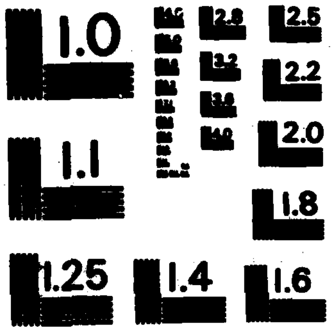
AD-A120 877 DESIGN NOTEBOOK FOR NAVAL AIR DEFENSE SIMULATION SURVEILLANCE SUPPLEMENT..(U) TRW DEFENSE SYSTEMS GROUP MCLEAN VA WATERWHEEL PROGRAM OFFICE.. 1/2

UNCLASSIFIED A G PURVES ET AL. 15 OCT 82 F/G 22/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

SSP-GOO-OG-0058-82
15 OCTOBER 1982

2

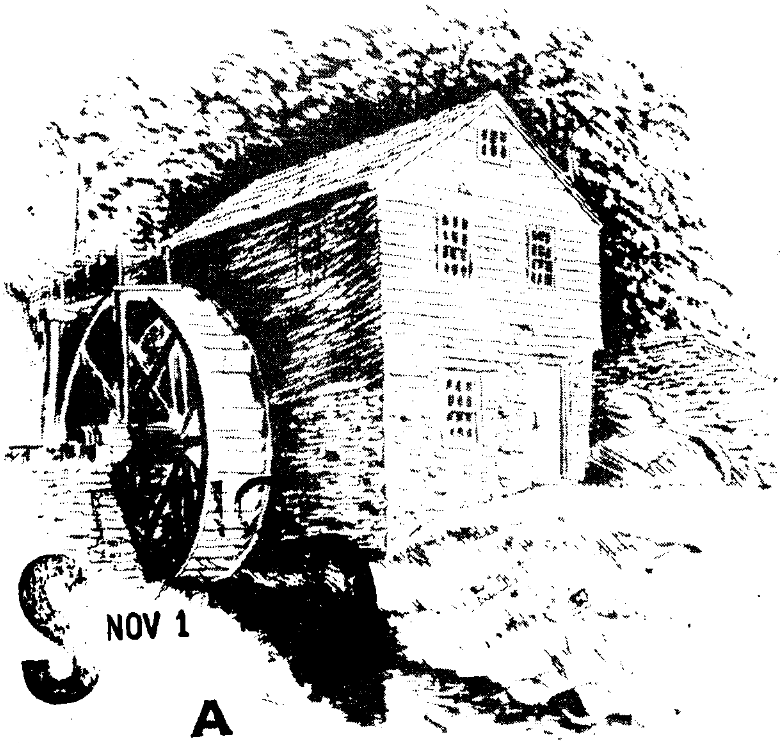
UNCLASSIFIED

SURVEILLANCE SUPPLEMENT
TO

DESIGN NOTEBOOK
FOR
NAVAL AIR DEFENSE SIMULATION

AD-A120 477

NADS



SPECIAL
PROGRAMS

TECHNICAL REPORT

This document has been approved for public release and sale; its distribution is unlimited.

PREPARED FOR
OFFICE OF THE CHIEF OF NAVAL OPERATIONS
OP-654

TRW
DEFENSE SYSTEMS GROUP
WATERWHEEL PROGRAM OFFICE
7600 COLSHIRE DRIVE • MCLEAN, VIRGINIA 22101

SSP-G00-0G-0058-82

SURVEILLANCE SUPPLEMENT

TO

DESIGN NOTEBOOK

FOR

NAVAL AIR DEFENSE SIMULATION (NADS)

FINAL REPORT

15 OCTOBER 1982

Prepared by:

A.G. Purves
L.T. Francis

Submitted by:

J. Sliney

Contract N00014-81-C-0226

TRW
WATERWHEEL PROGRAM OFFICE

FOREWARD

This document is a supplement to the Design Notebook for Naval Air Defense Simulation, SSP-G00-0G-0049-82 dated 15 September 1982. The supplement is a detailed description of the AAW Surveillance Simulation which was originally developed for OP-654 and was expanded under the current contract to model surveillance by radar as well as ELINT satellites of naval and air forces. Base watching, various forms of barriers and flight following tactics have been provided. Space based surveillance vehicles are limited to circular orbits.

INTRODUCTION

TRWs Waterwheel Program office has been developing the COVERAGE simulation for OP-654, OP-96, other customers, and under company IR&D funding for several years. Its current purpose and capability is to simulate detections of aircraft, by active and passive satellites and OTH radars. The COVERAGE simulation requires several additional programs as pre-processors and post-processors. The purpose of this documentation is to describe its capabilities and purposes, its relationship to other programs, and to give a sufficiently detailed description of the code to permit computer programmers to modify the code.

At this writing, the COVERAGE simulation is undergoing modifications as a part of current Waterwheel studies. In some cases, especially for subroutines BRBLD and BRRIER, the documentation describes modifications about to be made to the code rather than the current code which will shortly be obsolete.

TABLE OF CONTENTS

SECTION PAGE

TABLE OF CONTENTS (Cont.)

THIS PAGE INTENTIONALLY LEFT BLANK

1. OVERVIEW

1.1 BRIEF DESCRIPTION OF COVERAGE SIMULATION

(U) COVERAGE is a time-stepped Monte Carlo simulation which models detections of aircraft and ships by active and passive satellites. A high level flow diagram is shown in Figure 1-1. Satellite and target (i.e., aircraft and ship) motion is modeled explicitly. At each time step the position of the satellites and targets are updated and tests are performed to see if any of the satellites detect any of the targets. The tests for passive detection are:

- is the target emitting?
- does the emitter illuminate any spacecraft?
- is a ground station visible to the illuminated spacecraft?

The tests for active detection are:

- is the target emitting?
- does the emitter illuminate any spacecraft?
- is a ground station visible to the illuminated spacecraft?
- is the target in the spacecraft's detection annulus (of which the outer edge is power limited and the inner edge is clutter limited)?
- for aircraft targets, does the doppler return from the aircraft exceed the return from the background clutter?
- is the radar beam pointed at the target? (Note that this simulation models mechanical scanning only and not phased array.)
- a random draw against the probability of detection.

Active and passive satellites can be modeled simultaneously so that, for example, an active satellite can be cued to look at a target that was initially detected by a passive satellite.

(U) In each replication of a Monte Carlo run, the targets commence their courses at a random time, so that the satellites are in a different

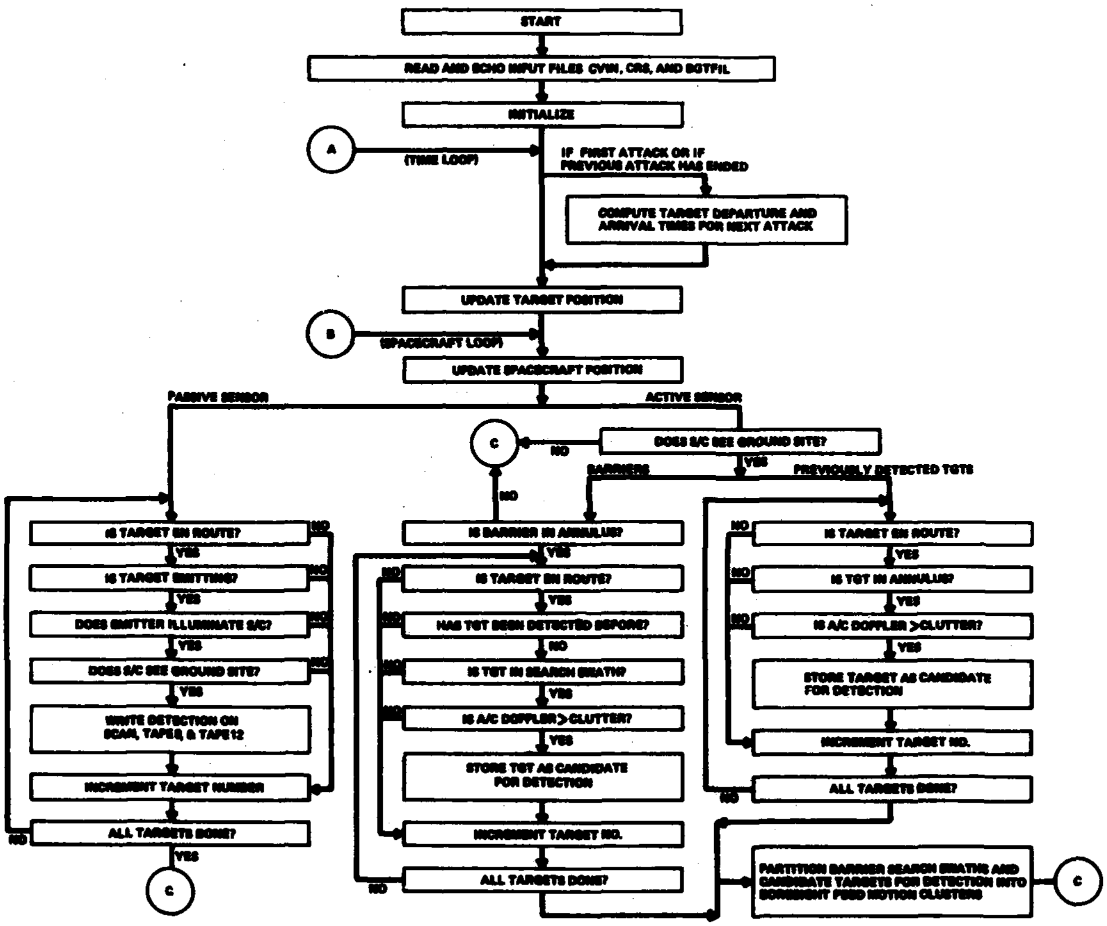

START
READ AND ECHO INPUT FILES CVIN, CRS, AND BGTFIL
INITIALIZE
A
(TIME LOOP)
IF FIRST ATTACK OR IF PREVIOUS ATTACK HAS ENDED
COMPUTE TARGET DEPARTURE AND ARRIVAL TIMES FOR NEXT ATTACK
UPDATE TARGET POSITION
B
(SPACECRAFT LOOP)
UPDATE SPACECRAFT POSITION
PASSIVE SENSOR
ACTIVE SENSOR
C
DOES S/C SEE GROUND SITE?
NO
YES
BARRIERS
PREVIOUSLY DETECTED TGTS
IS TARGET EN ROUTE?
IS TARGET EMITTING?
DOES EMITTER ILLUMINATE S/C?
DOES S/C SEE GROUND SITE?
WRITE DETECTION ON SCAN, TAPE8, & TAPE12
INCREMENT TARGET NUMBER
ALL TARGETS DONE?
IS BARRIER IN ANNULUS?
IS TARGET EN ROUTE?
HAS TGT BEEN DETECTED BEFORE?
IS TGT IN SEARCH SWATH?
IS A/C DOPPLER > CLUTTER?
STORE TGT AS CANDIDATE FOR DETECTION
INCREMENT TARGET NO.
ALL TARGETS DONE?
IS TARGET EN ROUTE?
IS TGT IN ANNULUS?
IS A/C DOPPLER > CLUTTER?
STORE TARGET AS CANDIDATE FOR DETECTION
INCREMENT TARGET NO.
ALL TARGETS DONE?
PARTITION BARRIER SEARCH SWATHS AND CANDIDATE TARGETS FOR DETECTION INTO BORESIGHT FEED MOTION CLUSTERS
YES
NO

Figure 1-1. (U) Flowchart for COVERAGE (U) (Page 1 of 2)

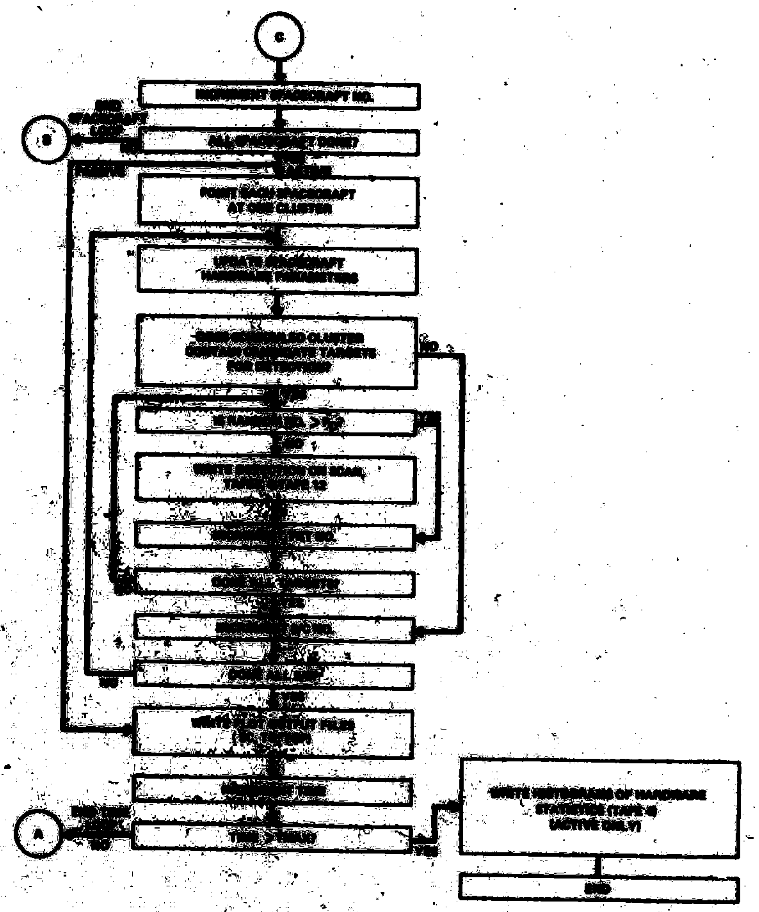

C
INCREMENT SPACECRAFT NO.
END SPACECRAFT LOOP
B
ALL SPACECRAFT DONE?
POINT EACH SPACECRAFT AT ONE CLUSTER
UPDATE SPACECRAFT HARDWARE PARAMETERS
NO
YES
NO
YES
NO
END TIME LOOP
A
TIME > TMAX?
YES
WRITE HISTOGRAMS OF HARDWARE STATISTICS (TAPE 4) (ACTIVE ONLY)
END

Figure 1-1. (U) Flowchart for COVERAGE (U) (Page 2 of 2) position when the bombers take off and when ships arrive in an area of interest.

(U) The simulation requires three input files and generates three types of output (see Sections 1.3 and 1.4). The input files describe:

- o satellite and target characteristics (COVIN)
- o target courses (CRS)
- o density of background aircraft traffic (BGTFIL)

The output files describe:

- o time of detection and target position when detected (OBS, DETIM, SCAN, COVECH)
- o satellite hardware performance (SCAN and COVHST)
- o data for input to a graphics program (SCPLT, TGPLT, OVPLT).

(U) COVERAGE is used in conjunction with other programs (see Figure 1-2). They are:

PROFIL - generates CRS input files for COVERAGE and AAWSIM

CRSGEN - creates a graphics input file from a CRS file

STATCV - processes DETIM to generate statistics on coverage gaps

WRNCOV - processes OBS to compute probability of initial detection as a function of target distance along course

AAWSIM - a one-on-one Monte Carlo simulation using CRS and OBS to model engagements between an interceptor assisted by early warning aircraft and a bomber in which the interceptor and early warning aircraft are launched and vectored based on external surveillance reports.

MAPPRO - a graphics program using DISSPLA subroutines used to draw snapshots of COVERAGE and AAWSIM scenarios.

