NASA Contractor Report 182025
DOT/FAA/DS-90/7

USER GUIDE FOR AN AIRBORNE WINDSHEAR DOPPLER RADAR SIMULATION (AWDRS) PROGRAM

C. L. Britt
Research Triangle Institute
Research Triangle Park, NC

Contract NAS1-17639
June 1990

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

TABLE OF CONTENTS

		Page
LIST OF FIGURES		iv
1.0	INTRODUCTION	1
2.0	GENERAL DESCRIPTION	2
3.0	PROGRAM STRUCTURE	7
	3.1 Overview	7
	3.2 Main Program	9
	3.3 Subroutines	9
4.0	INPUT FILES	14
	4.1 Overview	14
	4.2 Data File WXIN.DAT	14
	4.3 Filename File WXFILES.DAT	21
	4.4 Microburst Data Input Files	22
	4.5 Clutter Map Data Input Files and Map Specification Files	23
	4.6 Antenna Pattern Data (WXANT.DAT)	23
5.0	OUTPUT FILES	24
	5.1 Output File WXOUT.DAT	24
	5.2 User Named Output File #1 (OF1)	24
	5.3 User Named Output File #2 (OF2)	24
	5.4 User Named Output File #3 (OF3)	25
	5.5 I and Q Output Data (IQOUT.DAT)	26
6.0	EXAMPLE DATA RUN	27
	6.1 General Discussion	27
	6.2 Input Data	27
	6.3 Outputs	27
7.0	APPENDIX	
	A. Main Program Listing	43
	B. Subroutines	57
8.0	REFERENCES	87

LIST OF FIGURES

Page

1. Flow Diagram Showing the General Operation of the Windshear Radar Simulation Program 29

2. Logical Flow Diagram Showing the Main Calculation Loops in the Program........... 30

3. Sketch Showing the Breakdown of a Range Bin into Incremental Scattering Cells....... 31

4. Techniques for Calculation of In-Phase (I) and Quadrature (Q) Signals.............. 32

5. Input File WXIN.DAT. This File Defines the Parameters of the Simulation............ 33

6. Filename File WXFILES.DAT, Clutter Map Specification File WXRD1.DAT and Output Data File WXTEST.DAT........................ 34

7. Plot of the Doppler Spectra for a Range Bin 4km from the Aircraft in the Example Data Run........................... 35

8. Rain, Clutter and Discrete Powers Plotted vs. Radar Range. The Aircraft Touchdown Point is at 7km. No Discrete Targets were used in this run, hence the Discrete Plot is meaningless........................ 36

9. Rain-to-Clutter and Rain-to-Noise Power Ratios. The dBz Value (REF) of the Microburst Data along the Antenna Boresight is also Plotted............................ 37

10. Radar Derived Wind Velocity vs. Radar Range for Pulse-Pair (PP) and Spectral Average (SA) Processing. The "True" Velocity, or Velocity Along the Antenna Boresight Line, is also Plotted........................... 38

11. Hazard Index vs. Radar Range as derived from Pulse-Pair (PP), Spectral Averages (SA) and True Wind Velocity (TRUE). The Total Hazard Index including the Vertical Component (not measured by the radar) is also Plotted.................................. 39

12. Relative AGC Gain Vs. Radar Range. The 0 Level is the Average Value of Gain. Each Range Bin is assumed to have an Independent AGC Level.................................. 40

13. Plot of the Ratio of Ground Clutter Power Suppressed by the Clutter Filter Algorithm.......................... 41

1.0 INTRODUCTION

This report provides a description of the Airborne Windshear Doppler Radar Simulation (AWDRS) program developed for NASA-Langley Research Center by the Research Triangle Institute. The radar simulation program is a comprehensive calculation of the signal characteristics and expected outputs of an airborne coherent pulsed Doppler radar system viewing a low-level microburst along or near the approach path of the aircraft.

The simulation contains algorithms for direct calculation of radar signal returns from data files that provide, at any point in space, the radar reflectivity of moisture and the scattering cross section of ground clutter. The instantaneous radar signal amplitude is calculated by spatially integrating the radar equation over a large population of incremental scatterers. Further calculations simulate the signal processing done by a coherent radar in filtering the signal, providing Automatic Gain Control (AGC), forming in-phase (I) and quadrature (Q) base-band signal components, converting the I and Q signals to digital values, filtering the I and Q signals, and deriving Doppler velocity, spectral width, shear hazard and other radar outputs of interest.

The detailed nature of the simulation permits the quick evaluation of proposed trade-offs in radar system parameters and the evaluation of the performance of proposed configurations in various microburst/clutter environments. The simulation also provides a test bed for various proposed signal processing techniques for minimizing the effects of noise, phase jitter, and ground clutter and maximizing the useful information derived for avoidance of microburst windshear by aircraft.

2.0 GENERAL DESCRIPTION

Figure 1 is a block diagram indicating the major features of the simulation. Inputs to the program include an input file specifying the radar system parameters and when data files that contain the characteristics of the ground clutter and the microburst. The ground clutter data file consists of high-resolution (20m) calibrated Synthetic Aperture Radar (SAR) data of selected airport areas. The microburst data files comprise a database of reflectivity factors, x,y,z wind velocity components, and other meteorological parameters with a resolution of 30 meters. This data base is generated by a numerical convective cloud model driven by experimentally determined initial conditions, and represents selected time periods of the microburst development.

For each range bin, the simulation calculates the received signal amplitude level by integrating the product of the antenna gain pattern and scattering source amplitude and phase over a spherical-shell volume segment defined by the pulse width, radar range and ground plane intersection. The amplitude of the return from each incremental scatterer in the volume segment is proportional to either the square root of the normalized cross section of the ground clutter (from the clutter map) or the square root of the reflectivity factor of the moisture in the microburst (from the microburst data base). The phase of each incremental scatterer is the sum of a uniformly distributed ($0 - 2\pi$) random phase term, a phase term due to relative aircraft-scatterer radial velocity, and normally distributed random phase terms representing transmitter/receiver phase jitter and ground clutter random motion. The random phase terms simulating phase jitter and ground clutter motion are updated for each transmitted pulse while the uniformly distributed phase terms are updated for each sequence of pulses in a range bin. The phase terms representing aircraft-scatterer relative motion are linear functions of time.

Path attenuation for each incremental scatterer is determined by integrating the path losses over the transmission path. Empirical formulas from ref. [1] are used to determine the incremental path losses from the moisture content of the microburst. Aircraft ground velocity is assumed to be known accurately so that derived Doppler frequencies can be referenced to a value of zero corresponding to the aircraft ground velocity.

Simulated antenna patterns include a generic parabolic antenna with size and aperture illumination taper specified by input data, and a flat-plate array antenna with a pattern similar to that found in the current generation of X-band airborne weather radars.

A sequence of N pulses of in-phase (I) and quadrature (Q) signal amplitudes are calculated for each range bin as discussed above and subjected to AGC amplification and A/D quantization. A simulated fast-acting AGC is used to adjust the gain of the system on a bin-by-bin basis to achieve a wide dynamic signal amplitude range and to prevent signal saturation (due to clutter) prior to and during A/D conversion. The I and Q pulse stream is then digitally filtered to suppress ground clutter near zero Doppler frequencies and processed using both conventional pulse-pair and spectral averaging algorithms to derive the average velocity and spectral width of the scatterers in the range bin. Further processing of the velocity data provides information on wind shear and aircraft hazard factor.

Provision is made in the simulation to generate returns from a specified number of range bins over a specified azimuth scan so that simulated color displays of reflectivity, velocity, shear, spectral width, etc., can be examined. Other outputs of the simulation include plots of power levels, velocity, spectral width, shear hazard factor, and AGC levels vs. radar range. Doppler spectra of ground clutter and moisture as derived from the I and Q signals from each simulated range bin are also plotted.

Microburst Model

As mentioned above, the microburst model is a detailed numerical convective cloud and storm model that calculates the time history of the development of a microburst. The model uses a nonhydrostatic, compressible and unsteady set of governing equations which are solved on a three-dimensional staggered grid. The computation can be initiated from observed data and generates realistic wind fields that compare favorably with observed data such as that obtained in the JAWS study [2]. For the radar simulations to date, a 4x4 km. lattice of 30x30 meter grid spacing increments (2D axisymmetric version) has been generated at selected time periods. The model output includes the radar reflectivity factor (dBz), wind velocity components, temperature, equivalent potential temperature, pressure and moisture content (water vapor, ice, cloud droplets, rain, snow and hail/graupel). The model was developed by MESO, Inc. under NASA sponsorship and is described in detail in references [2] and [3].

Clutter Model

The ground clutter model used for the simulation cases is a high-resolution X-band SAR map of selected airport areas provided by Environmental Research Institute of Michigan (ERIM). The SAR map files are calibrated by ERIM personnel (using ground corner reflectors) to provide normalized cross section data with a resolution of 20m. In the simulations, the aircraft is positioned at a selected distance from the runway touchdown point on a specified (usually three-degree) glide slope.

A problem with the use of existing SAR data is associated with the variation of cross section with depression angle. The data were taken at depression angles ranging from approximately 18 to 40 degrees whereas for the simulated airborne radar the depression angles of interest are approximately 1 to 6 degrees.

To partially overcome this problem, a depression angle correction algorithm was developed by ERIM personnel that corrects the normalized cross section as a function of the angle at which the data were taken and the angle at which the cross section is desired. While it is felt that the correction algorithm is valid for flat surfaces such as asphalt or concrete, the literature [4] suggests that clutter sources such as trees, tall grass and urban areas have cross sections that do not vary significantly with depression angles in the ranges of interest. For this reason, certain areas (urban) of the clutter map are excluded from the depression angle correction and uncorrected cross section values are used in the simulation. Also, areas of the map with normalized cross sections greater than -6 dB are not corrected.

Moving Clutter Model

Discrete moving targets are generated by a supplemental program called WXDGEN. This program uses data on highway traffic densities and vehicular speeds to generate a large number of discrete targets on highways associated with a specific ground clutter map. Data for input to the WXDGEN program are obtained from high-resolution aerial photographs that coincide with the areas covered by the ERIM SAR maps.

The data base generated by WXDGEN can provide up to 80,000 discrete targets with coordinate locations, velocities, and RCS values assigned to each target. The data base also contains targets such as aircraft on departure or approach paths.

Since this data base is still under development, it is not used in version 2.0 of the simulation.

3.0 PROGRAM STRUCTURE

3.1 Overview

A logical flow diagram of the main computer program is shown in Figure 2. Initially, the input files are read, and the program variables are initialized. The input data provide the angular orientation and position of the radar system relative to the runway touchdown point and the appropriate coordinate rotation matrices are calculated for later use.

The first loop in the program is the antenna azimuth scan angle loop as shown in the Figure 2. Following the scan angle loop is the range bin loop which calculates the radar outputs in each range bin.

The range bin loop is further subdivided into three integration loops for the integration over range, azimuth angle, and elevation angle of a segment of a circular hemisphere. This segment is subdivided into small incremental cell volumes or areas representing the scatterers in the field of view. The breakdown of the range bin into incremental cells is sketched in Figure 3. The number of incremental cells in each range bin is specified in the input data file containing the simulation parameters.

For each incremental cell in the range bin, the in-phase, (I) and quadrature (Q) pulses are calculated using the equation given in Figure 4. In this equation, the amplitude of the return from scatterer "i" is calculated (using the radar equation) using either the scattering cross section obtained from the ground clutter map or the reflectivity factor from the microburst model, whichever is appropriate. Each scatterer in the range bin is assigned a random phase, a phase term representing the relative motion between the scatterer and the aircraft, and another normally distributed random phase term representing the system pulse-to-pulse phase error. This latter phase term is updated for each pulse, whereas the random phase associated with the scatterer is held constant over the series of pulses used to derive the velocity. The sum of the returns from all of the incremental scatterers in the range bin provides a single I and Q pulse amplitude. Finally, a random variable representing receiver noise is added to the signal in accordance with the (input) specified noise level of the system. This procedure continues until a series of pulses (as specified in the input data) is produced. This series of pulses is then processed to derive the average velocity of the scatterers in the radar field of view.

At this point, the series of I and Q pulses calculated by the simulation can be output to an I and Q data file if desired. This data may then be used for testing various signal processing algorithms for determining Doppler velocity, spectral width, or hazard index.

The program continues by taking a Fast-Fourier-Transform (FFT) of the I and Q pulse stream to determine the Doppler spectra for the range bin. This Doppler spectra is also written to an output file if desired. In addition, calculations are made on the I and Q pulse stream to determine power levels, average velocity using pulse-pair processing and spectral averaging, spectral width using pulse-pair processing and spectral averaging, and other parameters.

After the above calculations are complete, the next range bin is calculated and processing continues as shown in the flow chart. Upon completion of all range bin calculations, the windshear hazard factor [5] is calculated and added to the output file. In addition, data are output that permit development of simulated radar displays showing the radar outputs, (velocity, hazard factor, spectral width, or reflectivity) over a full azimuth scan of the radar.

3.2 Main Program

The main program (WXMAIN1.F) implements the functions described in the overview. Specialized functions are implemented in various subroutines. The major function of the main program is to implement the structure shown in Figure 2. A commented listing of the main program is given in Appendix A.

3.3 Subroutines

The following lists the subroutines used in the program and their function. The functions of the major subroutines used in the program are as follows:

1. RDDISC - Reads discrete target file and converts data to runway origin coordinates.

2. SETVAR - Places the output variables into an array called OV. The array OV is then written to an output file specified by an input data file.

3. FFAC - Calculates the hazard index by differencing adjacent velocity values calculated along a range line. The calculation uses velocities derived by pulse-pair processing, spectral averaging and true wind velocity.

4. AVRHAZ - Makes use of the OV array to average the hazard factor over several range bins (specified as input data range bins.)

5. DFILTER - Implements a high pass digital filter that operates on the I and Q signals and is used to suppress clutter signals.

6. EFILTER - This subroutine is used as an alternate to the above and implements either a first or second order Butterworth filter to operate on the I and Q signals for suppression of ground clutter.

7. FILCO - Calculates the coefficients necessary to implement the digital Butterworth filters used in subroutine EFILTER.

8. ADCONV - Calculates the AGC gain required to set the I and Q signals to a given average level and then quantizes the signals in accordance with the number of levels specified for the analog to digital converters. This subroutine assumes that AGC is implemented in each range bin separately.

9. ANTFLAT - Calculates the gain of a flat (Collins type) antenna with a beam width of 3.5 degrees. The algorithm implemented is obtained from data provided by Collins. Note that this algorithm is only valid for a frequency of 9.3 GHz.

10. HORIZ - Calculates the antenna pointing angle with respect to the horizon in degrees.

11. WXANAL - Calculates the average Doppler velocity in a range bin using a spectral averaging technique.

12. PPP - Calculates average Doppler velocity and spectral width using pulse pair processing algorithms.

13. ATTEN - Calculates the two-way path loss due to moisture using moisture values obtained from the microburst data base.

14. VELTRU - Calculates the "true" velocity component along the projected antenna beam center line. This true velocity component is then used in comparisons with radar calculated velocities.

15. GETBW - Calculates the 3dB beam width of the selected antenna in degrees.

16. READANT - Reads actual antenna pattern data in the antenna data file WXANT.dat.

17. ANTREAL - Calculates the antenna gain using input angles and data from the file read in with READANT.

18. READRWY - Reads runway specification file and map data associated with the specified ground clutter map.

19. RDCLUT5 - Reads the ground clutter map file specified in WXFILES.DAT.

20. CLUT5 - Gets a value of sigma zero from the ground clutter map at a location specified in the call statement. This routine also applies a correction factor for the angle of incidence.

21. STRNUM - Generates a uniformly distributed random number between 0 and 1.

22. BRSTWD3 - Calculates the three Cartesian wind components of the microburst given the spatial coordinates of a point. It also provides a reflectivity value at the spatial location specified.

23. ANTPAT - Calculates the antenna gain at a given input angle for a generic parabolic antenna with a diameter specified in input data.

24. NORMAL - This routine provides a random number in accordance with a normal distribution with a standard deviation of unity.

25. NORNG - Generates a sequence of numbers randomly distributed over the interval +/-3 from uniformly distributed random numbers.

26. CROSS - Takes the cross product of two vectors and outputs the product.

27. ROT - Creates rotation operator matrix.

28. EMAT - Creates a multiple rotation operator matrix with rotation through the Euler angles (roll, pitch and yaw).

29. GMTRA - Transposes an N x M matrix A to give a M x N matrix A^T.

30. GMPRD - Multiplies N x M matrix by an M x L matrix to provide an N x L matrix.

31. SPHER - Converts a vector defined in xyz rectangular coordinates to a vector defined in standard spherical coordinates.

32. MEG - Creates a matrix for conversion of a vector defined in a rectangular earth centered coordinates system to a vector in geographical coordinates (North, East, Down). The origin of the geographical coordinates is specified by its latitude and longitude (Not used in simulation).

33. BESSN - Finds the Bessel function of either integer or fractional order for any input value greater than zero.

34. GAMMA - This subroutine is used to find the Gamma function for any variable greater than zero. Used in subroutine BESSN.

36. FFT - Computes the fast Fourier transform of a complex vector of specified length.

37. PLYEVL - Library subroutine used by FFT subroutine.

38. FFRDR2 - This subroutine implements a function used in the FFT subroutine to permute a complex data vector in reverse binary order to normal order.

Detailed listings of these subroutines are provided in Appendix B.

4.0 INPUT FILES

4.1 Overview

There are several input data files necessary to run the program. These include:

1) Definition of parameters (WXIN.DAT)
2) Specification of user-defined filenames (WXFILES.DAT)
3) Microburst data base (3 files)
4) Ground clutter map data base (1 file)
5) Ground clutter map supplemental data (1 file)
6) Antenna pattern data (WXANT.DAT)

These files are discussed in sections 4.2 thru 4.6.

4.2 Data File WXIN.DAT

The file WXIN.DAT shown in Figure 5 provides the parameters that set up the simulation scenario. It consists of simulation parameters, radar parameters, parameters to define the microburst and clutter characteristics, and parameters to define signal processing algorithms to be used. The following adds explanatory notes to the items in the file. Most of the inputs are self-explanatory, and comments are added where required.

SIMULATION PARAMETERS	0.
A/C Distance to Touchdown (km)	7.
Aircraft Velocity (kts)	150.
Glideslope Angle (deg)	3.
No. of Complete Scans	1.
Time Between Scans (sec.)	5.
Roll Attitude (deg)	0.
Pitch Attitude (deg)	0.
Yaw Attitude (deg)	0.

