DRSAR/SA/N - 58

DESCRIPTION OF A COMPUTER PROGRAM (ZOT. 14) FOR GUIDANCE SIMULATION OF CANNON-LAUNCHED GUIDED PROJECTILES

RICHARD D. HEIDER

JANUARY 1977

US ARMY ARMAMENT COMMAND
SYSTEMS ANALYSIS DIRECTORATE
ROCK ISLAND, ILLINOIS 61201

DISPOSITION

Destroy this report when no longer needed. Do not return it to the originator.

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position.

WARNING

Information and data contained in this document are based on input available at the time of preparation. Because the results may be subject to change, this document should not be construed to represent the official position of the US Army Development & Readiness Command unless so stated.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DRSAR/SA/N-58	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DESCRIPTION OF A COMPUTER PROGRAM (ZOT.14) FOR GUIDANCE SIMULATION OF CANNON-LAUNCHED GUIDED PROJECTILES		5. TYPE OF REPORT & PERIOD COVERED Note - Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard D. Heider		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Armament Command Systems Analysis Directorate (DRSAR-SA) Rock Island, IL 61201		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Armament Command Systems Analysis Directorate (DRSAR-SA) Rock Island, IL 61201		12. REPORT DATE January 1977
		13. NUMBER OF PAGES 86
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

CLGP (Cannon-Launched Guided Projectile)
Guided Projectile
Guidance Simulation
Guidance Accuracy
Footprint
Performance Envelope

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The methodology of the ZOT.14 Guidance Simulation Model is discussed. The model is a five-degree-of-freedom (5DOF) digital simulation program in FORTRAN IV which is designed for studies of guidance accuracy and performance envelopes of various cannon-launched guided projectiles (CLGPs) under a wide variety of situations. It is designed to facilitate large-sample-size Monte-Carlo experiments at significantly less cost in time and money than is possible when using existing 6DOF models.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT)

I. PURPOSE

This note describes the methodology of the ZOT.14 program for the simulation of the post-enablement phase of flight of a cannon-launched guided projectile (CLGP). The mathematical models employed and their implementation in the program are discussed. A user's guide is appended (Appendix A).

II. GENERAL DESCRIPTION

The guidance simulation model ZOT.14 is a general-purpose, quick-running, continuous-time simulation model for the post-enablement phase of the flight of a guided projectile with proportional navigation. It incorporates a non-rolling-projectile flight model (which accounts in large part for its speed) with simplified flight equations, a model for the random motion of the designator spot about the target, a model for the evasive maneuver of the target, and a model for acquisition/loss of acquisition of the target.

The program is written in FORTRAN G for the IBM 360 computer.

III. APPLICATIONS

The following applications of this program have been utilized:

a. Projectile response and trajectory - the state of the projectile as a function of time for a given situation may be obtained.

b. Guidance accuracy - a sample distribution of impact locations about the target may be generated.

c. Footprint analysis - the locus of feasible target locations for a given combination of zone, QE, enablement time, and environmental variables may be obtained.

d. Parametric studies - the effect of varying system, environmental, or scenario parameters upon any of the above may be studied. Likewise, different systems may be compared under similar conditions.

IV. SCENARIO DESCRIPTION

The basic scenario consists of a flat ground plane upon which a target is located, a projectile which flies from its enablement point onward, presumably to acquire and home into the target, a laser designator, and an environment.

A. Initial Conditions. The simulation begins with the projectile in an "acquisition-enabled" state. It has not yet acquired the target but is capable of doing so as soon as a sufficiently strong signature is received. The positions and velocities of both projectile and target are specified; the attitude of the projectile is also specified.

B. Environment. The target is located upon a flat ground plane having a specified altitude with respect to sea level. Other variables characterizing ambient conditions include a cloud ceiling, visibility range, and wind vector.

C. Sequence of Events. The normal sequence of processes modeled is: search, acquisition, null, guidance, and impact. Loss of acquisition may occur at any time if weak or missing pulses occur or if the target leaves the field of view. Reacquisition may commence upon subsequent reappearance of the signature. Also, the projectile may be programmed to transition from the laser-designated mode to a passive infrared-tracking mode.

D. Noise Sources. The types of noise modeled are spot motion and pulse dropout, pitch-yaw gimbal cross-coupling, gyro drift, control surface misalignment, and IR detector jitter. The principal contributor to dispersion of impacts in the semi-active laser (SAL) guidance mode is spot motion. Gyro drift can be an important source of unguided delivery error for gliding projectiles. It is a source of a false steering signal during guidance for gliding or non-gliding projectiles. Control surface misalignment is modeled as another source of a false steering signal during guidance. The IR jitter is a white (uncorrelated) angular error internal to the detector used in the bifunctional passive IR (BF) mode and is the principal contributor to dispersion of impacts in the BF mode of guidance.

V. FLIGHT EQUATIONS

A. Inertial. Displacement of the projectile center of mass is governed by the vector equation

$$\vec{F} = m\,\vec{a}$$

where $\vec{F}$ is the vector sum of aerodynamic, gravitational, and rocket motor forces. The optional rocket booster motor is characterized by a rectangular thrust-time curve; that is, the thrust is either the nominal specified value or it is zero, depending upon the time. Likewise, the fuel mass decreases at a constant rate during burn.

Aerodynamic pitching and yawing of the projectile are approximated by a second-order differential equation

$$\tau_\alpha^2(\ddot{\alpha} + \ddot{\theta}) + 2\,\zeta_\alpha\tau_\alpha\,\dot{\alpha} + \alpha = K_\alpha\delta$$

where

α is the angle of attack (in yaw or pitch),

δ is the corresponding control surface deflection,

θ is the attitude of the velocity vector,

τ_α is a "time constant" varying with the projectile's aerodynamic state and depending on its design,

K_α is a gain also dependent on projectile state and design,

ζ_α is the natural pitch-rate damping coefficient of the projectile, a constant.

The implementation of these equations is by simple rectangular integration using a time step significantly smaller than the physical time constant involved, τ_α, which is normally of the order of a tenth of a second. (The time scales of translational accelerations and velocity-orientational accelerations are normally much larger.) The trajectories obtained using this simple scheme have compared favorably with those of other methods, including 4th-order Runge-Kutta. An earlier version of this program was successfully used to reproduce an actual guided trajectory (MGP-7) within the precision of the test data.

The computational scheme for the above equations may be described as follows: If we let the subscript j denote the value of a variable at time step j, and k denote the value at the succeeding time step k = j+1, then

$$x_k = x_j + h\, v_{xj},$$

$$v_{xk} = v_{xj} + h\, a_{xj},$$

where

$$a_{xj} = F_{xj}/m,$$

for translational component x, and

$$\alpha_k = \alpha_j + h\, \dot{\alpha}_j,$$

$$\dot{\alpha}_k = \dot{\alpha}_j + h[K_\alpha \delta_j/\tau_\alpha^2 - (\alpha_j/\tau_\alpha + 2\, \zeta_\alpha \dot{\alpha}_j)/\tau_\alpha]$$

$$- (\dot{\theta}_k - \dot{\theta}_j) + h\, F_r \ell/I_B$$

for yaw or pitch angle of attack. The last term above represents an angular acceleration due to the malthrust moment of a booster rocket motor. The constant h is the integration time step.

The sign conventions used in these equations are: positive pitch angle of attack is nose upward; positive yaw angle of attack is nose left;

positive control surface deflection is in the sense required to produce a positive angle of attack.

The projectile's roll attitude is fixed. For simplicity, the roll attitude is such that pitch and yaw are vertical and horizontal, respectively.

B. <u>Aerodynamic</u>. The aerodynamic forces are obtained using aero coefficient tables. Input to the tables are the mach number, angles of attack, and control deflections. The tables are interpolated linearly.

The normal force coefficients are computed as

$$C_{NY} = C_{N\alpha}(M,\alpha')\cdot\alpha_y + C_{N\delta}(M)\cdot\delta_y$$

$$C_{NP} = C_{N\alpha}(M;\alpha')\cdot\alpha_p + C_{N\delta}(M)\cdot\delta_p$$

where α' is the net angle of attack and M is the mach number.

The axial force coefficient is computed as one of the following, depending on the format of the tables provided by the developer:

$$C_A = C_{AO}(M) + C_{A\alpha}(M) \cdot \alpha'^2,$$

$$C_A = C_{AO}(M) + C_{A\delta}(M) \cdot (|\delta_y| + |\delta_p|),$$

$$C_A = C_{AO}(M) + C_1(M) \cdot [0.4\alpha_y - \delta_y)^2 + (.04\alpha_p - \delta_p)^2]$$

$$- C_2(M) \cdot [|\alpha_y|^3 + |\alpha_p|^3].$$

The speed of sound a and air density ρ are obtained from a subroutine which uses piecewise curve-fits with altitude as the independent variable. Then the projectile's mach number M is related to the air speed v as $M=v/a$.

The body forces are first computed as

$$F_{NY} = C_{NY} \cdot Aq \;,$$

$$F_{NP} = C_{NP} \cdot Aq \;,$$

$$F_A = C_A \cdot Aq \;,$$

where the dynamic pressure q is

$$q = \frac{1}{2}\rho v^2,$$

A is the projectile cross section, and v is the air speed. These forces are then resolved by rotation into the x,y,z frame of reference, after which the gravitational and motor components are added.

The tables also give $K_\alpha(M,\delta)$ for yaw and pitch and the static margin in calibers $x_{sm}(M)$, which are used to compute τ_α for yaw and pitch as

$$\tau_\alpha = \sqrt{\frac{8I_B}{\pi D^3[(C_{N\alpha} + C_{N\delta}/K_\alpha)\rho v^2 x_{sm}]}} ,$$

where I_B is the transverse moment of inertia and D is the reference diameter of the projectile.

All of the computations discussed in this section are repeated every time step.

VI. GUIDANCE AND CONTROL

A. Acquisition. Two acquisition models are incorporated, a deterministic acquisition range model and an energy-flux density model.

The first is used to determine when acquisition first occurs when either (1) the internal mode or (2) the mixed (external position, internal intensity) mode of spot motion is selected. The second model is used after entering within acquisition range in either of these modes. It is also used exclusively for (3) the external spot motion mode.

The acquisition range model is an equation giving the range at which the flux density equals the acquisition threshold for specific values of a set of variables. One of these variables, the angle between the line of sight and the normal to the designated surface, varies continually with each appearance of a spot; thus, the acquisition range is recomputed for every spot until it exceeds the actual slant range.

The flux density model is a more general model relating the flux density at the seeker to another set of factors. One of these, the scattering cross section, is either a constant defined at acquisition in spot motion modes (1) and (2), or a pulse-to-pulse variable obtained from the input cards in mode (3). The flux-density model requires the flux density to be both greater than an absolute threshold and not less than a specified fraction of the previous flux density to be visible.

Acquisition is established upon the reception of a given number of consecutive pulses. Loss of acquisition occurs if a given number of consecutive pulses go unseen. Figure 1 shows the logical sequence of acquisition and loss of acquisition.

B. Detector Characteristics. The detector thresholds, absolute and

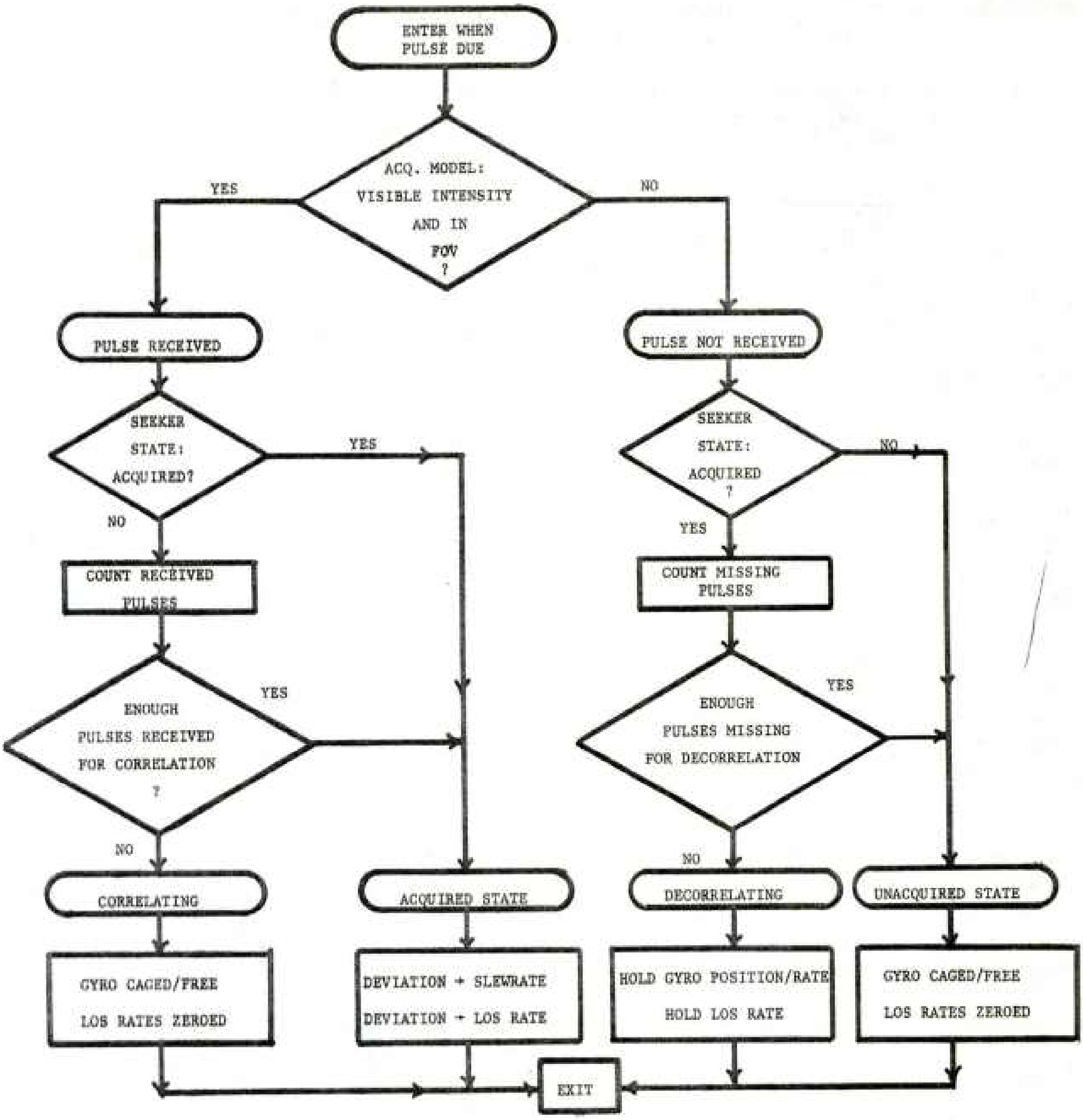


Figure 1. Flow Chart For Acquisition/Loss of Acquisition

relative, have already been mentioned. For a pulse to be received, it must meet those criteria and fall within the seeker's instantaneous field of view (FOV), which is a circular field. The position and intensity of a received pulse determine, by means of a table of transfer characteristics, the commanded gyro slew rate and the estimated line-of-sight (LOS) rate. Specifically, the detector characteristic is conceived in this program as an input-output relationship between two continuous-variable inputs and two outputs. The inputs are: (1) the angular difference in degrees in pitch (or yaw) between the LOS to the laser spot and the center of the FOV, and (2) the signal level of the received pulse in db above the threshold level. The outputs are: (1) the estimated LOS rate in degrees per second, which is input to the autopilot, and (2) the commanded gyro torquing rate in degrees per seond, which is input to the gyro torquing circuit. Moreover, the detector characteristic may depend upon two logical conditions: (1) whether the detector/autopilot is in the nulling or guiding mode, and (2) whether the detector is in the SAL or BF mode.

These transfer characteristics are entered as input data and are interpolated linearly with respect to deviation (apparent angular position of the spot on the seeker face) and signal level (db of energy flux density above threshold, 10 db = 1 decade). The interpolation scheme permits the user to "run out of the table" or extrapolate beyond the ranges of the tabulated independent variables--it simply assumes the value appropriate to the nearest tabulated value of the exceeded variable. (This "plateau" extrapolation scheme, incidentally, is also used in the aerodynamic tables.)

These tables are assumed to be (1) antisymmetric with respect to null (center of FOV) and (2) identical in yaw and pitch channels. Figure 2 shows a set of hypothetical transfer characteristics, and Table 1 shows the form in which these curves would be entered into the program. (This example does not represent any actual design.) The arrays are variable-dimensioned according to array sizes input. In addition, a second slew rate array is provided for the case that torquing behavior is different in the gyro-nulling phase (the period between acquisition and commencement of guidance) than in the guidance phase. If the BF detector has different characteristics than the SAL detector, another complete set of arrays (using the common set of deviations) is provided.

All abscissas for which there is a break point in any of the curves (Fig. 2) must be entered, along with the corresponding ordinates for every curve. Note that steps in ordinate are permitted by entering the abscissa twice, with the ordinates in the order of occurrence. Also note that the deviation and signal level 0 must be entered explicitly.

To the gyro slew rate thus defined by deviation and signal level are added the torquer-loop g-bias slew rate and the gyro drift rate. Torquer bias acts only during guidance. Drift acts when the gyro is in either the track or free mode; in the electrical-cage mode its only effect is to bias the gyro slightly off the caged position.

C. <u>Autopilot</u>. The ZOT.14 autopilot can be used to simulate all the

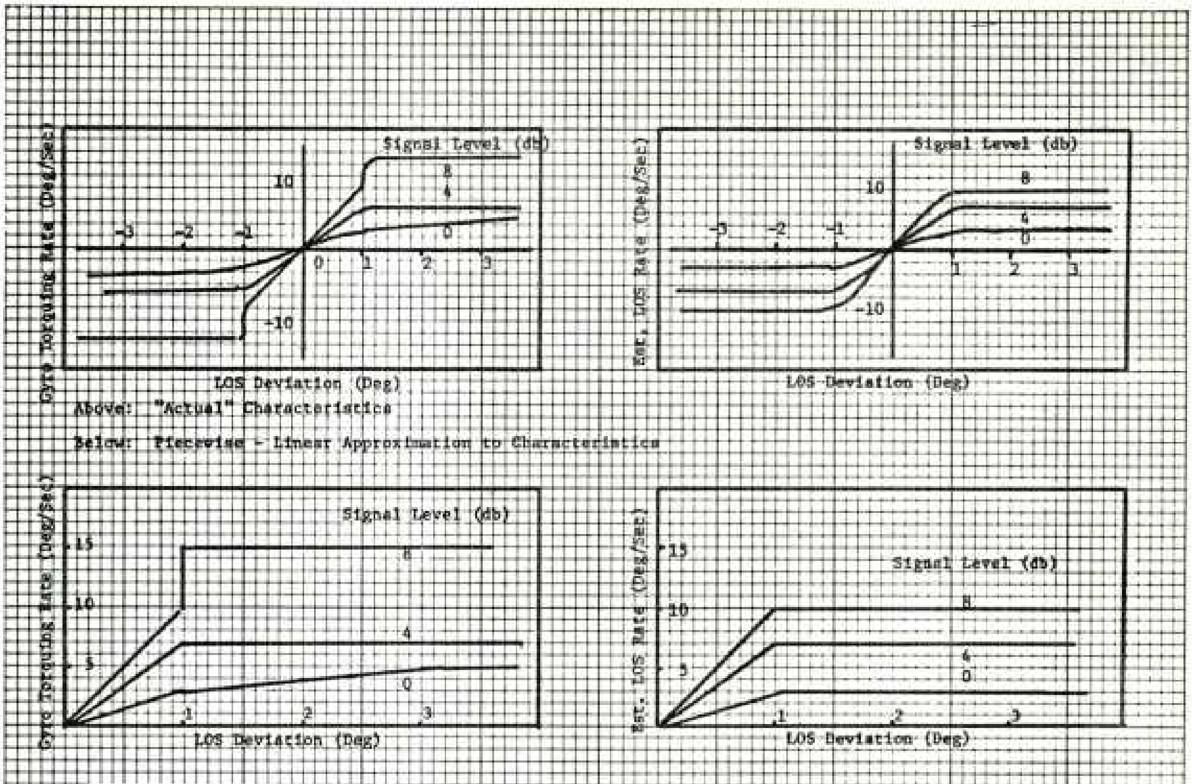


Figure 2. Hypothetical Detector Characteristic (Graphic Form)

TABLE 1. HYPOTHETICAL DETECTOR CHARACTERISTIC
(TABULAR FORM)

SIGNAL LEVEL (DB)	DEVIATION (Deg)			
	0	1	1	3
	SLEW RATE (Deg/Sec)			
0	0	3	3	5
4	0	7	7	7
8	0	10	15	15
	ESTIMATED LOS RATE			
0	0	3	3	3
4	0	7	7	7
8	0	10	10	10

non-gliding systems thus far encountered and the Martin ED gliding system, which keeps the body axis aligned with the "free" gyro axis. The pitch channel of the basic autopilot in the guidance phase is represented by Figure 3.

The input $\dot{\lambda}_p$ in Fig. 3, the estimated LOS rate, is the result of the seeker transfer outputs cross-coupled in pitch and yaw as follows:

$$\dot{\lambda}_p = \dot{\lambda}_p' + r\left|\frac{\Gamma_p}{\Gamma_{p(max)}}\right|\dot{\lambda}_y' \quad ,$$

where the primes indicate seeker transfer outputs, Γ_p is the pitch gimbal angle, and r is the cross-coupling coefficient.

The input $\dot{\Gamma}$ is obtained by differencing Γ at successive time steps, and Γ by differencing body-axis and gyro-axis orientation angles.

The first-order differential equation in the proportional-navigation channel (the upper channel in Fig. 3) is rectangularly integrated, as were the flight equations:

$$\delta_{PNk} = \delta_{PNj} + h\ (K_\delta\dot{\lambda} + \delta_B - \delta_{PNj})/\tau_\delta \quad ,$$

where δ_{PN} is the component of deflection due to the proportional navigation channel of the autopilot, and δ_B is the g-bias deflection.

The other differential equation, however, typically has a time constant τ_Q of about the same order as the time step used. Rather than decrease the time step by an order of magnitude or more, we have assumed that the input $\dot{\Gamma}$ is constant (with the value calculated above) during the present time step and solve the differential equation exactly:

$$\delta_{SDk} = \dot{\Gamma}[K_Q(1 - e^{-h/\tau_Q})] + \delta_{SDj}[e^{-h/\tau_Q}],$$

where δ_{SD} is the component of deflection due to the synthetic pitch-rate damping channel, and the expressions in brackets are precomputed constants.

Note in Figure 3 that the resulting deflections are limited in two places.

Different projectiles have not only different parameter values, but also different sequences of events in determining the bias δ_B and in freeing, caging, and tracking the gyro. These differences are discussed in Section X.

VII. TARGET MOTION

A. Target Model. The target is modeled as two distinct geometric

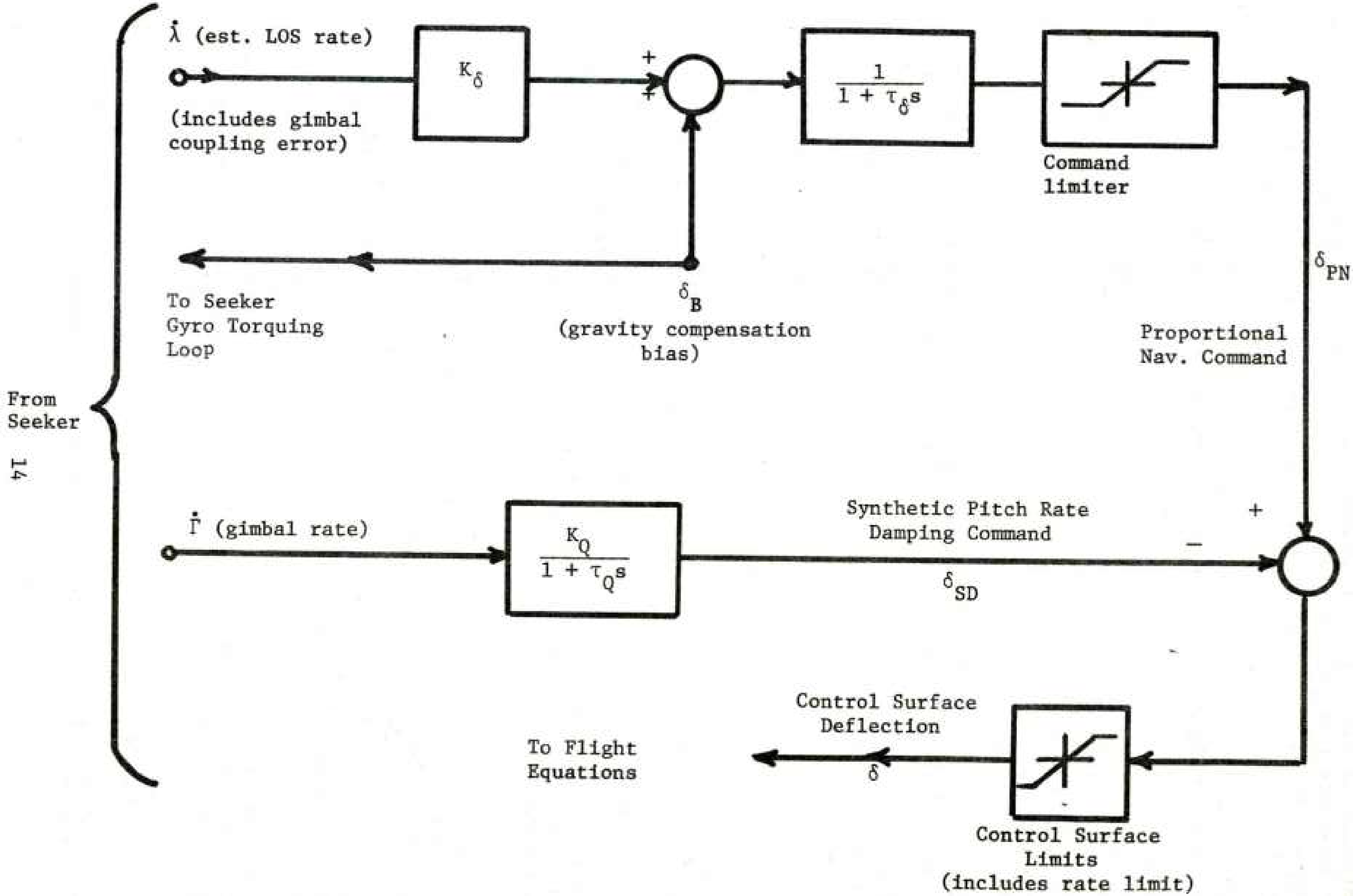


Figure 3. Basic ZOT.14 Autopilot In Guiding Mode

points:

(1) The "spot", which has a position determined jointly by a deterministic target maneuver model and a stochastic spot motion model; and

(2) The "target" or intended impact point, whose position is determined solely by the target maneuver model.

Note that while target (2) is the reference point for miss distances, target (1) is what the projectile homes on in the laser-guided mode. (There is also a bifunctional mode, in which the projectile may switch from tracking target (1) to target (2).)

B. Maneuver Model. This model provides the capability of four types of motion: (1) uniform motion; (2) uniform acceleration parallel to heading; (3) uniform turns (constant speed and radius); and (4) sinusoidal zigzags. Maneuvers may be programmed in any desired sequence, with the limitation that the total number of maneuvers of types 2, 3, and 4 for both targets together must not exceed ten. The two targets are programmed independently.

C. Spot Motion Model. The random motion of the spot about the nominal designation point is simulated by either of two means; in either case, it is necessary to do some preprocessing. The ERIM target reflectivity model, which utilizes a digital-Butterworth-filtered white noise source input to obtain a record of designator angular errors, a faceted target model, and an assumed designator-target-seeker geometry, produces a record of apparent spot positions in 3-dimensional space. This record may be used to drive ZOT.14 directly (external spot motion modes 2 and 3) or its autospectrum and spatial covariance matrix may be transmitted to ZOT.14, which creates a time series of spot positions having those properties (internal spot motion mode 1).

