SECTION II

PROGRAM DESCRIPTION

2-1. GENERAL.

2-2. This section describes the subprograms which perform the targeting and MSS functions. The primary program functions are described together with program inputs and a discussion of program controls. An index of subprograms is provided at the end of the section.

2-3. PRIMARY PROGRAM FUNCTIONS.

2-4. The dynamic operational control (DOC) of both targeting and MSS is performed by DØCNT. This subprogram establishes appropriate parameters; causes control cards to be read-in, interpreted, and stored; establishes subprograms in core, and then transfers control to one of the functional subprograms. At the completion of a functional operation, DØCNT again assumes control and terminates the program.

2-5. Production of the binary input tape to the TITAN I

Targeting Program is controlled by the binary input tape area. The input tape is generated using the missile trajectory (M/T) and radar launcher (R/L) tapes. The M/T tape contains information about missile parameters and trajectory selection criteria, and is generated or updated under control of the decimal correction (DEC) area. The R/L tape contains information on geographic and survey parameters, and is generated or updated under control of the R/L area.

2-6. TARGETING PROGRAM.

2-7. The major purpose of the targeting program is to determine, for a given ground guidance complex and for up to ten possible real targets, the corresponding ideal-earth aim points which are used by the ground guidance computer so that the actual impact points will coincide with the real targets. This is necessary since the guidance computer, when performing missile guidance computations, ignores the non-sphericity of the earth's gravitational field and the effects of atmospheric drag on re-entry. Hence in such an ideal earth, the missile is guided to a psuedo target or target aim point which is so offset from the real-earth target location, that the missile when aimed to this target aim point, actually impacts the real earth at the desired target.

2-8. OFFSET TARGET COMPUTATION (OTC). The main function of OTC is to determine the target aim point. The theoretical determination of the target aim point is described in [redacted]

The theory volume.

1. ~~theory~~ of this document. The actual ~~implementation~~ determination of the target aim point ~~determination~~ is performed under control of subprogram ØTCNT. After information describing the particular ground guidance complex, missile and trajectory data are read from tape, the program reads target data cards and sets itself up for target aim point determination of up to ten targets, one at a time. Concurrently, re-entry vehicle control data and supplementary data are determined for each target. The step required to perform these computations for a target is known as the computation step.

2-9. After completion of computations, the target aim point and related trajectory data are written on magnetic tape which is converted to a punched mylar plastic tape (the target tape) for use by the ground guidance computer at the launch site. The re-entry vehicle data is written on a magnetic tape which is converted to punched IBM cards. The data on the IBM cards is then punched on plastic RRU cards. The card conversion is verified for agreement between the two sets of cards before the RRU cards are sliced into segments (R/V cards) for use by the operational ground equipment (OGE) at the launch site.

2-10. Verification of computations is performed by testing the target aim point for each target. The mylar target tape is used to prepare a new magnetic tape. All tape conversions are performed off-line or independent of the targeting computations. This second magnetic tape and the IBM cards are read into the targeting computer, and the target aim points are tested by complete closed loop flight simulation. The test of target aim points provides a check of the output target tape for all possible errors not previously detected, including punching errors in the tape preparation. Only new target calculations are verified by a complete simulation; unchanged target data from an old target tape is verified by comparing the data on the new tape with data on the old tape.

2-11. Subprogram ØTCNT can be used for generating a new output target tape or for changing some of the target parameters of an old (previously prepared) tape, or both. When a new target tape is to be generated, the sequence of events is as described previously. When an old target tape is to be changed, ØTCNT must read the old tape and save all targeting data of the targets not being changed, before the new target tape is punched.

2-12. OUTPUT DATA VERIFICATION (TOT). Output tape readback and testing normally are performed semi-automatically as part of the OTC function. Output tape testing can be performed as a separate function, TOT, in order to provide for testing of target tapes which were not tested by the verification process of the OTC function. This occurs if continuous semi-automatic processing is not possible at time of target tape generation, or if it is desired to test old target tapes if input conditions or parameters change. TOT actually is performed in much the same way as the verification process of OTC, except that the ground guidance complex parameters and target data must be re-read into the targeting computer and a complete simulation is run for each target.

2-13. TARGET ACCESSIBILITY AREA DETERMINATION (TAA). The target accessibility area contains the targets which have a very high probability of being within range of missiles launched from a given launcher. The accessibility area is defined by launch azimuth limits and minimum and maximum range contours for specific launch site-radar configurations.

