(8) Steps 72-73. The unit roll axis vector $\vec{\xi}$ is defined. The angular acceleration vector $\vec{\alpha}$ and a position vector $\vec{r}$ are set to zero by expressions (29) and (30).

(9) Steps 74-77. The length of the current integration interval is checked for validity. A zero integration interval will cause a return to the user subprogram (step 104). If the interval length is valid, the missile altitude in relation to the upper limit of the atmosphere is determined. If the missile is above the upper limit of the atmosphere, the subprogram continues at step 105; otherwise, the subprogram continues at step 78.

(10) Steps 78-79. With the missile below the upper limit of the atmosphere in powered flight or in the re-entry stage, the sonic velocity at the current missile position is computed by expression (31).

(11) Steps 80-89). The X, Y, and Z components of wind velocity at the present missile position are computed by expressions (32), (33), and (34). The X, Y, and Z components of missile velocity relative to the air at the current missile position are evaluated by expressions (35), (36), and (37). VECMAG computes the magnitude of the missile velocity vector. The missile velocity unit vector coordinates X, Y, and Z are determined by dividing the coordinates of the missile velocity relative to the air by the magnitude of the missile velocity vector.

(12) Steps 90-92. The mach number is computed by dividing the magnitude of the current missile velocity relative to the air by the speed of sound at the current missile location. The dynamic pressure on the missile is evaluated by expression (38). Also, the angle of attack is converted from radians to degrees and the product of this value and the dynamic pressure on the missile is computed.

(13) Steps 93-95. IFLAG is set to identification integer 317. If the missile is in the re-entry stage of flight, the subprogram continues at step 107; otherwise, the subprogram continues at step 96.

(14) Steps 96-100. INTERP computes the current axial drag coefficient C_A by table interpolation as functions of mach number. The X, Y, and Z coordinates for the axial component of acceleration due to aerodynamic forces are evalu - ted by expressions (39), (40), and (41).

(15) Steps 101-104. The total axial thrust vector is evaluated by expression (42). The components of missile acceleration are computed by use of expressions (43), (44), and (45). Missile velocity components are placed in the FDX register. CUTIE is stepped by one and control is returned to the user subprogram.

(16) Steps 105-106. With the missile in powered flight above the upper limit of the atmosphere, the missile weight is computed and INTERP interpolates for the current axial position of the missile center of gravity as a function of

missile weight. The subprogram continues at step 101.

(17) Steps 107-112. With the missile in the re-entry stage of flight, and if range safety data has been requested, the subprogram continues at step 122. Otherwise, IFLAG is set to identification integer 316. If booster shell impact point is not to be determined, expression (47) is evaluated and the subprogram continues at step 124. Otherwise, the subprogram continues at step 113.

(18) Steps 113-116. INTERP performs four successive interpolations to obtain variables used in expression (46). C_{Dref} is a function of the mach number and ΔC_D, C_D', C_D'' are functions of altitude.

(19) Steps 117-120. The sum of two variables ($C_{Dref} + \Delta C_D$) is computed as a component of expression (48). If the missile re-entry velocity or re-entry angle is equal to a predetermined reference velocity or angle, expression (46) is not computed since ($C_{Dref} + \Delta C_D$) becomes the drag force coefficient. The subprogram continues at step 121. If the missile re-entry speed or angle is not identical to the reference speed or angle, the subprogram continues at step 120, where the drag force coefficient $\bar{C}_D$ for the re-entry vehicle total dynamic force vector is computed by use of expression (46).

(20) Step 121. The magnitude of the re-entry vehicle total aerodynamic drag force divided by missile speed is computed by use of expression (47). The subprogram contin-

ues at step 124.

(21) Steps 122-124. INTERP derives the current axial aerodynamic force coefficient of the missile as a function of mach number by interpolation. The magnitude of the aerodynamic drag force divided by the missile speed is computed. The X, Y, and Z components of the resultant force vector as a product of the drag force divided by missile speed and the respective velocities along each axis are evaluated by expressions (48), (49), and (50).

(22) Steps 125-126. With the missile in the ballistic or re-entry stage of flight, missile acceleration is computed as the sum of gravitational acceleration and axial aerodynamic drag force. CUTIE is stepped by one and control is returned to the user subprogram.

d. Expressions.

$$\text{Pressure or Density} = \frac{a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4}{\left[1 + b_1 + b_2 x^2 + b_3 x^3 + b_4 x^4\right]^4} \quad (1)$$

$$\ddot{X} = -g_o \frac{x_m}{r_m} H \quad (2)$$

$$\ddot{Y} = -g_o \frac{y_m}{r_m} H \quad (3)$$

$$\ddot{Z} = -g_o \frac{z_m}{r_m} (H + L) \quad (4)$$

for SW(21) = ØN and SW(20) = ØN

$$H = \frac{r^2}{r_m^{\,2}} + \frac{Jr^4}{r_m^{\,4}}\left[1 - 5(\sin L_{CM})^2\right] + \frac{dr^6}{r_m^{\,6}}$$