These parameters define the location, velocity and angular attitude of the aircraft relative to the runway touchdown point as defined in the ground clutter map. The roll, pitch and yaw angles are rotations in a right-handed sense about a coordinate system u, v, w with u along the aircraft velocity vector and w down. Also, the number of azimuth scans desired and the time interval between the scans is specified as shown.

Az Integration Range/2 (deg)	6.0
Az Integration Increment (deg)	.3
El Integration Range/2 (deg)	4.0
El Integration Increment (deg)	.2
Rng Integration Increment (m)	100.

The above five parameters define the integration range of the program with respect to the antenna beam center. Wider ranges provide more accuracy but require longer running times. The angles should be large enough to take into account the major sidelobes of the antenna for best accuracy. The angular increments should be less than the antenna beamwidth (approximately 3 deg). The number of radar scatterers used in the calculation are defined by the volume specified by the pulse width and angular ranges and the associated increments defined above.

Random Number Seed (0-1)	.224

This is a value used to initialize the random number generators in the program.

Runway Number	26
Right (1) or Left (2)	2.
	0.

These parameters specify the runway in the input clutter map that will be used as the coordinate reference point for the map. In the above example, runway 26L at Denver is specified.

MICROBURST & CLUTTER	0.
	0.
Along Track Offset from TD (km)	-2.
Cross Track Offset from TD (km)	0.

These numbers define the location of the microburst relative to the aircraft touchdown point.

Rain Standard Deviation (m/s)	1.
Clutter Standard Deviation (m/s)	.5

The above parameters define a random velocity component that is applied to the calculated radial component of the velocity of a scatterer. This random component is different for each radar pulse. The rain standard deviation (S.D.) simulates fine-grained turbulence and the clutter S.D. simulates moving grass, trees, etc.

Clutter Calc. Flag (1=ON, 0=OFF)	1.

This "switch" turns off the calculation of ground clutter, if desired.

Reflectivity Calc. Thres. (dBz) .1

The radar return from the rain scatterer is not calculated if the reflectivity of the scatterer (from the microburst data base) is below this threshold. A value of .1 is used for a wet microburst and -14 for a dry microburst. A large value (100.) is used to eliminate the microburst scatterers from the simulation (i.e., simulates "no microburst present").

Discrete Calc. Flag (1=ON, 0=OFF) 0.

This flag suppresses the reading of the discrete target input file and the calculation of the radar return from discrete moving targets if it is set to 0. Note that version 2.0 of the simulation program is not intended to use the discrete target file, since a discrete target file is not available for this version.

Minimum Reflectivity (dBz) -20.

This parameter defines the reflectivity "floor" for the microburst data base. For example, if the data base provides a reflectivity of -40 dBz, the value will be reset to the number provided above (-20 dBz).

Attenuation Code (0,1,2) 2.

This code defines the method of calculating path attenuation due to rain. Code "1" is the most accurate but increases running time considerably.

Code 0. = no rain attenuation calculated

Code 1. = attenuation is calculated for each incremental scatterer.

Code 2. = attenuation is calculated along beam center and this value used for each scatterer in the integration volume. This is an approximation that speeds up the calculation.

	0.
RADAR PARAMETERS	0.
Initial Radar Range (km)	1.
Number of Range Cells	50.
Antenna Az - if no scan (deg)	0.
Azimuth Scan Range/2 (deg)	0.
Azimuth Scan Increment (deg)	3.
Antenna Elevation (deg)	2.

These parameters define the angular scan range and the radar range characteristics. If no scan is used (Azimuth Scan Range/2 = 0.), the antenna azimuth is defined by the "Antenna Az-if no scan" parameter. The antenna elevation and azimuth angles are relative to the aircraft mechanical coordinates. If roll, pitch, and yaw are zero, the elevation angle will be relative to the specified glide slope with + angles indicating angles above the glide slope.

Transmitted Power (watts)	2000.
Frequency (GHz)	9.3
Pulse Width (microsecs)	1.
Pulse Interval (microsecs)	268.6
Receiver Noise Figure (db)	4.
Receiver Losses (db)	3.

The above specify radar parameters and are self-explanatory. The range of frequencies that can be specified is approximately 5-20 GHz when using antenna #1.

Antenna Type (1 = para., 2 = flat 3 = measured data)	3.

Type 1 is a generic parabolic antenna, type 2 is a calculated Collins-type flat antenna, and type 3 uses an input data set consisting of measured pattern data of a Collins flat antenna. (types 2 and 3 can only be used at 9.3 GHz).

Antenna Radius (m)	.381
Aperture Taper Parameter	.316

The above parameters define the parabolic antenna only.

RMS Trans. Phase Jitter (deg)	.2
RMS Trans. Freq. Jitter (Hz)	0.

The above specify the transmitter/receiver phase and frequency stability.

SIGNAL PROCESSING	0.
	0.
Number of Pulses	128.
Number of A/D bits	12.
AGC Gain Factor	.6

The above specify the number of pulses averaged to derive velocity (for pulse-pair processing) and the number of pulses used to determine the Doppler velocity spectra (for spectral the level at which the AGC gain for a range bin is set. For example, if the average signal level in a range bin is S_a, the product of the AGC gain and S_a is made to be equal to the gain factor specified. The simulated A/D sampling range of the resulting signal is -1.0 to 1.0.

Processing Threshold (db)	4.

The above sets the threshold relative to receiver noise above which the velocity and hazard factor calculations are made. If the signal (sum of rain and residual clutter power level above noise) after clutter filtering is below threshold, velocity and hazard factor are set to zero.

Clutter Filter Code (-2 to N)	2.
Clutter Filter Cutoff (m/s)	3.

The above are used to define the type of clutter filter and the filter cutoff in velocity units. If clutter is not calculated, these parameters are not applicable.

The clutter code is:

-2	Two-pole Butterworth filter
-1	Single-pole Butterworth filter
0	No Filter
1	Single pole high-pass filter
2	Two single-pole filters cascaded
.	Three single-pole filters cascaded
.	Etc

No. of Bins for F-factor Avr.	5.

The above indicates the number of range bins used to average the hazard factor. The values 0, 3, and 5 are used depending on the smoothing desired.

4.3 Filename File WXFILES.DAT

This file provides to the program a list of the file names for the microburst data and clutter data input files and user defined names for the output files. The structure of the file is as follows.

UFILE WFILE RFILE	Microburst data filenames
MFILE	Clutter map filename
DFILE	Discrete target file filename
RDFILE	Runway specification file for clutter map
OF1	User-named output file #1
OF2	User-named output file #2
OF3	User-named output file #3

The microburst data file names consists of three files. UFILE is the file containing the u component of microburst wind velocity, the WFILE contains the v component of the microburst wind velocity and the RFILE contains the reflectivity data base. MFILE is the data base for the clutter map from Denver Stapleton or from Willow Run Airport. RDFILE contains clutter map coordinates of the touchdown points of all runways used in the simulation. These coordinates are generally different for each clutter map used. DFILE is the discrete target file data base (not used with version #2 of the simulation - user must include a dummy file here).

The user-named output file #1 (OF1) contains the magnitude of the velocity spectra in each range bin for one range line (at the center of the scan if the radar is in an azimuth scan mode). This is a binary file with a structure as given in section 5.3.

Output file two (OF2) is a user named output file containing the parameters measured by the radar system such as Doppler velocities, spectral widths and power levels (clutter power, rain power, noise power, total power). This file also contains the derived hazard factor calculation. The format and structure of this file is discussed in detail in section 5.3.

OF3 is a condensed version of OF2 containing the variables: 1) radar range, 2) reflectivity, 3) average velocity and 4) hazard index. This file is used for making simulated range/azimuth radar displays of the output variables on a simulated color radar display. Format of this file discussed in section 5.4.

4.4 Microburst Data Input Files

There are nine microburst model files furnished with version 2 of the radar simulation program. Each microburst model consists of three files with a general file name structure as follows:

RRF#.DAT	-Radar Reflectivity
UC#.DAT	-Wind Component #1
WVC#.DAT	-Wind Component #2

In the above, the # sign is replaced by a number that specifies a specific microburst model data set. For example, RRF9.DAT, UC9.DAT, and WVC9.DAT are the data files for a "wet" microburst 9 minutes after initiation of the computer model that generates the data.

Three of the microburst models are "wet" microburst and six of the models are "dry" microburst. The "wet" microburst models are 9, 11 and 13 and the "dry" microburst models are 21, 23, 24, 26, 36, and 40. These numbers refer to the time after initiation of the NASA microburst model discussed in section 2.0.

4.5 Clutter Map Data Input Files and Map Specification Files

Version 2 of the radar simulation program is coded to use several clutter maps. These files are called:

MAP3E.DAT	Willow Run Airport
MAPD#.DAT	Denver Stapleton maps 1-6. The # sign is replaced with the number of map desired.

Each map has an associated input file WXRD#.DAT (Denver maps) or WXRDW.DAT for the Willow Run map. These map specification files give the runway coordinates, map size and scale factor data associated with a specific map.

4.6 Antenna Pattern Data (WXANT.DAT)

This input file consists of measured antenna pattern data taken in the NASA/LaRC anechoic chamber. The data consists of pattern data taken in the two principle planes of the antenna at .1 degree increments over a range of 180 degrees.

This actual pattern data is used in the simulation if antenna option #3 is chosen in the input data file WXIN.DAT. It should be noted that when this antenna is used in the simulation the results are only valid for a frequency of 9.3 GHz.

5.0. OUTPUT FILES

5.1 Output File WXOUT.DAT

This output file provides data giving the Doppler spectra for a selected range bin. It is only generated if the number of range bins is specified as 1. Thus, the file is useful for examining the spectra in single range bins.

The data output are a list of frequencies, spectral magnitudes, and the in-phase and quadrature pulses. The output format is given as Format 8 on the program listing.

5.2 User Named Output File #1 (OF1)

This output file consists of Doppler spectra data for a series of range bins along a range line at the center of the azimuth scan. The data consists of a list of spectral magnitudes starting with range bin 1 and continuing through all range bins. This file is a binary file and is useful for making plots of Doppler spectra versus range. The file is only written if the number of bins specified is greater than 1.

5.3 User Named Output File #2 (OF2)

This file contains the main output parameters of the simulation and consists of fifteen variables as follows:

(1) RRG2 - Radar Range (km)
(2) VPP - Velocity from Pulse Pair Processing (m/s)
(3) VSP - Velocity from Spectral Average (m/s)
(4) VTRU - True Velocity along Antenna Boresight (m/s)
(5) WPP - Spectrum Width from Pulse-Pair Processing (m/s)
(6) PRRN - Power Return from Rain (dBw)

(7) PRCL - Power Return from Clutter (dBw)

(8) PRNS - Power Return from Noise (dBw)

(9) PTOL - Total Power Received (dBw)

(10) FTOL - True Total Hazard Factor (Horizontal + Vertical Comp)

(11) GNDB - AGC Gain (Automatic Gain Control) (dB)

(12) FPP - Hazard Factor From Pulse Pair Processing (Horizontal Component)

(13) FSP - Hazard Factor From Spectral Average (Horizontal Component)

(14) FTRU - True Hazard Factor (Horizontal Component)

(15) ZZDBA - Reflectivity (dBz)

The format statement used for this output file is number 85 on the program code listing. Each series of output data are proceeded by header (Format 1388 on the program listing) giving the scan number antenna elevation, antenna azimuth, the scan increment and the number of range bins. An example of a printout of the data is given in Figure 6C.

5.4 User Named Output File #3 (OF3)

This file is a condensed version of file OF2 providing the variables:

(1) Radar Range (km)

(2) Radar Measured Power Level (dBw)

(3) Velocity from Pulse-Pair Processing (m/s)

(4) Derived Hazard Index

The file is used to make a simulated radar azimuth/range displays of power, velocity or hazard index. The variables above are output for each range line over an azimuth scan for several scans (as specified in input data).

5.5 I and Q Output Data (IQOUT.DAT)

This output file consists of raw I and Q data from the simulated radar. The output data format is given by number 902 in the program listing. This file is only output if the first 0 on the right-hand side of the input file WXIN.DAT is changed to 1. Since there is an I & Q value for each pulse in each range bin, this file can become large in long data runs.

6.0 EXAMPLE DATA RUN

6.1 General Discussion

To provide a check case and to illustrate the method of creating a simulation scenerio, a data run was made using a wet microburst and ground clutter from the Denver Stapleton airport area.

The situation simulated in this example is an aircraft on the landing approach path 7 km from touchdown observing the wet microburst at a range of 5 km from the aircraft (-2 km from the touchdown point). The antenna beam is tilted up 2 degrees from the glide slope angle of 3 degrees.

6.2 Input Data

The input data file WXIN.DAT is shown in Figure 5. Fifty range bins are used with a pulse width of 1 microsecond and with the first range bin at 1 km from the aircraft.

The "wet" microburst #11 is used in the example run. The ground clutter map used represents the Denver Stapleton area with the aircraft landing to the west on runway 26L. The WXFILES.DAT file shown in Figure 6, indicates the names of the microburst, ground clutter map and output files. This figure also shows the clutter map runway specification file used.

6.3 Outputs

Outputs from the example run (taken from the output file WXTEST.DAT are plotted in Figures 7-13.)

Figure 7 shows a plot of the Doppler velocity spectra from a range bin selected at 4 km ahead of the aircraft. This range bin shows the microburst winds flowing toward the aircraft. The stationary ground clutter in the spectra near zero velocity has been effectively suppressed by the clutter filter. Figure 8 plots the power levels at the receiver from ground clutter, rain and receiver noise. These power levels are converted to signal to clutter ratio (SCR), signal to noise ratio (SNR), and plotted in Figure 9. Also plotted on this figure is the dBz value of the rain obtained directly from the microburst data base.

Figure 10 plots the average velocity derived by the radar signal processor using both pulse-pair processing and spectral averaging techniques. The figure also shows the "true" velocity, which is the radar velocity components taken along the antenna boresight line. The windshear due to the microburst model is clearly evident in these velocity plots. Figure 11 shows the horizontal component of the hazard index derived from the velocity signatures given in Figure 10. In addition, this figure plots the true horizontal component of the hazard factor derived from the true velocity as well as the total hazard factor which includes both the vertical component (not measured by the radar) and the horizontal component along the antenna beam boresight line.

Figure 12 indicates the AGC gain required versus range along the selected range line. It should be recalled that each range bin is assumed to have an independent AGC. Figure 13 shows the effectiveness of the clutter filter in the simulated receiver in reducing the ground clutter signal level. At short ranges from the radar, the clutter suppression is not effective because the ground clutter signal moves out of the clutter filter notch due to geometric effects.

