NRL Technical Memorandum 7595-021
September 9, 1986

ABSTRACT INTERFACE SPECIFICATIONS FOR THE PHYSICAL MODELS MODULE

by

Bruce G. Labaw

Naval Research Laboratory
Washington, D. C.

ABSTRACT

CHAPTER 1

PM.ACM: AIRCRAFT MOTION

1. Introduction

This submodule of the Physical Models module contains programs that model aircraft motion. Aircraft location, velocity, and attitude with respect to the earth and airmass are derived from measures of aircraft motion from devices and other physical models modules.

The airframe coordinate system is a set of orthogonal axes stationary with respect to the body of the aircraft. The Y axis positive direction is forward (toward nose, from tail), the positive X axis points in the direction of the right wing, and the positive Z axis points from the aircraft center towards the top of the a/c. These are referred to as the Ya, Xa and Za axes.

2. Interface overview

2.1. Access Program Table

Program	*Parameters*	*Description*	*Undesired events*
The following programs have the same UE,			%%acm src not init%%
+G_AC_LOCN+	p1: =earth_locn=; O	!+a/c locn computed+!	
+G_AC_ALT+	p1: =AT.distance=; O	!+a/c alt asl+!	
+G_AC_VSPD_AIRSPD+	p1: =AT.speed=; O	!+v spd fm airspd+!	
+G_AC_VSPD_ALT+	p1: =AT.speed=; O	!+v spd fm alt+!	
+G_AC_VSPD_GNDSPD+	p1: =AT.speed=; O	!+v spd fm gndspd+!	
+G_AC_HVEL_AIRSPD+	p1: =AT.angrate_vec=; O	!+horiz veloc fm airspd+!	
+G_AC_HVEL_GNDSPD_1+	p1: =AT.angrate_vec=; O	!+horiz veloc fm gndspd 1+!	
+G_AC_HVEL_GNDSPD_2+	p1: =AT.angrate_vec=; O	!+horiz veloc fm gndspd 2+!	
+G_AC_HVEL_INCR+	p1: =AT.angrate_vec=; O	!+horiz veloc incr+!	
+G_AC_HVEL_LEVER+	p1: =AT.angrate_vec=; O	!+horiz veloc fm lever+!	
+G_AC_PITCH_RATE+	p1: =AT.angrate=; O	!+pitch rate+!	
+G_AC_ROLL_RATE+	p1: =AT.angrate=; O	!+roll rate+!	
+G_AC_HDG_RATE+	p1: =AT.angrate=; O	!+hdg rate+!	
+G_AC_NORMAL_ACCEL+	p1: =AT.accel=; O	!+normal accel+!	

+G_AC_GNDTRK_ANG+	p1: =AT.angle=; O	!+gnd trk angle+!
+G_AC_DRIFT_ANG+	p1: =AT.angle=; O	!+drift angle+!
+G_AC_AIR_VEL+	p1: =AT.velocity=; O	!+a/c airmass veloc+!
+G_AC_SPD_PU+	p1: =AT.speed=; O	!+a/c speed indic accel+!
+G_AC_V_DISP_PU+	p1: =AT.distance=; O	!+a/c vert disp indic accel+!
+G_AC_H_DISP_PU+	p1: =AT.distance=; O	!+a/c horiz disp indic accel+!
+G_AC_PTH_APPRCH+	p1: =AT.distance=; O	!+a/c path approach+!
+G_AC_V_DISP+	p1: =AT.distance=; O	!+a/c vert disp const vel+!
+G_AC_H_DISP+	p1: =AT.distance=; O	!+a/c horiz disp const vel+!
+G_AC_RHOX+	p1: =AT.angrate=; O	!+a/c rate East+!
+G_AC_RHOY+	p1: =AT.angrate=; O	!+a/c rate North+!
+G_AC_RHOZ+	p1: =AT.angrate=; O	!+a/c rate Up+!

2.2. ACCESS PROGRAM TABLES, INPUT

2.2.1. A/C INPUT, SOURCE SETTING

Each "#" of the input programs is replaced by an integer. Each +S_..._SRC+ program and ++S_..._INIT++ program affect the !!acm input!! with the same integer replacement: the set of legal !!acm input!!s defines the set of integers allowed in each program name. An "*" is used when referring to access name substrings of any of the following list: "AC_LOCN", "AC_ALT", "AC_H_VEL", "AC_V_VEL", "AC_T_HDG", "AC_PITCH", "AC_SPEED", "AC_ROLL", "MOM_ARM", "RATIO", "AC_FPA", "AC_AOA", "AC_AIRSPD", "AC_GNDSPD", "AC_DRIFT", "B_TO_P", "DISP", or "FLT_TIME".

Program	*Parameters*	*Description*	*Undesired events*
++S_*_#_INIT++	p1: =name=; I p2: =boolean=; I	!!src acm!! !!acm runtime change!!	%%acm already init%%
+S_*_#_SRC+	p1: =name=; I	!!src acm!!	%%acm src not init%% %%acm chg method%% %%itm in acmslist%%
++SAME_*++	p1: =name=; I p2: =acmslist=; I	!!src acm!!	%%acm already init%% %%err acmslist%%

%%acmslist not init%%
%%err acmslist%%

+S_*_ACM_SL_SRC+ p1: =name=; I !!src acm!!
p2: =acmslist=; I

Parameters

The input parameters of type "name" may be the name of a literal, constant, variable, or DB.dbitem. The typeclass of the entity or DB.dbitem, and a more complete description of the parameter p1 are given by the following tables.

Programs	Typeclass of p1	P1 description
++S_*_#_INIT++,		
+S_*_#_SRC+		
++SAME_*++, and		
+S_*_ACM_SL_SRC+ where		
"*" is replace by -		
"AC_LOCN"	earth_locn	!!src acm!! of !!a/c locn #!!
"AC_ALT"	AT.distance	!!src acm!! of !!a/c alt asl #!!
"AC_H_VEL"	AT.velocity	!!src acm!! of !!a/c h vel #!!
"AC_V_VEL"	AT.speed	!!src acm!! of !!a/c v vel #!!
"AC_T_HDG"	AT.angle	!!src acm!! of !!a/c t hdg #!!
"AC_PITCH"	AT.angle	!!src acm!! of !!a/c pitch #!!
"AC_SPEED"	AT.angle	!!src acm!! of !!a/c speed #!!
"AC_ROLL"	AT.angle	!!src acm!! of !!a/c roll #!!
"B_TO_P"	AT.orientation	!!src acm!! of !!body to platform #!!
"MOM_ARM"	AT.displacement	!!src acm!! of !!moment arm #!!
"RATIO"	real	!!src acm!! of !!ratio #!!
"AC_FPA"	AT.angle	!!src acm!! of !!a/c fpa #!!
"AC_AOA"	AT.angle	!!src acm!! of !!a/c aoa #!!
"AC_AIRSPD"	AT.speed	!!src acm!! of !!a/c airspd #!!
"AC_GNDSPD"	AT.speed	!!src acm!! of !!a/c gnd spd #!!
"AC_DRIFT"	AT.angle	!!src acm!! of !!a/c drift a #!!
"DISP"	AT.displacement	!!src acm!! of !!displacement #!!
"DIST"	AT.distance	!!src acm!! of !!dist #!!
"FLT_TIME"	timeint	!!src acm!! of !!a/c flt time #!!

++S_..._#_INIT++
+S_*_#_SRC+ The set of legal replacements for "#" is the same as that set forth in the definition of terms as given in the table below:

Programs	Term with the same "#" subset
"AC_LOCN"	!!a/c locn #!!
"AC_ALT"	!!a/c alt asl #!!
"AC_H_VEL"	!!a/c h vel #!!
"AC_V_VEL"	!!a/c v vel #!!
"AC_T_HDG"	!!a/c t hdg #!!
"AC_PITCH"	!!a/c pitch #!!
"AC_SPEED"	!!a/c speed #!!
"AC_ROLL"	!!a/c roll #!!
"B_TO_P"	!!body to platform #!!
"MOM_ARM"	!!moment arm #!!
"RATIO"	!!ratio #!!
"AC_FPA"	!!a/c fpa #!!
"AC_AOA"	!!a/c aoa #!!
"AC_AIRSPD"	!!a/c airspd #!!
"AC_GNDSPD"	!!a/c gnd spd #!!
"AC_DRIFT"	!!a/c drift a #!!
"DISP"	!!displacement #!!
"DIST"	!!dist #!!
"FLT_TIME"	!!a/c flt time #!!

Effects

++S_..._#_INIT++

++S_..._#_INIT++ disables %%acm src not init%% for the !!acm input!! corresponding to the program name.

If the !!acm runtime change!! is false, %%acm chg method%% is enabled. Values for the !!acm input!! are taken from p1 until an invocation of +S_..._#_SRC+ if !!acm runtime change!! is true, otherwise throughout the life of the program. %%acm already init%% is enabled for the !!acm input!!.

+S_..._#_SRC+

Until another invocation of this program the module will use p1 as the !!src acm!! for the !!acm input!!.

++SAME_...++

This program declares that the sources of information for the !!acm input!! members of the =acmslist= are always the same entity. Until +S_..._ACM_SL_SRC+ is invoked with p2, the source of information for all the members of p2 is p1. The !!acm runtime change!! state of each member of the sourcelist is true. %%itm in acmslist%% is enabled for each !!acm input!! in p2. %%acmslist not init%% is disabled for p2.

+S_..._ACM_SL_SRC+

Until another invocation of this program with the same p2 the module will use p1 to obtain an updated value for each !!acm input!! of p2.

2.3. VALUE SETTING PROGRAMS

Program	*Parameters*	*Description*	*Undesired events*
+RESET_MODEL_H_V+			

Effects

+RESET_MODEL_H_V+

The model computing !+horiz veloc incr+! is reset.

3. Local type definitions

=acmslist= Simple enumerated: a list (enclosed in parentheses, whitespace separator) of a set of !!acm input!! names that are members of the same term class in PM.ACM.4.2.1.

4. Dictionary

4.1. Dictionary of !+term+!s

All velocities are with respect to the earth unless otherwise stated. Angles are defined by use of a template listing the directions of two vectors and the sense of the angle measurement.

!+a/c airmass veloc+! The velocity of the aircraft with respect to the airmass. The aircraft airspeed, angle of attack, and airframe system to IMS system orientation difference are !!a/c airspd 3!!, !!a/c aoa 3!! and !!body to platform 1!!

!+a/c alt asl+! The a/c !!altitude asl!!. Previous !!altitude asl!! was !!a/c alt asl 1!!, with vertical velocity equal to !!a/c v vel 1!!.

!+a/c horiz disp const vel+! The !!a/c horiz disp!! of the a/c during a flight segment in which a horizontal speed of !!a/c gnd spd 5!! is maintained for !!a/c flt time 2!!.

!+a/c horiz disp indic accel+! The !!a/c horiz disp!! of the a/c during a flight segment beginning with the a/c speed of !!a/c speed 6!! and with an !!a/c pitch!! of !!a/c pitch 12!!, maintaining #max indic normal accel# during the flight segment and ending when the !!a/c pitch!! equals !!a/c pitch 10!! and the a/c speed is !!a/c speed 4!!.

!+a/c locn computed+! The current location of the a/c. The previous location of the a/c was !!a/c locn 1!!, and !!a/c h vel 1!! has been the a/c horizontal velocity.

!+a/c path approach+! The closest distance from a flight path segment to a calculated position (p1) . The flight path segment begins with the aircraft speed of !!a/c speed 7!!, aircraft ground speed of !!a/c gnd spd 3!! and #max indic normal accel# is maintained during the segment. The closest approach occurs before the !!a/c pitch!! is 90 degrees.

The position p1 is described by

(1) The distance along the a/c velocity vector from the a/c to a reference position p2 (on the ground) of !!displacement 1!!. The order of axes used in !!displacement 1!! is assumed to be distances positive in the East, North, and up directions.

(2) The !!a/c horiz disp!! from the p2 to p1 of !!displacement 2!! along the a/c ground track, and

(3) The !!p local vert!! displacement from p2 to p1 of !!displacement 3!! (positive if p2 is below p1).

!+a/c rate East+!
!+a/c rate North+!
!+a/c rate Up+! The aircraft rate, also known as the !!transport rate!! of the aircraft. The sources used in these calculations are as follows:

A/C Horizontal Velocity	!!a/c h vel 9!!
!+PM.hrc+! at A/C location	!!dist 1!!
A/C Height above Sea Level	!!a/c alt asl 5!!
Earth Ellipticity	!!ratio 1!!
Location of A/C	!!a/c locn 3!!

!+a/c speed indic accel+! The a/c speed after a flight segment beginning with the aircraft speed of !!a/c speed 5!! and with an !!a/c pitch!! of !!a/c pitch 11!!, maintaining #max indic normal accel# during the flight segment and ending when the !!a/c pitch!! equals !!a/c pitch 14!!.

!+a/c vert disp const vel+! The vertical displacement of the a/c during a flight segment in which a vertical speed of !!a/c v vel 7!! is maintained for !!a/c flt time 1!!.

!+a/c vert disp indic accel+! The vertical displacement of the a/c during a flight segment beginning with the a/c speed of !!a/c speed 2!! and with an !!a/c pitch!! of !!a/c pitch 13!! maintaining #max indic normal accel# during the flight segment and ending when the !!a/c pitch!! equals !!a/c pitch 9!! and the a/c speed is !!a/c speed 1!!. A displacement in the a/c !!p local vert!! (down) direction is negative.

!+drift angle+! The difference between the !!true heading!! of the Ya axis and the !!true heading!! of the a/c velocity vector.

Vector1	The projection of the Ya axis onto a/c !!p local horiz!!
Vector2	The projection of the a/c velocity vector A onto a/c !!p local horiz!!
Sign of angle	Positive clockwise (looking down) from Vector1 to Vector2

The a/c horizontal velocity is !!a/c h vel 4!!, and the true heading is !!a/c t hdg 4!!.

!+gnd trk angle+! The !!true heading!! of the aircraft velocity vector. The horizontal velocity of the a/c is !!a/c h vel 3!!.

!+hdg rate+! The rate at which !!a/c t hdg 2!! is changing. This is positive if !!a/c t hdg 2!! is increasing, negative if decreasing.

!+horiz veloc fm airspd+! The current velocity vector of the a/c projected into the a/c !!p local horiz!! plane. The present a/c state includes !!a/c t hdg 7!!, !!a/c roll 6!!, !!a/c pitch 15!!, !!a/c airspd 2!! and !!a/c aoa 1!!.

!+horiz veloc fm gndspd 1+! The current velocity vector of the a/c projected into the a/c !!p local horiz!! plane. The previous a/c horizontal velocity is !!a/c h vel 7!!, and the present a/c state includes !!a/c gnd spd 2!!.