$$\left[9(\sin L_{CM})^4 - 6(\sin L_{CM})^2 + \frac{3}{7}\right] \qquad (5)$$

$$L = \left[2J - \frac{dr^2}{7r_m^{\,2}} \; 28(\sin L_{CM})^2 - 12\,\frac{r^4}{r_m^{\,4}}\right] \qquad (6)$$

for SW(21) = ØN and SW(20) = ØFF

$$H = \frac{r^2}{r_m^{\,2}} + \frac{Jr^4}{r_m^{\,4}}\left[1 - 5(\sin L_{CM})^2\right] \qquad (7)$$

$$L = 2J\,\frac{r^4}{r_m^{\,4}} \qquad (8)$$

for SW(21) = ØFF and SW(20) = ØN

WWW.CHROMEHOOVES.NET

$$H = \frac{r^2}{r_m^{\,2}} + \frac{dr^6}{r_m^{\,6}}\left[9(\sin L_{CM})^4 - 6(\sin L_{CM})^2 + \frac{3}{7}\right] \qquad (9)$$

$$L = \left[-\frac{dr^2}{7r_m^{\,2}}\left[28(\sin L_{CM})^2 - 12\right]\frac{r^4}{r_m^{\,4}}\right] \qquad (10)$$

for SW(21) = ØFF and SW(20) = ØFF

$$H = \frac{r^2}{r_m^{\,2}} \qquad (11)$$

$$L = 0 \qquad (12)$$

$$F_T = F_{BSL} + K1\ (P_{SL} - P) \qquad (13)$$

$$F_T = F_{BSL} + K1\ (P_{SL} - P) + F_{VB} \qquad (14)$$

WWW.CHROMEHOOVES.NET

$$F_T = F_{VB} + F_{Tt_2}\left[1/(t_4 - t_2)\right]\int_{t_2}^{t_4} \sigma_B dt + F_{SR}\,(t_4 - t_3)/(t_4 - t_2) \qquad (15)$$

$$F_T = \left[F_{VB}\,(t_5 - t_4) + F_{VS}\,(t_7 - t_5) + F_{SR}\,(t_6 - t_4)\right]/(t_7 - t_4) \qquad (16)$$

$$F_T = \left[F_{SV} - K2P\right] + F_{VS} \qquad (17)$$

$$F_T = \left[F_{SV} - K2P\right] + F_{VB} \qquad (18)$$

$$F_T = F_{VB} \qquad (19)$$

$$F_T = 0 \qquad (20)$$

$$FDM = FWBG + FWBL + FWLB \qquad (21)$$

$$FDM = FWBG + FWBL + FPRGL + FPRGG \qquad (22)$$

$$FDM = \frac{FWBSD}{FT42} + FPRGL + FPRGG + \frac{(FWSRP)\,(FT43)}{(FT63)\,(FT42)} \qquad (23)$$

$$FDM = (FPRGL + FPRGG)\,FT54 + \frac{(FPSGL + FPSGG)\,FT75}{FT74} + \frac{(-FWSRP)\,(FT64)}{(FT63)\,(FT74)} - \frac{FWRC}{FT74} + \frac{FPRSL + FPRSG}{FT74} \qquad (24)$$

$$FDM = FPSGL + FPSGG + FPRFL + FPRFG \qquad (25)$$

$$FDM = FPRGL + FPRGG + (FPSSL + FPSSG)/(FT98) \qquad (26)$$

$$FDM = FPRGL + FPRGG \qquad (27)$$

$$FDM = 0 \qquad (28)$$

$$\vec{\alpha} = 0.0 \qquad (29)$$

$$\vec{r} = 0.0 \qquad (30)$$

$$c = \left[(1.4)(P)(144)\right]/D \qquad (31)$$

$$\dot{X}_W = V_{NW}\,X_m \sin L_{CM} + V_{EW}\,Y_m/r_{mxy} \qquad (32)$$

$$\dot{Y}_W = V_{NW}\, Y_m \sin L_{CM} - X_m\, V_{EW}/r_{mxy} \qquad (33)$$

$$\dot{Z}_W = V_{NW}\, r_{mxy}/r_m \qquad (34)$$

$$\dot{X}_{mr} = \dot{X}_m + \Omega\, \dot{Y}_m - \dot{X}_w \qquad (35)$$

$$\dot{Y}_{mr} = \dot{Y}_m - \Omega\, \dot{X}_m - \dot{Y}_w \qquad (36)$$

$$\dot{Z}_{mr} = \dot{Z}_m - \dot{Z}_w \qquad (37)$$

$$q = (1/2)\,(\rho\, S_M{}^2) \qquad (38)$$

$$D_{ax} = -\, qS/m\; C_D\, \xi_x \qquad (39)$$

$$D_{ay} = -\, qS/m\; C_D\, \xi_y \qquad (40)$$

$$D_{az} = -\, qS/m\; C_D\, \xi_z \qquad (41)$$

$$F_T' = F_T\, \xi \qquad (42)$$

$$\ddot{X} = G_x + DA_x + DN_x + (F_T')/M \qquad (43)$$

$$\ddot{Y} = G_y + DA_y + DN_y + (F_T')/M \qquad (44)$$

$$\ddot{Z} = G_z + DA_z + DN_z + (F_T')/M \qquad (45)$$

$$\overline{C}_D = C_{Dref} + C_D + \sum_{i=1}^{3} \sum_{j=1}^{3} \Big[(C_D'\, K_{ij} + C_D''\, \lambda_{ij})\, (S_{RE} - V_{ref})^{i-1}\, (\gamma_E - \gamma_{ref})^{j-1} \Big] \qquad (46)$$