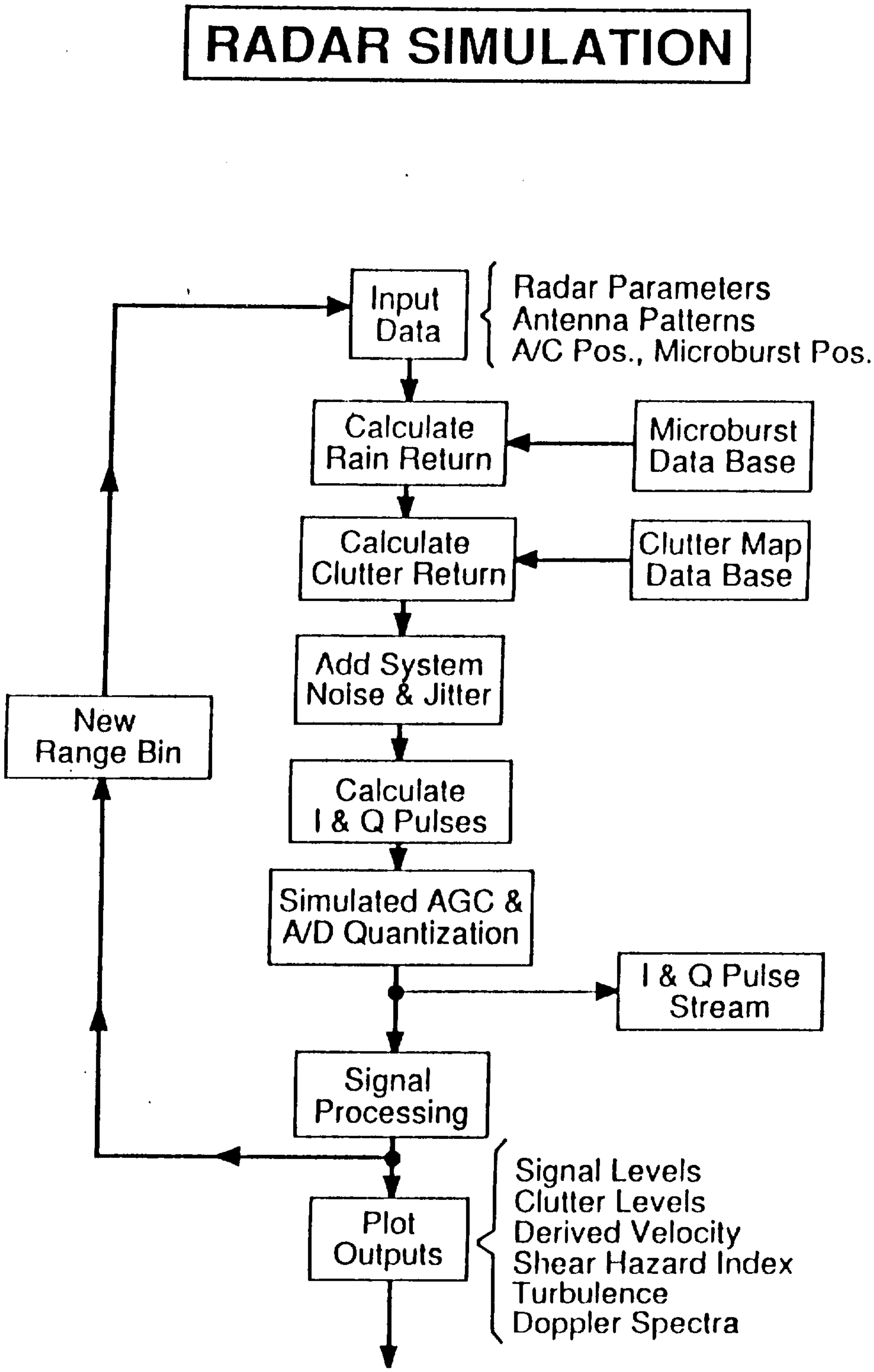


Figure 1 FLOW DIAGRAM SHOWING THE GENERAL OPERATION OF THE WINDSHEAR RADAR SIMULATION PROGRAM

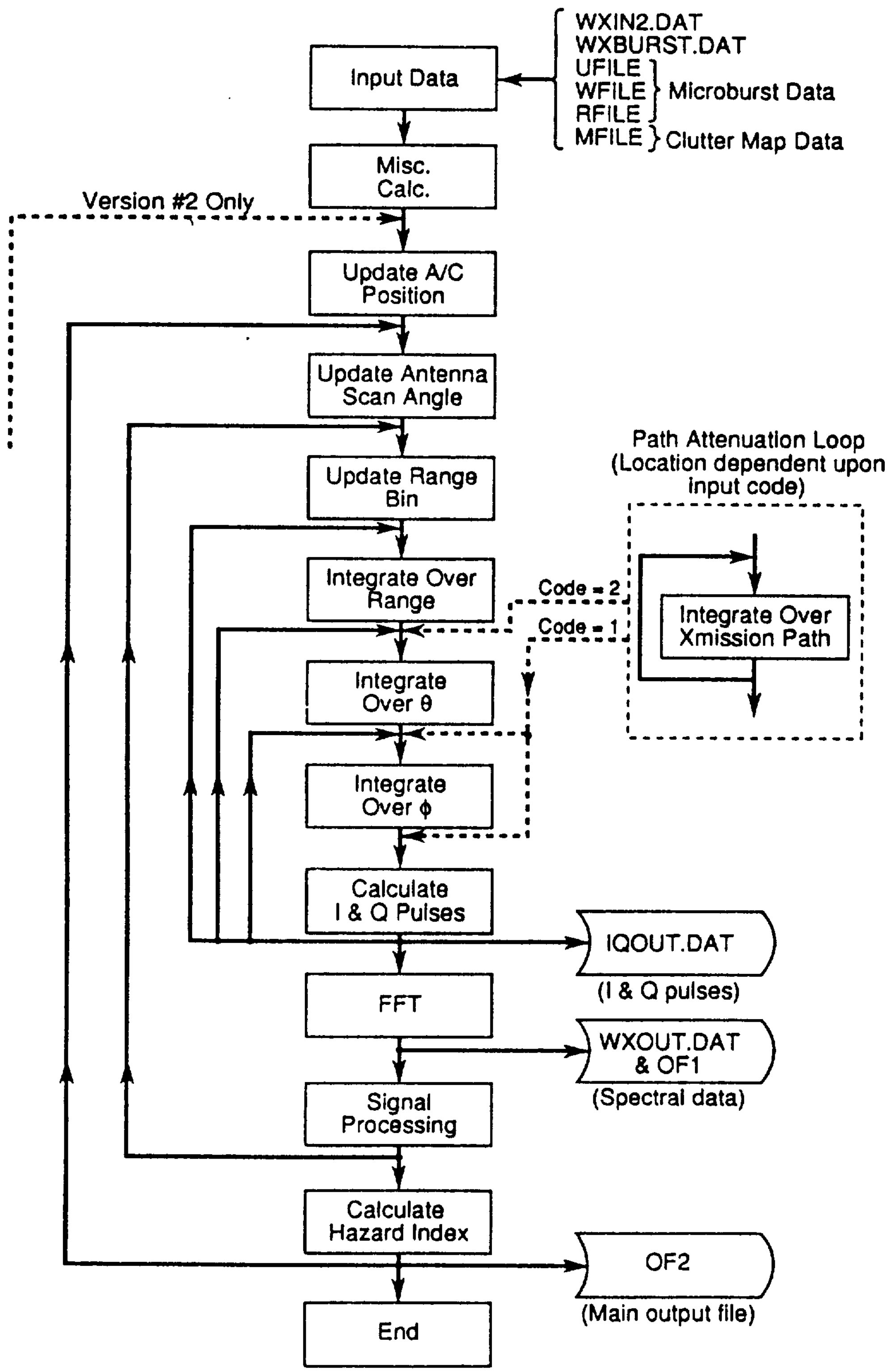


Figure 2 LOGICAL FLOW DIAGRAM SHOWING THE MAIN CALCULATION LOOPS IN THE PROGRAM

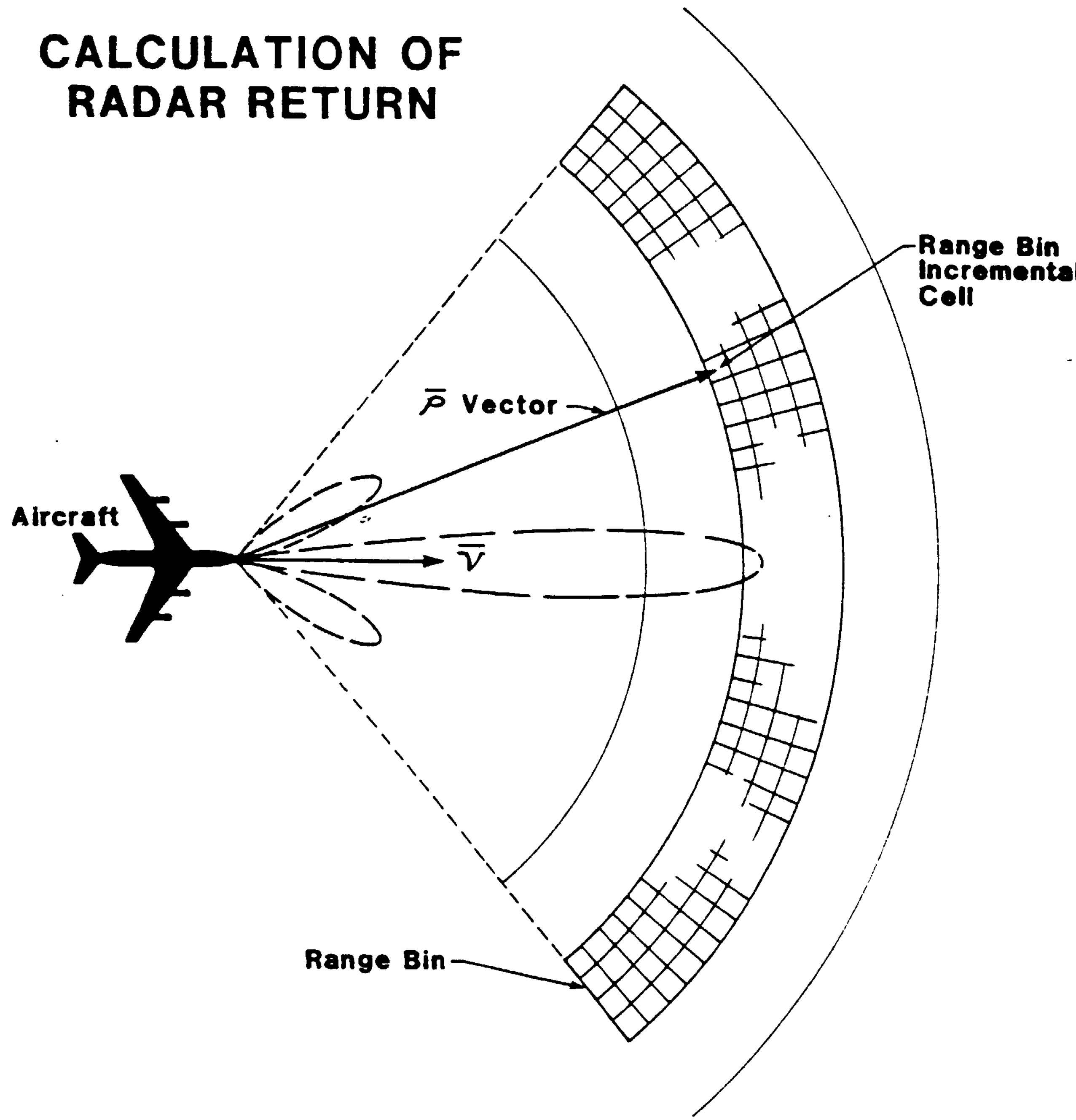
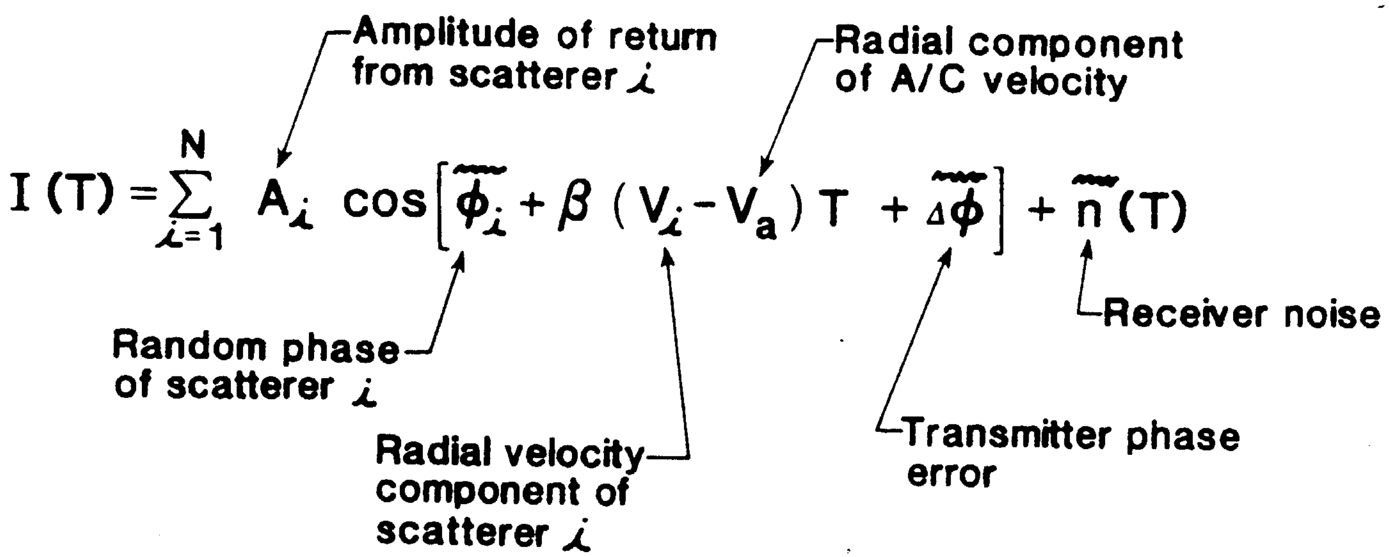


Figure 3 SKETCH SHOWING THE BREAKDOWN OF A RANGE BIN INTO INCREMENTAL SCATTERING CELLS.

CALCULATION OF I & Q



$$I(T) = \sum_{i=1}^{N} A_i \cos\left[\tilde{\phi}_i + \beta (V_i - V_a) T + \widetilde{\Delta\phi}\right] + \tilde{n}(T)$$

$$\beta = \frac{2\pi}{\lambda_o}$$

T = Time

Figure 4 TECHNIQUES FOR CALCULATION OF IN-PHASE (I) AND QUADRATURE (Q) SIGNALS

```
****SIMULATION PARAMETERS*****              0.
                                            0.   (I & Q data output flag)
A/C Distance to Touchdown (km)              7.
Aircraft Velocity (kts)                     150.
Glideslope Angle (deg)                      3.
No. of Complete Scans                       1.
Time Between Scans (sec.)                   5.
Roll Attitude (deg)                         0.
Pitch Attitude (deg)                        0.
Yaw Attitude (deg)                          0.
Az Integration Range/2 (deg)                6.0
Az Integration Increment (deg)              .3
El Integration Range/2 (deg)                4.0
El Integration Increment (deg)              .2
Rng Integration Increment (m)               100.
Random Number Seed (0-1)                    .224
Runway Number                               26.
Right (1) or Left (2)                       2.
                                            0.
****MICROBURST & CLUTTER******              0.
                                            0.
Along Track Offset from TD (km)             -2.
Cross Track Offset from TD (km)             0.
Rain Standard Deviation (m/s)               1.
Clutter Standard Deviation (m/s)            .5
Clutter Calc. Flag (1=ON,0=OFF)             1.
Discrete Calc. Flag (1=ON,0=OFF)            0.
Reflectivity Calc. Thres. (dBz)             1.
Minimum Reflectivity (dBz)                  -20.
Attenuation Code (0,1,2)                    2.
                                            0.
****RADAR PARAMETERS**********              0.
                                            0.
Initial Radar Range (km)                    1.0
Number of Range Cells                       50.
Antenna Az - if no scan (deg)               0.
Azimuth Scan Range/2 (deg)                  0.
Azimuth Scan Increment (deg)                3.
Antenna Elevation (deg)                     2.
Transmitted Power (watts)                   2000.
Frequency (GHz)                             9.3
Pulse Width (microsecs)                     1.
Pulse Interval (microsecs)                  268.6
Receiver Noise Figure (dB)                  4.
Receiver Losses (db)                        3.
Ant Type (1 , 2, or 3)                      3.
Antenna Radius (m)                          .381
Aperture Taper Parameter                    .316
RMS Trans. Phase Jitter (deg)               .2
RMS Trans. Freq Jitter (Hz)                 0.
                                            0.
****SIGNAL PROCESSING*********              0.
                                            0.
Number of Pulses                            128.
Number of A/D bits                          12.
AGC Gain Factor                             .6
Processing Threshold (dB)                   4.
Clutter Filter Code (-2 to N)               2.
Clutter Filter Cutoff (m/s)                 3.
No. of Bins for F-factor Avr.               5.
```

Figure 5 INPUT FILE WXIN.DAT. THIS FILE DEFINES THE PARAMETERS OF THE SIMULATION

```
UC11.DAT
WVC11.DAT
RRF11.DAT
MAPd1.DAT
WXDOUT1.DAT
WXRD1.DAT
WXSPEC.DAT
WXTEST.DAT
WXMAPS2.DAT
```

6a - WXFILES. DAT

```
8  270.  915  683  457  341  19.20  19.44  2
8  2  569  309  667  303
26 1  667  303  569  309
8  1  569  294  686  294
26 2  686  294  569  294
17 2  648  600  641  446
35 1  641  446  648  600
17 1  623  503  611  355
35 2  611  355  623  503
1213.08  1723.0
-6.0
```

6b - WXRD1.DAT

```
1     2.00     0.00     3.00   1
4.07,  10.84,  10.37,   9.96,-201.07, -92.12,-143.25,-140.22,-167.91,  0.000,   2.07,  0.000,  0.000,  0.000,  46.80
1.22,  -2.60,  -1.43,   1.12,-201.07,-201.07,-128.05,-139.69,-135.20,  0.000,   9.46,  0.000,  0.000,  0.000, -20.00
1.37,  -2.67,  -2.09,   1.29,-201.07,-201.07,-129.60,-139.65,-136.51, -0.018,  11.74, -0.001, -0.012, -0.007, -20.00
1.52,  -2.39,  -2.36,   1.51,-201.07,-201.07,-124.10,-140.16,-132.88, -0.026,   7.87, -0.010, -0.003, -0.012, -20.00
1.67,  -2.48,  -0.25,   1.81,-201.07,-201.07,-124.50,-139.72,-137.46, -0.038,  13.59, -0.013,  0.003, -0.017, -20.00
1.82,  -1.63,  -1.10,   2.21,-201.07,-201.07,-121.86,-139.59,-135.61, -0.054,  13.33, -0.010, -0.012, -0.024, -20.00
```

6c - WXTEST.DAT

Figure 6 FILENAME FILE WXFILES.DAT, CLUTTER MAP SPECIFICATION FILE WXRD1.DAT AND OUTPUT DATA FILE WXTEST.DAT

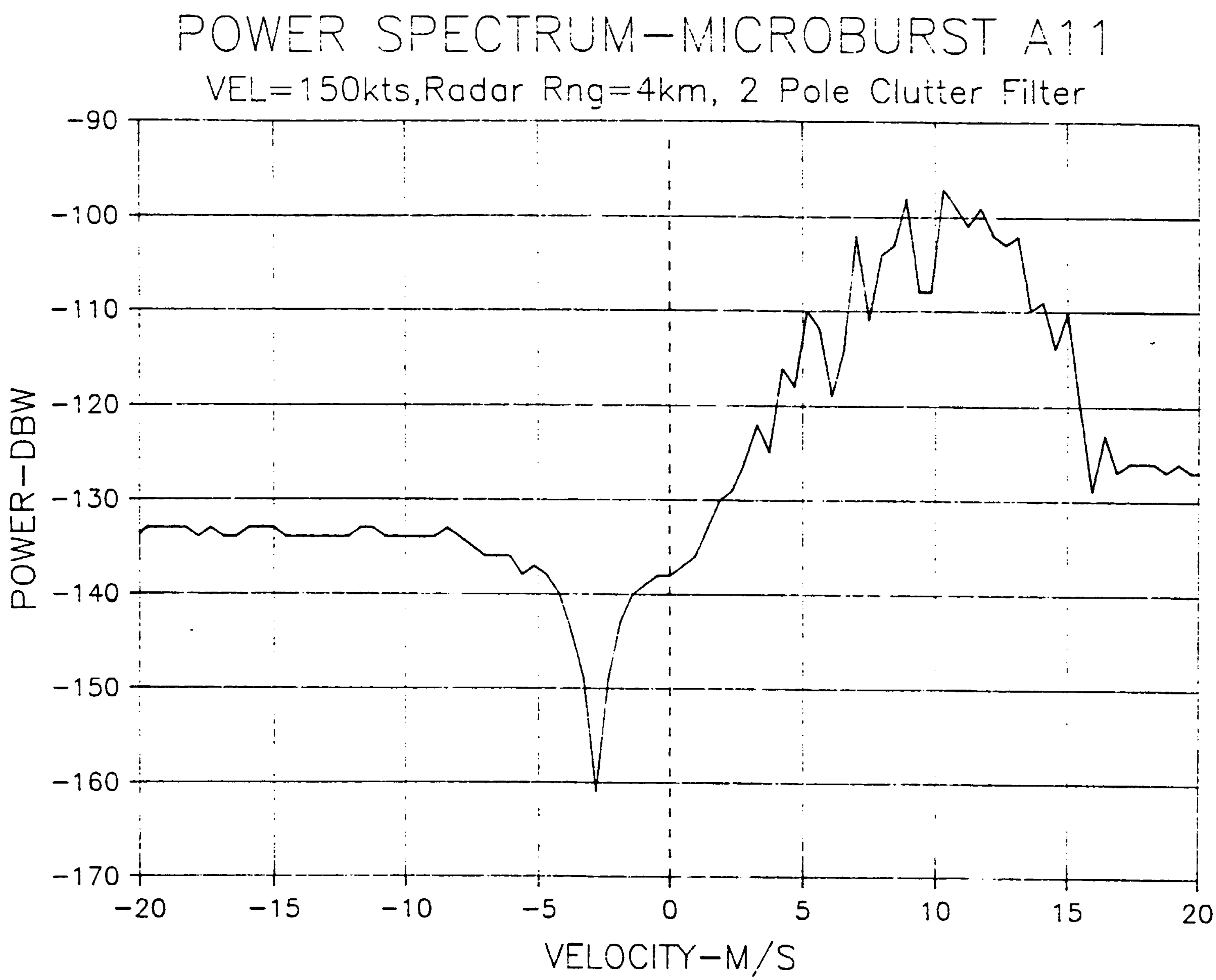


Figure 7 PLOT OF THE DOPPLER SPECTRA FOR A RANGE BIN 4KM FROM THE AIRCRAFT IN THE EXAMPLE DATA RUN.