!+horiz veloc fm gndspd 2+! The current velocity vector of the a/c projected into the a/c !!p local horiz!! plane. The present a/c state includes !!a/c t hdg 6!!, !!a/c gnd spd 4!! and !!a/c drift a 1!!.

!+horiz veloc fm lever+! The current velocity vector of the a/c projected into the a/c !!p local horiz!! plane. The previous a/c horizontal velocity is !!a/c h vel 2!!, and the present a/c state includes !!a/c v vel 3!!, !!moment arm 1!!, !!a/c t hdg 1!!, !!a/c roll 1!! and !!a/c pitch 1!!.

!+horiz veloc incr+!	The current velocity vector of the a/c projected into the a/c !!p local horiz!! plane. The !!a/c h vel 8!! is the source of the horizontal velocity used when reset, incremented or decremented by the value of !!a/c h vel incr 1!!.
!+normal accel+!	The acceleration of the a/c along the aircraft Za axis; positive in the Za positive direction. The aircraft state is described by !!a/c speed 3!!, !!a/c pitch 3!!, !!a/c roll 3!! and !!a/c t hdg 3!!.
!+pitch rate+!	The rate at which !!a/c pitch 2!! is changing. This is positive if !!a/c pitch 2!! is increasing, negative if decreasing.
!+roll rate+!	The rate at which !!a/c roll 2!! is changing. This is positive if !!a/c roll 2!! is increasing, negative if decreasing.
!+v spd fm airspd+!	The current speed of the a/c in the !!p local vert!! axis. The aircraft state for this model includes !!a/c airspd 1!! and !!a/c fpa 1!!.
!+v spd fm alt+!	The current speed of the a/c in the !!p local vert!! axis. The aircraft state for this model includes !!a/c v vel 2!! and !!a/c alt asl 2!!.
!+v spd fm gndspd+!	The current speed of the a/c in the !!p local vert!! axis. The aircraft state for this model includes !!a/c gnd spd 1!! !!a/c fpa 2!!.

4.2. Dictionary of !!term!!s

Some terms not defined here are defined in the PM.ECM. They include !!altitude asl!!, !!p local horiz!!, !!p local North!! and !!p local vert!!.

4.2.1. Dictionary of !!acm input!! !!term!!s

!!acm input!!	Input data to this module that may be used by one or more of its models. This may be any other !!term!! defined in this section.

The following series of terms are !!acm input!!s formed by replacement of a "#" by an integer. Each defined term with a "#" in the name forms a collection of terms called a term class. Individual terms are named by the replacement of the "#" in a term class by one of the integers listed with the term class definition.

Term class	Set of numbers that may textually replace "#" in legal !!term!!s	Definition
!!a/c airspd #!!	1..3	The speed of the a/c with respect to the local airmass (always non-negative)
!!a/c alt asl #!!	1..5	a/c !!altitude asl!!
!!a/c aoa #!!	1..3	!+DI.AOA+!
!!a/c drift a #!!	1	!+drift angle+!
!!a/c flt time #!!	1..2	Duration of an a/c flight segment
!!a/c fpa #!!	1..2	!+SS.flight path angle+!
!!a/c gnd spd #!!	1..5	The horizontal component of the a/c velocity (always nonnegative)
!!a/c h vel #!!	1..9	The current velocity vector of the a/c projected into the a/c !!p local horiz!! plane.
!!a/c h vel incr #!!	1	An incremental value of the a/c horizontal velocity vector.
!!a/c locn #!!	1..3	a/c location
!!a/c pitch #!!	1..15	!!a/c pitch!!
!!a/c roll #!!	1..6	!!a/c roll!!
!!a/c speed #!!	1..7	The speed of the a/c (always nonnegative)
!!a/c t hdg #!!	1..7	The !!true heading!! of the Ya axis
!!a/c v vel #!!	1..7	The current speed of the a/c in the !!p local vert!! axis.
!!body to platform #!!	1	The transformation from DI.Airframe system to DI.IMS system
!!displacement #!!	1..4	A =AT.displacement=
!!dist #!!	1	An =AT.distance=
!!moment arm #!!	1	The location of the a/c center of mass with respect to a device detecting a/c motion.
!!ratio #!!	1	A real

4.2.2. Dictionary of non-!!acm input!! !!term!!s

!!acm runtime change!! The source of this !!acm input!! may (will not) be changed during runtime if this state is true (false).

!!a/c horiz disp!! A displacement from one position to another in the direction of the projection of the a/c velocity vector on the a/c !!p local horiz!! plane. The displacement is positive from the first position in the direction of the projection of the velocity vector. The order of axes of the displacement are positive East, positive North, and positive Up directions.

!!a/c pitch!! The pitch of the aircraft.

Vector1	Ya axis
Vector2	The projection of the Ya axis onto a/c !!p local horiz!!
Sign of angle	positive when Ya is above the a/c !!p local horiz!!, otherwise negative.

!!a/c roll!! The aircraft roll.

Vector1	Xa axis
Vector2	The direction defined by the product of Ya cross !!p local vert!! (up).
Sign of angle	Positive when Xa is below a/c !!p local horiz!!, otherwise negative

!!src acm!! The variable, constant, literal, or =DB.dbitem= that is to be used by PM.ACM as the source for an !!acm input!!. When this module needs a value of the !!acm input!! it will use the value of this entity.

!!transport rate!! The velocity of an object over the surface of the earth, expressed as rotational changes in the East, North and Up axes, that would keep a coordinate frame aligned in the same orientation with respect to vertical and North. The rotation of the Earth is not a part of these terms.

!!true heading!! The angle formed by the projection of a vector (A) and North.

Vector1 !!p local North!!	
Vector2	The projection of a vector A onto a/c !!p local horiz!!
Sign of angle	Positive clockwise (looking down) from Vector1 to Vector2

5. Undesired event dictionary

5.1. Undesired event dictionary (system generation time)

%%acm already init%% An initial source and !!acm runtime change!! have already been specified for !!acm input!!.

%%acm chg method%% The !!acm runtime change!! for this input has been set false.

%%acm src not init%% At least one source of input for this access program has not been initialized.

%%acmslist not init%% ++SAME_...++ has not been invoked with =acmslist= p2.

%%err acmslist%% The members of the =acmslist= are !!acm input!! of a term class that is different from the term class listed in the definition of p1.

%%itm in acmslist%% The !!acm input!! has been declared, via ++SAME_...++ to be a member of a sourcelist of !!acm input!! that may not be reset separately.

5.2. Undesired event dictionary (runtime): None

6. System generation parameters

#max indic normal accel#	The maximum of indicated acceleration. Indicated acceleration is the sum of the Za component of !+gravity plus+! and !+normal accel+!.

CHAPTER 2

PM.ECM: EARTH CHARACTERISTICS MODULE

1. Introduction

The Earth Characteristics Module (ECM) hides models of physical earth characteristics (gravity, rotation, coriolis forces and magnetic variation), atmospheric characteristics (pressure and density) and earth geometry (latitude, longitude, distances and the direction of North, East and vertical for different positions). These programs provide information that is not measured directly by sensors on the aircraft.

Most physical characteristics of the earth and its atmosphere are position dependent. "Location" refers to a two-dimensional specification corresponding to =latitude= and =longitude=.

2. Interface overview

2.1. Model Value Access Program Table

2.1.1. Position specification

Program	*Parameters*	*Description*	*Undesired events*
The following programs have the same set of UE's,			
			%%src not init%%
+G_ALT_BARO+	p1: =AT.distance=; O	!+baro alt+!	
+G_ALT_SR+	p1: =AT.distance=; O	!+altitude sr+!	
+G_E_DIST+	p1: =AT.distance=; O	!+E dist+! from !!location 2!! to !!location 3!!	
+G_LOCN_BRG_GR+	p1: =earth_locn=; O	!+location brg+!	
+G_LOCN_DIST+	p1: =earth_locn=; O	!+location dist+!	
+G_LOCN_ELEV+	p1: =earth_locn=; O	!+location elev+!	
+G_LOCN_SR+	p1: =earth_locn=; O	!+location sr+!	
+G_LOCN_ORIENT+	p1: =earth_locn=; O	!+location or+!	
+G_LOCN_ANGRATE+	p1: =earth_locn=; O	!+location ang+!	
+G_N_DIST+	p1: =AT.distance=; O	!+N dist+! from !!location 13!! to !!location 14!!	
+G_ORIENT+	p1: =AT.orientation=; O	!+orient diff+! from !!location 15!! to !!location 16!!	
+G_SLT_RNG_L2+	p1: =AT.distance=; O	!+slt rng alt+!	
+G_ALT_DIFF+	p1: =AT.distance=; O	!+alt diff sr+!	

+G_GR_SR+	p1: =AT.distance=; O	!+ground range sr+!	
+G_SLT_RNG_GR+	p1: =AT.distance=; O	!+slt rng gr+!	

The following programs have the same set of UE's,

%dir undef%
%%src not init%%

+G_BRG_GR+	p1: =AT.angle=; O p2: =AT.distance=; O	!+t bearing+!, and !+ground range+! from !!location 4!! to !!location 5!!	
+G_ELEV_ANG_ALT+	p1: =AT.angle=; O	!+elev ang alt+!	

2.1.2. Earth Atmospheric Values

Program	*Parameters*	*Description*	*Undesired events*
The following programs have the same set of UE's,			%%src not init%%
+G_AIR_DENS+	p1: =AT.density=; O	!+air density+!	
+G_AIR_PRESS+	p1: =AT.pressure=; O	!+air pressure+!	
+G_DENS_RATIO+	p1: =real=; O	!+dens ratio+!	
+G_SOUND_SPD+	p1: =AT.speed=; O	!+sound speed+! at a surface whose !!altitude asl!! is equal to !!distance 20!!	
+G_WIND_VELOC+	p1: =AT.velocity=; O	!+wind velocity+!	

2.1.3. Other Earth Position Characteristics

Program	*Parameters*	*Description*	*Undesired events*
The following programs have the same set of UE's,			%%src not init%%
+G_CORIOL_E+	p1: =AT.accel=; O	!+coriolis E+!	
+G_CORIOL_N+	p1: =AT.accel=; O	!+coriolis N+!	
+G_CORIOL_V+	p1: =AT.accel=; O	!+coriolis V+!	
+G_GRAV+	p1: =AT.accel=; O	!+gravity plus+! at !!location 28!! and altitude !!distance 22!!	
+G_HRC+	p1: +AT.distance=; O	!+hrc+!	
+G_HRC_E+	p1: =AT.distance=; O	!+hrc East+!	
+G_HRC_N+	p1: =AT.distance=; O	!+hrc North+!	
+G_MAG_VAR+	p1: =AT.angle=; O	!+magnetic var+! at !!location 24!!	

+G_VERT_AX_D+	p1: =AT.distance=; O	!+vert ax dist+! at !!latitude 2!!

The following programs have the no UE's,

+G_ROT+	p1: =AT.angrate=; O	!+e rotation+!
+G_EQUAT_RAD+	p1: =AT.distance=; O	!+equator rad+!
+G_ELLIPTICITY+	p1: =real=; O	!+ellipticity+!

2.2. Input Programs

Each "#" of the input programs is replaced by an integer. Each +S_..._SRC+ program and +S_..._INIT+ program affect the "!!ecm input!!" with the same integer replacement: the set of legal "!!ecm input!!s defines the set of integers allowed in the program names. An "*" is used when referring to access name substrings of any of the following list: "ANGLE", "ANGRATE", "DIST", "LAT", "LOCN", "LONG", "ORIENT", "POSN", "PRESS", "TIME", or "VELOC".

Program	*Parameters*	*Description*	*Undesired events*
++S_*_#_INIT++	p1: =name=; I p2: =boolean=; I	!!src ecm!! !!runtime change!!	%%already init%%
+S_*_#_SRC+	p1: =name=; I	!!src ecm!!	%%src not init%% %%chg method%% %%itm in ecmlist%%
++SAME_*++	p1: =name=; I p2: =ecmlist=; I	!!src ecm!!	%%already init%% %%incons ecmlist%%
+S_*_SLIST_SRC+	p1: =name=; I p2: =ecmlist=; I	!!src ecm!!	%%incons ecmlist%% %%ecmlist not init%%

Parameters

The input parameters of type "name" may be the name of a literal, constant, variable, or DB.dbitem. The typeclass of the entity or DB.dbitem, and a more complete description of the parameter p1 are given by the following tables.

Programs	Typeclass of p1	P1 description
++S_*_#_INIT++,		
+S_*_#_SRC++,		
++SAME_*_++, and		
+S_*_SLIST_SRC+ where		
"*" is replace by -		
"ANGLE"	AT.angle	!!src ecm!! of !!angle #!!
"ANGRATE"	angrate	!!src ecm!! of !!angrate #!!
"DIST"	AT.distance	!!src ecm!! of !!distance #!!
"LAT"	latitude	!!src ecm!! of !!latitude #!!
"LOCN"	earth_locn	!!src ecm!! of !!location #!!
"LONG"	longitude	!!src ecm!! of !!longitude #!!
"ORIENT"	AT.orientation	!!src ecm!! of !!orientation #!!
"PRESS"	AT.pressure	!!src ecm!! of !!pressure #!!
"TIME"	timeint	!!src ecm!! of !!timeint #!!
"VELOC	AT.velocity	!!src ecm!! of !!velocity #!!

++S_*_#_INIT++
+S_*_#_SRC+

The set of legal replacements for "#" is the same as that set forth in the definition of terms as given in the table below:

Programs	Term with the same "#" subset
++S_*_#_INIT++ and	
+S_*_#_SRC++ where	
"*" is replace by -	
"ANGLE"	!!angle #!!
"ANGRATE"	!!angrate #!!
"DIST"	!!distance #!!
"LAT"	!!latitude #!!
"LOCN"	!!location #!!
"LONG"	!!longitude #!!
"ORIENT"	!!orientation #!!
"PRESS"	!!pressure #!!
"TIME"	!!timeint #!!
"VELOC	!!velocity #!!

Effects

++S_..._#_INIT++

++S_..._#_INIT++ disables %%src not init%% for the !!ecm input!! corresponding to the program name. (The name of the !!ecm input!! is obtained by substitution of the string "!!" for "++S_" and "_INIT++".) If the !!runtime change!! is false, %%chg method%% is enabled. Values for the !!ecm input!! are taken from p1 until an invocation of +S_..._#_SRC+ if !!runtime change!! is true, otherwise throughout the life of the program. %%already init%% is enabled for the !!ecm input!!.

+S_..._#_SRC+

Until another invocation of this program the module will use p1 to obtain an updated value for the !!ecm input!!.

++SAME_...++

This program declares that the sources of information for the !!ecm input!! members of the =ecmlist= are always the same entity. Until +S_..._SLIST_SRC+ is invoked with p2, the source of information is p1. The !!runtime change!! state of each member of the =ecmlist= is true. %%itm in ecmlist%% is enabled for each !!ecm input!! in p2. %%already init%% is enabled for all members of p2. %%ecmlist not init%% is

disabled for p2.