The internal mode operates in three logical stages: (1) generation of three independent gaussian white-noise sources by sampling standard normal deviates; (2) recreating the autospectrum by passing the white noise components through a digital Butterworth filter; and (3) recreating the covariance matrix by performing a linear transformation upon the independent filtered noise components. This mode has the advantage of producing an indefinitely long record using a small amount of input.

D. Pulse Dropout. Missing pulses may occur in the real world from several causes, such as drifting to a point hidden from the seeker's point of view (masking) or out of the seeker's field of view or to an area of very low reflectivity. When a missing pulse occurs in the ERIM model, zero-scattering cross section and zero position are output for that pulse; the ZOT.14 external spot motion mode interprets the zero cross section correctly as a missing pulse and the seeker model responds appropriately according to the seeker design specifications. In the internal mode, ZOT.14 provides an alternating stochastic renewal process of "see" and "dropped" time intervals.

E. Random Number Blocking. It may be desirable to reproduce identical spot position, pulse dropout, and IR jitter sequences for corresponding replications of several experiments. This requires that the same initial seed values be used in each experiment and that corresponding replications "use" the same number of laser and IR spots and pulse dropouts. This second requirement is met by requiring all replications to "use" a fixed "block" of random numbers regardless of the amount actually required within any one replication.

Block lengths for spot motion, pulse dropout, and IR jitter are named LENGTH, LENGT2, and LENGT3, respectively. These are input parameters and must be either all zero or all positive. If positive, they are the allocated block sizes; if these lengths are exceeded in any replication, the program prints a notification and aborts. If zero, they are internally defined at the successful completion of the first replication, allowing for two seconds of spot motion and IR jitter more than were actually used in that replication and ten cycles of pulse dropout more than were used. In practice, the user would make single-replication reference runs for each condition of an experiment expected to affect block lengths. The user would permit the program to define block lengths for each of these runs and would then use the maximum of each length in his experiment.

This routine operates in all spot motion modes. However, pulse dropout blocking operates only in internal mode (1) even though LENGT2 is defined for all modes.

VIII. FLIGHT TERMINATION

A. Decision Criteria. Termination of a replication occurs upon any of the following events:

(1) Excessive simulated time elapses. The time limit is 1.5 times initial slant range divided by initial projectile speed. The run is aborted.

(2) A random number block length is exceeded. The run is aborted.

(3) The projectile achieves closest approach to target (2). This point is determined by checking the sign of the dot product of the displacement vector from target to projectile with the projectile velocity vector. If the product is negative, the projectile is approaching the target; if positive, it is receding. The run is aborted if, at closest approach, the seeker has never acquired the target; otherwise, the run continues with impact scoring and the next replication.

(4) The projectile impacts the ground. Program action as in (3) above.

B. Scoring. When a replication terminates by criterion (3) or (4) and acquisition had occurred, the impact is scored; that is, the position of the point of closest approach is computed relative to target (2). This position is given both in the x, y, z simulation coordinates, and in the "impact plane," a hypothetical plane having origin at target (2), oriented normal to the relative velocity vector, and having yaw (horizontal) and pitch (vertical) coordinates.

The computations proceed as follows: let

x_r, y_r, z_r be the relative location of the projectile at the time step preceding the impact decision,

v_{xr}, v_{yr}, v_{zr} be the relative velocity at that time,

v_h be the horizontal resultant of that velocity,

v_p be the magnitude of that velocity,

x_{ca}, y_{ca}, z_{ca} be the relative point of closest approach,

r be the miss distance,

y be the yaw component of miss distance, and

p be the pitch component.

Then

$$x_{ca} = \frac{x_r(v_{yr}^2 + v_{zr}^2) - v_{xr}(v_{yr}y_r + v_{zr}z_r)}{v_p^2}$$

and similarly for y_{ca}, z_{ca} by cyclic permutation of coordinates. Then,

$$r = \sqrt{x_{ca}^2 + y_{ca}^2 + z_{ca}^2} ,$$

$$y = \frac{v_{xr}\, z_{ca} - v_{zr}\, x_{ca}}{v_h} , \text{ and}$$

$$p = \frac{-v_{xr}\, v_{yr}\, x_{ca} + v_h^2\, y_{ca} - v_{yr}\, v_{zr}\, z_{ca}}{v_h\, v_p} ,$$

given v_h and $v_p \neq 0$.

IX. REINITIALIZATION

After scoring, the next replication begins with reinitialization of the simulation. The unused portions of the previous replication's random number blocks are generated and discarded. Then, the projectile and target status are recalled for the reinitialization point, which is one of two points: (1) if the target location does not vary between replications and gyro drift is deterministic, the reinitialization point is the point at which acquisition was first established; otherwise (2) the reinitialization point is the original initialization point. When reinitialization is at point (1) above, sufficient spot motion is run to align the spot motion block with the simulation time of reinitialization. The pulse dropout block is not advanced, because the pulse dropout routine functions only after acquisition has first been established.

X. PROJECTILE MODELS

The various projectile designs accommodated by this program differ in the following qualitative respects:

(1) trajectory -- ballistic or gliding.

(2) gravity bias -- none, fixed, or adaptive. Bias may be applied through the proportional nagivation circuit alone or through the gyro torquing circuit as well.

(3) guidance enablement strategy -- this is the behavior of the projectile between acquisition and the commencement of proportional navigation guidance.

(4) rocket booster -- present or absent.

(5) bifunctional tracking -- present or absent.

(6) behavior upon loss of acquisition.

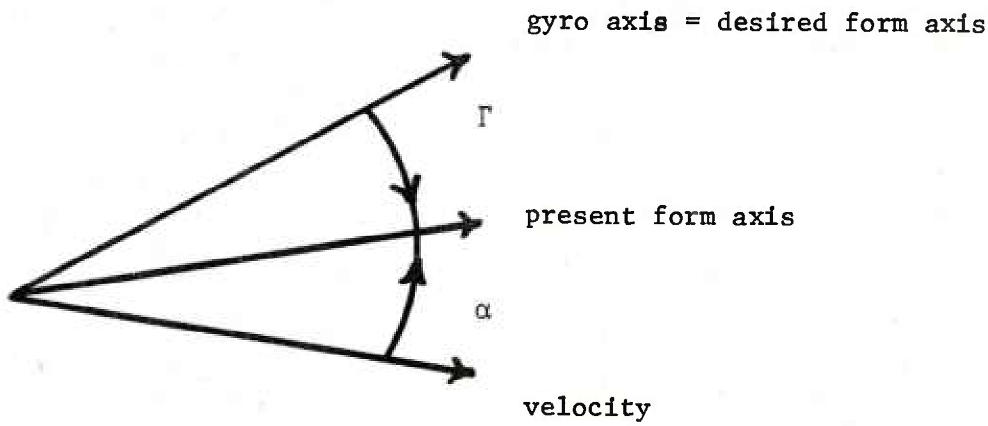
Each of these points is discussed in the following paragraphs.

A. Trajectory. The ballistic trajectory is typical of all but the Martin ED CLGP in glide mode. For the TI and Navy projectiles, the gyro is electrically caged to the projectile form axis between spinup and acquisition, while in the Martin AD case a gyro cant (or lookdown angle) was prescribed. The Martin ED ballistic mode is uncanted, like TI and Navy. The small lag in the electrically caged gyro may be accounted for, if desired, by means of the cant option. The cant is a pitchwise bias applied to gyro position relative to the form axis only during the preacquisition phase of flight. During ballistic flight, the control surfaces of the Navy projectile are in the trail position; all of the Army models have a synthetic pitch/yaw rate damping circuit which is active after spinup (thus, it is active after acquisition-enable), so that some control surface deflection may occur during "ballistic" flight.

The Martin ED glide mode projectile is initialized in glide trim; the gyro attitude and the deflection and trim angles required for zero transverse acceleration at initialization, given projectile position and velocity, are computed by an iterative procedure. The glide mode is an attitude-hold mode which "cages" the form axis on a free gyro, in effect the reverse of the ballistic situation. ZOT.14 maintains this attitude in an ad hoc fashion as follows: at each time step, the bias deflection required to align the axis is computed as

$$\delta_B = (\alpha - \Gamma)/K_\alpha \quad ,$$

where δ_B is the required bias, α is the present angle of attack, Γ is the present gimbal angle, and K_α is the ratio of trim angle of attack to control surface deflection. The variable δ_B passes through the proportional navigation (PN) channel before being applied to the control surfaces. The following figure and equation explain the rationale of the above equation:



$$\alpha_{desired} = K_\alpha \cdot \delta_{desired} = \alpha - \Gamma$$

B. Gravity Bias. G-bias is present only for the various Martin projectiles. For the AD version, it is a fixed value applied through the proportional navigation (PN) channel only; for the ED version, it is an internally-computed value applied through both PN and gyro torquer channels. However, since the computation of the bias is accomplished prior to the simulation's initial point, it is entered as a constant in the ballistic mode. Any value input in glide mode is ignored during the preacquisition glide, since the program computes the initial trim internally. The value of bias in glide mode is updated as long as the preacquisition glide continues. Thereafter, it is fixed at either the last computed value or the input value, as selected (there is no actual projectile which switches to a preset value -- the Martin ED holds the last computed value).

Note that G-bias is conceived as a deflection and not as an LOS rate. However, during active guidance, the torquer bias channel causes the gyro to drift at a fixed rate to introduce an artificial upward LOS rate into the PN channel. This is used in the Martin ED projectile to achieve the desired overbias. Since the optimum level of bias during guidance is somewhat greater than the one gee which glide requires, this is one way to increase the bias at the appropriate time; in fact, this is the way Martin has chosen. Also, if acquisition is lost, the bias returns to the one-gee level; for all biased models, the deflection bias remains on during a lost-acquisition period.

C. Guidance Enablement. Several strategies are implemented: the LOS rate input to the PN channel is shut off until either (1) the gyro has first been torqued according to its characteristic to within a nominal angular distance from null and a subsequent delay has then elapsed; or (2) the gyro has torqued according to its characteristics for a fixed time. The procedure of (1) is accomplished at reacquisition as well; strategy (2) permits immediate guidance upon reacquisition regardless of initial LOS error. During the nulling period, the projectile may, if specified, be steered in a "pursuit" mode in which the control deflection is equal to the deflection applied before acquisition (i.e., the glide bias if gliding and zero if ballistic) plus a term proportional to the gimbal angle. Note that during the nulling period, the gimbal angle rate gives rise to a small perturbing control deflection via the synthetic damping circuit, which in effect attributes all gimbal rates to body rates, not gyro rates.

The purpose of any nulling strategy (guidance enablement strategy) is to allow the gyro-LOS heading error initially existing upon acquisition to be eliminated before attempting PN guidance. Without some such strategy, the projectile will act upon erroneous steering information.

D. Rocket Booster. The Navy 5-inch guided projectile has a rocket booster, for which two effects are modeled: thrust and torque due to misalignment of the thrust vector and center of mass (malthrust). Both of these are modeled deterministically. The malthrust moment vector can be made to rotate about the form axis at a fixed rate to account for the effect of the rolling of the projectile's airframe upon the direction of the malthrust.

E. Bifunctional Tracking. For the study of a projectile which can initially acquire a laser spot and subsequently switch to a passive IR signature, the bifunctional option is provided. The transition occurs whenever (1) slant range to the actual target is less than a deterministic transition range and (2) the actual target is within the IR FOV. In this model no spot motion (a property of the designator, target, and projectile, jointly) is present; however, an internal noise source, IR detector jitter, is present. This noise is modeled as a normally-distributed, uncorrelated angular error in the detected LOS angles in yaw and pitch.

F. Behavior upon Loss of Acquisition. Ballistic models recage their gyros according to their torquing characteristics; gliding models let their gyros go free. Whatever deflection bias was present in guidance remains applied. During the decorrelation sequence immediately preceding loss of acquisition, gyros are freed and estimated LOS rates hold their previous values.

XI. PREPROCESSING

Preprocessing consists of three phases: scenario selection, zoning solution, and signature development.

A. Scenario Selection. In this first phase, the user chooses gun, designator, and target locations upon a ground plane. The nature of the gun, designator, target and projectile are all specified as well. Finally, the environment is characterized by such factors as ground plane altitude, visibility, and cloud height.

B. Zoning Solution. The second phase utilizes the exterior ballistic simulation program EXBAL to compute a family of trajectories from which a suitable solution for the problem can be taken.

C. Signature Development. The ERIM target reflectivity simulation model generates a record of apparent spot positions as seen from the seeker's point of view.

Input data include: the designator position and direction of approach of the projectile, in the target frame of reference; a geometric description of the surface of the target vehicle; and the dynamic characteristics of designator spot motion.

The record of spot positions is further processed to compute the spectral density and the covariance square root matrix, a matrix which is the matrix square root of the covariance matrix. This square root matrix is used to transform uncorrelated noise records into properly correlated noise records in ZOT.14. Alternatively, the ERIM spot position record may be used in ZOT.14, directly.

XII. POSTPROCESSING

The postprocessing program IMPAC may be used for further scoring of impacts. This program uses a simplified geometric description of the target vehicle and the record of impacts punched onto cards by ZOT.14 to score each impact as a hit or a miss. Hits may be further scored by computing the impact obliquity (angle between the projectile form axis and the normal to the target surface element struck).

Next page is blank.

APPENDIX A

USER'S GUIDE

Next page is blank.

INPUT

Input to ZOT.14 consists of a set of at least 15 punched cards. Some cards are optional, others required. The card set is assembled as follows:

CARD 1, required, format (8I10), contents by field:

(1)	MRUN	positive integer -- user's run identification number
(2)	MODEL	integer identifier of aero coefficient data -- 1 - Martin Marietta AD 2 - Texas Instruments AD 3 - Navy 5"/155mm sleeved round, AD 4 - Martin Marietta ED
(3)	NREPS	maximum number of replications to be run
(4)	NSAMPL	number of integration time steps between designator pulses
(5)	NPRINT	print control code 0 - print final results of replications only 1, 2 - print also intermediate results at intervals as specified by INTRVL and print notices of key events
(6)	INTRVL	print interval, number of integration time steps between printouts
(7)	NPUNCH	punch control code 0 - no punch 1 - punch results of each replication onto a pair of cards for IMPAC program
(8)	MEASUR	units of measure control code 1 - input interpreted as being in metric units (mks) 2 - input in English units

CARD 2, required, format (7I10,6X,4I1), contents by field:

(1)	ISEED	positive odd seed for spot motion generation
(2)	ISEED2	positive odd seed for pulse dropout generation
(3)	LENGTH	block length for spot motion; if zero, LENGTH will be defined at end of first replication
(4)	LENGT2	block length for pulse dropout; if LENGTH (above) is zero, LENGT2 will be defined at end of first replication

(5) MODESM spot motion mode code
1 - internal generation mode
2 - mixed external position and internal intensity
3 - external generation mode

(6) MSEEN number of consecutive seen pulses required for correlation (acquisition)

(7) MDROP number of consecutive unseen pulses required for decorrelation (loss of acquisition)

(8) NTD number of LOS deviations tabulated in the detector characteristic tables (see also CARD 10 and para. VI, B)

(9) NSL number of signal levels in detector characteristic tables

(10) MODDEP gyro torquing mode dependence code
1 - gyro torquing characteristic not mode dependent
2 - torquing characteristic different for nulling and guiding phases

(11) NMODES number of distinct detector characteristic sets
1 - SAL only, or IR detector characteristics are identical to SAL
2 - IR detector characteristics are different from SAL

CARD 3, required, format (8F10.0), contents by field:

(1) XP x-coordinate of initial projectile position [m;ft]*

(2) YP y-coordinate (altitude above ground)

(3) ZP z-coordinate (Note: (x,y,z) constitutes a right-hand coordinate system

(4) VXP x-component of initial projectile velocity with respect to earth [m/s;ft/s]

(5) VYP y-component

(6) VZP z-component

(7) H integration time step [sec]

(8) GROUND ground plane altitude above mean sea level [m;ft]

---

*The first unit in brackets applies when MEASUR=1, the second when MEASUR=2. If only one unit appears, it applies in either case.

CARD 4, required, format (8F10.0), contents by field:

(1)	M	projectile mass, exclusive of fuel if rocket-assisted [kg;lbm] (Note: all variables listed under F-formats are real-valued, regardless of initial letter of name.)
(2)	IB	projectile transverse moment of inertia [$kg \cdot m^2$; $slug \cdot ft^2$]
(3)	KDEL1	autopilot gain for SAL guidance [sec] K_{δ_1}
(4)	KDEL2	autopilot gain for IR guidance [sec] K_{δ_2}
(5)	TDEL	autopilot time constant τ_δ [sec]
(6)	ZALF	projectile natural pitch rate damping coefficient [dimensionless]
(7)	BIAS2	constant gravity-compensation deflection bias [deg]
(8)	CANT/KLDOT	for a ballistic-mode projectile, CANT is the gyro search cant angle, the angle downward from the projectile axis to the gyro axis [deg]; for a glide-mode projectile, KLDOT is the torquer-loop bias gain $K_{\dot{\lambda}}$ [sec^{-1}]

CARD 5, required, format (8F10.0); contents by field:

(1)	FOV(1)	instantaneous field of view for SAL, center of FOV to edge [deg]
(2)	FOV(2)	field of view for IR
(3)	COMLIM	limit on proportional navigation deflection commands [deg]
(4)	CONLIM	limit on control surface deflection [deg]
(5)	KQ	synthetic pitch-rate damping gain [sec]
(6)	TAUQ	synthetic damping time constant -- must be positive [sec]
(7)	DLNULL	null zone -- line-of-sight deviation must be less than DLNULL to accomplish null
(8)	WAIT	post null delay -- time during which null continues after achieving DLNULL [sec]

CARD 6, required, format (8F10.0), contents by field:

(1) FUELMS mass of fuel of rocket motor [kg; lbm]

(2) SIMPLS specific impulse of rocket [N•sec/kg; lbf•sec/lbm]

(3) THRUST rocket thrust [N;lbf]

(4) TBSTRT clock time (from launch) at burn start [sec]

(5) YLEVER rocket malthrust lever arm component creating positive yaw moment [m;in]

(6) PLEVER malthrust lever arm, pitch

(7) GDMAG if positive, deterministic gyro drift rate; if negative, absolute value is max random gyro drift rate [deg/sec]

(8) GDDIR if GDMAG positive, GDDIR is gyro drift direction CCW from the right [deg]; if GDMAG negative, GDDIR is seed number for gyro drift (positive odd)

CARD 7, required, format (8F10.0), contents by field:

(1) CSMA1 if CSMA2 $\leq$ 100.0, CSMA1 is the misalignment of the pitch control surface; if CSMA2 > 100.0, CSMA1 is the standard deviation of random control surface misalignment [deg]

(2) CSMA2 if $\leq$ 100.0, control surface misalignment is deterministic and CSMA2 is the misalignment of the yaw control surface [deg]; if > 100.0, misalignment is random and CSMA2 is the seed number (positive odd)

(3) ROLRAT angular rate at which the malthrust moment rotates about the form axis [deg/sec]

Remaining fields are not used.

CARD 8, required, format (8F10.0), contents by field:

(1) GAMAX max permissible gimbal angle [deg]

(2) XCUPLM gimbal cross-coupling coefficient at max gimbal angle [dimensionless]

(3) TIMEO clock time (from launch) at initialization of simulation [sec]

(4) AVGO mean length of a dropped-pulse string [sec]

(5) AVG1 mean length of a visible-pulse string [sec]

(6) VWIND wind speed [m/s;ft/s]

(7) THWIND azimuthal direction from which wind is blowing, measured CCW from +z axis [deg]

(8) GLIDON control code number, 3 digits followed by decimal:
1st digit = 0 for ballistic, 1 for glide mode
2nd digit = 0 for g-bias = BIAS2, 1 for g-bias = computed glide bias, during the guided phase of flight
3rd digit = 0 for null required only once, = 1 for null required on all acquisitions and reacquisitions

CARD 9, required, format (8F10.0), contents by field:

(1) P11 element of spot motion transform matrix [m;ft]

(2) P12

(3) P13

(4) P22

(5) P23

(6) P33

(7) FREQ corner frequency of spot motion power spectrum [hz]

(8) DR designation range [km;mi]

CARD 10, at least 1 card required, format (20F4.0), contents: first NTD* fields - vector of tabulated LOS deviations [deg], next NSL fields - vector of tabulated signal levels [db], next NTDxNSLxMODDEPxNMODES fields - table of estimated LOS rates as follows:

((((TMD(I,J,K,L),I=1,NTD),J=1,NSL),K=1,MODDEP), L=1,NMODES) [deg/sec]

next NTDxNSLxNMODES fields - table of commanded gyro torquing rates as follows:

(((TSR(I,J,L),I=1,NTD),J=1,NSL),L=1,NMODES) [deg/sec]

CARD 11, required, format (8F10.0), contents by field:

(1) THVCN angle between outward normal to designated target surface and the vertical [deg]

(2) PHVCN azimuthal angle of outward normal to designated target surface [deg]. Angle measured counterclockwise from Z - axis

---

*NTD, NSL, MODDEP, and NMODES are defined in CARD 2.

(3) LP laser pulse energy [J]

(4) THR detector threshold in units of [10^{-15} J/cm^2/pulse]

(5) REFL target surface reflectivity [dimensionless]

(6) VR visibility range [km;mi]

(7) CODE atmospheric water vapor code:
2. for 7.84 mm/km
3. for 1.87 mm/km

(8) CEIL cloud ceiling height above ground [m;ft]

CARD 12, required, format (8F10.0), contents by field:

(1) BMDIVG laser beam divergence [mrad]

(2) DYRANG detector lookdown instantaneous dynamic range [db]

(3) PRANGE visibility range due to particulate scattering only [km;mi]

(4) KAH attitude-hold gain

(5) SERRIR standard deviation of white angular noise in passive IR detector [mrad]

(6) SEEDIR seed for IR jitter (positive odd)

(7) RLNGT3 block length for IR jitter -- same rules as LENGT2

(8) not used

CARD 13, 2 cards required, format (8F10.0) --

CARD 13-1, contents by field:

(1) X(1) x-coordinate of initial reference spot location [m;ft]

(2) Y(1) y-coordinate

(3) Z(1) z-coordinate

(4) VEL(1) initial speed of reference spot location [m/s;ft/s]

(5) THETA(1) initial azimuthal heading of ref spot location [deg]

(6) XINCR x-wise increment in location of targets between replications [m;ft]

(7) ZINCR z-wise increment in location

(8) not used

CARD 13-2, contents by field:

(1) X(2) x-coordinate of initial location of desired impact point [m;ft]

(2) Y(2) y-coordinate

(3) Z(2) z-coordinate

(4) VEL(2) initial speed of desired impact point [m/s;ft/s]

(5) THETA(2) initial azimuthal heading [deg]

(6) RSWICH slant range within which projectile may switch to passive IR tracking mode [m;ft]

(7) XINCH position units override code -- if not zero, then the positions of both targets are interpreted as being in inches

(8) not used

CARD 14, required, format (8I10), contents by field:

(1) MANVRS(1) number of programmed evasive maneuvers for target 1 (the designator signature)

(2) MANVRS(2) number of evasive maneuvers for target 2 (the desired impact point)

Remaining fields are not used.

CARD 15, optional, format (8F10.0) --

The number of cards equals the sum MANVRS(1) + MANVRS(2). Cards are in the order of execution of maneuver for target 1, followed by order of execution for target 2.

Contents by field:

(1) duration of target's straight run preceding the maneuver [sec]

(2) maneuver code:
- 2. = acceleration/deceleration
- 3. = constant speed/radius turn
- 4. = sinusoidal zigzag

(3) for code 2, absolute value of acceleration [m/s^2]

for code 3, angle of turn, positive left (driver's perspective) [deg]
for code 4, number of complete sine waves

(4) for code 2, final speed [m/s]
for code 3, turning radius [m]
for code 4, not used

Remaining fields are not used.

CARD 16, optional, format (I10,4F10.0) --
Used only if MODESM = 2 or 3; then 1 card required.

Contents by field:

(1)	NSPOTS	number of spots in external record
(2)	CFACTR	multiplicative correction factor to be applied to scattering cross-sections of external record, to allow for changes in reflectivity
(3)	DESGPT(1)	distance forward from vehicle trailing edge to mean spot position, times -1 [in]
(4)	DESGPT(2)	height of mean spot position above ground [in]
(5)	DESGPT(3)	distance from centerline leftward to mean spot location [in]

Remaining fields are not used.

CARD 17, optional, format (20X,E10.3,3F10.0) --
Used only if MODESM = 2 or 3; then NSPOTS cards required.

Contents by field:

(1)	BRITE	scattering cross-section [in^2/ster]
(2)	YV(1)	distance forward from vehicle trailing edge to spot position [in]
(3)	YV(2)	distance from centerline leftward (from driver's perspective) to spot position [in]
(4)	YV(3)	height of spot position [in]

Card decks thus assembled may be stacked in series; the last deck should be followed by a blank card.

NOTES ON DATA SETUPS FOR VARIOUS APPLICATIONS

(1) Post-Enablement Phase of Unguided Trajectory.

Usually, all noise sources are zeroed out.* To prevent acquisition, do not zero laser energy, but rather set ceiling to zero. To prevent premature termination, place the targets considerably farther downrange than the projectile is expected to fly. Select one replication, NPRINT = 2, and an appropriate print interval (usually the equivalent of 0.5 to 2.0 seconds).

(2) Reference Guided Trajectory.

Usually, noise sources are zeroed out. Select other parameters as appropriate to the scenario. Usually, NREPS=1, NPRINT=2, INTRVL as required for a 0.1- to 0.5-sec print interval.

NOTE: If the time at which the projectile breaks cloud or comes within acquisition range is significantly later than the enablement time, and if the trajectory dispersion due to random gyro drift is negligible or not present, then much computer time can be saved by using a point just prior to breaking cloud or coming within acquisition range as the initial point in further runs studying the same scenario.

(3) Projectile Response.

These are usually runs of one to five replications. Some error or noise source of interest has been added to the previously noise-free scenario of a reference guided trajectory, and the behavior of the projectile is to be studied. The print interval is smaller than previously (unless the noise source is of the round-to-round type), perhaps equal to or even less than the pulse interval. Any number of modeled noise sources may be added in until the complete scenario of interest is simulated.

(4) Guidance Accuracy.

Guidance accuracy simulations usually account for all pertinent error sources. The print level is NPRINT=0, INTRVL=0, and the punch option NPUNCH=1 is taken if postprocessing with IMPAC is to be performed. The number of replications NREPS is selected with due consideration of the precision required in the estimates of guidance accuracy statistics; standard statistical techniques give the standard error of an estimate as a function of sample size.

(5) Footprint Analysis.

Footprints are generated by means of a series of noise-free runs in

---

*Valid seed numbers must be provided for spot motion and pulse dropout, even if these error sources are not used. Also, a valid seed number must be provided for IR jitter if the passive tracking mode is used.

which the target-location incrementing option is utilized. A typical starting procedure would be as follows:

21 replications, incrementing the (stationary) target by 50-m steps from the ballistic/pseudoballistic impact point toward the gun (uprange).

21 replications, incrementing in the same manner from the PBIP crossrange.

21 replications, incrementing downrange.

Presumably, at some point either the projectile misses the target by more than some arbitrary distance (say 5 feet or 1.5m) or the projectile is unable to acquire the target. The interval between two successive replications (a hit and a miss) plotted on a ground-plane grid, is one data "point" defining the boundary of the footprint.

If a run ends before finding a boundary point, the user sets up a new run picking up where the previous one left off. This is commonly required in locating the stretch point (farthest downrange point).

After these three initial points have been located, the user can use additional crosstrack-stepping runs to fill in the gaps in the boundary to any precision desired.

(6) Parametric Studies.

Studies of the effect of varying parameters may take the form of any of the above as appropriate.

OUTPUT

(1) Printout.

Printout consists of three parts: data, intermediate results, and final results.

(a) Data section

The printout begins with approximately three pages repeating the input data in a readable format.

(b) Intermediate results section

This part of the printout is controlled by NPRINT and INTRVL. The intermediate results may consist of only a line indicating where the projectile impacted relative to the target, or it may also contain notices of events such as acquisition/loss of acquisition, pulse dropout/reappearance, target evasive maneuver, etc., and notices of the projectile state at specified time intervals. These results are in order of replication, and within a replication, in chronological order.