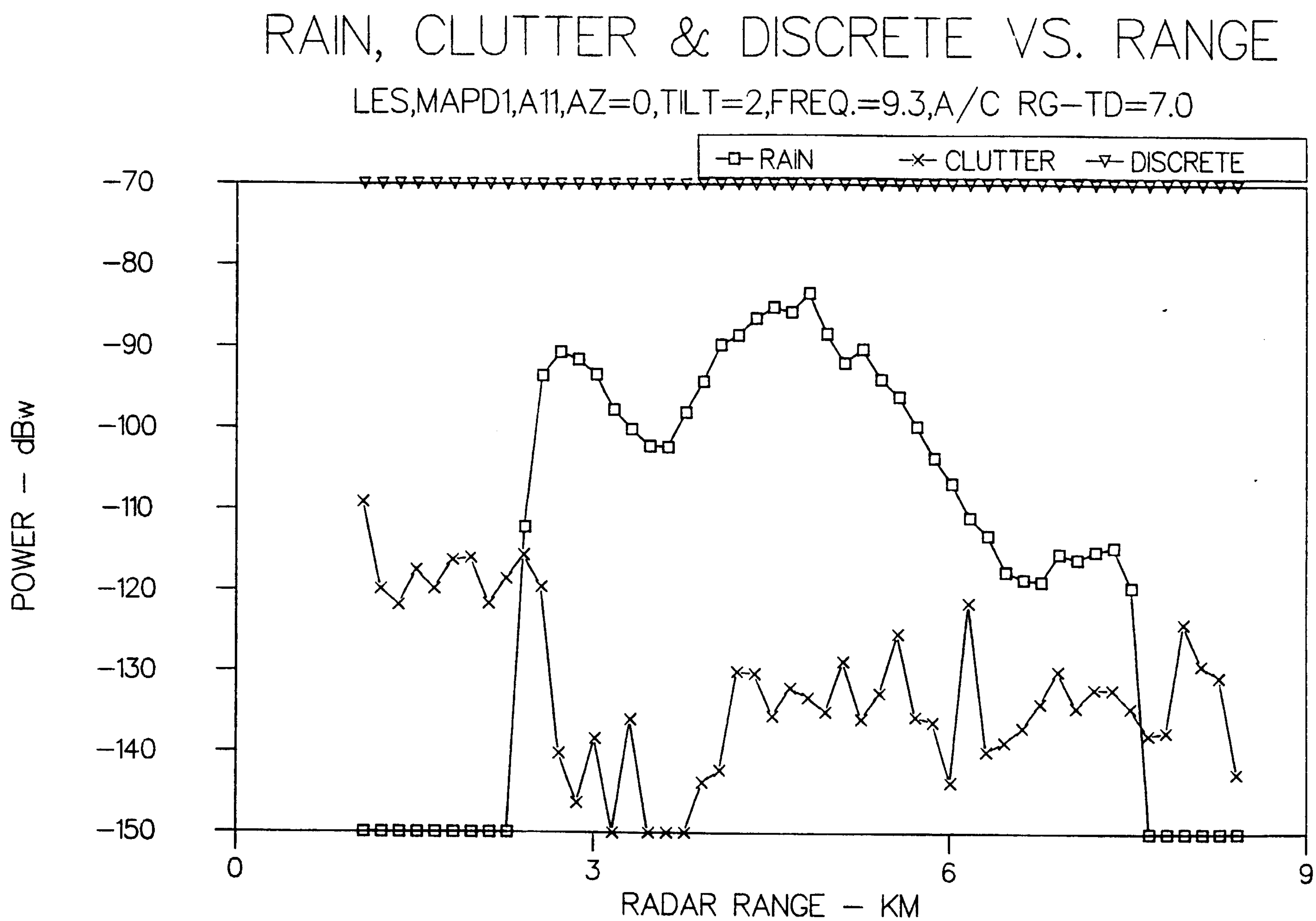


Figure 8 RAIN, CLUTTER AND DISCRETE POWERS PLOTTED VS. RADAR RANGE. THE AIRCRAFT TOUCHDOWN POINT IS AT 7KM. NO DISCRETE TARGETS WERE USED IN THIS RUN, HENCE THE DISCRETE PLOT IS MEANINGLESS.

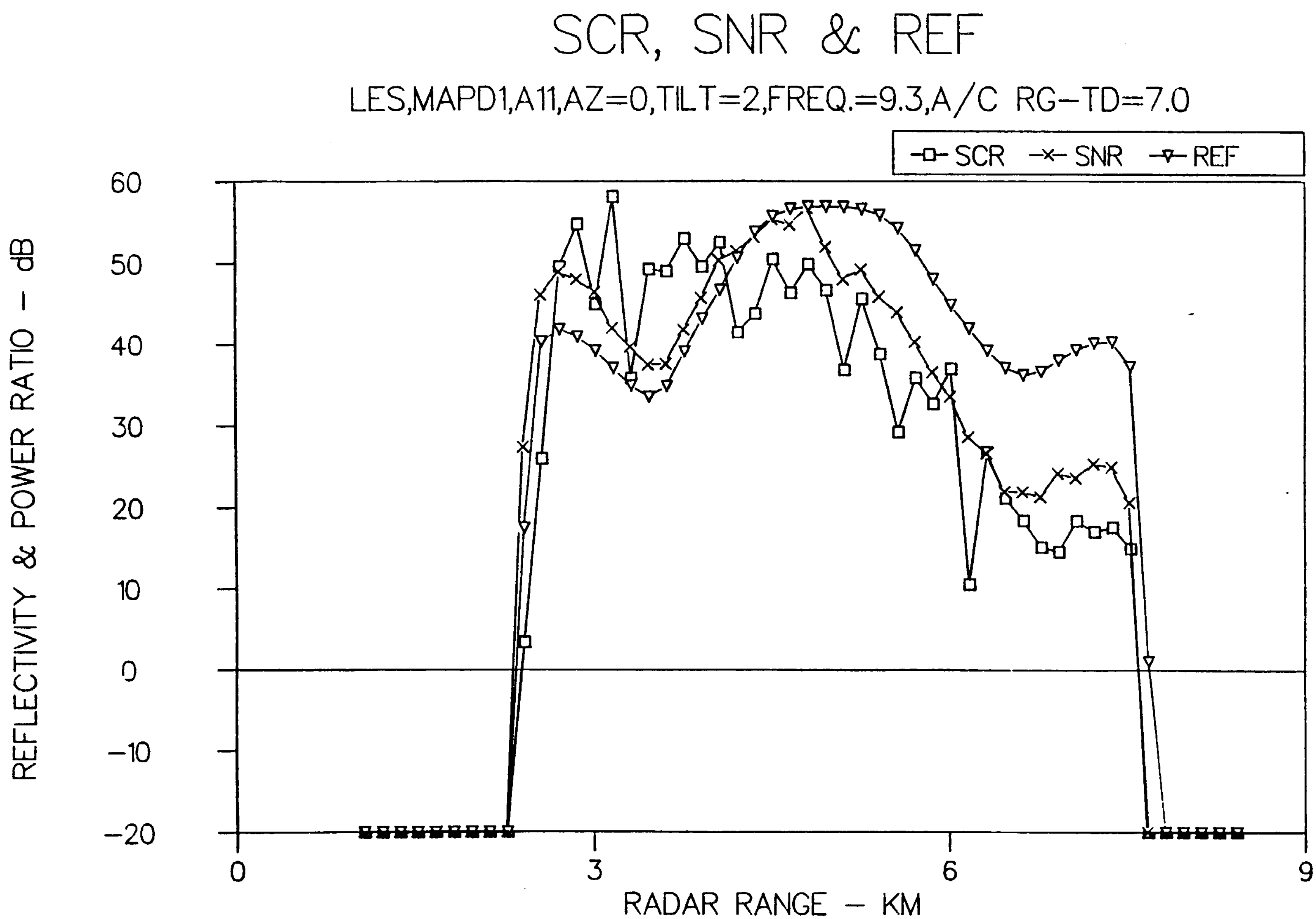


Figure 9 RAIN-TO-CLUTTER AND RAIN-TO-NOISE POWER RATIOS. THE dBz VALUE (REF) OF THE MICROBURST DATA ALONG THE ANTENNA BORESIGHT IS ALSO PLOTTED

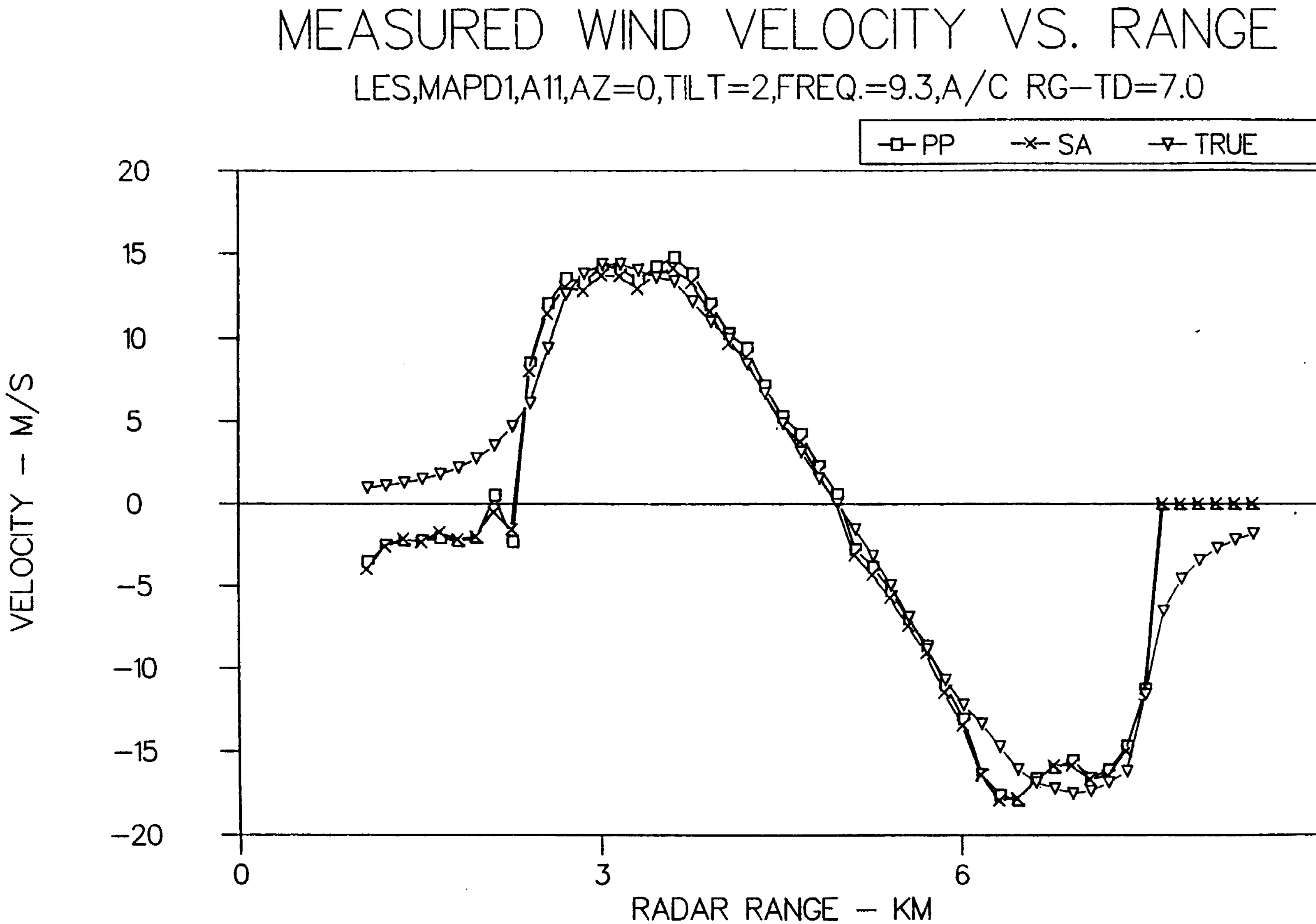


Figure 10 RADAR DERIVED WIND VELOCITY VS. RADAR RANGE FOR PULSE-PAIR (PP) AND SPECTRAL AVERAGE (SA) PROCESSING. THE "TRUE" VELOCITY, OR VELOCITY ALONG THE ANTENNA BORESIGHT LINE, IS ALSO PLOTTED.

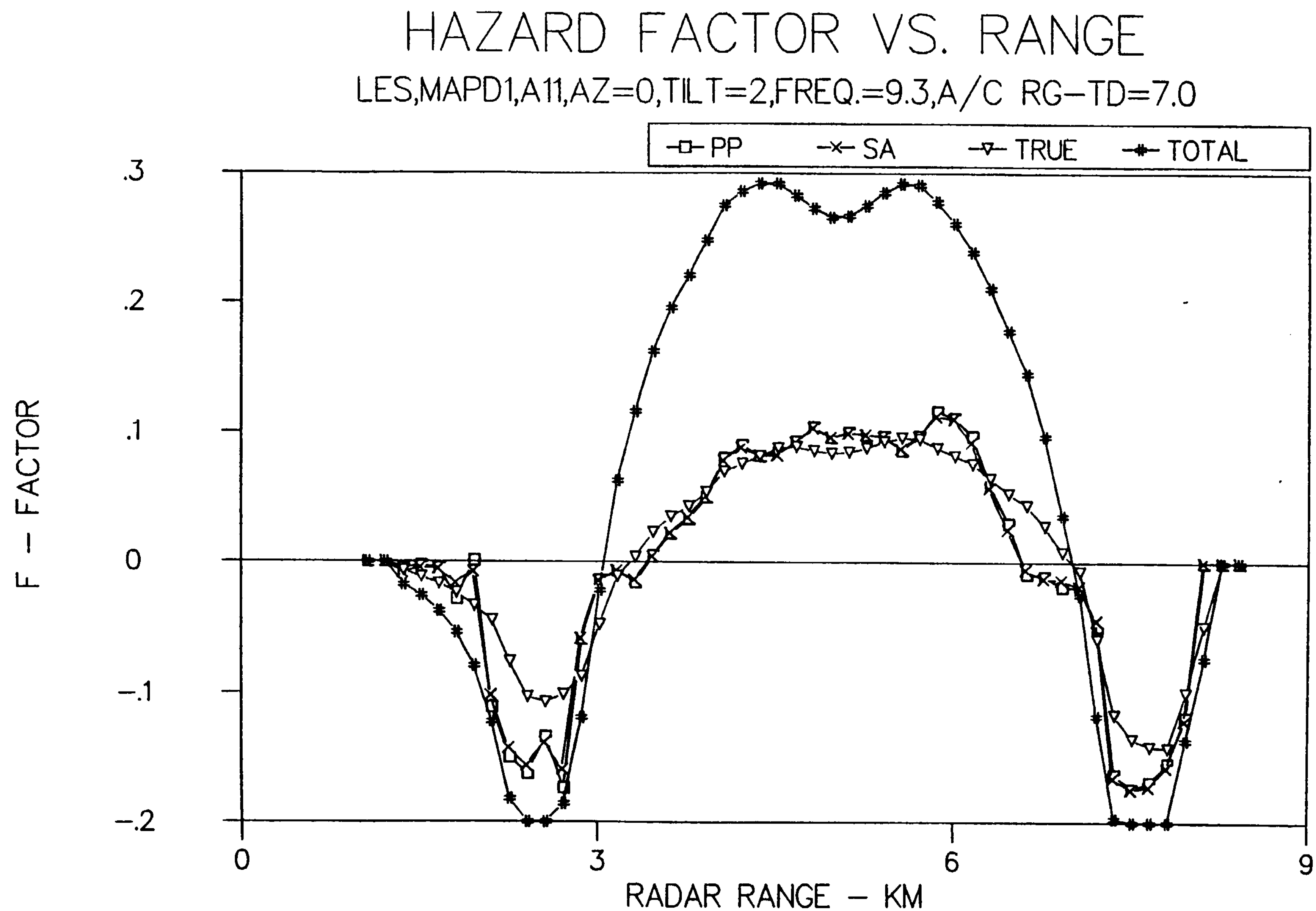


Figure 11 HAZARD INDEX VS. RADAR RANGE AS DERIVED FROM PULSE-PAIR (PP), SPECTRAL AVERAGES (SA) AND TRUE WIND VELOCITY (TRUE). THE TOTAL HAZARD INDEX INCLUDING THE VERTICAL COMPONENT (NOT MEASURED BY THE RADAR) IS ALSO PLOTTED.