+S_..._SLIST_SRC+ Until another invocation of this program the module will use p1 to obtain an updated value for each !!ecm input!! of p2.

3. Local type definitions

=earth_locn= A two dimensional array of AT.angles. The first dimension corresponds to SS.latitude, the second to SS.longitude.

=ecmlist= Simple enumerated: a list (enclosed in parentheses, whitespace separator) of a set of !!ecm input!! names of the same type.

4. Dictionary

Angles are defined by use of a template listing the directions of two vectors and the sense of the angle measurement.

4.1. !!ecm input!!

!!angle #!! Each term formed by replacement of "#" by an integer names an !!ecm input!! of type =AT.angle=. Valid replacements for "#" are listed in the table below, along with the value programs that may use the !!ecm input!!.

!!ecm input!!	Value program
!!angle 1!!	+G_LOCN_BRG_GR+
!!angle 2!!	+G_LOCN_ELEV_SRC+
!!angle 3!!	+G_LOCN_ELEV_SRC+
!!angle 4!!	+G_LOCN_SR+
!!angle 5!!	+G_ALT_DIFF+

!!angrate #!! Each term formed by replacement of "#" by an integer names an !!ecm input!! of type =AT.angrate=. Valid replacements for "#" are listed in the table below, along with the value programs that may use the !!ecm input!!.

!!ecm input!!	Value program
!!angrate 1!!	+G_LOCN_ANGRATE+
!!angrate 2!!	+G_LOCN_ANGRATE+
!!angrate 3!!	+G_LOCN_ANGRATE+
!!angrate 4!!	+G_CORIOL_E/N/V+
!!angrate 5!!	+G_CORIOL_E/N/V+
!!angrate 6!!	+G_CORIOL_E/N/V+

!!distance #!! Each term formed by replacement of "#" by an integer names an !!ecm input!! of type =AT.distance=. Valid replacements for "#" are listed in the table below, along with the value programs that may use the !!ecm input!!.

!!ecm input!!	Value program
!!distance 1!!	+G_ALT_BARO+
!!distance 2!!	+G_ALT_SR+
!!distance 3!!	+G_LOCN_BRG_GR+
!!distance 4!!	+G_LOCN_DIST+
!!distance 5!!	+G_LOCN_DIST+
!!distance 6!!	+G_LOCN_ELEV+
!!distance 7!!	+G_LOCN_SR+
!!distance 8!!	+G_LOCN_SR+
!!distance 10!!	+G_SLT_RNG_L2+
!!distance 11!!	+G_ALT_DIFF+
!!distance 12!!	+G_GR_SR+
!!distance 13!!	+G_GR_SR+
!!distance 14!!	+G_SLT_RNG_GR+
!!distance 15!!	+G_SLT_RNG_GR+
!!distance 16!!	+G_ELEV_ANG_ALT+
!!distance 17!!	+G_AIR_DENS+
!!distance 18!!	+G_AIR_PRESS+
!!distance 19!!	+G_DENS_RATIO+
!!distance 20!!	+G_SOUND_SPD+
!!distance 21!!	+G_ALT_SR+
!!distance 22!!	+G_GRAV+
!!distance 23!!	+G_HRC_E+
!!distance 24!!	+G_HRC_N+

!!ecm input!! Input data to this module that may be used by one of more of its models. This may be an !!angle #!!, !!angrate #!!, !!distance #!!, !!latitude #!!, !!location #!!, !!longitude #!!, !!orientation #!!, !!pressure #!!, !!timeint #!!, or !!velocity #!!.

!!latitude #!! Each term formed by replacement of "#" by an integer names an !!ecm input!! of type =latitude=. Valid replacements for "#" are listed in the table below, along with the value programs that may use the !!ecm input!!.

!!ecm input!!	Value program
!!latitude 2!!	+G_VERT_AX_D+

!!location #!! Each term formed by replacement of "#" by an integer names an !!ecm input!! of type =earth_locn=. Valid replacements for "#" are listed in the table below, along with the value programs that may use the !!ecm input!!.

!!ecm input!!	Value program
!!location 1!!	+G_ALT_SR+
!!location 2!!	+G_E_DIST+
!!location 3!!	+G_E_DIST+
!!location 4!!	+G_BRG_GR+
!!location 5!!	+G_BRG_GR+
!!location 7!!	+G_LOCN_BRG_GR+
!!location 8!!	+G_LOCN_DIST_+
!!location 9!!	+G_LOCN_ELEV+
!!location 10!!	+G_LOCN_SR+
!!location 11!!	+G_LOCN_ORIENT+
!!location 13!!	+G_N_DIST+
!!location 14!!	+G_N_DIST+
!!location 15!!	+G_ORIENT+
!!location 16!!	+G_ORIENT+
!!location 18!!	+G_SLT_RNG_L2+
!!location 19!!	+G_SLT_RNG_L2+
!!location 21!!	+G_HRC+
!!location 22!!	+G_ELEV_ANG_ALT+
!!location 23!!	+G_ELEV_ANG_ALT+
!!location 24!!	+G_MAG_VAR+
!!location 25!!	+G_LOCN_ANGRATE+
!!location 26!!	+G_CORIOL_E/N/V+
!!location 27!!	+G_ALT_SR+
!!location 28!!	+G_GRAV+
!!location 29!!	+G_HRC_E+
!!location 30!!	+G_HRC_N+

!!orientation #!!

Each term formed by replacement of "#" by an integer names an !!ecm input!! of type =AT.orientation=. Valid replacements for "#" are listed in the table below, along with the value programs that may use the !!ecm input!!.

!!ecm input!!	Value program
!!orientation 1!!	+G_LOCN_ORIENT+

!!pressure #!!

Each term formed by replacement of "#" by an integer names an !!ecm input!! of type =AT.pressure=. Valid replacements for "#" are listed in the table below, along with the value programs that may use the !!ecm input!!.

!!ecm input!!	Value program
!!pressure 1!!	+G_ALT_BARO+

!!timeint #!!

Each term formed by replacement of "#" by an integer names an !!ecm input!! of type =timeint=. Valid replacements for "#" are listed in the table below, along with the value programs that may use the !!ecm input!!.

!!ecm input!!	Value program
!!timeint 1!!	+G_LOCN_ANGRATE+

!!velocity #!!

Each term formed by replacement of "#" by an integer names an !!ecm input!! of type =AT.velocity=. Valid replacements for "#" are listed in the table below, along with the value programs that may use the !!ecm input!!.

!!ecm input!!	Value program
!!velocity 1!!	+G_CORIOL_E/N/V+
!!velocity 2!!	+G_WIND_VELOC+
!!velocity 3!!	+G_WIND_VELOC+

4.2. !!term!!s other than !!ecm input!!

!!alt diff!! — The difference in altitude from one !!position!! (A) to another (B). This may be either positive, zero or negative (negative when (A) is higher than (B)).

!!altitude asl!! — Altitude above sea level.

!!elev ang!! — The angle of elevation from one !!position!! (A) to another (B).

Vector1	The vector from (A) to (B)
Vector2	The projection of Vector1 onto the !!p local horiz!! of (A)
Sign of angle	Positive if the !!alt diff!! of the respective !!position!! is positive

!!p local East!!, !!p local horiz!!, !!p local m North!!, !!p local North!!, !!p local vert!!
North, East, Vertical, and Magnetic North vectors and the horizontal plane are defined for !!position!! static with respect to the earth. North, East and Vertical form an orthogonal set of axes; the specific direction is defined with reference to a model of the shape of the earth (reference 1). The particular model used is a secret of this module. !!p local vert!! may be approximated by the direction of the gravity vector. The sense of the vector results in down as positive. !!p local East!! is perpendicular to the axis of rotation, positive in the direction of rotation. North, East and Magnetic North are in the horizontal plane. Because of the curvature of the surface of the earth there is a horizontal surface defined as the locus of !!position!!s with the same !!altitude asl!! for any !!position!!. These may be locally approximated by the horizontal plane. These !!p local ...!! terms are the direction of vectors (with the exception of !!p local horiz!!, which is a plane) unique for all points.

!!position!! — A reference to a point in three-dimensional space characterized by an =earth_locn= and an altitude.

!!runtime change!! — The source of this !!ecm input!! may (will not) be changed during runtime if this state is true (false).

!!slt rng!! — The nonnegative straight-line =AT.distance= from one !!position!! (A) to another (B). The sign of the value is the same as the !!alt diff!! of the two !!position!!s, respectively.

!!src ecm!! — The variable, constant, literal, or =DB.dbitem= that is to be used by this module as the source for a !!ecm input!!. When this module needs a value of the !!ecm input!! it will use the value of this entity.

4.3. !+term+!s

!+air density+! — The air density at a !!position!! with !!altitude asl!! equal to !!distance 17!!.

!+air pressure+! The air pressure at a !!position!! with !!altitude asl!! equal to !!distance 18!!.

!+alt diff sr+! The !!alt diff!! from two !!position!!s (A) and (B) defined such that !!distance 11!! is the !!slt rng!! from (A) to (B) and !!angle 5!! is the !!elev ang!! from (A) to (B).

!+altitude sr+! !!altitude asl!! of a point (A) such that !!location 1!! is the location of (A); and !!distance 2!! is the !!slt rng!! from (A) to a point with location !!location 27!! and altitude !!distance 21!!.

!+baro alt+! Barometric !!altitude asl!! of a surface such that !!pressure 1!! is the sea level atmospheric pressure !!distance 1!! is barometric altitude computed with the assumption that the sea level pressure is #std slp#.

!+coriolis E+!
!+coriolis N+!
!+coriolis V+! The acceleration of an object at !!location 26!! with !!velocity 1!! velocity with respect to the earth due to the coriolis force of the earth reference frame. !!angrate 4!!, !!angrate 5!! and !!angrate 6!! are assumed to be the !!transport rate!! of the object in the East, North, and vertical axes. The !+coriolis *+! terms are accelerations with respect to the earth. Accelerations and velocities are expressed in the !!p local East!!, !!p local North!!, !!p local vert!! reference frame.

!+dens ratio+! The ratio of the air density at an !!altitude asl!! of !!distance 19!! to #std sl dens#.

!+E dist+! The length of the projection of the vector from one !!position!! (A) to another (B) onto the !!p local East!! vector of (A). !+E dist+! is negative if (B) is West of (A).

!+e rotation+! The inertial angular rate of the earth's rotation around its axis.

!+elev ang alt+! The !!elev ang!! from one !!position!! (A) at !!location 22!! to another (B) at !!location 23!! such that !!distance 16!! is the !!alt diff!! from (A) to (B).

!+ellipticity+! The ellipticity of an ellipsoid of revolution that approximates the shape of the earth.

!+equator rad+! The radius of the earth at the equator.

!+gravity plus+! The acceleration due to gravity and rotation of the earth of an object. This is negative in the !!p local vert!! direction and is acceleration with respect to the earth.

!+ground range+! The nonnegative =AT.distance= at sea level from one !!position!! to another. If this is an arc length of more than #2D range limit# the arc length between the !!position!! is used. For a =AT.distance= less than #2D range limit# the lengths defined by arc lengths or straight line =AT.distance= are nearly similar and it is a secret of this module which model is used. If the ground range is greater than #gr limit# then #gr limit# will be returned.

!+ground range sr+! The !+ground range+! from two !!position!!s (A) and (B) such that !!distance 12!! is the !!slt rng!! from (A) to (B) and !!distance 13!! is the !!alt diff!! from (A) to (B).

!+hrc+! The radius of curvature of the horizontal surface of the earth at sea level.

!+hrc East+! The radius of curvature of the horizontal surface through !!location 29!! and altitude !!distance 23!! in the plane formed by !!p local East!! and !!p local vert!!.

!+hrc North+! The radius of curvature of the horizontal surface through !!location 30!! and altitude !!distance 24!! in the plane formed by !!p local North!! and !!p local vert!!.

!+location ang+! The =earth_locn= (A) such that !!angrate 1!!, !!angrate 2!!, and !!angrate 3!! applied over !!timeint 1!! are the !+orient diff+! from !!location 25!! to (A). !!angrate 1!! is the rotation applied to the first axis, !!angrate 2!! to the second, and !!angrate 3!! to the third. If +G_LOCN_ANGRATE+ is not invoked with period !!timeint 1!! the value of !+location ang+! may not be as defined.

!+location brg+! The =earth_locn= (A) such that !!angle 1!! is the !+t bearing+! from (A) to !!location 7!! and !!distance 3!! is the !+ground range+! from (A) to !!location 7!!.

!+location dist+! The =earth_locn= (A) such that !!distance 4!! is the !+N dist+! from !!location 8!! to (A) and !!distance 5!! is the !+E dist+! from !!location 8!! to (A).

!+location elev+! The =earth_locn= of a point (A) such that Another point (B) is defined at !!location 9!!; !!distance 6!! is the !!alt diff!! from (A) to (B); !!angle 2!! is the !+t bearing+! from (A) to !!location 9!!; !!angle 3!! is the !!elev ang!! from (A) to (B).

!+location or+! The =earth_locn= (A) such that !!orientation 1!! is the !+orient diff+! from !!location 11!! to (A).

!+location sr+! The =earth_locn= of a point (A) such that Another point (B) is defined at !!location 10!!; !!distance 7!! is the !!alt diff!! from (A) to (B); !!angle 4!! is the !+t bearing+! from (A) to !!location 10!!; !!distance 8!! is the !!slt rng!! from (A) to (B).

!+magnetic var+! The magnetic variation of a location (A):

Vector1	!!p local North!! of (A)
Vector2	!!p local m North!! of (A)
Sign of angle	Positive clockwise (looking down) from Vector1 to Vector2

!+N dist+! The length of the projection of the vector from one !!position!! (A) to another (B) onto the !!p local North!! vector of (A). !+N dist+! is negative if (B) is South of (A).

!+orient diff+! The rotational difference of the coordinate systems associated with the second location (!!p local East!!, !!p local North!! and !!p local vert!! are the coordinate axes) in terms of the coordinate system of the first.

!+slt rng alt+! The !!slt rng!! from a point (A) at !!location 18!! to a point (B) at !!location 19!! such that !!distance 10!! is the !!alt diff!! from (A) to (B).

!+slt rng gr+! The !!slt rng!! from two !!position!!s (A) and (B) defined such that !!distance 14!! is the !+ground range+! from (A) to (B) and !!distance 15!! is the !!alt diff!! from (A) to (B).

!+sound speed+! The speed of sound at an !!altitude asl!! of !!distance 20!!

!+t bearing+! The true bearing from one location (A) to another (B).

Vector1	!!p local North!! at (A)
Vector2	The projection of the vector from (A) to (B) onto the !!p local horiz!! plane of (A).
Sign of angle	Positive clockwise (looking down) from Vector1 to Vector2

$0° \leq$!+t bearing+! $< 360°$.