(U) All these programs are written in FORTRAN. The most current versions of all these programs compile, load, and execute on CDC computers

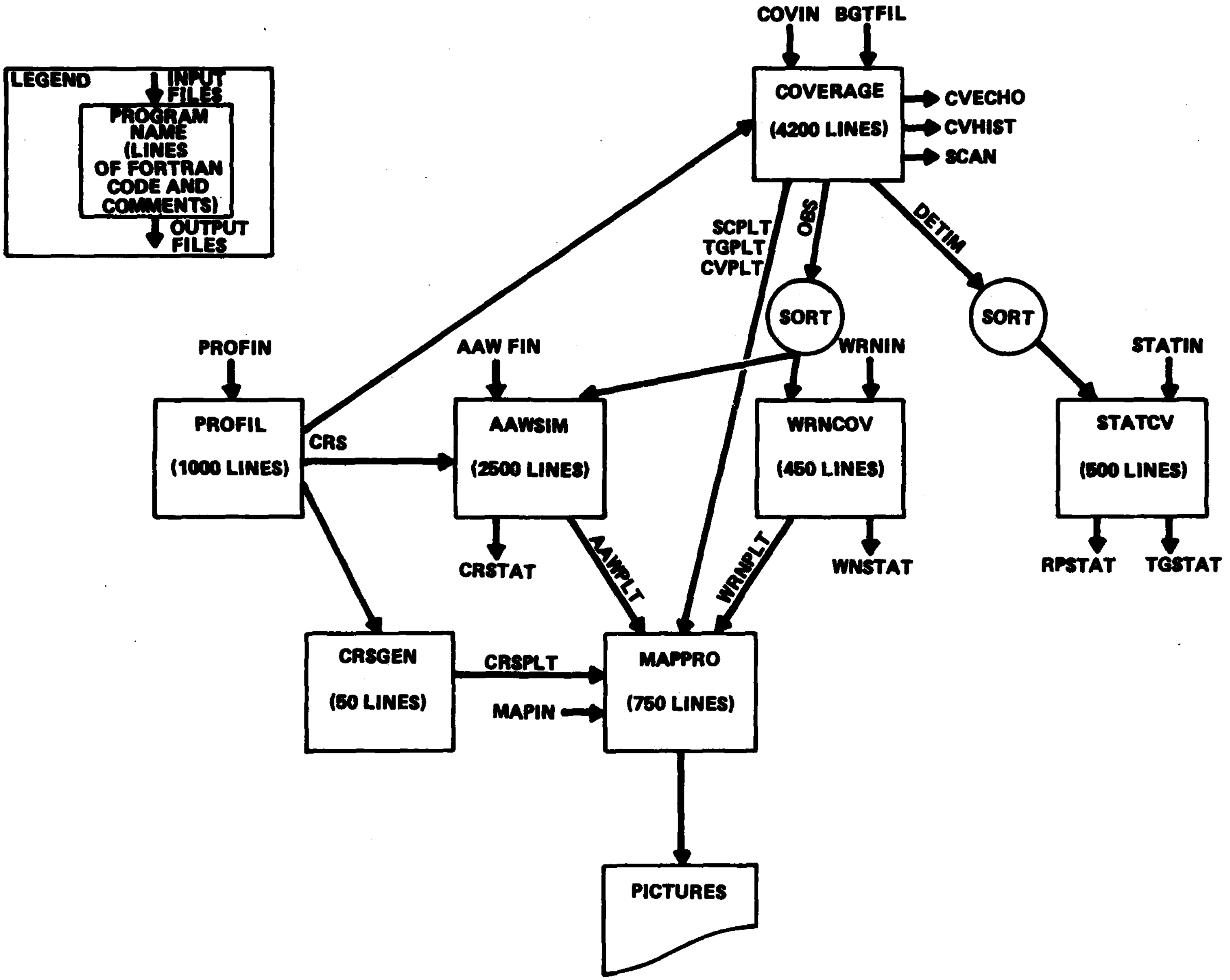


Figure 1-2. (U) Relationship of Coverage to Other Programs (U)

and on DEC VAX/780 VMS computers. Earlier versions of COVERAGE, AAWSIM, and COVSTAT have compiled, loaded, and executed on a MODCOMP Classic III computer.

(U) These programs have been under development for several years with funding from:

- o the Systems Analysis Division of the Chief of Naval Operations Program Planning Office (OP-96)
- o the Strategic and Theatre Nuclear Warfare Division of the Chief of Naval Operations Plans, Policy and Operations Office (OP-65)
- o contractor IR&D funds

1.2 PURPOSE OF THE COVERAGE SIMULATION

(U) The overall purpose of the COVERAGE simulation and related programs is to assess the utility of candidate surveillance architectures to Naval and Air Force anti-air warfare, primarily against the Soviet Backfire bomber, and to naval anti-surface warfare, primarily to support over-the-horizon targeting of the Tomahawk anti-ship missile. Production runs can be made to study mission utility as a function of:

- o the number of satellites
- o satellite orbital parameters
- o satellite performance characteristics
- o number and location of ground stations
- o C^3 time late
- o tradeoffs between satellites and non-space surveillance systems
- o barrier placement (active satellites only)
- o tasking (active satellites only)

(U) Passive satellite performance is modeled in terms of the spacecraft-target geometry required for detection. Active satellite performance is modeled in terms of slew angles and antenna feed motion required for mechanical scanning, energy balance, and the number of targets that can be tracked simultaneously.

(U) Non-space surveillance systems that can be modeled (and the program that models them) are:

- Early warning aircraft, e.g., the Navy E-2C and the Air Force E-3A (AAWSIM)
- Interceptors, e.g., F-14 and F-15 (AAWSIM)
- High frequency Over-the-Horizon Backscatter (OTH B) radar (to be added to COVERAGE)
- DEW Line radars (to be added to COVERAGE).

(U) Tasking of active satellites addresses how many carrier battle groups and how many bomber approaches can be adequately covered simultaneously.

(U) Mission utility for anti-air warfare, or more specifically, Naval Anti-air Warfare (AAW) and Air Force Airborne Tactical Warning (ATW) to defend North America, is quantified in terms of

- the probability of initial detection of a raid as a function of distance from the carrier battle group or from CONUS targets (quantified using COVWRN)
- the probability that an interceptor can visually identify and/or shoot at a bomber beyond some keepout range (quantified using AAWSIM)
- coverage gap statistics (quantified using COVSTAT)

Mission Utility for Naval Anti-Surface Warfare can be quantified, currently, in terms of coverage gap statistics and probability of initial detection.

COVIN Example

```
   TEST       10 SEP 82  DOVDEVT                <TITLE1
  1   4              1            <NATTAK,NCRS,LSEED
  F   T                           <BRTST,SIMARR   7/28
    .2000     .1000     .1000     <TLIM1,TLIM2,R1
    .4150   50.0000   4.0000      <R2,THRESH,BELEV
    .2540  0.90      0.80         <BW,PD,PC         6/07
  2   1   1   0   0   1           <IPRINT,IPRNT2,IFLG1,IHEAD,IPLOT,IPLOT2 6/29
  F   1                           <LAA,NSCAN
   0.0000    0.0000               <TPRINT(1:2)
  2   2   10.0000                 <NSW,ISW,TMAX
  4   T     .1000                 <NSC1,LTYP1,DTIM1
5.4897942E+07  0.00     45.00    0.00    0.00    0.00    4.00   50.00  <1
5.4897942E+07  0.00     45.00  180.00    0.00   90.00    4.00   50.00  <2
5.4897942E+07  0.00     45.00    0.00    0.00  180.00    4.00   50.00  <3
5.4897942E+07  0.00     45.00  180.00    0.00  270.00    4.00   50.00  <4
10.00  7.00  1.00 70.00 0.700          <BATCAP,PSOLAR,PSPC,PPLON,PPLOFF
  75.   30.   50.   5.0  .750          <CLSWID,TGTTRN,TGTSPD,DCPKPW,DUTCYC
 7.50 17.50 1.750  .750  .750 1.750    <SRCHRT,TTURN,TRKCYC,EFF,REQPEN
    0.     .25    .50    .75   1.00   1.25   1.50   1.75    <TIMTRK(1,1)
    0.      5.    12.    20.    30.    42.    57.    75.    <TIMTRK(1,2)
   7.5  7500.                                               <PWRTIM(1,1)
  1.75  750.                                                <PWRTIM(1,2)
  0   F     .1000                 <NSC2,LTYP2,DTIM2
  1   0   0   1                   <NSMCON,ITSTOR(1:2),NUMCT
       12         1         8         9        11         0         0   ISM-1
 12                               <NTRGTS
    0   40.00  180.00    0.00    0.00    0.00    1.00    1.00    0.00  <TG 1
    1    3.00   60.00    0.00    0.00    0.00    0.00   -1.00    0.00  <TG 2
    1    2.00   60.00    0.00    0.00    0.00    0.00   -1.00    0.00  <TG 3
    1    1.00   60.00    0.00    0.00    0.00    0.00   -1.00    0.00  <TG 4
    0   65.00    0.00    0.00    0.00    0.00    1.00    1.00    0.00  <TG 5
    0   43.00   15.00    0.00    0.00    0.00    2.00    2.00    0.00  <TG 6
    1    4.00   60.00    0.00    0.00    0.00    0.00   -2.00    0.00  <TG 7
    0   43.00   45.00    0.00    0.00    0.00    2.00    2.00    0.00  <TG 8
    0   20.00  155.00    1.00    0.00    0.00   -1.00    0.00    0.00  <TG 9
    0   40.00   25.00    1.00    0.00    0.00   -2.00    0.00    0.00  <TG 10
    0  -15.00  -45.00    1.00    0.00    0.00   -3.00    0.00    0.00  <TG 11
    0   20.00  155.00    1.00    0.00    0.00   -4.00    0.00    0.00  <TG 12
  4   0   1  12                   <RESET NSC1,NSC2,NSMCON,NTRGTS (IF NECESSARY)
```

CRS Example

```
  501     570       70         1      COURSE 1  TGT:     42.33  -74.98
0.00    69.00    32.00     0  500   0.00     0.00     0.00   -42.94   0.0
 .63    72.50    20.00     0  500    .63   316.51   316.51   -87.08   0.0
2.27    68.50   -20.00     0  500   1.64   820.11  1136.62  -129.33   0.0
5.03    50.00   -48.00     0  500   2.76  1380.33  2516.95  -133.41   0.0
6.18    43.00   -57.50     0  500   1.15   574.91  3091.86  -105.32   0.0
7.28    40.00   -69.00     0  500   1.10   547.67  3639.52  -100.27   0.0
7.85    39.00   -75.00     0  500    .57   284.64  3924.16  -104.09   0.0
  501     550       50         2      COURSE 2  TGT:     44.83  -92.97
0.00    69.00    32.00     0  500   0.00     0.00     0.00    -3.88   0.0
1.63    32.50    25.00     0  500   1.63   816.58   816.58   -29.23   0.0
2.62    86.00   -60.00     0  500    .98   491.75  1308.33  -140.52   0.0
5.84    60.00   -95.00     0  500   3.22  1611.65  2919.98   175.29   0.0
8.07    41.50   -93.00     0  500   2.23  1114.47  4034.45   176.85   0.0
  501     560       60         3      COURSE 3  TGT:     41.33 -103.98
0.00    56.00   160.00     0  500   0.00     0.00     0.00   143.81   0.0
1.42    46.00   170.00     0  500   1.42   708.62   708.62    94.71   0.0
3.16    43.00  -170.00     0  500   1.75   873.72  1582.33    83.14   0.0
4.92    43.00  -150.00     0  500   1.75   877.11  2459.44    90.86   0.0
6.72    41.00  -130.00     0  500   1.80   899.25  3358.69    90.00   0.0
9.15    38.00  -104.00     0  500   2.43  1214.70  4573.40   106.72   0.0
  501     560       60         4      COURSE 4  TGT:     20.00   65.00
0.00    45.00    35.00     0  500   0.00     0.00     0.00   108.61   0.0
4.09    27.37    71.66     0  500   4.09  2045.05  2045.05   137.33   0.0
4.37    25.66    73.37     0  500    .28   137.82  2182.87  -124.59   0.0
4.44    25.33    72.84  2000  500    .07    35.00  2217.87  -124.82   0.0
4.64    23.99    70.75     0  900    .20   140.00  2357.87  -125.69   0.0
4.86    22.02    67.83     0  900    .22   200.00  2557.87  -126.83   0.0
```

BGTFIL Example

```
$ TYPE BGTFIL.DAT
     100                  BACKGROUND TRAFFIC FILE (9/9/82) COM
   0   0   0  10   0   3   0   0   0   0   9   3   0   0   0   0   0   0
   0   0   7  15   8   0  12   0   0   0   0   0   0   0   0   0   0   0
   0   0  46   0   0   0  74   9   0   0   3   0   0   0   0   0   0   0
   0   0  33  15   4   3  21   0   0   0   0   0   0   0   0   0   0   0
   0   0   0  34   8  20   3   0   0   0   0   0   0   0   0   0   0   0
   0   0   0  29 161 149   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   7  44 146 424   3   0   0   0   0   0   0   0   0   0   0   0
   0   0   7  24 189 435  25   6   0   0   0   0   0   0   0   0   0   0
   0   0   0  29 287 523  89  32   0   0   0   0   0   0   0   0   0   0
  14   0   7  10 574 594  89  14   0   0   0   0   0   0   0   0   0   0
  14   0   0  10 527 255  21  17  31   6   0   0   7   0   0   0   0   0
  14   0   0  19 130  17   6  46   6   3   3   0   7   4   0   0   0   0
  14  11   0  19  71   7   6   3   3   0   0   6  27   0   0   0   0   0
  14  11   0  24  12   0   3   0   0   6  14  34   0   0   0   0   0   0
  14   5   0   5   0   3   3   0   0   3  14   0   0   0   0   0   0   0
  14   5   0   5   0   0   3   0   0   0   0   0   0   0   0   0   0   0
  14   5   0   0   0   7  12   6   0   0   0   0   0   0   0   0   0   0
  14   5   0  15  12  20   3   6   0   0   0   0   0   0   0   0   0   0
   0   0   0  19  12  10  21   9   3   0   0   0   0   0   0   0   0   0
   0   0   0  29  24  14   0   9   6   3   0   0   0   0   0   0   0   0
   0   0   0   0  16  17  18   9   0   0   0   0   0   0   0   0   0   0
   0   0   0   5  12  31  25  12   0   0   0   0   0   0   0   0   0   0
   0   0   0   0  12  31  28   0   0   0   0   0   0   0   0   0   0   0
   0   0   0  10   4  20  28   3   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   8   0  21   3   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0  31  35   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0  28  14   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   6  23   6   0   0   0   0   0   0   0   0   0
   0   0   0  10   0   0   6   6  28  28   0   0   0   0   0   0   0   0
   0   0   0   0   8  17  18   3  17   2
```

PLT Example

```
$ TYPE TGP.DAT
101.0
180.00,  40.00, 0, 0
180.00,  49.38, 0, 0
180.00,  58.75, 0, 0
180.00,  68.13, 0, 0
180.00,  77.50, 0, 0
180.00,  86.88, 0, 0
 -0.15,  83.75, 0, 0
  0.00,  74.38, 0, 0
  0.00,  65.00, 0, 0
-40.74,  58.39, 0, 0
-53.50,  64.82, 0, 0
-73.33,  69.07, 0, 0
-98.52,  69.77, 0, 0
120.74,  66.55, 0, 0
135.67,  60.70, 0, 0
144.95,  53.45, 0, 0
150.79,  45.50, 0, 0
154.59,  37.16, 0, 0
180.00,  40.00, 0, 0
0,0,0,0

101.5
180.00,  40.00, 12, 0
180.00,  49.38, 12, 0
180.00,  58.75, 12, 0
180.00,  68.13, 12, 0
180.00,  77.50, 12, 0
180.00,  86.88, 12, 0
 -0.15,  83.75, 12, 0
  0.00,  74.38, 12, 0
```

COVHST Example

```
                VAX-11 DEBUG Version 2.3-5

%DEBUG-I-INITIAL, language is FORTRAN, module set to 'COV'
DBG>GO
ROUTINE START AT COV
 0.0000000E+00   1.000000     UPDATE BGT DATA.
 0.0000000E+00          1  ECLIPSED.
 0.1000000              1  ECLIPSED.
  1.000000       2.000000     UPDATE BGT DATA.
  1.700000              2  ECLIPSED.
  1.800000              2  ECLIPSED.
  1.900000              2  ECLIPSED.
  2.000000       3.000000     UPDATE BGT DATA.
  2.000000              2  ECLIPSED.
  2.100000              2  ECLIPSED.
  2.200000              2  ECLIPSED.
  2.300000              2  ECLIPSED.
  3.000000       4.000000     UPDATE BGT DATA.
  4.000000       5.000000     UPDATE BGT DATA.
  5.000000       6.000000     UPDATE BGT DATA.
  6.000000       7.000000     UPDATE BGT DATA.
  7.000000       8.000000     UPDATE BGT DATA.
  8.000000       9.000000     UPDATE BGT DATA.
  9.000000       10.00000     UPDATE BGT DATA.
  9.700000              2  ECLIPSED.
  9.800000              2  ECLIPSED.
  9.900001              2  ECLIPSED.
  10.00000       11.00000     UPDATE BGT DATA.
  10.00000              2  ECLIPSED.
  10.10000              2  ECLIPSED.
  10.20000              2  ECLIPSED.
  10.30000              2  ECLIPSED.


CASE CPU TIME =     160.3
TOTAL CPU TIME =    161.9
FORTRAN STOP
%DEBUG-I-EXITSTATUS, is '%SYSTEM-S-NORMAL, normal successful completion
DBG>EXIT
```

1.3 INPUT

1.3.1 BGT FIL Input File

The BGT FIL is a file describing worldwide background aircraft traffic densities. There are 24 records, each record giving the densities at 000 HRS GMT, 0100 HRS GMT, ...,2300 HRS GMT for 10° x 10° latitude-longitude bins. Densities are in number of aircraft per million square miles. Each record has 17 rows corresponding to 10° latitude increments from ± 90° to - 90° and 36 columns corresponding to 10° longitude bins from 0° to 360°.

