420.400	-711.000

ENERGY MANUVERABILITY TABLES FOR AC= 2

G'S FOR RC TABLES

VA	GMAXT	EETCOT	GBIG	RC'S 1.000	2.000	3.000	4.000	5.000	6.000	7.000	8.000
301.840	1.000	0.	1.000	219.600	-1000.000	-1000.000	-1000.000	-1000.000	-1000.000	-1000.000	-1000.000
373.400	1.200	0.066	1.200	242.500	-970.000	-1000.000	-1000.000	-1000.000	-1000.000	-1000.000	-1000.000
431.200	2.200	0.146	2.200	350.600	289.700	-1158.800	-1000.000	-1000.000	-1000.000	-1000.000	-1000.000
539.000	3.300	0.188	3.300	451.400	423.100	338.100	-788.900	-1000.000	-1000.000	-1000.000	-1000.000
646.800	4.700	0.229	4.700	550.600	529.300	481.900	386.900	274.600	-1000.000	-1000.000	-1000.000
754.600	6.500	0.274	6.500	655.400	636.500	603.700	548.200	447.500	237.700	-237.700	-1000.000
862.400	7.330	0.271	7.330	762.100	731.000	694.600	649.700	507.400	417.900	347.400	192.100
970.200	7.330	0.241	7.330	879.200	832.800	782.900	673.300	546.200	378.600	330.100	206.800
1078.000	7.330	0.217	7.330	900.800	837.800	732.700	635.500	526.400	369.600	170.900	63.900
1185.800	7.330	0.197	7.330	997.100	935.700	837.800	692.500	529.200	374.500	185.300	-37.000
1293.600	7.330	0.181	7.330	1072.000	1016.400	929.800	787.600	618.700	415.600	162.200	-97.600
1401.400	7.330	0.167	7.330	1193.800	1146.600	1045.000	915.900	745.700	534.600	296.500	4.400
1509.200	7.330	0.155	7.330	1316.500	1277.300	1146.400	1048.700	866.300	670.100	414.900	138.800
1563.100	7.330	0.150	7.330	1384.500	1342.300	1235.900	1109.300	938.300	726.500	490.600	193.300

ALPMIS(A/C,MISSILE)

1	20.00	20.00	20.00	20.00	5.00	180.00
2	20.00	20.00	20.00	20.00	5.00	180.00

GMIS(A/C,MISSILE)

A/C \ MISSILE	1	2	3	4	5	6
1	8.00	8.00	8.00	8.00	8.00	8.00
2	8.00	8.00	8.00	8.00	8.00	8.00

```
$WRITAL
ALPMAX =      C.26179935E-CO,     0.26179935E-CO,
XLAMCA =      C.            ,     0.            ,
PI     =      C.            ,     0.            ,
TAL    =      C.            ,     0.            ,
TALPIS =      C.49999999E 01,     0.C9999999E C2,     0.            ,     C.            ,     C.3COCOCOOE C1,
              C.            ,     0.49999999E C1,     0.09999999E 02,     0.            ,     C.            ,
              C.3CCOOCCOE C1,     0.            ,
NUPIS  =            2,          2,          0,          0,          8,          0,          2,
                    2,          0,          0,          8,          0,
LCAP   =           25,
VSTR   =      C.75459999E C3,     0.75459999E C3,

RANCUN =      C.3CCOOCCOE C4,     0.3CCOOOOOE C4,
ALPCUN =      C.52359870E CO,     0.52359870E CO,
RTSST  =      C.89999999E 05,     0.89999999E 05,
TCELTS =      C.09999999E C1,     0.25000000E-CO,     0.25000000E-00,
O      =      C.09999999E 01,     0.09999999E C1,     0.09999999E 01,     C.09999999E C1,     C.09999999E 01,
              C.C9999999E C1,     0.C9999999E C1,     0.09999999E 01,     0.09999999E C1,     C.09999999E C1,
              C.C9999999E 01,     0.09999999E C1,     0.09999999E 01,     C.09999999E C1,     C.09999999E C1,
              C.C9999999E C1,     0.C9999999E C1,     0.09999999E 01,     0.09999999E C1,     C.09999999E 01,
              C.C9999999E C1,     0.C9999999E C1,
NUINC  =            2,
GB     =      C.            ,     0.            ,
SICKF  =      C.            ,     0.            ,
KCAPP  =            2,
TATRPR =      C.2CCOOCCOE C2,     0.36COOOOOE C3,    -0.C0000000E-19,    -C.000COOCOE-19,    -C.0COCOCOOE-19,
             -C.CCCCCCCOE-19,     0.2CCOOOOOE C2,     0.36000000E 03,    -0.00000000E-19,    -C.0COCOCOOE-19,
             -C.CCCCOCCOE-19,    -0.0CCOOOOOE-19,
RPRIME =      C.41COOCCOE C4,     0.6CCOOOOOE C3,    -0.00000000E-19,    -0.00000000E-19,    -C.0COCOCCCE-19,
             -C.CCOCOCCCE-19,     0.41COOOOOE C4,     0.60000000E 03,    -0.00000000E-19,    -0.COOCOCOOE-19,
             -C.CCCCOCCOE-19,    -0.COCOOOOOE-19,
MISRPR =            2,          5,  -8589934592,  -8589934592,  -8589934592,  -8589934592,          2,
                    5,  -8589934592,  -8589934592,  -8589934592,  -8589934592,

$ END
```

T	R	XB	YO	VB	PO	GB	BO	ALPB	I	PHIB	KO	XF	YF	VF	PF	GF	BF	ALPF	I	PHIF	KF
0.03	53625.	0.	0.	916.	0	1.00	0.	-75.0	1	-12C.0	1	-51802.	13880.	916.	5	1.00	C.23	6C.C	0	105.0	4
10.03	43532.	-0.	9160.	516.	0	1.CC	0.	-05.3	1	161.6	1	-43386.	12722.	986.	5	1.53	-C.04	-18.4	2	94.7	4
20.03	33967.	-C.	18320.	516.	0	1.CC	0.	-102.5	1	168.3	1	-33110.	10740.	1112.	2	1.86	-C.05	-11.7	2	77.1	4
30.03	25923.	-C.	27480.	516.	0	1.CC	0.	-122.2	1	175.6	1	-21937.	13668.	1213.	2	2.00	-C.05	-4.4	2	57.8	4
40.03	19368.	-0.	36640.	916.	0	1.0C	0.	-140.3	1	-18C.0	1	-12367.	21735.	1304.	2	1.58	-C.03	-C.	2	39.7	4
50.03	13682.	-C.	458C0.	916.	0	1.CC	0.	-156.0	1	-18C.0	1	-5395.	33229.	1391.	1	1.52	-C.03	-C.	2	23.2	4
6C.03	8956.	-C.	54960.	516.	0	1.00	0.	-169.0	1	-10C.0	1	-1585.	46145.	1252.	1	1.24	-C.02	-C.	2	10.2	5
70.03	5822.	-C.	64120.	516.	0	1.00	0.	-177.2	1	-18C.0	1	-284.	50305.	1145.	1	1.04	-C.01	-C.	2	2.8	5
8C.03	4245.	-C.	73280.	516.	0	1.CC	0.	-179.6	1	-18C.0	1	-31.	69015.	991.	1	1.00	-C.00	-C.	2	0.4	5
90.03	2512.	-3785.	79666.	811.	9	6.93	-0.27	-128.4	0	179.9	9	-1922.	77980.	935.	1	6.97	-C.24	-C.1	0	51.6	5
10C.03	3879.	-3528.	74206.	753.	8	6.45	-0.27	171.7	3	107.3	11	-7786.	74620.	875.	3	6.94	-C.25	-72.7	0	-8.3	5
11C.03	5243.	-1711.	79131.	754.	8	6.48	-0.27	-130.6	3	153.4	11	-1403.	73894.	875.	1	6.94	-C.25	-26.6	0	49.4	5
120.03	2C93.	-5693.	75485.	754.	8	6.49	-0.27	-153.7	3	121.1	11	-5789.	77576.	735.	3	5.74	-C.25	-58.9	0	26.3	5
13C.C3	6584.	-601.	77279.	755.	8	6.50	-0.27	-135.4	3	118.9	11	-4712.	72135.	709.	3	5.17	-C.23	-61.1	0	44.6	5
140.03	4587.	-5588.	77618.	755.	8	6.5C	-0.27	-90.1	3	137.5	11	-1600.	76282.	621.	1	4.19	-C.21	-42.5	C	89.9	5
150.C3	5938.	-1161.	75202.	755.	8	6.50	-0.27	-111.9	3	55.4	11	-6804.	77054.	608.	8	3.26	-C.16	-124.6	3	68.1	11
160.03	8392.	-4662.	75310.	755.	2	6.50	-0.27	-43.8	0	63.6	5	-7430.	71388.	622.	8	3.45	-C.17	-11C.4	3	136.2	11
170.03	3064.	-4288.	73444.	656.	1	4.15	-0.23	-4.2	0	9C.0	5	-1816.	71779.	658.	1C	4.08	-C.19	-9C.C	C	175.8	5
180.03	5135.	491.	75693.	697.	3	5.08	-0.23	-88.5	0	7.8	5	-4551.	76668.	651.	8	4.77	-C.22	-172.2	3	51.5	11
190.03	7371.	-4369.	78217.	724.	2	5.75	-0.25	-22.7	0	59.0	5	-7530.	71558.	700.	8	4.57	-C.22	-121.C	3	157.3	11