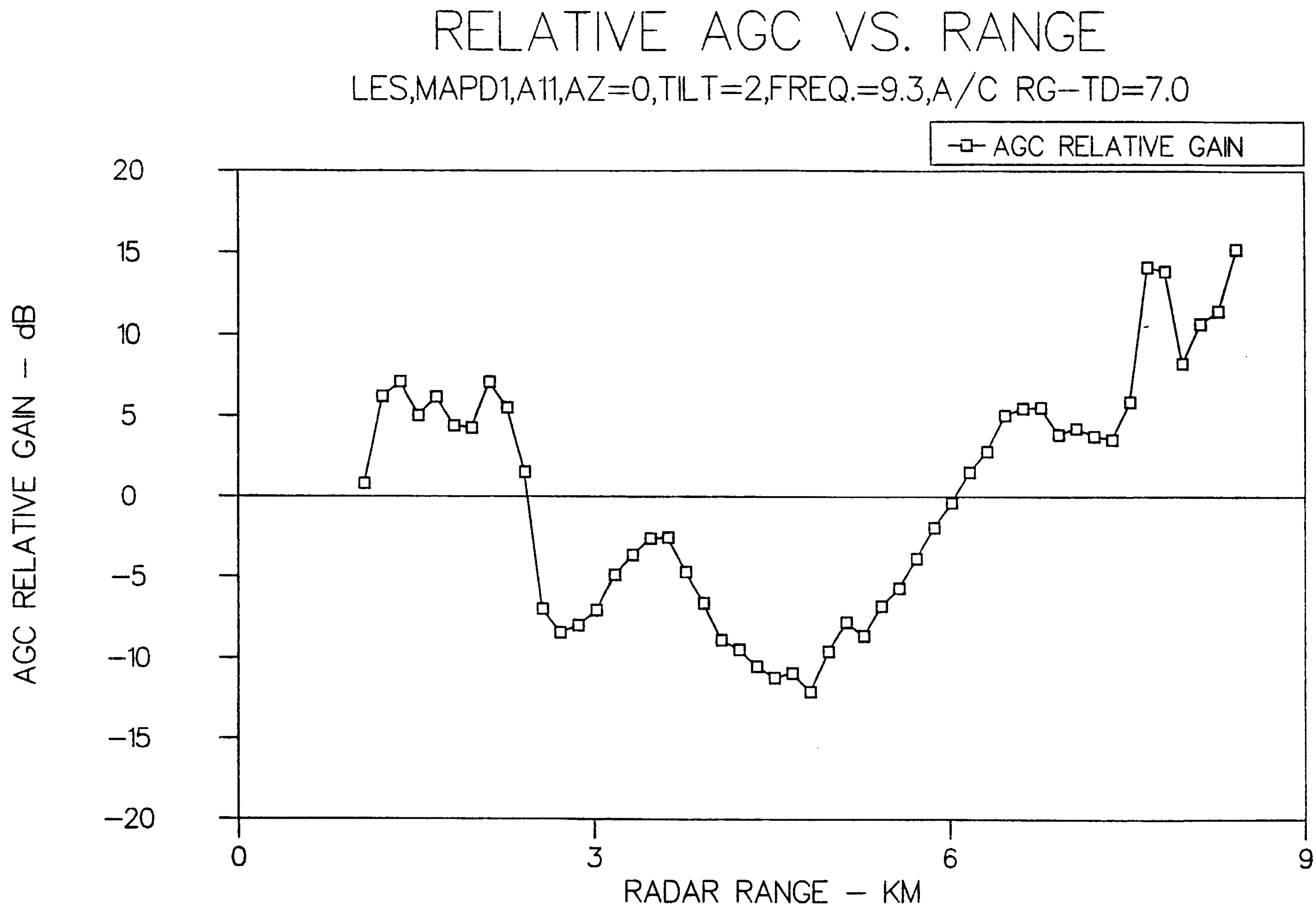


Figure 12 RELATIVE AGC GAIN VS. RADAR RANGE. THE 0 LEVEL IS THE AVERAGE VALUE OF GAIN. EACH RANGE BIN IS ASSUMED TO HAVE AN INDEPENDENT AGC LEVEL.