OBS Output File

The observation output file is written on (unit 12) every time a sensor detects an aircraft target. Data written are:

Column 1	Time of detection (hours) since simulation start;
Column 2	Number of sensor making the detection;
Column 3	Course number;
Column 4	Attack (replication) number;
Column 5	Target number (row on TARGTS array);
Column 6	Target latitude (degrees);
Column 7	Target longitude (degrees);
Column 8	Time (hours) since target began course;
Column 9	Bearing in degrees (relative to north) from target to sensor;
Column 10	Distance (n.mi.) of target from destination;
Column 11	1 - target is classified; 0 - target is not classified;

1.3.2 COVIN Input File

NATTAK	Integer	-	Number of replications;
NCRS	Integer	-	Number of aircraft and ship courses to be read in from CRS file;
LSEED	Integer	Odd number	Random number seed;
BRTST	Logical	Always FALSE	Obsolete;
SIMARR	Logical	TRUE	All bombers arrive at missile launch points simultaneously;
		FALSE	All bombers take off simultaneously;
TLIM1	Real	Hours	(Passive sensor only) Emission duration immediately after aircrafts take off;
TLIM2	Real	Hours	(Passive sensor only) Aircraft emission duration immediately prior to missile launch;
R1, R2	Real	Hours	(Passive sensor only) Aircraft emitter usage ratio between emissions at beginning and end of flight. Emitters are used R1 hours out of every R2 hours;
THRSH	Real	Knots	(Active sensor only) Minimum discernable velocity;
BELEV	Real	Degrees	(Active sensor only) Outer annulus edge at which aircraft barriers are searched (earth grazing angle);
BW	Real	Degrees	(Active sensor only) Beamwidth;
PD	Real	0 - 10	(Active sensor only) Probability of detection for single dwell;
PC	Real	0 - 10	(Active sensor only) Probability of classification - obsolete;
TPRINT	Real	Hours	Obsolete;
TPRINT2	Real	Hours	Obsolete;

IFLG1	Integer	0	Do not compute rotation of node;
		≠ 0	Compute rotation of node;
IHEAD	Integer	0	Suppresses input echo on unit 6;
		1	Prints input echo on unit 6;
IPLOT	Integer	0	Suppresses output on SCPLT and TGPLT files (in main program);
		1	Prints output on SCPLT and TGPLT files (in main program);
IPLOT2	Integer	0	Suppresses output on DETIM file (in SETL)
		1	Prints output on DETIM file (in SETL);
LAA	Logical	TRUE	(Active sensor only) Debug output on SRT, AA, and CLM arrays;
		FALSE	Suppresses debug output;
NSCAN	Integer	-	Replication for which SCAN, CVPLT, SCPLT, and TGPLT output is desired. Zero if no output is desired for these files;
TPRINT	Real	-	Obsolete
NSW	Integer	1	Satellite coverage area is a circle (one edge);
		2	Satellite coverage area is an annulus (two edges);
ISW	Integer		Method of inputing satellite coverage area edges;
		1	Earth central angle for each edge;
		2	Earth grazing angle for each edge;
		3	Spacecraft look angle (from nadir) for each edge;
TMAX	Real	Hours	Time for each replication or attack; should be longer than the longest course time;
NSC1	Integer	1-8	Number of type 1 sensors;
LTYP1	Logical	TRUE	Type 1 sensors are active;
		FALSE	Type 1 sensors are passive;

DTIM1	Real	Hours	Time increment for Type 1 spacecraft;
OESC			Orbital parameters for spacecraft or dimensions of coverage area for OTH radars and DEW Line;
			Orbital parameters (initial conditions);
OESC(1, J)	Real	Feet	Length of semi-minor axis;
OESC(2, J)	Real	0.-1.0	Eccentricity;
OESC(3, J)	Real	Degrees	Inclination;
OESC(4, J)	Real	Degrees	Longitude of ascending node;
OESC(5, J)	Real	Degrees	Argument of perigee;
OESC(6, J)	Real	Degrees	Mean anomaly;
			OTH B/DEW Line Dimensions;
OESC(1, J)	Real	Degrees	Latitude of coverage center;
OESC(2, J)	Real	Degrees	Longitude of coverage center;
OESC(3, J)	Real	Degrees	Heading (from north) from center of coverage area axis of symmetry (or bisector);
OESC(4, J)	Real	Degrees	Half angle of width of coverage area;
OESC(5, J)	Real	n.mi.	Radius to inner edge;
OESC(6, J)	Real	n.mi.	Radius to outer edge;
ORBPAR(NSC, 1-2, ISW)	Real	Degrees	Satellite coverage area dimensions (See ISW);
BATCAP	Real	Kilowatt Hrs (KWH)	(active satellite only) Maximum storage capacity of battery;
PSOLAR	Real	KW	(active satellite only) Average power input due to insolation;
PSPC	Real	KW	(active satellite only) Average power required for base load of spacecraft;
PPLON	Real	KW	(active satellite only) Power required with payload on;
PPLOFF	Real	KW	(Active satellite only) Power requied with payload off;

CLSWID	Real	Degrees	(active sensor only) Maximum feed angle from boresight (rotational degrees about suborbital point);
TGTTRN	Real	Degrees	Maximum expected aircraft turn;
TGTSPD	Real	Knots	Maximum expected target speed change;
DCPKPW			Obsolete;
DUTCYL			Obsolete;
SRCHRT(3)	Real	Degrees/Minute	(active sensor only) Search rate for barrier search for: 1 Aircraft barrier at outer edge of annulus; 2 Aircraft barrier at inner edge of annulus; 3 Ship barrier at outer edge only;
TTURN	Real	Minutes	(active sensor only) Feed turn time during search;
TRKCYL	Real	Hours	(active sensor only) Track cycle time (minimizes energy usage);
EFF			Obsolete;
REQPEN	Real	Ratio	Feed turn penalty for reacquisition;
TIMTRK			Obsolete;
PURTIM			Obsolete;
NSCZ	Integer	2-20	Number of type 2 spacecraft;
LTYPZ	Logical	FALSE	Type 2 spacecraft must be passive;
DTIMZ	Real	Hrs	Time step for type 2 spacecraft;
NSMCON	Integer	-	Number of ground station sets;
ISMTAB	Ground station set specifications;		
ISMTAB(1, J)	Integer	1-15	Set Number;
ISMTAB(2, J)	Integer	-	Target one start number;
ISMTAB(3, J)	Integer	-	Target one end number;
ISMTAB(4, J)	Integer	-	Target two start number (first G/S);

ISMTAB(5, J)	Integer	-	Target two end number (last G/S);
ISMTAB(6, J)	Integer	-	Spacecraft start number (0 = any S/C);
ISMTAB(7, J)	Integer	-	Spacecraft end number (0 - any S/C);

Notes on ISMTAB;

1. Target numbers refer to rows on TARGTS array;
2. Ground stations should be entered sequentlially at the end of the TARGTS array;
3. Entries 1-3 are used for passive sensors only and;
4. If active satellites are modelled, ISMTAB refers to active satellite ground stations only;

ITSTOR	Integer	Zero	Never use;
NUMCT	Integer	Always 1	Number of simultaneous observation by spacecraft required for detection;
NTRGTS	Integer	1-35	Number of targets to be input to the TARGTS array;

TARGTS	Real	Input area for:
		o Aircraft
		o Ships
		o Aircraft barrier;
		o Ship barrier;
		o Ground stations;	

TARGTS Input for aircraft and ships;

TARGTS(J, 1) 1 Aircraft raid count;

1 Ships raid count;

TARGTS(J, 2) Course number;

TARGTS(J, 3) (Passive sensor only) Emitter scan width in azimuth;

TARGTS(J, 4) (Passive sensor only) Emitter elevation limits 1 or 2;

TARGTS(J, 5) (Passive sensor only) Emitter's lower elevation limit (grazing angle degrees above local horizon);

TARGTS(J, 6) (Passive sensor only) Emitter's upper elevation limit (grazing angle degrees above local horizon);

TARGTS(J, 7) Course start time (hours) - computed by program. Input a zero;

TARGTS(J, 8) For passive sensors, input time (hours) after takeoff at which emitter first turns on;
For active sensors, enter any negative number; it will be replaced with the time of initial detection;

TARGTS(J, 9) Time (in hours) of course duration; is computed by program; input a zero;

Aircraft and Ship Barriers:

Note: Ship barriers must be circles centered at a carrier battle group; aircraft barriers can be circles or annular sections centered anywhere;

TARGTS(J, 1) Must be zero;

TARGTS(J, 2) Latitude (degrees) of barrier center;

TARGTS(J, 3) Longitude (degrees) of barrier center;

TARGTS(J, 4) Inner radius (n.mi.) of barrier (must be zero for circular barrier);

TARGTS(J, 5) Outer radius (n.mi.) of barrier;

TARGTS(J, 6) Bearing (degrees from north) of annular barrier axis of symmetry;

TARGTS(J, 7) Barrier ID. Must be an odd number from 1 to 19 for aircraft barrier and any integer from 21 to 30 for ship barriers;

TARGTS(J, 8) Barrier width (degrees);

TARGTS(J, 9) Barrier type (1 for aircraft, 2 for ships);

Ground Stations:

TARGTS(J, 1) Must be zero;

TARGTS(J, 2) Latitude (degrees);

TARGTS(J, 3) Longitude (degrees);

TARGTS(J, 4) Elevation coverage limits (1 or 2);

TARGTS(J, 5) Earth Grazing angle (degrees) of lower elevation limit);

TARGTS(J, 6) Earth Grazing angle (degrees) of upper elevation limit;

TARGTS(J, 7) ID; Must be any negative integer (entered as a real);

TARGTS(J, 8-9) Not used;

Revision data:

I	Integer	≤ NSC1	Number of type 1 sensors to be modelled;
J	Integer	≤ NSC2	Number of type 2 spacecraft to be modelled;
IDO	Integer	≤ NSMCON	Number of ground station sets to be modelled;
K	Integer	≤ NTARGTS + NSMCON	Number targets to be modelled;

1.3.3 COURSE FILE

(U) This file defines all bomber flight paths to be simulated in terms of state vectors. Each state vector consists of:

1) Time of course change (hours)
2) Lattitude of the course change (degrees)
3) Longitude of the course change (degrees)
4) Target acceleration after course change (knots/hr.)
5) Target velocity after course change (knots)
6) Time since last course change (hours)
7) Distance flown since last course change (nautical miles)
8) Total distance flown thus far (nautical miles)
9) Target heading after course change (degrees)
10) Distance from carrier (nautical miles)

A separate program (PROFILE) generates elements one through nine for each the state vectors. AAWSIM computes the tenth element of each state vector.

1.4 OUTPUT

1.4.1 DETIM Output File

DETIM is written on (unit 8) by subroutine SETL every time a new contact (or detection) begins of a barrier, aircraft, or (for passive sensors only) a ground station. Data written are:

Column 1 Target number (rows or TARGTS array);

Column 2 Attack/replication number;

Column 3 Time (hours) since aircraft began course or since (for ground stations and barriers) replication began;

Column 4 Down or gap time (hours) since end of last contact and start of this contact;

Column 5 0 - target has not been classified;
1 - target is classified;

1.4.2 TGPLT, SCPLT, and CVPLT Output Files

These files have a special format required by the MAPPRO graphics program. This format is

```
FILID
data rows
0,0,0,0
FILID
data rows       repeated an unlimited number of times
0,0,0,0
   :
End-of-file
```

The format for each data row is LONGITUDE, LATITUDE, symbol number, heading where all angles are in degrees. Symbol numbers are the following integers:

1	Causes a DISSPLA symbol to be drawn for each data row;
0	A line connects all the points indicated on the data rows;
- 1	Do not use;
- 2	Do not use;
- 3	Draws Backfire symbol;
- 4	Draws F-14 symbol;
- 5	Draws CV symbol;
- 6	Draws OTH B symbol;
- 6	Draws other symbols satellites, RPV, E-2C, or an "X";

FILIDs must be ascending real numbers. Free format is used.

TGPLT writes a data row to show the current position of every bomber at every simulator time step. The FILID is the simulator time at which the data row is written.

SCPLT writes a data row at every time step for

- o suborbital point of each satellite;
- o boresight aimpoint for each cluster (active satellite only)

o intersections of annulus edges with aircraft and ship barrier edges;

Again, the FILID is the simulator time at each time step. CVPLT writes scenaric data rows. Data is indicated by the FILID.

FILID	DATA
1 - 50	Aircraft and ship courses;
51 - 100	Destination of aircraft and ships;
101 - 150	Barrier centers;
200	Aircraft and ship positions when detected;

1.4.3 SCAN Output Files

The SCAN file shows the status of each sensor at each time step, especially by indicating which targets are observed. Data written for each sensor are:

Column 1 Time (hours);

Column 2 For each target in TARGTS array o if not detected;
9 if aircraft cannot be detected because velocity component along line of sight is less than the minimum discernable velocity (active satellite only);
7 aircraft (ship) not detected because probability of detection is less than one (active satellite only);
A-Z indicates
- o barrier aircraft, or ship is detectable;
- o cluster ID;

Column 3 # indicates a scheduling conflict
blank indicates no scheduling conflict (active satellite only);

Column 4 letter indicates which cluster was scheduled;(The remaining data are for active satellites only.)

Column 5 Boresight slew angle since last time step (rotation angle degrees about boresight);

Column 6 Off-boresight feed angle required to scan cluster (rotation angle degrees about suborbital point);

Column 7 Aircraft track count (the number of aircraft being tracked at current time step);

Column 8 Battery state charge (0 to 100%);

Also, there is a summary column, which shows for each target.

8 target is detected
0 target is not observable
0 I 8 target is observable by I satellites but is not detected because of scheduling conflicts;

THIS PAGE INTENTIONALLY LEFT BLANK

2.0 OVERVIEW OF MAIN PROGRAM

This chapter describes the functions performed by the main program code. The code is referred to in terms of the FORTRAN DO/LOOP and other labels and subroutines. The main program is long and performs much of the simulation logic. The overall structure of the main program (called COV) is:

Declaration Statements	
Comments describing inputs	
Data statements	
FORMAT statements	
Input and initialization	(labels 10-200)
Time loop	(labels 200-1000)
Sensor loop	(labels 295-600)
Target detection loops	(labels 380-500)

Target detection loops include logic or subroutine which test for the following sensor - target combinations. Under the LOCATION heading, names refer to subroutines and numbers refer to FORTRAN labels and DO LOOPs in the main program.