200.03	1424.	-3379.	72267.	693.	1	5.51	-0.25	-21.9	0	74.5	5	-1955.	72223.	703.	8	5.04	-C.23	-105.5	3	158.1	11
21C.C3	6850.	332.	76294.	722.	3	5.71	-0.25	-67.4	0	19.5	5	-6509.	75955.	704.	8	5.04	-C.23	-16C.5	3	112.6	11
220.03	5649.	-5137.	75452.	69C.	1	2.6C	-0.12	-2.3	2	79.9	5	-6875.	70087.	704.	1C	5.04	-C.23	-10C.1	0	177.7	9
23C.03	2610.	-1586.	7C874.	7CT.	8	5.34	-0.24	-96.4	3	-2.6	11	-2627.	734C3.	7C4.	8	5.04	-C.23	177.4	3	83.6	11
240.03	7479.	-1C81.	76773.	725.	3	5.76	-0.25	-41.7	0	39.4	5	-8400.	74740.	704.	8	5.04	-C.23	-14C.6	3	138.3	11
250.03	3324.	-5539.	72830.	669.	5	4.45	-0.21	-3.8	0	91.6	5	-5982.	69506.	704.	5	5.04	-C.23	-88.4	0	176.2	7
260.03	5246.	-1C79.	7C379.	716.	3	5.56	-0.25	-85.9	0	7.0	5	-3975.	74753.	704.	8	5.04	-C.23	-173.C	3	94.1	11
27C.03	7019.	-3C47.	75486.	726.	2	5.79	-0.25	-17.6	0	61.3	5	-9689.	73218.	704.	1C	5.04	-C.23	-118.7	0	162.4	9
28C.03	1258.	-6310.	7C5C7.	669.	1	5.11	-0.24	-36.8	0	57.5	5	-5198.	69920.	704.	8	5.04	-C.23	-122.5	3	143.2	11
29C.03	6989.	-857.	7C989.	722.	3	5.71	-0.25	-63.6	0	22.8	5	-5930.	75797.	7C4.	8	5.04	-C.23	-157.2	3	116.4	11
30C.03	5284.	-5469.	74131.	682.	1	2.78	-0.13	-2.6	2	82.5	5	-10197.	71772.	704.	1C	5.04	-C.23	-97.5	0	177.4	9

TAWARE = 83.5 TLAST = 3CC.2 F = 107151. EPSLCN = 45. CRICPCINT = 10

I	MIS	T	TMIS	R	ALPHA	PHI
1	1	83.50	0.25	4095.04	-C.	C.19
1	2	83.50	0.25	4095.04	-C.	C.19
1	3	1CCC.CO	0.	0.	C.	C.
1	4	1CCC.CO	0.	0.	C.	C.
1	5	1CCC.CO	0.	0.	C.	C.
1	6	1CCC.CO	0.	0.	C.	C.
2	1	163.CO	2.75	7239.95	-18.41	72.55
2	2	1CCC.CO	0.	0.	C.	C.
2	3	1CCC.CO	C.	0.	C.	C.
2	4	1CCC.CO	0.	0.	C.	C.
2	5	1CCC.CO	0.	0.	C.	C.
2	6	1CCC.CO	0.	0.	C.	C.

CRIC PT	AIRCRAFT	MIS. TYPE	TIME FIRED	RANGE	ALPHA	PHI
10	1	1	83.500	4095.036	-0.	0.186
10	1	2	83.5CC	4095.036	-C.	0.186
10	2	1	163.C00	7239.953	-18.407	72.554
10	2	1	19C.50C	7211.751	-18.944	61.960
10	C	C	C.	0.	0.	0.

SICK TIME(1)=2C5.25 SICK TIME(2)=212.25

APPENDIX A

THE PROBABILITY OF DETECTION

Since the ATAC-2 model initiates all engagements with the fighter detecting the bomber and since the relevant probabilities of kill are conditional on detection, the probability of detection as a function of the initial relative heading angle, ϵ , is of importance. This section contains the derivation of that probability. The scenario is one that is consistent with the random search of an area with no prior knowledge of the presence of an enemy. This situation is also consistent with the method of initialization of the combatants by the ENGAGEMENT Model.

Suppose then that the fighter is searching some large area A in which the bomber exists but its position is unknown to the fighter. Let the bomber's position in this area be a random variable with bivariate distribution function

$$P(x, y) = \{Pr \quad X < x \, , \, Y < y\} \, ,$$

where X and Y are the random variables defining the bomber's position in A . Let the bomber's velocity vector be parallel to the x-axis. Now A is such that the bomber's position in A is invariant with time, i.e., the area A moves with the bomber. Hence,

$$\int_A dP(x, y) = 1 \quad .$$

($P(x, y)$ will later be taken to be uniform for simplicity although this need not be the case.)

A.1 Stationary Target

Suppose for the moment that the fighter is searching the area A for a stationary target. The velocity $\overline{V}_B$ is zero and the target does not move. In an amount of time t the fighter's detection pattern will have swept out an area $a = r\,V_F\,t$ where r is the normal projection of the fighter's detection pattern onto the normal to the fighter's velocity vector. Further it may be said that the stationary target will be detected in time t if and only if the target is in the area a at time $t' \leq t$. But the target is in the area a with probability

$$\int_a dP(x,\, y) \quad .$$

And if $P(x, y)$ is a uniform distribution, then this probability is just

$$a/A = \frac{r\,V_F\,t}{A} \quad .$$

The above may now be generalized to account for the moving target.

A.2 Non-Stationary Target

Suppose now, as is generally the case, that the bomber has a velocity $\overline{V}_B$ while the fighter is searching for the target with velocity $\overline{V}_F$. Put in relative coordinates the fighter searches the area A for a stationary target with velocity $\overline{V^*} = \overline{V}_F - \overline{V}_B$, the relative velocity. With this in mind the situation is completely analogous to the stationary target case above. Again the fighter sweeps out an area in A equal to a. By analogy then the area a is $Y^*\,V^*\,t$ where Y^* is the normal projection onto the normal to the velocity vector with which the fighter searches for s ststionary target, namely $\overline{V}^*$. Then, if $P_D(\varepsilon)$ is the probability of detection st some initial relative heading angle ε snd $P(x, y)$ is uniform over A ,

$$P_D(\varepsilon) = \frac{Y^* \, V^* \, t}{A} .$$

V^* is obtained from V_F , V_B and ε by the Law of Cosines, i.e.,

$$V^* = \left[V_F^2 + V_B^2 - 2 \, V_F \, V_B \cos \varepsilon \right]^{1/2} .$$

Let D be the distance that the target travels in time t , analogous to a penetration distance, then

$$P_D(\varepsilon) = (1/A) \, Y^* \, V^* \, D/V_B$$

$$= (1/A) \, F \, D ,$$

where

$$F = Y^* \, V^*/V_B .$$

Now the probability of no detection is $1 - P_D(\varepsilon)$. And if n fighters search the area A independently with the same relative velocity, then the probability that no detection occurs is

$$\left[1 - P_D(\varepsilon)\right]^n .$$

while the probability that at least one detects the target is

$$1 - \left[1 - P_D(\varepsilon)\right]^n .$$

An approximation to this quantity is given by the exponential function:

$$\left[1 - P_D(\varepsilon)\right]^n \simeq - e^{-(n/A)\ F\ D} \quad .$$

Then letting $\delta = n/A$ be the density of fighters in the area A and taking the above approximation

$$P_D(\varepsilon) = 1 - e^{-\delta F\ D} \quad .$$

It is well to note some of the tacit assumptions throughout the derivation of $P_D(\varepsilon)$. For example, it was assumed that the fighter remained within the area A for the total time t . For small values of t or large A this assumption is not unreasonable. Also it was assumed that no detection can take place in zero time; detection is accomplished by relative motion between the aircraft. Therefore, if $\overline{V}*$ the relative velocity vector is zero, then $P_D(\varepsilon)$ will also be zero. (For non-zero velocities of the aircraft this only occurs at $\overline{V}_F = \overline{V}_B$ and $\varepsilon = 0$.)

APPENDIX B

GEOMETRIC CONSIDERATIONS

This Appendix defines the connection between the inertial and relative coordinate systems. Also, the equations of relative motion are derived.

B.1 Inertial Vs. Relative Coordinates

Consider then two aircraft labeled F end B with respective speeds V_F and V_B at some fixed altitude, see Figure B.1-1. The inertial positions of the aircraft are defined by (x_F , y_F) and (x_B , y_B), while the directions of their headings are given by β_F and β_B , measured as shown. Let θ be the direction of the line of sight between the two aircraft. As previously stated α_i is the tracking angle of aircraft i measured from the inner line of sight to the heading of aircraft i , while ϕ_i , the angle-off, is measured from the outer line of sight to the heading of aircraft $j \neq i$.

The following conventions are adopted:

1. all angles are between $-\pi$ and π inclusive,
2. angles measured in a counterclockwise manner are positive while angles measured in a clockwise manner are negative.

Now the inertial coordinate system uniquely defines the positions of the aircraft with respect to any observer. Seven parameters are used -- x_F , x_B , y_F , y_B , β_F , β_B and θ . In this system θ is redundant since it may be obtained from the others; by observation

$$\theta = \mathrm{Tan}^{-1} \left\{ \frac{y_B - y_F}{x_F - x_B} \right\} . \qquad \text{(B.1-1)}$$