$$F_{DA} = q\, \overline{C}_D\, \overline{A/W} \qquad (47)$$

$$\overline{F}_{DX} = (F_{DA}/V)\; V_x \qquad (48)$$

$$\overline{F}_{DY} = (F_{DA}/V)\; V_y \qquad (49)$$

$$\overline{F}_{DZ} = (F_{DA}/V)\; V_z \qquad (50)$$

2-243. SUBPROGRAM P24 (CLØØP). CLØØP performs closed loop flight simulation. The FORTRAN II reference statement is CALL CLØØP.

a. Inputs. The inputs are as follows:

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
FSPPS	2,3	Current missile position vector	$\vec{r}_m$	feet
FSPVL	2,3	Current missile velocity vector	$\vec{V}_M$	ft/sec
FTFSP	2	Current time of flight since liftoff	t_f	seconds
GØMGA	2	Rotation rate of earth	Ω	rad/sec
GAIR	2	Altitude above mean sea level of upper limit of atmosphere	$h_{a\ell}$	feet
GSYNC	2	Time interval between end of last FSIMLC simulation interval and end of current GCDSIM cycle		seconds
GAZEZ	2	Time interval after SECO when A_o and E_o are measured	t_{AoEo}	seconds
GDELT	2,3	Length of FSIMLC output intervals, powered, ballistic re-entry	t_{RS}	seconds
ILEVL	3	Number of output intervals per simulation interval, powered, ballistic, re-entry	I_o/sim	positive integer
TMFLT	13.4	Total time of flight vs earth-fixed range and launch azimuth from delta matrix	t_{f8}	seconds
GLXQA	21	Constraint - table of maximum angle of attack vs dynamic pressure		degs vs lbs/ft^2

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
GCØDT	2,2	Mean SECO and VECO transmission delay times (not including whoosh times)	t_{td}	seconds
PRDRV	2	Radar distance to center of earth	r_r	feet
PRCCL	2	Radar cosine of geocentric latitude	$\cos L_{CR}$	pure number
PRSCL	2	Radar sine of geocentric latitude	$\sin L_{CR}$	pure number
FPRFL	2,1	Stage II Sustainer LOX flow rate	$\dot{L}_S$	slugs/sec
FPRFG	2,1	Stage II Sustainer fuel flow rate	$\dot{F}_S$	slugs/sec
GLFVC	2	Constraint - Sustainer burning time margin	F_{VCON}	seconds
XS(6)	1	System constant	S_3	
XS(58)	1	System constant	S_{29}	

b. Outputs. The outputs include the outputs of all subprograms used by CLØØP and the following:

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
LEVEL	1	Number of output intervals per simulation interval	L_O/SIM	positive number
XLØM	2,7	Magnitude of steering commands		quanta
XLØT	2,7	Type of steering commands		positive integer
GALST	2	Last value of radar azimuth		radians
GADØT	4	Current estimates of radar antenna slew rate	$\dot{A}_{est}$	rad/sec
GADMX	2	Current maximum value of antenna slew rate	$\dot{A}_{MAX}$	rad/sec

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
GADXT	2	Time of flight from liftoff to GADMX	t_{Amax}	seconds
GLØØK	2	Sine of smallest too-small look angle after look angle has become at least 5 degrees	β_{Lmin}	pure number
GLØKT	2	Time of flight from liftoff to GLØØK	$t\beta_{Lmin}$	seconds
GXLPH	2	Current largest excessive angle of attack	θ_{max}	degrees
GXLPQ	2	Dynamic pressure at time of GXLPH	$q\theta_{max}$	lbs/ft^2
GXLPT	2	Time of flight from liftoff to GXLPH	$t\theta_{max}$	seconds
GAQMX	2	Maximum value of dynamic pressure q during flight prior to re-entry vehicle separation	Q_{maxNCS}	lbs/ft^2
GMXQT	2	Time of flight from liftoff to GAQMX	t_{qmax}	seconds
GAQST	2	Value of q at booster jettison point	Q_{BJT}	lbs/ft^2
GNQVA	2	Actual value of integral of qVa from liftoff to booster jettison		lbs/ft
FAZRØ	2	A_o at time GAZEZ after SECO	A_{oS}	radians
FEZRØ	2	E_o at time GAZEZ after sustainer cutoff	E_{oS}	radians
FSPEC	2	Special requested interrupt point time from liftoff	t_{IN}	seconds
GELVE	2	Elevation angle nearest VECO	E_{VECO}	radians
GAFVC	2	Stage II fuel remaining at vernier-commanded cutoff	F_{VECO}	slugs
GBJSM	2,15	Booster-jettison point summary data table		