!+vert ax dist+!	Distance from a !!position!! (A) to the intersection of the plane perpendicular to the !!p local North!! of (A) containing (A) with the earth axis of rotation.
!+wind velocity+!	The velocity of the airmass with respect to the earth of a sensor that has !!velocity 2!! with respect to the earth and !!velocity 3!! with respect to the airmass. !+wind velocity+!, !!velocity 2!! and !!velocity 3!! are in the !!p local East!!, !!p local North!!, !!p local vert!! reference frame.

5. Undesired event dictionary

%%already init%%	An initial source and !!runtime change!! have already been specified for a !!ecm input!!. This !!ecm input!! may be a member of a =ecmlist=.
%%chg method%%	The !!runtime change!! for this input has been set false.
%dir undef%	The user has referenced a direction from one =earth_locn= to itself.
%%incons ecmlist%%	The members of the =ecmlist= are !!ecm input!! of a type that is different from p1.
%%itm in ecmlist%%	The !!ecm input!! has been declared, via ++SAME_...++ to be a member of a =ecmlist= of !!ecm input!! that may not be reset separately.
%%ecmlist not init%%	++SAME_...++ has not been invoked with =ecmlist= p2.
%%src not init%%	At least one source of input for this access program has not been initialized.

6. System generation parameters

#2D range limit#	Type: distance. Min-Max: 22.7 nmi - 36 nmi. The maximum !+ground range+! that should be derived using a two dimensional model of the earth.
#gr limit#	Type: distance. Min-Max: 3500 nmi - 3800 nmi.
#max wind magnitude#	Type: speed. Min-Max: 128 fps - 512 fps. The maximum magnitude of the wind.
#std sl dens#	Air density at sea level in a standard atmosphere.
#std slp#	Air pressure at mean day sea level in a standard atmosphere.

CHAPTER 3

PM.HFM: HUMAN FACTORS MODULE

1. Introduction

This submodule contains models of those human (pilot) factors that must be considered when determining the behavior of the software. It includes models of how a human perceives continuous motion from discrete steps, and how long a human takes to react to a stimulus.

2. Interface overview

2.1. ACCESS PROGRAM TABLE

Program	*Parameters*	*Description*	*Undesired events*
+G_PILOT_REACT_TIME+			
	p1: =timeint=; O	!+pilot react time+!	
			None
+G_UPDATE_FREQ+			
	p1: =timeint=; O	!+symbol update period+!	
			None

3. Local type definitions None

4. Dictionary

!+pilot react time+!
: The minimum amount of time required for a pilot to react to a stimulus, plus a safety margin, assuming that the pilot notices the stimulus immediately.

!+symbol update period+!
: The maximum time interval between updates on a display in order to achieve the illusion of continuous motion.

5. Undesired event dictionary None

6. System generation parameters

#symbol update period#
: System generation time value of !+symbol update period+!.

CHAPTER 4

PM.TBM: TARGET BEHAVIOR MODULE

1. Introduction

This submodule contains models for tracking of target movement relative to the aircraft. Future target positions are extrapolated from previous reports of target position and any reports of target heading and speed. Aircraft position, attitude, and velocity are assumed known to this module. This module may be used to keep track of more than one target. Integers chosen by this module are used to refer to individual targets. Input parameters to this module may be given as variables, constants, literals, or =DB.dbitem=s.

2. Interface overview

2.1. Access Program Table

Program	*Parameters*	*Description*	*Undesired events*
+ASGN_TGT_IDENT+			
	p1: =integer=; O	!!tgt ident!!	
			None
The following programs have the same UE,			
			%unknown target%
+FREE_TGT+	p1: =integer=; I	!!tgt ident!!	
+S_TGT_POSN+			
	p1: =integer=; I	!!tgt ident!!	
	p2: =timeint=; I	!!transit time!!	
	p3: =earth_locn=; I	target location	
	p4: =AT.distance=; I	target !!ECM.altitude asl!!	
+S_TGT_POSN_REL+			
	p1: =integer=; I	!!tgt ident!!	
	p2: =timeint=; I	!!transit time!!	
	p3: =AT.angle=; I	!!tgt azimuth!!	
	p4: =AT.angle=; I	!!tgt elevation!!	
	p5: =AT.distance=; I	!!tgt slt rng!!	
+S_TGT_SPD+			
	p1: =integer=; I	!!tgt ident!!	
	p2: =timeint=; I	!!transit time!!	
	p3: =AT.speed=; I	target speed	
+S_TGT_HDG+			
	p1: =integer=; I	!!tgt ident!!	
	p2: =timeint=; I	!!transit time!!	
	p3: =AT.angle=; I	target !!true heading!!	

+S_TGT_V_SPD+

	p1: =integer=; I	!!tgt ident!!
	p2: =timeint=; I	!!transit time!!
	p3: =AT.speed=; I	!!tgt v spd!!

%unknown target%
%posn unknown%
%incomp motion%

+G_TGT_POSN_REL+

	p1: =integer=; I	!!tgt ident!!
	p2: =timeint=; I	!!transit time!!
	p3: =AT.angle=; O	!!tgt azimuth!!
	p4: =AT.angle=; O	!!tgt elevation!!

%unknown target%
%posn unknown%
%incomp motion%

+G_TGT_SR+	p1: =integer=; I	!!tgt ident!!
	p2: =timeint=; I	!!transit time!!
	p3: =AT.distance=; O	!!tgt slt rng!!

Effects

+ASGN_TGT_IDENT+
The programs +S_TGT_POSN+, +S_TGT_POSN_REL+, +S_TGT_SPD+, +S_TGT_HDG+, +S_TGT_V_SPD+, +G_TGT_POSN_REL+, +G_TGT_SR+ and +FREE_TGT+ may be invoked with !!tgt ident!! p1 until after the next invocation of +FREE_TGT+ with !!tgt ident!! p1.

+FREE_TGT+ The only interface program of this module that may be subsequently invoked with !!tgt ident!! p1 is +ASGN_TGT_IDENT+. This restriction only applies until that invocation.

+S_TGT_POSN+
The location of p1 at time p2 is assumed by this module to be the earth location p3, altitude p4.

%posn unknown% is disabled for p1.

+S_TGT_POSN_REL+
%posn unknown% is disabled for p1.

+S_TGT_HDG+
The !!true heading!! of the target p1's direction of motion at time p2 is assumed by this module to be p3.

+S_TGT_V_SPD+
The speed in the aircraft !!p local vert!! of target p1 relative to the earth at time p2 is assumed by this module to be p3.

+S_TGT_SPD+
The speed of target p1 relative to the earth at time p2 is assumed by this module to be p3.

3. Local type definitions None

4. Dictionary

!!tgt azimuth!! The !!azimuth!! from the aircraft to target p1 at !!transit time!! p2.

!!tgt elevation!!
The !!elevation!! from the aircraft to target p1 at !!transit time!! p2.

!!tgt ident!! An =integer= index in the range 1..#num targets#.

!!tgt slt rng!! The slant range from the aircraft to target p1 at !!transit time!! p2.

!!tgt v spd!! Vertical speed of target relative to earth (down is negative).

!!transit time!! The elapsed time since +S_TGT_POSN_REL+ or +S_TGT_POSN+ was first called with this p1.

5. Undesired event dictionary

%incomp motion%
The target motion is incompletely specified since %unknown target% was last disabled. The target motion is completely specified by either +S_TGT_SPD+ with zero p3, or the combination of +S_TGT_SPD+, +S_TGT_HDG+, and +S_TGT_V_SPD+.

%posn unknown%
Neither +S_TGT_POSN_REL+ nor +S_TGT_POSN+ have been invoked since the last invocation of +ASGN_TGT_IDENT+.

%too many targets%
The number of invocations of +ASGN_TGT_IDENT+ minus the number of invocations of +FREE_TGT+ is greater than #num targets#.

%unknown target%
+ASGN_TGT_IDENT+ has not been invoked with p1 since the last inovcation of +FREE_TGT_IDENT+ for this !!tgt ident!! or since sysgen time if +FREE_TGT_IDENT+ has never been invoked for this !!tgt ident!!.

6. System generation parameters

#num targets# The maximum number of targets that this module will have to track simultaneously.

CHAPTER 5

PM.WBM: WEAPON BEHAVIOR MODULE

1. Introduction

This submodule contains models of weapon behavior and programs for all weapons. These models predict weapon trajectories of the !!current wpn!!.

2. Interface overview

2.1. Access program tables

2.1.1. Release Weapon Trajectories

Program	*Parameters*	*Description*	*Undesired events*
+G_RLS_T_TRAV+	p1: =timeint=; O	!+rls t travel+!	%not rls weap% %%wbm src not init%%
+G_RLS_B_BEARING+	p1: =AT.angle=; O	!+burst bearing+!	
+G_RLS_WITHIN_RANGE+	p1: =boolean=; O	!+rls within rng+!	%not rls or sk% %%wbm src not init%%
+G_RLS_RANGE+	p1: =AT.distance=; O	!+rls range+!	%not rls or wall% %%wbm src not init%%
+G_MAX_RANGE_ANGLE+	p1: =AT.angle=; O	!+max range angle+!	%not rls weap%
			%wpn out of rng% %not rls weap% %%wbm src not init%%

+G_RLS_PITCH_UP+			
	p1: =AT.angle=; O	!+upper rls pitch+!	
+G_RLS_PITCH_LO+			
	p1: =AT.angle=; O	!+lower rls pitch+!	
+G_RLS_PITCH_SHR+			%wpn out of rng% %not shrike% %%wbm src not init%%
	p1: =AT.angle=; O	!+sk rls pitch+!	

2.1.2. Launch Weapon Trajectories

Program	*Parameters*	*Description*	*Undesired events*
+G_SR_A/G_I_P+			%not lnch weap% %%wbm src not init%%
	p1: =AT.distance=; O	!+slt rng imp pt+!	
+G_A/G_AIM_PT+			
	p1: =AT.angle=; O	!+a/g imp pt az+!	
	p2: =AT.angle=; O	!+a/g imp pt el+!	
+G_A/A_IMP_AZ+			
	p1: =AT.angle=; O	!+a/a imp pt az+!	
+G_A/A_IMP_EL+			
	p1: =AT.angle=; O	!+a/a imp pt el+!	

2.1.3. Sources of Module Input

Each "#" of the input programs is replaced by an integer. Each +S_..._#_SRC+ program and ++S_..._#_INIT++ program affect the !!wbm input!! with the same integer replacement: the set of legal !!wbm input!!s defines the set of integers allowed in each program name. An "*" is used when referring to access name substrings of any of the following list: "TRAJ_ALT", "TRAJ_A_DIFF", "WPN_AIRSPD", "WPN_AOA", "WPN_D_A", "WPN_PTCH", "WPN_ROLL", "WPN_A_VEL", "WPN_SR_IP", or "WPN_T_T".

Program	*Parameters*	*Description*	*Undesired events*
++S_*_#_INIT++			%%wbm already init%% %%itm in wlist%%
	p1: =name=; I	!!src wbm!!	
	p2: =boolean=; I	!!wbm runtime change!!	
++S_TRAJ_RNG_#_INIT++			%%wbm already init%%
	p1: =name=; I	!!src wbm!! of !!traj range #!!	

```
                p2: =boolean=; I        !!wbm runtime change!!

++S_WPN_HDG_#_INIT++
                p1: =name=; I           !!src wbm!! of !!w a/c hdg #!!
                p2: =boolean=; I        !!wbm runtime change!!
```

```
                                                                %%wbm chg
                                                                   method%%
                                                                %%itm not init%%
                                                                %%itm in wlist%%
                                                                %%wlist member
                                                                   init%%
+S_*_#_SRC+     p1: =name=; I           !!src wbm!!
```

```
                                                                %%itm not init%%
                                                                %%wbm chg
                                                                   method%%
+S_TRAJ_RNG_#_SRC+
                p1: =name=; I           !!src wbm!! of !!traj range #!!

+S_WPN_HDG_#_SRC+
                p1: =name=; I           !!src wbm!! of !!w a/c hdg #!!
```

```
                                                                %%wlist in
                                                                   wlist%%
                                                                %%err wlist%%
++SAME_*++      p1: =name=; I           !!src wbm!!
                p2: =wbmslist=; I
```

```
                                                                %%err wlist%%
                                                                %%wlist not
                                                                   init%%
+S_*_WBM_SL_SRC+
                p1: =name=; I           !!src wbm!!
                p2: =wbmslist=; I
```

Parameters

The input parameters of type "name" may be the name of a literal, constant, variable, or DB.dbitem. The typeclass of the entity or DB.dbitem, and a more complete description of the parameter p1 are given by the following tables.

Programs	Typeclass of p1	P1 description
++S_TRAJ_RNG_#_INIT++	distance	!!src wbm!! of !!traj range #!!
++S_WPN_HDG_#_INIT++	angle	!!src wbm!! of !!w a/c hdg #!!
+S_TRAJ_RNG_#_SRC+	distance	!!src wbm!! of !!traj range #!!
+S_WPN_HDG_#_SRC+	angle	!!src wbm!! of !!w a/c hdg #!!
++S_*_#_INIT++,		
+S_*_#_SRC+,		
++SAME_*++ and		
+S_*_WBM_SL_SRC+ where		
"*" is replaced by:		
"TRAJ_ALT"	distance	!!src wbm!! of !!begin traj alt #!!
"TRAJ_A_DIFF"	distance	!!src wbm!! of !!traj alt diff #!!
"WPN_AIRSPD"	speed	!!src wbm!! of !!w a/c airspd #!!
"WPN_AOA"	angle	!!src wbm!! of !!w a/c aoa #!!
"WPN_D_A"	angle	!!src wbm!! of !!w a/c drft a #!!
"WPN_PTCH"	angle	!!src wbm!! of !!w a/c pitch #!!
"WPN_ROLL"	angle	!!src wbm!! of !!w a/c roll #!!
"WPN_A_VEL"	velocity	!!src wbm!! of !!w a/c vrt air #!!
"WPN_SR_IP"	distance	!!src wbm!! of !!w sr ip #!!
"WPN_T_T"	timeint	!!src wbm!! of !!weap tt #!!

++S_..._#_INIT++
+S_..._#_SRC+

The set of legal replacements for "#" is the same as that set forth in the definition of terms as given in the table below:

Programs	Term with the same "#" subset
++S_TRAJ_RNG_#_INIT++	!!traj range #!!
++S_WPN_HDG_#_INIT++	!!w a/c hdg #!!
+S_TRAJ_RNG_#_+	!!traj range #!!
+S_WPN_HDG_#_+	!!w a/c hdg #!!
++S_*_#_INIT++ and	
+S_*_#_SRC+, where	
"*" is replaced by:	
"TRAJ_ALT"	!!begin traj alt #!!
"TRAJ_A_DIFF"	!!traj alt diff #!!
"WPN_AIRSPD"	!!w a/c airspd #!!
"WPN_AOA"	!!w a/c aoa #!!
"WPN_D_A"	!!w a/c drft a #!!
"WPN_PTCH"	!!w a/c pitch #!!
"WPN_ROLL"	!!w a/c roll #!!
"WPN_A_VEL"	!!w a/c vrt air #!!
"WPN_SR_IP"	!!w sr ip #!!
"WPN_T_T"	!!weap tt #!!