SENSOR	TARGET	LOCATION
active satellites	aircraft barriers	BARRIER (in LOOP 385)
	aircraft	BARRIER, LOOP 500, and 828-855
	ship barriers	BARRIRS
	ships	BARRIRS, LOOP 500, and 828-855)
	ground stations	355-380
passive satellites	aircraft, ships, and ground stations	LOOP 500
OTH, B radar & DEW/LINE	aircraft	FXSENS

Other important sections of code in COV, and the associated sensor, if applicable, are:

SENSOR	FUNCTION	LOCATION
--	Calculate start & end time for each replication and takeoff times for aircraft	200-210
all satelites	Solar eclipsing	220-230 ECLPSE
	Update position of ships and aircraft at each time step	260-295
all satellites	Update spacecraft position (in terms of suborbital point)	300-320 CORBIT
active satellites	Test for ground station visibility	355-380
active satellites	Identify candidate search areas at which boresight should be pointed	FSTEP 500-580, including CLUSTR
passive satellites	Test for ground station visibility	LOOP 625
active satellites	Decide at which search area to point the boresight of each spacecraft	CLSTPP, just after 675
--	Output SCAN and plotting files	900-997

The following test describe COV sequentially from label 10 to the end in greater, but not complete, detail:

LABEL/LOOP/SUBROUTINE	DESCRIPTION
LOOP 10	Rewind files.
COMBLK	Initialize earth gravitational constant (GMU), earth rotation rate (WE) and earth radius (RE) in distance, time, angle units of feet, hours, and degrees.
SECOND	Save current value of CPU time.

ZERRV
ZERIV
Set arrays to zero.

20-30
Read COVIN file (describes sensors and scenarios).

40-45
Echo COVIN file on COVOUT.

45-50
Determine time step (DTIME) and IRATIO, which makes it possible toupdate and test the two types of satellites at different time steps (DTIM1 and DTIM2). TIME is subsequently set to zero.

LOOP 75
Read the CRS file (describes routes followed by aircraft and ships) and compute distance between starting point of each aircraft or ship course and the location of the carrier or base that the aircraft or ship is attacking.

LOOP 87
Write aircraft/ship courses on COVPLT file.

LOOP 90
Write locations of carriers or bases that aircraft or ships are attacking on COVPLT file.

LOOP 100
Echo CRS file on COVSUM file.

105-140
Write header for SCAN file. Contents of header is determined by the number of sensors and the number of targets (i.e., aircraft and ship barrier, aircrft, ships, and ground stations.

LOOP 155
Spacecraft initialization. DNODEH is procession of the mode in degrees. XM is spacecraft velocity and HA and HP are apogee and perigee.

LOOP 165
Converts some entries of the TARGTS array from real to integer.

165-185
BRBLD
BRBLDS
UPR1
JDORGF
LINTRP
BRBLD and BRBLDS set up arrays that are referred to in processing aircraft and ship barriers. UPR1 a uniform random number generator and is called with JDORGF to compute a random starting date for eclipsing. Remaining code initializes variables for computing active satellite energy balance. LINTRP is a linear interpolation routine.

185-200	Compute TMAX, the time at which the simulation ends.
200	Start of time loop, in which sensor and target positions are updated and tests are made to see if any sensors detect any targets.
200 to 220	For each attack (i.e., replication), compute simulated time at which this replication begins (SATTAK) and ends (TATTAK) and at what time each of the aircraft/ships begins its course (TARGTS (NTAR,7)).
220-230 BGTINP ECLPSE	BGTINP updates clock and reads in new aircraft background density data if hour has changed. ECLPSE checks for solar eclipsing of satellite.
235-250	Initialization required each time the time loop is updated.
250-295	Update position of moving targets and DELTIM of fixed and moving targets. DELTIM is the difference between the current TIME and the time the replication began (for fixed targets) or the time the aircraft or ship began its course (for moving targets).
260-263 BGNCRS ENDCRS	Check if aircraft/ship has begun or ended course. BGNCRS is called just before an aircraft begins its course so that coverage statistics will include the detection or gap occurring when the aircraft begins its course. ENDCRS is called for the same purpose just after an aircraft ends its course. Otherwise, if for example, an aircraft course took three hours and the first detection did not occur until two hours into the flight, that two hour gap would not be inlucded in CVSTAT statistics if BGNCRS were not used.
263-275	Interpolation between two state vectors from the CRS bit to get aircraft/ship position, speed, and heading.
275-295	When target happens within six minutes from a state vector, position, course, and speed are obtained directly from the state vector instead of from interpolation.
LOOP 500	Sensor loop. Performs spacecraft, OTH B, and DEW Line tests for current time.

295-300 Ships sensor loop when needed if the two types of spacecraft are tested at different time increments (DTIM1 and DTIM2).

300-320
CORBIT Update spacecraft suborbital point for current time step.

320-350 Compute earth central angle for inner and outer edge of spacecraft detection annulus based on spacecraft altitude, and satellite look angle from nadir (BETA) or earth elevation angle above horizon (EPS).

355-370 Test if active satellites have a ground station visible. If not, skip rest of sensor loop.

FXSENS If current sensor is an OTH B radar or the DEWLSC line barrier, test for detection of aircraft and then ship to end of sensor loop.

LOOP 385
BRRIER Test ifany aircraft barriers can be searched at current time by current spacecraft active. If it is, store boundaries of search area in SRT array for spacecraft scheduling.

LOOP Test if any ship barriers can be searched at current time by current spacecraft, if ctive. If so, store search area boundaries on SRT array for spacecraft scheduling.

LOOP 500 For active satellites, tests for detection of aircraft that have already been initially detected, either in a barrier, by a passive satellite, by the DEW line, or by an OTH B radar. For passive satellites, test for detection of a ship or aircraft and for ground station visibility.

405-425 Retrieve target position, course, speed and DECTIM from TPOS.

425-440 Test is target is in annulas by comparing earth central angles between suborbital point and target and suborbital point and nular edges.

445-465
SETL

Complete detection tests for passive satellites, namely: is aircraft/ship emitting? If emitting, is it emitting in the direction of the aircraft?

Emitter on/off usage is described by TLIM1, TLIM2, R1, and R2 (see Table). If tests are passed and ground station visibility is not being tested, the detection is written as CVSCAN, CVOBS, and SETL is called. SETL updates coverage statistics array for current replication and may cause write on CVTAP8 (input to CVSTAT). If ground station visibility is being tested,then target observability is indicated on the IRFTAB array. If current target is a ground station, then its observability is noted in ISCAN and IRFTAB and SETL is called.

465-475
JMEQ
FSTEP

Perform doppler test for active detection of aircraft by calling JMEQ. JMEQ computer DDOP, the component of aircraft velocity along the line-of-sight between the target and the spacecraft. For detection to occur, the absolute value of DDOP must exceed the doppler return from clutter, which is input as THRESH. If the aircraft is detectable, then FSTEP is called to store aircraft position for subsequent use in deciding where to point the boresight (in subroutines CLUSTR and CLSTPP).

500

End sensor loop.

500-565

By now, the SRT array holds the spacecraft look angle (from radar) and the bearing (relative to spacecraft velocity) for each target and for each search area end point detectable from the current spacecraft. The AA array is now computed. It stores the suborbital-point-centered angle between every pair of targets (and search area end points) observable by the current spacecraft. This information is used by CLUSTR. If search area endpoints exist, then the search area ID is stored on the AA diagonal, which otherwise would be zero.

565-570

Optional listing of AA for debug purposes.

570-580
CLUSTR

Computes boresight pointing options for current spacecraft by partioning detectable targets and search area into clusters. Clusters are groups of targets and search areas which can be scanned froma fixed boresight position by moving the antenna feed. After CLUSTR has been called for all spacecraft, then CLSTPP is called (after label 678) to decide where to aim the boresight of each spacecraft. For fixed sensors, i.e., OTH B and DEW line, all observable targets are assumed to be in one cluster and the CLUSTR functions are performed in FXSENS.

580-630
SETL

Ground station logic for passive spacecraft. The ISMTAB array specifies that particular spacecraft (ISMTAB(6-7,I)) can report detections of particular target (ISMTAB(2-3,I)) only if the spacecraft sees a target and particular ground station locations (ISMTAB(4-5,I)) simultaneously. For targets and ground stations, ISMTAB refers to rows on the TARGTS/ITRGTS array. If detection occurs, SETL is called to update coverage statistics and, perhaps, write on CVTAP8, JSCAN is set, and the latitude and longitude of the observed target is written on tape 8.

650-675

ITSTOR is never used.

675-LOOP 815
CLUSTR

Decide at which cluster to aim active satellite boresight. This process is called scheduling. CLSTPP knows the clusters of all spacecraft.

LOOP 815

Update 'he ISCAN summary (last column) to show how many sensors detect each target.

LOOP 875

Each sensor has now been assigned, or scheduled, to look at a particular cluster. This loop updates active spacecraft hardware requirement parameters (state of change, feed angle, lew angle, and number of aircraft to be tracked) and determines, based on a random draw against an input probability of detction, whether a particular ship or aircraft is detected. If detection occurs, ISCAN is set, CVOBS is written on, and SETL is called.

D0875 - D0855
STATUD

SCSCH (3,IDO) stores the number of the scheduled clusters (in terms of a row on the CLM/CLM2/CLM3 arrays). STATUD is called twice to update histogram statistics. The first call updates the number of aircraft that must be tracked, which is the sum of the background and threat aircraft in search areas and in (smaller search) areas where threat aircraft are being reacquired and tracked. The second call updates the ratio of the number of aircraft that must be tracked (track requirements) to the number of aircraft that there is time to track (track capacity). A message is written whenever the ratio is greater than one.

D0855 - 830
UPRI (TSS)
RAN (VAX)

For each aircraft or ship target, a uniform random draw is made to test against an input probability of detection (PD). If the draw is greater than PP, detection does not occur and ISCAN is set to -7, which shows up in the SCAN printout as a +7.

830-855
SETL

If an aircraft is being detected (in a search area) for the first time,the time of the initial detection is stored in the eighth element of the TARGTS/ITRGTS row for that target. For detected targets ISCAN is set to the number of the scheduled cluster, SETL is called, and CVOBS is written on.

855-870
STATUD

Based on the boresight aim points at the current and previous time steps, spacecraft slew angle (centered at the suborbital point) is computed and stored in CLBG (5,5, NSC) and STATUD is called to update slew angle histogram statistics. STATUD is also called to update histogram statistics on cluster width (the suborbital point centered angle between the targets or search area corners at the edges of a cluster).

870-875
ECLPSF
PWRBAL
STATUD

Update power balance statistics for current spcecraft. ECLPSF is called to determine if current spacecraft is eclipsed, in which case solar panels are not charged.

875-900

For targets currently being detected, set time of last detection(TARGTS(NTAR,11)) to current time.

90-995

Write on output files CVSCAN, CVPLT, CVTG, and CVSC if DOUT is true. DOUT prevents writes at the smaller time step in DTIM1 and DTIM2 are different.

900-920
CTOALF

Write on CVSCAN file. CTOALF converts positive numbers stored in ISCAN to alphabet letters. Different SCAN formats are used depending on whether one or two types of spacecraft are being simulated. CVSCAN shows, for each sensor, which targets are detected, the different clusters, which cluster is scheduled, and hardware requirements parameters.

920-955

Writes longitude and latitude of current position of detected moving targets on CVPLT file.

955-985

Writes longitude, latitude, and heading of suborbital points of spacecraft on CVSC file. If active spacecraft are being simulated, the longitude and latitude of boresight aim points are written. If active spacecraft are simulated and their annulus edges intersect barriers (thus creating search areas), the longitudes and latitudes of these intersections, which are the search area endpoints, are also written on SCPLT.

985-995

Write longitude, latitude, and bearing of all moving targets at current time step on TGPLT file.

995-1000

End of time loop. If TMAX is exceeded, exit loop and end simulation. Otherwise, increment the number of time steps (ITIME), time, clock, and date.

1000-1050
STATPT
SUMMRY

Call final statistics subroutines, write endfiles, rewind files, and stop. STATPT writes histogram dates on tape 4 and SUMMRY computes coverage statistics for last replication only and writes then on CVECHO.

THIS PAGE INTENTIONALLY LEFT BLANK

3. SUBROUTINES

3.1 INPUT

3.1.1 SUBROUTINE BRBLD

PURPOSE:

Reads input data about aircraft barriers from targets array and stores it in BRREF/IBRREF for processing aircraft barriers in BRRIER. Barriers are modeled as circles or annular sections.

INPUTS:

NTRGTT — Number of input rows to TARGTS array inputs for aircraft barriers;

TARGTS (NTAR,1) — Must be zero if row contains barrier information;

TARGTS (NTAR,2) — Longitude of barrier center (deg);

TARGTS (NTAR,3) — Latitude of barrier center (deg);

TARGTS (NTAR,4) — Radius to inner edge of barrier (n.mi.);

TARGTS (NTAR,5) — Radius to outer edge of barrier (n.mi.);

TARGTS (NTAR,6) — Bearing of bisector of barrier (deg. from north);

TARGTS (NTAR,7) — Fixed target ID; for aircraft barriers must be an odd number from 1 to 19;

TARGTS (NTAR,8) — Barrier width (half angle);

TARGTS (NTAR,9) — Not used

For circular aircraft barriers, TARGTS (NTAR,4) must be zero and the sixth and eighth entries are not read. Annular barriers must have two rows on the TARGTS array, the first of which must have the inputs described above and the second of which must have the first entry set to zero are the seventh set to the same barrier ID.

TARGTS array inputs for aircraft.

TARGTS (NTAR,1) — Must be greater than zero;

TARGTS (NTAR,2) — Course number in CRS input file;

TARGTS (NTAR,3) — Not used

TARGTS (NTAR,4) — Not used

TARGTS (NTAR,5) — Not used

TARGTS (NTAR,6) — Not used

TARGTS (NTAR,7) — Time after start of replication that bomber begins course, is not an input but is computed by program;

TARGTS (NTAR,8) Set to any negative number. If and when aircraft is detected in a barrier, this entry is set to the time at which initial detection occurs.

TARGTS (NRAT,9) Length of flight (hours);

OUTPUTS

NUMBR Number of aircraft barriers in TARGTS array;

IBRREF (1,IBR) Barrier ID;

IBRREF (2,IBR) Number of (first) row in TARGTS array with information about barrier center, radii, and bearings;

IBRREF (3,IBR) Number of (second) row in TARGTS array, with barrier ID only;

IBRREF (4,IBR) Latitude of pole of a great circle going through the barrier center and defining one side of the barrier;

IBRREF (5,IBR) Longitude of pole first described;

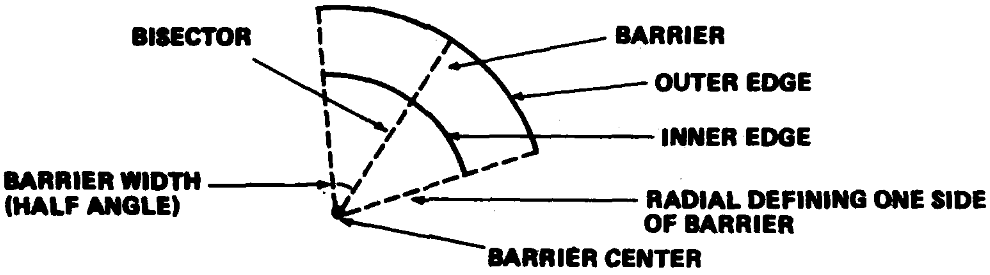
IBRREF (6,IBR) Latitude of pole of great circle of radial defining other side of barrier;

IBRREF (7, IBR) Longitude of pole just described;

Has a shorter current detection period than that of any of the targets seen by the other cluster;

METHOD:

Aircraft barriers are defined as annular sectors:



This subroutine searches the TARGTS array to build the BRREF array. The poles for the radials are computed by adding 90 to the bearing of a radial and finding a point 1/4 of an earth radius away from the center along that bearing. If the inner radius is zero, then a circular barrier is assumed and no poles are calculated. The data described here are used in subroutine BRRIER.

3.1.2 SUBROUTINE BRBLDS

PURPOSE:

Sets up a reference array, BREFS, for use by subroutine BRRIRS in processing ship barriers. Like BRBLD, which is for aircraft barriers, BRBLDS gets the data for the reference array by searching the TARGTS array. Ship barriers are models as circles centered at a carrier battle group with an input radius specified on the TARGTS array. They are searched only at the outer edge of the annulus. Ship barriers are used to model radar satellite searches of enemy ships for providing over-the-horizon targeting of Tomahawk.