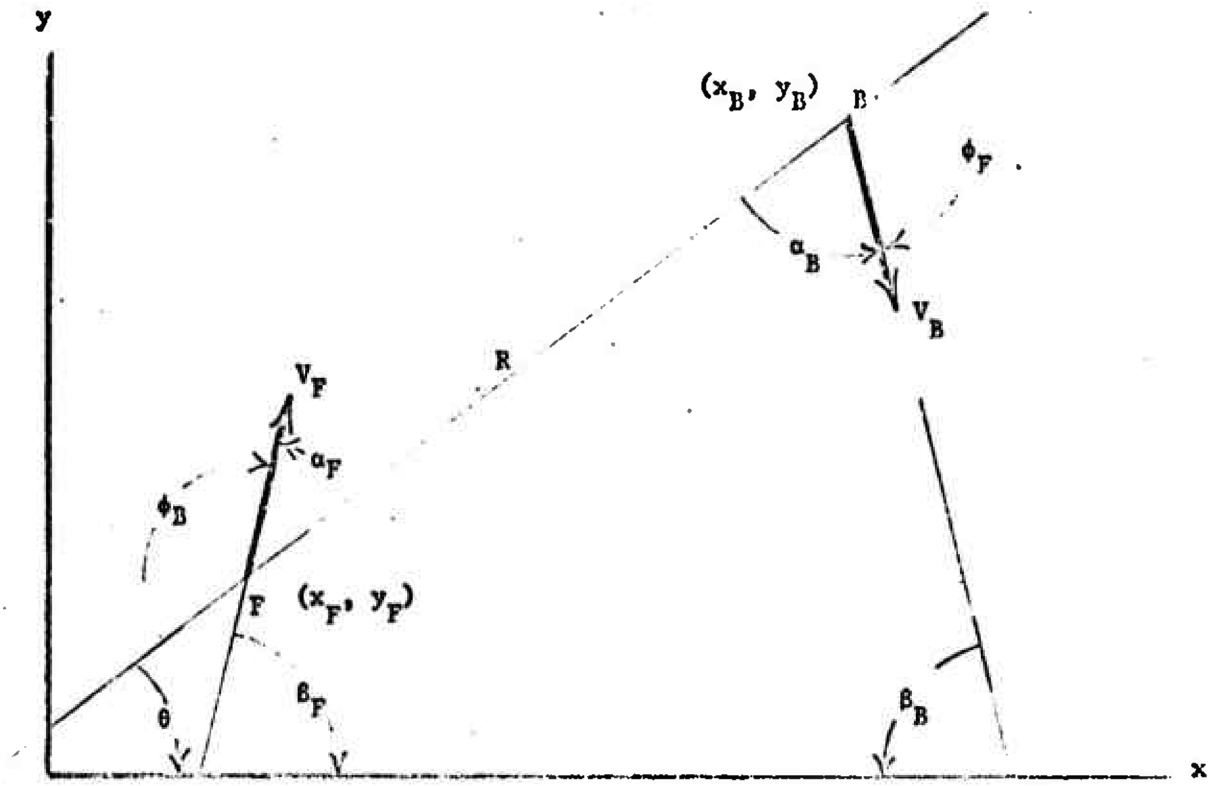


Figure B.1-1 Inertial and Relative Geometry

This may also be taken as a definition of θ. However, the relative coordinate system makes use of five parameters -- R, α_F, α_B, ϕ_F and ϕ_B and does not uniquely define the aircraft's inertial position. One of the angles α_F, ϕ_B and one of the angles α_B, ϕ_F are redundant. Again by observation

$$\alpha_B = \phi_F - \pi \operatorname{sgn}(\phi_F)$$

and

$$\phi_B = \alpha_F - \pi \operatorname{sgn}(\alpha_F) \quad .$$

Thus, only three parameters are needed to define the relative coordinate system; a distance (range) and two angles.

Yet these variables define the positions for an observer that is fixed with respect to an aircraft. Indeed, translations and rotations of the inertial system give rise to the same relative position. The object of the above is to show that the relative coordinates may be uniquely determined by the inertial coordinates. The reverse is, of course, not true. The following gives this correspondence between the inertial and relative coordinate system explicitly.

B.2 Deriving the Relative Coordinates From the Inertial Coordinates

Suppose that the relative coordinates as a function of the inertial coordinates are desired. The range R is given by

$$R = \left[(x_B - x_F)^2 + (y_B - y_F)^2\right]^{1/2} \qquad (B.2\text{-}1)$$

Also by reference to Figure B.1-1 it is seen that

$$\alpha_F = \theta - \beta_F \qquad (B.2\text{-}2)$$

and

$$\phi_F = \theta - \beta_B \quad . \qquad (B.2\text{-}3)$$

Thus the relative coordinate system is defined from the inertial system.

B.3 The Equations of Relative Motion

In this section the equations of relative motion are derived. These equations are expressions for the time derivatives of the relative parameters in terms of the relative parameters.

Firstly, consider the closure rate or rate of change of the range, $\dot{R}$. From (B.2-1) differentiating with respect to time gives

$$R \frac{dR}{dt} = (x_B - x_F) \frac{d}{dt} (x_B - x_F)$$

$$+ (y_B - y_F) \frac{d}{dt} (y_B - y_F) \quad .$$

But

$$\frac{d}{dt} (x_B - x_F) = V_B \cos \beta_B - V_F \cos \beta_F$$

and

$$\frac{d}{dt} (y_B - y_F) = V_B \sin \beta_B - V_F \sin \beta_F \; .$$

Also

$$(x_B - x_F)/R = \cos \theta$$

and

$$(y_B - y_F)/R = \sin \theta \qquad .$$

Thus

$$\frac{dR}{dt} \equiv \dot{R} = V_B(\cos \beta_B \cos \theta - \sin \beta_B \sin \theta)$$

$$- V_F(\cos \beta_F \cos \theta - \sin \beta_F \sin \theta) \quad .$$

Since $\alpha_F = \theta - \beta_F$ and $\phi_F = \theta - \beta_B$, $\dot{R}$ is finally given by

$$\dot{R} = V_B \cos \phi_F - V_F \cos \alpha_F \quad . \qquad (B.3\text{-}1)$$

From (B.2-2) and (B.2-3) the angular rates $\dot{\alpha}_F$ and $\dot{\phi}_F$ are easily obtained as

$$\dot{\alpha}_F = \dot{\theta} - \dot{\beta}_F \qquad (B.3\text{-}2)$$

and

$$\dot{\phi}_F = \dot{\theta} - \dot{\beta}_B \quad . \qquad (B.3\text{-}3)$$

Now, $\dot{\beta}_F$ and $\dot{\beta}_B$ are the turning rates of the respective aircraft, which are at the control of the respective pilots. However, $\dot{\theta}$, the turning rate of the line of sight, must be obtained. To do so note that

$$x_B - x_F = R \cos\theta \quad .$$

Then

$$\frac{d}{dt}(x_B - x_F) = \dot{R}\cos\theta - R\dot{\theta}\sin\theta$$

$$= V_B \cos\beta_B - V_F \cos\beta_F \quad .$$

With this and the expression for $\dot{R}$, (B.3-1), rearrangement gives

$$R\dot{\theta}\sin\theta = V_F(\cos\beta_F - \cos\alpha_F \cos\theta)$$

$$- V_B(\cos\beta_B - \cos\phi_F \cos\theta) \quad .$$

But $\cos\beta_B = \cos(\theta - \phi_F)$ and $\cos\beta_F = \cos(\theta - \alpha_F)$; hence

$$R\dot{\theta}\sin\theta = V_F \sin\theta \sin\alpha_F - V_B \sin\theta \sin\phi_F \quad ,$$

and finally the turning rate of the line of sight is given by

$$\dot{\theta} = \frac{V_F \sin\alpha_F - V_B \sin\phi_F}{R} \quad . \qquad \text{(B.3-4)}$$

APPENDIX C

THE DEL PURSUIT COURSE

The Decreasing Lag Pursuit Course forms one of the basic tactical considerations of the ATAC-2 Model. The course dictates the manner in which each aircraft pursues its enemy; the method of attack. This section describes and defines this course. Also, some of the implications of the course are shown.

C.1 Rationale

The development of the DEL Pursuit Course was motivated by the assumed objectives of an attacking aircraft engaged in air-to-air combat. These objectives are two-fold; it is assumed that in general a pursuer desires

1) to avoid a nose-on attack, and
2) to attack the enemy from behind.

The nose-on attack is avoided to help prevent the enemy from firing its weapons. Further, the pursuer's weapons are in general not as effective when fired at an aircraft traveling towards it. Also, being in the front hemisphere of an enemy while traveling towards it is an unstable situation. On the other hand, a steady state situation off the tail of an enemy at a range and angle-off at which its more lethal weapons may be launched is a very desirable situation. Of course, counter examples to the desirability of these objectives under more specific situations may be pointed out; these assumptions are for a rather general case.

A pursuit course that helps an aircraft accomplish these objectives is one that tends to avoid head-on attacks while helping acquire the tail of the enemy. Such a course is the DEL Pursuit Course. This course dictates that a pursuer point more and more towards the enemy when approaching a tail chase.

The DEL Pursuit Course specifies the amount by which an aircraft will deviate from a pure pursuit course (pointing at the target) as a function of where it is with respect to the target. When in the front hemisphere of its enemy the DEL Pursuit Course requires the pursuer to deviate by a certain amount. Then as the pursuer swings around, decreasing the angle-off, this deviation decreases. Finally, as a tail chase is approached the deviation goes to zero, so that a pure pursuit course is flown. The DEL Pursuit Course is defined by specifying α_i (the tracking angle) as a function of ϕ_i, the angle-off. Figure C.1-1 gives the general form of the function.

There are two parameters in the DEL Pursuit Course function; α_{MAX} and ϕ^*. The function decreases between π and ϕ^* for decreasing $|\phi_i|$ and is zero for $|\phi_i| \leq \phi^*$. Thus, there is a cone, with half-angle ϕ^*, off the tail of the target in which the pursuer will fly a pure pursuit course (setting $\alpha_i = 0$). Further, α_{MAX} is the largest deviation and this value occurs only at $\phi_i = \pm \pi$ or when the enemy points directly at the aircraft flying a DEL Pursuit Course. A graphic description is shown in Figure C.1-2.