++S_..._#_INIT++

++S_..._#_INIT++ disables %%wbm src not init%% for the !!wbm input!! corresponding to the program name.

If p2 is false, %%wbm chg method%% is enabled, and values for the !!wbm input!! are taken from the present value of p1. If p2 is true, values for the !!wbm input!! are taken from the present value of p1 only until an invocation of +S_..._#_SRC+ for the same input.

%%wbm already init%% is enabled for the !!wbm input!!.

%%wlist member init%% is enabled for the !!wbm input!!.

%%itm not init%% is disabled for the !!wbm input!!.

+S_..._#_SRC+

Until another invocation of this program the module will use p1 as the !!src wbm!! for the !!wbm input!!.

++SAME_*++ This program declares that the sources of information for the !!wbm input!! members of the wbmslist are always the same entity. Until +S_..._WBM_SL_SRC+ is invoked with p2, the source of information for all the members of p2 is p1. The !!wbm runtime change!! state of each member of the wbmslist is true. %%itm in wlist%% is enabled for each !!wbm input!! in p2. %%wlist not init%% is disabled for p2.

%%itm in wlist%% is enabled for all members of wbmslist p2.

%%wlist in wlist%% is enabled for wbmslist p2.

+S_*_WBM_SL_SRC+

Until another invocation of this program with the same p2 the module will use p1 to obtain an updated value for each !!wbm input!! of p2.

3. Local type definitions

=wbmslist= Simple enumerated: a list (enclosed in parentheses, whitespace separator) of a set of !!wbm input!! names that are members of the same term class in PM.WBM.4.2.

4. Dictionary

4.1. !+term+!s

All !+term+!s refer to the trajectory behavior of the !!current wpn!!.

!+a/a imp pt az+!
!+a/a imp pt el+!

The !!azimuth!! and !!elevation!! from the a/c at !!wpn sep!! to the !!imp posn!! of an !!a/a launch wpn!!. The following terms are assumed to describe the a/c state at !!wpn sep!!: !!w a/c aoa 4!!, !!w a/c airspd 4!!, !!w a/c pitch 4!! and !!w a/c roll 5!!. The !!imp posn!! is assumed to be !!w sr ip 2!! from the !!wpn sep posn!! , and the time interval of weapon travel is assumed !!weap tt 2!!.

!+a/g imp pt az+!
!+a/g imp pt el+!

The !!azimuth!! and !!elevation!! from the a/c at !!wpn sep!! to the !!imp posn!! of an !!a/g launch wpn!!. The following terms are assumed to describe the a/c state at !!wpn sep!!: !!w a/c aoa 5!!, !!w a/c airspd 3!!, !!w a/c pitch 3!! and !!w a/c roll 6!!. The !!imp posn!! is assumed !!w sr ip 1!! from the !!wpn sep posn!!.

!+burst bearing+!

!+t bearing+! from !!wpn sep posn!! to the !!traj end posn!! for a !!release wpn!!. The time interval of weapon travel is assumed to be !!weap tt 1!!. At !!wpn sep!! the a/c state used by the model is !!w a/c hdg 1!!, !!w a/c roll 3!! and !!w a/c vrt air 1!!.

!+lower rls pitch+!

!+upper rls pitch+!
The lower and upper aircraft pitches such that an aircraft at !!wpn sep!! with either of these pitches and other a/c state descriptions would cause the !!release wpn!! to follow a trajectory through the !!traj end posn!!. The other a/c state descriptions assumed by the model are !!w a/c vrt air 2!!, !!w a/c roll 4!!, !!w a/c aoa 3!! and !!begin traj alt 3!!. The difference between the !!wpn sep posn!! and the !!traj end posn!! is also assumed to be !!traj alt diff 3!! and !!traj range 1!!.

!+max range angle+!
The a/c pitch at the time of !!wpn sep!! that would result in maximum horizontal travel of a !!release wpn!!.

!+rls t travel+! Difference between the time of !!wpn sep!! and when the !!release wpn!! travels through !!traj end posn!!, with the following assumptions. At !!wpn sep!! the a/c state is !!begin traj alt 2!!, !!w a/c pitch 2!!, !!w a/c roll 2!!, !!w a/c aoa 2!!, !!w a/c airspd 2!!, and !!w a/c drft a 2!!. The !!traj end posn!! is also assumed to be !!traj alt diff 2!! from the !!wpn sep posn!!.

!+rls range+! Horizontal distance from the !!wpn sep posn!! to the !!traj end posn!! for a !!release wpn!! or !!walleye!!. At !!wpn sep!! the model assumes that the a/c state is !!begin traj alt 1!!, !!w a/c pitch 1!!, !!w a/c roll 1!!, !!w a/c aoa 1!!, !!w a/c airspd 1!!, and !!w a/c drft a 1!!. The !!traj end posn!! is also assumed to be !!traj alt diff 1!! from the !!wpn sep posn!!.

!+rls within rng+!
True if the the horizontal distance from the !!wpn sep posn!! to the !!traj end posn!! would be greater than or equal to !!traj range 1!! for a !!release wpn!! or !!shrike!!. This model assumes the a/c state at !!wpn sep!! to be !!w a/c vrt air 2!!, !!w a/c roll 4!!, !!w a/c aoa 3!!, !!begin traj alt 3!!, with pitch of !+max range angle+!. The altitude difference between !!wpn sep posn!! and the !!traj end posn!! is also assumed !!traj alt diff 3!!.

!+sk rls pitch+! An aircraft at !!wpn sep!! with this pitch and other a/c state descriptions would cause the !!shrike!! to follow a trajectory through the !!traj end posn!!. This model assumes an a/c state of !!w a/c vrt air 2!!, !!w a/c roll 4!!, !!w a/c aoa 3!! and !!begin traj alt 3!!. The difference between the !!wpn sep posn!! and the !!traj end posn!! is assumed !!traj alt diff 3!! and !!traj range 1!!.

!+slt rng imp pt+!
The slant range from the a/c at !!wpn sep!! to the !!imp posn!! of an !!a/g launch wpn!!. At !!wpn sep!! the a/c is assumed to have !!w a/c aoa 5!!, !!w a/c airspd 3!!, !!w a/c pitch 3!!, and !!w a/c roll 6!!. A (device estimate) slant range to !!traj end!! is !!w sr ip 1!!.

4.2. Dictionary of !!wbm input!! !!term!!s

!!wbm input!! Input to this module that may be used by one or more of its models. This may be any other !!term!! defined in this section.

The following series of terms are !!wbm input!!s formed by replacement of a "#" by an integer. Each defined term with a "#" in the name forms a collection of terms called a term class. Individual terms are named by the replacement of the "#" in a term class by one of the integers listed with the term class definition.

Term class	Set of numbers that may textually replace "#" in legal !!term!!s	Definition
!!begin traj alt #!!	1..3	a/c !!altitude asl!!
!!traj alt diff #!!	1..3	The altitude of the !!wpn sep posn!! minus the altitude of the !!traj end posn!!
!!traj range #!!	1	The horizontal distance between the !!wpn sep posn!! and the !!traj end posn!!
!!w a/c airspd #!!	1..4	!!a/c airspd!!
!!w a/c aoa #!!	1..5	!+DI.AOA+!
!!w a/c drft a #!!	1..2	!+drift angle+!
!!w a/c hdg #!!	1	!!a/c heading!!
!!w a/c pitch #!!	1..4	!!a/c pitch!!
!!w a/c roll #!!	1..6	!!a/c roll!!
!!w a/c vrt air #!!	1..2	!+a/c airmass veloc+!
!!w sr ip #!!	1..2	Slant range from !!wpn sep posn!! to !!traj end posn!!.
!!weap tt #!!	1..2	Weapon time of travel from !!wpn sep!! to !!traj end!!

4.3. !!term!!s other than Input

!!a/g launch wpn!!
A !!launch wpn!! whose !!traj end!! is at the ground level.

!!a/a launch wpn!!
A !!launch wpn!! whose !!traj end!! is not at the ground level.

!!current wpn!! The weapon whose characteristics are on the interface of DI.WCM and release conditions are on the DI.WRS interface.

!!launch wpn!! A weapon whose weapon class is GN or RK.

!!release wpn!! A weapon whose weapon class is HD, MF, MD, OD, OR, SH, SL, SM, SOD, or SSH.

!!shrike!! A weapon whose weapon class is SK.

!!src wbm!! The variable, constant, literal, or =DB.dbitem= that is to be used by PM.ACM as the source for an !!wbm input!!. When this module needs a value of the !!wbm input!! it will use the value of this entity.

!!traj end!! The end of the !!current wpn!! trajectory after separation from the a/c.

!!traj end posn!!
The position of !!traj end!!.

!!walleye!! A weapon whose weapon class is WL.

!!wbm runtime change!!
The source of this !!wbm input!! may (will not) be changed during runtime if this state is true

(false).

!!wpn sep!! The separation of the !!current wpn!! from the a/c. For the weapon whose weapon class is GN this is the separation of the bullet from the a/c.

!!wpn sep posn!!
The position of the a/c at !!wpn sep!!.

5. Undesired event dictionary

%not lnch weap%
!!current wpn!! is not a !!launch wpn!!.

%not rls weap%
!!current wpn!! is not a !!release wpn!!.

%not rls or sk%
!!current wpn!! is not a !!release wpn!! nor a !!shrike!!.

%not rls or wall%
!!current wpn!! is not a !!release wpn!! nor a !!walleye!!.

%not shrike% !!current wpn!! is not a !!shrike!!.

%wpn out of rng%
!+rls within rng+! is not true.

%%err wlist%%
The members of the wbmslist p2 are !!wbm input!! of a term class that is different from the term class listed in the definition of p1.

%%itm in wlist%%
The !!wbm input!! is a member of the wbmslist of a previous invocation of ++SAME_...++.

%%itm not init%%
The !!wbm input!! has not been set with ++S_..._#_INIT++.

%%wbm already init%%
++S_..._#_INIT++ has been invoked for the !!wbm input!!.

%%wbm chg method%%
The !!wbm runtime change!! for this input is false.

%%wbm src not init%%
At least one source of input for this access program is not a member of a wbmslist of a previous invocation of ++SAME_...++ and also has not been set with ++S_..._#_INIT++.

%%wlist in wlist%%
One or more members of the wbmslist p2 are members of a wbmslist of a previous invocation of ++SAME_...++.

%%wlist member init%%
++S_..._#_INIT++ has been invoked for one or more members of the wbmslist p2.

%%wlist not init%%
++SAME_...++ has not been invoked with wbmslist p2.

6. System generation parameters None

APPENDIX A

DESIGN ISSUES

(1) The implementation of this module presents a problem because the ranges and resolutions of the sources is not strictly known until system generation time. One solution is to write templates or parameterized modules, in which the translator fills in the ranges and resolutions at system generation time. This places a large proportion of the writing of this module into sysgen time. An alternative is to use the information provided by other modules of their intention to invoke the source setting programs. This sacrifices ease of change, but enables a larger proportion of the code to be written before the program is assembled. The latter course enables writing of pseudocode that is more explicit about ranges and resolutions before sysgen time, and is allowed in the implementation of this module.

1. PM.ACM

(1) Should input to the PM programs be explicitly listed on the interface? We have in other modules excluded placing a list of inputs on the interface document. This was done because the association of input with output was a part of the secret of the module. For the PM modules, the list of inputs is not a secret. The source of the input may change during the execution of the program, and that is a secret of a module outside of the PM. The users of PM programs must designate sources of inputs.

(2) Should the simplifying assumptions about the world necessary for modeling with reasonable speed be on the interface? This question is divided into two parts, the assumptions embodied in a choice of inputs, and secondly the choice of constants and algorithm given the input list. As discussed above, the input list should be on the interface, but the algorithm and physical constants need not be known outside of the module, and constitute the module's secrets.

(3) The pullup programs in this module are a portion of the algorithm in the original program. The pullup programs in the original program contain a mixture of assumptions about the weapon characteristics, weapon trajectories and aircraft behavior. The programs on this interface allow a user to model the aircraft behavior during the pullups.

(4) Some of the output is calculated from a representation of the history of the value of sources. In previous versions the filtering was explicit on the interface, with the period and a representation of the filtering time constant as parameters to this module. This version of the interface hides the details of the filtering, or even whether filtering is done at all. The only indications of filtering are in the accuracy specifications. The users of this module need not know more.

(5) There was previously language on the interface differentiating a model and an implementation of a model. Through heated debate the line separating the two was judged to be arbitrary, and the distinction was to be avoided for clarity if possible. Hopefully the present interface uses the word "model" in a way that is clear to its readers.

2. PM.ECM

(1) There are several different models that produce calculations of the same output quantity that are different because of different assumptions of the models. The users of this module do not need to know which model is chosen.

(2) There are several methods of updating module-local values of variables. The proposals under consideration were direct parameter passing, supplying a program name that could be invoked to supply a current value and the present scheme. Use of input parameters with the output value requesting programs is inefficient if the values of the input parameters do not change often. A scheme for communicating the names of programs outside PM that would be invoked when PM needs the values was dropped because it should not be

the responsibility of PM to update the values. The present scheme is more efficient while still retaining flexibility for the values and even sources of input to change.

(3) Originally there was a program on this interface returning the local rotation of the earth in coordinates that represented local East, North and Vertical of a location on the earth's surface. Use of this program was found to be more expensive than merely putting the rotation rate on the interface in the quite common case of conversion of the angular rotation to a coordinate frame that does not correspond to East, North and Vertical on a location on the earth's surface. The alternative of leaving the old program on the interface and merely calling for the angular rate at the North or South pole was rejected as also having an excessive overhead.

(4) We would like to hide the secret of our model of the shape of the earth. We cannot completely hide this, because the implementation of PM.ACM may be significantly more efficient if it is known that our model is an ellipsoid of revolution. This is the reason for the term !+ellipticity+!.

3. PM.HFM

(1) The current OFP has no specific human reaction time built in to it; rather, there is a safety margin for the entire aircraft pullup that includes the aircraft reaction time also. We will have to guess what fraction of that is the human reaction time.

(2) The case could be made that update rates are device-dependent. For example, a device with a long-persisting image would require a lower rate than one with a short-lived image to simulate continuous motion. However, we believe that that is true only in the extreme, and that for likely A-7 replacement devices the value is human-dependent only.