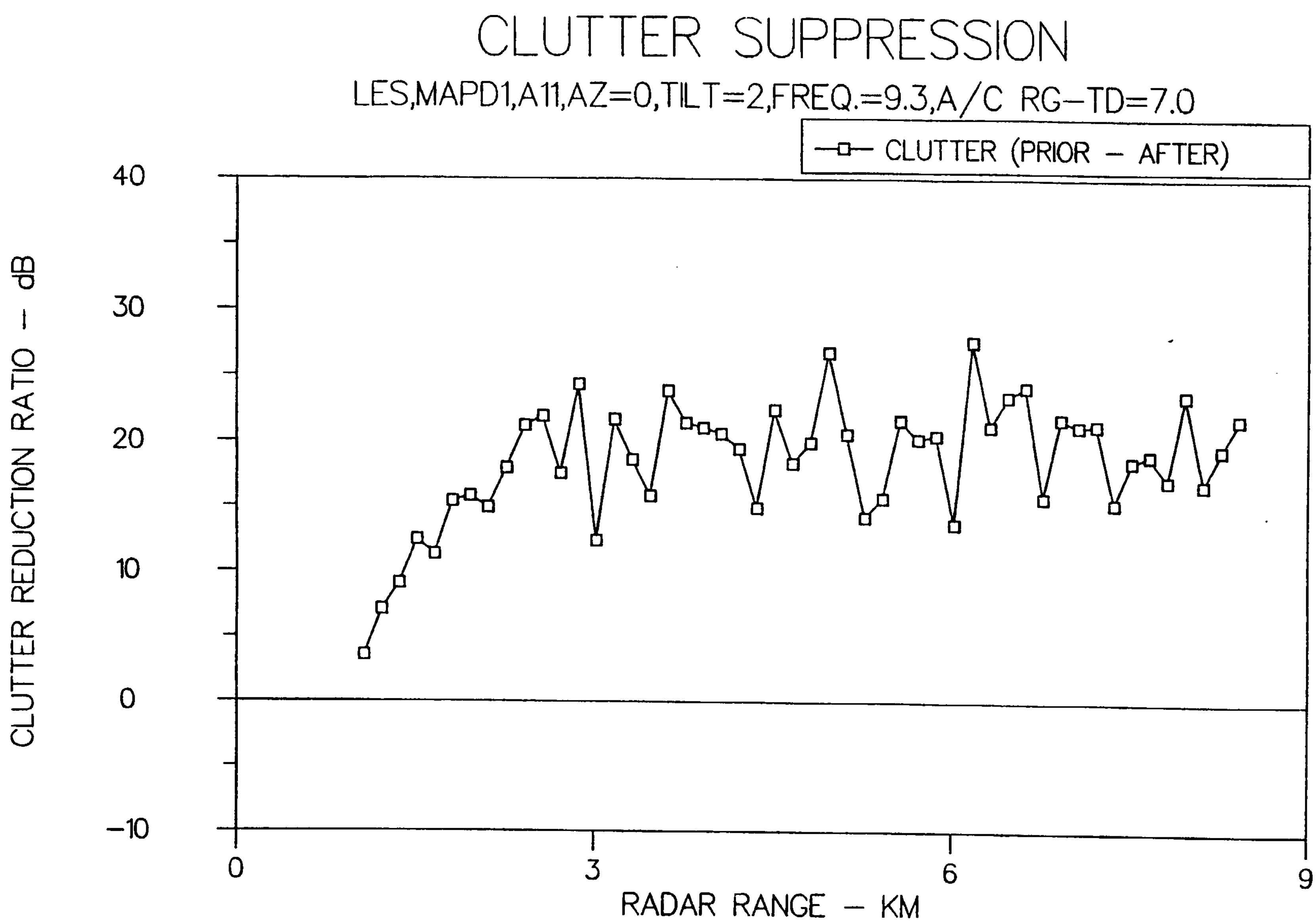


Figure 13 PLOT OF THE RATIO OF GROUND CLUTTER POWER SUPPRESSED BY THE CLUTTER FILTER ALGORITHM

APPENDIX A

MAIN PROGRAM LISTING

```
      PROGRAM WXMAIN1
C*********************************************************************
C
C     WINDSHEAR RADAR SIMULATION PROGRAM (VERSION 2.0)
C
C     THIS PROGRAM WAS WRITTEN FOR NASA, LANGLEY RESEARCH
C     CENTER UNDER CONTRACT NAS1-17639 BY:
C
C           CHARLES L. BRITT, Ph.D
C           RESEARCH TRIANGLE INSTITUTE
C           610 THIMBLE SHOALS BLVD., SUITE 203B
C           NEWPORT NEWS, VA 23606
C           Ph. 804-864-1786 or 804-873-1164
C
C     REVISION 8/9/89 - ADDS ANT PATTERN
C     REVISION 8/23/89 - ADDS DISCRETE TARGETS & NEW MAP
C     ROUTINES
C
C*********************************************************************
      IMPLICIT INTEGER (I-N)
      INTEGER*2 MAP(1000000),IDATA
      CHARACTER MFILE*20,DFILE*20,RWFILE*20,OF3*20,RORL*8
      CHARACTER HEAD*33,UFILE*20,WFILE*20,RFILE*20,OF1*20,OF2*20
      DIMENSION R1(3,3),R2(3,3),R3(3,3),R4(3,3),R5(3,3)
      DIMENSION R6(3,3),R7(3,3),RT(3,3),RE(3,3),RFIN(3,3)
      DIMENSION RHO(3),RHO1(3),RHOS(3),VRN(3),AI(4),OV(15,100)
      DIMENSION RV(3,3),VU(3),VX(3),RHOA(3),RFIN2(3,3)
      DIMENSION PCLUT(512),PRAIN(512),PTOT(512),PNSE(512)
     2    ,PDISC(512)
      DIMENSION CC1(512),CC2(512),ANTDAT(1800,2)
      DIMENSION EE(3000),VVV(3000),AAA(3000),RHOBM(3),VP(3)
      DIMENSION E(5000),VV(5000),AA(5000),VSAVE(3,2),RP(3)
      DIMENSION U1(512),U2(512),IMAGDB(512),IDATA(512),RPS(3)
      DIMENSION XXD(80,1000),YYD(80,1000),RCSD(80,1000),
     2  VXD(80,1000),VYD(80,1000),NND(80),AAD(1000),EED(1000),
     3  VVD(1000),UPD(80,1000)
      DIMENSION CA(2),IRUSE(6),RMR(3,3),RHOMO(3),RRM(3,3),RHORW(3)
      COMMON/DISC/XXD,YYD,UPD,RCSD,VXD,VYD,NND,NSEG,SFRG,SFAZ,IRES
      COMMON/WIND/UC(101,101),WC(101,101),IRUN,TIME,ISTOP,JSTOP,DR
     2    ,DZ,ZZ(101,101)
      COMMON/CONST/PI,DTR,RTD,CC,FREQ,XLAM
      COMMON/ANT/C1,C2,XK,C12,GAINO
      COMMON/ATTN/CON1,CON2,BETA
      COMMON/BFIL/FC11,FC12,FC21,FC22,FC23
      COMMON/ANTR/ANTDAT
      COMMON/CLUT/RMR,RHOMO,SFR,SFA,IRCODE,CA,IRUSE,JMAXO
     2  ,SIGURB,AREAC,RRM
      DATA AG,XNMTM,XKTTMS,XKMTM,XKB,XTO/
     2  9.80616,1853.2,.5148,1000.,
     3  1.38E-23,290./
C
```

```
      OPEN(4,FILE='WXFILES.DAT',STATUS='OLD')
      READ(4,3333)UFILE
      READ(4,3333)WFILE
      READ(4,3333)RFILE
      READ(4,3333)MFILE
      READ(4,3333)DFILE
      READ(4,3333)RWFILE
      READ(4,3333)OF1
      READ(4,3333)OF2
      READ(4,3333)OF3
 3333  FORMAT(A20)
      OPEN(8,FILE='WXIN.DAT')
C
C ******INPUT DATA************************************
C
      WRITE(*,*)'READING INPUT DATA'
      READ(8,18)DUM1,DUM,RG,VELKT,GSLOP,XNS,XTM,ROL,PIT,YAW,THTMAX,
     2 DELTHT,PMAX,DELP,DELRG,XRN,XW,XRL,DUM,DUM,DUM,YOFF,XOFF,
     3 VELSD,CLUSD,XICLUT,XIDISC,ZTHRES,DBZMIN,XIATTN,DUM,DUM,DUM,
     4 RRG,XNBINS,AZI,AZSCAN,DSCAN,EL,PWRTX,FGZ,PWM,PIM,XNFDB,
     5 XLOSDB,ANTFG,A,B,PHSERR,FREQERR,DUM,DUM,DUM,XN,XNBITS,GFAC,
     6 PTHRES,XIFIL,FILFAC,XNHAVR
      OPEN(5,FILE='WXOUT.DAT')
      OPEN(15,FILE=MFILE,STATUS='OLD')
      OPEN(9,FILE=UFILE,STATUS='OLD')
      OPEN(10,FILE=WFILE,STATUS='OLD')
      OPEN(11,FILE=RFILE,STATUS='OLD')
      OPEN(7,FILE=OF1,FORM='BINARY')
      OPEN(12,FILE=OF3)
      OPEN(13,FILE=OF2)
      OPEN(14,FILE='IQOUT.DAT')
      OPEN(16,FILE='WXANT.DAT',STATUS='OLD')
      OPEN(17,FILE=DFILE,STATUS='OLD')
      OPEN(18,FILE=RWFILE,STATUS='OLD')
      N=1
      CALL BRSTWD3(U,V,W,0.,0.,0.,0.,0.,N)
C
C *****CONSTANTS**************************************
C
      PI=3.141592654
      PI2=2.*PI
      DTR=PI/180.
      RTD=1./DTR
      CC=2.997925E8
      XRN=.352
      XKK=sqrt(.9313)
      NBINS=1
      CONSQR2=1./(2.**.5)
      XX=0.
 18   FORMAT(33X,F10.3)
      IWHICH=INT(XW + .1)
```

```
      IRL=INT(XRL + .1)
      IDISC=INT(XIDISC +.1)
      NSCNS=INT(XNS +.1)
      DELTM=XTM
      CLOSE(8)
      OPEN(8,FILE='WXIN.DAT')
      NIN=60
      IQOUT=INT(DUM1 + .1)
      DUM=0.
      WRITE(*,1776)
 1776 FORMAT(//)
      DO 821 I=1,NIN
      READ(8,19)HEAD,DUM
      WRITE(*,191)HEAD,DUM
 821  CONTINUE
      WRITE(*,1776)
 19   FORMAT(A33,F10.3)
 191  FORMAT(1X,A33,F12.3)
      CALL READRWY(IWHICH,IRL,IMAX,JMAX)
C      WRITE(*,*)IMAX,JMAX
      ICLUT=INT(XICLUT+.1)
      IF(ICLUT .EQ. 1)CALL RDCLUT5(IMAX,JMAX,MAP)
C      WRITE(*,*)'BACK FROM RDCLUT ROUTINE'
      CALL READANT
      IF(IDISC .EQ. 1)CALL RDDISC
C
C ******MISC CALCULATIONS*****************************
C
      NHAVR=INT(XNHAVR + .1)
      NLINE=INT(2.*AZSCAN/DSCAN)+1
      NLINE2=NLINE/2
      IF(NLINE2 .LE. 0)NLINE2=1
      ALPHA=10.**(2.660*ALOG10(FGZ)-4.5631)
      BETA=10.**(-.1669*ALOG10(FGZ)+.2586)
      CON1=4.34E3
      CON2=ALPHA/CON1
      WRITE(*,*)'MISC CALCULATIONS'
      IATTN=INT(XIATTN + .1)
      IF(XIFIL .GE. 0.)IFIL=INT(XIFIL + .1)
      IF(XIFIL .LT. 0.)IFIL=-1*INT(ABS(XIFIL) + .1)
      IANT=INT(ANTFG + .1)
      PFAC=10.**(PTHRES/10.)
      ZFAC=10.**(ZTHRES/10.)
      ROO=RG*XKMTM
      XOFF=XOFF*XKMTM
      YOFF=YOFF*XKMTM
      RRGO=RRG
      NBINS=INT(XNBINS+.1)
      NLEV=INT(2.**(XNBITS-1.) + .1)
      THTMIN=-1.*THTMAX
      NTHT=INT((THTMAX-THTMIN)/DELTHT)
```

```
      VEL=VELKT*XKTTMS
      FREQ=FGZ*1.E9
      PWIDTH=PWM*1.E-6
      PRR=PIM*1.E-6
      NTIME=INT(XN+.1)
      XLAM=CC/FREQ
      C1=PI2*A*A
      C2=2.*PI2/(XLAM*XLAM)
      C12=C1*C2
      XK=PI2/XLAM
      GAINO=C12/2.
      DELT=PRR
      XLOS=10.**(XLOSDB/10.)
      CU=4.*PI*FREQ/CC
      VU(1)=VEL
      VU(2)=0.
      VU(3)=0.
      VELIN1=VEL
      CONR=((PI**5./XLAM**4.)*XKK**2.)*1.E-18
      VRN(1)=0.
      VRN(3)=0.
      XBW=1./PWIDTH
      XNF=10.**(XNFDB/10.)
      XPNSE=XNF*XTO*XBW*XKB
      SDNSE2=SQRT(XPNSE/2.)
      SDNSE=SQRT(XPNSE)
      BINSZ=PWIDTH*CC/2.
      NRG=INT(BINSZ/DELRG + .1)
      SDPHS=PHSERR*DTR
      SDFREQ=FREQERR
      VMAX=CC/(4.*FREQ*PRR)
      CALL FILCO(VMAX,FILFAC)
      FC=VEL/(AG*BINSZ)
      CF=4.*PI*DELT*FILFAC/XLAM
      IF(CF .GE. 1.)WRITE(*,*) 'ERROR IN FILTER FACTOR'
      WRITE(*,*)'CALCULATING MATRICES'
C
C ******CALCULATE MATRICES****************************
C
      CALL ROT(3,-90.,R3)
      CALL ROT(1,180.,R1)
      CALL GMPRD(R3,R1,RT,3,3,3)
      CALL ROT(2,-GSLOP,R1)
      CALL GMPRD(R1,RT,R3,3,3,3)
      CALL GMTRA(R3,RV,3,3)
      CALL EMAT(ROL,PIT,YAW,RE)
      CALL GMPRD(RE,R3,R4,3,3,3)
C
C****************AIRCRAFT MOTION LOOP**************
C
      DO 1998 ITM=1,NSCNS
```

```
      RO=ROO-VEL*(ITM-1)*DELTM
      IF(RO .LE. 0.)GO TO 1998
       ZO=-RO*SIN(GSLOP*DTR)
       YO=RO*COS(GSLOP*DTR)
       XO=0.
      BX=0. + XOFF
      BY=RO*COS(GSLOP*DTR) + YOFF
C
C****************AZ SCAN LOOP**********************
C
      DO 544 JS=1,NLINE
      AZ=-AZSCAN + (JS-1)*DSCAN
      IF(NLINE .EQ. 1)AZ=AZI
      CALL ROT(3,AZ,R5)
      CALL ROT(2,EL,R6)
      CALL GMPRD(R6,R5,R7,3,3,3)
      CALL GMPRD(R7,R4,RFIN,3,3,3)
      CALL ROT(2,90.,R1)
      CALL ROT(3,90.,R2)
      CALL GMPRD(R2,R1,R3,3,3,3)
      CALL GMPRD(R3,RFIN,RFIN2,3,3,3)
      CALL GMPRD(RV,VU,VX,3,3,1)
C
C *****ANTENNA ANGLE WITH RESPECT TO HORIZON*******
C
      CALL HORIZ(RFIN,VP,RPS)
      HANG=90.-RPS(3)
      THTST=90.-RPS(2)-THTMAX
C      WRITE(*,*)HANG,VP(1),VP(2),VP(3)
      PHI1=RPS(3)-PMAX
      PHI2=RPS(3)+PMAX
      VELIN=VELIN1*SIN(RPS(2)*DTR)
      CALL GETBW(IANT,A,B, BMWID)
      BMWID2=BMWID/2.
C      WRITE(*,*)IANT, BMWID
C
C *****CALCULATE RET. SIGNALS FROM RAIN AND CLUTTER*******
C
c      WRITE(7,6)NTIME,DELT,VEL,FREQ
  6   FORMAT(I5,3E12.4)
C
C******RANGE BIN LOOP********************************
C
      DO 320 III=1,2
      DO 321 JJJ=1,3
 321  VSAVE(JJJ,III)=0.
 320  CONTINUE
      IF(ITM.EQ.1.AND.NLINE.EQ.NLINE2)
     2     WRITE(13,1388)NLINE,EL,AZ,DSCAN,NBINS
      IF(NLINE.GT.1)
     2     WRITE(12,1389)NLINE,EL,AZ,DSCAN,NBINS,DELTM,NSCNS
```

```
 1389   FORMAT(1X,I4,3F8.2,I4,F8.2,I4)
        DO 888 MM=1,NBINS
        RRG=RRGO+(MM-1)*BINSZ/XKMTM
        RRGC=RRG + BINSZ/(2.*XKMTM)
        WRITE(*,458)MM,RRG,AZ,ITM
 458    FORMAT(1X,/,1x,
      2   'CALCULATING BIN NO.',i4,'  RG = ',f5.1,'  AZ = ',F5.1,
      3   '  SCAN = ' ,I4,/)
        CALL VELTRU(RFIN,VX,VELIN,RO,RRGC,ZO,BX,BY,FVER,VTRU)
C
C *****RANGE CALCULATIONS****************************
C
        SRMIN=RRG*XKMTM
        SRMAX=SRMIN + PWIDTH*CC/2.
        RRNG=SRMIN + (SRMAX-SRMIN)/2.
        TDELAY=2.*RRNG/CC
        DELRG=(SRMAX-SRMIN)/NRG
        CP=PWRTX*XLAM*XLAM/((4.*PI)**3.* RRNG**4.* XLOS)
        CPRT=SQRT(CP)
        XKMAX=NTHT*NRG
        K=0
        KK=0
        KNTZ=0
        IF(IQOUT .EQ. 1)THEN
           WRITE(14,901)MM,RRNG
           WRITE(*,*)'IQ DATA WRITTEN TO FILE'
        ENDIF
 901    FORMAT(I4,E12.4)
C
C *****RANGE LOOP**************************************
C
 112    FORMAT(1X,3I10,9X,3I10)
        XLOSS=1.
        DO 577 MMM=1,3
 577    RP(MMM)=VP(MMM)*RRNG
        IF(IATTN .EQ. 2)CALL ATTEN(RP,ZO,BINSZ,BX,BY, XLOSS)
        SUMZZ=0.
 1388   FORMAT(1X,I4,3F8.2,I4)
        DO 10 I=1,NRG
        SR=SRMIN + DELRG*(I-1) + DELRG/2.
        PSI=ASIN(ZO/SR)
        TEST=PHI2
        PSID=ABS(PSI)*RTD + 90.
        IF(TEST .GE. PSID)PHI2=PSID
C        WRITE(*,*)TEST,PSID,PHI2,PHI1
        NPHI=INT((PHI2-PHI1)/DELP)
        NPROD=NPHI*NTHT*NRG
        AI(1)=90.-ABS(PSI)*RTD
        AI(2)=AI(1)*AI(1)
        AI(3)=AI(2)*AI(1)
        AI(4)=AI(3)*AI(1)
```

```
      SRCPS=SR*COS(PSI)
      AREAC=SR*DELRG*DELTHT*DTR/COS(PSI)
      AREA=SR*DELRG*DELTHT*DTR
      CVOL=AREA*SR*DELP*DTR
 100  FORMAT(1X,2I4)
C
C******THETA LOOP***************************************