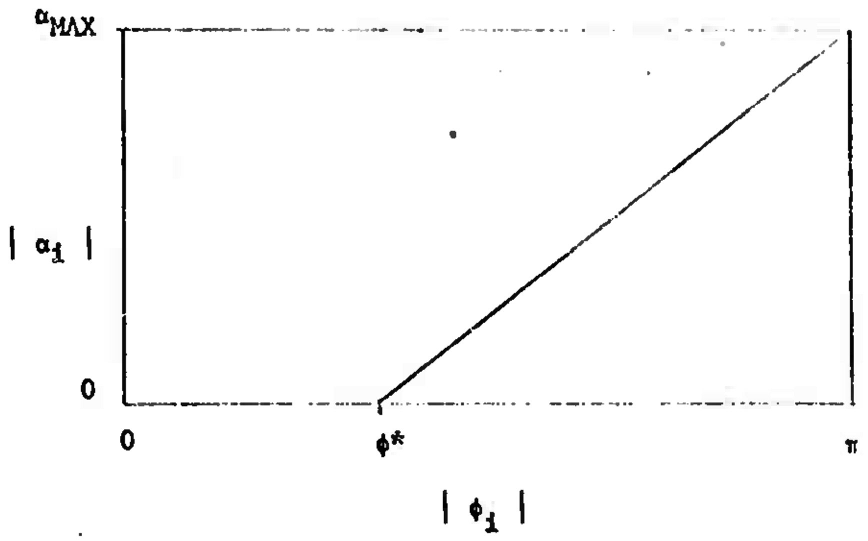


Figure C.1-1 Defining Function of the DEL Pursuit Course

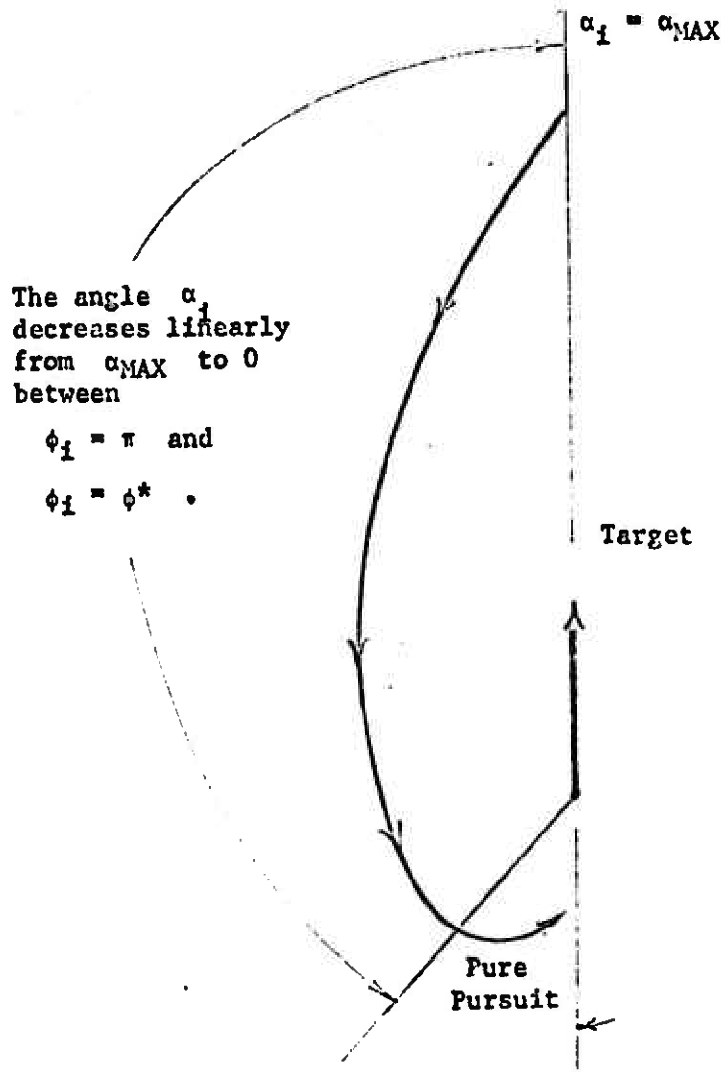


Figure C.1-2 A DEL Pursuit Course

C.2 Formulation

To specify the general form of the DEL Pursuit Course it is necessary to define some terms. In general, the term lag implies pointing to the rear of the target while the term lead implies pointing ahead. With this in mind consider Figure C.2-1. It is seen in (a) that F is pointing ahead of B or leading B while in (b) that F is pointing behind or lagging B. These positions are generalized by the following definitions:

1) a lag position is one in which the velocity vectors lie on opposite sides of the line of sight, and

2) a lead position is one in which both velocity vectors lie on the same side of the line of sight.

Lag or lead courses are courses in which the positions at all times are lag or lead, respectively. This is interpreted symbolically as follows: A lag position implies that $\operatorname{sgn}(\alpha_1) \neq \operatorname{sgn}(\phi_1)$; a lead position implies that $\operatorname{sgn}(\alpha_1) = \operatorname{sgn}(\phi_1)$.

Now it will be noticed, again in Figure C.2-1 that in (a) B also lags F while in (b) B leads F. Indeed, this is the general case as the definition implies. That is, if F lags/leads B, then correspondingly B lags/leads F. F cannot lag B while B leads F, or vice-versa. It is well to note four singularities at this point. They occur when $\alpha_1 = 0$ or π , and $\phi_1 = 0$ or π (see Figure C.2-2). Here both velocity vectors lie on the same line. And in any of these cases it is said that both aircraft lead and lag at the same time.

To give a general expression for the tracking angle of an aircraft flying a DFL Pursuit Course it is necessary to define the desired tracking angle η as the angle to which α_i will be set whenever aircraft i is flying a

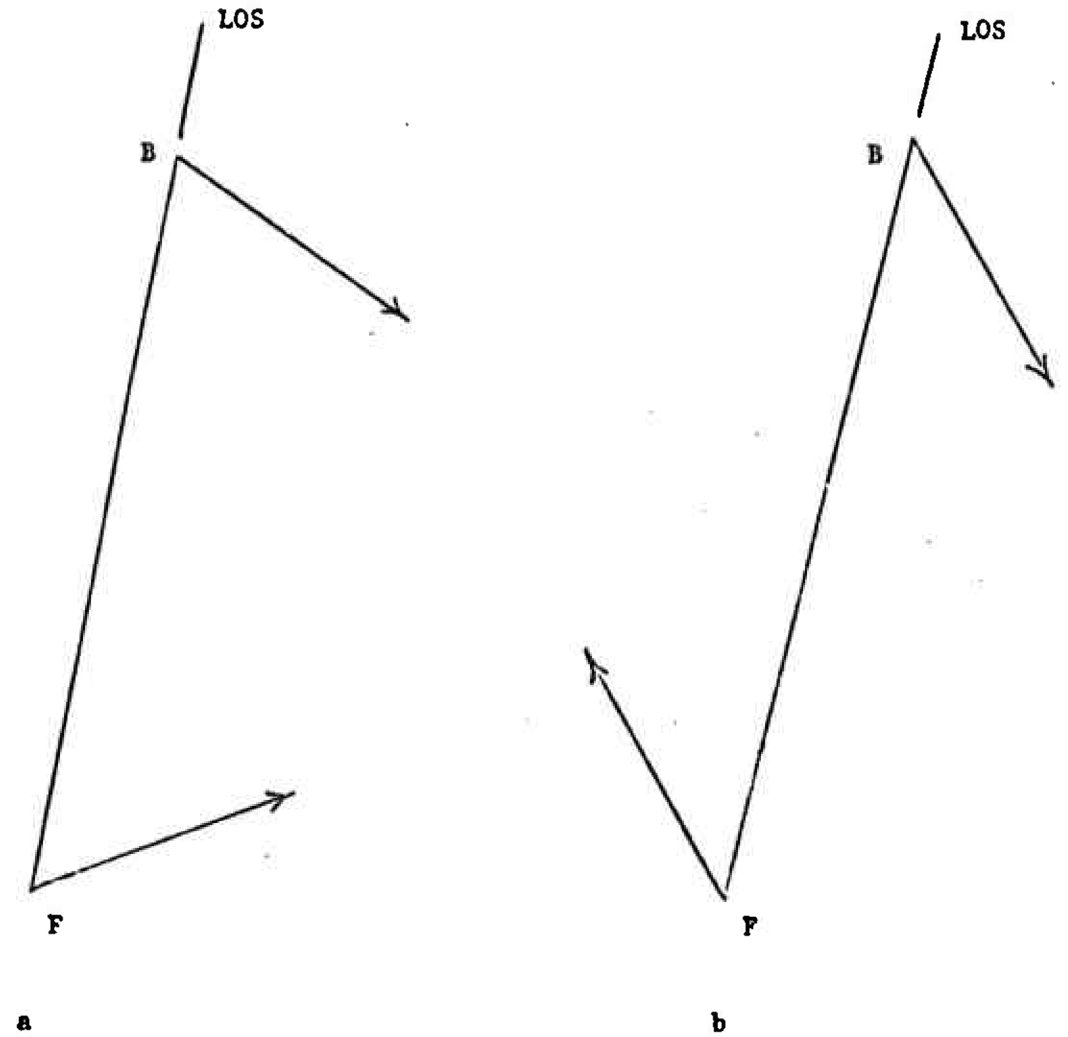


Figure C.2-1 Lead (a) and Lag (b) Positions

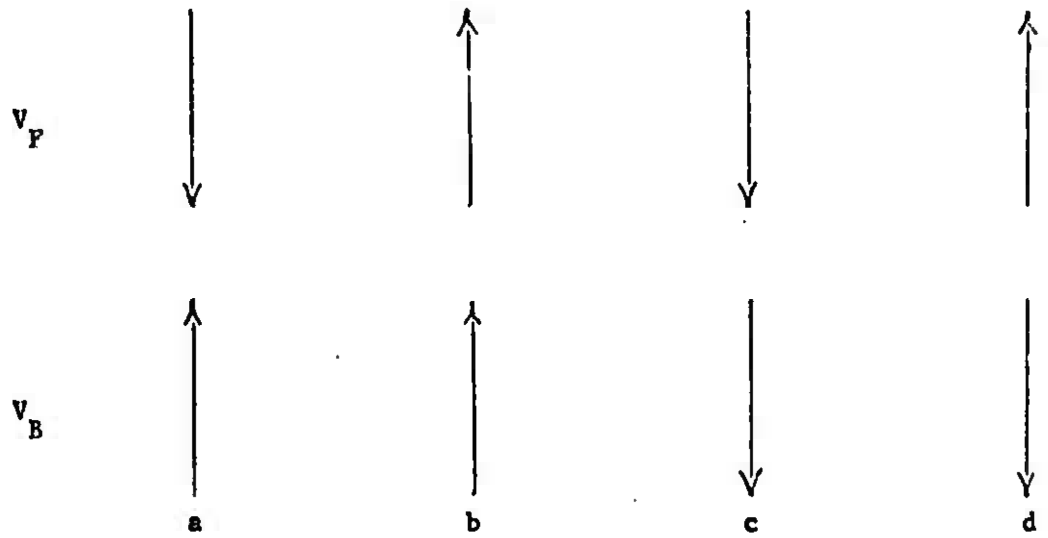


Figure C.2-2 Singular Cases

DEL Pursuit Course. And symbolically expressing Figure C.1-1 in slope-intercept form gives

$$\eta = \begin{cases} \operatorname{sgn}(\phi_i) \dfrac{\alpha_{MAX}(i)\,(|\phi_i| - \phi^*)}{\phi^* - \pi} & , \text{ if } |\phi_i| > \phi^* , \\ 0 & , \text{ if } |\phi_i| \leq \phi^* . \end{cases} \qquad \text{(C.2-1)}$$