(c) Final results section

This part is printed if multiple replications are specified and all replications were completed. It consists of a list of guidance accuracy statistics, including ordered miss distances.

(2) Punched Card Output.

This output is used as input to the IMPAC program. It consists of two cards per replication containing the relative position, the velocity, and the attitude of the projectile at closest approach to the target.

Next page is blank

PROGRAM LISTING

Next page is blank

```
      IMPLICIT REAL*8(A-H, O-Z)
      REAL*8 M, IB, KDEL, KDEL1, KDEL2, KALF(2), LRT, MILE, MXDLOS
      REAL*8 KQ, KLDOT, MP, KA
      REAL*4 NORM, NORM1, SLRNG1, VXR, VZR, SLRNG2, VAPOR, HRANGE,PRANGE
      REAL*4 MACHNO, URN, SERRIR, DUMMY, AMAX, AMIN, S
      REAL*4 CHAMAX, CHAMIN, SOMA
      DIMENSION R(100), Y(100), P(100), S0(12), S1(12), S2(12)
      DIMENSION MOONAM(4), TEXT3(2), TEXT4(2), TEXT5(2), TEXT6(2)
      DIMENSION TEXT7(3), FOV(2), TEXT8(2)
      DIMENSION VTD(9), VSL(9), TMD(9,9,2), TSR(9,9,2,2)
      EQUIVALENCE (S1(1),XP), (S1(2),YP), (S1(3),ZP),  (S1(4),VXP),
     1 (S1(5),VYP), (S1(6),VZP), (S1(7),ALPHAY), (S1(8),ALPHAP), (S1(9),
     2 DALPHY), (S1(10),DALPHP), (S1(11),DELTAY), (S1(12),DELTAP)
      EQUIVALENCE (S2(1),XP2), (S2(2),YP2), (S2(3),ZP2), (S2(4),VXP2),
     1 (S2(5),VYP2), (S2(6),VZP2), (S2(7),ALPHY2), (S2(8),ALPHP2), (S2(
     2 9),DALFY2), (S2(10),DALFP2), (S2(11),DELTY2), (S2(12),DELTP2)
      DATA POUND / 2.2046D0 /, SLFTSQ / .7375600 /, FOOT / 3.2808D0 /,
     1  G / 9.80665D0 /,  MILE / .6213700 /
      DATA TEXT1, TEXT2, TEXT3, TEXT4 / 'METERS) ', 'FEET)    ',
     1 'SPOT MOT', 'ION     ', 'PULSE DR', 'OPOUT   ' /
      DATA TEXT5, TEXT6 / 'METRIC  ', 'ENGLISH ', 'NO  ', 'YES ' /
      DATA TEXT7 / 'INTERNAL', 'MIXED   ', 'EXTERNAL' /
      DATA TEXT8 / 'IR JITTE','R       ' /
      DATA MOONAM / 'M/AD', ' TI ', 'NAVY', 'M/ED' /
      COMMON / BLK1 / XS, YS, ZS,                HN,        GROUND
      COMMON / BLK2 / XT, YT, ZT, KDEL, KDEL1, KDEL2, SLRNG1, SLRNG2
      COMMON / BLK3 / XCA, YCA, ZCA, RMISS, YMISS, PMISS, VXR, VZR
      COMMON / BLK4 / H, PI, PRF, MEASUR
      COMMON / BLK5 / PGYRO, TGYRO, CANT, BIASP, BIAS2, OLAMDY, OLAMDP,
     1              DROPS, KACQ, NRVS, IPRINT, ISTEP, NPRINT,JSTEP,NO
      COMMON / BLK6 / AVG0, AVG1
      COMMON / BLK7 / PHVCN, THVCN, LRT, B, CEIL, OR
      COMMON / BLK8 / FOV, MODESM, KOND, ITARG, NSW
      COMMON / BLK9 / FNY, FNP, FA,TALF(2),VSQ,CNA, AQ, KALF, IB, MODEL
      COMMON / BLK10 / QPGYRO, QTGYRO
      COMMON / BLK11 / CND, CAO , MACHNO
      COMMON / BLK12 / BRIGHT, TIME0, NULL, IVIS, INCR
      COMMON / BLK13 / DRATIO, EFACTR, E, THR
      COMMON / BLK14 / OBETAY, OBETAP
      COMMON / BLK15 / VAPOR, HRANGE, PRANGE
      COMMON / BLK16 / DLNULL, WAIT,         MSEEN, MOROP
      COMMON / BLK17 / BETAY,BETAP,IGLIDE,IBIAS,INULL,NSEEN,JREP
      COMMON / BLK18 / SERRIR, IRSEED, LENGT3, NRVS3
C
C
C
    1 FORMAT(8I10)
    2 FORMAT(8F10.0)
    3 FORMAT(7I10, 6X, 4I1)
   12 FORMAT(' *** ERROR -- PROJECTILE FAILS TO APPROACH TARGET WITHIN T
     1IME LIMIT')
   13 FORMAT(  '0TIME', 6X, 'X', 7X, 'Y', 7X, 'Z', 5X, 'VX', 5X, 'VY',
     1 5X, 'VZ', 5X,'AY', 5X, 'AP', 4X, 'DAY', 4X, 'DAP', 3X, 'DELY',
     2 2X, 'DELP', 3X, 'DLY', 3X, 'DLP', 6X, 'XT', 6X, 'ZT', 5X, 'RSR',
     3 2X, 'RNAV')
   14 FORMAT(1X, F5.2, 3F8.1, 3F7.1, 4F7.3, 4F6.3, 3F8.1, F5.2)
   20 FORMAT( '0*** AT STEP ', I4, ' PROJECTILE IMPACTS BALLISTICALLY.
     1RUN TERMINATED.')
   21 FORMAT(  '0*** REDEFINE BLOCK SIZE ', I5, ' FOR ', 2A8)
   22 FORMAT(  '0*** ERROR -- BLOCK SIZE EXCEEDED FOR ', 2A8)
```

```
23 FORMAT(15X, 'X=', F7.1, ', Y=', F7.1, ', Z=', F7.1, ';  VX=', F6.100005800
 1 , ', VY=', F6.1, ', VZ=', F6.1, ';  XT=', F7.1, ', ZT=', F7.1)      00005900
24 FORMAT(/ T6, 15( '-'), 'CLOSEST APPROACH', 16('-'), T54, 4('-'),    00006000
 1 'TARGET', 5('-'), T70, 'SHORT-', T77, 21('-'), 'SUMMATIONS',        00006100
 2 23('-')/ ' REP', T10, 'X', T18, 'Y', T26, 'Z', T34, 'R', T42,       00006200
 3 'Y', T50, 'P', T59, 'X', T67, 'Z', T71, 'FALL', T84, 'SR', T95,     00006300
 4 'SY', T106, 'SP', T116, 'SY2', T127, 'SP2')                         00006400
25 FORMAT(1X, I3, 6F8.3, 2F8.1, 7X, 5F11.4)                             00006500
26 FORMAT(1X, I3, 6F8.3, 2F8.1, F7.2, 5F11.4)                           00006600
27 FORMAT('ZOT.14 RUN', I4, ', REP', I3, T36, 3F15.6)                  00006700
28 FORMAT(// 66(' -') // ' FINAL RESULTS (IN ',  A8, '--' / T40, 'YAW00006800
 1 ERROR', 5X, 'MEAN', 2X, F8.4, 5X, 'STD. DEV.  ', F8.4 / T38, 'PIT00006900
 2CH ERROR', 5X, 'MEAN  ', F8.4, 5X, 'STD. DEV.  ', F8.4 / T35, 'ABS00007000
 3OLUTE ERROR', 5X, 'MEAN  ', F8.4, 5X, 'STD. DEV.  ', F8.4 )         00007100
29 FORMAT(T54, 'BASED ON ', I6, ' OBSERVED SPOT POSITIONS')            00007200
30 FORMAT(// ' SORTED YAW ERRORS --'/ )                                00007300
31 FORMAT(10(3X,F10.6))                                                 00007400
32 FORMAT(// ' SORTED PITCH ERRORS --'/ )                              00007500
33 FORMAT(// ' SORTED ABSOLUTE ERRORS --'/)                            00007600
34 FORMAT(// 10X, I4, ' SHOTS FELL SHORT OF TARGET.' )                 00007700
35 FORMAT(10X, 'AVERAGE NUMBER OF DROPPED-PULSE STRINGS WAS ', F10.4,00007800
 1 ' PER REPLICATION.')                                                 00007900
36 FORMAT(// 66(' -'))                                                  00008000
37 FORMAT (T38, 'SPOT MOTION  X  MEAN  ', F8.4, 5X, 'STD. DEV.  ',     00008100
 1 F8.4, / T51, 'Y  MEAN  ', F8.4, 5X, 'STD. DEV.  ', F8.4/ T51,      00008200
 2 'Z  MEAN  ', F8.4, 5X, 'STD. DEV.  ', F8.4)                         00008300
41 FORMAT ('1', T64, 'ZOT.14'// T33, 'PROGRAM FOR SIMULATION OF THE P00008400
 1OST-ENABLEMENT PHASE OF CLGP FLIGHT'// T39, 'FEATURES:', T53, 'ACQ00008500
 2UISITION/REACQUISITION LOGIC'/ T53, 'DESIGNATOR SPOT MOTION WITH P00008600
 3ULSE DROPOUT'/ T53, 'MANEUVERABLE TARGET'/ T53, 'BIFUNCTIONAL TRAC00008700
 4KING'// 66(' -') )                                                   00008800
42 FORMAT (/ T60, 'RUN NUMBER ', I3// T53, 'CLGP MODEL IS ', A4, ' VE00008900
 1RSION'// 66(' -')// T53, 'PROGRAM CONTROL PARAMETERS'// T42, I10, 00009000
 2' REPLICATIONS'/ T42, I10, ' MILLISECONDS BETWEEN DESIGNATOR PULSE00009100
 3S'/ T42, I10, ' PRINT CONTROL' )                                     00009200
43 FORMAT (T42, I10, ' MILLISECOND PRINT INTERVAL' )                   00009300
44 FORMAT (T42, F10.4, ' SECOND INTEGRATION INTERVAL'/ T53, 'PUNCHED 00009400
 1OUTPUT OPTION: ', A4 )                                               00009500
45 FORMAT (T33, 'SEED VALUE ', I10, ' AND BLOCK LENGTH ', I6, ' FOR '00009600
 1, 2A8 )                                                              00009700
46 FORMAT (T46, 'INPUT DATA ARE IN ',  A8, 'SYSTEM'/ )                 00009800
47 FORMAT (66(' -')// T52, 'PROJECTILE DESIGN PARAMETERS'// ' PARAMET00009900
 1ER',T60, 'METRIC UNITS', T90, 'ENGLISH UNITS'// ' MASS', T60, F10.00010000
 24, ' KG', T90, F10.4, ' LB'/ ' PITCH MOMENT OF INERTIA', T60, F10.00010100
 34, ' KG*M**2', T90, F10.4, ' SLUG*FT**2'/ ' K.DELTA FOR SEMI-ACTIV00010200
 4E MODE', T60, F10.4, ' SEC'/ T10, 'FOR PASSIVE MODE', T60, F10.4, 00010300
 5 ' SEC')                                                             00010400
48 FORMAT('0TAU.DELTA',T60,F10.4,' SEC'/' ZETA.ALPHA',T60,F10.4/'0GRA00010500
 1VITY BIAS ANGLE', T60, F10.4, ' DEG UPWARD'/                         00010600
 3                       ' GYRO CANT ANGLE',T60,F10.4,' DEG DOWNWARD'/00010700
 4'0GYRO FIELD OF VIEW, LASER',T60,F10.4,' DEG PLUS OR MINUS'/T22,'I00010800
 5NFRARED',T60,F10.4,' DEG PLUS OR MINUS'/' MAX CONTROL SURFACE DEFL00010900
 6ECTION',T60,F10.4,' DEG PLUS OR MINUS'/)                             00011000
49 FORMAT(/66(' -')// T52, 'PROJECTILE INITIAL CONDITIONS'// ' POSITI00011100
 1ON', T32, 'X', T60, F10.3, ' M', T90, F10.3, ' FT'/ T32, 'Y (ABOVE00011200
 2 GROUND LEVEL)', T60, F10.3, ' M', T90, F10.3, ' FT'/ T32, 'Z', T600011300
 30, F10.3, ' M', T90, F10.3, ' FT'// ' VELOCITY', T32, 'X', T60,     00011400
 4 F10.3, ' M/SEC', T90, F10.3, ' FT/SEC'/ T32, 'Y', T60, F10.3,      00011500
 5 ' M/SEC', T90, F10.3, ' FT/SEC' )                                   00011600
50 FORMAT(T32,'Z',T60,F10.3,' M/SEC',T90,F10.3,' FT/SEC'//' CONTROL S00011700
```

```
     1URFACE DEFLECTIONS AND ANGLES OF ATTACK ARE ZERO.'//66(' -')//' TI00011800
     2ME AT THIS POINT OF ENABLEMENT IS ',F6.2,' SEC.'//' GROUND LEVEL (00011900
     3ABOVE SEA LEVEL)',T60,F10.3,' M',T90,F10.3,' FT'/' WIND VELOCITY',00012000
     4T60,F10.3,' M/SEC',T90,F10.3,' FT/SEC'/' WIND DIRECTION IS FROM',T00012100
     560,F10.3,' DEG CC FROM +Z AXIS.'//66(' -'))                        00012200
   51 FORMAT (/    T48, 'DESIGNATION AND ACQUISITION PARAMETERS'//      00012300
     1 ' DESIGNATION RANGE', T60, F10.4, ' KM', T90, F10.4, ' MI'/      00012400
     3 ' SPOT MOTION GENERATION MODE IS ', A8)                          00012500
   52 FORMAT(' SPOT POSITION',T50,'| ',3(F10.4,1X),'|',4X,'| ',3(F10.4,100012600
     1X),'|'/T7,'TRANSFORMATION',T50,'|     0.0000 ',2(F10.4,1X), '| M, 00012700
     2|     0.0000 ',2(F10.4,1X),'| FT'/T12,'MATRIX P',T50,'| ',2('    000012800
     3.0000 '),F10.4,' |    | ',2('    0.0000 '),F10.4,' |'//' DESIGNATI00012900
     4ON CORNER FREQUENCY', T60, F10.4, ' HZ' )                         00013000
   53 FORMAT (' MEAN TIME IN DROPPED STATE', T60, F10.4, ' SEC'/T12,    00013100
     1 'IN SEE STATE', T60, F10.4, ' SEC'/)                             00013200
   54 FORMAT(' CONSECUTIVE SEEN PULSES FOR ACQUISITION', T60, I5/ ' CONS00013300
     1ECUTIVE DROPPED PULSES FOR LOSS OF ACQUISITION', T60, I5/         00013400
     2                                                      ' NULL ZONE 00013500
     3FOR COMMENCEMENT OF GUIDANCE', T60, F10.4, ' DEG PLUS OR MINUS'/  00013600
     4 ' POST-NULL DELAY', T60, F10.4, ' SEC' / )                       00013700
   55 FORMAT(' MAX CONTROL SURFACE GUIDANCE COMPONENT', T60, F10.4, ' DE00013800
     1G PLUS OR MINUS'/' K,Q', T60, F10.4, ' SEC'/' TAU,Q', T60, F10.4, 00013900
     2 ' SEC'/)                                                         00014000
   56 FORMAT (' MAX LOS RATE CROSS-COUPLING COEFFICIENT', T60, F10.4/   00014100
     1 ' MAX GIMBAL ANGLE',     T60, F10.4, ' DEG'/)                    00014200
   57 FORMAT(1X,66('$*')/' **** ERROR **** GIMBAL',' ANGLE OUT OF LIMITS00014300
     1 -- YAW=', F10.4, ', PITCH=', F10.4, ' AT STEP', I?, '.  EXECUTION00014400
     2 CONTINUING.'/ 1X, 66('$*'))                                      00014500
   58 FORMAT ('ZOT.14 RUN', I4, T41, F10.4)                             00014600
   59 FORMAT (2I4, T16, 2F10.6, 3F15.6)                                 00014700
   60 FORMAT (' TORQUER BIAS GAIN KLDOT',T60,F10.4,' SEC**(-1)'/)       00014800
   61 FORMAT ('0COMPUTED  INITIAL ATTITUDE OF GLIDING PROJECTILE --'/   00014900
     1 '   BIAS = DELTA =', F10.6, ', ALPHA =', F10.6, ', THETA,GYRO =', 00015000
     2 F10.6, ' DEGREES AFTER ',I2,' ITERATIONS WITH DDELTA =',F10.7,   00015100
     3 'RAD'/' INITIAL MACH NUMBER ', F10.4/)                           00015150
   62 FORMAT(' THIS IS NOT A ROCKET-ASSISTED PROJECTILE.'/)             00015151
   63 FORMAT( ' ROCKET-ASSISTED PROJECTILE:'/T5,'ADDED MASS OF FUEL',   00015152
     1 T60,F10.3,' KG',T90,F10.3,' LB'/T5,'SPECIFIC IMPULSE',T60,F10.3, 00015153
     2 ' NEWTON-SEC/KG', T90,F10.3,' LBF-SEC/LBM'/T5,'THRUST',T60,F10.3,00015154
     3 ' NEWTONS',T90,F10.3,' LB')                                      00015155
   64 FORMAT(T5,'YAW MALTHRUST LEVER ARM',T60,F10.5,' M',T90,F10.5,' IN'00015156
     1 /T5,'PITCH MALTHRUST LEVER ARM',T60,F10.5,T90,F10.5/T5,'TIME BURN00015157
     2 STARTS',T60,F10.3,' SEC'/T5,'DURATION OF BURN',T60,F10.3/T5,'TIME00015158
     3 BURN ENDS',T60,F10.3/T5,'LEVER ARM ROLL RATE',T60,F10.3,' D/S'/) 00015159
   65 FORMAT(' GYRO DRIFT RATE', T60, F10.5, ' DEG/SEC'/ ' GYRO DRIFT DI00015160
     1RECTION', T60, F10.3, ' DEG CCW FROM RIGHT'/)                     00015161
   66 FORMAT(' MAX GYRO DRIFT RATE', T60, F10.5, ' DEG/SEC'/ ' GYRO DRIF00015162
     1T SEED NUMBER', T60, I10/)                                        00015163
   67 FORMAT(' DURING NULL THE ATTITUDE-HOLD CHANNEL WILL REMAIN ON WITH00015164
     1 GAIN KAH =', F8.3)                                               00015165
   68 FORMAT( ' PITCH CONTROL SURFACE MISALIGNMENT ', T60,F10.4, ' DEG'/00015170
     1 ' YAW CONTROL SURFACE MISALIGNMENT', T60, F10.4)                 00015171
   69 FORMAT(' CONTROL SURFACE MISALIGNMENT STD. DEV.', T60, F10.4, 'DEG00015172
     1'/ ' MISALIGNMENT SEED VALUE', T60, I10)                          00015173
C                                                                       00015200
C                                                                       00015300
C--   THE MAIN PROGRAM PERFORMS INPUT PERTAINING TO PROGRAM CONTROL AND 00015400
C     TO THE PROJECTILE.  IT INITIALIZES THE SYSTEM AND INTEGRATES THE  00015500
C     FINITE-DIFFERENCE EQUATIONS OF THE PROJECTILE FLY-IN.  IT ACCUMULAT00015600
C     MISS STATISTICS AND PRINTS OUT A REPORT.                          00015700
```

```
C
      PI = 3.14159264359D0
      RADIAN = PI / 180.
      TWOPI = 2.*PI
C
  100 CONTINUE
C--   DATA INPUT SECTION
C--   READ PROGRAM CONTROL PARAMETERS
      READ (5,1) MRUN, MODEL, NREPS, NSAMPL, NPRINT, INTRVL, NPUNCH,
     1      MEASUR
      IF (MRUN.LE.0)    STOP
      READ (5,3) ISEED, ISEED2, LENGTH, LENGT2, MODESM, MSEEN, MDROP,
     1      NTD, NSL, MODDEP, NMODES
C--   READ PROJECTILE INITIAL STATE
      READ (5,2) (S1(I), I=1,6), H, GROUND
C--   INITIALIZE REMAINING STATE-VECTOR ELEMENTS
      DO 110 I=7,12
  110 S1(I) = 0.
      DO 120 I=1,12
  120 S0(I) = S1(I)
C--   READ PROJECTILE DESIGN PARAMETERS
      READ (S,2) MP, IB, KDEL1, KDEL2, TDEL, ZALF, BIAS2, CANT, FOV,
     1      COMLIM, CDNLIM, KG, TAUQ, DLNULL, WAIT
      READ (5,2) FUELMS, SIMPLS, THRUST, TBSTRT, YLEVER, PLEVER, GDMAG,
     1      GDDIR
      READ(S,2) CSMA1, CSMA2, ROLRAT
      READ (5,2) GAMAX, XCUPLM, TIME0, AVG0, AVG1, VWIND, THWIND,GLIDON
C--   READ DESIGNATION PARAMETERS
      READ(S,2) P11, P12, P13, P22, P23, P33, FREQ, DR
      IF (AVG1.LE.0.D0)    AVG1 = 1000.
      NSAMPT = NSAMPL*H*1000. + 0.5
      WRITE (6,41)
      WRITE (6,42) MRUN, MODNAM(MODEL), NREPS, NSAMPT, NPRINT
      IF (NPRINT.LE.0)    GD TO 130
      INTVLT = INTRVL*H*1000. + 0.5
      WRITE (6,43) INTVLT
  130 CONTINUE
      WRITE (6,44) H, TEXT6(NPUNCH+1)
      WRITE (6,45) ISEED, LENGTH, TEXT3, ISEED2, LENGT2, TEXT4
      WRITE (6,46) TEXT5(MEASUR)
      GO TD (140,160), MEASUR
  140 CONTINUE
C--   MEASUR=1    =>    INPUT IS METRIC
      DO 150 I=1,6
  150 S2(I) = S1(I)*FOOT
      EMP = MP*PDUND
      EIB = IB*SLFTSQ
      EDR = DR*MILE
      EP11 = P11*FDOT
      EP12 = P12*FOOT
      EP13 = P13*FOOT
      EP22 = P22*FOOT
      EP23 = P23*FDDT
      EP33 = P33*FOOT
      EGR = GROUND*FODT
      EVW = VWIND*FOOT
      EFUELM = FUELMS*PDUND
      ESIMPL = SIMPLS/G
      ETHRST = THRUST*PDUND/G
      EYLVR = YLEVER*FDDT*12.
```

```
      EPLVR = PLEVER*FOOT*12.                                            00021014
      GO TO 180                                                          00021100
  160 CONTINUE                                                           00021200
C--  MEASUR=2    =>    INPUT IS ENGLISH                                  00021300
      DO 170 I=1,6                                                       00021400
      S2(I) = S1(I)                                                      00021500
      S1(I) = S1(I)/FOOT                                                 00021600
  170 S0(I) = S1(I)                                                      00021700
      EMP = MP                                                           00021800
      MP = MP/POUND                                                      00021900
      EI8 = I8                                                           00022000
      I8 = I8/SLFTSQ                                                     00022100
       EDR = DR                                                          00022200
      DR = DR/MILE                                                       00022300
      EP11 = P11                                                         00022400
      EP12 = P12                                                         00022500
      EP13 = P13                                                         00022600
      EP22 = P22                                                         00022700
      EP23 = P23                                                         00022800
      EP33 = P33                                                         00022900
      P11 = P11/FOOT                                                     00023000
      P12 = P12/FOOT                                                     00023100
      P13 = P13/FOOT                                                     00023200
      P22 = P22/FOOT                                                     00023300
      P23 = P23/FOOT                                                     00023400
      P33 = P33/FOOT                                                     00023500
      EGR = GROUND                                                       00023600
      GROUND = GROUND/FOOT                                               00023700
      EVW = VWIND                                                        00023800
      VWIND = VWIND/FOOT                                                 00023900
      EFUELM = FUELMS                                                    00023901
      FUELMS = FUELMS/POUND                                              00023902
      ESIMPL = SIMPLS                                                    00023903
      SIMPLS = SIMPLS*G                                                  00023904
      ETHRST = THRUST                                                    00023905
      THRUST = THRUST*G/POUND                                            00023906
      EYLVR = YLEVER                                                     00023907
      YLEVER = YLEVER/(FOOT*12.)                                         00023908
      EPLVR = PLEVER                                                     00023909
      PLEVER = PLEVER/(FOOT*12.)                                         00023910
  180 CONTINUE                                                           00024000
      KLDOT = 0.                                                         00024100
C--  GLIDON GENERATES 3 INTEGER CONTROL VARIABLES:                        00024200
C          IGLIDE = 0 FOR NO GLIDE, 1 FOR GLIDE                          00024250
C          IBIAS = 0 FOR FIXED BIAS DURING GUIDANCE, 1 FOR COMPUTED      00024300
C          INULL = 0 FOR 1-TIME NULL, 1 FOR NULL EACH REACQUISITION      00024350
      IGLIDE = 1.D-2*GLIDON + .001                                       00024400
      GLIDON = GLIDON - 100.*IGLIDE                                      00024450
      IBIAS = 1.D-1*GLIDON + .01                                         00024500
      GLIDON = GLIDON - 10.*IBIAS                                        00024550
      INULL = GLIDON + .1                                                00024600
      IF (IGLIDE.EQ.0)    GO TO 185                                      00024650
      IF (IBIAS.EQ.0)   GO TO 182                                        00024700
      KLDOT = CANT                                                       00024750
      BIAS2 = 0.                                                         00024800
  182 CANT = 0.                                                          00024830
  185 CONTINUE                                                           00024860
      WRITE (6,47) MP, EMP, I8, EI8, KDEL1, KDEL2                        00024900
      WRITE (6,48) TDEL, ZALF, BIAS2,          CANT, FOV, CONLIM         00025000
      WRITE (6,55) COMLIM, KQ, TAUQ                                      00025100
```

```
C--  COMPUTE ROCKET-BURN PARAMETERS                                     00025101
      TFBURN = 0.                                                       00025102
      IF (THRUST.GT.0.D0)   TFBURN = SIMPL5*FUELMS/THRUST               00025103
      IF (TFBURN.EQ.0.D0)   TBSTRT = -10000.                            00025104
      TBSTOP = TBSTRT + TFBURN                                          00025105
      IF (TFBURN.GT.0.D0)   GO TO 9180                                  00025106
      WRITE (6,62)                                                      00025107
      GO TO 9182                                                        00025108
 9180 WRITE(6,63)FUELMS,EFUELM,SIMPLS,ESIMPL,THRUST,ETHRST              00025109
      WRITE(6,64)YLEVER,EYLVR,PLEVER,EPLVR,TBSTRT,TFBURN,TBSTOP,ROLRAT  00025110
      IF (ROLRAT.NE.0.D0)   GO TO 9181                                  00025111
      DYMT = H*THRUST*YLEVER/IB                                         00025112
      DPMT = H*THRUST*PLEVER/IB                                         00025113
      GO TO 9182                                                        00025114
 9181 OMT = H*THRUST*OSQRT(YLEVER**2 + PLEVER**2)/IB                    00025115
      AMT = OATAN2(PLEVER,YLEVER)                                       00025116
 9182 CONTINUE                                                          00025117
C--  COMPUTE GYRO-DRIFT PARAMETERS                                      00025118
      IF (GOMAG.LT.0.D0)   GO TO 918S                                   00025119
C--  GYRO ORIFT IS DETERMINISTIC                                        00025120
      IGO = 0                                                           00025121
      WRITE (6,65) GDMAG, GODIR                                         00025122
      GO TO 9186                                                        00025123
 918S CONTINUE                                                          00025124
C--  GYRO-DRIFT IS RANDOM                                               00025126
      IGO = 1                                                           00025128
      GOMAG = -GDMAG                                                    00025130
      ISEED3 = GDDIR                                                    00025132
      WRITE (6,66)  GDMAG,  ISEED3                                      00025134
      GOMAG = GDMAG*RAOIAN                                              00025136
 9186 CONTINUE                                                          00025138
C--  COMPUTE CONTROL-SURFACE MISALIGNMENT PARAMETERS                    00025140
      IF (CSMA2.GT.100.D0)   GO TO 9287                                 00025142
C--  MISALIGNMENT IS DETERMINISTIC                                      00025144
      IMA = 0                                                           00025146
      WRITE(6,68) CSMA1, CSMA2                                          00025148
      DELPMA = CSMA1*RADIAN                                             000251S0
      OELYMA = CSMA2*RADIAN                                             00025152
      GO TO 9288                                                        00025154
 9287 CONTINUE                                                          00025156
C--  MISALIGNMENT IS RANDOM                                             000251S8
      IMA = 1                                                           00025160
      ISEEOM = CSMA2                                                    00025162
      WRITE(6,69) CSMA1, ISEEOM                                         00025164
      SOMA = CSMA1*RAOIAN                                               00025166
      CMAMAX = 3.*SOMA                                                  00025168
      CMAMIN = -CMAMAX                                                  00025170
 9288 CONTINUE                                                          00025172
      WRITE (6,56) XCUPLM, GAMAX                                        00025200
      WRITE (6,S4) MSEEN, MDROP,          OLNULL, WAIT                  00025300
      WRITE (6,60) KLDOT                                                00025400
      CALL SKRTBL (NTD, NSL, MOOOEP, NMOOES, VTD, VSL, TMO, TSR)        00025500
      WRITE (6,49) (S1(I), S2(I), I=1,S)                                00025600
      WRITE (6,50) S1(6), S2(6), TIMEO, GROUND, EGR                     00025700
     1   , VWIND, EVW, THWINO                                           00025800
      WRITE (6,S1) OR, EDR,            TEXT7(MODESM)                    00025900
      IF (MODESM.EQ.1)   WRITE (6,53) AVGO, AVG1                        00026000
      IF (MODESM.EQ.1)   WRITE (6,52) P11,P12,P13,EP11,EP12,EP13, P22,  00026100
     1     P23,EP22,EP23, P33,EP33, FREQ                                00026200
C--  REAO ACQUISITION PARAMETERS                                        00026300
```

```
      CALL AQDATA (KAH)                                                 00026400
      S = SERRIR * I.D-3                                                00026420
      AMAX = 4. * S                                                     00026425
      AMIN = - AMAX                                                     00026430
      HN = H*NSAMPL                                                     00026500
C--   READ TARGET MOTION PARAMETERS                                     00026600
      CALL DODGER(I,SI,TGTHDG)                                          00026700
C                                                                       00026800
C                                                                       00026900
C                                                                       00027000
C--   INITIALIZATION SECTION                                            00027100
      M = MP                                                            00027110
      IF (TIME0.LT.TBSTOP)    M = MP+ FUELMS*(TBSTOP-TIME0)/TFBURN      00027111
      IF (TIME0.LT.TBSTRT)    M = MP + FUELMS                           00027112
      F2 = DEXP(-H/TAUQ)                                                00027200
      FI = 1.D0 - F2                                                    00027300
      ROLRAT = ROLRAT*RADIAN                                            00027350
      BIAS2 = BIAS2*RADIAN                                              00027400
      CANT = CANT*RADIAN                                                00027500
      FOV(I) = FOV(1)*RADIAN                                            00027600
      FOV(2) = FOV(2)*RADIAN                                            00027700
      CONLIH = CONLIM*RADIAN                                            00027800
      COMLIM = COHLIM*RADIAN                                            00027900
      DLNULL = DLNULL*RADIAN                                            00028000
      GAMAX = GAMAX*RADIAN                                              00028100
      RXCUPL = XCUPLH/GAHAX                                             00028200
C--   IF GYRO DRIFT IS DETERMINISTIC, DEFINE IT NOW                     00028210
      IF (IGD.EQ.1)   GO TO 9187                                        00028211
      GDHAG = GDHAG*RADIAN                                              00028212
      GDDIR = GDDIR*RADIAN                                              00028213
      YGD = -H*GDMAG*DCOS(GDDIR)                                        00028214
      PGD = +H*GDHAG*DSIN(GDDIR)                                        00028215
 9187 CONTINUE                                                          00028216
C--   IF CONTROL SURFACE MISALIGNMENT IS DETERMINISTIC, IT IS ALREADY   00028250
C     DEFINED . . .                                                     00028251
      WX = VWIND * DSIN(THWIND*RADIAN)                                  00028300
      WZ = VWIND * DCOS(THWIND*RADIAN)                                  00028400
      DIST = DSORT( (XP-XT)**2 + (YP-YT)**2 + (ZP-ZT)**2)               00028500
      VP = DSQRT( VXP**2 + VYP**2 + VZP**2 )                            00028600