4. PM.TBM

(1) The programs that define the target position, speed, and heading take a time parameter (as opposed to having the module assume that the information is true when the program is called) because the data might be old. In an extreme example, the observation may come from another aircraft or even a ground spotter, be relayed to our aircraft, and be entered through the panel. By the time the data would get to this module, the observation would be several minutes old.

(2) The decision was made to base elapsed time from the first invocation of a program specifying the position of a target. The other choices considered were basing the time on the previous invocation of such a program, or the first time the !!tgt ident!! was allocated for a target. The time of first allocation may be far removed from the time the aircraft is close to the target and thus was not used. Basing the time on the previous invocation results in an undefined value the first time that one of the position programs is invoked for a target. Therefore the third choice of time elapsed (positive or negative) since the first invocation of one of the two programs setting the position or relative position was adopted.

(3) Output estimates of slant range, azimuth, and elevation are defined with the same term as the input estimates. This was done to simplify the interface specification. The alternative of defining the differences between the terms other than one is input and the other output to this module was rejected as revealing a secret of the module.

(4) A previous version hid the design decision that members of the set of targets are mapped to the =integer=s. Since this is a decision that is not likely to change, it was made visible on this interface.

(5) This module is unimplemented for the following reasons. It can be ascertained from our Requirements document that in the OFP there is only one target at a time, and that it is assumed to be stationary. By our design rules, then, that knowledge is incorporated into a Behavior-Hiding module (in our case, the Reference Point submodule of Shared Services). Since that module "knows" that none of the features provided by PM.TBM are needed, it need not ever invoke the facilities of this module. Should the requirements change one day and this module become necessary, then it will be implemented and SS.SYSVAL.REFPT will be changed to make use of it -- as is the case with any other unimplemented feature anywhere in our system.

5. PM.WBM

(1) The differences in trajectory calculations between air to ground guns and rockets are hidden. The differences in trajectory calculations are not needed outside of this module. Other differences between air to ground guns and rockets are not needed on this interface.

(2) At one time we offered slant range to impact point for release weapons. This may be calculated from the information at the present interface, if the state of the aircraft is known.

(3) Users need not request information through this module that is available directly from the DI.WCM. This includes the maximum range of a launch weapon.

(4) There need not be a program for resetting iterative calculations that need to be notified when a new weapon is designated. This module may obtain that information directly from DI.WCM. The calculations in question include range calculations for medium drag weapons and slant range for launch air to ground weapons.

(5) At one time it was assumed that a module outside of this one translated the weap_class of DI.WCM and other information into trajectory behavior classes . After discussion it was recognized that the module responsible for calculation of the weapons trajectories should also classify these trajectories into drag types. These behavior drag types are a secret of this module.

(6) The minimum time to release for multiple-release weapons is a policy, not a behavior of the weapon-launching system. As such, it is not on this interface.

(7) The slant range to impact offered on this interface is the slant range in the earth coordinate system. The slant range in calculated in the airmass coordinate system uncorrected for wind may be calculated from this value. It is this latter value that is used in the NWC-2 version of the OFP to determine whether a target is within useful range.

(8) The downrange miss distance rate (used for calculation of the release time interval) is assumed calculated outside of this module, because changes in motion of the aircraft are significant contributions to this quantity.

(9) Weapon trajectory calculations start from the time that the weapon leaves the aircraft. Timing constraints such as !+DI.rack delay time+! !+DI.preparation time+! and !+DI.rack curve class+! are not used by this module. Calculations of the lead time between sending pulses to the DI module and the actual release are not calculated by this module.

(10) All of the output quantities are specifically unfiltered unless they are described as filtered or smoothed (azimuth and elevation of air to ground weapons).

(11) There are two or more parts to the trajectory calculations of GN and RK. These are the calculation of initial velocity upon leaving the aircraft, and the remaining trajectory. The calculation of initial velocity is more easily expressed in the reference frame of the aircraft body, but the remaining trajectory is more easily expressed in an earth or airmass coordinate system. Since the users need the result of the trajectory eventually converted to the aircraft body reference frame it was decided to have the results expressed on this interface in that reference frame. An alternative considered was to not hide on this interface the fact that there are two parts to the trajectory calculation.

(12) Previous versions used the drag characteristics of release weapons on the interface to disallow behavior calculations that were defined but not implemented. All of the calculations that were offered for low drag weapons have a physical meaning and are computable for all release weapons. Drag types have been removed from this interface, because the differences are not behavior, but are a policy decision whether to calculate certain trajectory quantities for a set of weapons.

(13) The maximum range angle in previous versions was assumed to be constant for all (low drag) release weapons. This assumption was relaxed because this may be likely to change.

(14) The air to air launch calculations in NWC-2 assume that the elevation angle to impact point is equal to the lead angle (the angle by which to lead the target to compensate for the target motion during weapon travel), and this elevation angle is not calculated. This elevation angle calculation could be unimplemented in the final (SCR) version.

APPENDIX B

IMPLEMENTATION NOTES

1. PM.ACM

(1) This is a list of any input dependencies of !+term+!s not completely described by the definitions in PM.ACM.4.1.

!+a/c path approach+!

#max indic normal accel# and !+ECM.gravity plus+! at a/c are used in addition to the inputs listed in the !+term+! definition.

!+a/c speed indic accel+!

#max indic normal accel# and !+ECM.gravity plus+! at a/c are used in addition to the inputs listed in the !+term+! definition.

!+a/c horiz disp indic accel+!

#max indic normal accel# and !+ECM.gravity plus+! at a/c are used in addition to the inputs listed in the !+term+! definition.

!+a/c vert disp indic accel+!

#max indic normal accel# and !+ECM.gravity plus+! at a/c are used in addition to the inputs listed in the !+term+! definition.

(2) !+PM.a/c rate East+! = - (second component of !!a/c h vel 9!! × 1 radian) / (!!dist 1!! + !!a/c alt asl 5!!)

A range of -/+ (1:P-12 - 1:P-26) radians/second and resolution of 1:P-26 radians/second is sufficient for !+a/c rate East+!. The multiplication by 1 radian is only for type conversion; a type conversion with no loss of resolution should replace the multiplication if such a conversion were available.

(3) !+PM.a/c rate North+! = (first component of !!a/c h vel 9!! × 1 radian) × (1 - 2 × !!ratio 1!! × (Cosine(latitude of !!a/c locn 3!!))^2) / (!!dist 1!! + !!a/c alt asl 5!!)

A range of -/+ (1:P-12 - 1:P-26) radians/second and resolution of 1:P-26 radians/second is sufficient for !+a/c rate North+!. The multiplication by 1 radian is only for type conversion; a type conversion with no loss of resolution should replace the multiplication if such a conversion were available.

(4) !+PM.a/c rate Up+! = !+PM.a/c rate North+! × Tangent(latitude of !!a/c locn 3!!)

A range of -/+ (1:P-11 - 1:P-25) radians/second and resolution of 1:P-25 radians/second is sufficient for !+a/c rate Up+!.

(5) The Ya vector, depressed by the !!a/c aoa 3!!, is taken as the velocity with respect to the airmass. A small angle approximation is made so that the Ya component of the airspeed is !!a/c airspd 3!!, and the Za component of the airspeed is (!!a/c airspd 3!! × (- +R_ANGLE_RAD+(!!a/c aoa 3!!))). This vector is transformed into the platform coordinates by use of !!body to platform 1!!, and the vertical component is ignored (set to zero). The result is !+ a/c airmass veloc+!.

(6) !+v spd fm gndspd+! is (!!a/c gnd spd 1!! × TAN(!!a/c fpa 2!!)) when the !!a/c fpa 2!! is between -76° and 76°. At all times !+v spd fm gndspd+! is in the range from #SS.vvel min# to #SS.vvel max#, if it is outside of that range the closest sysgen value is used. The implementation of this program may involve use of the SIN and COS of the angle instead of the TAN, selectively choosing the order of multiplication and division to get intermediate products whose bounds are known while maintaining accuracy.

2. PM.EMC

(1) In the following three implementation notes concerning the coriolis components the division by 1 radian is only for type conversion. If an appropriate type conversion is available, it should be used, thus eliminating any loss of accuracy.

(2) !+PM.coriolis E+! =
[- (2 × !+PM.e rotation+! × Sine(latitude of !!location 26!!) × Second component of !!velocity 1!!) + (2 × !+PM.e rotation+! × Cosine(latitude of !!location 26!!) × Third component of !!velocity 1!!) + (!!angrate 5!! × Third component of !!velocity 1!!) - (!!angrate 6!! × Second component of !!velocity 1!!)] / 1 radian

(3) !+PM.coriolis N+! =
[(2 × !+PM.e rotation+! × Sine(latitude of !!location 26!!) × First component of !!velocity 1!!) - (!!angrate 4!! × Third component of !!velocity 1!!) + (!!angrate 6!! × First component of !!velocity 1!!)] / 1 radian

(4) !+PM.coriolis V+! =
[-(2 × !+PM.e rotation+! × Cosine(latitude of !!location 26!!) × First component of !!velocity 1!!) - (!!angrate 5!! × First component of !!velocity 1!!) + (!!angrate 4!! × Second component of !!velocity 1!!)] / 1 radian

(5) !+location ang+! is a latitude/longitude pair that is computed as follows:

```
Retained data SAVE: angle, initial value rad(0)

IF +ABSV+((+R_ANGLE_RAD+(!!angrate 2!! × !!timeint 1!!)) <
        Cosine(latitude of !!location 25!!))
THEN
        SAVE := !!angrate 2!! × !!timeint 1!! / Cosine(latitude of !!location 25!!))
ENDIF
Longitude of !+location ang+! := Longitude of !!location 25!! + SAVE
Latitude of !+location ang+! := Latitude of !!location 25!! - (!!angrate 1!! × !!timeint 1!!)
```

(6) All subtraction and addition of entities of type longitude is AT.MOD2

(7) This note is for !+ground range+! from !!location 4!! to !!location 5!!, abbreviated GRLOC, and !+t bearing+! from !!location 4!! to !!location 5!!, abbreviated BRGLOC. The algorithm, ranges and resolutions are sufficient to implement the programs on the interface.

The following is a list of entities referred to in this note. Some of these ranges and resolutions are given for data that is not retained. In these cases the entities are named for convenience and specification of ranges and resolutions, and need not be named entities in the code.

Entity Name	Type	Range	Resolution
Aj	REAL	0 to (1 - Res)	1:P-30
Ajfirst	REAL	0 to (4 - Res)	1:P-28
BRGLOC	ANGLE	0 to (1 - Res) circles	1:P-15 circles
Denompartial	REAL	-+ (2 - Res)	1:P-28
Denominator	REAL	-+ (2 - Res)	1:P-28
#2D range limit#	DISTANCE	22.7 nmi to 36 nmi	1:P-5 nmi
FLATTEST	ANGLE_SQ	Constant	value/1000
FLAT	BOOLEAN		
GRLOC	DISTANCE	0 to (1:P13 - 1:P-2)nmi	1:P-2 nmi
#gr limit#	DISTANCE	3500 nmi to 3800 nmi	10 nmi
KSS1	REAL	Constant	1:P-23
KSS2	REAL	Constant	1:P-28
KSS3	REAL	Constant	1:P-31
LATDIF	ANGLE	-+ (1 circle - Res)	1:P-30 circles
Numerator	REAL	-+ (2 - Res)	1:P-29
PHICLM	ANGLE	-+ (2 circles - Res)	1:P-29 circles
Theta1	REAL	-+ (4 - Res)	1:P-28
Theta2	REAL	-+ (1:P-3 - Res)	1:P-33
Thetaj	REAL	-+ (2 - Res)	1:P-29
Xj	REAL	-+ (0.5 - Res)	1:P-31

CONSTANTS:

Constant	Value
#2D range limit#	32 nmi.
#gr limit#	3606 nmi.
FLATTEST	+ANGLE_SQ_R_RAD+((#2D range limit#/ !+equator rad+!)ˆ2)
KSS1	2.0000065914
KSS2	0.0833005696
KSS3	0.009779758

Code for either value:

PHICLM := (Longitude(!!location 5!!) - Longitude(!!location 4!!)) × Cosine(Latitude(!!location 4!!))
LATDIF := Latitude(!!location 5!!) - Latitude(!!location 4!!)

FLAT := (PHICLMˆ2 + LATDIFˆ2) < FLATTEST
IF FLAT THEN
GRLOC := +R_ANGLE_RAD+(SQRT(LATDIFˆ2 + PHICLMˆ2)) × !+hrc+! at !!location 4!!
BRGLOC := <MOD1 ArcTangent(PHICLM/LATDIF)>
ELSE { not FLAT}
Ajfirst := [Sine(Longitude(!!location 4!!)) × Cosine(Latitude(!!location 4!!)) - Sine(Longitude(!!location 5!!)) × Cosine(Latitude(!!location 5!!))]ˆ2 + [Sine(Latitude(!!location 5!!)) - Sine(Latitude(!!location 4!!))]ˆ2 + [Cosine(Longitude(!!location 5!!)) × Cosine(Latitude(!!location 5!!)) - Cosine(Longitude(!!location 4!!)) × Cosine(Latitude(!!location 4!!))]ˆ2

Aj := MIN(Ajfirst, (1 - 1:P-30))
Xj := SQRT(Aj) / 2.00000000
Theta2 := KSS2 + Aj × KSS3
Theta1 := KSS1 + Aj × Theta2
Thetaj := Xj × Theta1
Numerator := Sine(Longitude(!!location 5!!) - Longitude(!!location 4!!)) × Cosine(Latitude(!!location 5!!)) × Cosine(Latitude(!!location 4!!))

Denompartial := Sine(Latitude(!!location 4!!)) × Aj / 2.0000000
Denominator := Sine(Latitude(!!location 5!!)) - Sine(Latitude(!!location 4!!)) + Denompartial
BRGLOC := <MOD1 Arctangent(Numerator/Denominator)>
GRLOC := Thetaj × !+equator rad+!

ENDIF

(8) !+hrc+! is calculated using the equation referenced in [MRC] page 6-6 and the [GLOSS] definition of RM.

(9) The values of 20,925,741 feet {[MRC] page 6-11}, resolution 2,000 feet, or 6076.116 nautical miles {[GLOSS] definition of KNM}, resolution 0.6 nmi are sufficient values for !+equator rad+!.

(10) !+gravity plus+! at the indicated location and altitude may be calculated as follows {reference [MRC] and [NWC-2] }

!+gravity plus+! := 32.0882 ft/secˆ2 + 0.169682 ft/secˆ2 × [Sine(Latitude(!!location 28!!))]ˆ2 - 3.1E-6 secˆ-2 × !!distance 22!!

(11) A resolution of 1:P-27 radians/second is sufficient for !+e rotation+!.

(12) A resolution of 1:P-22 is sufficient for !+ellipticity+!.