Now since $0 \leq \phi^* \leq \pi/2$ (see derivation in Section C.4) the term $\operatorname{sgn}(\phi_i)\,(|\phi_i| - \phi^*)/(\phi^* - \pi)$ will always have the opposite sign of ϕ_i. Then the final sign of η will be determined by the sign of $\alpha_{MAX}(i)$. If $\alpha_{MAX}(i) > 0$, then $\operatorname{sgn}(\eta) \neq \operatorname{sgn}(\phi_i)$, whereas if $\alpha_{MAX}(i) < 0$, then $\operatorname{sgn}(\eta) = \operatorname{sgn}(\phi_i)$. Also, if $\operatorname{sgn}(\eta) = +1$ a lag course will be called for, while if $\operatorname{sgn}(\eta) = -1$ a lead course will be called for. Again due to the fact that F's leading B implies B leads F, it is impossible to require one aircraft to lead and one to lag. Hence, a requirement is that $\alpha_{MAX}(F)$ and $\alpha_{MAX}(B)$ have the same sign for a lag.

Equation (C.2-1) is an abbreviated form. For purposes of generality the capability of specifying a constant deviation angle was introduced. This is accomplished by subtracting the quantity $\operatorname{sgn}(\phi_i)\,\lambda_i$ from both forms of the expression given in (C.2-1), where λ_i is the desired constant lag (lead) angle of aircraft i.

Thus the final form of the DEL Pursuit Course is given by

$$\eta = \begin{cases} \mathrm{sgn}(\phi_i) \left\{ \dfrac{\alpha_{MAX}(i)\ (|\ \phi_i\ | - \phi^*)}{\phi^* - \pi} - \lambda_i \right\} & , \text{ if } |\ \phi_i\ | > \phi^* \ , \\ - \mathrm{sgn}(\phi_i)\ \lambda_i & , \text{ otherwise} \ . \end{cases} \qquad (C.2\text{-}2)$$

Note that in the generalized form, the largest lag or lead angle is not $|\ \alpha_{MAX}(i)\ |$ but $|\ \alpha_{MAX}(i) + \lambda_i\ |$.

C.3 <u>Implications of the DEL Pursuit Course</u>

The final form of the DEL Pursuit Course is rather general. For example, a pure pursuit course will be flown by aircraft i if $\alpha_{MAX}(i)$ and λ_i are both set to zero, thus giving η as zero. Also, a deviated pursuit course (with constant deviate angle) may be specified by setting $\alpha_{MAX}(i)$ to zero and λ_i to the desired deviate angle. However, the usual situation is to set $\lambda_i = 0$ so that at, say, $\phi_i = \pm\pi$ the aircraft will deviate by setting $|\ \alpha_i\ | = \alpha_{MAX}(i)$ and tend to swing around behind its enemy, linearly decreasing its tracking angle down to zero at $\phi_i = \pm\ \phi^*$. Thereafter aircraft i will fly a pure pursuit course, setting $\alpha_i = 0$.

Since $\alpha_{MAX}(i)$ is a parameter of the function defining the tracking angle at all points along the DEL Pursuit Course, its magnitude influences how well the attacker will get to the rear of its enemy; whether or not the objective is obtained. In Figure C.3-1 various DEL Pursuit Course paths are drawn against a non-maneuvering target, for different values of $\alpha_{MAX}(i)$. These paths are drawn in relative (r, ϕ) coordinates where r corresponds to the range between the aircraft and ϕ the angle-off. Of course, the initial point and relative velocities influence the shape; also, the

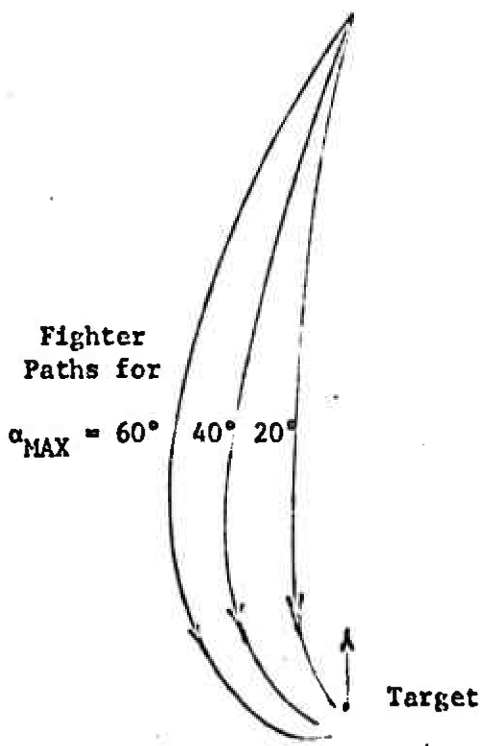


Figure C.3-1 DEL Pursuit Course Paths for Different Values of $\alpha_{MAX}(i)$

conclusion is that increasing the value of $\alpha_{MAX}(i)$ tends to force the path out and away from the target. The path is longer and takes more time to execute, but at any given time, points on the paths with larger values of $\alpha_{MAX}(i)$ are further from the target.

An interesting implication of the DEL Pursuit Course is seen as one considers what happens when both aircraft fly such courses. Suppose then that they do both fly DEL Pursuit Courses specified by (C.2-1) for $i = F, B$. The notation may be abbreviated to require that α_i be a linear function of ϕ_i for $i = F, B$. Then

$$\alpha_F = K_F \phi_F + C_F$$

and

$$\alpha_B = K_B \phi_B + C_B \quad ,$$

where K_F, C_F, K_B and C_B are constants. Then differentiating with respect to time gives

$$\dot{\alpha}_F = K_F \dot{\phi}_F$$

and

$$\dot{\alpha}_B = K_B \dot{\phi}_B \quad .$$

Since α_i and ϕ_j are negative supplements of each other

$$\dot{\alpha}_B = \dot{\phi}_F$$

and

$$\dot{\alpha}_F = \dot{\phi}_B \quad .$$

hence

$$\dot{\alpha}_F = K_F \dot{\alpha}_B = K_F K_B \dot{\phi}_B$$

$$= K_F K_B \dot{\alpha}_F \qquad .$$

And from this it is concluded that either $K_F K_B = 1$ or $\dot{\alpha}_F = \dot{\alpha}_B = 0$. But K_F and K_B are chosen. Hence, whenever $K_F K_B \neq 1$ then $\dot{\alpha}_F = \dot{\alpha}_B = 0$ and the aircraft fly at constant tracking angles whenever they are both flying DEL Pursuit Courses. Thus, if both aircraft have the turning capability to fly the DEL Pursuit course, both aircraft are thwarted from reducing their angles-off.

C.4 Derivation of ϕ^*

This section contains a brief derivation of the angle ϕ^*. This angle is used as a parameter of the DEL Pursuit Course function. It is defined as the angle of maximum lateral g's of a pursuer flying pure pursuit. It assumes a constant pursuer velocity and a linear non-maneuvering target with constant velocity greater than half the pursuer's velocity. Suppose F is pursuing B with pure pursuit at constant velocity $V_F < 2 V_B$. Also suppose that $\dot{\beta}_B = 0$. Since F is on pure pursuit $\dot{\alpha}_F = \alpha_F = 0$. Hence

$$\dot{\theta} = \frac{-V_B}{R} \sin \phi_F \qquad ,$$

$$\dot{R} = V_B \cos \phi_F - V_F \qquad ,$$

and

$$\dot{\beta}_F = \dot{\theta} \qquad .$$

Further, since F is at constant velocity, the point of maximal lateral g's is also the point of maximal turning rate, $\dot{\beta}_F$. And a necessary condition for $\dot{\beta}_F$ to be maximal is that $\ddot{\beta}_F = 0$. And

$$\ddot{\beta}_F = \ddot{\theta} = \frac{\partial\dot{\theta}}{\partial R}\frac{dR}{dt} + \frac{\partial\dot{\theta}}{\partial\phi_F}\frac{d\phi_F}{dt}$$

$$= \frac{V_B}{R^2}\sin\phi_F\,(V_B\cos\phi_F - V_F) + \frac{V_B^2}{R^2}\sin\phi_F\cos\phi_F$$

$$= \frac{V_B}{R^2}\sin\phi_F\,(2V_B\cos\phi_F - V_F) \quad .$$

And by definition

$$\frac{d\dot{\theta}}{dt}\bigg|_{\phi_F = \phi^*} = 0 \quad ,$$

hence there are two roots; namely $\phi^* = 0$ and $\phi^* = \mathrm{Cos}^{-1}(V_F/2V_B)$. The point of $\phi_F = 0$ is also the point of zero range, except for the special case in which the fighter started pursuit at $\phi_F = 0$ -- an uninteresting situation. Thus, the applicable non-zero root is

$$\phi^* = \mathrm{Cos}^{-1}(V_F/2V_B) \quad .$$

Note that only necessary conditions have been given; sufficient conditions may be demonstrated by showing that this is an absolute maximum. Also, the case of $V_F \geq 2V_B$ has not been dealt with. For this condition it can be shown that the lateral g's of the pursuer increase without bound. Hence, no ϕ^* exists for this case. In that case, ϕ^* is set to zero.