      TIHE = DIST/VP                                                    00028700
C--   THE SIMULATION LIMITS ITSELF TO 1.5 TIMES THE NUMBER OF STEPS REQUI00028800
C     TRAVERSE THE REMAINING SLANT RANGE AT THE INITIAL VELOCITY.       00028900
      MAXSTP = I.5*TIHE/H                                               00029000
      PRF = I./HN                                                       00029100
      VXPW = VXP + WX                                                   00029200
      VZPW = VZP + WZ                                                   00029300
      VH = DSQRT( VXPW**2 + VZPW**2 )                                   00029400
      OPGYRO = DATAN2(VXPW, VZPW)                                       00029500
      QTGYRO = DATAN2(VYP, VH) - CANT                                   00029600
      IF (IGLIDE.EO.0)    GO TO 188                                     00029700
C--   GLIDE IS SELECTED -- CALCULATE INITIAL BIAS AND ATTITUDE          00029800
C     A SUFFICIENT CONDITION FOR NUMERICAL STABILITY OF THE FOLLOWING   00029850
C     ITERATIVE PROCEDURE IS THAT K.ALPHA NOT INCREASE WITH DELTA.      00029851
      VSQ = VH**2 + VYP**2                                              00029900
      ITS = 0                                                           00029950
  186 CALL PARAMS (SI)                                                  00030000
      BIASP = H*G*VH/( DSORT(VSO)*AO*((CNA-CAO)*KALF(2)+CND) )          00030100
      DDELTA = DABS( BIASP-DELTAP )                                     00030150
      DELTAP = BIASP                                                    00030300
      ALPHAP = KALF(2)*DELTAP                                           00030400
```

```
      ITS = ITS + 1                                                     00030420
      IF (ITS.GE.20)   GO TO 187                                        00030440
      IF (DDELTA.GT.0.0002)   GO TO 186                                 00030460
  187 CONTINUE                                                          00030480
      GAMAP = 0.                                                        00030500
      QTGYRO = DATAN2(VYP,VH) + ALPHAP                                  00030600
      OUT1 = BIASP/RADIAN                                               00030700
      OUT2 = ALPHAP/RADIAN                                              00030800
      OUT3 = QTGYRO/RADIAN                                              00030900
      WRITE (6,61) OUT1, OUT2, OUT3, ITS, DDELTA, MACHNO                00031000
      SO(8) = ALPHAP                                                    00031100
      SO(12) = DELTAP                                                   00031200
      WRITE (6,67) KAH                                                  00031250
  188 CONTINUE                                                          00031300
C--  COMPUTE SPOT MOTION DIGITAL FILTER COEFFICIENTS OR READ ARRAY OF SP00031600
      MD= MODESM                                                        00031700
      T = HN                                                            00031800
      CALL SPOT1(MO, P11, P12, P13, P22, P23, P33, T, FREQ, TGTHDG)     00031900
      SNORM = NORM(ISEED)                                               00032000
      ISTEP = 0                                                         00032100
      CALL DODGER (2, S1, TGTHDG)                                       00032200
      IF (IGD.EQ.1)    INCR = 1                                         00032220
      IF (INCR.EQ.1)    CALL DODGER (3,S1,TGTHDG)                       00032250
      NSF = 0                                                           00032300
      NSPOTS = 0                                                        00032400
      LOGIC = 0                                                         00032500
      IF(LENGTH.GT.0)    LOGIC = 1                                      00032600
      IFLGGY = 0                                                        00032700
      SUMXS = 0.                                                        00032800
      SUMYS = 0.                                                        00032900
      SUMZS = 0.                                                        00033000
      SUMXS2 = 0.                                                       00033100
      SUMYS2 = 0.                                                       00033200
      SUMZS2 = 0.                                                       00033300
      SUMPCH = 0.                                                       00033400
      SUMYAW = 0.                                                       00033500
      SUMRAD = 0.                                                       00033600
      SUMPSQ = 0.                                                       00033700
      SUMYSQ = 0.                                                       00033800
      DROPS = 0.                                                        00033900
      KACQ = 1                                                          00034000
      NO = 0                                                            00034100
      JSTEP = 0                                                         00034200
C                                                                       00034300
C                                                                       00034400
C                                                                       00034500
C--  BEGIN REPLICATIONS                                                 00034600
      DO 500 IREP =1,NREPS                                              00034700
      NRVS3 = 0                                                         00034720
      JREP = IREP                                                       00034750
      BIASY = 0.                                                        00034760
      IF (IGLIDE.EQ.0)    BIASP = 0.                                    00034800
      NULL = 1                                                          00034900
      E = THR                                                           00035000
      IF (IREP.GT.1)    GO TO 190                                       00035100
      IF ((P11**2 + P12**2 + P13**2 + P22**2 + P23**2 + P33**2).EQ.0.D0)00035200
     1     GO TO 200                                                    00035300
      IF (MODESM.NE.1)    GO TO 200                                     00035400
      N = S.*PRF                                                        00035500
      CALL SPOTMO (N, XSS, YSS, ZSS)                                    00035600
```

```
      GO TO 200                                                           00035700
  190 CONTINUE                                                            00035800
      IF (INCR.EQ.0)    GO TO 195                                         00035810
      KACQ = 1                                                            00035820
      GO TO 200                                                           00035830
  195 CONTINUE                                                            00035840
      KACQ = 2                                                            00035850
      NSEEN = NSEEN - 1                                                   00035900
  200 CONTINUE                                                            00036500
C--   IF GYPO DRIFT IS RANDOM, DETERMINE IT NO.                           00036510
      IF (IGO.EQ.0)    GO TO 205                                          00036511
      GD = GOMAG*URN(ISEED3)                                              00036512
      GODIR = TWOPI*URN(ISEED3)                                           00036513
      YGO = -H*GD*DCOS(GODIR)                                             00036514
      PGD = +H*GD*DSIN(GODIR)                                             00036515
  205 CONTINUE                                                            00036516
C--   IF CONTROL SURFACE MISALIGNMENT IS RANDOM, DETERMINE IT NOW         00036550
      IF (IMA.EQ.0)    GO TO 206                                          00036552
      CALL NORMXX(CMAMIN, CMAMAX, 0., SDMA, DUMMY, ISEEDM)                00036554
      DELPMA = DUMMY                                                      00036556
      CALL NORMXX(CMAMIN, CMAMAX, 0., SDMA, DUMMY, ISEEDM)                00036558
      DELYMA = DUMMY                                                      00036560
  206 CONTINUE                                                            00036562
      N = JSTEP/NSAMPL                                                    00036600
      IF (N.GT.0)    CALL SPOTMO (N, XSS, YSS, ZSS)                       00036700
      XS = XSS                                                            00036800
      YS = YSS                                                            00036900
      ZS = ZSS                                                            00037000
      ISTEP = JSTEP                                                       00037100
      NRVS = NO                                                           00037200
      IPRINT = 1                                                          00037300
      PGYRO = QPGYRO                                                      00037400
      TGYRO = QTGYRO                                                      00037500
      PGYROX = PGYRO                                                      00037600
      TGYROX = TGYRO                                                      00037700
      DPGYRO = 0.                                                         00037800
      DTGYRO = 0.                                                         00037900
      GAMAY = 0.                                                          00038000
      DO 210 I=1,I2                                                       00038100
  210 S1(I) = S0(I)                                                       00038200
      VXPW = VXP + WX                                                     00038210
      VZPW = VZP + WZ                                                     00038220
      VH = DSQRT(VXPW**2 + VZPW**2)                                       00038300
      THVW = DATAN2(VYP, VH)                                              00038400
      PHVW = DATAN2(VXPW, VZPW)                                           00038450
      DTHVW = 0.                                                          00038460
      DPHVW = 0.                                                          00038462
      GAMAP = THVW - TGYRO + ALPHAP                                       00038500
      BETAP = THVW    + ALPHAP                                            00038510
      BETAY = DATAN2(VXPW,VZPW)                                           00038511
      ADYMT = 0.                                                          00038512
      ADPMT = 0.                                                          00038513
      DLAMDY = 0.                                                         00038600
      DLAMDP = 0.                                                         00038700
      DCOMY = 0.                                                          00038800
      DCOMP = BIASP                                                       00038900
      DY11 = 0.                                                           00039000
      DP11 = DCOMP                                                        00039100
      DY41 = 0.                                                           00039200
      DP41 = 0.                                                           00039300
```

```
      DY42 = 0.                                                         00039400
      DP42 = 0.                                                         00039500
      IF (IREP.LE.1)   GO TO 220                                        00039600
      CALL DODGER (4,S1, TGTHDG)                                        00039700
      GO TO 245                                                         00039800
C                                                                       00039900
C                                                                       00040000
C                                                                       00040100
  220 CONTINUE                                                          00040200
C--  BEGIN STEP CALCULATIONS                                             00040300
      VSQ = VXPW**2 + VYP**2 + VZPW**2                                  00040600
      VP = DSQRT(VSQ)                                                   00040700
      CALL PARAMS (S1)                                                  00040800
C--  CONVERT BODY FORCES FROM BODY FRAME OF REFERENCE TO ZOT COORDINATES00040900
      FD = -FA - FNP*ALPHAP - FNY*ALPHAY                                00041000
      FLP = -FA*ALPHAP + FNP                                            00041100
      FLY = -FA*ALPHAY - FNP*ALPHAY*ALPHAP + FNY                        00041200
      VH = DSQRT( VXPW**2 + VZPW**2 )                                   00041300
      FX = FD*VXPW/VP + FLY*VZPW/VH - FLP*VXPW*VYP/(VH*VP)              00041400
      FY = FD*VYP/VP + FLP*VH/VP                                        00041500
      FZ = FD*VZPW/VP - FLY*VXPW/VH - FLP*VYP*VZPW/(VH*VP)              00041600
C--  ADD ROCKET THRUST & MOMENT                                          00041610
      TIME = TIME0 + H*ISTEP                                            00041612
      M = MP + FUELMS                                                   00041613
      IF (TIME.LT.TBSTRT)   GO TO 223                                   00041614
      IF (TIME.GE.TBSTOP)   GO TO 221                                   00041616
      FX = FX + THRUST*DSIN(BETAY)*DCOS(BETAP)                          00041618
      FY = FY + THRUST*DSIN(BETAP)                                      00041620
      FZ = FZ + THRUST*DCOS(BETAY)*DCOS(BETAP)                          00041622
      IF (ROLRAT.NE.0.D0)   GO TO 9220                                  00041623
      ADYMT = DYMT                                                      00041624
      ADPMT = DPMT                                                      00041625
      GO TO 9221                                                        00041626
 9220 AROLL = AMT + ROLRAT*(TIME-TBSTRT)                                00041627
      ADYMT = DMT*DCOS(AROLL)                                           00041628
      ADPMT = DMT*DSIN(AROLL)                                           00041629
 9221 CONTINUE                                                          00041630
C--  UPDATE PROJECTILE MASS                                              00041631
      M = MP + FUELMS*(TBSTOP-TIME)/TFBURN                              00041632
      GO TO 223                                                         00041633
  221 CONTINUE                                                          00041634
      M = MP                                                            00041636
      ADYMT = 0.                                                        00041638
      ADPMT = 0.                                                        00041640
  223 CONTINUE                                                          00041642
C--  INTEGRATE STATE DIFFERENTIAL EQUATIONS                              00042000
      XP2 = XP + H*VXP                                                  00042100
      YP2 = YP + H*VYP                                                  00042200
      ZP2 = ZP + H*VZP                                                  00042250
      VXP2 = VXP + H* FX/M                                              00042300
      VYP2 = VYP + H*(FY/M - G)                                         00042400
      VZP2 = VZP + H* FZ/M                                              00042500
      VXPW = VXP2 + WX                                                  00042510
      VZPW = VZP2 + WZ                                                  00042512
      VH = DSQRT(VXPW**2 + VZPW**2)                                     00042514
      THVW2 = DATAN2(VYP2,VH)                                           00042516
      PHVW2 = DATAN2(VXPW,VZPW)                                         00042518
      DTHVW2= (THVW2 - THVW)/H                                          00042520
      DPHVW2= (PHVW2 - PHVW)/H                                          00042522
      ALPHY2 = ALPHAY + H*DALPHY                                        00042600
```

```
      ALPHP2 = ALPHAP + H*DALPHP
      DALFY2 = DALPHY + H*(KALF(1)*DELTAY/TALF(1)**2 - (ALPHAY/TALF(1)
     1      + 2.*ZALF*DALPHY) / TALF(1) ) + ADYMT
     2      - (DPHVW2-DPHVW) * DCOS(BETAP)
      DALFP2 = DALPHP + H*(KALF(2)*DELTAP/TALF(2)**2 - (ALPHAP/TALF(2)
     1      + 2.*ZALF*DALPHP) / TALF(2) ) + ADPMT -(DTHVW2-DTHVW)
      DY12 = DY11 + H*(DCOMY - DY11) / TDEL
      DP12 = DP11 + H*(DCOMP - DP11) / TDEL
      BETAP = THVW2 + ALPHP2
      BETAY = PHVW2 + ALPHY2/DCOS(BETAP)
      IF (DY12.GT.+COMLIM)   DY12 = +COMLIM
      IF (DY12.LT.-COMLIM)   DY12 = -COMLIM
      IF (DP12.GT.+COMLIM)   DP12 = +COMLIM
      IF (DP12.LT.-COMLIM)   DP12 = -COMLIM
      PGYROX = PGYROX + DPGYRO
      TGYROX = TGYROX + DTGYRO
      IF (KACQ.EQ.5) TGYROX = TGYROX - H*KLDOT*(BIASP-DELPMA)
      IF (KACQ.EQ.5) PGYROX = PGYROX-H*KLDOT*(BIASY-DELYMA)/DCOS(TGYROX)
      GAMAY2 =-PGYROX + BETAY
      IF (GAMAY2.GT.+PI)   GAMAY2 = GAMAY2 - TWOPI
      IF (GAMAY2.LT.-PI)   GAMAY2 = GAMAY2 + TWOPI
      GAMAY2 = GAMAY2*DCOS(BETAP)
      GAMAP2 =-TGYROX + BETAP
      GYDOT = (GAMAY2-GAMAY)/H
      GPDOT = (GAMAP2-GAMAP)/H
      IF (IFLGGY.NE.0)   GO TO 222
      IF ((DABS(GAMAY2).LT.GAMAX).AND.(DABS(GAMAP2).LT.GAMAX)) GO TO 222
      GAMAY = GAMAY2/RADIAN
      GAMAP = GAMAP2/RADIAN
      WRITE (6,57) GAMAY, GAMAP, ISTEP
      IFLGGY = 1
  222 CONTINUE
      DY42 = KQ*GYDOT*F1 + DY41*F2
      DP42 = KQ*GPDOT*F1 + DP41*F2
      DELTY2 = DY12 - DY42
      DELTP2 = DP12 - DP42
      IF ((KACQ.NE.5).OR.(NULL.EQ.3))   GO TO 227
      DELTY2 = DELTY2 - KAH*GAMAY2
      DELTP2 = DELTP2 - KAH*GAMAP2
  227 CONTINUE
C--  IMPACT CHECK
      PDOTV = (XP2-XT)*VXP2 + (YP2-YT)*VYP2 + (ZP2-ZT)*VZP2
      IF ((PDOTV.GE.0.D0).OR.(YP2.LE.0.D0 ))   GO TO 390
      ISTEP = ISTEP + 1
      IF (ISTEP.LE.MAXSTP)  GO TO 230
      WRITE (6,12)
      GO TO 100
  230 CONTINUE
C--  PROJECTILE IS STILL FLYING IN
      DO 240 I=1,12
  240 S1(I) = S2(I)
      DY11 = DY12
      DP11 = DP12
      DY41 = DY42
      DP41 = DP42
      GAMAY = GAMAY2
      GAMAP = GAMAP2
      THVW = THVW2
      PHVW = PHVW2
      DTHVW = DTHVW2
```

```
      OPHVA = OPH**2                                                    00048116
C--  CANARD LIMITER                                                     00048200
      IF (DELTAY.GT.+CONLIM)  DELTAY = +CONLIM                          00048300
      IF (DELTAY.LT.-CONLIM)  DELTAY = -CONLIM                          00048400
      IF (DELTAP.GT.+CONLIM)  DELTAP = +CONLIM                          00048500
      IF (DELTAP.LT.-CONLIM)  DELTAP = -CONLIM                          00048600
      N = ISTEP - ISTEP/NSAMPL*NSAMPL                                   00048700
      IF (N.NE.0)   GO TO 250                                           00048800
C--   PERIODIC TARGET POSITION COMPUTATION                              00048900
C    (DODGER CALLS SPOTMO FOR SPOT POSITION)                            00049000
      CALL DODGER (2,S1, TGTHDG)                                        00049100
C--  COLLECT STATISTICS ON SPOT POSITION                                00049200
      SUMXS = SUMXS + XS                                                00049300
      SUMYS = SUMYS + YS                                                00049400
      SUMZS = SUMZS + ZS                                                00049500
      SUMXS2 = SUMXS2 + XS**2                                           00049600
      SUMYS2 = SUMYS2 + YS**2                                           00049700
      SUMZS2 = SUMZS2 + ZS**2                                           00049800
      NRVS = NRVS + 1                                                   00049900
      NSPOTS = NSPOTS + 1                                               00050000
C--  OBTAIN NEW LINE-OF-SIGHT RATES                                     00050100
  245 CONTINUE                                                          00050200
      XPGYRO = PGYROX                                                   00050300
      XTGYRO = TGYROX                                                   00050400
      PGYRO = PGYROX                                                    00050500
      TGYRO = TGYROX                                                    00050600
      CALL TRACK (S1,S0,ISEED2,TS*,ISEE,AMAX,AMIN,S)                    00050700
      IF (KACQ.EQ.6)    GO TO 100                                       00050800
C--  OBTAIN NEW GYRO POSITION DIFFERENTIALS                             00050900
      OPGYRO = (PGYRO - XPGYRO) / NSAMPL                                00051000
      DTGYRO = (TGYRO - XTGYRO) / NSAMPL                                00051100
      IF ( (IGLIDE.EQ.0) .AND. (KACQ.NE.5) )    GO TO 246               00051110
      DTGYRO = DTGYRO + PGO                                             00051111
      OPGYRO = DPGYRO + YGD/DCOS(TGYRO)                                 00051112
  246 CONTINUE                                                          00051113
C--  IF GLIDING, COMPUTE UPDATED BIAS DURING PRE-ACQUISITION OR USE OLD 00051200
C    BIAS DURING LOST-ACQUISITION.                                      00051300
      IF (IGLIDE.EQ.0)    GO TO 247                                     00051400
      IF (KACQ.GE.3)    GO TO 247                                       00051500
      BIASP = (ALPHAP -  GAMAP) / KALF(2)                               00051600
      BIASY = (ALPHAY - GAMAY) / KALF(1)                                00051601
  247 CONTINUE                                                          00051800
C--  CONVERT L. O. S. RATES INTO COMMANDED DEFLECTIONS                  00051900
      XCUPLY = DABS(GAMAY)*RXCUPL                                       00052000
      XCUPLP = OABS(GAMAP)*RXCUPL                                       00052100
      ELAMDY = OLAMOY + XCUPLY*DLAMDP                                   00052200
      ELAMOP = DLAMOP + XCUPLP*DLAMOY                                   00052300
      OCOMY = KDEL*ELAMDY + BIASY                                       00052400
      DCOMP = KDEL*ELAMDP + BIASP                                       00052500
  250 CONTINUE                                                          00052600
      IF (NPRINT.EQ.0)   GO TO 220                                      00052700
C--  PERIODIC OUTPUT                                                    00052800
      N = ISTEP - ISTEP/INTRVL*INTRVL                                   00052900
      IF (N.NE.0)   GO TO 220                                           00053000
      IF (IPRINT.EQ.1)    WRITE (6,13)                                  00053100
      IPRINT = 0                                                        00053200
      AKALFA = KALF(1)                                                  00053230
      IF ( OABS(ALPHAP) .GT. DABS(ALPHAY) )    AKALFA = KALF(2)         00053260
      RNAV = ( CNA*AKALFA + CND ) * KDEL*AQ / (M*VP)                    00053300
      TIME = TIME0 + ISTEP*H + .001                                     00053400
```

```
      WRITE (6,14) TIME, SI, DLAMDY, OLAMDP, XT, ZT, SLRNG2, RNAV       00053500
      GO TO 220                                                         00053600
C--   END STEP CALCULATIONS                                             00053700
C                                                                       00053800
C                                                                       00053900
C                                                                       00054000
  390 CONTINUE                                                          00054100
C--   PROJECTILE HAS IMPACTED                                           00054200
      IF (KACQ.GT.2)   GO TO 400                                        00054300
C--   BALLISTIC IMPACT                                                  00054400
      WRITE (6,20) ISTEP                                                00054500
      WRITE (6,23) (SI(I), I=I,6), XT, ZT                               00054600
      GO TO I00                                                         00054700
  400 CONTINUE                                                          00054800
C--   GUIOEO IMPACT                                                     00054900
C--   RANDOM NUMBER BLOCKING SECTION                                    00055000
      IF (LOGIC.GT.0)  GO TO 4I0                                        00055100
      LENGTH = NRVS + 2.*PRF + 0.5                                      00055200
      LENGT2 = NSW + I0                                                 00055300
      LENGT3 = NRVS3 + 4.*PRF + 0.S                                     00055350
      WRITE (6,2I) LENGTH, TEXT3, LENGT2, TEXT4, LENGT3, TEXT8          00055400
      LOGIC = I                                                         00055500
      GO TO 430                                                         00055600
  4I0 CONTINUE                                                          00055700
      IF (NRVS-LENGTH) 430,440,420                                      00055800
  420 CONTINUE                                                          00055900
      WRITE (6,22)  TEXT3                                               00056000
      GO TO 100                                                         00056100
  430 CONTINUE                                                          00056200
      N = LENGTH - NRVS                                                 00056300
      CALL SPOTMO (N, XSS, YSS, ZSS)                                    00056400
  440 CONTINUE                                                          00056500
      IF (NSW-LENGT2) 460,470,4S0                                       00056600
  4S0 CONTINUE                                                          00056700
      WRITE (6,22) TEXT4                                                00056800
      GO TO I00                                                         00056900
  460 CONTINUE                                                          00057000
      NSW = NSW + I                                                     00057100
      OO 46S I=NSW,LENGT2                                               00057200
      CALL SWITCH(ISEE, TSW, ISEED2)                                    00057300
  46S CONTINUE                                                          00057400
  470 CONTINUE                                                          00057500
      IF (FOV(2).EQ.0.00)   GO TO 474                                   00057510
      IF (NRVS3-LENGT3) 472,474,47I                                     00057515
  47I WRITE(6,22) TEXT8                                                 00057520
      GO TO I00                                                         00057525
  472 N = (LENGT3 - NRVS3) * 2                                          00057530
      I = IRSEEO                                                        00057535
      OO 473 J=I,N                                                      00057540
  473 CALL RANDMM(I,OUMMY)                                              00057545
      IRSEED = I                                                        00057550
  474 CONTINUE                                                          00057555
      IF (NPRINT.EQ.0)   GO TO 47S                                      00057600
      IF (IPRINT.EQ.I)   WRITE (6,I3)                                   00057700
      TIME = TIMEO + ISTEP*H + .00I                                     00057800
      AKALFA = KALF(1)                                                  00057830
      IF ( OABS(ALPHAP) .GT. OABS(ALPHAY) )   AKALFA = KALF(2)          00057860
      RNAV = ( CNA*AKALFA + CNO ) *KOEL*AQ / (M*VP)                     00057900