C
      DO 20 J=1,NTHT
      KK=KK+1
C
C *****CALCULATE CLUTTER RETURN**********************
C
      THTR=(THTST + DELTHT*(J-1) + DELTHT/2.)*DTR
      RHO(1)=SRCPS*SIN(THTR)
      RHO(2)=SRCPS*COS(THTR)
      RHO(3)=ZO
      CALL GMPRD(RFIN2,RHO,RHOA,3,3,1)
      CALL SPHER(RHOA,RHOS)
      ANTANG=RHOS(3)
      ANTTHT=RHOS(2)
      IWRT=1
      IF(IANT .EQ. 1)CALL ANTPAT(A,B,ANTANG,GAIN)
      IF(IANT .EQ. 2)CALL ANTFLAT(A,B,ANTANG,GAIN)
      IF(IANT .EQ. 3)CALL ANTREAL(ANTANG,ANTTHT,GAIN)
      GDB=10.*ALOG10(GAIN)
 333  FORMAT(1X,6F8.2)
      XM=RHO(1)
      YM=RHO(2)-YO
      SIGODB=-90.
      IF(ICLUT .EQ. 1)CALL CLUT5(XM,YM,IMAX,JMAX,MAP,AI, SIGODB)
      IWRT=0
C      WRITE(*,111)I,J,XM,YM,AI(1),SIGODB,ANTANG,ANTTHT,GDB
 111  FORMAT(1X,2I5,7F8.1)
      SIGO=10.**(SIGODB/10.)
      IF(IATTN .EQ. 1)CALL ATTEN(RHO,ZO,BINSZ,BX,BY, XLOSS)
      CSIGC=SIGO*AREAC*CP*XLOSS
      SRSIGC=SQRT(CSIGC)
      EE(KK)=SRSIGC*GAIN
      CALL STRNUM(XRN)
      AAA(KK)=PI2*XRN
      CALL NORMAL(XX)
      DCLUV=XX*CLUSD
      VVV(KK)=(((VX(1)*RHO(1)+VX(2)*RHO(2)+VX(3)*RHO(3))/SR-VELIN)
     2      + DCLUV)*CU
C
C *****PHI LOOP - CALCULATE RAIN RETURN******************
C
 308  DO 30 L=1,NPHI
      PHI=PHI2-(L-1)*DELP - DELP/2.
      PHIR=PHI*DTR
```

```
      RHO(1)=SR*SIN(THTR)*SIN(PHIR)
      RHO(2)=SR*COS(THTR)*SIN(PHIR)
      RHO(3)=SR*COS(PHIR)
      AXX=RHO(1)
      AYY=RHO(2)
      AZZ=RHO(3)-ZO
      DBZ=ZTHRES
      CALL BRSTWD3(U,V,DBZ,BX,BY,AXX,AYY,AZZ,3)
      ZZZ=10**(DBZ/10.)
      IF(DBZ.LT.ZTHRES)GO TO 30
      IF(DBZ .LE. DBZMIN)DBZ=DBZMIN
      K=K+1
      CALL GMPRD(RFIN2,RHO,RHOA,3,3,1)
      CALL SPHER(RHOA,RHOS)
      ANTANG=RHOS(3)
      ANTTHT=RHOS(2)
      IF(IANT .EQ. 1)CALL ANTPAT(A,B,ANTANG,GAIN)
      IF(IANT .EQ. 2)CALL ANTFLAT(A,B,ANTANG,GAIN)
      IF(IANT .EQ. 3)CALL ANTREAL(ANTANG,ANTTHT,GAIN)
      CALL NORMAL(XX)
      DV=XX*VELSD
C       WRITE(*,877)BX,BY,AXX,AYY,AZZ,RO,ZO,SRMIN,SRMAX,SR,SRCPS
 877  FORMAT(1X,11F6.0)
      CALL BRSTWD3(U,V,W,BX,BY,AXX,AYY,AZZ,2)
      VRN(1)=U
      VRN(2)=V + DV
      VRN(3)=W
      VXX=VX(1)-VRN(1)
      VYY=VX(2)-VRN(2)
      VZZ=VX(3)-VRN(3)
      VV(K)=((VXX*RHO(1)+VYY*RHO(2)+VZZ*RHO(3))/SR -VELIN)*CU
C
C *****CALCULATE RAIN AMPLITUDE********************
C
      IF(IATTN .EQ. 1)CALL ATTEN(RHO,ZO,BINSZ,BX,BY, XLOSS)
      ETA=ZZZ*CONR
      TEST=ABS(ANTANG)-BMWID2
      ZADD=0.
      IF(TEST .LE. 0.)THEN
        ZADD=ZZZ
        KNTZ=1+KNTZ
      ENDIF
      SUMZZ=SUMZZ + ZADD
      SRETA=SQRT(ETA)
      CSIG=ETA*CVOL*CP*XLOSS
      SRSIG=SQRT(CSIG)
      E(K)=SRSIG*GAIN
      CALL STRNUM(XRN)
      AA(K)=XRN*PI2
  30  CONTINUE
  20  CONTINUE
```

```
   10 CONTINUE
      KMAX=K
      IF(KNTZ .GT. 1 .AND. SUMZZ .GT. 0.)THEN
       ZZAVR=SUMZZ/KNTZ
       ZZDBA=10.*ALOG10(ZZAVR)
      ELSE
       ZZDBA=DBZMIN
      ENDIF
      KKMAX=KK

C
C*************DISCRETE TARGET RETURN*******************
C
      KKD=0
      IF(IDISC .NE. 1)GO TO 1869
      DO 1868 KS=1,NSEG
      KTMAX=NND(KS)
      DO 1867 KT=1,KTMAX
      RHO(1)=XXD(KS,KT)+XO
      RHO(2)=YYD(KS,KT)+YO
      RHO(3)=UPD(KS,KT)+ZO
      RGD=SQRT(RHO(1)**2+RHO(2)**2+RHO(3)**2)
      IF(RGD.LT.SRMIN.OR.RGD.GT.SRMAX)GO TO 1865
      IF(RHO(2) .LT. 200.)GO TO 1865
      KKD=KKD+1
      CALL GMPRD(RFIN2,RHO,RHOA,3,3,1)
      CALL SPHER(RHOA,RHOS)
      ANTANG=RHOS(3)
      ANTTHT=RHOS(2)
      IF(ANTANG .GE. 85.)ANTANG =85.
      IF(IANT .EQ. 1)CALL ANTPAT(A,B,ANTANG,GAIN)
      IF(IANT .EQ. 2)CALL ANTFLAT(A,B,ANTANG,GAIN)
      IF(IANT .EQ. 3)CALL ANTREAL(ANTANG,ANTTHT,GAIN)
      GDB=10.*ALOG10(GAIN)
      SIGO=10.**(RCSD(KS,KT)/10.)
      IF(IATTN .EQ. 1)CALL ATTEN(RHO,ZO,BINSZ,BX,BY, XLOSS)
      CSIGD=SIGO*CP*XLOSS
      SRSIGD=SQRT(CSIGD)
      EED(KKD)=SRSIGD*GAIN
      CALL STRNUM(XRN)
      AAD(KKD)=PI2*XRN
      VXX=VX(1)+VXD(KS,KT)
      VYY=VX(2)+VYD(KS,KT)
      VZZ=VX(3)
      VVD(KKD)=(((VXX*RHO(1)+VYY*RHO(2)+VZZ*RHO(3))/RGD-VELIN))*CU
      GEE=20.*ALOG10(EED(KKD))
C     WRITE(*,*)KKD,KS,KT,RHO(1),RHO(2),RGD,GEE,gdb,VVD(KKD)/CU
 1865 continue
 1867 CONTINUE
 1868 CONTINUE
 1869 CONTINUE
```

```
      KDMAX=KKD
      WRITE(*,*)'          NRG       NTHT      NPHI    SAMPLES:     RAIN
     2 CLUTTER   DISCRETE '
      WRITE(*,112)NRG,NTHT,NPHI,KMAX,KKMAX,KDMAX
C
C ******TIME LOOP************************************
C
      WRITE(*,*)'WRITING OUTPUT PULSE TRAIN'
      TIME=0.
      DO 50 II=1,NTIME
      TT=TIME + DELT*(II-1)
      CALL NORMAL(XX)
      DELPHS=XX*SDPHS
      CALL NORMAL(XX)
      DELFREQ=XX*SDFREQ*PI2
      DPHASE=DELPHS + DELFREQ*TDELAY
C
C*******SUM LOOP*************************************
C
      SUM1=0.
      SUM2=0.
      SUM1C=0.
      SUM2C=0.
      SUM1D=0.
      SUM2D=0.
      XMMDB=-99.
C
C *****RAIN CELLS*************************************
C
      DO 60 KRN=1,KMAX
      PHASE= AA(KRN)+VV(KRN)*TT + DPHASE
      XXX=E(KRN)
      SUM1 = SUM1 + XXX*COS(PHASE)
 60   SUM2 = SUM2 + XXX*SIN(PHASE)
      PRAIN(II)=SUM1*SUM1 + SUM2*SUM2

C
C *****CLUTTER CELLS**********************************
C
      DO 62 KKC=1,KKMAX
      PHASE= AAA(KKC)+VVV(KKC)*TT + DPHASE
      XXX=EE(KKC)
      SUM1C = SUM1C + XXX*COS(PHASE)
 62   SUM2C = SUM2C + XXX*SIN(PHASE)
      PCLUT(II)=SUM1C*SUM1C + SUM2C*SUM2C
      CC1(II)=SUM1C
      CC2(II)=SUM2C
C
C******DISCRETE TARGETS******************************
C
      IF(IDISC .NE. 1)GO TO 621
```

```
      DO 620 KKKD=1,KDMAX
      PHASE= AAD(KKKD)+VVD(KKKD)*TT + DPHASE
      XXX=EED(KKKD)
      SUM1D = SUM1D + XXX*COS(PHASE)
 620  SUM2D = SUM2D + XXX*SIN(PHASE)
 621  PDISC(II)=SUM1D*SUM1D + SUM2D*SUM2D
C
C *****ADD RECEIVER NOISE*****************************
C
      CALL NORMAL(XX)
      PN1=XX*SDNSE2
      SUM1=SUM1+PN1
      CALL NORMAL(XX)
      PN2=XX*SDNSE2
      SUM2=SUM2+PN2
      PNSE(II)=PN1*PN1 + PN2*PN2
      U1(II)=SUM1+SUM1C+SUM1D
      U2(II)=SUM2+SUM2C+SUM2D
      PTOT(II)=U1(II)*U1(II) + U2(II)*U2(II)
 50   CONTINUE
C
C********ANALOG TO DIGITAL CONVERSION***************
C
      WRITE(*,*)'A/D CONVERSION OF I & Q SIGNALS'
      CALL ADCONV(U1,U2,NTIME,NTIME,NLEV,GFAC,SDNSE,GN)
C
C********FILTER I & Q PULSES************************
C
      WRITE(*,*)'CLUTTER FILTER',IFIL,NTIME
      IF(IFIL .GT. 0)THEN
        DO 8888 JJJ=1,IFIL
        CALL DFILTER(CC1,CC2,CF,NTIME)
        WRITE(*,*)'ONE STAGE CLUTTER FILTER COMPLETED'
 8888   CALL DFILTER(U1,U2,CF,NTIME)
      ELSEIF(IFIL .LT. 0)THEN
        CALL EFILTER(U1,U2,IFIL,NTIME)
        CALL EFILTER(CC1,CC2,IFIL,NTIME)
      ELSEIF(IFIL .EQ. 0)THEN
        WRITE(*,*)'NO CLUTTER FILTER'
      ENDIF
 787  CONTINUE
C
C********AVERAGE POWER CALCULATIONS*****************
C
      WRITE(*,*)'AVERAGE POWER CALCULATIONS'
      PRRN=1.E-18
      PRCL=1.E-18
      PDIS=1.E-18
      PRNS=0.
      PTOL=0.
      PRCP=1.E-18
```

```
      PTEST=1.E-30
      DO 782 JJJ=1,NTIME
      PRCP=CC1(JJJ)**2 + CC2(JJJ)**2 + PRCP
      PRRN=PRRN + PRAIN(JJJ)
      PRCL=PRCL + PCLUT(JJJ)
      PRNS=PRNS + PNSE(JJJ)
      PDIS=PDIS + PDISC(JJJ)
      PTOL=PTOL + PTOT(JJJ)
  782 CONTINUE
      PRES=PTOL-PRRN-PRCL
      IF(PRES .LE. PTEST)PRES=PTEST
      STHRES=PFAC*PRNS/NTIME
      PRRN=10.*ALOG10(PRRN/NTIME)
      PRCL=10.*ALOG10(PRCL/NTIME)
      PRNS=10.*ALOG10(PRNS/NTIME)
      PTOL=10.*ALOG10(PTOL/NTIME)
      PRES=10.*ALOG10(PRES/NTIME)
      PRCP=10.*ALOG10(PRCP/NTIME)
      PDIS=10.*ALOG10(PDIS/NTIME)
      GNDB=10.*ALOG10(GN)
C
C *******SIGNAL PROCESSING CODE**********************
C
      WRITE(*,*)'SIGNAL PROCESSING'
      CALL PPP(U1,U2,NTIME,PRR,VEL,STHRES,VPP,WPP)
 85   FORMAT(1X,F7.2,1H,,F7.2,1H,,F7.2,1H,,F7.2,1H,,F7.2,
     2    1H,,F7.2,1H,,F7.2,1H,,F7.2,1H,,F7.2,1H,,F7.3,1H,,F7.2,
     3    1H,,F7.3,1H,,F7.3,1H,,F7.3,1H,,F7.2)
      CALL WXANAL(U1,U2,NTIME,DELT,VEL,STHRES,IMAGDB,VSP)
C
C*****CALCULATE F-FACTOR FROM VELOCITIES*******************
C
      IF(MM.GE.3)CALL FFAC(FC,VSAVE,VPP,VSP,VTRU,FPP,FSP,FTRU)
C
C*****WRITE OUTPUT FOR PLOTS OF DATA VS RANGE**************
C
      IF(IQOUT .EQ. 1)WRITE(14,902)(U1(I),U2(I),I=1,NTIME)
 902  FORMAT(2E12.5)
      RRG2=RRNG/XKMTM
      FTOL=FVER+FTRU
      CALL SETVAR(RRG2,VPP,VSP,VTRU,PDIS,PRRN,PRCL,PRNS,PRCP,FTOL,
     2  WPP,FPP,FSP,FTRU,ZZDBA,MM,OV)
C      WRITE(*,85)RRG2,VPP,VSP,VTRU,PDIS,PRRN,PRCL,PRNS,PRCP,FTOL,
C     2   GNDB,FPP,FSP,FTRU,ZZDBA
      NN=NTIME
      DO 889 I=1,NN
      IF(I.LE.NN/2)II=I+NN/2
      IF(I.GT.NN/2)II=I-NN/2
      IDATA(II)=IMAGDB(I)
 889  CONTINUE
C
```

```
C ******WRITE OUTPUT SPECTRAL DATA FOR PLOT IF NBINS=1 *******
C
      IF(NBINS .NE. 1)GO TO 990
      DELF=1./(NTIME*DELT)
      DELV=DELF*CC/(2.*FREQ)
      WRITE(*,*)'WRITING PLOT OUTPUT DATA'
      DO 989 I=1,NTIME
      OUT1=+(I-1)*DELV -(NTIME/2)*DELV
      OUT2=IDATA(NTIME-I+1)*1.
      WRITE(5,8)OUT1,OUT2,U1(I),U2(I)
 8    FORMAT(1X,F8.2,1H,,F8.2,1H,,E10.2,1H,,E10.2)
989   CONTINUE
C
C ******WRITE DATA FOR SPECTRAL PLOTS IF NBINS GT 1******
C
990   CONTINUE
      IF(NBINS .EQ. 1)GO TO 888
      IF(NLINE .EQ. NLINE2)WRITE(7)(IDATA(I),I=1,NTIME)
C
C*********SAVE VELOCITY FROM THIS RANGE BIN*********
C
      DO 322 III=1,3
322   VSAVE(III,2)=VSAVE(III,1)
      VSAVE(1,1)=VPP
      VSAVE(2,1)=VSP
      VSAVE(3,1)=VTRU
888   CONTINUE
C
C**********COMPUTE HAZARD INDEX AND OUTPUT DATA*****

      CALL AVRHAZ(OV,NHAVR,NBINS)
      IF(ITM.EQ.1.AND.NLINE.EQ.NLINE2)
     2   WRITE(13,85)((OV(K,MMM),K=1,15),MMM=1,NBINS)
      IF(NLINE.GT.1)THEN
         DO 1999 MMM=1,NBINS
1999     WRITE(12,86)OV(1,MMM),OV(2,MMM),OV(9,MMM),OV(12,MMM)
      ENDIF
86    FORMAT(4F8.3)
544   CONTINUE
1998  CONTINUE
      WRITE(*,*)'PROGRAM COMPLETED'
      END
```