3. PM.WBM

(1) Any limiting of input variables must be done inside of the module. One of the variables that is limited in the NWC-2 calculations is the vertical travel of the weapon.

(2) The airmass reference frame is usually used for trajectory calculations.

APPENDIX C

ASSUMPTIONS LISTS

1. BASIC ASSUMPTIONS

1.1. PM.ACM

(1) The airframe coordinate system is a set of orthogonal axes stationary with respect to the body of the aircraft.

(2) Requesters of information from this module need not be the suppliers of input. The users of the +G...+ programs do not care if a particular kind of input or input source is used to calculate an output value.

(3) A series of programs that assume a constant indicated acceleration are sufficient to model a/c pullup behavior.

(4) The inputs on the present interface, plus PM.ECM programs modeling the earth characteristics such as wind, gravity, and earth inertial velocity are sufficient to calculate the output terms.

(5) It is not necessary to know the thrust or drag of the aircraft during a flight with constant indicated acceleration of #max indic normal accel#.

1.2. PM.ECM

(1) Requestors of information from this module need not be the suppliers of input. The users of the +G...+ programs do not care if a particular kind of input or input source is used to calculate an output value.

(2) Users of this module do not need to know information about the models used by this module.

(3) The sum of the earth's gravity and centripetal force results in an acceleration along !!p local vert!! in the negative direction.

(4) !!altitude asl!! is measured along the !!p local vert!! axis from sea level.

(5) The input states listed in the definition of the output !+term+!s are sufficient to describe the !+term+! with acceptable accuracy.

1.3. PM.HFM

(1) If a series of slightly different static scenes is displayed in rapid succession, the human eye perceives continuous motion. The perception depends on the similarity of the successive scenes, and the display update rate.

(2) In the A-7 OFP, the static scenes used to simulate continuous motion include a small symbol that moves small amounts across a fixed background, and a background that moves in small amounts past a fixed point. Each display will be similar enough to its predecessor to preserve the illusion of motion, given an update rate that is sufficiently fast.

(3) There is a minimum amount of time required for a human to react to a stimulus. "React" means to take some action, such as moving a control. This submodule provides a reaction time that takes into account that minimum time interval, plus a generally accepted safety margin. The action that the pilot takes is to avoid a catastrophe.

(4) The reaction time is produced assuming that the pilot will notice the stimulus as soon as it occurs. It does not include any time required for the stimulus to come to the pilot's attention.

(5) The reaction time and update frequency provided by this module will suffice for any human pilot. There will never be a case when the values should be tailored to a particular pilot or class of pilots.

1.4. PM.TBM

(1) The position of a target can be predicted by knowledge of target position at one time, velocity, and time elapsed since target was at known position.

(2) User programs will provide sufficiently frequent and accurate reports of the position, speed, and heading of a target for the production of sufficiently accurate predictions of target movement.

(3) Target speed and heading may be extrapolated from position reports if not reported explicitly or as frequently as positions. Target acceleration and direction changes between reports can be ignored. Straight paths are assumed between position or heading change reports.

(4) Attitude of a target will not be a factor in tracking the position of the target.

(5) Target speed and heading need not be calculated for the user. Position and distance at specified time intervals is sufficient.

(6) It is more convenient for the users of this module to receive position relative to the aircraft than to receive position relative to the earth.

(7) Any likely implementation of this module would have an explicit function from target to =integer=.

1.5. PM.WBM

(1) Each release weapon may have a maximum range angle independent of the altitude or other state of the aircraft.

(2) For release weapons, weapon release velocity (airspeed, heading, and pitch of weapon at time of release) is determined by weapons platform (aircraft) velocity pitch and heading at time of release and weapon ejection velocity.

(3) A maximum of one weapon is on the currently active weapon station. This is the weapon whose characteristics are on the interface of DI.WCM and the same weapon whose release conditions are on the DI.WRS interface.

(4) For any set of release parameters of a release weapon trajectory excluding pitch angle (airspeed, height of release, height of burst), there is one pitch angle of release for each release weapon or Shrike that will result in maximum horizontal range.

There are two a/c pitch angles for release weapons that will result in trajectories through positions with the same range and height of burst if the range is smaller than this maximum horizontal range. These angles diverge from the maximum range angle as the range is decreased. The larger angle is referred to as the upper solution or loft angle while the other angle is referred to as the lower solution or dive angle.

There is only one such pitch angle for the Shrike.

(5) This module may determine when the value of the !!current wpn!! changes.

(6) Weapon behavior need be modelled for only one weapon at a time, the !!current wpn!!.

(7) Weapons are either release, launch, Shrike, or Walleye.

(8) The users of this module know the difference between air to air launch weapons and air to ground launch weapons. Different assumptions are used in the calculation of the two classes of weapons. These different assumptions are the result of a policy decision that is not the secret of this module.

(9) Requesters of information from this module need not be the suppliers of input. The users of the +G...+ programs do not care if a particular kind of input or input source is used to calculate an output value.

(10) The inputs on the present interface, plus PM.ECM programs modeling the earth characteristics such as wind and gravity, Filter programs, DI.WRS and DI.WCM are sufficient to calculate the output terms.

(11) The azimuth to the impact point for air to ground weapons will not be used without the elevation to that point, also the inverse is true. The same is not true for the elevation and the azimuth to the impact point for air to air weapons.

2. UNDESIRED EVENTS ASSUMPTIONS

2.1. PM.ECM

(1) Sources of input will be initialized only once.

(2) The source of input value will be known to be constant or vary at runtime.

(3) The direction from one =earth_locn= to itself is undefined. Sources will not specify such direction, nor will users of this module ask for this type of direction.

(4) The user may declare the sources of a list of data items to be the same entity, and thereafter sources of the data items may not vary individually.

(5) The sources of input must be known to this module before they are used.

2.2. PM.TBM

(1) User programs will not request tracking of more targets than a value specified at system generation time.

(2) A user program will not reference an unassigned target identifier.

(3) A user program will not request a position for a target which does not have a previously reported position.

(4) Target motion is not sufficiently described without specification of all three components of motion or the equivalent.

2.3. PM.WBM

2.3.1. Sysgen-time

(1) The sources of input must be known to this module before they are used.

(2) Sources of input will be initialized only once.

(3) The source of input value will be known to be constant or vary at runtime.

(4) The user may declare the sources of a list of data items to be the same entity, and thereafter sources of the data items may not vary individually.

2.3.2. Run-time

(1) Release weapons, launch weapons, Shrike and Walleye each have their sets of weapons behavior measures. The values of those measures will only be requested when an appropriate weapon is designated.

(2) For any set of release parameters of a release weapon trajectory excluding pitch angle (airspeed, height of release, height of burst), there is a limit to the range of a weapon. For ranges greater than this limit there are no possible pitches that would result in a trajectory through the indicated position (burst height and range).

APPENDIX D

UNIMPLEMENTED FEATURES

Not all the capabilities described in this document have been provided in the current version of the Physical Models. A few facilities, which are not currently needed by the application program, have not been implemented. An attempt to use an absent facility will result in an undesired event in the development version. The unimplemented features are described below.

1. PM.ACM

(1) The programs that set the source of !!a/c pitch 9!!, !!a/c pitch 5!!, !!a/c roll 5!!, !!a/c roll 6!!, and !!a/c t hdg 5!! are unimplemented. are unimplemented.

2. PM.TBM

(1) This module is completely unimplemented.

3. PM.WBM

(1) The programs that set the source of !!w a/c roll 1!!, !!w a/c drft a 1!!, !!w a/c roll 2!!, !!w a/c drft a 2!!, !!w a/c roll 3!!, !!w a/c roll 4!!, and !!w a/c aoa 4!! are unimplemented.

(1) The following undesired events are not implemented. If used under the conditions corresponding to the undesired events, effect programs will have no effect and value programs will return terms with undefined values.

%not lnch weap%
%not rls weap%
%not rls or sk%
%not rls or wall%
%not shrike%
%wpn out of rng%

4. PM.WBM

!!a/c airspd!!
!!a/c heading!!
!!azimuth!!
!!elevation!!
!!imp posn!!

PM.INDEX: INDICES TO DOCUMENT

1. Access Programs (including programs that report events)

+ASGN_TGT_IDENT+ 10-23
+FREE_TGT+ 10-23
+G_A/A_IMP_AZ+ 10-27
+G_A/A_IMP_EL+ 10-27
+G_AC_AIR_VEL+ 10-2
+G_AC_ALT+ 10-1
+G_AC_DRIFT_ANG+ 10-2
+G_AC_GNDTRK_ANG+ 10-2
+G_AC_HDG_RATE+ 10-1
+G_AC_H_DISP+ 10-2
+G_AC_H_DISP_PU+ 10-2
+G_AC_HVEL_AIRSPD+ 10-1
+G_AC_HVEL_GNDSPD_1+ 10-1
+G_AC_HVEL_GNDSPD_2+ 10-1
+G_AC_HVEL_INCR+ 10-1
+G_AC_HVEL_LEVER+ 10-1
+G_AC_LOCN+ 10-1
+G_AC_NORMAL_ACCEL+ 10-1
+G_AC_PITCH_RATE+ 10-1
+G_AC_PTH_APPRCH+ 10-2
+G_AC_RHOX+ 10-2
+G_AC_RHOY+ 10-2
+G_AC_RHOZ+ 10-2
+G_AC_ROLL_RATE+ 10-1
+G_AC_SPD_PU+ 10-2
+G_AC_V_DISP+ 10-2
+G_AC_V_DISP_PU+ 10-2
+G_AC_VSPD_AIRSPD+ 10-1
+G_AC_VSPD_ALT+ 10-1
+G_AC_VSPD_GNDSPD+ 10-1
+G_A/G_AIM_PT+ 10-27
+G_AIR_DENS+ 10-12
+G_AIR_PRESS+ 10-12
+G_ALT_BARO+ 10-11
+G_ALT_DIFF+ 10-11
+G_ALT_SR+ 10-11
+G_BRG_GR+ 10-12
+G_CORIOL_E+ 10-12
+G_CORIOL_N+ 10-12
+G_CORIOL_V+ 10-12
+G_DENS_RATIO+ 10-12
+G_E_DIST+ 10-11
+G_ELEV_ANG_ALT+ 10-12
+G_ELLIPTICITY+ 10-13
+G_EQUAT_RAD+ 10-13

+G_GRAV+ 10-12
+G_GR_SR+ 10-12
+G_HRC+ 10-12
+G_HRC_E+ 10-12
+G_HRC_N+ 10-12
+G_LOCN_ANGRATE+ 10-11
+G_LOCN_BRG_GR+ 10-11
+G_LOCN_DIST+ 10-11
+G_LOCN_ELEV+ 10-11
+G_LOCN_ORIENT+ 10-11
+G_LOCN_SR+ 10-11
+G_MAG_VAR+ 10-12
+G_MAX_RANGE_ANGLE+ 10-26
+G_N_DIST+ 10-11
+G_ORIENT+ 10-11
+G_PILOT_REACT_TIME+ 10-22
+G_RLS_B_BEARING+ 10-26
+G_RLS_PITCH_LO+ 10-27
+G_RLS_PITCH_SHR+ 10-27
+G_RLS_PITCH_UP+ 10-26
+G_RLS_RANGE+ 10-26
+G_RLS_T_TRAV+ 10-26
+G_RLS_WITHIN_RANGE+ 10-26
+G_ROT+ 10-13
+G_SLT_RNG_GR+ 10-12
+G_SLT_RNG_L2+ 10-11
+G_SOUND_SPD+ 10-12
+G_SR_A/G_I_P+ 10-27
+G_TGT_POSN_REL+ 10-24
+G_TGT_SR+ 10-24
+G_UPDATE_FREQ+ 10-22
+G_VERT_AX_D+ 10-13
+G_WIND_VELOC+ 10-12
+RESET_MODEL_H_V+ 10-4
+S_AC_AIRSPD_ACM_SL_SRC+ 10-3
++S_AC_AIRSPD_#_INIT++ 10-2
+S_AC_AIRSPD_#_SRC+ 10-2
+S_AC_ALT_ACM_SL_SRC+ 10-3
++S_AC_ALT_#_INIT++ 10-2
+S_AC_ALT_#_SRC+ 10-2
+S_AC_AOA_ACM_SL_SRC+ 10-3
++S_AC_AOA_#_INIT++ 10-2
+S_AC_AOA_#_SRC+ 10-2
+S_AC_DRIFT_A_ACM_SL_SRC+ 10-3
++S_AC_DRIFT_A_#_INIT++ 10-2
+S_AC_DRIFT_A_#_SRC+ 10-2
+S_AC_FPA_ACM_SL_SRC+ 10-3
++S_AC_FPA_#_INIT++ 10-2
+S_AC_FPA_#_SRC+ 10-2
+S_AC_GNDSPD_ACM_SL_SRC+ 10-3
++S_AC_GNDSPD_#_INIT++ 10-2
+S_AC_GNDSPD_#_SRC+ 10-2
+S_AC_H_VEL_ACM_SL_SRC+ 10-3
++S_AC_H_VEL_#_INIT++ 10-2
+S_AC_H_VEL_#_SRC+ 10-2
+S_AC_LOCN_ACM_SL_SRC+ 10-3

++S_AC_LOCN_#_INIT++ 10-2
+S_AC_LOCN_#_SRC+ 10-2
+S_AC_PITCH_ACM_SL_SRC+ 10-3
++S_AC_PITCH_#_INIT++ 10-2
+S_AC_PITCH_#_SRC+ 10-2
+S_AC_ROLL_ACM_SL_SRC+ 10-3
++S_AC_ROLL_#_INIT++ 10-2
+S_AC_ROLL_#_SRC+ 10-2
+S_AC_SPEED_ACM_SL_SRC+ 10-3
++S_AC_SPEED_#_INIT++ 10-2
+S_AC_SPEED_#_SRC+ 10-2
+S_AC_T_HDG_ACM_SL_SRC+ 10-3
++S_AC_T_HDG_#_INIT++ 10-2
+S_AC_T_HDG_#_SRC+ 10-2
+S_AC_V_VEL_ACM_SL_SRC+ 10-3
++S_AC_V_VEL_#_INIT++ 10-2
+S_AC_V_VEL_#_SRC+ 10-2
++SAME_AC_AIRSPD++ 10-3
++SAME_AC_ALT++ 10-3
++SAME_AC_AOA++ 10-3
++SAME_AC_DRIFT_A++ 10-3
++SAME_AC_FPA++ 10-3
++SAME_AC_GNDSPD++ 10-3
++SAME_AC_H_VEL++ 10-3
++SAME_AC_LOCN++ 10-3
++SAME_AC_PITCH++ 10-3
++SAME_AC_ROLL++ 10-3
++SAME_AC_SPEED++ 10-3
++SAME_AC_T_HDG++ 10-3
++SAME_AC_V_VEL++ 10-3
++SAME_ANGLE++ 10-13
++SAME_ANGRATE++ 10-13
++SAME_B_TO_P++ 10-3
++SAME_DISP++ 10-3
++SAME_DIST++ 10-13
++SAME_FLT_TIME+ 10-3
++SAME_LAT++ 10-13
++SAME_LOCN++ 10-13
++SAME_LONG++ 10-13
++SAME_MOM_ARM++ 10-3
++SAME_ORIENT++ 10-13
++SAME_POSN++ 10-13
++SAME_PRESS++ 10-13
++SAME_RATIO++ 10-3
++SAME_TIME++ 10-13
++SAME_TRAJ_A_DIFF++ 10-28
++SAME_TRAJ_ALT++ 10-28
++SAME_VELOC++ 10-13
++SAME_WPN_AIRSPD++ 10-28
++SAME_WPN_AOA++ 10-28
++SAME_WPN_A_VEL++ 10-28
++SAME_WPN_D_A++ 10-28
++SAME_WPN_PTCH++ 10-28
++SAME_WPN_ROLL++ 10-28
++SAME_WPN_SR_IP++ 10-28
++SAME_WPN_T_T++ 10-28