C--   SLANT RANGE USEO HERE IS THE VALUE CALCULATEO AT MOST RECENT PULSE 00058000
      WRITE (6,I4) TIME, SI, DLAMOY, OLAMDP, XT, ZT, SLRNG2, RNAV       00058100
```

```
  475 CONTINUE                                                          00058200
C--   COMPUTE MISS DISTANCE                                             00058300
      CALL DODGER (5.SI, TGTHDG)                                        00058400
      SUMYAW = SUMYAW + YMISS                                           00058S00
      SUMPCH = SUMPCH + PMISS                                           00058600
      SUMRAD = SUMRAD + RMISS                                           00058700
      SUM YSG = SUMYSQ + YMISS**2                                       00058800
      SUMPSQ = SUMPSQ + PMISS**2                                        00058900
      IF((IREP.EQ.I).OR.(NPRINT.GT.0))  WRITE (6,24)                    00059000
      IF (YT+YCA.LT. 0.D0 ) GO TO 480                                   00059100
      WRITE (6,2S) IREP, XCA, YCA, ZCA, RMISS, YMISS, PMISS, XT, ZT,    00059200
     I      SUMRAD, SUMYAW, SUMPCH, SUMYSQ. SUMPSQ                      000S9300
      GO TO 490                                                         000S9400
  480 CONTINUE                                                          00059500
      T = -(YT+YCA) / VYP                                               00059600
      XSF = XCA + VXR*T                                                 00059700
      ZSF = ZCA + VZR*T                                                 000S9800
      SF = DSQRT( XSF**2 + YT**2 + ZSF**2 )                             00059900
      WRITE (6,26) IREP, XCA, YCA. ZCA, RMISS, YMISS, PMISS, XT, ZT, SF,00060000
     I      SUMRAD, SUMYAW, SUMPCH, SUMYSG, SUMPSQ                      00060I00
      NSF = NSF + I                                                     00060200
  490 CONTINUE                                                          00060300
      R(IREP) =  RMISS                                                  00060400
      Y(IREP) = YMISS                                                   00060500
      P(IREP) = PMISS                                                   00060600
      IF (NPUNCH.LE.0)   GO TO 500                                      00060700
      IF (IREP.EQ.I)   WRITE (7.58) MRUN, TGTHDG                        00060800
      OUTI = RMISS*FOOT                                                 00060900
      OUT2 = YMISS*FOOT                                                 0006I000
      OUT3 = PMISS*FOOT                                                 0006II00
      OUT4 = VXP*FOOT                                                   0006I200
      OUTS = VZP*FOOT                                                   0006I300
      OUT6 = -VYP*FOOT                                                  0006I400
      WRITE(7,27) MRUN, IREP, OUT1, OUT2, OUT3                          0006IS00
      ALPHAY = ALPHAY/RADIAN                                            00061600
      ALPHAP = ALPHAP/RADIAN                                            0006I700
      WRITE (7,59) MRUN, IREP, ALPHAY, ALPHAP. OUT4, OUT5, OUT6         0006I800
  S00 CONTINUE                                                          0006I900
C--   ENO REPLICATION                                                   00062000
C                                                                       00062I00
C                                                                       00062200
C                                                                       00062300
C--   MAKE FINAL REPORT                                                 00062400
      IF (NREPS.LE.I)   GO TO 5S0                                       00062S00
      RREPS = NREPS                                                     00062SS0
      RSPOTS = NSPOTS                                                   00062SS5
      SIGP = 0.00                                                       00062S60
      SIGY = 0.00                                                       0006256S
      SIGR = 0.00                                                       00062570
      SIGXS = 0.00                                                      0006257S
      SIGYS = 0.00                                                      00062S80
      SIGZS = 0.00                                                      0006258S
      PBAR = SUMPCH/RREPS                                               00062600
      YBAR = SUMYAW/RREPS                                               00062700
      RBAR = SUMRAO/RREPS                                               00062800
      SUMRSQ = SUMYSQ + SUMPSQ                                          00062900
      HOLO = (RREPS*SUMPSQ - SUMPCH**2) / (RREPS*(RREPS-I.00))          00063000
      IF (HOLO.GT.0.00)   SIGP = DSQRT(HOLO)                            00063050
      HOLO = (RREPS*SUMYSG - SUMYAW**2) / (RREPS*(RREPS-I.00))          00063I00
      IF (HOLO.GT.0.00)   SIGY = DSQRT(HOLO)                            00063I50
```

```
      HOLD = (RREPS*SUMRSQ - SUMRAD**2) / (RREPS*(RREPS-1.D0))
      IF (HOLD.GT.0.D0)   SIGR = DSQRT(HOLD)
      XSBAR = SUMXS/RSPOTS
      YSBAR = SUMYS/RSPOTS
      ZSBAR = SUMZS/RSPOTS
      HOLD = (RSPOTS*SUMXS2 - SUMXS**2) / (RSPOTS*(RSPOTS-1.D0))
      IF (HOLD.GT.0.D0)   SIGXS = DSQRT(HOLD)
      HOLD = (RSPOTS*SUMYS2 - SUMYS**2) / (RSPOTS*(RSPOTS-1.D0))
      IF (HOLD.GT.0.D0)   SIGYS = DSQRT(HOLD)
      HOLD = (RSPOTS*SUMZS2 - SUMZS**2) / (RSPOTS*(RSPOTS-I.D0))
      IF (HOLD.GT.0.D0)   SIGZS = DSQRT(HOLD)
      WRITE (6,28) TEXT1,        YBAR, SIGY, PBAR, SIGP, RBAR, SIGR
      WRITE (6,37) XSBAR, SIGXS, YSBAR, SIGYS, ZSBAR, SIGZS
      WRITE (6,29) NSPOTS
      NM1 = NREPS - 1
      DO S40 I=I,NMI
      IP1 = I + 1
      DO S30 J=IPI,NREPS
      IF (Y(I).LE.Y(J))  GO TO S10
      HOLD = Y(I)
      Y(I) = Y(J)
      Y(J) = HOLD
  S10 CONTINUE
      IF (P(I).LE.P(J))  GO TO S20
      HOLD = P(I)
      P(I) = P(J)
      P(J) = HOLD
  S20 CONTINUE
      IF (R(I).LE.R(J))  GO TO S30
      HOLD = R(I)
      R(I) = R(J)
      R(J) = HOLD
  530 CONTINUE
  S40 CONTINUE
  SS0 CONTINUE
      WRITE (6,30)
      WRITE (6,31)  (Y(I),I=1,NREPS)
      WRITE (6,32)
      WRITE (6,31)  (P(I),I=1,NREPS)
      WRITE (6,33)
      WRITE (6,3I)  (R(I),I=I,NREPS)
      WRITE (6,34) NSF
      IF (NREPS.GT.1)   DROPS = DROPS/NREPS
      WRITE (6,3S) DROPS
      IF (NREPS.LE.1)  GO TO I00
      YBAR = YBAR*FOOT
      PBAR = PBAR*FOOT
      RBAR = RBAR*FOOT
      SIGY = SIGY*FOOT
      SIGP = SIGP*FOOT
      SIGR = SIGR*FOOT
      XSBAR = XSBAR*FOOT
      YSBAR = YSBAR*FOOT
      ZSBAR = ZSBAR*FOOT
      SIGXS = SIGXS*FOOT
      SIGYS = SIGYS*FOOT
      SIGZS = SIGZS*FOOT
      WRITE (6,28) TEXT2,        YBAR, SIGY, PBAR, SIGP, RBAR, SIGR
      WRITE (6,37) XSBAR, SIGXS, YSBAR, SIGYS, ZSBAR, SIGZS
      DO S60 I=1,NREPS
```

```
      R(I) = R(I)*FOOT
      Y(I) = Y(I)*FOOT
      P(I) = P(I)*FOOT
  S60 CONTINUE
      WRITE (6,30)
      WRITE (6,31) (Y(I),I=1,NREPS)
      WRITE (6,32)
      WRITE (6,31) (P(I),I=1,NREPS)
      WRITE (6,33)
      WRITE (6,31) (R(I),I=1,NREPS)
      WRITE (6,36)
      GO TO 100
      END
C--  TRACK  --  TRACK  --  TRACK  --  TRACK  --  TRACK  --  TRACK  --  T
      SUBROUTINE TRACK(SI,S0,ISEE02,TSW,ISEE,AMAX,AMIN,S)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 SLRNG1, SLRNG2, TRANS,RACQKM,AMAX,AMIN,S
      REAL*8 LOSRY, LOSRP, MXOLOS, KOEL, KOEL1, KOEL2, LRT
      OIMENSION SI(12), S0(I2), FOV(2)
      COMMON / BLKI / XS, YS, ZS,                 HN,      GROUND
      COMMON/BLK2/ XA, YA, ZA, KOEL, KOEL1, KOEL2, SLRNGI, SLRNG2
      COMMON/BLK4/ H, P1, PRF, MEASUR
      COMMON/BLKS/ PGYRO, TGYRO, CANT, BIAS, BIAS2, LOSRY, LOSRP,
     1   OROPS, KACQ, NRVS, IPRINT, ISTEP, NPRINT, JSTEP, NO
      COMMON / BLK7 / PHVCN, THVCN, LRT, B, CEIL, OR
      COMMON / BLKB / FOV, MOOESM, KONO, ITARG, NSW
      COMMON / BLKI0 / QPGYRO, QTGYRO
      COMMON / BLKI2 / CROSS, TIMEO, NULL, IVIS, INCR
      COMMON / BLK13 / ORATIO, EFACTR, E, THR
      COMMON / BLK14 / OBETAY, OBETAP
      COMMON / BLK16 / OLNULL, WAIT,           MSEEN, MDROP
      COMMON / BLK17 / BETAY,BETAP,IGLIDE,IBIAS,INULL,NSEEN,JREP
C
    1 FORMAT(/ ' *** SWITCH TO PULSE STATE ', II, ' AT STEP ', I4, T90,
     1 'TIME ', FI0.3)
    2 FORMAT(/ ' *** TARGET HAS SLIPPEO OUT OF FIELD OF VIEW ANO ACQUISI
     1TION BROKEN AT STEP ', I4, T90, 'TIME ', F10.3)
    3 FORMAT ('0*** AT STEP ', I4, ' GUIOANCE COMMENCES.', T90, 'TIME ',
     1 F10.3 )
   1S FORMAT(/ ' *** AT STEP ', I4, ' PROJECTILE IS WITHIN ACQUISITION R
     1ANGE.', T90, 'TIME ', F10.3)
   16 FORMAT(/ ' *** AT STEP ', I4, ' TARGET HAS BEEN SEEN THE REQUIRED
     1', I2, ' PULSES.  ACQUISITION COMMENCING.', T90, 'TIME ', F10.3)
   17 FORMAT(/ ' *** AT STEP ', I4, ' PROJECTILE IS SO SITUATEO THAT ACQ
     1UISITION IS NO LONGER POSSIBLE.  RUN TERMINATEO.')
   18 FORMAT(/ ' *** SWITCH FROM PULSE STATE ', I1, ' AT TIME ', F7.4)
   19 FORMAT(/ ' *** AT STEP ', I4, ' REACQUISITION SEQUENCE COMMENCES A
     1FTER ', I3, ' UNSEEN PULSES.', T90, 'TIME ', F10.3)
   23 FORMAT(1SX, 'X=', F7.1, ', Y=', F7.1, ', Z=', F7.1, ';  VX=', F6.1
     1 , ', VY=', F6.1, ', VZ=', F6.1, ';  XT=', F7.1, ', ZT=', F7.I,
     2 '; GYRO T=', F6.3, ', P=', F6.3 )
C
C--  TRACK CONTROLS THE POSITION OF THE SEEKER GYRO ANO COMPUTES LINE-OF
C    (LOS) RATES.  IT ALSO CONTROLS ACQUISITION, LOSS OF ACQUISITION, RE
C    ITION AND PULSE STATE.
C
C--  THE CONTROL VARIABLE  ISEE  INOICATES THE STATE OF THE PULSE:
C         0   =>   PULSE IS OROPPEO
C         1   =>   PULSE IS VISIBLE
C--  THE CONTROL VARIABLE  KACQ  INDICATES THE STATE OF THE SEEKER:
```

```
C          1    =>    TARGET HAS NOT BEEN DETERMINED TO BE IN ACQ. RANGE    00074800
C          2    =>    TARGET IS IN RANGE BUT HAS NOT BEEN DETERMINED TO BE 00074900
C          3    =>    DROPPED STATE PHASE 1: GYRO HOLDS LAST POSITION, LOS 00075000
C                     REMAINS AT LAST VALUE                                 00075100
C          4    =>    DROPPED STATE PHASE 2: GYRO RETURNS TO CAGED POSITION00075200
C                     RATE = 0                                              00075300
C          5    =>    ACQUIRED STATE                                        00075400
C          6    =>    ACQUISITION IS IMPOSSIBLE: RUN WILL TERMINATE         00075500
C--   THE CONTROL VARIABLE  NULL  INDICATES THE STATE OF THE CONTROLLER 000075600
C     THE ACQUIRED SEEKER STATE (KACQ=5):                                   00075700
C          1    =>    CONTROLS REMAIN CAGED WHILE SEEKER PERFORMS NULL MANE00075800
C          2    =>    CONTROLS REMAIN CAGED FOR AN ADDITIONAL TIME AFTER C000075900
C                     OF NULL MANEUVER.                                     00076000
C          3    =>    CONTROLS ARE RELEASED TO FOLLOW SEEKER COMMANDS.      00076100
C          *** NOTE ***  DURING A REACQUISITION SEQUENCE, BIAS REMAINS AP00076200
C                        TO PITCH CHANNEL                                   00076300
C                                                                           00076400
C--   SAVE OLD GYRO ANGLES                                                  00076500
      XPGYRO = PGYRO                                                        00076600
      XTGYRO = TGYRO                                                        00076700
      IF (KACQ.EQ.5)   NDROP = 0                                            00077400
C--   IF PROJECTILE IS TRACKING TARGET 2, THEN KOND=3. ITARG=2              00077500
C     AND THE PULSE STATE IS IMPLICITLY "SEE".                              00077600
      IF (KOND.EQ.3)   GO TO 110                                            00077700
      IF (KACQ.GT.2)   GO TO 190                                            00077800
  110 CONTINUE                                                              00077900
C--   LOOK ANGLES                                                           00078000
      DX = XA - SI(1)                                                       00078100
      DY = YA - SI(2)                                                       00078200
      DZ = ZA - SI(3)                                                       00078300
      RANGE = DSQRT( DX*DX + DZ*DZ )                                        00078400
      VH = DSQRT( SI(4)**2 + SI(6)**2 )                                     00078500
      PDOTVH = DX*SI(4) + DZ*SI(6)                                          00078600
      IF (PDOTVH.LT.0.00)   RANGE = -RANGE                                  00078700
      PHLOOK = DATAN2( DX, DZ )                                             00078800
      THLOOK = DATAN2( DY, RANGE )                                          00078900
      GO TO (120,130,290,290,250), KACQ                                     00079000
  120 CONTINUE                                                              00079100
C--   ACQUISITION CANNOT BEGIN UNTIL PROJECTILE IS BELOW CLOUD CEILING.     00079200
      IF (SI(2).GT.CEIL)   GO TO 130                                        00079300
C--   CHECK FOR TARGET IN ACQUISITION RANGE                                 00079400
      CALL AQRANG (PHLOOK, THLOOK, RACQ)                                    00079500
      RSLANT = DSQRT( RANGE**2 + DY**2 )                                    00079600
      IF (RSLANT.GT.RACQ)   GO TO 130                                       00079700
      KACQ = 2                                                              00079800
      NSEEN = 0                                                             00079900
      IF ((JREP.GT.1).AND.(NPRINT.EQ.0))   GO TO 130                        00079950
      TIME = TIMEO + ISTEP*H                                                00080000
      WRITE (6,15) ISTEP, TIME                                              00080100
      WRITE (6,23) (SI(I), I=1,6), XA, ZA, XTGYRD, XPGYRO                   00080200
      IPRINT = 1                                                            00080300
  130 CONTINUE                                                              00080400
      IF (IGLIDE.EQ.1)   GO TO 135                                          00080500
C--   ESTIMATE FUTURE CAGED GYRO POSITION                                   00080600
      THCAGE = BETAP + HN*DBETAP                                            00080700
      PHCAGE = BETAY + HN*DBETAY                                            00080800
      IF (KACQ.NE.4)   THCAGE = THCAGE - CANT                               00080900
  135 CONTINUE                                                              00081000
      IF (KACQ.EQ.2)   GO TO 160                                            00081100
  140 CONTINUE                                                              00081200
```

```
C--  GYRO FREE IF GLIDING
      IF (IGLIDE.EQ.1)   GO TO 150
C--  GYRO IS CAGED
      TGYRO = THCAGE
      PGYRO = PHCAGE
      IF (KACQ.EQ.4)   GO TO 260
  150 CONTINUE
C--  COMPUTE L. O. S. RATE -- CAGEO OR GLIDING
      LOSRY = 0.
      LOSRP = 0.
      RETURN
  160 CONTINUE
C--  CHECK FOR TARGET IN F. O. V.
      IF (MODESM.EQ.1)   GO TO 165
      CALL DENSTY (KSEE, SGLEVL)
      IF (KSEE.EQ.0)   GO TO 140
  165 CONTINUE
      CALL AQSEE (XPGYRO,XTGYRO,PHLOOK, THLOOK, INFOV)
      IF (INFOV.NE.1)   NSEEN = 0
      GO TO (180,140,170), INFOV
  170 CONTINUE
C--  ACQUISITION IMPOSSIBLE
      KACO = 6
      WRITE (6,17) ISTEP
      WRITE (6,23) (S1(I), I=I,6), XA, ZA, XTGYRO, XPGYRO
      RETURN
  180 CONTINUE
C--  SATISFY CONSECUTIVE-PULSE REQUIREMENT FOR ACQUISITION
      NSEEN = NSEEN + 1
      IF (NSEEN.LT.MSEEN)   GO TO 140
C--- ACQUISITION COMMENCES
      KACO = 5
      NOROP = 0
      NULLEO = 0
      TIME = TIMEO + ISTEP*H
      IF (INULL.EQ.0)   TGUIDE = TIME + WAIT
      TSW = TIME
      ISEE = 0
      CALL SWITCH (ISEE, TSW, ISEED2)
      NSW = 1
      IF ((JREP.GT.1).AND.(NPRINT.EQ.0))   GO TO 182
      WRITE (6,16) ISTEP, MSEEN, TIME
      WRITE (6,23) (S1(1), I=1,6), XA, ZA, XTGYRO, XPGYRO
      IPRINT = 1
  182 CONTINUE
      IF (JREP.GT.1)   GO TO 250
      IF (INCR.EO.1)   GO TO 187
      JSTEP = ISTEP
      NO = NRVS
      QPGYRO = PGYRO
      OTGYRO = TGYRO
      OO 185 I=1,12
  185 SO(I) = SI(I)
      CALL OOOGER (3, S1, TGTHOG)
  187 CONTINUE
      IF (MOOESM.EQ.3)   GO TO 250
C--  COMPUTE CROSS-SECTION FOR INTERNAL OR MIXED MODE
      RACQKM = RACQ*.001
      CROSS = THR*RACQ**2/(EFACTR*TRANS(RACQKM))
      GO TO 250
```

```
  190 CONTINUE
C--   (PROJECTILE IS IN A POST-ACQUISITION STATE)
C--   PULSE STATE SWITCH CHECK
      IF (MODESW.EQ.1)   GO TO 199
C--   EXTERNAL GENERATION CHECK -- MODES 2, 3
      JSEE = ISEE
      CALL OENSTY (KSEE, SGLEVL)
      ISEE = KSEE
      IF (ISEE.EQ.JSEE)    GO TO 210
      IF ( ISEE.EQ.0)    DROPS = DROPS + 1.
      IF (NPRINT.EQ.0)    GO TO 210
      TIME = TIMEO + ISTEP*H
      WRITE (6,1) ISEE, ISTEP, TIME
      IPRINT = 1
      GO TO 210
C--   INTERNAL GENERATION CHECK
  199 CONTINUE
      TNOW = ISTEP*H + TIMEO
  200 CONTINUE
      IF (TNOW.LT.TSW)    GO TO 210
      IF (NPRINT.EQ.0)    GO TO 350
      WRITE (6, I8) ISEE, TSW
      IPRINT = I
  350 CONTINUE
      CALL SWITCH (ISEE, TSW, ISEED2)
      NSW = NSW + 1
      IF (ISEE.EQ.0)    DROPS = DROPS + 1.
      GO TO 200
  210 CONTINUE
      IF (ISEE.EQ.0)    NSEEN = 0
      IF (KACQ-4) 21S,280,240
  21S CONTINUE
C--   SEEKER IS IN OROP PHASE 1
      IF (ISEE.EQ.1)    GO TO 110
  220 CONTINUE
C--   GYRO POSITION OOES NOT CHANGE; COMMANOEO CONTROL SURFACE OEFLECTION
C     THE SAME, IE,  THE L. O. S. RATES OO NOT CHANGE.
      NOROP = NOROP + I
      IF (NOROP.LT.MDROP)    RETURN
      KACQ = 4
      GO TO 130
  240 CONTINUE
      IF (ISEE.EQ.1)    GO TO 110
      KACQ = 3
      GO TO 220
  250 CONTINUE
C--   NEW GYRO ANGLES (WITHOUT LIMITING)
      PGYRO = PHLOOK
      TGYRO = THLOOK
C--   CHECK THAT TARGET IS STILL IN F. O. V.
      K = KONO
      KONO = 1TARG
      CALL AQSEE (XPGYRO, XTGYRO, PHLOOK, THLOOK, INFOV)
      KONO = K
      IF (INFOV.EQ.I)    GO TO 260
C--   TARGET HAS SLIPPEO OUT OF FOV
      IF (NPRINT.EQ.0)    GO TO 2SS
      TIME = TIMEO + ISTEP*H
      WRITE (6,2) ISTEP, TIME
      WRITE (6,23) (S1(I), I=1,6), XA, ZA, XTGYRO, XPGYRO
```

```
      IPRINT = 1
  255 CONTINUE
      KACQ = 3
      NDROP = 0
      NSEEN = 0
      PGYRO = XPGYRO
      TGYRO = XTGYRO
      GO TO 220
C--  TARGET IS STILL IN FOV
  260 CONTINUE
C--  COMPUTE DETECTED L. O. S. DEVIATION FROM GYRO AXIS
C--  DETECTED DEVIATIONS ARE INPUT TO CONTROL SURFACE COMMAND
C--  COMPUTE GYRO SLEW RATES
      DEVY = PGYRO - XPGYRO
      DEVP = TGYRO - XTGYRO
      IF (DEVY.GT.+PI)   DEVY = DEVY - PI - PI
      IF (DEVY.LT.-PI)   DEVY = DEVY + PI + PI
      DCOSTG = DCOS(TGYRO)
      ADEVY = DEVY*DCOSTG
      ADEVP = DEVP
      IF (MODESM.EQ.1)   CALL DENSTY (KSEE, SGLEVL)
      M = 1
      IF ((KACQ.EQ.5).AND.(NULL.EQ.3))   M = 2
      CALL LOOKUP (M, ADEVY, SGLEVL, ELOSRY, SLRTY, AMAX, AMIN, 5)
      CALL LOOKUP (M, ADEVP, SGLEVL, ELOSRP, SLRTP, AMAX, AMIN, 5)
      DGYROY = SLRTY*HN
      DGYROP = SLRTP*HN
      PGYRO = XPGYRO + DGYROY/DCOSTG
      TGYRO = XTGYRO + DGYROP
      IF (PGYRO.GT.+PI)   PGYRO = PGYRO - PI - PI
      IF (PGYRO.LT.-PI)   PGYRO = PGYRO + PI + PI
C--  NEW GYRO ANGLES ARE INPUT TO NEXT GUIDANCE ITERATION
      IF (KACQ.EQ.4)   GO TO 150
C--  LOS RATE IS BLANKED DURING AND IMMEDIATELY AFTER NULL MANEUVER.
C--  FOR INULL=0, NULL OCCURS ONLY ONCE AND ON TIME BASIS ONLY.
  262 CONTINUE
      IF (INULL.EQ.1)   GO TO 265
      IF (NULL.EQ.3)   GO TO 277
      TIME = TIME0 + ISTEP*H
      IF (TIME.GE.TGUIDE)   GO TO 275
      GO TO 150
  265 CONTINUE
      GO TO (271,274,277), NULL
  271 CONTINUE
      IF (DSQRT(DGYROY**2+DGYROP**2).GT.DLNULL)   GO TO 150
      NULL = 2
      NULLED = 1
      TNULL = 0.
  274 CONTINUE
      IF (TNULL.GE.WAIT)   GO TO 275
      TNULL = TNULL + HN
      GO TO 150
C--  GUIDANCE COMMENCES
  275 CONTINUE
      NULL = 3
      NULLED = 1
      IF (IBIAS.EQ.0)   BIAS = BIAS2
      IF (NPRINT.EQ.0)   GO TO 277
      TIME = TIME0 + ISTEP*H
      WRITE (6,3) ISTEP, TIME
```

```
  190 CONTINUE
C--  (PROJECTILE IS IN A POST-ACQUISITION STATE)
C--  PULSE STATE SWITCH CHECK
      IF (MODESM.EQ.1)   GO TO 199
C--  EXTERNAL GENERATION CHECK -- MODES 2, 3
      JSEE = ISEE
      CALL DENSTY (KSEE, SGLEVL)
      ISEE = KSEE
      IF (ISEE.EQ.JSEE)    GO TO 210
      IF ( ISEE.EQ.0)    DROPS = DROPS + 1.
      IF (NPRINT.EQ.0)    GO TO 210
      TIME = TIMEO + ISTEP*H
      WRITE (6,1) ISEE, ISTEP, TIME
      IPRINT = 1
      GO TO 210
C--  INTERNAL GENERATION CHECK
  199 CONTINUE
      TNOW = ISTEP*H + TIMEO
  200 CONTINUE
      IF (TNOW.LT.TSW)    GO TO 210
      IF (NPRINT.EQ.0)    GO TO 350
      WRITE (6, 18) ISEE, TSW
      IPRINT = 1
  350 CONTINUE
      CALL SWITCH (ISEE, TSW, ISEE02)
      NSW = NSW + 1
      IF (ISEE.EQ.0)    DROPS = OROPS + 1.
      GO TO 200
  210 CONTINUE
      IF (ISEE.EQ.0)    NSEEN = 0
      IF (KACQ-4) 215,280,240
  215 CONTINUE
C--  SEEKER IS IN DROP PHASE 1
      IF (ISEE.EQ.1)    GO TO 110
  220 CONTINUE
C--  GYRO POSITION DOES NOT CHANGE; COMMANOED CONTROL SURFACE DEFLECTION
C    THE SAME, IE.  THE L. O. S. RATES DO NOT CHANGE.
      NDROP = NDROP + 1
      IF (NDROP.LT.MOROP)    RETURN
      KACQ = 4
      GO TO 130
  240 CONTINUE
      IF (ISEE.EQ.1)    GO TO 110
      KACQ = 3
      GO TO 220
  250 CONTINUE
C--  NEW GYRO ANGLES (WITHOUT LIMITING)
      PGYRO = PHLOOK
      TGYRO = THLOOK
C--  CHECK THAT TARGET IS STILL IN F. O. V.
      K = KONO
      KONO = ITARG
      CALL AQSEE (XPGYRO, XTGYRO, PHLOOK, THLOOK, INFOV)
      KONO = K
      IF (INFOV.EQ.1)    GO TO 260
C--  TARGET HAS SLIPPED OUT OF FOV
      IF (NPRINT.EQ.0)    GO TO 255
      TIME = TIMEO + ISTEP*H
      WRITE (6,2) ISTEP, TIME
      WRITE (6,23) (S1(I), I=1,6), XA, ZA, XTGYRO, XPGYRO
```

```
      IPRINT = 1
  255 CONTINUE
      KACQ = 3
      NDROP = 0
      NSEEN = 0
      PGYRO = XPGYRO
      TGYRO = XTGYRD
      GO TO 220
C--  TARGET IS STILL IN FOV
  260 CONTINUE
C--  COMPUTE DETECTED L. O. S. DEVIATION FROM GYRO AXIS
C--  DETECTED DEVIATIONS ARE INPUT TO CONTROL SURFACE COMMAND
C--  COMPUTE GYRO SLEW RATES
      DEVY = PGYRO - XPGYRO
      DEVP = TGYRO - XTGYRO
      IF (DEVY.GT.+PI)   OEVY = DEVY - PI - PI
      IF (DEVY.LT.-PI)   DEVY = DEVY + PI + PI
      OCOSTG = DCOS(TGYRO)
      ADEVY = DEVY*DCOSTG
      ADEVP = DEVP
      IF (MODESM.EQ.I)   CALL OENSTY (KSEE, SGLEVL)
      M = I
      IF ((KACQ.EQ.5).AND.(NULL.EQ.3))   M = 2
      CALL LODKUP (M, ADEVY, SGLEVL, ELOSRY, SLRTY, AMAX, AMIN, 5)
      CALL LOOKUP (M, ADEVP, SGLEVL, ELOSRP, SLRTP, AMAX, AMIN, 5)
      OGYROY = SLRTY*HN
      DGYROP = SLRTP*HN
      PGYRD = XPGYRO + DGYROY/DCOSTG
      TGYRO = XTGYRO + DGYROP
      IF (PGYRO.GT.+PI)   PGYRO = PGYRO - PI - PI
      IF (PGYRO.LT.-PI)   PGYRO = PGYRO + PI + PI
C--  NEW GYRD ANGLES ARE INPUT TO NEXT GUIDANCE ITERATION
      IF (KACQ.EQ.4)   GO TO 150
C--  LOS RATE IS BLANKED DURING AND IMMEDIATELY AFTER NULL MANEUVER.
C--  FOR INULL=0, NULL OCCURS ONLY ONCE AND ON TIME BASIS ONLY.
  262 CONTINUE
      IF (INULL.EQ.1)   GO TO 265
      IF (NULL.EQ.3)   GO TO 277
      TIME = TIMEO + ISTEP*H
      IF (TIME.GE.TGUIDE)   GO TO 275
      GO TO 150
  265 CONTINUE
      GO TO (271,274,277), NULL
  271 CONTINUE
      IF (DSQRT(DGYROY**2+DGYRDP**2).GT.DLNULL)   GD TO 150
      NULL = 2
      NULLED = 1
      TNULL = 0.
  274 CONTINUE
      IF (TNULL.GE.WAIT)   GO TD 275
      TNULL = TNULL + HN
      GO TO 150
C--  GUIDANCE COMMENCES
  275 CONTINUE
      NULL = 3
      NULLED = 1
      IF (IBIAS.EQ.0)   BIAS = BIAS2
      IF (NPRINT.EQ.0)   GO TO 277
      TIME = TIMEO + ISTEP*H
      WRITE (6,3) ISTEP, TIME
```

```
      WRITE (6,23) (S1(I), I=1,6), XA, ZA, XTGYRO, XPGYRO               00098600
      IPRINT = 1                                                        00098700
  277 CONTINUE                                                          00098800
C--  GENERATE CONTROL SURFACE COMMAND (L. O. S. RATE)                    00098900
      LOSRY = ELOSRY                                                    00099000
      LOSRP = ELOSRP                                                    00099100
      RETURN                                                            00099200
  280 CONTINUE                                                          00099300
C--  SEEKER IS IN DROP PHASE 2                                          00099400
      IF (ISEE.EQ.1)   GO TO 110                                        00099500
C--  GYRO WILL RETURN TO CAGED POSITION, LOS RATES TO ZERO.             00099600
C    BIAS REMAINS APPLIED TO PITCH CHANNEL.                             00099700
      NDROP = NDROP + 1                                                 00099800
      GO TO 130                                                         00099900
  290 CONTINUE                                                          00100000
C--  ATTEMPT TO REACQUIRE                                               00100100
      K = KOND                                                          00100200
      KOND = ITARG                                                      00100300
      CALL AQSEE (XPGYRO, XTGYRO, PHLOOK, THLOOK, INFOV)                00100400
      KOND = K                                                          00100500
      GO TO (310,300,300), INFOV                                        00100600
  300 CONTINUE                                                          00100700
      IF (KACQ.EQ.3)   GO TO 220                                        00100800
      NDROP = NDROP + 1                                                 00100900
      GO TO 130                                                         00101000
  310 CONTINUE                                                          00101100
      IF (KACQ.NE.3)   GO TO 315                                        00101200
      KACQ = 5                                                          00101300
      IF (NULLED.EQ.0)   GO TO 262                                      00101350
      GO TO 275                                                         00101400
  315 CONTINUE                                                          00101500
C--  SATISFY CONSECUTIVE-PULSE REQUIREMENT FOR REACQUISITION             00101600
      NSEEN = NSEEN + 1                                                 00101700
      IF (NSEEN.GE.MSEEN)   GO TO 320                                   00101800
      IF (KACQ.EQ.4)   GO TO 130                                        00101900
      RETURN                                                            00102000
  320 CONTINUE                                                          00102100
C--  REACQUISITION                                                      00102200
      KACQ = 5                                                          00102300
      NULL = 1                                                          00102400
      TIME = TIME0 + ISTEP*H                                            00102420
      IF (INULL.EQ.1)   GO TO 330                                       00102440
      IF (NULLED.EQ.0)   TGUIDE = TIME + WAIT                           00102460
  330 CONTINUE                                                          00102480
      IF (NPRINT.EQ.0)   GO TO 250                                      00102500
      WRITE (6,19) ISTEP, NDROP, TIME                                   00102700
      WRITE (6,23) (S1(I), I=1,6), XA, ZA, XTGYRO, XPGYRO               00102800
      IPRINT = 1                                                        00102900
      GO TO 250                                                         00103000
      END                                                               00103100
C--  DODGER  --  DODGER  --  DODGER  --  DODGER  --  DODGER  --  DODGER 00103200
      SUBROUTINE DODGER (KALL, S1, TGTHDG)                              00103300
      REAL*8 XT, YT, ZT, XS, YS, ZS, P1, KDEL, KDEL1, KDEL2, HN, XCA,   00103400
     1 YCA, ZCA, RMISS, YMISS, PMISS, QKD, XP, YP, ZP, VXP, VYP, VZP,   00103500
     2 SDX, SDY, SDZ, H, S1(12),DSQRT,GROUND,   FREQ, PRF               00103600