INPUTS:

For ship barriers, TARGTS must have the following inputs:

TARGTS (1,NTAR)	Zero;
TARGTS (2,NTAR)	Latitude of barrier center;
TARGTS (3,NTAR)	Longitude of barrier center;
TARGTS (7,NTAR)	Barrier ID; must be a number from 21 to 30 (implies a maximum of ten ship barriers);
TARGTS (8,NTAR)	Radius of barrier (n.mi.);

To input ships, TARGTS must have:

TARGTS (NTAR,1)	A negative number;
TARGTS (NTAR,2)	Number of course input in CRS file;
TARGTS (NTAR,7)	Time, (in hrs) at which ship begins its course (computed by program);
TARGTS (NTAR,8)	Time at which ship is initially detected (computed by program);
TARGTS (NTAR,9)	Duration of ship course (hrs);

OUTPUTS:

NUMBRS	Number of ship barriers input in TARGTS array;
IBREFS (1,IBR)	Barrier ID number;
IBREFS (2,IBR)	Row in TARGTS array describing this barrier;
IBREFS (3,IBR)	Not used
IBREFS (4,IBR)	Not used

IBREFS (5,IBR)	Not used
BREFS (6,IBR)	Latitude of barrier center (degrees);
BREFS (7,IBR)	Longitude of barrier center (degrees);
BREFS (8,IBR)	Barrier radius (n.mi.);
BREFS (9,IBR)	Barrier radius (degrees);

METHOD:

As part of program initialization, BRBLDS tests every row of the TARGTS array for a zero in the first entry and a number between 21 and 30 in the seventh entry. If these tests succeed, then the row describes a ship barrier and the appropriate information is put into the BREFS array.

3.2 SUBROUTINE CORBIT

Subroutine CORBIT calculates the suborbital point and the velocity of the suborbital point.

Spacecraft position and velocity in geocentric inertial spherical coordinates are translated into earth geographical coordinates. The elements used to describe the circular orbit for an earth satellite are listed in table 1.

CORBIT first calculates the orientation of the orbit plane, that is, the longitude of the ascending node at the time of interest,

ON = TIME(DNODE-WE) + OO

and the argument of perigee; the rotation of theorbit within the orbital plane

PM = (XNU * TIME + OM) MOD 360°

where MOD is the remainder function.

The position of the satellite at the time of interest thus calculated is translated to earth coordinates.

TABLE 1

Symbol	Element	Definition
DI	Inclination	Angle between orbit plane and earth's equatorial plane
XNU	Mean Angular Motion	Angular velocity in the orbital plane y/a^3
A	Distance	Distance from earth center to the satellite
DNODE	Precession of ascending node	Angular velocity of the orbit plane about the line normal to the earth's equatorial plane and through the origin (earth center)
PM	Argument of perigee	The rotation of the orbit within the orbital plane
OO	Right ascension of the ascending node	Longitude of ascending node at reference time
OM	Mean anomaly at epoch	360° x $\Delta\tau/p$ where P is the period and t is the time difference between the epoch and the last perigee passage before epoch
To	Epoch	Time for which elements are specified
ON	Longitude of ascending node	Orientation of the orbital plane at the time of interest
WE	Earth rotation rate (degrees/hour)	
GMU	Gravitational constant	

The coordinate of the point is (PM, 0) in inertial coordinates

In earth reference the old North pole is at $(\mathrm{ON}-90^\circ, 90-\mathrm{DI})$, the coordinates of the point in the new system $Q_o' = 90^\circ$

$\mathrm{COS}(90 - \mathrm{LAT}) = \mathrm{COS}(\mathrm{DI}) * 0 + \mathrm{SIN}(\mathrm{DI})\mathrm{COS}(\mathrm{PM} - 90)$

$\mathrm{SIN}(\mathrm{LAT}) = \mathrm{SIN}(\mathrm{DI})\ \mathrm{SIN\ PM}$

$\mathrm{LAT} = \mathrm{SIN}^{-1}\quad \mathrm{SIN}(\mathrm{DI})\mathrm{SIN\ PM})$

$\mathrm{SIN}(\mathrm{LON}-90) = \mathrm{SIN}(90 - \mathrm{PM})\ \mathrm{SIN}\ 0^\circ/\mathrm{SIN}(90 - \mathrm{LAT})$

$\mathrm{COS}(\mathrm{LON}) = \mathrm{COS}(\mathrm{PM})/\mathrm{COS}(\mathrm{LAT})$

$\mathrm{LON} = \mathrm{COS}^{-1}\left\{\mathrm{COS}(\mathrm{PM})/\mathrm{COS}(\mathrm{LAT})\right\}$

3.3 ACTIVE SENSOR LOGIC

3.3.1 SUBROUTINE BGT

PURPOSE:

This subroutine is responsible for computing the background traffic in a specified area.

INPUT ARGUMENTS:

AREA, ALAT, ALON, BGTDEN

Three calling sequence inputs are used to define the area of interest, and a fourth variable, the background density array, is provided through X8 Common.

AREA	Area of region in square miles;
ALAT	Latitude representative of region in degrees;
ALON	Longitude representative of region in degrees;
BGTDEN	Background traffic density data, stored in an 18 x 36; array, in aircraft per million square miles. (BGTDEN is contained in COMMON X8);

OUTPUT ARGUMENTS: TRAFIC

The aircraft count resulting from the computations of this routine are returned via the calling sequence variable TRAFIC.

TRAFIC	Number of aircraft in the region.

METHOD:

This routine computes the aircraft count in a region based upon the area and the density associated with the region. The region's latitude and longitude are converted into array subscripts by the following formulae:

```
LAT =    9  - 1F1X(1.E-6 + ALAT/10.)    if ALAT ≥ 0.
         10 - 1FIX(1.E-6 + ALAT/10.)    if ALAT < 0.
LON =    18 + 1F1X(1.E-6 + ALON/10.)    if ALON ≥ 0.
         19 + 1F1X(1.2-6 + ALON/10.)    if ALON < 0.
```

The traffic count is then computed by the following statement, which includes the conversion from aircraft per million square miles:

```
TRAFIC = BGTDEN(LAT,LON) x AREA/10.E6
```

REFERENCES:

The routine is called from BGTSCN and CLUSTR.

3.3.2 Subroutine BGTINP

PURPOSE:

This routine is responsible for keeping the background traffic data current with the simulation time. The background traffic file BGTFIL contains densities by hour for the ten degree square segments covering the surface of the globe. As model time progresses, the density data must be revised for each hour.

INPUT ARGUMENTS:

CLOCK, BGTTIM, BGTDEN

The routine receives the simulated clock time (24 hour, with fractions of an hour) through the calling sequence. It maintains BGTTIM, the time associated with the current density data set, and BGTDEN, the 18 by 36 array of background traffic densities, the density data set. These latter two items are input from the BGTFIL on FORTRAN logical unit number 20 and are maintained in COMMON X8.

CLOCK	24.00 hour simulation clock time
BGTTIM	24.00 hour time, in hourly increments, associated with BGTDEN
BGTDEN	Background traffic density, stored in an 18 x 36 array, in aircraft per million square miles.

OUTPUT ARGUMENTS:

BGTTIM, BGTDEN

These arguments are the current time and the current background traffic density data as updated from the BGTFIL as necessary. They are defined above.

METHOD:

The CLOCK time, maintained in the main program, is compared to the BGTTIM value. Initially, BGTTIM is set to -10.E20. If CLOCK is greater than BGTTIM, the next record for the BGTFIL file (BGTTIM and BGTDEN) is read and the times are again compared. If BGTTIM exceeds 24., the file is rewound before it is read.

REFERENCES:

This routine is called from COV, the main routine, early in the time looo.

3.3.3 Subroutine BGTSCN

PURPOSE:

This routine is responsible for accumulating the quantity of background traffic in a scan area bounded by points A and B along a spacecraft annulus of depth DISTDP.

INPUT ARGUMENTS:

The following is a list of input arguments passed by the argument list:

SCLAT Spacecraft's suborbital point latitude in degrees;

SCLON Spacecraft's suborbital point longitude in degrees;

ALAT Latitude of one scan area boundry in degrees;

ALON Longitude of one scan area boundry in degrees;

BLAT Latitude of other scan area boundry in degrees;

BLON Longitude of other scan area boundry in degrees;

S Earth central angle of annular radius in degrees;

LINNER Logical variable, TRUE for inner annulus, FALSE for outer annulas search;

DISTDP Depth of search area in degrees

OUTPUT ARGUMENTS: TOTTRF

TOTTRF Total number of background aircraft traffic found in the scan area.

METHOD:

A series of ten degree wide subregions is defined within the scan area. The midpoints of these subregions and their areas are used to define the density blocks for routine BGT. The final subregion may be less than ten degrees in width, being the remainder of the region. The area of a subregion is defined to be depth times the width times the degree-to-nautical-mile conversion squared.

REFERENCES:

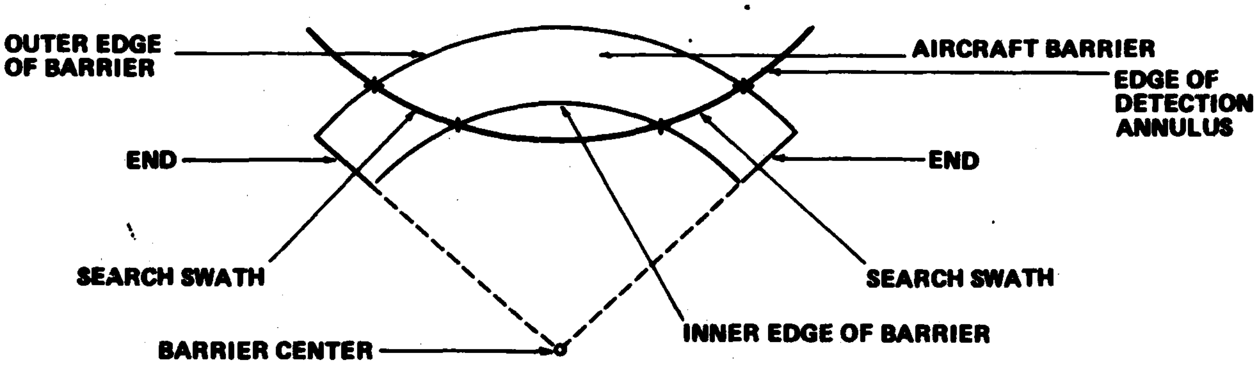
BGTSCN is called by BRRIER Subroutine.

BGTSCN calls BGT, BRNG, DGLIM, FLALOB and TRI.

3.3.4 SUBROUTINE BRRIER

PURPOSE:

Determines if a satellite detection annulus edge intersects an aircraft barrier. If there is an intersection, BRRIER tests to see if any aircraft are detected in the search swath along the annulus edge and between the two points of intersection. Because aircraft barriers can be annulur, a satellite detection annulus edge can intersect a barrier twice.



Aircraft barriers are searched at the inner and outer edge of the annulus.

INPUTS:

BRREF	Array: See BRBLD for description;
TARGTS	Array: See BRDLD for description;
IBR	Aircraft barrier ID;
BELEVO	Outer annulus edge at wheel barriers are searched (degree earth grazing angle);
BELEVI	Inner edge of annulus at which aircraft barriers are searched (degrees, earth grazing angle);
TPOS (1,NTAR)	Aircraft latitude;
TPOS (2,NTAR)	Aircraft longitude;
TPOS (4,NTAR)	Time since aircraft began course;
BW	Beamwidth of radar beam;

OUTPUTS:

NACS — Number of aircraft barriers being searched by current spacecraft;

NTSCH — Number of aircraft and ship barriers being searched by current spacecraft;

Locations of annulus barrier intersections defining search swaths are stored in:

PT (1, NTSCH, NSC) Latitude of first intersection;
PT (2, NTSCH, NSC) Longitude of first intersection;
PT (3, NTSCH, NSC) Latitude of second intersection;
PT (4, NTSCH, NSC) Longitude of second intersection;

TPOS (1, BROW) Latitude of first intersection;
TPOS (2, BROW) Longitude of first intersection;
TPOS (3, BROW) Latitude of second intersection;
TPOS (5, BROW) Longitude of second intersection;

Where BROW is

- first row in TARGTS array with the barrier ID (for the first swath)
- second row in TARGTS array with this barrier ID (for the second swath)

Note: TPOS is changed every time the sensor loop is incremented while PT is changed only when the time loop is incremented. PT is used to write latitudes and longitudes of swath limits on the SCPLT file.

RES (1, NISCH) Width of search swath, measured in spacecraft centered degrees;
RES (2, NISCH) Search cycle time (minutes);
RES (3, NISCH) Available dwell time for a single bean (minutes);
RES (4, NISCH) Width of search swath, measured in rotational degrees about suborbital point;

RES (5, NISCH) Annulus edge bordering search swath (earth central angle in degrees);

RES (6, NISCH) Number of aircraft (includes background traffic plus threat aircraft) that are in search swath and that need to be tracked.

METHOD:

BRRIER performs three functions for each aircraft barrier for each spacecraft sensor at each time step:

1. Determines if the inner or outer edge of the annulus intersects the barrier; thus creating a search swath;
2. Completes the entries in the RES array for each search swath.;
3. Determines if there are any threat aircraft in the search swath being detected for the first time;

If there are two search swaths, then the ID for the first one is the odd number input as the barrier ID in the TARGTS array and the ID of the second one is the input ID number plus one.

To determine if there is a search swath, BRRIER uses CONES to test for the intersections of the annulus edge with the following four circles:

1. The circle defining the inner edge of the barrier;
2. The cirle defining the outer edge of the barrier;
3. The great circle defining each end of the barrier. See description of BRBLD.

If these intersections exist, they are then tested against the barrier specifications input on the TARGTS array. This is done by transforming the intersections to a new coordinate system whose pole is the barrier center and whose zero meridian is the barrier's bisector (or axis of symmetry). After this transformation, the colatitude and longitude of each intersection are the distance from the barrier center and the bearing relative to the barrier bisector. Then, for an intersection to be on the barrier, the colatitude must be on or between the inner and outer edges of the barrier and the bearing must be less than or equal to the half-angle of the barrier width (see BRBLD). Intersections passing these tests are input to the SRT array for scheduling by calling FSTEP.

The entries in RES are then computed for each swath. Search swath width in rotational degrees is computed by converting the points of intersection to a new coordinate system whose pole is the spacecraft suborbital point and whose zero meridian is along the spacecraft's inertial velocity vector. The depth of the search swath is the depth of the radar beam footprint or the distance traveled by the suborbital point in one simulation time step, whichever is larger. The period or time available to scan (or search) the swath is the time it takes the annulus edge to travel over a distance equal to the depth of the swath. This period is also called the search cycle time; to maintain a 100% probability of sweeping over an aircraft as it goes through the search swath, the search swath must be scanned at least once per cycle time. Available beam dwell time is the ratio of cycle time to swath width (in spacecraft centered degrees) times the beamwidth. The number of aircraft tracked is the number of threat aircraft detected (described below) plus one-half the number of background aircraft in the swath (because it is assumed that half the aircraft in a barrier are headed towards the Soviet side of the barrier and half are headed away from the Soviet side).

Finally, the tests to determine if any threat aircraft, which are input in the TARGTS array, are detected in the swath are:

1. Has the aircraft begun its course? If yes, then;
2. Has the aircraft ended its course? If no, then;
3. Is the aircraft inside the search swath? If yes, then;
4. Does the aircraft have a doppler return greater than the minimum discernable velocity?

If all these tests are passed, then the aircraft's position relative to the spacecraft is input to the SRT array by calling FSTEP.

3.3.5 SUBROUTINE BRRIRS

PURPOSE:

Models detections of ship barriers by radar satellites. A ship search swath is defined by the segment of the outer edge of an annulus that is inside a ship barrier. Because ship barriers are always circular, there is never more than one search swath per barrier.