2-14. Two launch azimuths are specified for each TAA run. Normally these may be chosen to be the maximum and minimum boundaries for a permissible launch azimuth sector. If more than one sector exists or the sector is extremely wide, additional runs may be made to determine the additional or intermediate points. The permissible launch azimuths for specified site-radar configurations are based on system constraints determined by independent engineering studies. These constraints are based on factors such as maximum radar azimuth slew rate, terrain profiles, and signal strength requirements.

2-15. The TAA range limits for the specified site-radar configuration are determined by open loop flight simulations along the specified launch azimuth using maximum and minimum burnout times. The simulated trajectories are based on kick angles obtained from the maximum and minimum range entries of the delta matrix.

2-16. MISSILE SYSTEM SIMULATION (MSS).

2-17. In order to permit system studies of missile flight, the MSS program permits simulations under special test conditions. An optional function available to MSS is the extraction of range safety data.

2-18. SPECIAL FLIGHT SIMULATION (SIM). The major purpose of the simulation program is to perform flight simulations with versatility. SIM may also make corrections to Common locations using the DEC control area. Subprogram SIMNT operates the various modes of simulation available individually or in combinations as specified by a special control card. The six possible modes of simulation are as follows:

a. Closed loop simulation

b. Open loop simulation

c. Real-earth ballistic flight simulation from re-entry vehicle separation to either point of atmospheric re-entry or detonation

d. Real-earth re-entry simulation from point of atmospheric re-entry to point of detonation

e. Ideal-earth ballistic flight simulation from any arbitrary point in space to either point of impact or target aim point

f. Perturbed flight simulation

2-19. The closed loop and open loop modes of operation differ in the method of generating steering orders and cutoff commands during the guidance phase. The closed loop mode uses the guidance equations while the open loop mode uses a table of predetermined steering orders and specified times. However, both modes simulate the real-earth missile flight from launch and can be terminated at either point of re-entry vehicle separation, point of atmospheric re-entry, or point of impact.

2-20. Perturbed missile flights are made by changing the missile and/or system parameters prior to execution of a special function. This is accomplished by changing up to 20 selected groups of constants by some multiple of the standard deviation for the constant. These flights can be used for either system studies or as the basis for generating range safety data.

2-21. For each mode of operation it is also possible to change missile and/or system parameters. This is accomplished by use of special input cards which can alter the contents of registers in the Common Area as desired.

2-22. RANGE SAFETY DATA EXTRACTION (RSD). Certain range safety data pertinent to the missile flight simulations is performed by the MSS function for simulation. The optional production of such range safety data is a separate function under control of subprogram RSDNT.

2-23. Data previously stored on special magnetic tapes describes the trajectories simulated by MSS. The data stored on these tapes is read and processed by RSD. The outputs of RSD are printed out for specified times during the powered phase of flight and include missile position and velocity data in range safety coordinates and instantaneous impact point data in geodetic latitude and longitude.

2-23A. SYSTEM FLOW DIAGRAMS.

2-23B. Figures 2-1 through 2-1F illustrate the overall flow diagrams for the system. These include Radar-Launcher Tape Generation, Missile-Trajectory Tape Generation, Binary Tape Generation, Offset Target Computation, Output Data Verification, Target Accessibility Area Determination, and Missile System Simulation and Range Safety Data. Each step in the flow diagram is numbered and all inputs and outputs necessary for the appropriate step are shown.

2-23C. The subprograms required to perform the action described in each step are listed in their respective calling sequence below the dotted line. The order of enclosures used to differentiate the sequence is braces, { }, brackets, [ ], and parentheses, ( ). Each subprogram that is called will eventually return to the subprogram that called it. An example of a calling sequence is SWAP {CLØØP [ØLØØP (LØCALT, FSIMLC {FLYER [PØLY (ACCELR), RØLL (ACCELR)], INTERP}, RANGLE, PRESET, MØNTØR)]}. The arrow direction indicates the calling and returning of the various subprograms:


```
SWAP → CLØØP → ØLØØP → LØCALT
                 ↙
               ØLØØP → FSIMLC → FLYER → PØLY ⇄ ACCELR
                                  ↙
                               FLYER ⇄ RØLL ⇄ ACCELR
                        ↙
                   FSIMLC ⇄ INTERP
                 ↙
               ØLØØP → RANGLE
                 ↙
               ØLØØP → PRESET
                 ↙
               ØLØØP → MØNTØR
                 ↙
SWAP ← CLØØP ← ØLØØP
```


 Changed 31 May 1962