      REAL*8 PHGYRO, THGYRO, PHLOOK, THLOOK                             00103700
      REAL*8 FOV(2), BRIGHT, TIME0, XSS, YSS, ZSS                       00103800
      REAL*8 TGTHDG,  PGYRO, TGYRO, CANT, BIAS, BIAS2, DLAMDY, DLAMDP,  00103900
     1     MXDLDS, DROPS                                                00104000
      DIMENSION PAR(10,4), MANVRS(2), MSEQ(2), MTYPE(2), X(2), Y(2),    00104100
```

```
     1 Z(2), VEL(2), THETA(2), EX(2), EY(2), EZ(2), EVEL(2), DTHETA(2),
     2 T(2), VX(2), VZ(2), TEND(2), X0(2), Z0(2), VX0(2), VZ0(2), AX(2),
     3 AZ(2), PHI(2), XC(2), ZC(2), R(2), XX(2), ZZ(2), VXX(2), VZZ(2),
     4 THETAF(2), QHT(2), QWS(2), QT(2), QTE(2), QX(2), QZ(2), QVX(2),
     S QVZ(2), QW(2), QTH(2), Q1(2), Q2(2), Q3(2), Q4(2), Q5(2), Q6(2)
      COMMON / BLK1 / XS, YS, ZS,                  HN,      GROUND
      COMMON / BLK2 / XT, YT, ZT, KDEL, KDEL1, KDEL2, SLRNG1, SLRNG2
      COMMON / BLK3 / XCA, YCA, ZCA, RMISS, YMISS, PMISS, VXR, VZR
      COMMON / BLK4 / H, P1, PRF, MEASUR
      COMMON / BLKS / PGYRO, TGYPO, CANT, BIAS, BIAS2, DLAMDY, DLAMOP,
     1          DROPS, KACQ, NRVS, IPRINT, ISTEP, NPRINT, JSTEP, NO
      COMMON / BLK8 / FOV, MODE, KONO, ITARG, NSW
      COMMON / BLK12 / BPIGHT, TIMEO, NULL, IVIS, INCR
      OATA FOOT, AMP / 3.2808, S.066 /
C
C
C
    1 FORMAT (8F10.0)
    2 FORMAT (8I10)
    3 FORMAT ( 66(' -')// T53, 'TARGET MOTION SPECIFICATIONS'// ' TARGET
     1 1 IS THE APPARENT TARGET CREATEO BY THE LASER SPOT.  (X(1), Y(1),
     2 Z(I)) IS THE POINT OF REFERENCE FOR SPOT POSITION.'/ T7, 'WITH IN
     3TERNAL SPOT MOTION GENERATION, IT IS THE CENTROID OF SPOT POSITION
     4S; WITH EXTERNAL GENERATION, IT IS THE NOMINAL DES-')
    4 FORMAT ( T7, 'IGNATION POINT, NOT NECESSARILY THE TRUE CENTROID','
     1.  IN EITHER CASE, THE BELOW REFER TO THE DETERMINISTIC ANO NOT TH
     2E'/ T7, 'RANOOM COMPONENT OF MOTION.'// T7, 'POSITION', T32, 'X(1)
     3', T60, F10.3, ' M', T90, F10.3, ' FT' )
    S FORMAT ( T12, 'AFTER CONTINUING ', F5.2, ' SEC, ACCELERATE/OECELER
     1ATE AT ', FS.2, ' M/SEC/SEC TO ', FS.2, ' M/SEC.' )
    6 FORMAT ( T12, 'AFTER CONTINUING ', FS.2, ' SEC, TURN LEFT ', F6.1,
     1 ' OEG WITH RAOIUS ', F6.2, ' M.' )
    7 FORMAT ( T12, 'AFTER CONTINUING ', FS.2, ' SEC, PERFORM ZIGZAG ',
     1 F3.1, ' TIMES.' )
    8 FORMAT ( T32, 'Y(1) (ABOVE GROUND LEVEL)', T60, F10.3, ' M', T90,
     1 F10.3, ' FT'/ T32, 'Z(1)', T60, F10.3, ' M', T90, F10.3, ' FT'//
     2 T7, 'VELOCITY', T32, 'VEL(1)', T60, F10.3, ' M/SEC', T90, F10.3,
     3 ' FT/SEC'/ T7, 'OIRECTION', T32, 'THETA(1)', T60, F10.3, ' OEG CC
     4W FROM +Z AXIS'// T7, 'TARGET 1 PERFORMS ', I1, ' MANEUVERS -' )
    9 FORMAT ( // ' TARGET 2 IS THE POINT OF AIM WITH RESPECT TO WHICH M
     1ISS OISTANCES ARE COMPUTEO.  IN THE BIFUNCTIONAL MOOE OF OPERATION
     2 THE'/ T7, 'PROJECTILE HOMES ON TARGET 2 AFTER APPROACHING WITHIN
     3THE TRANSITION RANGE.  THERE IS NO RANOOM COMPONENT OF MOTION.'//)
   10 FORMAT ( T7, 'POSITION', T32, 'X(2)', T60, F10.3, ' M', T90, F10.3
     1 , ' FT'/ T32, 'Y(2) (ABOVE GROUNO LEVEL)', T60, F10.3, ' M', T90,
     2 F10.3, ' FT'/ T32, 'Z(2)', T60, F10.3, ' M', T90, F10.3, ' FT'//
     3 T7, 'VELOCITY', T32, 'VEL(2)', T60, F10.3, ' M/SEC', T90, F10.3,
     4 ' FT/SEC' )
   11 FORMAT ( T7, 'DIRECTION', T32, 'THETA(2)', T60, F10.3, ' OEG CCW F
     1ROM +Z AXIS'// T7, 'TARGET 2 PERFORMS ', I1, ' MANEUVERS -' )
   12 FORMAT ( T7, 'THE TRANSITION RANGE IS ', F10.4, ' M OR ', F10.4,
     1' FT FROM TARGET 2.'// 66(' -')/ )
   13 FORMAT ('0*** AT STEP ', I4, ' REMAINING SLANT RANGE TO TARGET 2 I
     1S ', F10.4, ' METERS.' )
   14 FORMAT ('0*** AT STEP ', I4, ' TARGET 2 IS IN INFRAREO FIELO OF VI
     1EW.  TRANSITION TO PASSIVE TRACKING OCCURS WITH STATE:'/ ' PROJ X'
     2, F10.3, ' Y', F10.3, ' Z', F10.3, ' * VX', F10.3, ' VY', F10.3,
     3 ' VZ', F10.3, ' * TGT X', F10.3, ' Z', F10.3 /T90, 'TIME ',
     4 F10.3)
   1S FORMAT ('0*** AT STEP ', I4, ' TARGET ', I1, ' GOES FROM MANEUVER
```

```
     1TYPE ', I1, ' TO MANEUVER TYPE ', I1, ' WITH STATE:'/ 10X, 'X=',  00110200
     2 F10.3, '  Z=', F10.3, '  VEL=', F10.3, '  THETA=', F10.3, ' DEG'/00110300
     3 T90, 'TIME ', F10.3)                                            00110400
   16 FORMAT ('0*** TIME OF CLOSEST APPROACH = ', F8.4, ' SECONDS.')   00110500
   17 FORMAT ('0TARGET POSITION INCREMENT PER SHOT  X', T60, F10.4, ' M'00110600
     1 , T90, F10.4, ' FT'/ T37, 'Z', T60, F10.4, T90, F10.4/)         00110700
C                                                                      00110800
C                                                                      00110900
C--  DODGER CONTROLS THE MOTION OF THE TWO TARGETS.                     00111000
C                                                                      00111100
      XP = S1(1)                                                       00111200
      YP = S1(2)                                                       00111300
      ZP = S1(3)                                                       00111400
      VXP = S1(4)                                                      00111500
      VYP = S1(5)                                                      00111600
      VZP = S1(6)                                                      00111700
      GO TO (100,200,300,400,500), KALL                                00111800
C                                                                      00111900
C--  1    11   1    1    1    1    1    1    1    1    1    1    1    1 00112000
C                                                                      00112100
  100 CONTINUE                                                         00112200
C--  KALL=1 => INPUT                                                   00112300
      READ (5,1) X(1), Y(1), Z(1), VEL(1), THETA(1), XINCR, ZINCR      00112400
      READ(5,1) X(2), Y(2), Z(2), VEL(2), THETA(2),       RSWICH,XINCH00112500
C--  IF XINCH IS 1.0, TARGET POSITIONS ARE GIVEN IN INCHES.  IN ANY CASE00112600
C    VELOCITY IS IN M/SEC WHEN MEASUR=1, OR FT/SEC WHEN MEASUR=2 .     00112700
      INCR = 1                                                         00112730
      IF ((XINCR.EQ.0.).AND.(ZINCR.EQ.0.))    INCR = 0                 00112760
      IF (XINCH.NE.1.)    GO TO 110                                    00112800
      IF (MEASUR.GT.1)    GO TO 105                                    00112900
C--  CONVERT TARGET POSITIONS FROM INCHES TO METERS                     00113000
      DO 101 I=1,2                                                     00113100
      X(I) = X(I) * .0254                                              00113200
      Y(I) = Y(I) * .0254                                              00113300
  101 Z(I) = Z(I) * .0254                                              00113400
      GO TO 110                                                        00113500
  105 CONTINUE                                                         00113600
C--  CONVERT TARGET POSITIONS FROM INCHES TO FEET                       00113700
      DO 106 I=1,2                                                     00113800
      X(I) = X(I) * .0833                                              00113900
      Y(I) = Y(I) * .0833                                              00114000
  106 Z(I) = Z(I) * .0833                                              00114100
  110 CONTINUE                                                         00114200
      READ (5,2) MANVRS                                                00114300
      IF (MEASUR.GT.1)    GO TO 130                                    00114400
C--  MEASUR=1  =>  METRIC UNITS                                        00114500
      DO 120 I=1,2                                                     00114600
      EX(I) = X(I)*FOOT                                                00114700
      EY(I) = Y(I)*FOOT                                                00114800
      EZ(I) = Z(I)*FOOT                                                00114900
      EVEL(I) = VEL(I)*FOOT                                            00115000
  120 CONTINUE                                                         00115100
      ERSW = RSWICH*FOOT                                               00115200
      EXINCR = XINCR*FOOT                                              00115300
      EZINCR = ZINCR*FOOT                                              00115400
      GO TO 150                                                        00115500
  130 CONTINUE                                                         00115600
C--  MEASUR=2  =>  ENGLISH UNITS                                       00115700
      DO 140 I=1,2                                                     00115800
      EX(I) = X(I)                                                     00115900
```

```
      EY(I) = Y(I)
      EZ(I) = Z(I)
      EVEL(I) = VEL(I)
      X(1) = X(I)/FOOT
      Y(I) = Y(1)/FOOT
      Z(1) = Z(1)/FOOT
      VEL(I) = VEL(I)/FOOT
  140 CONTINUE
      ERSW = RS*ICH
      RS*ICH = RS*ICH/FOOT
      EXINCR = XINCR
      EZINCR = ZINCR
      XINCR = XINCR/FOOT
      ZINCR = ZINCR/FOOT
  150 CONTINUE
      WRITE (6,3)
      WRITE (6,4) X(I), EX(1)
      WRITE (6,8) Y(I), EY(1), Z(I), EZ(1), VEL(I), EVEL(I), THETA(I),
     1          MANVRS(1)
      MI = MANVRS(1) + I
      M2 = MANVRS(I) + MANVRS(2)
      IF (MANVRS(2).EQ.0)    M2 = M2 + 1
      DO I60 I=I,M2
      IF (1.NE.M1)    GO TO 155
      WRITE (6,9)
      WRITE (6,I0) X(2), EX(2), Y(2), EY(2), Z(2), EZ(2), VEL(2),
     I         EVEL(2)
      WRITE (6,11) THETA(2), MANVRS(2)
      IF (MANVRS(2).EQ.0)    GO TO 160
  155 CONTINUE
      READ (5,1)  (PAR(1,J), J=1,4)
      IF (PAR(1,2).EQ.2.00)    WRITE (6,5) PAR(I,I), PAR(I,3), PAR(1,4)
      IF (PAR(I,2).EQ.3.00)    WRITE (6,6) PAR(1,I), PAR(I,3), PAR(I,4)
      IF (PAR(1,2).EQ.4.00)    WRITE (6,7) PAR(1,1), PAR(1,3)
  160 CONTINUE
      WRITE (6,17) XINCR, EXINCR, ZINCR, EZINCR
      WRITE (6,I2) RSWICH, ERSW
      MSEQ(I) = 1
      MSEQ(2) = M1
      XT = X(2)
      YT = Y(2)
      ZT = Z(2)
      HALFPI = 0.5*PI
      OMEGA = 0.2*PI
      RADIAN = PI/I80.
      KOND = I
      ITARG = 1
      KDEL = KDELI
      DO 170 1=I,2
      THETA(1) = THETA(I)*RADIAN
      T(I) = -HN
      MTYPE(I) = I
      VZ(I) = VEL(1)*COS(THETA(1))
      VX(1) = VEL(I) *SIN(THETA(I))
      TEND(1) = I000.
      IF (MANVRS(I).GT.0)    TEND(1) = PAR(MSEQ(1), 1)
      X(1) = X(I) - HN*VX(I)
      Z(I) = Z(I) - HN*VZ(1)
  I70 CONTINUE
      TGTHDG = THETA(1)
```

```
      RETURN
C
C--  2    2    2    2    2    2    2    2    2    2    2    2    2    2
C
  200 CONTINUE
C--  KALL=2   =>   TARGET MOTION
      DO 275 I = ITARG, 2
  205 CONTINUE
      T(I) = T(I) + HN
      IF (T(I) .GT. TEND(I))   GO TO 230
C--  MANEUVER CONTINUING
      MT = MTYPE(I)
      GO TO (210,215,220,225), MT
  210 CONTINUE
C--  CONTINUE STRAIGHT RUN
      X(I) = X(I) + HN*VX(I)
      Z(I) = Z(I) + HN*VZ(I)
      GO TO 275
  215 CONTINUE
C--  ACCELERATION CONTINUING
      X(I) = X0(I) + T(I)*( VX0(I) + 0.5*AX(I)*T(I) )
      Z(I) = Z0(I) + T(I)*( VZ0(I) + 0.5*AZ(I)*T(I) )
      VX(I) = VX0(I) + AX(I)*T(I)
      VZ(I) = VZ0(I) + AZ(I)*T(I)
      GO TO 275
  220 CONTINUE
C--  TURN CONTINUING
      THETA(I) = THETA(I) + DTHETA(I)
      PHI(I) = PHI(I) + DTHETA(I)
      Z(I) = ZC(I) + R(I)*COS(PHI(I))
      X(I) = XC(I) + R(I)*SIN(PHI(I))
      VZ(I) = VEL(I)*COS(THETA(I))
      VX(I) = VEL(I)*SIN(THETA(I))
      GO TO 275
  225 CONTINUE
C--  ZIGZAG CONTINUING
      XX(I) = XX(I) + HN*VXX(I)
      ZZ(I) = ZZ(I) + HN*VZZ(I)
      SWT = SIN(OMEGA*T(I))
      HOLD = X(I)
      X(I) = XX(I) + AX(I)*SWT
      VX(I) = (X(I)-HOLD)/HN
      HOLD = Z(I)
      Z(I) = ZZ(I) + AZ(I)*SWT
      VZ(I) = (Z(I)-HOLD)/HN
      GO TO 275
  230 CONTINUE
C--  CHANGE MANEUVERS
      T(I) = 0.
      TIME = TIME0 + ISTEP*H
      MT = MTYPE(I)
      GO TO (235,255,260,265), MT
  235 CONTINUE
C--  CHANGING FROM STRAIGHT RUN
      MTYPE (I) = PAR(MSEQ(I), 2)
      IF (NPRINT.EQ.0)   GO TO 236
      OTHETA = THETA(I) / RADIAN
      WRITE (6,15) ISTEP, I, MT, MTYPE(I), X(I), Z(I), VEL(I), OTHETA
     I  , TIME
      IPRINT = I
```

```
  235 CONTINUE
      IF (MTYPE(I) - 3)   240,245,250
  240 CONTINUE
C--  CHANGING TO ACCELERATION
      DELV = PAR(MSEQ(I), 4) - VEL(I)
      ACCEL = PAR(MSEQ(I), 3)
      IF (DELV.LT.0.)   ACCEL = -ACCEL
      TEND(I) = DELV/ACCEL
      X0(I) = X(I)
      Z0(I) = Z(I)
      VX0(I) = VX(I)
      VZ0(I) = VZ(I)
      AX(I) = ACCEL*SIN(THETA(I))
      AZ(I) = ACCEL*COS(THETA(I))
      GO TO 205
  245 CONTINUE
C--  CHANGING TO TURN
      ANGLE = PAR(MSEQ(I), 3) * RADIAN
      R(I) = PAR(MSEQ(I), 4)
      SIGN = 1.
      IF (ANGLE.LT.0.)   SIGN = -1.
      XC(I) = X(I) + SIGN*R(I)*COS(THETA(I))
      ZC(I) = Z(I) - SIGN*R(I)*SIN(THETA(I))
      PHI(I) = THETA(I) - SIGN*HALFPI
      DTHETA(I) = SIGN*HN*VEL(I)/R(I)
      TEND(I) = SIGN*ANGLE*R(I)/VEL(I)
      THETAF(I) = THETA(I) + ANGLE
      GO TO 205
  250 CONTINUE
C--  CHANGING TO ZIGZAG
      XX(I) = X(I)
      ZZ(I) = Z(I)
      VXX(I) = VX(I)
      VZZ(I) = VZ(I)
      TEND(I) = PAR(MSEQ(I), 3) * 10.
      AZ(I) = -AMP*SIN(THETA(I))
      AX(I) =  AMP*COS(THETA(I))
      GO TO 205
  255 CONTINUE
C--  FROM ACCELERATION TO STRAIGHT RUN
      VX(I) = VX0(I) + AX(I)*TEND(I)
      VZ(I) = VZ0(I) + AZ(I)*TEND(I)
      VEL(I) = PAR(MSEQ(I), 4)
      GO TO 270
  260 CONTINUE
C--  FROM TURN
      THETA(I) = THETAF(I)
      VZ(I) = VEL(I) * COS(THETA(I))
      VX(I) = VEL(I) * SIN(THETA(I))
      GO TO 270
  265 CONTINUE
C--  FROM ZIGZAG
      VX(I) = VX0(I)
      VZ(I) = VZ0(I)
  270 CONTINUE
C--  SET UP STRAIGHT RUN
      MTYPE(I) = 1
      IF (NPRINT.EQ.0)   GO TO 271
      OTHETA = THETA(I) / RADIAN
     WRITE (6,15) ISTEP, I, MT, MTYPE(I), X(I), Z(I), VEL(I), OTHETA
```

```
     1  + TIME
        IPRINT = 1
  271 CONTINUE
        MSEQ(I) = MSEQ(I) + 1
        TEND(I) = 1000.
        MMM = MANVRS(1)
        IF (I.EQ.2)    MMM = MMM + MANVRS(2)
        IF (MSEQ(I).LE.MMM)    TEND(I) = PAR(MSEQ(I), 1)
        GO TO 205
  275 CONTINUE
        CALL SPOTMO (1, XSS, YSS, ZSS)
        XS = XSS
        YS = YSS
        ZS = ZSS
        SLRNG1 = DSQRT((XP-X(1)-XS)**2 + (YP-Y(1)-YS)**2 +(ZP-Z(1)-ZS)**2)
        SLRNG2= DSQRT((XP-X(2))**2 + (YP-Y(2))**2 + (ZP-Z(2))**2 )
        IF (ITARG.NE.1)    GO TO 295
        IF (KOND.EQ.2)    GO TO 291
        IF(SLRNG2.LT.RS*ICH)    GO TO 290
  280 CONTINUE
        XT = X(1) + XS
        YT = Y(1) + YS
        ZT = Z(1) + ZS
        RETURN
  290 CONTINUE
C--   PROJECTILE IS IN TRANSITION RANGE AND WILL SWITCH TO PASSIVE HOMING
C     TARGET 2 IS WITHIN INFRARED FIELD OF VIEW.
        KOND = 2
        IF (NPRINT.GT.0)    WRITE (6,13)    ISTEP, SLRNG2
        IF (NPRINT.GT.0)    IPRINT = 1
  291 CONTINUE
        DX = X(2) - XP
        DY = Y(2) - YP
        DZ = Z(2) - ZP
        DH = SQRT( DX**2 + DZ**2 )
        PHLOOK = ATAN2(DX, DZ)
        THLOOK = ATAN2(DY, DH)
        PHGYRO = PGYRO
        THGYRO = TGYRO
        CALL AQSEE (PHGYRO, THGYRO, PHLOOK, THLOOK, INFOV)
        IF (INFOV.NE.1)    GO TO 280
C--   TARGET IS IN I.R. F.O.V.  TRANSITION OCCURS.
        KACQ = 5
        KOND = 3
        NULL = 1
        IF (NPRINT.GT.0)    WRITE(6,14)  ISTEP, (SI(I), I=1,6),X(2), Z(2)
        IF (NPRINT.GT.0)    IPRINT = 1
        ITARG = 2
        YT = Y(2)
        KDEL = KDEL2
  295 CONTINUE
        XT = X(2)
        ZT = Z(2)
        RETURN
C
C--  3    3    3    3    3    3    3    3    3    3    3    3    3    3
C
  300 CONTINUE
C--   KALL=3   =>   SAVE ACQUISITION CONDITIONS
        KONDQ = KOND
```

```
      ITQ = ITARG
      QKO = KOEL
      QSLR1 = SLRNG1
      QSLR2 = SLRNG2
      DO 340 I=1,2
      QMT(I) = MTYPE(I)
      QMS(I) = MSEQ(I)
      QT(I) = T(I)
      QTE(I) = TEND(I)
      QX(I) = X(I)
      QZ(I) = Z(I)
      QVX(I) = VX(I)
      QVZ(I) = VZ(I)
      QV(I) = VEL(I)
      QTH(I) = THETA(I)
      MT = MTYPE(1)
      GO TO (340,310,320,330), MT
  310 CONTINUE
      Q1(I) = X0(I)
      Q2(I) = Z0(I)
      Q3(I) = VX0(I)
      Q4(I) = VZ0(I)
      Q5(I) = AX(I)
      Q6(I) = AZ(I)
      GO TO 340
  320 CONTINUE
      Q1(I) = R(I)
      Q2(I) = XC(I)
      Q3(I) = ZC(I)
      Q4(I) = PHI(I)
      Q5(I) = OTHETA(I)
      Q6(I) = THETAF(I)
      GO TO 340
  330 CONTINUE
      Q1(I) = XX(I)
      Q2(I) = ZZ(I)
      Q3(I) = VXX(I)
      Q4(I) = VZZ(I)
      Q5(I) = AX(I)
      Q6(I) = AZ(I)
  340 CONTINUE
      RETURN
C
C--  4    4    4    4    4    4    4    4    4    4    4    4    4    4
C
  400 CONTINUE
C--  KALL=4    =>   LOAO ACQUISITION CONOITIONS
      KONO = KONDQ
      ITARG = ITQ
      KOEL = QKO
      SLRNG1 = QSLR1
      SLRNG2 = QSLR2
      DO 440 I=1,2
      MTYPE(I) = QMT(I)
      MSEQ(I) = QMS(I)
      T(I) = QT(I)
      TEND(I) = QTE(I)
      QX(I) = QX(I) + XINCR
      QZ(I) = QZ(I) + ZINCR
      X(I) = QX(I)
```

```
      Z(I) = QZ(I)
      VX(I) = QVX(I)
      VZ(I) = QVZ(I)
      VEL(I) = QV(I)
      THETA(I) = QT-(I)
      MT = MTYPE(I)
      GO TO (440,410,420,430), MT
  410 CONTINUE
      Q1(I) = Q1(I) + XINCR
      Q2(I) = Q2(I) + ZINCR
      X0(I) = Q1(I)
      Z0(I) = Q2(I)
      VX0(I) = Q3(I)
      VZ0(I) = Q4(I)
      AX(I) = Q5(I)
      AZ(I) = Q6(I)
      GO TO 440
  420 CONTINUE
      Q2(I) = Q2(I) + XINCR
      Q3(I) = Q3(I) + ZINCR
      R(I) = Q1(I)
      XC(I) = Q2(I)
      ZC(I) = Q3(I)
      PHI(I) = Q4(I)
      DTHETA(I) = Q5(I)
      THETAF(I) = Q6(I)
      GO TO 440
  430 CONTINUE
      QI(I) = QI(I) + XINCR
      Q2(I) = Q2(I) + ZINCR
      XX(I) = Q1(I)
      ZZ(I) = Q2(I)
      VXX(I) = Q3(I)
      VZZ(I) = Q4(I)
      AX(I) = QS(I)
      AZ(I) = Q6(I)
  440 CONTINUE
      RETURN
C
C--  5    5    5    5    5    5    5    S    S    S    5    5    5    5
C
  500 CONTINUE
C--  KALL=5   =>   COMPUTE CLOSEST APPROACH
      XR = XP - X(2)
      YR = YP - Y(2)
      ZR = ZP-Z(2)
      VXR = VXP - VX(2)
      VYR = VYP
      VZR = VZP - VZ(2)
      VSQ =  VXR**2 + VYR**2 + VZR**2
      XCA = ( XR*( VYR**2 + VZR**2 ) - VXR*( VYR*YR + VZR*ZR ) ) / VSQ
      YCA = ( YR*( VZR**2 + VXR**2 ) - VYR*( VZR*ZR + VXR*XR ) ) / VSQ
      ZCA = ( ZR*( VXR**2 + VYR**2 ) - VZR*( VXR*XR + VYR*YR ) ) / VSQ
      VP  = SQRT(VSQ)
      VH = SQRT(VSQ-VYR**2)
      RMISS =DSQRT( XCA**2 + YCA**2 + ZCA**2 )
      YMISS =(VXR*ZCA - VZR*XCA) / VH
      PMISS = (-VXR*VYR*XCA + VH**2 * YCA - VYR*VZR*ZCA) / (VH*VP)
      DIST =DSQRT( (XR-XCA)**2 + (YR-YCA)**2 + (ZR-ZCA)**2 )
      TT= DIST/VP
```

```
      PDOTV = XR*,XR + YR*,YR + ZR*,ZR
      IF (PDOTV.GT.0.)  TT = -TT
      XT = X(2) + VX(2)*TT
      ZT = Z(2) + VZ(2)*TT
      TT = TIMEO + ISTEP*H + TT
      IF (NPRINT.GT.0)   WRITE (6,16) TT
      TGTHDG = THETA(2)/RADIAN
      RETURN
      END
C--   AQDATA  --  AQDATA  --  AQDATA  --  AQDATA  --  AQDATA  --  AQDATA
      SUBROUTINE AQDATA (KAM)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 LP, LRT, MILE, KAM
      REAL*4 TRANS, SDR, VAPOR, HRANGE, PRANGE, SERRIR
      COMMON/BLK4/ H, PI, PRF, MEASUR
      COMMON / BLK7 / PHVCN, THVCN, LRT, B, CEIL, DR
      COMMON / BLK8 / FOV(2), MODESM, KOND, ITARG, NSM
      COMMON / BLK13 / DRATIO, EFACTR, E, THR
      COMMON / BLK15 / VAPOR, HRANGE, PRANGE
      COMMON / BLK18 / SERRIR, IRSEED, LENGT3, NRVS3
      DATA MILE, FOOT / .62137D0, 3.2808D0 /
C
    1 FORMAT(8F10.0)
    2 FORMAT ( ' POLAR ANGLES OF NORMAL TO TARGET SURFACE:  THETA', T60,
     1 F10.4, ' DEG FROM +Y AXIS'/ T45, 'PHI', T60, F10.4, ' DEG FROM +Z
     2 AXIS'/ ' TARGET REFLECTIVITY', T60, F10.4/ ' LASER PULSE ENERGY',
     3 T60, F10.4, ' JOULES')
    3 FORMAT ( ' DETECTOR THRESHOLD', T60, F10.4, '*10**(-15) J/CM**2/PU
     1LSE'/ ' VISIBILITY RANGE', T60, F10.4, ' KM', T90, F10.4, ' MI'/
     2 ' PRECIPITABLE WATER VAPOR CODE', T60, F10.4/ T12, 'NOTE:  2 =>
     37.84 MM/KM,  3 => 1.87 MM/KM'/ ' CLOUD CEILING (ABOVE GROUND LEVEL
     4)', T60, F10.4, ' M', T90, F10.4, ' FT'/ )
    4 FORMAT (' BEAM DIVERGENCE', T60, F10.4, ' MILLIRADIANS' / ' SEEKER
     1 DYNAMIC RANGE', T60, F10.4, ' DECIBELS' / ' VISIBILITY RANGE DUE
     2TO PARTICULATE ONLY', T60, F10.4, ' KM', T90, F10.4, ' MI' / )
    5 FORMAT(T5, 'ERROR -- PROGRAM ABORTED DUE TO INVALID KODE.' )
    9 FORMAT ('0PASSIVE IR MODE ANGULAR WHITE-NOISE STD. DEV. IS', F10.4
     1, ' MILLIRADIANS'/' RANDOM NUMBER SEED FOR ANGULAR NOISE IS ',I10/
     2 ' BLOCK LENGTH IS ', I10/)
C
C--   AQDATA READS IN ACQUISITION MODEL DATA AND PERFORMS PRELIMINARY CAL
C
      READ (5,1) THVCN, PHVCN, LP, THR, REFL, VR, CODE, CEIL
C--   THVCN IN DEGREES COLATITUDE FROM Y-AXIS; PHVCN IN DEGREES AZIMUTH (
C     ND-RULE) FROM Z-AXIS; AR, DR, VR IN KM; LP IN J/PULSE.  THR IN UNIT
C     10**-15 J/SQ CM/PULSE.
      IF (MEASUR.EQ.2)   GO TO 50
C--   MEASUR=1  =>  METRIC UNITS
      EVR = VR*MILE
      ECEIL = CEIL*FOOT
      GO TO 60
   50 CONTINUE
C--   MEASUR=2  =>  ENGLISH UNITS
      EVR = VR
      ECEIL = CEIL
      VR = VR/MILE
      CEIL = CEIL/FOOT
   60 CONTINUE
      WRITE (6,2) THVCN, PHVCN, REFL, LP
      WRITE (6,3)THR, VR, EVR, CODE, CEIL, ECEIL
```

```
C--  CONVERT ANGLES TO RADIANS                                          00157400
      RADIAN = PI/180.                                                  00157500
      THVCN = THVCN*RADIAN                                              00157600
      PHVCN = PHVCN*RADIAN                                              00157700
      KODE = CODE                                                       00157800
      LRT = LP*REFL/THR                                                 00157900
      HRANGE = VR                                                       00158000
      READ (5,1) BMDIVG, DYRANG, PRANGE, KAM, SERRIR, SEEDIR, RLNGT3    00158100
      IRSEED = SEEDIR                                                   00158130
      LENGT3 = RLNGT3                                                   00158160
      IF (MEASUR.EQ.2)   GO TO 65                                       00158200
      EPRANG = PRANGE*MILE                                              00158300
      GO TO 67                                                          00158400
   65 CONTINUE                                                          00158500
      EPRANG = PRANGE                                                   00158600
      PRANGE = PRANGE/MILE                                              00158700
   67 CONTINUE                                                          00158800
      WRITE (6,4) BMOIVG, DYRANG, PRANGE, EPRANG                        00158900
      WRITE (6,9) SERRIR, IRSEED, LENGT3                                00158950
      ORATIO = 10.D0**(-.1*DYRANG)                                      00159000
      THR = THR * 1.0-15                                                00159100
      OLGVR = OLOG(VR)                                                  00159200
      IF (KODE.EQ.2)   GO TO 100                                        00159300
      IF (KODE.EQ.3)   GO TO 110                                        00159400
      WRITE (6,5)                                                       00159500
      STOP                                                              00159600
  100 CONTINUE                                                          00159700
      B =(-3.040095 + 0.2133597*OR**2 - 4.022480*DLGVR**2 + 32.406036*  00159800
     1  OLGVR + 0.7171896*DLGVR*DR - 6.471279*DR) * ( 1. + 2./VR**2 )   00159900
      VAPOR = 7.84                                                      00160000
      GO TO 120                                                         00160100
  110 CONTINUE                                                          00160200
      B =(-8.464915 + 0.1546823*OR**2 - 4.554651*DLGVR**2 + 38.560562*  00160300
     1 OLGVR + 0.9938782*DLGVR*OR - 6.641898*OR) * ( 1. + 2./VR**2 )    00160400
      VAPOR = 1.87                                                      00160500
  120 CONTINUE                                                          00160600
      SOR = OR                                                          00160700
      TAULT = TRANS(SOR)                                                00160800
      EFACTR = 8.2146260-8 * LP * TAULT / (OR*BMDIVG)**2                00160900
      RETURN                                                            00161000
      END                                                               00161100
C--  AQRANG  --  AQRANG  --  AQRANG  --  AQRANG  --  AQRANG  --  AQRANG 00161200
      SUBROUTINE AQRANG (PHLOOK, THLOOK, RACQ)                          00161300
      IMPLICIT REAL*8 (A-H, O-Z)                                        00161400
      REAL*8 LRT, LOSRY, LOSRP, MXOLOS                                  00161500
      COMMON / BLK5 / PGYRO, TGYRO, CANT, BIAS, BIAS2, LOSRY, LOSRP,    00161600
     1            OROPS, KACQ, NRVS, IPRINT, ISTEP, NPRINT, JSTEP, NO   00161700
      COMMON/BLK7/ PHNORM, THNORM, LRT, B, CEIL, OR                     00161800
C                                                                       00161900
C--  AQRANG COMPUTES THE ACQUISITION RANGE.                             00162000
C                                                                       00162100
    1 FORMAT(' TARGET NOT VISIBLE FROM PROJECTILE', 1H', 'S VIEWPOINT AT00162200
     1 STEP ', I5 )                                                     00162300
C                                                                       00162400
      COSANG = -OCOS(THNORM)*OSIN(THLOOK) -                             00162500
     1     OSIN(THNORM)*OCOS(THLOOK)*OCOS(PHLOOK-PHNORM)                00162600
      IF (COSANG) 100,100,110                                           00162700
  100 CONTINUE                                                          00162800
      RACQ = 0.                                                         00162900
      IF (NPRINT.LT.2)   RETURN                                         00163000
```

```
      WRITE (6,1) ISTEP
      IPRINT = 1
      RETURN
  110 CONTINUE
      Z =(LRT*COSANG)**0.4
      RACQ = Z*B*1000.
      RETURN
      END
C--   AQSEE  --  AQSEE  --  AQSEE  --  AQSEE  --  AQSEE  --  AQSEE  --  A
      SUBROUTINE AQSEE (PHGYRO, THGYRO, PHLOOK, THLOOK, INFOV)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION FOV(2)
      COMMON/BLK4/ H, PI, PRF, MEASUR
      COMMON/BLK8/ FOV, MODE, KOND, ITARG, NS.
C
C--   AQSEE DETERMINES WHETHER TARGET IS IN FIELD OF VIEW.
C
      I = KONO
      IF (KOND.EQ.3)   I = 2
      COSLIM = DCOS(FOV(I))
      COSANG = DSIN(PHLOOK)*DCOS(THLOOK)*DSIN(PHGYRO)*DCOS(THGYRO)
     1      + OSIN(THLOOK)*OSIN(THGYRO)
     2      + OCOS(PHLOOK)*OCOS(THLOOK)*DCOS(PHGYRO)*DCOS(THGYRO)
      IF (COSANG-COSLIM) 20,20,10
   10 CONTINUE
      INFOV = 1
      RETURN
   20 CONTINUE
      IF (THGYRO-THLOOK) 30,40,40
   30 CONTINUE
      INFOV = 3
      RETURN
   40 CONTINUE
      O = DABS(PHGYRO-PHLOOK)*DCOS(THLOOK)
      IF ( (D.GT.FOV(I)) .AND. (D.LT.(2.*PI-FOV(I ))) )   GO TO 30
      INFOV = 2
      RETURN
      ENO
C--   SWITCH  --  SWITCH  --  SWITCH  --  SWITCH  --  SWITCH  --  SWITCH
      SUBROUTINE SWITCH (ISEE, TSW, ISEED)
      REAL*8 TSW, AVG0, AVG1
C--   REAL*4 ALOG, URN
      COMMON / BLK6 / AVG0, AVG1
C
C--   SWITCH CREATES OROPPPO -PULSE STRINGS IN THE INTERNAL MODE OF SPOT
C
      IF (ISEE.EQ.1)   GO TO 10
      ISEE = 1
      TSW = TSW - AVG1*ALOG( URN(ISEED) )
      RETURN
   10 CONTINUE
      ISEE = 0
      TSW = TSW - 0.5*AVG0*( ALOG( URN(ISEEO) ) + ALOG( URN(ISEED) ) )
      RETURN
      END
C--   SPOT1  --  SPOT1  --  SPOT1  --  SPOT1  --  SPOT1  --  SPOT1  --  S
      SUBROUTINE SPOT1(MOOE,P11,P12,P13,P22,P23,P33,T,FREQ, TGTHOG)
C**** THIS PROGRAM GENERATES LASER SPOT MOTION OURING TRACKING
C**** RELATIVE TO AN INTENDED CENTER OF OESIGNAIION IN AN INERTIAL
C**** FRAME OF REFERENCE--XS IS THE RANGEWISE COORDINATE.
```

```
C**** YS IS ELEVATION OR HEIGHT, AND ZS IS CROSSTRACK OR DEFLECTION
C**** IN A RIGHT-HAND COORDINATE SYSTEM.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 SN(3), NORMI, XX(3,2505),METERS, BRITE(2505),SPR
      DIMENSION X(3,3),Z(3,3),FOV(2),AM(3,3),YV(3),ZV(3)
      DIMENSION DESGPT(3)
      EQUIVALENCE (YV(1),X(1,1))