INPUTS:

SHELEV Annulus edge at which ship searches are conducted. Expressed in terms of earth grazing angle (degrees);

IBR Number of ship barrier being tested;

BREFS ARRAY See BRBLD for description;

NSC Number of current spacecraft;

OUTPUTS:

NSHS Number of ship barriers being searched by current spacecraft;

NTSCH Number of aircraft and ship barriers being searched by current spacecraft;

Each ship search swath is bounded by the intersections of the search annulus and the circle describing the barrier. The latitude and longitude (in degrees) of each intersection is stored as follows:

TPOS (1, NTAR)	Latitude of first intersection;
TPOS (2, NTAR)	Longitude of first intersection;
TPOS (3, NTAR)	Latitude of second intersection;
TPOS (5, NTAR)	Longitude of second intersection;

PT (1, NTSCH, NSC)	Latitude of first intersection;
PT (2, NTSCH, NSC)	Longitude of first intersection;
PT (3, NTSCH, NSC)	Latitude of second intersection;
PT (4, NTSCH, NSC)	Longitude of second intersection;

RES (1, NTSCH) Width of search swath, measured in spacecraft centered degrees;

RES (2, NTSCH) Search cycle time;

RES (3, NTSCH) Available dwell time for a single beam;

RES (4, NTSCH) Width of search swath, measured in rotational degrees about the suborbital point;

RES (5, NTSCH) Annulus edge bordering search swath; expressed as an earth central angle (in degrees) from the suborbital point;

RES (6, NTSCH) Not used;

ISCAN (NTAR, NSC) Set to one if barrier is searched by spacecraft; is used on the SCAN output file;

STBER 1, STBER 1 Bearings (in degrees) from suborbital point to intersections of annulus edge and ship barrier;

ST Earth central angle between suborbital point and annulus edge at which barrier search is performed;

METHOD:

Subroutine CIRSCT is called to compute location of two intersections between annulus edge at which ship barriers are searched and circular ship barrier. If there is no intersection then CIRSCT sets IPOINT to zero, which causes BARRIRS to return immediately. If the two circles do intersect, then

- o NSHS and NTSCHS are incremented;
- o The latitude and longitude of the intersections are stored in TPOS and PT;
- o ISCAN (NTAR, NSC) is set to one, when NTAR is the row of the TARGTS array describing this barrier and NSC is the number of the current spacecraft;
- o Subroutine FSTEP is called to enter the two points of intersection into the SRT array for clustering and scheduling;
- o The first five entries in RES are computed;

3.3.6 SUBROUTINE CLUSTR

PURPOSE:

Partition all the search areas in the annulus of a single active satellite into groups (or clusters) which can be mechanically scanned by moving the antenna feed of the satellite boresight. There are two types of search areas, barrier search areas and aircraft reacquisition and tracking areas. A barrier search area is a swath, one beam footprint deep, along the edge of the annulus which is inside a barrier. Aircraft reacquisition and tracking areas are much smaller areas of uncertainty which are searched to detect aircraft that have been previously detected. For each cluster, the following are computed:

- o boresight cumpoint
- o background traffic in search areas
- o energy used to scan cluster

INPUTS:

CLSWID	degrees	Angle, centred at suborbital point and expressed in rotation degrees about that point, defining the maximum width of a cluster, i.e., the maximum width that can be scanned from a fixed boresight position by moving the antenna feed.
NTD		Total number of targets detectble in annulus. In the context, targets include: o intersections of barrier boundaries with annulus o aircraft inside annulus

AA (35,35)

A square symmetric zero diagonal array containing the suborbital point centered angle between all pairs of targets inside the annulus. For barrier search areas, aircraft not yet detected, and ships in barrier search areas, the diagonal element is reset from zero to the ID number of the search area.

NSC

Number of current spacecraft

SRT (3,35)

Stores for each target in annulus:

1. Row of target on TARGTS array
2. Bearing, relative to spacecraft velocity bearing, from suborbital point to target (degrees)
3. Spacecraft look angle, from nadir (degrees).

OUTPUTS:

CLM(36,10)
CLM2(5,10)
CLM3(25,10)

These arrays can store up to 41 items of information about each cluster:

CLM(1,ICL) Cluster number

CLM(2,ICL) Number of spacecraft viewing cluster

CLM(3,ICL) Cluster number for this spacecraft

CLM(4&5,ITCL) Bearings from spacecraft suborbital point to targets that are gratest angle apart;

CLM(6&7,ITCL)		Spacecraft look angles to see targets that are greatest angle apart. 4-6 are for the target tha is closest to the suborbital point; 5-7 are for the target that is furthest from the suborbital point;
CLM(8,ITCL)		Bearing to boresight AIM point
CLM(9,ITCL)		Spacecraft look angle to boresight AIM point;
CLM(10,ITCL)		Maximum angular separation of targets in cluster;
CLM(11,ITCL)		Number of targets in cluster
CLM(12-36,ITCL)		Number of each target in cluster; Initially the number is the target's rank in the SRT array. Maximum number of targets currently allowed in a cluster is 25.;
CLM2(1,JCL)	hours	Earliest time of last detection across all targets in the cluster;
CLM2(2,JCL)	hours	Minimum duration of current detection interval across all targets in cluster;
CLM3(1,ITCL)		Barrier number;
CLM3(2,ITCL)		Earth central angle of search (inner or outer edge of annulus) in degrees;
CLM3(3,ITCL)		Search cycle time (minutes) from barrier;
CLM3(4,ITCL)		Search amplitude in spacecraft degrees;
CLM3(5,ITCL)		Search time (minutes), hopefully less than cycle time;
CLM3(6,ITCL)		Search energy (KJ) expended during time step;
CLM3(7,ITCL)		Background traffic in search area (inc targets);
CLM3(8,ITCL)		Time P/L on during search (hrs);

CLM3(9,ITCL)		Number of targets to be reacquired;
CLM3(10,ITCL)		Energy (KJ) used to reacquire targets;
CLM3(11,ITCL)		Time (minues) to reacquire targets;
CLM3(12,ITCL)		Background traffic in reacquisition areas (inc targets);
CLM3(13,ITCL)		Background traffic in tracking areas (inc targets);
CLM3(14,ITCL)		Number of raid targets to be tracked;
CLM3(15,ITCL)		Track time/cycle time ratio;
CLM3(16,ITCL)		Maximum number of targets that can be tracked in the available track time;
CLM3(17,ITCL)		Track energy (KJ);
CLM3(18,ITCL)		Total background traffic encountered;
CLM3(19,ITCL)		Time P/L on during tracking (hrs);
CLM3(20,ITCL)		Time P/L on during reacquisition (hrs);
CLM3(21,ITCL)		Total energy used (KJ) to search, reacquire, and track targets;
CLM3(22,ITCL)		Total time P/L on during period;
CLM3(23,ITCL)		Not used;
CLM3(24,ITCL)		Not used;
CLM3(25,ITCL)		Not used;
CLCTR(2,5,8)		
CLCTR(1,IC2,NSC)	degrees	Longitude of boresight aimoint;
CLCTR(2,ICL,NSC)	degrees	Latitude of boresight aimpoint;
CLBG(5,5,9)	degrees	Stores spacecraft centered angles between all pairs of boresight aimpoints in annulus; The diagonal, which otherwise would be zero, stores:
CLBG(1,1,NSC)		Is # if there is a scheduling conflict, i.e., if this spacecraft can see more than one cluster;
CLBG(2,2,NSC)		Number of clusters in annulus;

CLBG(3,3,NSC) Number of clusters for which this spacecraft is responsible. (Another spacecraft may be able, for example, to see one of the clusters visible to this spacecraft.)

CLBG(4,4,NSC) Total angle to visit all clusters.

METHOD:

Subroutine CLUSTR partitions th targets in the AA array into clusters, which are defined and stored in the CLM, CLM2, and CLM3 arrays. Once all the targets in a cluster are identified, the boresight aimpoint (in the center of the cluster) and the energy required to scan the cluster are computed.

The following rules guide the building of a cluster:

1. A cluster contains no more than one barrier search area swath.

2. The entire swath (defined as the annulus edge between two annulus - barrier intersections) must be in one cluster, even if the resulting cluster width is greater than CLSWID. (If this occurs, the spacecraft must slew while the antenna feed is moving).

3. A cluster will generally contain an aircraft barrier or a ship barrier and several aircraft track or reacquisition search areas which are approximated by the current, true position of the aircraft.

4. Clusters are first built around barrier search swaths and then around unclustered aircraft. The first member of a cluster is called the pivot.

5. If the pivot is a barrier search swath intersection or as yet undetected aircraft or ship appearing in a barrier search swath, then all other members of that barrier search swath are included in the cluster. CLSWID is replaced by the width of the barrier search swath if it is greater than the search swath.

6. After the pivot target(s) are in the cluster, all other aircraft in the AA array are tested to see if they can also be in the cluster. To be in the cluster, the candidate aircraft must be less than CLSWID degrees away from all targets already in the cluster.

7. The process of identifying a pivot and testing all other unassigned AA array entries against that pivot continues until all entries of the AA array are clustered.

8. The diagonal element in the AA array for an individual target is set to 99 when that target has been included in a cluster.

The boresight is aimed at a spacecraft look angle that bisects the spacecraft centered angle from the inner to the outer edge of the annulus and at bearing angle that bisects the width of the cluster.

The logic to compute background traffic in search areas and the time and energy to scan a cluster is still developmental and is not correct as of this writing.

3.3.7 SUBROUTINE CLSTPP

PURPOSE:

Subroutine CLSTPP (cluster port processor) is the schedular for active satellites. If an active satellite can view more than one cluster (see description of subroutine CLUSTR), then the schedule decides at which cluster to aim the boresight.

INPUTS:

CLBG (5,5,9) — See description of subroutine CLUSTR. CLSTPP checks to see if two satellites see the same cluster, in which case the third diagonal element is decremented for one of the spacecraft and the first diagonal element may be set to a blantk.

CLM (36,10) — See description in CLUSTR. For each cluster (i.e., for each row on the CLM array), CLSTPP uses:

- CLM (1,ICL): Cluster number;
- CLM (2,ICL): Number of spacecraft viewing cluster;
- CLM (3,ICL): Cluster number for this spacecraft;
- CLM (11,ICL): Number of targets in this cluster;
- CLM (12-36): Target number (i.e., row on targets array) of each target in cluster;

Note: In this context, "target" can mean the endpoints of a barrier search swath.

CLM3 (1,ICL) — Earliest time of last detection across all targets in cluster.

CLM2 (2,ICL) Minimum duration of current detection interval across all targets in cluster.

CLM3 (1,ICL) ID number of barrier search swath

XLN (NTAR,1) Time (hours) at which last detection occurred.

XLN (NTAR,11) Time (hours) at which current detection began; is set to zero if target was not detected at last time step.

ITCL Total number of clusters seen by all active sensors (e.g. active spacecraft, OT4 B radars, and DEW line) at current time step. (By definition, there is no more than one cluster per fixed sensor.)

NSCI The number of active sensors.

OUTPUTS:

SCSCH (9,8) Shows how each active sensor is scheduled.

SCSCH (1,NSL) 1 if sensor sees no more than one cluster; 0 if sensor sees none or several clusters;

SCSCH (2,NSC) 1 if sensor has been assigned to a cluster; 0 if sensor has not been assigned to a cluster;

SCSCH (3,NSC) Row in CLM/CLM2/CLM3 arrays containing cluster to which this sensor has been assigned;

SCSCH (4,NSC) Number of cluster to which spacecraft has been assigned. This number can be less than SCSCH (3,NSC), but only if two or more sensors see the same cluster (i.e., have matching clusters).

SCSCH (5,NSC) Number of targets in cluster;

SCSCH (6 and 7,NSC) First and last rows in CLM/CLM2/CLM3 containing clusters that can be seen by this spacecraft;

SCSCH (8,NSC) Earliest time of last detection across all targets in assigned cluster;

SCSCH (9,NSC) Minimum duration of current detection interval across all targets in assigned cluster;

ICLST (9, 10) Shows which active sensors are assigned to each cluster;

ICLST (1,ICL) Total number of active sensors assigned to this cluster;

ICLST (2-9,ICL) Active sensor ID numbers of sensors scheduled to view this cluster;

ICLWID (8) Width of cluster (in terms of rotation degrees about spacecraft suborbital point) which has been scheduled for each spacecraft;

METHOD:

CLSTPP assigns or schedules active sensors to look at one cluster per time step. For OTH B radar and DEW Line this is trivial, because by definition OTH B and the DEW Line each see only one cluster. Scheduling spacecraft is harder, because, in general, each spacecraft has at each time step more than one cluster (in its annulus) that needs to be scanned. Recall that a cluster is a group of aircraft and up to one aircraft or ship barrier search swath that can be scanned from a fixed boresight position by moving the antenna feed. When an active spacecraft has a scheduling conflict, the conflict is resolved based on the following priorities (listed in descending order of importance).

1. Do not schedule a cluster if it is already scheduled to another sensor (implies that two sensor see the same cluster) unless the current sensor has in its cluster a target which:

- is not seen by the other sensor;
- has longer detection gap than the targets by the other sensor;
- has a shorter current detection period than that of any of the targets seen by the other cluster;

2. Choose the cluster that has target with the longests detection gap unless the maximum gap is less than .9 hrs.

3. Choose the cluster which has the target with the shortest detection period if:

- the target was observed at the last time step;
- The detection period is less than .3 hrs;

4. Choose the cluster that has the most targets.

It occasionally happens that a sensor is not scheduled to see a cluster - all the clusters it sees are already scheduled to other sensors. The scheduler assigns sensors in ascending order -1, 2, ..., NSCI; it does not unassign nor reassign sensors. (This would be a desirable improvement.) When a sensor is scheduled to a particular cluster, this is recorded in the SCSCH array.

3.3.8 Subroutine ECLPSF

PURPOSE:

This routine is responsible for updating the SUNVEC data to the current simulation date/time. It is called within the early stages of the Time Loop of COV. The SUNVEC is used to determine the eclipse status of the spacecraft within that Time Loop.

INPUT ARGUMENTS:

DATE	The current Julian date and time, the latter expressed as a fraction of a day.

OUTPUT ARGUMENTS:

SUNVEC	The x-, y- and z- unit vector for the sun's position relative to the Earth.

METHOD:

The algorithm is embeded in SUN and the other utility routines called.

REFERENCES:

ECLPSF is called by COV. It calls the following utility routines: SUN, OTOC, EQEC, and RMMUL.

3.3.9 Function ECLPSF

PURPOSE:

ECLPSF is a logical function intended to return a TRUE value when the spacecraft position input is in solar eclipse during the current time, and otherwise, FALSE.

INPUT ARGUMENTS:

SUNVEC	The x-, y- and z- unit vector of the sun's position relative to the Earth at the current time.
SCLON	Longitude of suborbital point in degrees
SCLAT	Latitude of suborbital point in degrees
SCRAD	Orbital radius in feet.

OUTPUT ARGUMENTS:

ECLIPSE	A logical variable, set TRUE for eclipse, otherwise, FALSE.

METHOD:

The unit vector for the spacecraft is computed and the dot product of this vector with the solar unit vector is taken. From the comparison of the result with the horizon of the spacecraft, the eclipse state can be determined.

REFERENCES:

The routine is used in COV. It calls DOT and MAGN.

3.3.10 SUBROUTINE FSTEP

PURPOSE:

Performs the first step of active satellite clustering and scheduling, which is to store the bearing and elevation angles of detectable search areas in the SRT array in order of their bearing relative to the spacecraft's velocity heading.

INPUTS: (all angles in degrees)

ST Earth central angle from suborbital point to a search area location;

STBER Bearing (from north) from suborbital point to search area location;

SVBER Spacecraft velocity bearing;

NTAR Search area ID number (row on TARGTS array that describes the search area). A search area can be a swath through an aircraft or ship barrier or the estimated current position of an aircraft.

OUTPUTS: (all angles in degrees)

SRT(1, I) Search area ID number;

SRT(2, I) Bearinq, relative to spacecraft velocity heading, from spacecraft suborbital point to search area location;

SRT(3, I) Spacecraft look angle from nadir to target;

Note: all detectable search areas are entered in the SRT array in ascending order based on bearing from suborbital point to search area. Bearings can vary from -180° to ± 180°.