APPENDIX D

TURNING RATE

The turning rate of an aircraft, $\dot{\beta}$, is the time rate of change of the heading angle, the direction of travel. As such it is related to the g's and speed of the aircraft. From equation (4.6-2) an aircraft at speed V pulling G total g's will turn at an angular rate of

$$\dot{\beta} = (32.2/V)\ (G^2 - 1)^{1/2} \quad .$$

But the maximum g's an aircraft can sustain are in general a function of speed described in Appendix E. Hence, the maximum turning rate that an aircraft can sustain is also a function of speed. Now, throughout the development of ATAC-2 various considerations have led to the conclusion that this $\dot{\beta}$ function of speed is a better measure of performance and capability than is the g function of speed. Some of these considerations are now given.

To show the importance of the turning rate capability of an aircraft, a rather contrived situation is considered first. This is an extreme example. Suppose two dissimilar aircraft have agreed to a duel under the "old code". They start back to back, fly linearly for say 10 seconds, (at possibly differing speeds) then at the same time start to turn, and finally fire their missiles, (see Figure D-1). Suppose also that each has the same weapon. The weapon may be fired at any range but with an angular constraint. The firing aircraft must point at its target to within say plus or minus 5 degrees to fire. Further suppose that each weapon has a zero time of flight and constant kill probability. Then, since each aircraft

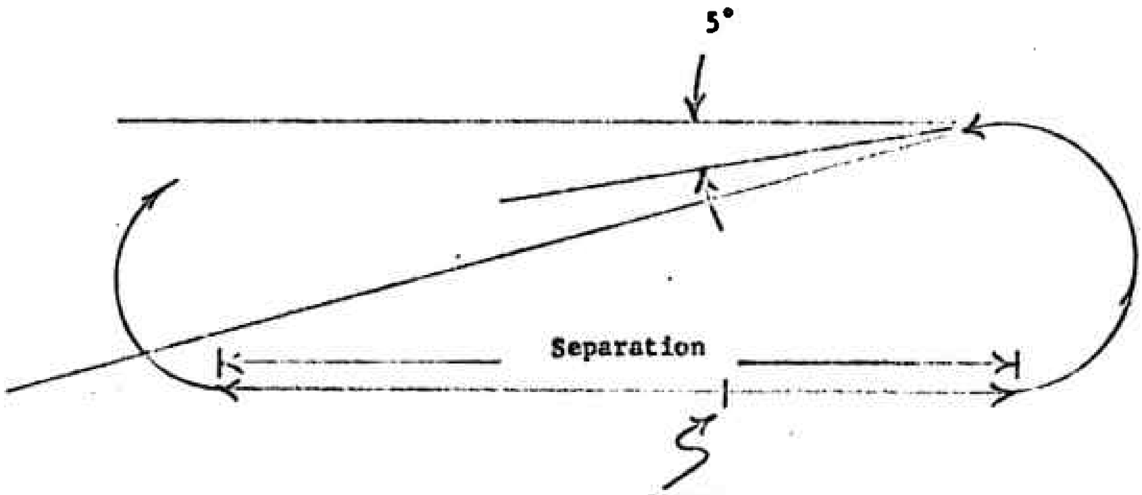


Figure D-1 A Duel

starts its turn at the same time, the aircraft that turns the quickest fires first and thereby wins the duel. It matters little how many g's each pulls but rather the time it takes to turn through about 180 degrees.

Indeed this is a contrived situation. However, consider what happens without the agreement to duel by the code. What happens after each aircraft has passed its opponent head-on? Each aircraft tries to turn around and fire at its opponent before being fired upon. Then the distance between the centers of the combatant's circles of turn has decreased from the above example. However, the situation is quite analogous. And again the objective of each combatant is to turn as fast as possible so as to point near its opponent first. To be sure the situations are not completely similar. However, closer analysis will show that for a vast range of relative parameters, it is the aircraft that turns the quickest that points at its enemy first. This then is one reason turning rate is an important consideration of performance. An aircraft desires to turn as fast as possible, irrespective of from what initial angle it started, so as to point near its target and be able to fire weapons.

Another reason for the desire of high turning rate is general maneuvering. In Appendix F it is shown that one of the necessary conditions for one aircraft to follow another in steady state while on pure pursuit is that it turns as fast as the target - that $\dot{\beta}$ of the pursuer equals $\dot{\beta}$ of the evader. This requirement holds even if the pursuer is allowed to fly a deviated pursuit course (one in which its tracking angle is a constant not necessarily zero). It must still be able to turn as fast as its opponent in order to hold its range and angle-off constant. Further, it is seen that the pursuer in steady state on pure pursuit pulls less g's than the evader at all non-zero steady state ranges. From the evader's standpoint it is also the case that high turning rate helps. For the greater the disparity between the turning rate of the evader over the pursuer, the sooner will the evader extricate itself from this unhealthy position of having the enemy on its tail. Merely pulling more g's than the pursuer does not necessarily help its situation.

Thus, it is for these and other reasons that one is led to consider high turning rate capability as beneficial and the $\dot{\beta}$ function of speed as an important indicator of performance. And this then is the justification for many of the tactics of the model that require an aircraft in certain situations to tend to V^*, the speed at which the sustainable turning rate is an absolute maximum.

APPENDIX E

SPECIFIC POWER FUNCTION

In general, the performance of an aircraft is thought to ba its ability to climb and dive, to turn, to speed up and slow down as well as its maximum speed, etc. Further, most all of these abilities are inter-related; the ability to climb is a function of lateral g level, the ability to turn is a function of speed, the maximum speed is a function of altitude, and so on. Yet there is a concept which helps unify many of these abilities; this is the concept of specific power, P_S . The literature abounds with various treatments of specific power, see for example Boyd [Ref. 4], so that only a brief description will be given here. The method in which ATAC-2 makes use of the concept of specific power, however, is discussed in detail.

The specific[1)] power of an aircraft is defined to be the time rate of change of the specific energy level. This includes the energy due to altitude (potential) and the energy due to speed (kinetic). Thus, for an aircraft at altitude h and velocity V the specific energy is given by

$$E_S = h + V^2/(2 \cdot 32.2)$$

then

$$P_S \equiv d\,E_S/dt = dh/dt + (V/32.2)\,dV/dt \quad \text{(E-1)}$$

---

1) Throughout the term specific is with respect to the weight of the aircraft.

Then specific power is composed of two components; the power of altitude change and the power of speed change. An aircraft at a positive specific power level will increase its energy level. It may do this by linearly accelerating ($dh/dt = 0$) at a rate of $dV/dt = P_S (32.2/V)$, for example. Or the aircraft could accomplish this by climbing at constant speed ($dV/dt = 0$) at a rate of $dh/dt = P_S$. In fact the aircraft could combine the two (climbing and increasing its speed) in any manner that is consistent with (E-1). A completely analogous situation applies for a negative specific power. Here the aircraft will lose energy. This may be accomplished by losing altitude in any combination with losing speed again consistent with (E-1). Thus, the specific power of an aircraft specifies the rate of change of the energy level of the aircraft: The rate of change of the altitude and of the speed at a given apeed.

By definition then the specific power of an aircraft has units of power (ft-lbs/sec) divided by weight (lbs) or ft/sec. This is reflected in (E-1) for dh/dt and $(V/32.2)\ dV/dt$ also have units of feet per second.

One might logically question then how the specific power function of an aircraft is characterized. How is it specified? [Ref. 4] shows that for an aircraft with thrust T, drag D, weight w and speed V the level of specific power is

$$P_S = (T - D)\ V/w \quad . \qquad \text{(E-2)}$$

Thus to specify the specific power of sn aircraft, knowledge of engine and aerodynamic properties at various speeds is required. Now the thrust of an aircraft is a function of the speed, altitude and power setting. Also the drag of an aircraft is a function of speed, altitude and lateral g's. Consequently, the specific power of an aircraft is a function of nearly every other possible state parameter, namely, speed, power setting, altitude and lateral g's. At a fixed power setting and a fixed altitude specific power is a function of speed and lateral g's. This is built into the ATAC-2 model. The aircraft is assumed to be at a fixed power setting and at a constant altitude. Henceforth, then the specific power function is taken to be a function of speed and lateral g's.

An example of the specific power function of a fictitious, though not atypical, aircraft is shown in Figure E-1. Each line shown gives the value of the specific power function along the ordinate for the associated speed along the abscissa, and total g level labeled. (The total g's of an aircraft are the lateral g's plus the g due to gravity, added vectorially.) Also associated with this diagram is a fixed altitude and power setting. Now for a 1g level the function is positive at all speeds shown. Thus, the aircraft will accelerate at this constant altitude. However, for higher g levels, say about 3.5 g's, the specific power function is negative for all speeds indicated. Thus, the aircraft will decelerate. From equation (E-1) it is seen that the acceleration of the aircraft at this constant altitude is given by 32.2 P_S/V. And if the function is negative the aircraft will decelerate, while it will gain speed if the function is positive.