      COMMON/ BLKI2 / BRIGHT, TIMED, NULL, IVIS, INCR
      DATA METERS / .0254   / ,PI/3.I41592600/
C
    1 FORMAT (I10,4F10.0)
    2 FORMAT (20X, E10.3, 3F10.0)
    3 FORMAT (1X, I4, ' EXTERNALLY-GENERATED SPOT POSITIONS READ IN.  CR
     1OSS-SECTION CORRECTION FACTOR IS ',F7.3 / ' DESIGNATION POINT IN F
     2ACET COORDINATE SYSTEM IS (', 3F8.2, ') INCHES'/ 'I')
    4 FORMAT ('I')
      IVIS = 1
      C = DCOS(TGTHDG)
      S = DSIN(TGTHDG)
      IF(MODE.GT.I) GO TO 70
C--   MODE=I  =>  SPOT MOTION GENERATEO INTERNALLY
      OO 20 I=1,3
      DO 10 J=1,3
      X(I,J) = 0.
   10 Z(I,J) = 0.
   20 CONTINUE
      WD = 2.*PI*FREQ
      ZA = DSQRT(0.500)
      WA = DTAN(0.5*WD*T)
      XDENOM = I. / ( 1. + WA*(2.*ZA+WA) )
      A0 = WA*WA * XDENOM
      A1. = 2. * A0
      A2 = A0
      B1 = 2. * (WA*WA-1.) * XDENOM
      B2 = ( 1. - WA*(2.*ZA-WA) ) * XDENOM
      AA1 = B1
      AA2 = 1. - B1*B1 - B2*B2
      BB1 = 1. + B2
      BB2 = -2.*B1*B2
      CC1 = A0*A1 + A2*(A1-B1*A0)
      CC2 = -2.*( B1*CCI + A0*A2*B2 ) + A0*A0 + AI*A1 + A2*A2
      ANUM = CC1*BB2 - CC2*BB1
      DENOM = AA1*BB2 - AA2*BB1
      R = DSQRT(OENOM/ANUM)
C--   THE INTERNAL-SPOT-MOTION TRANSFORM MATRIX PROVIOES CORRECT VARIANCE
C     COVARIANCE  OF SPOT POSITION IN FACET COORDINATES AND ROTATES
C     INTO DIRECTION OF TARGET MOTION  IN ZOT COORDINATES
      AM(1,I) = -P11*S + P13*C
      AM(1,2) = P23*C
      AM(1,3) = P33*C
      AM(2,1) = P12
      AM(2,2) = P22
      AM(2,3) = 0.
      AM(3,1) = -P11*C - PI3*S
      AM(3,2) = -P23*S
      AM(3,3) = -P33*S
      WRITE (6,4)
      RETURN
C--   MODE=2 OR 3  =>  SPOT MOTION IS EXTERNALLY GENERATED
```

```
C--  MODE 2 IS MIXED:  EXTERNAL POSITION, INTERNAL INTENSITY             00175100
C--  MODE 3 IS EXTERNAL POSITION AND INTENSITY                           00175200
  70 CONTINUE                                                            00175300
C--  **NOTE**  EXTERNAL GENERATION ROUTINE ASSUMES TARGET VEHICLE IS FAC00175400
C              IN THE  -X  DIRECTION IN THE ZOT COORDINATE SYSTEM.      00175500
C              THE INPUT DATA ARE REFERENCED TO A SYSTEM IN WHICH  +X  100175600
C              FORWARD WITH RESPECT TO THE VEHICLE.  +Y  IS TO THE LEFT.00175700
C              IS UP.                                                    00175800
C              THE ORIGIN IS AT THE TRAILING EDGE OF THE TANK, ON THE CE00175900
C              AND AT GROUND LEVEL.                                      00176000
C--  **NOTE**  INPUT IS IN INCHES AND IS CONVERTED TO METERS.            00176100
C              THE INPUT POSITION IS ALSO TRANSLATED TO A DESIGNATION-  00176200
C              POINT-CENTERED FRAME AND ROTATED INTO THE DIRECTION OF   00176300
C              TARGET MOTION.                                            00176400
C                                                                        00176500
      READ (5,1)  NSPOTS, CFACTR, DESGPT                                 00176600
      WRITE (6,3) NSPOTS, CFACTR , DESGPT                                00176700
C--  DESGPT IN FACET COORDINATES IN INCHES                               00176800
C    CONVERT TO ERIM COORDINATES                                         00176900
      DESGPT(1) = - DESGPT(1)                                            00177000
      HOLD =  DESGPT(2)                                                  00177100
      DESGPT(2) = DESGPT(3)                                              00177200
      DESGPT(3) = HOLD                                                   00177300
      DO 90 I = 1, NSPOTS                                                00177400
      READ (5,2) BRITE(I), (YV(J),    J=1,3)                             00177500
      BRITE(I) = BRITE(I)*CFACTR                                         00177600
      DO 80 J=1,3                                                        00177700
   80 YV(J) = (YV(J) - DESGPT(J))*METERS                                 00177800
      XX(I,1) = -S*YV(1) - C*YV(2)                                       00177900
      XX(2,1) = C*YV(1) - S*YV(2)                                        00178000
      XX(3,1) = YV(3)                                                    00178100
   90 CONTINUE                                                           00178200
      I=0                                                                00178300
      RETURN                                                             00178400
C                                                                        00178500
C--  SPOTMD  --  SPOTMD  --  SPOTMD  --  SPOTMD  --  SPOTMD  --  SPOTMD 00178600
      ENTRY SPOTMD(N,XS,YS,ZS)                                           00178700
      IF (N.LE.0)   RETURN                                               00178800
      IF (MODE.GT.1)   GO TO 100                                         00178900
C--  MODE 1, INTERNAL                                                    00179000
      DO 60 K=1,N                                                        00179100
      DO 50 I=1,3                                                        00179200
      DO 40 J=1,2                                                        00179300
      JJ = 4 - J                                                         00179400
      X(I,JJ) = X(I,JJ-1)                                                00179500
   40 Z(I,JJ) = Z(I,JJ-1)                                                00179600
      SPR=R                                                              00179700
      Z(I,1)=NORM1(0.,SPR)                                               00179800
   50 X(I,1) = A0*Z(I,1) + A1*Z(I,2) + A2*Z(I,3) - B1*X(I,2) - B2*X(I,3)00179900
   60 CONTINUE                                                           00180000
      CALL MAVMPY(AM,YV,ZV,3)                                            00180100
      XS=ZV(1)                                                           00180200
      YS=ZV(2)                                                           00180300
      ZS=ZV(3)                                                           00180400
      RETURN                                                             00180500
  100 CONTINUE                                                           00180600
C--  MODE 2 OR 3, EXTERNAL                                               00180700
      I = I + N                                                          00180800
  110 CONTINUE                                                           00180900
      IF (I.LE.NSPOTS)   GO TO 120                                       00181000
```

```
      I = I - NSPOTS
      GO TO 110
  120 CONTINUE
      XS = -XX(1,1)
      ZS = XX(2,I)
      YS = XX(3,1)
      IF (MODE.EQ.3)   GO TO 130
      IVIS = 1
      IF (BRITE(I).EQ.0.)    IVIS = 0
      RETURN
  130 BRIGHT = BRITE(1)
      RETURN
      END
C--  MAVMPY  --  MAVMPY  --  MAVMPY  --  MAVMPY  --  MAVMPY  --  MAVMPY
      SUBROUTINE MAVMPY(AM,BM,CM,NELMTS)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C**** CM IS THE PRODUCT OF MATRIX AM AND COLUMN VECTOR BM.
      DIMENSION AM(NELMTS,NELMTS),BM(NELMTS),CM(NELMTS)
      DO 20 I=1,NELMTS
      CM(I)=0.0
      DO 10 K=1,NELMTS
   10 CM(I)=CM(I)+AM(I,K)*BM(K)
   20 CONTINUE
      RETURN
      END
C--  SOUND  --  SOUND  --  SOUND  --  SOUND  --  SOUND  --  SOUND  --  S
      SUBROUTINE SOUND(  Y, A, RHO)
C
C     SUBROUTINE COMPUTES THE SPEED OF SOUND IN M/SEC
C     VERSUS ALTITUDE IN METERS.  ALSO COMPUTED IS THE
C     ACCELERATION DUE TO GRAVITY IN M/SEC/SEC AND THE
C     AIR DENSITY IN KG/M**3 AND THE ABSOLUTE VISCOSITY
C     OF THE AIR IN KG/M/SEC.  NOTE THAT REYNOLD'S NUMBER
C     PER METER IS GIVEN BY A*RHO*EMACH/VISCO.
C
C     G=9.84 *(6.378E6/(6.378E6+Y))**2
      D=6.356766E6+Y
      IF(Y.LE.11019.07) GO TO 1
      IF(Y.LE.20063.12) GO TO 2
      IF(Y.LE.32161.9) GO TO 3
      IF(Y.LE.47350.09) GO TO 4
      IF(Y.LE.52428.88) GO TO 5
      IF(Y.LE.61591.03) GO TO 6
      IF(Y.LE.79994.14) GO TO 7
      RHO=0.4636*EXP(-0.12207E-3*Y)
C     T=TEMPERATURE IN DEGREES KELVIN
      T=180.65
    8 A=20.053*SQRT(T)
C     VISCO=0.00467*(T+110.)*(T/217.78)**1.5
C     THIS IS THE SUTHERLAND VISCOSITY LAW.
      RETURN
    1 RHO=1.224999+Y*(-.1176033E-3+Y*(.433719E-8+Y*(-.7461659E-13
     1 +Y*(.5537603E-18-.9572727E-24*Y))))
      T=(1.831702E9-4.103083E4*Y)/D
      GO TO 8
    2 RHO=1.990142+Y*(-.2940114E-3+Y*(.1993974E-7+Y*(-.7637263E-12
     1 +Y*(.1615921E-16-.1476764E-21*Y))))
      T=216.65
      GO TO 8
```

```
3 RHO=1.61561+Y*(-.2357495E-3+Y*(.1308607E-7+Y*(-.3519651E-12
 1 +Y*(.5798729E-17-.3626654E-22*Y))))
  T=(1.250058E9+6.553416E3*Y)/D
  GO TO 8
4 RHO=1.10944+Y*(-.1140029E-3+Y*(.4817401E-8+Y*(-.1039241E-12
 1 +Y*(.1138793E-17-.5052135E-23*Y))))
  T=(8.839083E8+1.7938E4*Y)/D
  GO TO 8
5 RHO=.8974979E-1+Y*(-.417905E-5+Y*(.3529753E-10+Y*(.1177144E-14
 1 +Y*(-.2567072E-19+.1449113E-24*Y))))
  T=270.65
  GO TO 8
6 RHO=.1029082E-1+Y*(.1081853E-5+Y*(-.8523619E-10+Y*(.2075003E-14
 1 +Y*(-.2164824E-19+.860425E-25*Y))))
  T=(2.381562E9-1.233889E4*Y)/D
  GD TO 8
7 RHO=0.4636*EXP(-0.12207E-3*Y)
  T=(3.157088E9-2.493041E4*Y)/D
  GO TO 8
  END
C--  PARAM5  --  PARAM5  --  PARAM5  --  PARAM5  --  PARAM5  --  PARAM5
  SUBROUTINE PARAM5 (51)
  REAL*8 KALF(2),IB,51(12),FNY,FA,TALF(2),ALPHA,DSQRT,V5Q,CNA,AQ,
 1 FNP
  REAL*8 CNO, DAB5 , CAO
  REAL*8          X5, Y5, Z5,            HN,       GROUNO
  REAL*8 ALFAY, ALFAP
  REAL*8 ALFAMX
  REAL*4 MACHNO, MACH1
  OIMENSION NMACH(4), NALPHA(4), VMACH(7,4), VALPHA(7,4),TKA(7,7,4),
 1     TCNA(7,7,4), TCND(7,4), TCAO(7,4), TCAD(7,4), TX5M(7,4)
  OIMENSIDN NMCAO(4), VMCAD(7,4), VDELTA(7,4), NOELTA(4)
  OATA VDELTA / 22*0., .2094, 5*0. /
  DATA NDELTA / 1,1,1,2 /
  DATA NMCAD / 7, 5, 7, 7 /
  OATA NMACH / 3, 5, 7, 3 /, NALPHA / 1, 1, 1, 3 /
  OATA VMCAO / .4, .625, .75, .812, .875, .938, 1.,
 1 .6,.7,.8,.9,1.,2*0.,.8,.85,.9,1.,1.05,1.1,2.,.5,
 2 .6, .7, .8, .9, 1., 1.1 /
  OATA VMACH / .4,.8,1.,4*0.,.6,.7,.8,.9,1.,2*0.,.7,.8,.925,1.,1.06,
 1 1.1,1.3,.5,.8,1.,4*0. /
  OATA VALPHA / 21*0., .0873, .1745, .2618, 4*0. /
  DATA TKA / 1.4, 1.2, 1.1, 46*0., .68, .64, .6, .7, .83, 44*0.,
 1 .8 ,.7,.58,.55,3*.54,42*0.,1.75,1.76,1.70,
 2      4*0., 1.102, 1.088, 1.076, 39*0. /
  DATA TCNA / 10.8, 12.3, 15.6, 46*0.,
 2 15.,15.8,16.8,18.,18.5,44*0.,9.19,9.27,10.1,11.25,11.82,11.51,10.
 335,42*0.,16.1,17.6,19.1,4*0.,16.1,16.6,19.1,4*0.,14.4,14.7,
 4 19.0,32*0. /
  DATA TCND / -6.0, -7.0, -10.0, 4*0., 2.6, 2.7, 2.8, 2.5, 2.45,
 1 2*0.,.42,.85,1.22,.8,.63,.6,.97,-4.6,-5.2,-5.3,4*0. /
  OATA TCAD / .26, .27, .31, .34, .4, .61, .62,
 1 .24,.25,.27,.3,.58,2*0.,.285,.31,.362,.664,.696,.703,.622,.37,.37
 2 ,.38,.4,.51,.72,.79 /
  DATA TCAD / 1.2, 1.5, 1.8, 4*0.,
 * 1.8,1.8,1.75,2.,2.3,2*0.,1.27,1.62,1.32,.79,1.38,1.76,2.9,
 1 7*0. /
  OATA TX5M / 1.2,1.4,1.5,4*0.,.7,.7,.75,.5,.5,2*0.,.98,1.,1.12,1.26
 1, 1.41,1.30,1.1,3*.55,4*0. /
  DATA A,        CONST / .01887,        683.8251 /
```

```
      DATA ALFAMX / .26179900 /                                         00192800
C--   CONST IS 8/(PI*D**3)  IN M**(-3)                                  00192900
      COMMON / BLKI / XS, YS, ZS,                  HN,       GROUND     00193000
      COMMON / BLK9 / FNY, FNP, FA, TALF, VSQ, CNA, AQ, KALF, IB, M     00193100
      COMMON / BLK11 / CND, CAO, MACHNO                                 00193200
C                                                                       00193300
C--   PARAMS COMPUTES VALUES OF AERODYNAMIC COEFFICIENTS BY A TABLE-LOOKU00193400
C     METHOD.                                                           00193500
C                                                                       00193600
      ALT = SI(2) + GROUND                                              00193700
C--   OBTAIN MACH NUMBER AND AIR DENSITY                                00193800
      CALL SOUND (ALT, MACH1,  RHO)                                     00193900
      MACHNO =DSQRT(VSQ) / MACH1                                        00194000
C                                                                       00194100
C--   M IS THE CLGP MODEL NUMBER:                                       00194200
C          1  IS THE MARTIN AD VERSION                                  00194300
C          2  IS TEXAS INSTRUMENTS                                      00194400
C          3  IS NAVY                                                   00194500
C          4  IS MARTIN ED VERSION                                      00194600
C                                                                       00194700
C--   TABLE LOOKUP                                                      00194800
      NM = NMACH(M)                                                     00194900
      NA = NALPHA(M)                                                    00195000
      ND = NDELTA(M)                                                    00195050
C--   MACH NUMBER INTERPOLATION . . . . . .                             00195075
      DO 50 I=I, NM                                                     00195100
      DIFF = VMACH(I,M) - MACHNO                                        00195200
      IF(DIFF.LE.0.)   GO TO 50                                         00195300
      IF (I.EQ.I)   GO TO I00                                           00195400
      IMACH = I-I                                                       00195500
      JMACH = I                                                         00195600
      RMACH =  DIFF/( VMACH(JMACH,M) - VMACH(IMACH,M) )                 00195700
      GO TO I20                                                         00195800
   50 CONTINUE                                                          00195900
      IMACH = NM                                                        00196000
      JMACH = NM                                                        00196100
      GO TO 1I0                                                         00196200
  100 CONTINUE                                                          00196300
      IMACH = 1                                                         00196400
      JMACH = I                                                         00196500
  I10 CONTINUE                                                          00196600
      RMACH = 0.                                                        00196700
  I20 CONTINUE                                                          00196800
C--   ALPHA INTERPOLATION . . . . . . . . . . .                         00196850
      ALPHA = DSQRT( S1(7)**2 + S1(8)**2 )                              00196900
      DO I50 I=1, NA                                                    00197000
      DIFF = VALPHA(I,M) - ALPHA                                        00197100
      IF (DIFF.LE.0.)   GO TO I50                                       00197200
      IF (I.EQ.I)   GO TO 200                                           00197300
      IALPHA = I-1                                                      00197400
      JALPHA = I                                                        00197500
      RALPHA =  DIFF/( VALPHA(JALPHA,M) - VALPHA(IALPHA,M) )            00197600
      GO TO 220                                                         00197700
  150 CONTINUE                                                          00197800
      IALPHA = NA                                                       00197900
      JALPHA = NA                                                       00198000
      GO TO 2I0                                                         00198100
  200 CONTINUE                                                          00198200
      IALPHA = I                                                        00198300
      JALPHA = I                                                        00198400
```

```
  210 CONTINUE                                                          00198500
      RALPHA = 0.                                                       00198600
  220 CONTINUE                                                          00198700
C--  DELTA INTERPOLATIONS . .    .    .    .    .    .    .            00198701
      DO 245 II=1,2                                                     00198702
      DELTA = DABS(SI(II+10))                                           00198703
      DO 225 I=1,ND                                                     00198704
      DIFF = VDELTA(I,M) - DELTA                                        00198705
      IF (DIFF.LE.0.)   GO TO 225                                       00198706
      IF (I.EQ.I)   GO TO 230                                           00198707
      IDELTA = 1-I                                                      00198708
      JDELTA = 1                                                        00198709
      RDELTA = DIFF/( VDELTA(JDELTA,M) - VDELTA(IDELTA,M) )             00198710
      GO TO 240                                                         00198711
  225 CONTINUE                                                          00198712
      IDELTA = ND                                                       00198713
      JDELTA = ND                                                       00198714
      GO TO 235                                                         00198715
  230 CONTINUE                                                          00198716
      IDELTA = I                                                        00198717
      JDELTA = 1                                                        00198718
  235 CONTINUE                                                          00198719
      RDELTA = 0.                                                       00198720
  240 CONTINUE                                                          00198721
      KALF(II) = TKA(IMACH,IDELTA,M)*RMACH*RDELTA +                     00198722
     I           TKA(IMACH,JDELTA,M)*RMACH*(1.-RDELTA) +                00198723
     2           TKA(JMACH,IDELTA,M)*(1.-RMACH)*RDELTA +                00198724
     3           TKA(JMACH,JDELTA,M)*(1.-RMACH)*(1.-RDELTA)             00198725
  245 CONTINUE                                                          00198800
      CNA = TCNA(IMACH,IALPHA,M)*RMACH*RALPHA + TCNA(IMACH,JALPHA,M)*   00198900
     1RMACH*(1.-RALPHA) + TCNA(JMACH,IALPHA,M)*(1.-RMACH)*RALPHA +      00199000
     2     TCNA(JMACH,JALPHA,M)*(I.-RMACH)*(1.-RALPHA)                  00199100
      CND = TCND(IMACH,M)*RMACH + TCND(JMACH,M)*(1.-RMACH)              00199200
      CAD = TCAD(IMACH,M)*RMACH + TCAD(JMACH,M)*(1.-RMACH)              00199300
      XSM = TXSM(IMACH,M)*RMACH + TXSM(JMACH,M)*(1.-RMACH)              00199400
C--  SEPARATE TABLE LOOKUP FOR CAO                                      00199500
      NM = NMCAO(M)                                                     00199600
      DO 250 I=1, NM                                                    00199700
      DIFF = VMCAD(I,M) - MACHNO                                        00199800
      IF (DIFF.LE.0.)   GO TO 250                                       00199900
      IF (I.EQ.I)   GO TO 300                                           00200000
      IMACH = I - I                                                     00200100
      JMACH = I                                                         00200200
      RMACH = DIFF / ( VMCAO(JMACH,M) - VMCAO(IMACH,M) )                00200300
      GO TO 320                                                         00200400
  250 CONTINUE                                                          00200500
      IMACH = NM                                                        00200600
      JMACH = NM                                                        00200700
      GO TO 310                                                         00200800
  300 CONTINUE                                                          00200900
      IMACH = I                                                         00201000
      JMACH = I                                                         00201100
  310 CONTINUE                                                          00201200
      RMACH = 0.                                                        00201300
  320 CONTINUE                                                          00201400
C--  COMPUTE PARAMETERS                                                 00201500
      CAO = TCAO(IMACH,M)*RMACH + TCAO(JMACH,M)*(I,-RMACH)              00201600
      CNY = CNA*SI(7) + CND*SI(11)                                      00201700
      CNP = CNA*SI(8) + CND*SI(I2)                                      00201800
C--  THE CA EQUATION DIFFERS AMONG MODELS.                              00201900
```

```
      IF (M.EQ.1)    GO TO 400
      IF (M.EQ.4)    GO TO 330
      CA = CAO + CAO*( SI(II)**2 + SI(12)**2 )
      GO TO 410
  330 CONTINUE
      IF (MACHNO.LE.0.5)    GO TO 360
      IF (MACHNO.GT.0.8)    GO TO 370
      RMACH = (MACHNO-0.5) / 0.3
      CI = 4.*(1.-RMACH) + 5.*RMACH
      C2 = 5.
      GO TO 390
  360 CONTINUE
      CI = 4.
      C2 = 5.
      GO TO 390
  370 CONTINUE
      IF (MACHNO.GT.1.0)    GO TO 380
      RMACH = (MACHNO-0.8) / 0.2
      CI = 5.*(I.-RMACH) + 6.5*RMACH
      C2 = 5.*(1.-RMACH) + 4.*RMACH
      GO TO 390
  380 CONTINUE
      C1 = 6.5
      C2 = 4.
  390 CONTINUE
      ALFAY = S1(7)
      ALFAP = S1(8)
      IF (ALFAY.GT.ALFAMX)    ALFAY = +ALFAMX
      IF (ALFAY.LT.-ALFAMX)   ALFAY = -ALFAMX
      IF (ALFAP.GT.+ALFAMX)   ALFAP = +ALFAMX
      IF (ALFAP.LT.-ALFAMX)   ALFAP = -ALFAMX
      CA = CAO + C1*( (.4*ALFAY-SI(1I))**2 + (.4*ALFAP-S1(12))**2 )
     I     - C2*( OABS(ALFAY)**3 + OABS(ALFAP)**3 )
      GO TO 410
  400 CONTINUE
      CA = CAO + CAO*( OABS( S1(1I) ) + DABS( S1(I2) ) )
  410 CONTINUE
      AQ = A*RHO*VSQ*0.5
      FNY = AQ*CNY
      FNP = AQ*CNP
      FA = AQ*CA
      00 420 I=I,2
  420 TALF(I) = OSQRT( CONST*IB / ( (CNA+CND/KALF(I)) *RHO*VSQ*XSM) )
      RETURN
      END
C--  OENSTY  --  DENSTY  --  DENSTY  --  OENSTY  --  OENSTY  --  OENSTY
      SUBROUTINE DENSTY (KSEE, SGLEVL)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 KDEL, KOEL1, KDEL2
      REAL*4 SLRNGI, SLRNG2, TRANS
      COMMON / BLK2 / XT, YT, ZT, KOEL, KDELI, KDEL2, SLRNG1, SLRNG2
      COMMON / BLKI2 / CROSS, TIME0, NULL, IVIS, INCR
      COMMON / BLK13 / DRATIO, EFACTR, E, THR
C
C--  DENSTY CALCULATES THE RECEIVEO ENERGY DENSITY ANO DETERMINES WHETHE
C    PULSE IS OF VISIBLE INTENSITY.
C
      ELAST = E
      TAUTP = TRANS(SLRNG1*.001)
      E = (EFACTR * CROSS * TAUTP / SLRNG1**2) * IVIS
```

```
      IF (E.GE.THR)   GO TO 100
      E = THR
   50 KSEE = 0
      RETURN
  100 CONTINUE
      RATIO = E / ELAST
      IF (RATIO.LT.DRATIO)   GO TO 50
      KSEE = 1
      SGLEVL = 10.D0 * DLOG10(E/THR)
      RETURN
      END
C--  TRANS  --  TRANS  --  TRANS  --  TRANS  --  TRANS  --  TRANS  --  T
      FUNCTION TRANS(RANGE)
      COMMON / BLK15 / VAPOR, HRANGE, PRANGE
      DATA RMIN / 10.97394 /, XRATIO / .518868 /
C--  XRATIO  IS RATIO OF SCATTERER SIZE TO WAVELENGTH
C--  RANGE = BEAM PATH LENGTH
C--  HRANGE = VISIBILITY RANGE DUE TO ATMOSPHERIC WATER VAPOR
C--  PRANGE = VISIBILITY RANGE DUE TO PARTICULATE
C--       THE ABOVE RANGES IN KILOMETERS . . .
      TAU1 = 1. - ERF(.195*SQRT(.1*VAPOR*RANGE))
      EXPON = 1.3
      IF (HRANGE.LT.RMIN)   EXPON = .585*HRANGE**.3333333
      TAU2 = EXP(-(XRATIO**EXPON)*3.912*RANGE/HRANGE)
      TAU3 = EXP(-3.912*RANGE/PRANGE)
C--  A TAU4 IS REQUIRED IF RAINFALL IS CONSIDERED
      TRANS = TAU1 * TAU2 * TAU3
      RETURN
      END
C--  SKRTBL  --  SKRTBL  --  SKRTBL  --  SKRTBL  --  SKRTBL  --  SKRTBL
      SUBROUTINE SKRTBL (NTD, NSL, MODDEP, NMODES, VTD, VSL, TMD, TSR)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 SERRIR, S, AMIN, AMAX, DEVIAT
      DIMENSION VTD(NTD), VSL(NSL), TMD(NTD,NSL,NMODES),
     1     TSR(NTD,NSL,MODDEP,NMODES)
      DATA RADIAN / 1.745329250-2 /
      COMMON / BLK8 /  FOV(2), MODESM, KOND, ITARG, NSW
      COMMON / BLK18 / SERRIR, IRSEED, LENGT3, NRVS3
C
    1 FORMAT (20F4.0)
    2 FORMAT (//' TABLES OF DETECTOR TRANSFER CHARACTERISTICS FOR LASER
     1GUIDANCE'/' DEVIATION (DEG)', T25, 10(1X,F10.4))
    3 FORMAT (1X, F10.4, T25, 10(1X, F10.4) )
    4 FORMAT ( ' SIGNAL LEVEL (DB)', T25, 'SLEW RATES (UNGUIDED MODES, D
     1EG/SEC)')
    5 FORMAT (' SIGNAL LEVEL (DB)', T25, 'ESTIMATED LOS RATES (DEG/SEC)'
     1)
    6 FORMAT (' SLEW RATES FOR GUIDED MODES ARE SAME AS ABOVE')
    7 FORMAT (' SIGNAL LEVEL (DB)', T25, 'SLEW RATES (GUIDED MODES, DEG/
     1SEC)')
    8 FORMAT (//' TABLES OF DETECTOR TRANSFER CHARACTERISTICS FOR PASSIV
     1E IR GUIDANCE'/' DEVIATION (DEG)', T25, 10(1X, F10.4))
   11 FORMAT (' CHARACTERISTICS FOR PASSIVE IR MODE ARE SAME AS FOR LASE
     1R MODE')
C
      READ (5,1) VTD, VSL, TMD, TSR
      WRITE (6,2) VTD
      DO 10 I=1,NSL
      WRITE (6,5)
   10 WRITE(6,3) VSL(I), (TMD(J,I,1), J=1,NTD)
```

```
      WRITE (6,4)
      DO 20 I=1,NSL
   20 WRITE (6,3) VSL(I), (TSR(J,I,1,1), J=I,NTD)
      IF (MODDEP.EQ.1)    WRITE (6,6)
      IF (MODDEP.EQ.1)    GO TO 50
      WRITE (6,7)
      DO 30 I=1,NSL
   30 WRITE (6,3) VSL(I),(TSR(J,I,2,1), J=1,NTD)
   50 CONTINUE
      IF (NMODES.EQ.1)    GO TO 58
      WRITE (6,8) VTD
      DO 51 I=1,NSL
      WRITE(6,5)
   51 WRITE (6,3) VSL(I), (TMD(J,I,2), J=1,NTD)
      WRITE (6,4)
      DO 52 I=1,NSL
   52 WRITE (6,3) VSL(I), (TSR(J,I,1,2), J=1,NTD)
      IF (MODDEP.EQ.1)    WRITE (6,6)
      IF (MODDEP.EQ.1)    GO TO 58
      WRITE (6,7)
      DO 53 I=1,NSL
   53 WRITE (6,3) VSL(I), (TSR(J,I,2,2), J=1,NTD)
      GO TO 59
   58 WRITE (6,11)
   59 CONTINUE
      DO 70 I=1,NTD
      VTD(I) = VTD(I)*RADIAN
      DO 69 L=1,NMODES
      DO 65 J=I,NSL
      TMD(I,J,L) = TMD(I,J,L)*RADIAN
      DO 60 K=I, MODDEP
   60 TSR(I,J,K,L) = TSR(I,J,K,L)*RADIAN
   65 CONTINUE
   69 CONTINUE
   70 CONTINUE
      RETURN
C
C--  LOOKUP  --  LOOKUP  --  LOOKUP  --  LOOKUP  --  LOOKUP  --  LOOKUP
      ENTRY LOOKUP (MODSKR,TRUDEV,SGLEVL,ELDSRT,SLWRAT,AMAX,AMIN,S)
      L = ITARG
      IF (NMODES.EQ.1)    L = I
      DETDEV = TRUDEV
      IF (ITARG.EQ.I)    GO TO I00
      ISEED = IRSEED
      CALL NORMXX (AMIN, AMAX, 0., S, DEVIAT, ISEED)
      IRSEED = ISEED
      DETDEV = TRUDEV + DEVIAT
      NRVS3 = NRVS3 + 1
  I00 CONTINUE
      IF (DETDEV.LT.0.D0)    GO TO 110
      ATD = DETDEV
      SIGN = 1.
      GO TO 120
  110 ATD = -DETDEV
      SIGN = -1.
  120 CONTINUE
      DO 130 I=2,NTD
      IF (VTD(I).LE.ATD)    GO TO I30
      ITD = I
      RTD = (ATD-VTD(I-1)) / (VTD(I)-VTD(I-1))
```

```
      GO TO 140
  130 CONTINUE
      ITD = NTD
      RTD = 1.
  140 CONTINUE
      M = MODSKR
      IF (MODDEP.EQ.1)    M=1
      IF (NSL.EQ.1)    GO TO 160
      DO 150 I=2,NSL
      IF (VSL(I).LE.SGLEVL)    GO TO 150
      ISL = I
      RSL = (SGLEVL-VSL(I-1)) / (VSL(I)-VSL(I-1))
      GO TO 170
  150 CONTINUE
      ISL = NSL
      RSL = 1.
  170 CONTINUE
      ELOSRT = SIGN*( TMD(ITD,ISL,L)*RTD*RSL + TMD(ITD,ISL-1,L)*RTD*
     1     (1.-RSL) + TMD(ITD-1,ISL,L)*(1.-RTD)*RSL + TMD(ITD-1,ISL-1,L)
     2     *(1.-RTD)*(1.-RSL) )
      SLWRAT = SIGN*( TSR(ITD,ISL,M,L)*RTD*RSL + TSR(ITD,ISL-1,M,L)*
     1     RTD*(1.-RSL) + TSR(ITD-1,ISL,M,L)*(1.-RTD)*RSL +
     2     TSR(ITD-1,ISL-1,M,L)*(1.-RTD)*(1.-RSL)  )
      RETURN
  160 CONTINUE
      ELOSRT = SIGN * ( TMD(ITD,1,L)*RTD + TMD(ITD-1,1,L)*(1.-RTD)   )
      SLWRAT = SIGN * ( TSR(ITD,1,M,L)*RTD + TSR(ITD-1,1,M,L)*(1.-RTD) )
      RETURN
      END
C--  NORMXX  --  NORMXX  --  NORMXX  --  NORMXX  --  NORMXX  --  NORMXX
      SUBROUTINE NORMXX (AMIN, AMAX, AMEAN, SIGMA, X, ISEED)
C
C     THIS SUBROUTINE GENERATES A NORMAL DEVIATE
C
      CALL RANDMM(ISEED,X)
      Z = X
      CALL RANDMM(ISEED,X)
      X = (((-2.0*ALOG(Z))**0.5)*(COS(6.283*X)))*SIGMA + AMEAN
      IF (X.LT.AMIN) X = AMIN
      IF (X.GT.AMAX)  X = AMAX
      RETURN
      END
C--  RANDMM  --  RANDMM  --  RANDMM  --  RANDMM  --  RANDMM  --  RANDMM
      SUBROUTINE RANDMM(ISEED,X)
C
C     THIS SUBROUTINE GENERATES UNIFORM DEVIATES
C
      ISEED = ISEED*65539
      IF(ISEED)2266,2266,2277
 2266 ISEED = ISEED + 2147483647 + 1
 2277 X = ISEED
      X = X*.4656613E-9
      RETURN
      END
```