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
GVCSM	2,15	Vernier-commanded cutoff-point summary data table		
GSCSM	2,15	Sustainer-commanded cutoff-point summary data table		
GNSSM	2,9	Re-entry vehicle separation-point summary data table		

The following switch settings also are outputs:

COMMON TAG	ITEM (switch in ØN state)
SW(3)	Booster propellant exhaustion to be monitored; if ØFF, stage II
SW(4)	Special interrupt time of flight simulation requested
SW(5)	Interrupt occurred based on altitude criterion
SW(6)	Interrupt occurred based on special time criterion
SW(7)	Interrupt occurred based on normal interrupt time criterion
SW(10)	RSDØRE to be initialized for data recording of new target
SW(11)	RSDØRE to end recording of offset data for current target
SW(12)	Stage II fuel or LOX exhaustion occurred before vernier cutoff
SW(22)	FSIMLC to be initialized for starting new flight simulation
SW(23)	RADSIM to be initialized for starting new flight simulation
SW(24)	Guided flight simulation in progress
SW(25)	FSIMLC to interrupt on propellant exhaustion

COMMON TAG	ITEM (switch in ØN state)
SW(26)	FSIMLC to interrupt on time of flight
SW(27)	FSIMLC to interrupt on altitude during ascent
SW(28)	FSIMLC to interrupt on altitude during descent
SW(29)	FSIMLC to interrupt on other end criteria
SW(30)	FSIMLC to complete previously interrupted simulation interval
SW(34)	WNDTRP to use launch area meteorological data; if ØFF, impact area
SW(36)	Booster stage simulation in progress
SW(37)	Sustainer stage simulation in progress
SW(38)	Vernier stage simulation in progress
SW(39)	Ballistic stage simulation in progress
SW(40)	Re-entry stage simulation in progress
SW(42)	Radar coast flag set
SW(50)	Antenna slew rate exceeded constraint during current flight
SW(62)	TEST to check only outputs of GGDSIM to CLØØP; if ØFF, check other outputs of GGDSIM
SW(65)	FXTIM was set for sustainer cutoff
SW(66)	FXTIM was set for vernier cutoff
SW(67)	Gimbal angle was excessive at least once during this flight

COMMON TAG	ITEM (switch in ØN state)
SW(68)	Angle of attack was excessive at least once during this flight
SW(134)	Look angle constraint was exceeded
SW(136)	Missile was in orbit after cutoff in TAA flight
SW(151)	Constraint was exceeded in last simulation of this target

c. Closed Loop Simulation Timing.

(1) To more fully understand the synchronization between the components of the targeting program, figure 2-6 is included in the description of CLØØP. This figure illustrates the timing during closed loop simulation and the relationship in time between the components of the guidance system as data is processed. The basic time is the length of a guidance computer cycle, 0.390412 second. To simplify the timing problem, the integration interval used by the flight simulator corresponds to a guidance computer cycle.

(2) During closed loop simulation, the relationship in time between the components of the guidance system is established at liftoff. The position, velocity, and attitude of the missile at liftoff establish the initial conditions for the equation of missile motion. The flight simulator integrates the equation of motion to obtain the position of the missile 0.292809 second after liftoff. All subsequent integration intervals correspond to the length of a guidance computer cycle. The missile position data is adjusted by