NTD Number of search areas detectable by current spacecraft;

METHOD:

This subroutine is an intermediate step which is prerequisite for subroutine CLUSTR. Subroutine CLUSTR groups detectable search areas for an active satellite into groups (called clusters) which can be scanned from a fixed boresight position by moving the antenna feed. To do this, CLUSTR requires that all detectable search areas be stored in clockwise order around the annulus. FSTEP puts search areas in this order. A search area is:

- o An aircraft barrier search swath;
- o A ship barrier search swath;
- o The current position of an aircraft that has already been initially detected;

For each search swath, two points are stored in SRT, namely, the two intersections of the barrier with the annulus edge bordering the search swath. Currently, the model approximates a search area to reacquire a previously detected aircraft by the aircraft's current, actual position. This approximation is not valid if the aircraft maneuvers between detections; the search area should be based on an a position extrapolated from the last observed position and should have some area of uncertainty.

A search area is detectable if it is in the spacecraft's detection annulus.

3.3.11 SUBROUTINE FXSENS

PURPOSE:

Models detections of aircraft by fixed sensors, e.g., Over-the-Horizon Backscatter (OTH B) radar and the DEW Line. The approach is to model the areas covered by these two systems as annular sectors and to compute when aircraft are in these sectors.

INPUTS:

(All angles in degrees;)

SLAT	Latitude of annulus center;
SLON	Longitude of annulus center;
SHD	Bearing from north of axis of symmetry (or bisector) of annular sector;
SWD	Angulur width of annular sector (half angle);
SIR	Distance from annulus center to inner edge of annular sector (n.mi.);
SOR	Distance from annulus center to outer edge of annulus sector;
TARGTS (NTAR, 1)	Greater than zero if target is an aircraft;
TARGTS (NTAR, 7)	Time at which aircraft begins flight;
TPOS(1, NTAR)	Aircraft's current latitude;
TPOS(2, NTAR)	Aircraft's current longitude;
TPOS(3, NTAR)	Aircraft's current bearing;
TPOS(4, NTAR)	Time since aircraft started course;

OUTPUTS:

CLM(1, J)	Cluster number (usually equal to J);
CLM(2, J)	Sensor number;
CLM(3, J)	Number of clusters seen by this sensor (is always one for OTH B radar and DEW Line);
CLM(11, J)	Number of aircraft seen by this sensor;

CLM (12-35, J) Aircraft ID (i.e., row in TARGTS array) for each aircraft seen by sensor;

METHOD:

To determine if an aircraft is in the coverage area of an OTH B radara or the DEW Line, FXSENS checks all rows in the TARGTS array to find if:

o row represents an aircraft; if yes, then;

o has the aircraft begun its flight? if yes, then;

o has the aircraft ended its flight; if not, then the current position of the aircraft is transformed to a new coordinate system whose pole is the center of the annular coverage sector (e.g. the OTH B radar transmitter is the center of the annular sector covered by the OTH B radar system) and whose zero meridian is the axis of symmetry of the annular sector. In the new coordinate system, the aircraft's colatitude is its distance from the center and its longitude is its angulur separation from the center of the annular coverage sector. The tests to determine if the aircraft is in the annular coverage sector are, therefore:

o is aircraft's new colatitude greater than the distance to the inner edge of the coverage sector; if yes, then;

o is the new colatitude less than the distance to the outer edge of the coverage sector; if yes, then;

o is the absolute value of the aircraft's new longitude less than the half-angle width of the coverage sector;

If all these tests are passed, then the aircraft is in the coverage area of the current fixed sensor.

If the fixed sensor sees one or more aircraft, then FXSENS creates a cluster, similar to the clusters created for spacecraft by subroutine CLUSTR.. However, whereas for spacecraft there may be several clusters of which only one can be observed at a time; for OTH B radar or the DEW Line, there is only one cluster, i.e., all aircraft in the coverage area can be seen in one simulator time step. A cluster is created by making the entries shown under outputs in a new row of the CLM array. This makes it unnecessary to call CLUSTR for fixed sensors. However, subroutine CLSTPP, which schedules spacecraft to look at the highest priority cluster, uses the CLM array to assign spacecraft to clusters which do not have aircraft that are seen by a fixed sensor. Thus, fixed sensors can simplify spacecraft scheduling.

3.3.12 SUBROUTINE JMEQ

PURPOSE:

Computes an aircraft's velocity component along the line of sight from the aircraft tothe spacecraft.

INPUTS: (all angles in degrees)

ST	Earth central angle from suborbital point to aircraft position;
DBER	Suborbital-point-centered angle between spacecraft velocity bearing and great circle to aircraft position;
VT	Aircraft velocity (knots)

OUTPUTS:

DDOP	Aircraft's velocity component along the line of sight from the aircraft to the spacecraft (knots);

METHOD:

Compute the aircraft centered angle between the aircraft velocity vector and the aircraft-to-spacecraft line-of-sight radial. This is done using a right spherical triangle on a unit sphere whose center is the aircraft. Two adjacent (and perpendicular) sides are the arcs corresponding to DBER and the elevation angle (above the aircraft's local horizon) to the spacecraft. The third arc length corresponds to the angle between the velocity vector and the line-of-sight from the aircraft to the spacecraft. The aircraft velocity component along the line-of-sight is its speed times the cosine of the angle just computed.

3.3.13 SUBROUTINE PWRBAL

PURPOSE:

This routine is responsible for maintaining the state of charge of the battery. The state of charge is updated each time increment based upon the amount of time the payload is used during the period and the solar eclipse state.

INPUT ARGUMENTS:

The following parameters are input via the calling sequence:

TUSED Amount of time payload was used in hours.

ECLIPS Logical variable indicating eclipse state, TRUE for eclipse; FALSE, otherwise.

SOC State of charge, as a fraction, at start of time interval.

The following parameters are input via Common X2 or X8

BATCAP Maximum battery capacity in kilowatt hours.

PSOLAR Power input from solar panels in kilowatts

PPLON Power consumed while payload is on, including power consumption of platform, in kilowatts.

PPLOFF Power consumed while payload is off, including power consumption of platform, in kilowatts.

DUTCYC Duty cycle of payload.

DTIME Time interval in hours.

OUTPUT ARGUMENTS:

SOC The state of charge at the end of the time interval

METHOD:

The input power is determined by the eclipse state, 0 for TRUE, PSOLAR for FALSE. If the payload was used during the period, the consumption rate is set to PPLON for a period of DTYCYC * TUSED. The period the payload was off is either the entire interval or the remainder of the interval (DTIME-DTYCYC * TUSED) and power consumption, PPLOFF. The net power is then computed (Power in minus Power out) as well as an ETA factor. If the battery is discharging, the change in state of charge is the net change times the period divided by ETA times battery capacity. If the battery is charging, the change in state of charge is equal to net change times ETA times .7576 times the period divided by the battery capacity. The new SOC is then the old SOC plus the change in state of charge. If the payload was used, there is a period that it was not used during the interval. The power consumption during that period is computed next as already described for the payload off.

REFERENCES:

This routine is called by COV.

3.4 OUTPUT

3.4.1 SUBROUTINE BGNCRS

PURPOSE:

BGNCRS calls SETL EPS1 before moving target begins course so that coverage statistics computed by subroutine summary and the post-processor STATCV include gaps and detections occurring at the beginning of the target's course. (Note that, in the statistics, the initial gap or detection will be EPS1 plus the duration of the gap or detection.)

ENDCRS calls SETL EPS1 after moving target begins course so that coverage statistics computed by subroutine summary and the post-processor STATCV include gaps and detections occurring at the end of the target's course. (Note that, in the statistics,the final gap or detection will be EPS1 plus the duration of the gap or detection.)

INPUTS:

STPROF	hours	Time (since start of simulation) that moving target begins course.
EDPROF	hours	Duration of target's course.
EPS1	hours	See above. Normally set to .0001 hours.
NSMCON	-	Number of target-ground station sets on ISMTAB array.
NTRGTT	-	Number of targets on the TARGTS/ITRGTS array plus NSMCON.
DTIME	hours	Time step size for simulation.
TIME	hours	Current time being simulated.

OUTPUTS:

TADJ	hours	Added to TIME; is the time at which SETL is called and results in SETL being called EPS1 before course begins or EPS1 after course ends. Is reset to 999 at end of subroutine.

METHOD:

The main program, COV, calls BGNCRS/ENDCRS two time steps before/after course begins/ends. For BGNCRS, a modular divide determines how much time elapses in the next time step before the course begins; EPS1 is subtracted from the time period to compute TADJ. For ENDCRS, a modular divide determines how much time elapsed in the previous time step before the course ended; EPS1 is added to this period to compute TADJ. After SETL is called, TADJ is reset to 999.

REFERENCES:

Calling routines: COV
Called routines: SETL

3.4.2 Subroutine CTOALF

PURPOSE:

This routine is responsible for converting the numeric cluster number found in ISCAN array to either an alpha or a digit (in alpha form). The SCSCH(4, I) volumn is also converted to alpha. The ISA vector is set from the rounded value of the CLBG(5,5,I) vector. These values are used in the SCAN printout.

INPUT ARGUMENTS:

The following inputs are passed via the calling sequence:

NSC1 Number of appropriate S/C;

NTRGTS Number of targets;

SCSCH Array whose (4, I) column represents the selected cluster id;

The ISCAN array and the CLBG array are passed via the X4 Common.

OUTPUT ARGUMENTS:

The following outputs are passed via the calling sequence:

ISCSCH Array whose (4, I) column represents the converted ID for the selected cluster;

ISA Vector containing the rounded value of the CLBG (5, 5, I) column.

The ISCAN array is also used for output.

METHOD:

ISA is derived from the following:

ISA(I) = IFIX(.5 + CLBG(5, 5, I))

ISCAN is derived from the following:

'.' if ISCAN (J,I) = 0

-ISCAN(J,I) as an alpha coded digit (1-9)

ISCAN(J,I) if ISCAN(J,I) < 0

'A'-'Z' if 0 < ISCAN(J,I) ≤ 26

'*' if ISCAN(J,I) > 26

ISCSCH(4,I) is transformed in the same way ISCAN is.

REFERENCES:

This routine is called from COV.

3.4.3 SUBROUTINE SETL

PURPOSE:

Gathers statistics on the duration of detections and of gaps between detections. At the start of each new detection, SETL also writes on unit 8 the time at which the detection begins and the duration of the coverage gap since the last detection.

INPUTS:

TIME — Current clock time (hours);

DTIME — Simulator time step (hours);

DTIMX — Larger time step if two types of spacecraft are modeled;

SATTAK — Time at which current replication started (hours);

TADJ — Amount by which time has been adjusted by ENCRS or BGNCRS;

NTAR — I (row on TARGTS array) for object being detected;

IPLOT2 — Must be equal to or greater than one to cause a write on unit 8;

DELTIM — Time since aircraft began flight or since start of replication (for fixed targets);

NUMCT — Number of simultaneous contacts required for detection;

OUTPUTS: (all times in hours)

XLN(I, 1) — Time at which last contact occurred;

XLN(I, 2) — Maximum time out of contact;

XLN(I, 3) — End time of maximum time out of contact;

XLN(I, 4) — Maximum time out of contact;

XLN(I, 5) — End of minimum time out of contact;

XLN(I, 6) — Total time out of contact;

XLN(I, 7) — Number of contacts;

XLN(I, 8) — Sum of squares of down times;

XLN(I, 9) — Time at which first detection began;

XLN(I, 10) Time at which last detection began;

XLN(I, 11) Time at which current contact began (zero if currently out of contact);

Data written on unit 8:

NTAR ID of aircraft or barrier being detected;

IATTAK Replication number;

DLTIM DELTIM (see above);

DT Time since last contact ended;

ICLS (NTAR) 1 if aircraft is classified; 0 if otherwise;

METHOD:

SETL is called everytime a ground station, an aircraft, or a barrier is detected. If the detection is the first detection after a gap, then the SLM array is updated and unit 8 written on; otherwise a RETURN is executed. Special tests are required if a detection occurs exactly at the start of a replication or at the first time step into a replication. BGNCRS and ENDCRS call SETL just before and after aircraft flights so gaps before the first detection and after the last detection are counted in the statistics. However, when ENDCRS calls SETL, DT is always less than the time step, which would cause an immediate return. However, this return is prevented if TADJ is negative, and TADJ is negative only if SETL is called from ENDCRS.

Special format statements are used to write a zero before single digit positive numbers so sort utilities will sort on a zero instead of a blank.

3.4.4 Subroutine STATPT

PURPOSE:

This routine calls STHIST for each of five terms to printout the post-run statistics. The terms are abbreviated SA, CW, SC, EB, and RC.

INPUT ARGUMENTS:

The STATS Common provides all the inputs to this routine. For each of the five terms, the following data are provided where ** is an abbreviation for one of the five terms:

NCNT**	Total of all frequency counts
**MIN	Minimum value
**MAX	Maximum value
**SUM	Sum of values
**SUM2	Sum of values squared
I**VEC	Frequency count vector

OUTPUT ARGUMENTS:

For each term, the following data are returned:

**SUM	Mean
**SUM2	Stand deviation

METHOD:

The routine contains data statements for the labels for each of the five terms. The routine calls STHIST for each term after heading each term's report.

REFERENCES:

STATPT is called by COV. It calls STHIST.

3.4.5 Subroutine STATUD

PURPOSE:

This routine is responsible for maintaining the statistics for each of the five terms. The count, minimum, maximum, sum and sum square are maintained and the frequency bins are incremented as appropriate. The terms are abbreviated SA, CW, SC, EB and RC.

INPUT ARGUMENTS:

The calling sequence provides the following arguments:

MODE	The term identifier (1 for SA, 2 for CW, 3 for SC, 4 for EB and 5 for RC)
VALUE	The value of the term.

The STATS Common is used to hold these statistics, as defined in STATUP.

OUTPUT ARGUMENTS:

The updated STATS Common is the only output for this routine.

METHOD:

Depending upon the value of MODE, the statistics are updated for the term. For the basic statistics, the formulae are as follows where ** represents the term's abbreviation:

```
NCNT** = NCNT** + 1
**MIN  = AMIN1(**MIN, VALUE)
**MAX  = AMAX1(**MAX, VALUE)
**SUM  = **SUM + VALUE
**SUM2 = **SUM2 + VALUE * VALUE
```

For this frequency array, the maintenance is dependent upon MODE.

```
For MODE = 1,
    I = MINO(18, MAXO(1, 1 + IFIX(-1.E-10 + VALUE/10.)))
For MODE = 2,
    I = MINO(11, MAXO(1, 1 + IFIX(-1.E-10 + VALUE/10.)))
For MODE = 3,
     I = MINO(12, MAXO(2, 2 + IFIX(-1.E-10 + VALUE/10.)))
     or I = 1 if VALUE ≤ 0
For MODE = 4,
     I = MINO(11, MAXO(1, 1 + IFIX(-1.E-10 + VALUE/10.)))
For MODE = 5,
    I = MINO(9, MAXO(1, 1 + IFIX(-1.E-10 + VALUE/25.)))
    and I***VEC (I) = I***VEC(I) + 1
```

REFERENCES:

The routine is called by COV.

3.4.6 SUBROUTINE SMMRY

PURPOSE:

Computes statistics summarizing one replication using data stored in the XLN array.

INPUTS

XLN(I, 1)	Time at which last contact occurred;
XLN(I, 2)	Maximum time out of contact;
XLN(I, 3)	End time of maximum time out of contact;
XLN(I, 4)	Maximum time out of contact;
XLN(I, 5)	End of minimum time out of contact;
XLN(I, 6)	Total time out of contact;
XLN(I, 7)	Number of contacts;
XLN(I, 8)	Sum of squares of down times;
XLN(I, 9)	Time at which first detection began;
XLN(I, 10)	Time at which last detection began;
XLN(I, 11)	Time at which current contact began (zero if currently out of contact);
NTRGTS	Number of aircraft and barrier input in TARGTS array;

OUTPUTS: (all times in hours)

All outputs are written on unit 6.