More information than just the acceleration capability of an aircraft may be obtained from this diagram. The maximum speed V_{max}, is just the limit of the lines indicated on the right in Figure E-1. This limit at

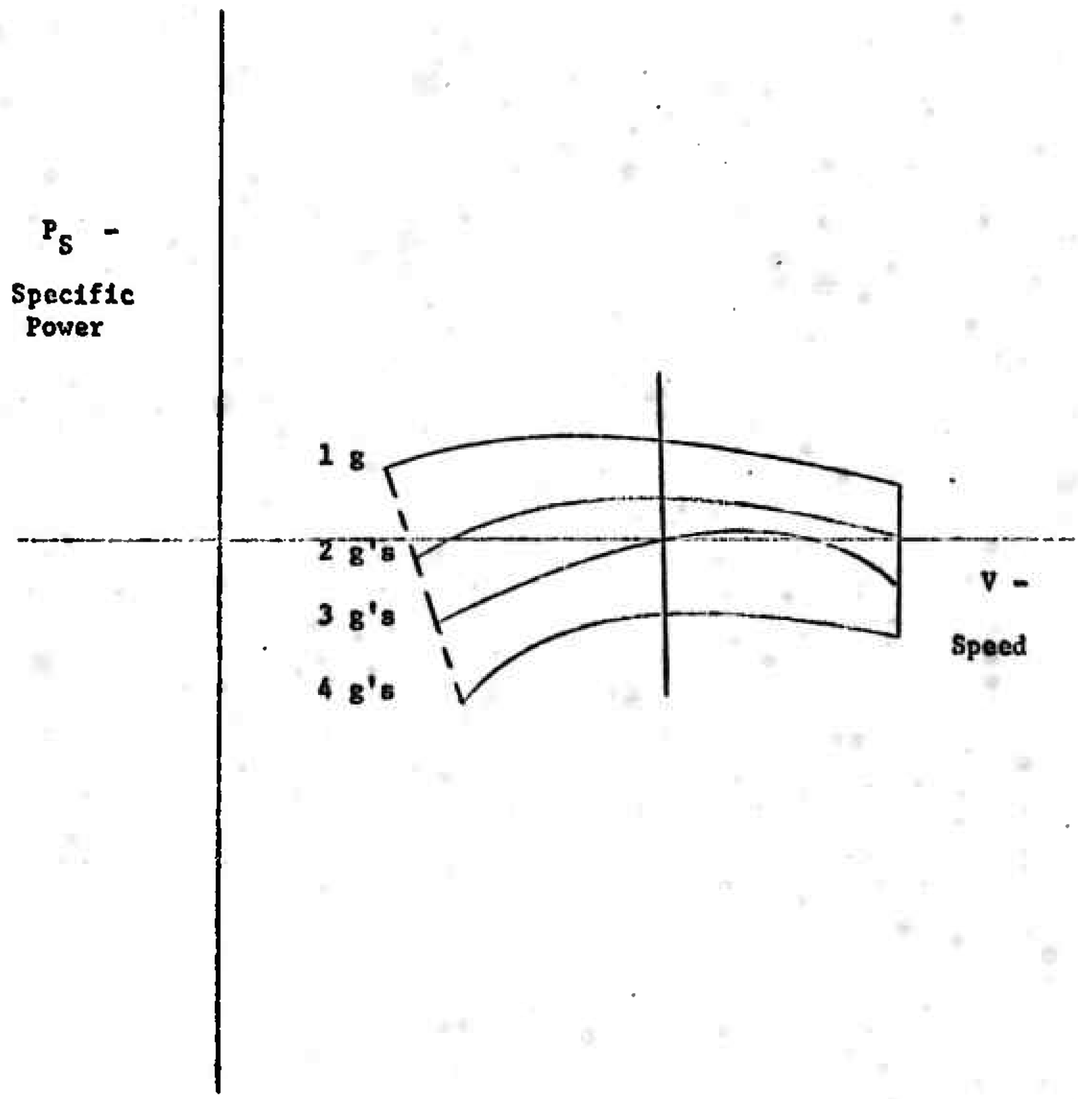


Figure E-1 A Specific Power Function of Velocity and g's for an Aircraft at Constant Altitude and Power Setting this altitude is a heat limit. At some other altitude, however, the maximum speed may be a power limit. The minimum speed V_C is given by the extent of the lines on the left as shown. Also, the maximum g's that the aircraft may sustain is indicated by the extent of the g lines. This is the limit to which the aircraft's structure is able to sustain g's. In this example the limit is given by 4 g's. Also, more noteworthy information is available from this specific power function. The dotted line at the left indicates the limit of the speed an aircraft may fly at while at a given g level. This is the stall limit; if the aircraft flies at less than this speed, it begins to lose its lift, starts to buffet and finally stalls. Yet the concept may be viewed from the other way also, namely, as the maximum amount of g's the aircraft may sustain at a given speed without stalling. Then the whole lower line (dotted on the left, solid otherwise) may be viewed as the maximum total g's the aircraft may attain. This g limit is itself a function of speed as indicated. The symbol used for this function is G(V) .

There is one important concept which has yet to be discussed. This is the concept of the g(V) function. To illustrate this concept an example is useful. Consider an aircraft at some speed and 1 total g. Suppose the aircraft begins to pull more and more g's so that it continues to lose specific power (see E-1). As the aircraft continues to pull more g's it passes a singular transition point. This is the point at which the specific power function is zero; the point where the function changes sign. Clearly an aircraft that pulls more g's at that speed will lose speed. At the constant speed vertical line this value is about 3.0 g's. At other velocities the relevant value is, of course, different. Thus, it is a function of velocity. Hence g(V) is the total g's as a function of speed, V , at which the specific power function is zero. This may be viewed as the sustainable g's at the given speed; for the aircraft will neither accelerate nor decelerate at g(V) total g's. Then, if an aircraft wishes to pull its maximum g's at, say, speed V , and not decelerate, it can pull no more than g(V) total g's.

In summary then the specific power function, indeed, provides a great deal of knowledge about the performance of the aircraft. The value of the function itself gives the rate at which the aircraft will accelerate at some g level and velocity. Also, the limits of the function give the maximum velocity and the maximum total g's the aircraft may achieve. Finally, the g(V) function is obtained from the specific power function; it is the g level at which the velocity V is just maintained.

APPENDIX F

STEADY STATE CONDITIONS

This appendix gives the derivation of the necessary steady state conditions. Also, the applicability of the steady state conditions to ATAC-2 is discussed.

These conditions are applicable when one aircraft follows another that turns. One aircraft tries to maneuver by turning while the other tries to stay behind it. Under this situation then, suppose F is following B in the steady state defined to be the condition of constant range, $\dot{R} = 0$, and constant angle-off $\dot{\phi}_F = 0$. Since the fighter is assumed on pure pursuit, $\dot{\alpha}_F = 0$. Then from (B.3-2), (B.3-3), and (B.3-4) it is seen that

$$\left.\begin{aligned} \dot{\phi}_F &= 0 \Longrightarrow \dot{\beta}_F = \dot{\theta} \\ \dot{\alpha}_F &= 0 \Longrightarrow \dot{\beta}_B = \dot{\theta} \end{aligned}\right\} \Longrightarrow \dot{\beta}_F = \dot{\beta}_B \qquad \text{(F-1)}$$

and

$$R = \frac{V_B \sin \phi_F}{\dot{\beta}_B} \quad . \qquad \text{(F-2)}$$

From (B.3-1),

$$\dot{R} = 0 \Longrightarrow V_F = V_B \cos \phi_F \quad . \qquad \text{(F-3)}$$

These then are the necessary conditions for F to follow B in steady state on pure pursuit.

To observe some of the singularities of this situation consider Figure F-1. Suppose that F desires to follow B in steady state at a range R' and angle-off ϕ' and a speed V_F'. Then among the three parameters V_F', ϕ', and R' the fighter is free to pick only one of these, for the other two will be determined by this choice. For example, suppose ϕ' is chosen. Then (F-3) gives the V_F' associated with this ϕ' while (F-2) gives the R'; assuming, of course, that the bomber continues at V_B and $\dot{\beta}_B$.

Yet even this choice of one of the three parameters is limited. It is seen from Figure F-2 that the fighter's speed has an upper bound of V_B; it can fly no faster than the target in steady state. Also, if $V_C(F)$ is the minimum speed of the fighter then there is a definite upper bound to ϕ' given by $\cos^{-1}(V_C(F)/V_B)$. And it is seen from Figure F-3 that an upper bound on R' is the bomber's radius $r_B = V_B/\dot{\beta}_B$. So there are definite limits to the choice of R', ϕ' and V_F'.

Further considerations of the above conditions influence the tactics of the model. Since $\dot{\beta}_F$, the turning rate of the fighter, is bounded by the maximum turning rate (determined by the speed), the bomber can prevent the fighter from obtaining a steady state condition by making $\dot{\beta}_B$ larger than this bound. For example, suppose two identical aircraft engage each other. Also suppose that the implied $\dot{\beta}$ function of speed for both aircraft has an absolute maximum at V^*. Then if the bomber flies at speed V^*, turning at its maximum rate, the fighter would not be able to obtain a steady state at any non-zero range.