Changed 31 October 1962

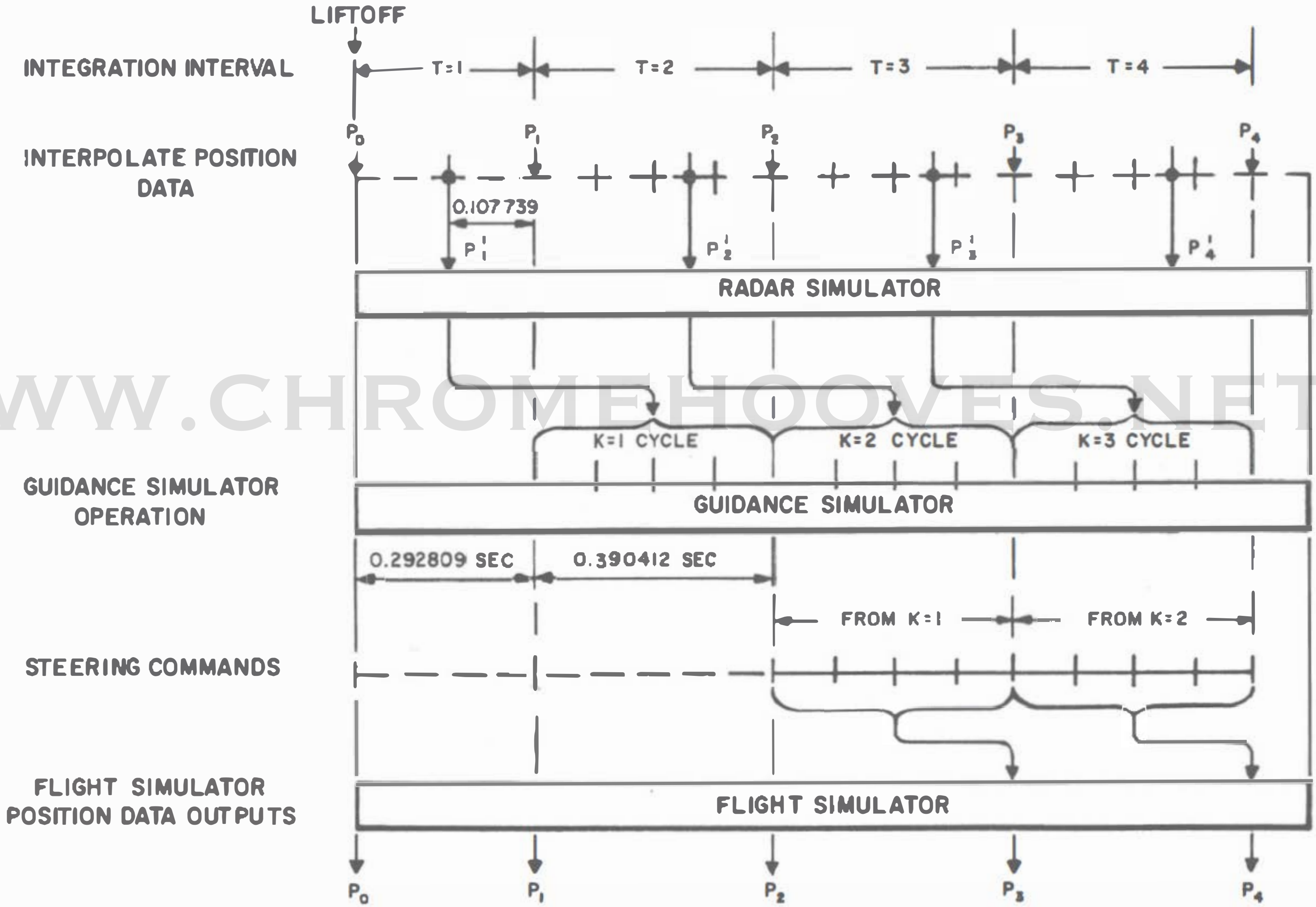

Figure 2-6. Closed Loop Simulation Timing

quadratic interpolation to be valid 0.107739 second before the start of a guidance computer cycle.

(3) The adjusted missile position data is converted to radar coordinates by the radar simulator. The radar data is used by the guidance simulator to determine steering orders and cutoff commands. Since the steering orders computed during the k^{th} cycle are not transmitted until the $k+1^{st}$ cycle, their use by the flight simulator is delayed for one cycle.

d. Program Logic. FD P24. CLØØP is the control subprogram of closed loop flight simulation. It is used by the OTC, TOT, TAA, and SIM functions. Linkage between subprograms of the A area with CLØØP is achieved by setting a dummy subprogram in the B1 area which contains the same number of transfer vectors as CLØØP. This dummy subprogram is referred to as dummy CLØØP and is not described in this volume.

(1) Steps 1-17. IFLAG is set to identification integer 1624. If SWAP is to call CLØØP, DP converts missile position from inertial coordinates to radar coordinates. Otherwise, the subprogram continues at step 125. Preliminary operations are performed before the first simulation interval. These include establishing missile parameters, setting internal controls, clearing time sequencing devices, and initializing switch settings. If RSDØRE is to record data from GGDSIM, initializations are performed, ØLGSIM computes pitch, yaw, and roll turning rates, S_{29} is set to

the powered flight integration interval, and the subprogram continues at step 18. Otherwise, GGDSIM performs setup phase computations for the guidance simulator.

(2) Steps 18-21. PRESET sets up FSIMLC for start of this flight. The length of the current integration interval is computed, the simulated radar data is set to the current missile position, and RSDØRE stores current trajectory data for range safety use.

(3) Steps 22-24. FSIMLC performs flight simulation for the first simulation interval and the necessary outputs are stored for use by the radar simulator. RADSIM performs radar simulation for this interval after the times have been computed at which the position data was valid.

(4) Steps 25-29. SINE, CØSINE, and VCDØTP are used to compute the cosine of the current look angle. If AAFLG is set ØN, the subprogram continues at step 30. If the vernier cutoff command has been reached, AAFLG is set ØN and the cosine of the current look angle is stored in the vernier-commanded cutoff point data table. The subprogram continues at step 30.

(5) Steps 30-38. IFLAG is set to identification integer 1624. If the current time of flight since liftoff is less than the total time of flight vs earth fixed range and launch azimuth extracted from the delta matrix, the subprogram continues at step 39. Otherwise, a test is performed

to determine if the magnitude of current total thrust is less than 1500 pounds. If so and A is less than 1, TAUT is set to the current time of flight plus 5 and A is set to 1. If not, the subprogram continues at step 39. If the current time of flight is less than TAUT, the subprogram continues at step 39. Otherwise, a test is performed to determine if the current look angle is less than five degrees. If so, SW(134) is set ØN to signify that the look angle constraint was exceeded and the look angle and time of occurrence are stored for future use. The subprogram continues at step 39.

(6) Steps 39-45. The current estimates of radar antenna slew rate are computed. If $\dot{A}_{EST} \cdot t_{RS}$ is greater than or equal to 6, the current estimates of radar antenna slew rate are re-computed. If the estimated value is the maximum rate ever encountered, it is stored. If the estimated value is greater than the constraint, SW(50) is set ØN. The subprogram continues at step 46.