The following is computed for each target:

Maximum value of down time;
Time at which maximum down time ended;
Minimum value of down time;
Time at which the minimum down time ended;
Total down time;
Number of contacts;
Mean time between contacts;
Standard deviation of down times;

Percent of time covered;
Average revisits per day;
Start time of first contact;
Start time of last contact;

In addition the minimum, average, and maximum values across all targets are compiled for:

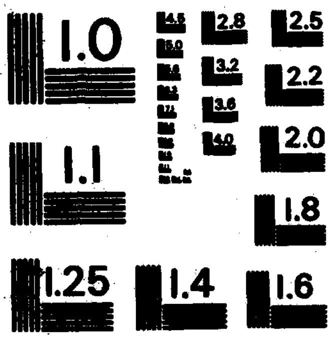
Maximum time between contacts;
Minimum time between contacts;
Average time between contacts;
Standard deviaton of down time;
Average duration of contacts;
Percent coverage;
Average revisits per day;
Average revisit interval;

METHOD:

Method is implied by the outputs.

AD-A120 877 DESIGN NOTEBOOK FOR NAVAL AIR DEFENSE SIMULATION 2/2
SURVEILLANCE SUPPLEMENT..(U) TRW DEFENSE SYSTEMS GROUP
MCLEAN VA WATERWHEEL PROGRAM OFFICE..
UNCLASSIFIED A G PURVES ET AL. 15 OCT 82 F/G 22/2 NL

END
FILMED
DTIC


1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

3.5 UTILITIES

3.5.1 SUBROUTINE ANGLED

PURPOSE:

Computes the angular separation between two points in degrees.

INPUTS:

A(3), B(3) -- Unit vectors.

OUTPUT:

ANG -- Angular separation between A and B in degrees.

METHOD:

Standard trigonometry.

REFERENCES:

No external references

3.5.2 SUBROUTINE BRNG

PURPOSE: Given two pts., A and B, find the gearing from pt. A to pt. B in radians and in degrees.

INPUT ARGUMENTS:

ALAT	Lattitude of pt. A (degrees)
ALON	Longitude of pt. A (degrees)
BLAT	Lattitude of pt. B (degrees)
BLON	Longitude of pt. B (degrees)

OUTPUT ARGUMENTS:

RBRNG	Bearing from pt. A to pt. B in radians
DBRNG	Bearing from pt. A to pt. B in degrees

METHOD:

Given two pts. A and B,

1) IF ALAT = 90.0 or BLAT = -90° THEN the bearing is equal to PI.

2) IF ALAT = -90 of BLAT = 90° THEN the bearing is equal to 0.

3) Otherwise

   a) Find the distance from pt. A to pt. B.

   b) IF the difference between ALON and BLON is not 180°, then the bearing from pt. A to pt. B is defined as follows:

$$\cos(\text{Bearing}) = \frac{\sin(\text{BLAT}) - \sin(\text{ALAT})*\cos(\text{DIST})}{\cos(\text{ALAT})*\sin(\text{DIST})}$$

where

DIST	Distance from A to B in radians
Bearing	Bearing from pt. A to pt. B

c) IF the difference between ALON and BLON equals 180° then the bearing is defined as follows:

```
IF ALAT and BLAT are approximately equal then
        if ALAT BLAT then bearing = ¶
        if ALAT BLAT then bearing = 0
IF. ALAT and BLAT do not approximate each other then
        if ALAT 0. then bearing = 0
        if ALAT<0 then bearing = ¶
```

REFERENCES:

Subroutine BRNG is used by COVERAGE and PROFILE.

3.5.3 SUBROUTINE CIRSCT

PURPOSE:

Given two points on the earth, each with a different radius, compute the points of intersection.

INPUTS:

LATA, LONA -- Latitude and longitude on point A.
RADA -- Radius of circle at point A in NMI.
LATB, LONB -- Latitude and longitude of point B.
RADB -- Radius of circle at point B in NMI.

OUTPUTS:

LAT1, LON1 -- Intersect point 1.
LAT2, LON2 -- Intersect point 2.
IPOINT -- Number of intersections (0 or 2).

METHOD:

This routine will never allow only one point of intersection. If the two points are located less than 0.6 degrees of each other, then it is assumed there is no intersection.

REFERENCES:

The following subroutines are called:

1. VEC
2. CONES
3. ANGLED
4. RADECM

THIS PAGE INTENTIONALLY LEFT BLANK

3.5.4 SUBROUTINE COMBLK

PURPOSE:

To set up constants and units for COVERAGE.

3.5.5 SUBROUTINE CONES

PURPOSE:

Computes the two unit vectors defined by the intersection of two given cones.

INPUTS:

A(3) -- Unit vector, axis of cone 1.
B(3) -- Unit vector, axis of cone 2.
COS1 -- Cosine of the half angle for cone 1.
COS2 -- Cosine of the half angle for cone 2.

OUTPUTS:

V1(3), V2(3) -- The two unit vectors defined by the intersection of the two cones. If the two cones do not intersect, then V1(1) is set to 2.0.

METHOD:

Standard trigonometry

REFERENCES:

No external references.

3.5.6 SUBROUTINE DIS

PURPOSE: Given two pts., A and B, find the distance from pt. A to pt. B in radians, degrees, and nautical miles.

INPUT ARGUMENTS:

ALAT	Lattitude of pt. A
ALON	Longitude of pt. A
BLAT	Lattitude of pt. B
IOPT	= 1 implies coordinates of A and B are input in degrees
	= 2 implies coordinates of A and B are input in radians

OUTPUTS ARGUMENTS:

RDIST	Distance from pt. A to pt. B in radians
DDIST	Distance from pt. A to pt. B in degrees
DISTNM	Distance from pt. A to pt. B in nautical miles

METHOD:

If pt. A and pt. B are two points on a unit sphere, then the arc-length distance between them is defined as follows

$$\cos(\mathrm{DIST}) = \sin(\mathrm{ALAT})\sin(\mathrm{BLAT}) + \cos(\mathrm{ALAT})\cos(\mathrm{BLAT})\cos(\mathrm{ALON}-\mathrm{BLON})$$

where $0° \leq \mathrm{DIST} \leq 180°$

output DIST in radians, degrees, and nautical miles.

REFERENCES:

Subroutine DIST is used by COVERAGE and PROFILE.

3.5.7 SUBROUTINE FLALOB

PURPOSE: Given the coordinates of a pt. A, the bearing from pt. a to another pt., B, find the coordinates of pt. B.

INPUT ARGUMENTS:

ALAT	Lattitude of pt. A
ALON	Longitude of pt. A
DIST	Distance from pt. A to pt. B
BRNG	Bearing from pt. A to pt. B
IOPT	= 1 implies that all inputs and outputs are in degrees = 2 implies that all inputs and outputs are in radians

OUTPUT ARGUMENTS:

BLAT	Lattitude of pt. B
BLON	Longitude of pt. B

METHOD:

The coordinates of pt. B can be defined as follows:

$$\sin(BLAT) = \cos(BRNG)\cos(ALAT)\sin(DIST) + \sin(ALAT)\cos(DIST)$$

$$BLON = SIGN(DLONG,\ BRNG)$$

where DLONG is the difference between ALON and BLON

$$\cos(DLONG) = \frac{\cos(DIST) - \sin(BLAT)\sin(ALAT)}{\cos(BLAT)\cos(ALAT)}$$

BLON is then adjusted such that

$$0 \leq |BLON| \leq 180°$$

REFERENCES:

FLALOB is used by COVERAGE.

3.5.8 Subroutine HISTGM

PURPOSE:

HISTGM is a utility routine responsible for generating a horizontally arrayed histogram bar chart on a character printer (as opposed to a graphics display device). Both self-scale and user-scale options are available. Both percent and absolute levels are printed.

INPUT ARGUMENTS:

NCOUNT	Total of all frequency counts;
IFREQ	Array of frequency counts (rows);
NROW	Number of bins or rows;
MAX	Scaling term: > 0 for the maximum scale (frequency for any row); ≤ 0 for self-scale;
LABEL	Array of eight character per row alphanumeric row labels;
IUNIT	The FORTRAN logical unit number of the file on which the histogram is to be written.

OUTPUT ARGUMENTS:

None

METHOD:

If MAX is less than or equal to zero, the routine finds the maximum frequency for all the rows.

All frequencies will be scaled by 50./maximum.
The histogram is a bar chart whose length is computed by:

```
J = IFIX(FLOAT(IFREQ(I))*(50./Maximum) + .5)
J = MINO(50, MAXO(0,J))
```

J asterisks are printed out for the bar, preceeded with the eight character label and followed by the value of IFREQ and the percent of NCOUNT represented by IFREQ for each row.

REFERENCES:

HISTGM is called by STHIST routine.

THIS PAGE INTENTIONALLY LEFT BLANK

3.5.9 SUBROUTINE JDORGF

PURPOSE:

To convert a date from julian to year, month, day, hours, minutes, seconds or from year, month, day, hours, minutes, seconds to julian date.

3.5.10 SUBROUTINE LINTRP

PURPOSE:

LINTRP is a utility routine responsible for performing a bounded linear interpolation of the value of a dependent variable given the independent variable and a table of dependent and independent variables. The independent variable must be monitorically increasing.

INPUT ARGUMENTS:

X1	Independent variable
X	Table of independent variables
Y	Table of dependent variables
N	Length of tables

OUTPUT ARGUMENTS:

Y1	Dependent variable associated with X1.

METHOD:

$$Y1 = \begin{cases} Yo \text{ if } X1 \leq Xo \\ Yi +(Yi - Yi\text{-}1 (X1 - Xi\text{-}1)/(Xi - Xi\text{-}1) \text{ for } Xi\text{-}1 \leq X1 < Xi \\ Yn \text{ if } X1 \geq Xn \end{cases}$$

3.5.11 SUBROUTINE MAGN

PURPOSE:

To compute the magnitude of vector A.

INPUTS:	A	A three element array.

OUTPUTS:

B	Magnitude of A;
C	Magnitude of A squared;

METHOD:

Magnitude of vector A is the square root of the sum of the squares of the three elements of A.

REFERENCES:

MAGN is used by COVERAGE.

3.5.12 SUBROUTINE RADECM

PURPOSE:

Computes the right ascension and declination of a given vector, and its magnitude.

INPUTS:

ALPHA -- Right Ascension in degrees.
DELTA -- Declination in degrees.
VMAG -- Magnitude of V.

METHOD:

Standard trigonometry

REFERENCES:

No external references.

3.5.13 SUBROUTINE RMMUL

PURPOSE:

To find the product of two matrices.

INPUTS:

A	Matrix of size L x M;
B	Matrix of size M x N;

OUTPUTS:

C	Product of matrices A and B; (Matrix of size L x N)

REFERENCES:

RMMUL is used by COVERAGE.

3.5.14 SUBROUTINE SPHTRN

PURPOSE: Given the coordinates of a point P in the old system and the coordinates of a new pole in the old coordinate system, find the coordinates of point P in the new system.

INPUT ARGUMENTS:

PHI	Longitude of pt. in old system
TH	Colatitude of pt. in old system
PHIO	Longitude of the new pole
THO	Colatitude of the new pole
PPHIO	Bearing of the azimuthal reference

OUTPUT ARGUMENTS:

PPHI	Longitude of pt. in the new system
PTH	Colatitude of pt. in the new system

METHOD:

The new azimuthal reference is Ø° 0. Therefore the colatitude of the pt. P in the new system can be defined as follows:

cos(PTH) = cos(THO)cos(THO)cos(TH)+sin(THO)sin(TH)cos(PHIO)

where PTH is the colatitude of pt. P in the new system.

sin(PPHI) = -sin(PHIO)sin(TH)/sin(PTH)

REFERENCES:

SPHTRN is used by COVERAGE.

3.5.15 Subroutine STHIST

PURPOSE:

This utility routine is responsible for computing the mean and standard deviation of a set of data and outputting the minimum and maximum, total count, mean and standard deviation and frequency histogram.

INPUT ARGUMENTS:

NCOUNT Total of all frequency counts, N;
IFREQ Array of frequency counts;
NROW Number of rows or bins into which frequency has been divided;
MAX Explicit scale for histogram if greater than 0;
LABEL Eight character row labels for histogram;
IUNIT FORTRAN logical unit number for output;
XMIN Minimum value of X;
XMAX Maximum value of X;
SUM SUM of X;
SUM2 SUM of X squared;

OUTPUT ARGUMENTS:

SUM Mean value of X
SUM2 Standard deviation of X

METHOD:

The mean and standard deviation of X are computed as follows for NCOUNT great than 1:

```
SUM =    SUM/FLOAT(NCOUNT)
SUM2 =   SQRT((SUM2 - FLOAT(NCOUNT)*SUM*SUM)/FLOAT(NCOUNT-1))
```

The routine writes the NCOUNT, minimum, maximum, mean and standard deviation on device IUNIT. It then uses routine HISTGM to generate a frequency histogram.

REFERENCES:

STHIST is called by STATPT routine and it calls HISTGM.

3.5.16 SUBROUTINE SUN

PURPOSE:

To compute the orbital elements of the sun for a given julian date.

INPUTS:

TOTJD	JULIAN DATE

OUTPUTS:

OES(6)	Orbital elements of the sun;

REFERENCES:

SUN is used by coverage.

3.5.17 SUBROUTINE TRI

PURPOSE:

Given either two sides and an angle of a triangle or two angles and a side of a triangle, an option flag, and an ambiguity flag, find the remaining sides and angles of the triangle.

INPUT - OUTPUT MATRIX:

DEFINE P, Q, R - angles (degrees)
X, Y, Z - sides (degrees) such that
x is opposite P,
y is opposite Q, and
z is opposite R.
L - option Flag
M - ambiguity Flag

To OUTPUT: ANGLES, SIDES (RAD)	INPUT: L	 M	 ANGLES, SIDES (RAD)
R(QUAD 1 or 2), X, Y	1	1	P, Q, Z
R(QUAD 3 or 4), X, Y	1	2	P, Q, Z
Y(QUAD 2 or 3), R, Z	2	2	P, Q, X
X(QUAD 1 or 2), Q, R	3	1	P, Y, Z
X(QUAD 3 or 4), Q, R	3	2	P, Y, Z
Q(QUAD 1 or 4), R, Z	4	1	P, X, Y
Q(QUAD 2 or 3), R, Z	4	2	P, X, Y
P(QUAD 1 or 2), Q, R	5	1	X, Y, Z
P(QUAD 3 or 4), Q, R	5	2	X, Y, Z
X(QUAD 1 or 2), Y, Z	6	1	P, Q, R
X(QUAD 3 or 4), Y, Z	6	2	P, Q, R

METHOD:

Simple trigonometry.

3.5.18 SUBROUTINE VEC

PURPOSE:

Computes the unit vector corresponding to a given right ascension and declination in degrees.

INPUTS:

ALF -- Right ascension in degrees.
DEL -- Declination in degrees.

OUTPUTS:

V(3) -- Unit vector corresponding to ALF and DEL.

METHOD:

Standard trigonometry

REFERENCES:

No external references.

3.5.19 SUBROUTINE ZERRV

PURPOSE:

To initialize a real array.

INPUTS:

A	A real array of dimension N;
N	Dimension of A;

OUTPUTS:

$A(I) = 0.0$ where $1 \le I \le N$;

REFERENCES:

This routine is used by COVERAGE and AAWSIM.

3.5.20 SUBROUTINE ZERIV

PURPOSE:

To Initialize an integer array.

INPUTS:

IA	An integer array of dimension N;
N	Dimension of IA;

OUTPUTS:

$IA(I) = 0$ where $1 \leq I \leq N$;

REFERENCES:

This routine is used by COVERAGE and AAWSIM.

3.6 FUNCTIONS

The following functions are utilized in the program:

ACOSD Given the cosine of an angle, find the angle in degrees.

ANGMOD Given an angle in degrees, output the angle such that $0^\circ \leq \text{ANGLE} < 360^\circ$.

ASIND Given the sine of an angle, find the angle in degrees.

ATAND Given the tangent of an angle, find the angle in degrees.

ATAN2D Given the sine and cosine of an angle, find the angle in degrees.

COSD Given an angle in degrees, find the cosine of the angle.

DGLIM Given an angle in degrees, output the angle in degrees such that the absolute value is less than or equal to 180.

PILIM Given an angle in radians, output the angle in radians such that the absolute value is less than or equal to PI.

PIMOD Given an angle in radians, find the angle such that $0 \leq \text{ANGLE} < 2\pi$

SIND Given an angle in degrees, find the sine of the angle.

TAND Given an angle in degrees, find the tangent of the angle.