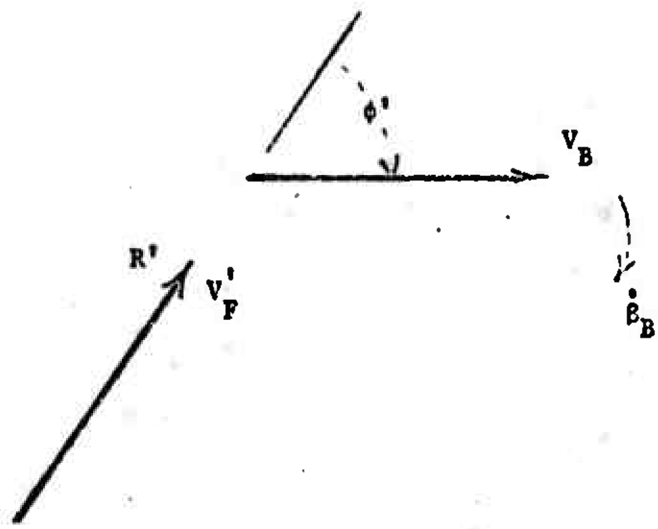


Figure F-1 Steady State Parameters

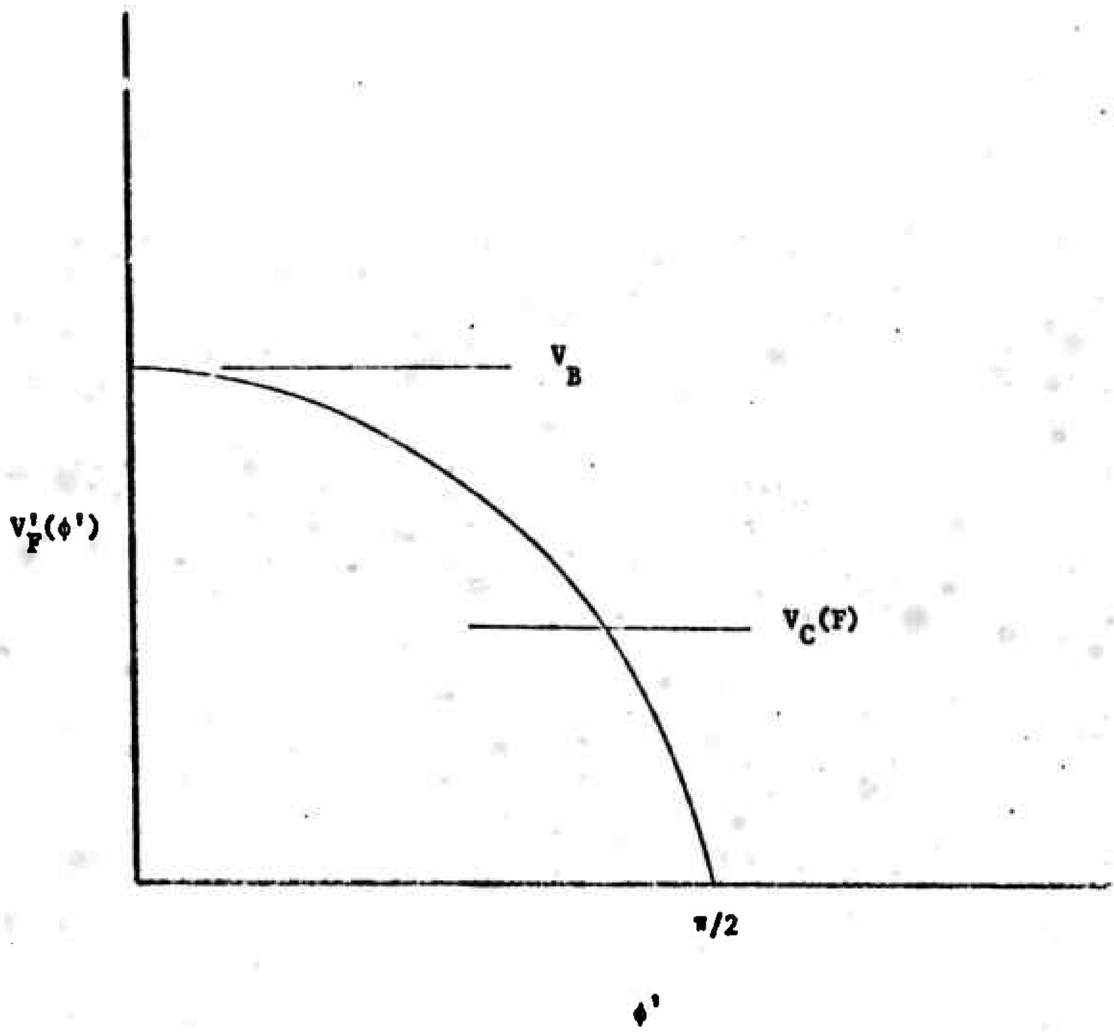


Figure F-2 Steady State Velocity as Function of Angle-Off

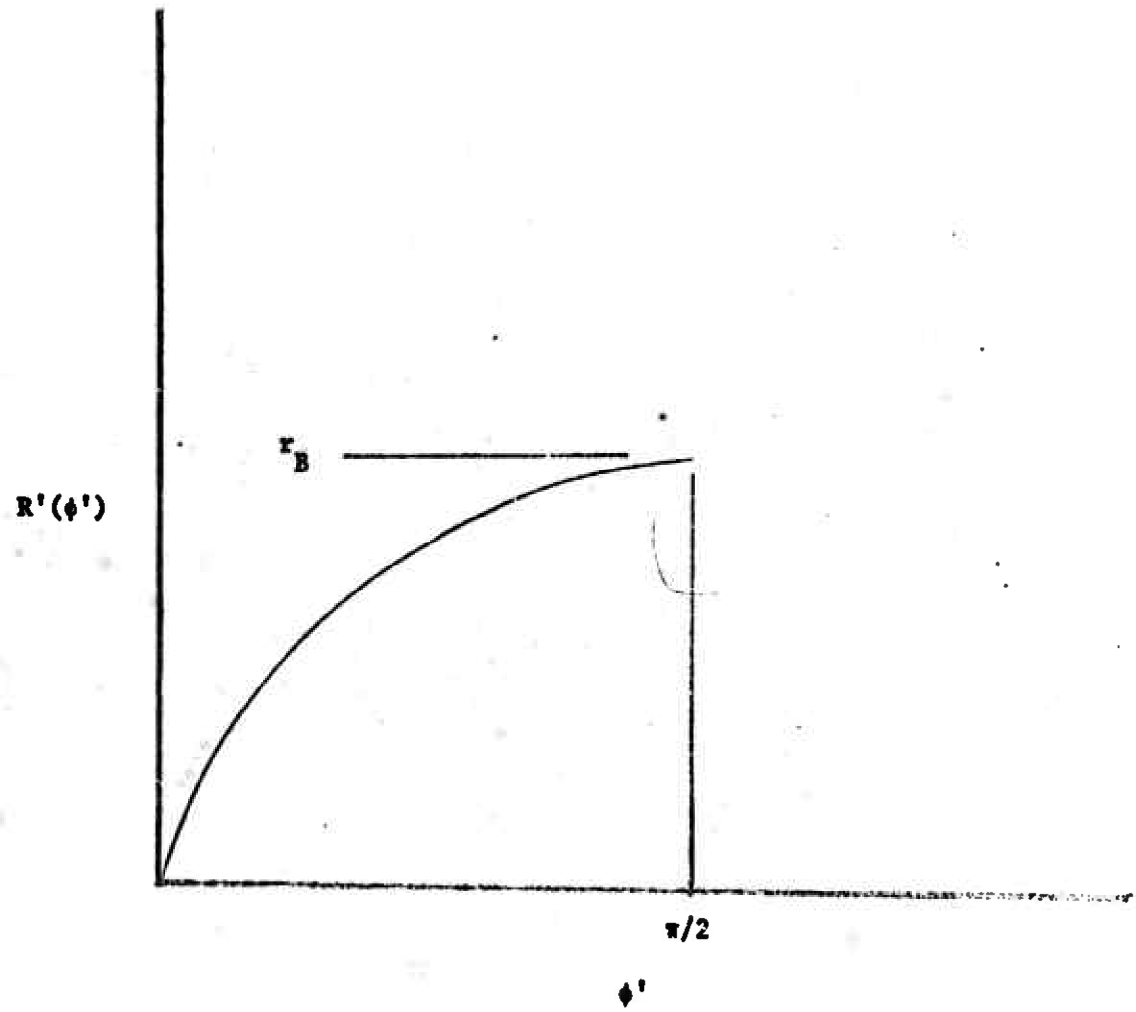


Figure F-3 Steady State Range as Function of Angle-Off

Of course, similar considerations may be made with dissimilar aircraft by comparing their $\dot{\beta}$ functions of speed. In fact, one can determine before hand which, if either, of two combatsnts is able to follow another in the steady state. Further, the range of the parameters R', ϕ', V_P' which will accomplish a steady state can be evaluated, by noting the $\dot{\beta}$ function of speed, and applying equations (F-1), (F-2), snd (F-3).

A word of caution may be in order here. To say that the steady state is unattainable for one aircraft or another is indeed not to imply that one is inferior. For the transient state or near steady state may be quite easily attained and is often quite sufficient in order to remain behind s target for a long time.

LIST OF REFERENCES

[1] Cerone, J. R., T. J. Harris, B. J. Jacobs, W. J. Strauss, Fighter Vs. Fighter Combat: ATAC-2 Model, Volume I, II, Caywood-Schiller, Associates, October, 1966.

[2] Harris, T. J., B. J. Jacobs, W. J. Strauss, Fighter Vs. Fighter Combat: ATAC-2 Model: Double Search, CSA Report Nr. 67-R-102, Caywood-Schiller, Associates, September, 1967.

[3] Personal Communications, Dr. Hitchcock, U. S. Naval Air Development Center, Johnsville, Pennsylvania, March, 1967.

[4] Boyd, J. R., T. P. Christie, J. E. Gibson, Energy-Maneuverability (U), Air Force Systems Command, Eglin Air Force Base, APGC-TR-66-4, Volume 1, March, 1966 (Secret).

[5] Rutowski, E. S., Energy Approach to the General Aircraft Performance Problem, J. Aero. Sci., Volume 21, No. 3, March, 1954.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
Deputy for Development Planning (ASBEE-10)
Aeronautical Systems Division
Wright-Patterson, AFB, Ohio 45433

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP
N/A

3. REPORT TITLE
Air-to-Air Combat Model Program & Appendices Technical Details

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Progress Report (Oct 66 - Nov 67)

5. AUTHOR(S) (First name, middle initial, last name)
Terrell J. Harris, Barton J. Jacobs, Walter J. Strauss

6. REPORT DATE
November 1967

7a. TOTAL NO. OF PAGES
201

7b. NO. OF REFS
5

8a. CONTRACT OR GRANT NO.
F-33-(615)-3707

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
CSA Report 67-101 Vol III, Vol IV

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT
Number 1: Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.

HERMAN WOOD
STINFO Officer (ASBO-?)
Management Operations Division
Deputy for Development Planning

11. SUPPLEMENTARY NOTES
See Also: Air-to-Air Combat Model Description & Development General Information.

12. SPONSORING MILITARY ACTIVITY
Cost-Effectiveness Branch (ASBEE-10)
Operations Research Section
Deputy for Development Planning

13. ABSTRACT

ATAC-2 is a simulation model designed to help evaluate fighters in air-to-air combat. The model treats the one vs. one dogfight which arises from a random search situation. Both aircraft in the combat are (usually) aggressive. The two principal outputs from the model are the probability a given aircraft is killed in the fight and the expected number of enemy aircraft an aircraft kills over its useful life. Combat is restricted to a fixed altitude. The maneuvers are dynamic in that each aircraft responds to the situation at each moment in a duel depending on the information it has about an opponent's activities. (U)

Inputs include, for each aircraft, search and tracking radar characteristics, passive radar sensors, optical capability, IFF, energy-maneuverability data, weapon loadings, weapon characteristics, and weapon kill probabilities. (U)

The rationale for the model specifics are presented. Flow charts and program listings are included. The model has been run repeatedly on an IBM 7094. (U)

DD FORM 1 NOV 65 1473

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
OPERATIONS RESEARCH						
SIMULATION						
AIR-TO-AIR COMBAT MODEL						
COST EFFECTIVENESS						
TACTICS						
DYNAMIC MODEL						