(7) Steps 46-49. RSDØRE stores trajectory data for RSD. If RSDØRE is to record data from GGDSIM, ØLGSIM computes pitch, yaw, and roll turning rates and the subprogram continues at step 50. Otherwise, GGDSIM performs ground guidance simulation and the subprogram continues at step 50.

(8) Steps 50-56. If P is less than or equal to 5 and t is less than or equal to S_3, the roll steering commands, if they exist, for each of the four guidance intervals are stored and the subprogram continues at step 57. Otherwise,

pitch and yaw steering commands, if they exist, are stored. If both pitch and yaw steering commands exist for any of the four guidance intervals, an exit is made to ILLCMD to print a notification of an illegal command.

(9) Steps 57-72. If P is less than 15, the subprogram continues at step 73. If P is greater than 21, IFLAG is set to identification integer 1624 and a test is performed to determine if the next normal interrupt point is vernier cutoff. If not, and if counter v is less than S_{29}, the time from liftoff to vernier cutoff is computed and stored as the next normal interrupt point, SW(66) is set ØN and the subprogram continues at step 73. If the next normal interrupt point is vernier cutoff, or if counter v is equal to or greater than S_{29}, the subprogram continues at step 73. If $14 < P < 22$, IFLAG is set to identification integer 1624, and a test is performed to determine if the next normal interrupt point is sustainer cutoff. If not, and if counter v is less than S_{29}, the time from liftoff is computed and stored as the next normal interrupt point. A special interrupt point after sustainer cutoff is set up when A_o and E_o are measured, SW(65) and SW(4) are set ØN, and the subprogram continues at step 73. If the next normal interrupt point is sustainer cutoff, or if counter v is equal to or greater than S_{29}, the subprogram continues at step 73.

(10) Steps 73-79. FSIMLC performs flight simulation. If the missile is below the upper limit of the atmosphere

and the constraint on the angle of attack has been exceeded, SW(68) is set ØN. If the largest excessive angle of attack previously encountered has been exceeded, the current angle of attack, time of flight, and dynamic pressure are stored. IFLAG is set to identification integer 1624.

(11) Steps 80-94. If an interrupt occurred based on an altitude criterion, RSDØRE stores trajectory data at the interrupt point for RSD and the subprogram continues at step 95. Otherwise, a test is performed to determine if an interrupt occurred based on a special time criterion. If so, DP computes A_o and E_o, SW(4) is set ØFF, and the subprogram continues at step 124. If not, and if an interrupt occurred based on a normal interrupt time, the subprogram continues at step 94. If an interrupt did not occur based on a normal interrupt time, a test is made to determine if the sustainer burning time margin has been exceeded. Subsequent tests determine if stage II LOX or fuel have been exhausted. If either has been exhausted, SW(12) is set ØN and control is transferred to step 24. If neither have been exhausted, control is transferred directly to step 24.

(12) Steps 95-103. The current stage of flight is determined and switches are set to indicate if any change occurs. During sustainer, SW(37) is set ØN and during vernier, SW(38) is set ØN. At vernier cutoff, DP computes the elevation angle nearest vernier cutoff. At vernier thrust decay, SW(28) is set ØN. After appropriate switches have

been set, PRESET sets up flight simulation for the next interval.

(13) Steps 104-109. IFLAG is set to identification integer 1624. If an interrupt occurred based on a normal interrupt time, and if the missile is at stage II gas generator start, SW(26) is set ØFF and both SW(25) and SW(3) are set ØN. If not at stage II gas generator start, these switch settings are reversed. SW(26) ØFF indicates interruption is not to be based on time of flight. SW(25) ØN and SW(3) ØN indicate interruption is to be based on propellant exhaustion, and booster propellant exhaustion is to be monitored.

(14) Steps 110-114. If the missile is not prior to re-entry vehicle separation, SW(5), SW(6), and SW(7) are set ØFF and the dynamic pressure and its time of occurrence are stored. RANGLE computes the re-entry angle and speed of the missile relative to the surface of the rotating earth, the re-entry vehicle separation-point summary data table is set up, and the subprogram goes to step 129. If the missile is prior to re-entry vehicle separation, the subprogram continues at step 115.

(15) Steps 115-121. If the missile is at booster jettison, the booster-jettison point data table is completed, the dynamic pressure is stored, and the subprogram continues at step 124. If the missile is at sustainer thrust decay, the sustainer-cutoff summary data table is set up, and the

subprogram continues at step 122. If the missile is at vernier thrust decay, the vernier-commanded cutoff point data table is completed, the stage II fuel remaining is stored, SW(38) is set ØFF, SW(39) is set ØN, and the subprogram continues at step 124. Otherwise, the subprogram goes directly to step 124.

(16) Steps 122-124. If the minimum burning time is 125 or more seconds, SW(151) is set ØN, ILLCMD prints an error statement, and the subprogram continues at step 129. Otherwise, SW(5), SW(6), and SW(7) are set ØFF, SW(30) is set ØN, and the subprogram continues at step 73.