++S_ANGLE_#_INIT++ 10-13
+S_ANGLE_SLIST_SRC+ 10-13
+S_ANGLE_#_SRC+ 10-13
++S_ANGRATE_#_INIT++ 10-13
+S_ANGRATE_SLIST_SRC+ 10-13
+S_ANGRATE_#_SRC+ 10-13
+S_B_TO_P_ACM_SL_SRC+ 10-3
++S_B_TO_P_#_INIT++ 10-2
+S_B_TO_P_#_SRC+ 10-2
+S_DISP_ACM_SL_SRC+ 10-3
++S_DISP_#_INIT++ 10-2
+S_DISP_#_SRC+ 10-2
++S_DIST_#_INIT++ 10-13
+S_DIST_SLIST_SRC+ 10-13
+S_DIST_#_SRC+ 10-13
+S_FLT_TIME_ACM_SL_SRC+ 10-3
++S_FLT_TIME_#_INIT++ 10-2
+S_FLT_TIME_#_SRC+ 10-2
++S_LAT_#_INIT++ 10-13
+S_LAT_SLIST_SRC+ 10-13
+S_LAT_#_SRC+ 10-13
++S_LOCN_#_INIT++ 10-13
+S_LOCN_SLIST_SRC+ 10-13
+S_LOCN_#_SRC+ 10-13
++S_LONG_#_INIT++ 10-13
+S_LONG_SLIST_SRC+ 10-13
+S_LONG_#_SRC+ 10-13
+S_MOM_ARM_ACM_SL_SRC+ 10-3
++S_MOM_ARM_#_INIT++ 10-2
+S_MOM_ARM_#_SRC+ 10-2
++S_ORIENT_#_INIT++ 10-13
+S_ORIENT_SLIST_SRC+ 10-13
+S_ORIENT_#_SRC+ 10-13
++S_POSN_#_INIT++ 10-13
+S_POSN_SLIST_SRC+ 10-13
+S_POSN_#_SRC+ 10-13
++S_PRESS_#_INIT++ 10-13
+S_PRESS_SLIST_SRC+ 10-13
+S_PRESS_#_SRC+ 10-13
+S_RATIO_ACM_SL_SRC+ 10-3
++S_RATIO_#_INIT++ 10-2
+S_RATIO_#_SRC+ 10-2
+S_TGT_HDG+ 10-23
+S_TGT_POSN+ 10-23
+S_TGT_POSN_REL+ 10-23
+S_TGT_SPD+ 10-23
+S_TGT_V_SPD+ 10-24
++S_TIME_#_INIT++ 10-13
+S_TIME+SLIST_SRC+ 10-13
+S_TIME_#_SRC+ 10-13
++S_TRAJ_A_DIFF_#_INIT++ 10-27
+S_TRAJ_A_DIFF_#_SRC+ 10-28
+S_TRAJ_A_DIFF_WBM_SL_SRC+ 10-28
++S_TRAJ_ALT_#_INIT++ 10-27
+S_TRAJ_ALT_#_SRC+ 10-28
+S_TRAJ_ALT_WBM_SL_SRC+ 10-28

++S_TRAJ_RNG_#_INIT++ 10-27
+S_TRAJ_RNG_#_SRC+ 10-28
++S_VELOC_#_INIT++ 10-13
+S_VELOC_SLIST_SRC+ 10-13
+S_VELOC_#_SRC+ 10-13
++S_WPN_AIRSPD_#_INIT++ 10-27
+S_WPN_AIRSPD_#_SRC+ 10-28
+S_WPN_AIRSPD_WBM_SL_SRC+ 10-28
++S_WPN_AOA_#_INIT++ 10-27
+S_WPN_AOA_#_SRC+ 10-28
+S_WPN_AOA_WBM_SL_SRC+ 10-28
++S_WPN_A_VEL_#_INIT++ 10-27
+S_WPN_A_VEL_#_SRC+ 10-28
+S_WPN_A_VEL_WBM_SL_SRC+ 10-28
++S_WPN_D_A_#_INIT++ 10-27
+S_WPN_D_A_#_SRC+ 10-28
+S_WPN_D_A_WBM_SL_SRC+ 10-28
++S_WPN_HDG_#_INIT++ 10-28
+S_WPN_HDG_#_SRC+ 10-28
++S_WPN_PTCH_#_INIT++ 10-27
+S_WPN_PTCH_#_SRC+ 10-28
+S_WPN_PTCH_WBM_SL_SRC+ 10-28
++S_WPN_ROLL_#_INIT++ 10-27
+S_WPN_ROLL_#_SRC+ 10-28
+S_WPN_ROLL_WBM_SL_SRC+ 10-28
++S_WPN_SR_IP_#_INIT++ 10-27
+S_WPN_SR_IP_#_SRC+ 10-28
+S_WPN_SR_IP_WBM_SL_SRC+ 10-28
++S_WPN_T_T_#_INIT++ 10-27
+S_WPN_T_T_#_SRC+ 10-28
+S_WPN_T_T_WBM_SL_SRC+ 10-28

2. Local type definitions

=acmslist= 10-5
=earth_locn= 10-15
=ecmlist= 10-15
=wbmslist= 10-30

3. Dictionary Terms

!+a/a imp pt az+! 10-30
!+a/a imp pt el+! 10-30
!!a/a launch wpn!! 10-32
!+a/c airmass veloc+! 10-5
!!a/c airspd #!! 10-8
!!a/c alt asl #!! 10-8
!+a/c alt asl+! 10-5
!!a/c aoa #!! 10-8
!!a/c drift a #!! 10-8

!!a/c flt time #!! 10-8
!!a/c fpa #!! 10-8
!!a/c gnd spd #!! 10-8
!!a/c h vel #!! 10-8
!!a/c h vel incr #!! 10-8
!!a/c horiz disp!! 10-8
!+a/c horiz disp const vel+! 10-5
!+a/c horiz disp indic accel+! 10-5
!!a/c locn #!! 10-8
!+a/c locn computed+! 10-5
!+a/c path approach+! 10-5
!!a/c pitch #!! 10-8
!!a/c pitch!! 10-8
!+a/c rate East+! 10-5
!+a/c rate North+! 10-5
!+a/c rate Up+! 10-5
!!a/c roll #!! 10-8
!!a/c roll!! 10-9
!!a/c speed #!! 10-8
!+a/c speed indic accel+! 10-6
!!a/c t hdg #!! 10-8
!!a/c v vel #!! 10-8
!+a/c vert disp const vel+! 10-6
!+a/c vert disp indic accel+! 10-6
!!acm input!! 10-7
!!acm runtime change!! 10-8
!+a/g imp pt az+! 10-30
!+a/g imp pt el+! 10-30
!!a/g launch wpn!! 10-32
!+air density+! 10-18
!+air pressure+! 10-19
!!alt diff!! 10-18
!+alt diff sr+! 10-19
!!altitude asl!! 10-18
!+altitude sr+! 10-19
!!angle #!! 10-15
!!angrate #!! 10-15
!+baro alt+! 10-19
!+burst bearing+! 10-30
!+coriolis E+! 10-19
!!current wpn!! 10-32
!+dens ratio+! 10-19
!!displacement #!! 10-8
!!dist #!! 10-8
!!distance #!! 10-15
!+drift angle+! 10-6
!+E dist+! 10-19
!+e rotation+! 10-19
!!ecm input!! 10-16
!!elev ang!! 10-18
!+elev ang alt+! 10-19
!+ellipticity+! 10-19
!+equator rad+! 10-19
!+gnd trk angle+! 10-6
!+gravity plus+! 10-19
!+ground range+! 10-19

!+ground range sr+! 10-19
!+hdg rate+! 10-6
!+horiz veloc fm airspd+! 10-6
!+horiz veloc fm gndspd 1+! 10-6
!+horiz veloc fm gndspd 2+! 10-6
!+horiz veloc fm lever+! 10-6
!+horiz veloc incr+! 10-7
!+hrc+! 10-19
!+hrc East+! 10-19
!+hrc North+! 10-19
!!latitude #!! 10-16
!!launch wpn!! 10-32
!!location #!! 10-16
!+location ang+! 10-20
!+location brg+! 10-20
!+location dist+! 10-20
!+location elev+! 10-20
!+location or+! 10-20
!+location sr+! 10-20
!+lower rls pitch+! 10-30
!+magnetic var+! 10-20
!+max range angle+! 10-31
!!moment arm #!! 10-8
!+N dist+! 10-20
!+normal accel+! 10-7
!+orient diff+! 10-20
!!orientation #!! 10-17
!!p local East!! 10-18
!!p local horiz!! 10-18
!!p local m North!! 10-18
!!p local North!! 10-18
!!p local vert!! 10-18
!+pilot react time+! 10-22
!+pitch rate+! 10-7
!!position!! 10-18
!!pressure #!! 10-17
!!ratio #!! 10-8
!!release wpn!! 10-32
!+rls range+! 10-31
!+rls t travel+! 10-31
!+rls within rng+! 10-31
!+roll rate+! 10-7
!!runtime change!! 10-18
!!shrike!! 10-32
!+sk rls pitch+! 10-31
!!slt rng!! 10-18
!+slt rng alt+! 10-20
!+slt rng gr+! 10-20
!+slt rng imp pt+! 10-31
!+sound speed+! 10-20
!!src acm!! 10-9
!!src ecm!! 10-18
!!src wbm!! 10-32
!+symbol update period+! 10-22
!+t bearing+! 10-20
!!tgt azimuth!! 10-25

!!tgt elevation!! 10-25
!!tgt ident!! 10-25
!!tgt slt rng!! 10-25
!!tgt v spd!! 10-25
!!timeint #!! 10-17
!!traj end!! 10-32
!!traj end posn!! 10-32
!!transit time!! 10-25
!!transport rate!! 10-9
!!true heading!! 10-9
!+upper rls pitch+! 10-31
!+v spd fm airspd+! 10-7
!+v spd fm alt+! 10-7
!+v spd fm gndspd+! 10-7
!!velocity #!! 10-17
!+vert ax dist+! 10-21
!!walleye!! 10-32
!!wbm input!! 10-31
!!wbm runtime change!! 10-32
!+wind velocity+! 10-21
!!wpn sep!! 10-33
!!wpn sep posn!! 10-33

4. Undesired Events

%%acm already init%% 10-9
%%acm chg method%% 10-9
%%acm src not init%% 10-9
%%acmslist not init%% 10-9
%%already init%% 10-21
%%chg method%% 10-21
%dir undef% 10-21
%%ecmlist not init%% 10-21
%%err acmslist%% 10-9
%%err wlist%% 10-33
%incomp motion% 10-25
%%incons ecmlist%% 10-21
%%itm in acmslist%% 10-9
%%itm in ecmlist%% 10-21
%%itm in wlist%% 10-33
%%itm not init%% 10-33
%not lnch weap% 10-33
%not rls or sk% 10-33
%not rls or wall% 10-33
%not rls weap% 10-33
%not shrike% 10-33
%posn unknown% 10-25
%%src not init%% 10-21
%too many targets% 10-25
%unknown target% 10-25
%%wbm already init%% 10-33
%%wbm chg method%% 10-33
%%wbm src not init%% 10-33
%%wlist in wlist%% 10-33

%%wlist member init%% 10-33
%%wlist not init%% 10-34
%wpn out of rng% 10-33

5. System Generation Parameters

#2D range limit# 10-21
#gr limit# 10-21
#max indic normal accel# 10-10
#max wind magnitude# 10-21
#num targets# 10-25
#std sl dens# 10-21
#std slp# 10-21
#symbol update period# 10-22

References

[AT] Clements, P., Faulk, S., and Parnas, D.; Interface Specifications for the SCR(A-7E) Applications Data Types Module; NRL Memorandum Report 8734

[DI] Parker, A., Heninger, K., Parnas, D., and Shore, J.; Abstract Interface Specifications for the A-7E Device Interface Module; NRL Memorandum Report 4385

[EC] Britton, K., Clements, P., Parnas, D., and Weiss, D.; Interface Specifications for the SCR(A-7E) Extended Computer Module; NRL Memorandum Report 4843

[GEO]
Bomford, Brigadier G., GEODESY, 2nd ed., Oxford, 1962.

[GLOSS]
OFP-2: *OFP NWC-2 Glossary of Terms for A-7E Navigation and Weapon Delivery OFP* (U), for Tape 7, revision B, 1 Feb 1977

[MRC]
Amlicke, B., Fox, J.; A-7 Flight Software Analysis; Mission Research Corporation report MRC/WDC-R-053; Sept. 1983

[NAV]
Crews, L., Hall, C; A-7D/E Aircraft Navigation Functions; NWC TN 404-176 March 1975

[NWC-2]
OFP NWC-2 Program Listing and Cross References (U), NWC-2 Tape 7 Revision C, 3 Oct 1977

[REQ]
Heninger, K., Kallender, J., Parnas, D., and Shore, J.; Software Requirements for the A-7E Aircraft; NRL Memorandum Report 3876; 27 November 1978.

[WPN]
A-7E Aircraft Weapon Delivery Equations (U), by R. George, NWC TM 2926, September 1976

Acknowledgements

Table of Contents

Chapter 1 PM.ACM: AIRCRAFT MOTION 10-1
Chapter 2 PM.ECM: EARTH CHARACTERISTICS MODULE 10-11
Chapter 3 PM.HFM: HUMAN FACTORS MODULE 10-22
Chapter 4 PM.TBM: TARGET BEHAVIOR MODULE 10-23
Chapter 5 PM.WBM: WEAPON BEHAVIOR MODULE 10-26
Appendix A DESIGN ISSUES 10-35
Appendix B IMPLEMENTATION NOTES 10-39
Appendix C ASSUMPTIONS LISTS 10-43
Appendix D UNIMPLEMENTED FEATURES 10-46