DISTRIBUTION LIST

Copy No.	
1	Department of Defense
	Office of the Defense Director, Research & Engineering
	ATTN: Assistant Director (Test Resources)
	Room 3D 116, The Pentagon
	Washington, DC 20301
	HQ, Department of the Army
1	ATTN: DAMO-RDQ
1	DAMA-CSM-CM
	Washington, DC 20310
1	Commander
	US Army Materiel Development & Readiness Command
	ATTN: DRCPM-SA
	5001 Eisenhower Avenue
	Alexandria, VA 22333
	Commander
	US Army Armament Command
1	ATTN: DRSAR-EN
1	DRSAR-RD
2	DRSAR-SA
6	DRSAR-SAM
1	DRSAR-PA
1	DRSAR-AS
	Rock Island, IL 61201
	Commander
	US Army Test and Evaluation Command
1	ATTN: DRSTE-SY
1	DRSTE-TA
1	DRSTE-ME
	Aberdeen Proving Ground, MD 21005
	Commander
	US Army Electronics Command
1	ATTN: DRSEL-SA
1	DRSEL-RD
	Fort Monmouth, NJ 07703
	Commander
	US Army Missile Command
1	ATTN: DRDMI-CM
1	DRDMI-D
1	DRDMI-RGN
2	DRDMI-CAWS-FO
1	DRDMI-RGT
	Redstone Arsenal, AL 35809

Copy No.	
	Commander US Army Tank Automotive Command
1	ATTN: DRSTA-S
1	DRSTA-R
1	DRSTA-V
	Warren, MI 48090
	Commander HQ, US Army Aviation Systems Command
1	ATTN: DRSAV-E
1	DRSAV-W
1	DRSAV-D
1	DRSAV-WR (COL Powers)
	P. O. Box 209, Main Office St. Louis, MO 63166
	Commander Rock Island Arsenal
1	ATTN: SARRI-RL
1	DRDAR-LCA-PP
1	SARRI-RLP-L
	Rock Island, IL 61201
1	Commander Watervliet Arsenal ATTN: SARWV-RD Watervliet, NY 12189
	Commander Picatinny
1	ATTN: SARPA-PA
1	SARPA-AD
1	SARPA-ND
1	SARPA-TS
	Dover, NJ 07801
	Commander Edgewood Arsenal
1	ATTN: SAREA-DE-W
1	SAREA-DE-N
	Aberdeen Proving Ground, MD 21010
1	Project Manager for Selected Ammunition US Army Materiel Development & Readiness Command ATTN: DRCPM-SA2 Dover, NJ 07801

DISTRIBUTION LIST (Cont)

Copy No.	
1	Project Manager for M110E2 ATTN: DRCPM-M110E2 Rock Island, IL 61201
1	Project Manager for Cannon Artillery Weapons System ATTN: DRCPM-CAWS Dover, NJ 07801
1	Product Manager for Advanced Attack Helicopter Systems US Army Aviation Systems Command St. Louis, MO 63166
1	Product Manager for AH-1 Cobra Series Aircraft US Army Materiel Development & Readiness Command P. O. Box 209 St. Louis, MO 63166
	Commander US Army Materiel Systems Analysis Activity
1	ATTN: DRXSY-GS
1	DRXSY-C
1	DRXSY-T
1	DRXSY-D
1	DRXSY-AAM
	Aberdeen Proving Ground, MD 21005
	Commander US Army Ballistic Research Laboratories
1	ATTN: DRXBR-EB
1	DRXBR-TB
1	DRXBR-VL
	Aberdeen Proving Ground, MD 21005
1	Commander Army Ordnance Center & School ATTN: ATSL-CTD-MF Aberdeen Proving Ground, MD 21005
1	Commander Human Engineering Laboratories ATTN: DRXHE-SP Aberdeen Proving Ground, MD 21005
1	Commander US Army Aberdeen Research & Development Center ATTN: STEAP-TL Aberdeen Proving Ground, MD

Copy No.	
1	Commander US Army Armor School Fort Knox, KY 40121
1	Commandant US Army Field Artillery School Fort Sill, OK 73503
1	Commandant US Army Infantry School Fort Benning, GA 31905
1	Commander US Army Missile & Munitions Center & School Redstone Arsenal, AL 35809
1	Commander Modern Army Selected System Test, Evaluation & Review Fort Hood, TX 76544
1	Commander US Army Combat Developments Experimentation Command Fort Ord, CA 93941
1	Commander US Army TRADOC Systems Analysis Activity White Sands Missile Range White Sands, NM 88002
1	Commander Harry Diamond Laboratories ATTN: AMXDO-PP Washington, DC 24038
1	Director Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, VA 22209
1	Commander US Army Concepts Analysis Agency 8120 Woodmont Avenue Bethesda, MD 20014
1	Commander Combat Systems Group ATTN: CDMSD-M Fort Leavenworth, KS 66027

DISTRIBUTION LIST (Cont)

Copy No.	
1	Commander USA Combined Arms Combat Developments Activity ATTN: ATCACC Fort Leavenworth, KS 66027
1	Commander Defense Logistics Studies Information Exchange Fort Lee, VA 23801
	Commander US Army Mobility Equipment Research & Development Center
1	ATTN: STSFB-B
1	STSFB-ZN Fort Belvoir, VA 22060
1	Commander US Army Armor Training Center ATTN: AMSTB-CO-MM Fort Knox, KY 40121
1	Commander US Naval Weapons Laboratory Code GS, Mr. Godius Dahlgren, VA 22448
	Chief, Naval Operations
1	ATTN: OP 96
1	OP 982F Washington, DC 00350
1	Commander Naval Weapons Center Code 4072 Mr. E. Breitenstein China Lake, CA 93555
	Commander US Army White Sands Missile Range
1	ATTN: STEWS-TE-MD
1	STEWS-TE-PC White Sands Missile Range, NM 88002
1	Chief, Naval Materiel ATTN: MAT 0362 Washington, DC 20360
1	Director US Marine Corps Development and Education Center Quantico, VA 22134

Copy No.

12	Defense Documentation Center Cameron Station Alexandria, VA 22314