(17) Steps 125-129. IFLAG is set to identification integer 1624. If SWAP is to call ØLØØP, open loop flight simulation is performed by ØLØØP, CUTIE is stepped by one, and control is returned to the user subprogram.

2-246. SUBPROGRAM C70 (FLYER). FLYER maintains history values of position, velocity, mass, time, and their derivatives and selects the appropriate integrator. FLYER also monitors maximum dynamic pressure, and integrates and monitors the heat factor (FVA). The FORTRAN II reference statement is CALL FLYER.

a. Inputs. The inputs are register blocks FINT (0,1,2,3) and FDER (0,1,2) containing history values to be updated by FLYER. The following registers are also inputs:

COMMON TAG	DIMENSION	ITEM
LAG	1	Runge-Kutte step number
FDLT	2	Back up integrator interval of simulation interval
FDYNQ	2	Dynamic pressure
FAQMX	2	Maximum value of q during flight so far
FTFSP	2	Current total time of flight since liftoff
FVA	2	Magnitude of current missile velocity relative to local air mass
FQVAX	2	Maximum value of q (FVA) product so far (during re-entry)
FNQVA	2	Integral of q (FVA) product since liftoff
FALT	2	Altitude history value maintained by integrator
DALTP	2,2	Two previous altitude changes during integration
FMALT	2	Altitude of missile above ellipsoid

b. Outputs. The outputs consist of updated tables containing history values maintained by the integrator and computed values as follows:

COMMON TAG	DIMENSION	ITEM
FAQMX	2	Maximum value of dynamic pressure q during flight so far
FMXQT	2	Time of flight since liftoff to occurrence of FAQMX
QVA	2	Product of q and magnitude of current missile velocity relative to local air mass (FVA)
FQVAX	2	Maximum value of q (FVA) product during re-entry
GQVXT	2	Time of flight since liftoff to occurrence of FQVAX
FNQVA	2	Integral of qVA since liftoff
ALTPR	1	Altitude of missile at start of previous integration
DALTP	2,2	Previous and second previous altitude change during integration
FINT 3	2,4,4	History values maintained by integrator
FINT 2	2,4,4	
FINT 1	2,4,4	
FINT 0	2,4,4	
FDER 2	2,2,4	
FDER 1	2,2,4	
FDER 0	2,2,4	

c. Program Logic. FD C70

(1) Steps 1-3. FDLT is examined to determine whether back up integrator interval of interrupted simulation interval is negative or positive. If positive, control is transferred to step 16. If negative, Runge-Kutta step number LAG is set at five and Runge-Kutta integrator is selected to integrate for values of position, velocity, mass, and time. RK performs the Runge-Kutta integration. Control is transferred to step 24.

(2) Steps 4-15. ROUND rounds the double-precision missile position value to single precision, ~~and CUTIE is stepped by one~~. The current value of dynamic pressure q is compared with the previously recorded value of q. If greater, the current value and the corresponding time of flight measured from liftoff are stored before computing product of current dynamic pressure and current missile velocity magnitude relative to local air mass q (FVA). If the current value of q is not greater than the previously recorded value, no new values are stored and the product q (FVA) is computed directly. This value of q (FVA) is compared with the previously recorded value. If greater, the current value and the corresponding time of flight measured from liftoff are stored before computing the integral of q (FVA) with respect to time since liftoff. If the current value of q (FVA) is not greater than the previously recorded value, no new values are stored and the integral of q (FVA) with respect to time since liftoff is computed directly. The missile altitude

at the start of the previous integration is placed in ALTPR and registers containing previous and second previous altitude changes during integration are updated. Control is returned to the user subprogram.

(3) Steps 16-19. If the back up integrator interval of the interrupted simulation integral is not zero and if the Runge-Kutta step number is other than negative, control is transferred to step 19. If the back up integrator interval is not zero and if the Runge-Kutta step number is negative or, if the back up integrator interval is zero, IFLAG is set to identification integer 370. The subprogram exits to RLLBCK for return to the previous checkpoint.

(4) Steps 20-25. If the Runge-Kutta number is zero, PØLY integrates for position, velocity, mass, and time before updating history values. If the number is not zero, the Runge-Kutta step is reduced by one; RK integrates for position, velocity, mass, and time; and history values are updated. ~~RØLL updates the output interpolation table~~. Control is transferred to step 4.

2-247. SUBPROGRAM P08 (FSIMLC). FSIMLC simulates the flight of a missile during a given time interval. The FORTRAN II reference statement is CALL FSIMLC.

a. Inputs. The inputs are as follows:

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
FDELT	2	Length of output intervals	t_{Loi}	seconds
FLIX	2	Number of output intervals per integration interval		
FMALT	2	Current missile altitude above earth ellipsoid	h_m'	feet
FSPPS	2,3	Current missile position vector	$\vec{r}_m$	feet
FTFEB	2,1	Expected time of booster usable fuel exhaustion from liftoff	t_{BFE}	seconds
FTLEB	2,1	Expected time of booster usable LOX exhaustion from liftoff	t_{BLE}	seconds
FTFSP	2	Current time of flight since liftoff	t_f	seconds
FTFES	2,1	Expected time of stage II fuel exhaustion from liftoff	t_{FE}	seconds
FTLES	2,1	Expected time of stage II LOX exhaustion from liftoff	t_{LE}	seconds
FXTIM	2	Next normal sequencing interrupt time from liftoff	t_{NSIN}	seconds
FHUP	2	Flight interrupt specified by altitude during ascent	h_{up}	feet
FSPEC	2	Special requested interrupt point time from liftoff	t_{IN}	

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
FHDN	2	Flight interrupt specified by altitude during descent	h_{dn}	feet
FXX	80	Outputs interpolation table		
LIX	1	Number of output intervals per integration interval		
LEVEL	1	Number of output intervals per simulation interval	L_o/SIM	
DALTP	2,2	Previous and second previous altitude change during integration	$h_{\Delta 1}, h_{\Delta 2}$	

The following switch settings also are inputs:

COMMON TAG	ITEM (switch in ØN state)
SW(3)	Booster propellant exhaustion is to be monitored. If ØFF, stage II propellant exhaustion is to be monitored
SW(4)	A special interrupt time of flight simulation is requested in FSPEC
SW(22)	FSIMLC is to be initialized for starting a new flight simulation
SW(24)	Guided flight simulation (powered phase) is in progress
SW(25)	FSIMLC is directed to interrupt on propellant exhaustion
SW(26)	FSIMLC is directed to interrupt on time of flight
SW(27)	FSIMLC is directed to interrupt on altitude during ascent
SW(28)	FSIMLC is directed to interrupt on altitude during descent

COMMON TAG	ITEM (switch in ØN state)
SW(29)	FSIMLC is directed to interrupt on other end criteria
SW(30)	FSIMLC is directed to complete a previously interrupted simulation interval

b. Outputs. The outputs are as follows:

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
FDLT	2	Length of current integration interval	t_{CIi}	seconds
FDEL	2	Remaining time to integrate up to interrupt point	t_{Irem}	seconds
FALT	2	Missile altitude above ellipsoid at beginning of current integration interval	h'_{MI}	feet
FSPP	2,3	Interrupt point missile position vector	$\vec{r}''_m$	feet
FSPP(1-4)	2,3	Missile position vector components at end of four output intervals	$\vec{r}_{m1-4}$	feet
FSPV(1-4)	2,3	Missile velocity vector components at end of the four output intervals	V_{M1-4}	ft/sec

The following switch settings also are outputs:

COMMON TAG	ITEM (switch in ØN state)
SW(5)	Interrupt based on an altitude criteria
SW(6)	Interrupt based on a special interrupt criteria
SW(7)	Interrupt based on a normal sequence time criteria

c. Program Logic. FD PC8

(1) Steps 1-8. IFLAG is set to identification integer 1608. INTRØG interrogates SW(30) to determine if a previously interrogated simulation interval is to be completed. If ØN, SW(30) is set ØFF, the length of the current integration interval is negated as a control for FLYER. and the subprogram continues at step 9. If ØFF. MATTIT computes missile attitude vectors for one simulation interval. An indicator is set to LINT to mark the first interval and the length of the current integration interval is computed.

(2) Steps 9-19. The missile altitude at the beginning of the current integration interval is set to the current altitude. FLYER maintains the table of history values. Missile position is computed and IFLAG is set to identification integer 1608. INTRØG interrogates SW(29) to determine if simulation is to be interrupted. If ØFF, the subprogram continues at step 20. If ØN. simulation is not to be interrupted on other end criteria and SW(29) is set ØFF. Values of usable booster propellants and expected time of exhaust are determined if the missile is prior to re-entry vehicle separation. Time remaining to interrupt point is set to zero and current flight information is stored as interrupt point data. Control is returned to the user subprogram.

(3) Steps 20-40. If the missile is prior to re-entry vehicle separation. EXAUST computes the values of usable

propellants and expected time of exhaustion; otherwise, EXAUST is skipped and work area GRASE is cleared for use. The switches which determine the reason for interrupt point are interrogated. If simulation is to be interrupted at propellant exhaustion (SW(25) = ØN), at time of flight SW(26) = ØN), at altitude during ascent (SW(27) = ØN), and/or at special time criteria (SW(4) = ØN), these times are computed. If simulation is not to be interrupted during descent (SW(28) = ØFF), the subprogram continues at step 47; otherwise, the subprogram continues at the next step.

(4) Steps 41-46. If simulation is to be interrupted during descent, but the missile is not descending, the subprogram continues at step 56. If the missile is below altitude of descent interrupt, expression (1) is computed. If expression (1) is greater than 10, expressions (2), (3), and (4) are computed. If less than or equal to 10, the time remaining to interrupt point based on altitude descent is computed.

(5) Steps 47-55. The time remaining to the next interrupt point is set to the smallest of the times previously computed. If the minimum time is not negative, the subprogram continues at step 56. If negative, the length of the current integration interval is set to this value. SW(7) is set ØN if interrupt is due to propellant exhaustion or time of flight criteria; SW(6) is set ØN if interrupt is due to special interrupt time; SW(5) is set ØN if interrupt