APPENDIX B

```
      SUBROUTINE RDDISC
C*************************************************************
C     READS DISCRETE TARGET FILE
C     AND CONVERTS DATA TO RUNWAY ORIGIN COORDINATES
C     ---MUST BE CALLED AFTER RUNWAY DATA ARE READ
C*************************************************************
      IMPLICIT INTEGER (I-N)
      DIMENSION XXD(80,1000),YYD(80,1000),RCSD(80,1000),
     2  VXD(80,1000),VYD(80,1000),NND(80),UPD(80,1000)
      DIMENSION R1(3),R2(3),RMR(3,3),RHOMO(3),RRM(3,3),IRUSE(6)
      DIMENSION CA(2)
      COMMON/DISC/XXD,YYD,UPD,RCSD,VXD,VYD,NND,NSEG,SFRG,SFAZ,IRES
      COMMON/CLUT/RMR,RHOMO,SFR,SFA,IRCODE,CA,IRUSE,JMAXO
     2  ,SIGURB,AREAC,RRM
      READ(17,*)NSEG,ANG,JMAXO,IMAXO,JMAXR,IMAXR,SFRG,SFAZ,IRES
      READ(17,*)(NND(I),I=1,NSEG)
      READ(17,*)((XXD(I,J),J=1,NND(I)),I=1,NSEG)
      READ(17,*)((YYD(I,J),J=1,NND(I)),I=1,NSEG)
      READ(17,*)((UPD(I,J),J=1,NND(I)),I=1,NSEG)
      READ(17,*)((RCSD(I,J),J=1,NND(I)),I=1,NSEG)
      READ(17,*)((VXD(I,J),J=1,NND(I)),I=1,NSEG)
      READ(17,*)((VYD(I,J),J=1,NND(I)),I=1,NSEG)
      WRITE(*,*)'DISCRETE TARGET FILE READ'
C
C  **********CONVERT TO COORDINATES AT RUNWAY ORIGIN**********
C
      DO 10 I=1,NSEG
      DO 20 J=1,NND(I)
      R1(1)=XXD(I,J)-RHOMO(1)
      R1(2)=-YYD(I,J)-RHOMO(2)
      R1(3)=0.
      CALL GMPRD(RMR,R1,R2,3,3,1)
      XXD(I,J)=R2(1)
      YYD(I,J)=R2(2)
      R1(1)=VXD(I,J)
      R1(2)=-VYD(I,J)
      R1(3)=0.
      CALL GMPRD(RMR,R1,R2,3,3,1)
      VXD(I,J)=R2(1)
      VYD(I,J)=R2(2)
 20   CONTINUE
 10   CONTINUE
      RETURN
      END
C
      SUBROUTINE P(XMAT)
      DIMENSION XMAT(3,3)
      WRITE(*,*)'  '
      WRITE(*,12)XMAT(1,1),XMAT(1,2),XMAT(1,3)
      WRITE(*,12)XMAT(2,1),XMAT(2,2),XMAT(2,3)
      WRITE(*,12)XMAT(3,1),XMAT(3,2),XMAT(3,3)
```

```
   12  FORMAT(1X,3F8.2)
       RETURN
       END
C***********************************************************************
       SUBROUTINE SETVAR(RRG2,VPP,VSP,VTRU,WPP,PRRN,PRCL,PRNS,PTOL,
     2        FTOL,GNDB,FPP,FSP,FTRU,ZZDBA,MM,OV)
C***********************************************************************
       IMPLICIT INTEGER(I-N)
       DIMENSION OV(15,100)
       OV(1,MM)=RRG2
       OV(2,MM)=VPP
       OV(3,MM)=VSP
       OV(4,MM)=VTRU
       OV(5,MM)=WPP
       OV(6,MM)=PRRN
       OV(7,MM)=PRCL
       OV(8,MM)=PRNS
       OV(9,MM)=PTOL
       OV(10,MM)=FTOL
       OV(11,MM)=GNDB
       OV(12,MM)=FPP
       OV(13,MM)=FSP
       OV(14,MM)=FTRU
       OV(15,MM)=ZZDBA
       RETURN
       END
C
C****************************************************************
       SUBROUTINE FFAC(FC,VSAVE,VPP,VSP,VTRU,FPP,FSP,FTRU)
C****************************************************************
C      CALCULATES F-FACTOR OR HAZARD INDEX
C****************************************************************
       IMPLICIT INTEGER (I-N)
       DIMENSION VSAVE(3,2)
       FPP=-FC*(VPP-VSAVE(1,1))
       FSP=-FC*(VSP-VSAVE(2,1))
       FTRU=-FC*(VTRU-VSAVE(3,1))
       RETURN
       END
C
C**********************************************
       SUBROUTINE AVRHAZ(OV,NN,NBINS)
C**********************************************
       IMPLICIT INTEGER (I-N)
       DIMENSION OV(15,100),FPP(100),FSP(100),FTR(100),FTO(100)
       IF(NN .LE. 2)RETURN
       JSTART=(NN+1)/2
       JSTOP=NBINS-JSTART+1
       DO 20 J=1,NBINS
       IF(J.GE.JSTART.AND.J.LE.JSTOP)THEN
        IDEX2=J+(NN-1)/2
```

```
      IDEX1=J-(NN-1)/2
      FPP(J)=0.
      FSP(J)=0.
      FTR(J)=0.
      FTO(J)=0.
      DO 55 K=IDEX1,IDEX2
      FPP(J)=OV(12,K)/NN+FPP(J)
      FSP(J)=OV(13,K)/NN+FSP(J)
      FTR(J)=OV(14,K)/NN+FTR(J)
 55   FTO(J)=OV(10,K)/NN+FTO(J)
     ELSE
      FPP(J)=0.
      FSP(J)=0.
      FTR(J)=0.
      FTO(J)=0.
     ENDIF
 20  CONTINUE
     DO 30 J=1,NBINS
     OV(12,J)=FPP(J)
     OV(13,J)=FSP(J)
     OV(14,J)=FTR(J)
     OV(10,J)=FTO(J)
 30  CONTINUE
     RETURN
     END
C
C***********************************************************
     SUBROUTINE DFILTER(U1,U2,F,NT)
C***********************************************************
C     THIS SUBROUTINE IS A HIGH-PASS DIGITAL FILTER FOR
C     THE I & Q SIGNALS USED TO SUPPRESS CLUTTER SIGNALS
C***********************************************************
     IMPLICIT INTEGER (I-N)
     DIMENSION U1(*),U2(*),FU1(512),FU2(512)
     C1=1.-F
     FU1(1)=U1(1)
     FU2(1)=U2(1)
     DO 785 I=2,NT
     FU1(I)=C1*FU1(I-1) + U1(I) - U1(I-1)
     FU2(I)=C1*FU2(I-1) + U2(I) - U2(I-1)
 785 CONTINUE
     DO 786 I=1,NT
     U1(I)=FU1(I)
     U2(I)=FU2(I)
 786 CONTINUE

     RETURN
     END
C
C***********************************************************
     SUBROUTINE EFILTER(U1,U2,IFIL,NT)
```

```
C*************************************************************
C     THIS SUBROUTINE IS A 1ST OR 2ND ORDER BUTTERWORTH FILTER FOR
C     THE I & Q SIGNALS USED TO SUPPRESS CLUTTER SIGNALS
C*************************************************************
      IMPLICIT INTEGER (I-N)
      DIMENSION U1(*),U2(*),FU1(512),FU2(512)
      COMMON/BFIL/FC11,FC12,FC21,FC22,FC23
      FU1(1)=U1(1)
      FU2(1)=U2(1)
      FU1(2)=U1(2)
      FU2(2)=U2(2)
      ILO=-2
      IHI=-1
      IF(IFIL .LT. ILO .OR. IFIL .GT. IHI)THEN
        WRITE(*,*)'RETURN FROM EFILTER - NO FILTERING DONE'
        RETURN
      ELSEIF(IFIL .EQ. -2)THEN
        DO 785 I=3,NT
        FU1(I)=FC21*FU1(I-1)+FC22*FU1(I-2)+FC23*
     2         (U1(I)-2.*U1(I-1)+U1(I-2))
        FU2(I)=FC21*FU2(I-1)+FC22*FU2(I-2)+FC23*
     2         (U2(I)-2.*U2(I-1)+U2(I-2))
785     CONTINUE
        WRITE(*,*)'TWO POLE BUTTERWORTH FILTER COMPLETED'
      ELSEIF(IFIL .EQ. -1)THEN
        DO 789 I=2,NT
        FU1(I)=FC11*FU1(I-1)+FC12*(U1(I)-U1(I-1))
        FU2(I)=FC11*FU2(I-1)+FC12*(U2(I)-U2(I-1))
789     CONTINUE
        WRITE(*,*)'SINGLE POLE BUTTERWORTH FILTER COMPLETED'
      ENDIF
      DO 786 I=1,NT
      U1(I)=FU1(I)
      U2(I)=FU2(I)
786   CONTINUE
      RETURN
      END
C
C*************************************************************
      SUBROUTINE FILCO(VMAX,VCO)
C*************************************************************
C     CALCULATES COEFFICIENTS FOR 1ST AND 2ND
C     ORDER DIGITAL BUTTERWORTH FILTERS
C*************************************************************
      IMPLICIT INTEGER (I-N)
      COMMON/BFIL/FC11,FC12,FC21,FC22,FC23
      DATA B1,C1,B2,C2,D2/.293408,.353296,.677496,.253921,.144106/
      PI=3.14159
      R=VCO/VMAX
      ALP= -COS(.5*(1.+PI*R))/COS(.5*(1.-PI*R))
      FC11=-(ALP+B1)/(1.+ALP*B1)
```

```
      FC12=C1*(1.-ALP)/(1.+ALP*B1)
      DEM=1. + ALP*B2 + ALP*ALP*C2
      FC21=-(2.*ALP*(1.+C2) + B2*(1.+ALP*ALP))/DEM
      FC22=-(ALP*ALP + ALP*B2 + C2)/DEM
      FC23=(D2*(1.-ALP)**2)/DEM
C      WRITE(*,4)FC11,FC12,FC21,FC22,FC23
 4    FORMAT(1X,5F15.4)
      RETURN
      END
C
C***********************************************
      SUBROUTINE ADCONV(U1,U2,NP,NSAMP,NLEV,GFAC,SDNSE,GAIN)
C***********************************************
C         THIS SUBROUTINE CALCULATES THE
C         GAIN REQUIRED TO SET THE SIGNALS
C         TO A +/- UNITY LEVEL AND THEN QUANTIZES THE SIGNALS
C         IN ACCORDANCE WITH THE NUMBER OF LEVELS OF THE A/D.
C         GFAC IS THE LEVEL DESIRED FOR AVERAGE SIGNAL
C         THIS ROUTINE ASSUMES AGC IN EACH RANGE BIN
C***********************************************
      IMPLICIT INTEGER (I-N)
      DIMENSION U1(*),U2(*)
C
C ********CALCULATE AVERAGE POWER & REQUIRED GAIN***
C
C      WRITE(*,*)NP,NSAMP,NLEV,GFAC,SDNSE
      GAINM=GFAC/SDNSE
      PAV=0.
      DO 5 I=1,NSAMP
 5    PAV=PAV+(U1(I)*U1(I) + U2(I)*U2(I))
      GAIN=GFAC/SQRT(PAV)
C      WRITE(*,*)GAIN,GAINM,PAV
      IF(GAIN .GE. GAINM)GAIN=GAINM
      GAIN1=GAIN*NLEV
C
C*********QUANTIZE SIGNALS AND RENORMALIZE TO ORIGINAL LEVELS******
C
      DO 10 I=1,NP
      T1=U1(I)
      T2=U2(I)
      XNUM=FLOAT(INT(GAIN1*U1(I)))
      U1(I)=XNUM/GAIN1
      XNUM=FLOAT(INT(GAIN1*U2(I)))
      U2(I)=XNUM/GAIN1
C     WRITE(*,*)T1,U1(I),T2,U2(I)
 10   CONTINUE
      RETURN
      END
C
C***************************************************
      SUBROUTINE ANTFLAT(A,B,ANG,GAIN)
```

```
C***************************************************************
C       CALCULATES GAIN OF FLAT (COLLINS TYPE) ANTENNA
C       BEAMWIDTH IS 3.5 DEGREES
C       A & B ARE NOT USED,ANG IS IN DEGREES,GAIN IN DB
C***************************************************************
        IMPLICIT INTEGER (I-N)
        DATA A1,A2,A3,A4,A5,A6,A7,A8,A9/1.75,7.,12.25,14.,
     2    15.75,28.,40.25,42.,43.75/
        GO=34.5
        G1=9.5
        AA=ABS(ANG)
        GAIN=0.
        IF(AA.LT.A1)THEN
          GAIN=GO-5.*ALOG10(EXP(5.55*(AA/3.5)**2))
        ELSEIF(AA.GE.A1.AND.AA.LT.A2)THEN
          GAIN=GO+11.3-8.175*AA
        ELSEIF(AA.GE.A2.AND.AA.LT.A3)THEN
          GAIN=G1+11.3-8.175*(A4-AA)
        ELSEIF(AA.GE.A3.AND.AA.LT.A5)THEN
          GAIN=G1-5.* ALOG10(EXP(5.55*((A4-AA)/3.5)**2))
        ELSEIF(AA.GE.A5.AND.AA.LT.A6)THEN
          GAIN=G1+11.3-8.175*(AA-A4)
        ELSEIF(AA.GE.A6.AND.AA.LT.A7)THEN
          GAIN=G1+11.3-8.175*(A8-AA)
        ELSEIF(AA.GE.A7.AND.AA.LT.A9)THEN
          GAIN=G1-5.*ALOG10(EXP(5.55*((A8-AA)/3.5)**2))
        ELSEIF(AA.GE.A9)THEN
          GAIN=G1+11.3-8.175*(AA-A8)
 55     ENDIF
        IF(GAIN .LT. 0.)GAIN=0.
        GAIN=10.**(GAIN/10.)
        RETURN
        END
C
C***************************************************************
        SUBROUTINE HORIZ(RFIN,H2,HO)
C***************************************************************
C       CALCULATES ANTENNA POINTING ANGLE WITH RESPECT
C       TO HORIZON IN DEGREES
C***************************************************************
        IMPLICIT INTEGER (I-N)
        DIMENSION RFIN(3,3),R1(3,3),H2(3),H1(3),HO(3)
        DATA H1/1.,0.,0./
        CALL GMTRA(RFIN,R1,3,3)
        CALL GMPRD(R1,H1,H2,3,3,1)
C        WRITE(*,*)H2(1),H2(2),H2(3)
        CALL SPHER(H2,HO)
C        WRITE(*,*)HO(1),HO(2),HO(3)
        RETURN
        END
C
```

```
C ********************************************************************
      SUBROUTINE WXANAL(U1,U2,N,DT,VEL,STHRES,IMAGDB,VSP)
C*********************************************************************
C     CALCULATES VELOCITY USING SPECTRAL AVERAGING
C*********************************************************************
      IMPLICIT INTEGER (I-N)
      DIMENSION U1(*),U2(*),IWK(10)
      DIMENSION RE(512),XIMAG(512),XMAG(512),IMAGDB(*)
      DIMENSION X(512)
      COMPLEX Z(512)
      COMMON/CONST/PI,DTR,RTD,CC,FREQ,XLAM
      XKTTMS=.5148
C
      DO 10 I=1,N
      Z(I)=CMPLX(U1(I),U2(I))
 10   CONTINUE
C
      DELF=1./(N*DT)
      DELV=DELF*CC/(2.*FREQ)
      XN=N
      M=(ALOG10(XN)/ALOG10(2.)) + .5
      CALL FFT(Z,M,IWK)
C
      DO 20 I=1,N
      RE(I)=REAL(Z(I))
      XIMAG(I)=AIMAG(Z(I))
      XMAG(I)=(RE(I)*RE(I) + XIMAG(I)*XIMAG(I))/(XN*XN)
      IMAGDB(I)=INT(10.*ALOG10(XMAG(I)))
 20   CONTINUE
C
      S=0.
      DO 889 I=1,N
      IF(I.LE.N/2)II=I+N/2
      IF(I.GT.N/2)II=I-N/2
      X(II)=XMAG(N-I+1)
 889  CONTINUE
C
      SUM1=0.
      SUM2=0.
      DO 200 I=2,N
      SUM1=SUM1 + X(I)
      VELI=(I-1-N/2)*DELV
      SUM2=SUM2 + VELI*X(I)
 200  CONTINUE
      S=SUM1*N/(N-1)
      VNN2=(N/2)
      VSP=SUM2/SUM1
      IF(S .LT. STHRES)VSP=0.
C     WRITE(*,*)VSP,S,STHRES
      RETURN
      END
```

```
C
C***************************************************************
       SUBROUTINE PPP(U1,U2,N,PRR,VEL,STHRES,VPP,WPP)
C***************************************************************
C      CALCULATES VELOCITY & WIDTH USING PULSE-PAIR
C***************************************************************
       IMPLICIT INTEGER (I-N)
       DIMENSION U1(*),U2(*)
       COMMON/CONST/PI,DTR,RTD,CC,FREQ,XLAM
       SUMT=1.E-50
       XKTTMS=.5148
       PI4=4.*PI
       CONS=XLAM/(2.*PI*PRR*1.414)
       SUMR=0.
       SUMI=0.
       SUMSQ=0.
       M=N-1
       VPP=0.
       WPP=0.
       DO 10 I=1,M
       SUMR=SUMR + U1(I)*U1(I+1)  + U2(I)*U2(I+1)
       SUMI=SUMI + U1(I)*U2(I+1)  - U1(I+1)*U2(I)
       SUMSQ=SUMSQ + U1(I)*U1(I)  + U2(I)*U2(I)
 10    CONTINUE
       IF(SUMSQ .LT. SUMT)SUMSQ=SUMT
C       IF(SUMR .LT. SUMT)SUMR=SUMT
C       IF(SUMI .LT. SUMT)SUMI=SUMT
       VPP=(XLAM/(PI4*PRR))*ATAN2(SUMI,SUMR)
       R1=(1./M)*SQRT(SUMR*SUMR + SUMI*SUMI)
       S=(1./M)*SUMSQ
       IF(S .LT. STHRES)VPP=0.
       WPP=0.
       IF(R1 .EQ. 0.)GO TO 45
       TEST=ALOG(S/R1)
       SGN=-1.
       IF(TEST .GT. 0.)SGN=1.
       WPP=SGN*CONS*SQRT((ABS(TEST)))
C       IF(WPP .GT. 5.)VPP=0.
       IF(S .LT. STHRES)WPP=0.
C       WRITE(*,*)VPP,WPP,R1,S,STHRES
 45    RETURN
       END
C
C*****************************************************
       SUBROUTINE ATTEN(RH,ZO,BINSZ,BX,BY, XLOSS)
C*****************************************************
C      CALCULATES 2-WAY PATH LOSS DUE TO MOISTURE
C*****************************************************
       IMPLICIT INTEGER (I-N)
       DIMENSION RH(3)
       COMMON/ATTN/CON1,CON2,BETA
```

```
      DUM=0.
      TMAX=16.E6
      XLOSS=1.
      X=RH(1)
      Y=RH(2)
      Z=RH(3)
      XMAG=SQRT(X*X+Y*Y+Z*Z)
      SUM=0.
      SO=XMAG
      DO 10 I=1,50
      S=SO-BINSZ*(2.*I - 1.)
C      WRITE(*,6)SO,S
  6   FORMAT(1X,2F10.0)
      IF(S .LE. 1000.)GO TO 35
      CON=S/XMAG
      XX=X*CON
      YY=Y*CON
      ZZ=Z*CON-ZO
      IF(ZZ .LE. 0.)GO TO 10
      TEST=(XX-BX)*(XX-BX) + (YY-BY)*(YY-BY)
      IF(TEST .GE. TMAX)GO TO 10
      CALL BRSTWD3(DUM,DUM,DBZ,BX,BY,XX,YY,ZZ,3)
      IF(DBZ .LT. 1.)GO TO 10
      ZZZ=10**(DBZ/10.)
      RR=(ZZZ/200.)**.625
      XK=CON2*(RR**BETA)
      SUM=SUM - XK*BINSZ*2.
C      WRITE(*,40)I,XX,YY,ZZ,S,DBZ,RR,XK,SUM
 10   CONTINUE
 40   FORMAT(1X,I3,4F7.0,2F5.1,2E10.2)
C*************2-WAY LOSS FACTOR*******************
 35   XLOSS=EXP(2.*SUM)
C      ATTNDB=10.*ALOG10(XLOSS)
C      WRITE(*,*)XLOSS,ATTNDB
      RETURN
      END
C
C ***************************************************
      SUBROUTINE VELTRU(RFIN,VX,VELIN,ROO,RRG,ZO,BX,BY,FVER,VTRU)
C****************************************************
C     CALCULATES VELOCITY COMPONENT ALONG PROJECTED
C     ANTENNA BEAM CENTER
C****************************************************
      IMPLICIT INTEGER (I-N)
      DIMENSION RFIN(*),VX(*),RHOBM2(3),R1(3,3),RHOANT(3)
      XKTTMS=.5148
      DO 10 I=1,3
 10   RHOANT(I)=0.
      VTRU=0.
      RHOANT(1)=RRG*1000.
      CALL GMTRA(RFIN,R1,3,3)
```

```
      CALL GMPRD(R1,RHOANT,RHOBM2,3,3,1)
      XX=RHOBM2(1)
      YY=RHOBM2(2)
      ZZZ=RHOBM2(3)
      RHOMAG=SQRT(XX*XX + YY*YY + ZZZ*ZZZ)
      ZZ=ZZZ-ZO
      IF(ZZ .LE. 0.)go to 20
      CALL BRSTWD3(U,V,W,BX,BY,XX,YY,ZZ,2)
      VXX=VX(1)-U
      VYY=VX(2)-V
      VZZ=VX(3)-W
      FVER=-W/(VELIN*XKTTMS)
      VTRU=((VXX*XX+VYY*YY+VZZ*ZZZ)/RHOMAG)-VELIN
C      WRITE(*,*)XX,YY,ZZ,U,V,W,RHOMAG,VELIN
 20   RETURN
      END
C
C*************************************************
        SUBROUTINE GETBW(ITYPE,A,B,BW)
C*************************************************
C       GETS 3 DB BEAMWIDTH OF ANTENNAS IN DEGREES
C*************************************************
        GO=0.
        BW=0.
        DO 10 I=1,75
        ANG=(I-1)*.1
        IF(ITYPE .EQ. 1)CALL ANTPAT(A,B,ANG,GAIN)
        IF(ITYPE .EQ. 2)CALL ANTFLAT(A,B,ANG,GAIN)
        IF(ITYPE .EQ. 3)CALL ANTREAL(ANG,90.,GAIN)
        GDB=10.*ALOG10(GAIN)
        IF(I .EQ. 1)GO=GDB
        TEST=GO-GDB
        IF(TEST .GT. 3.) GO TO 20
 10     CONTINUE
 20     BW=ANG*2.
        RETURN
        END
C
C*************************************************
      SUBROUTINE READANT
C*********************************************
C     READS ANTENNA PATTERN DATA FROM UNIT 16
C*********************************************
      IMPLICIT INTEGER(I-N)
      DIMENSION ANTDAT(1800,2)
      COMMON/ANTR/ANTDAT
      DO 10 I=1,1800
      READ(16,20)DUM,ANTDAT(I,1),ANTDAT(I,2)
C      WRITE(*,*)I,ANTDAT(I,1),ANTDAT(I,2)
 10   CONTINUE
 20   FORMAT(3F10.2)
```

```
      WRITE(*,*)'ANTENNA DATA READ'
      RETURN
      END

      SUBROUTINE ANTREAL(ANG,THT,GAIN)
C**************************************************
C     CALCULATES ANTENNA GAIN FROM DATA FILE
C**************************************************
      IMPLICIT INTEGER(I-N)
      DIMENSION ANTDAT(1800,2)
      COMMON/CONST/PI,DTR,RTD,CC,FREQ,XLAM
      COMMON/ANTR/ANTDAT
      DATA GAINO/34.6/
      THTR=THT*DTR
      IANG=INT(10.*ANG)
      E=COS(THTR)
      H=SIN(THTR)
      IPHIE=IANG+900
      IPHIH=IPHIE
      IF(E.LE.0.)IPHIE=900-IANG
      IF(H.LE.0.)IPHIH=900-IANG
      IE=1801-IPHIE
      IH=1801-IPHIH
C      write(*,*)ang,tht,ie,ih
      WE=E*E
      WH=H*H
      GE=GAINO+ANTDAT(IE,1)
      GH=GAINO+ANTDAT(IH,2)
      PE=10.**(GE/10.)
      PH=10.**(GH/10.)
      GAIN=PE*WE + PH*WH
      RETURN
      END
C
      SUBROUTINE READRWY(IWHICH,IRL,IMAX,JMAX)
C********************************************************
C     READS RUNWAY SPECIFICATION DATA FOR A GIVEN MAP
C********************************************************
      IMPLICIT INTEGER (I-N)
      DIMENSION IRWDATA(10,6),IRUSE(6),CA(2),RMR(3,3),RHOMO(3)
      DIMENSION RRM(3,3)
      COMMON/CLUT/RMR,RHOMO,SFR,SFA,IRCODE,CA,IRUSE,JMAXO
     2    ,SIGURB,AREAC,RRM
      COMMON/CONST/PI,DTR,RTD,CC,FREQ,XLAM
      READ(18,*)NR,XLOOK,JMAXO,IMAXO,JMAXR,IMAXR,SFRG,SFAZ,IRCODE
      NTIMES=1
      SFR=SFRG
      SFA=SFAZ
      DO 10 I=1,NR
      READ(18,*) (IRWDATA(I,J),J=1,6)
 10   CONTINUE
```

```
      READ(18,*)(CA(J),J=1,2)
      READ(18,*)SIGURB
C
  5   WRITE(*,*)'RUNWAY IS    ',IWHICH,IRL
      IRWNO=99
      DO 20 I=1,NR
      IF(IWHICH .EQ. IRWDATA(I,1).AND.IRL.EQ.IRWDATA(I,2))
     2   IRWNO=I
 20   CONTINUE
      IF(IRWNO .EQ. 99)THEN
         WRITE(*,*)'NO RUNWAY SELECTED'
         GO TO 46
      ENDIF
       DO 30 J=1,6
 30      IRUSE(J)=IRWDATA(IRWNO,J)
      IF(IRCODE .EQ. 2)THEN
        IMAX=IMAXR
        JMAX=JMAXR
      ELSE
        IMAX=IMAXO
        JMAX=JMAXO
      ENDIF
C      write(*,*)NR,XLOOK,JMAXO,IMAXO,JMAXR,IMAXR,SFRG,SFAZ,IRCODE
C      write(*,*)imax,jmax,irwno
C ***********CALCULATIONS8**********************
      XT=(IRUSE(6)-IRUSE(4))*SFAZ
      YT=(IRUSE(3)-IRUSE(5))*SFRG
      YLEN=SQRT(XT*XT+YT*YT)
      THETA=ATAN2(XT,YT)*RTD
      CALL ROT(3,-THETA,RMR)
      RHOMO(1)=IRUSE(4)*SFAZ
      RHOMO(2)=-IRUSE(3)*SFRG
      RHOMO(3)=0.
      CALL GMTRA(RMR,RRM,3,3)
C      WRITE(*,*)RHOMO(1),RHOMO(2),THETA
 46   RETURN
      END
C**********************************************
      SUBROUTINE RDCLUT5(IMAX,JMAX,MAP)
C**********************************************
C     READS CLUTTER MAP
C**********************************************
      IMPLICIT INTEGER (I-N)
      INTEGER*4 K
      INTEGER*2 MAP(*),IN(1000)
      WRITE(*,*)'READING CLUTTER MAP DATA'
      DO 20 J=1,JMAX
      READ(15,57)(IN(I),I=1,IMAX)
      DO 40 II=1,IMAX
      K=(J-1)*IMAX + II
      MAP(K)=IN(II)
```

```
   40     CONTINUE
 C        WRITE(*,8)K
   20     CONTINUE
   8      FORMAT(1X,I10)
  57      FORMAT(10I5)
          WRITE(*,*)'CLUTTER MAP COMPLETED, K = ',K
          RETURN
          END
C*****************************************************
          SUBROUTINE clut5(X,Y,IMAX,JMAX,MAP,ANG, SIGODB)
C*****************************************************
C         GETS VALUE OF SIGO FROM CLUTTER MAP
C           ORIGIN OF COORDINATES AT SPECIFIED RUNWAY
C*****************************************************
          IMPLICIT INTEGER (I-N)
          INTEGER*4 K
          INTEGER*2 MAP(*)
          DIMENSION CO(4),ANG(*),RHOMP(3),RRM(3,3),RHORW(3)
          DIMENSION IRWDATA(10,6),IRUSE(6),RMR(3,3),RHOMO(3),CA(2)
          COMMON/CONST/PI,DTR,RTD,CC,FREQ,XLAM
          COMMON/CLUT/RMR,RHOMO,SFR,SFA,IRCODE,CA,IRUSE,JMAXO
     2      ,SIGURB,AREAC,RRM
          DATA CO/.331,-4.339E-2,9.416E-4,-6.180E-6/
          SFRG=SFR
          SFAZ=SFA
          RHORW(1)=X
          RHORW(2)=Y
          RHORW(3)=0.
          CALL GMPRD(RRM,RHORW,RHOMP,3,3,1)
          II=INT(RHOMP(1)/SFA) + IRUSE(4)
          JJ=INT(-RHOMP(2)/SFR) + IRUSE(3)
 C         WRITE(*,98)II,JJ,II*SFA,JJ*SFR
  98      FORMAT(1X,2I5,2F10.1)
          I=INT((II+1)/IRCODE + .5)
          J=INT((JJ+1)/IRCODE + .5)
          JJJ=IRCODE*J
          IF(JJJ.GE.JMAXO)JJJ=JMAXO
          IF(JJJ.LT.1)JJJ=1
          IF(I.GT.IMAX)I=I-IMAX
          IF(I.LT.1)I=IMAX+I
C          IF(J.GT.JMAX)J=J-JMAX
          IF(J.LT.1)J=JMAX+J
          IF(I.GT.IMAX)I=IMAX
          IF(I.LT.1)I=1
          IF(J.GT.JMAX)J=JMAX
          IF(J.LT.1)J=1
          K=(J-1)*IMAX + I
C
C  ***********INCIDENCE ANGLE CORRECTION TO SIGO*****************
C
          A=ATAN((CA(1) + SFRG*(JMAXO-JJJ))/CA(2))*RTD
```

```
      CORR=CO(4)*(ANG(4)-A**4)+CO(3)*(ANG(3)-A**3)+CO(2)*(ANG(2)
     2   -A**2)+CO(1)*(ANG(1)-A)
C
C******************AVERAGE MAP CELLS IF REQUIRED***************
C
      TEST=SFAZ*SFRG*IRCODE*IRCODE
      ITST=INT(AREAC/TEST)
           IS=0
           IE=0
           JS=0
           JE=0
           NA=1
      IF(ITST.GE.4)THEN
           IS=-1
           IE=1
           JS=-1
           JE=1
           NA=9
      END IF
      AVR=0.
      DO IX=I+IS,I+IE
      DO JX=J+JS,J+JE
      K=(JX-1)*IMAX + IX
      VAL=MAP(K)*1.0
      PWR=10.**(VAL/100.)
      AVR=AVR + PWR/NA
      END DO
      END DO
      VAL=10.*ALOG10(AVR)
      XFAC=1.
C
C   **********DO NOT CORRECT URBAN AREA OR DISCRETE SCATTERERS****
C
      TEST=SIGURB
      IF(VAL .GE. TEST)XFAC=0.
      SIGODB=VAL + CORR*XFAC
      IF(SIGODB .LT. -60.)SIGODB=-60.
c      WRITE(*,100)X,Y,ANG(1),A,VAL,CORR,SIGODB,I,J,K,II,JJ,JJJ
 100  FORMAT(1X,2F7.1,5F9.2,6I7)
      RETURN
      END
```

```
      SUBROUTINE STRNUM(YY)
      BB=1.
      P1=YY*317.
      YY = AMOD(P1,BB)
      RETURN
      END
C
      SUBROUTINE BRSTWD3(U,V,W,BX,BY,AX,AY,AZ,N)
C
C-----------------------------------------------------------------------
C
C   PURPOSE:  CALCULATE THE THREE CARTESIAN WIND COMPONENTS OF A TIME-
C             FROZEN SYMMETRIC MICROBURST BY SPECIFYING THE LOCATION
C             OF AN AIRPLANE RELATIVE TO USER'S RUNWAY COORDINATE
C             SYSTEM.  THIS SYSTEM IS ASSUMED TO BE A RIGHT-HANDED
C             ONE WITH Z-AXIS POINTING AWAY FROM THE GROUND.
C             OTHERWISE, THE CALCULATED WIND COMPONENTS NEED TO BE
C             MODIFIED ACCORDINGLY.
C
C    OUTPUT:  U,V, AND W (IN METERS PER SECOND) ARE THE TWO HORIZONTAL
C             AND THE VERTICAL (DEFINED POSITIVE UPWARD) WIND COM-
C             PONENTS IN USER'S RUNWAY COORDINATE SYSTEM.
C
C     INPUT:  (1) BX AND BY (IN METERS) ARE THE HORIZONTAL SURFACE
C                 COORDINATES FOR THE LOCATION OF THE MICROBURST CENTER
C                 RELATIVE TO THE RUNWAY COORDINATES.
C             (2) AX, AY, AND AZ (IN METERS) ARE THE COORDINATES FOR THE
C                 LOCATION OF THE AIRPLANE RELATIVE TO THE RUNWAY.
C             (3) N IS A COUNTER THAT WHEN N=1 WIND DATA ARE READ INTO T
C                 PROGRAM.  THE DATA MUST BE READ AT THE FIRST CALL OF T
C                 SUBROUTINE AND ONLY AT THE FIRST TIME DURING EACH RUN.
C                 DATA FILE SHOULD BE ON TAPE5 SPECIFIED IN THE MAIN
C                 PROGRAM.
C
C             IF N=1 THE DATA ARE READ INTO ARRAYS. IF N=2 THE WIND
C             COMPONENTS ARE CALCULATED.  IF N=3 THE REFLECTIVITY IS
C             CALCULATED AND APPEARS AS W IN THE SUBROUTINE CALL.
C
C SUBROUTINE
C    CALLED:  PRNTOUT
C
C-----------------------------------------------------------------------
C
      COMMON/WIND/ UC(101,101),WC(101,101),IRUN,TIME,ISTOP,JSTOP,DR,DZ
     2     ,ZZ(101,101)
C      DIMENSION DUM(101,101)
C      EQUIVALENCE (DUM(1,1),WC(1,1))
C      DATA END /'END '/
      CHARACTER*4 HD
C
```

```
C              UC AND WC ARE WIND COMPONENTS IN THE CYLINDRICAL
C              COORDINATE SYSTEM.
C
      FLAG=0
      IF (N.NE.1) GO TO 10
C
C   READ IN THE WIND DATA OF A SELECTED TIME SLICE OF A MICROBURST
C     AND PRINT OUT INFORMATION CHARACTERIZING THE TIME-FROZEN
C     WIND FIELD.
C
C       READ(9,1)IRUN,TIME,ISTOP,JSTOP,IT,DR,DZ
      READ(9,1)IRUN,TIME,ISTOP,JSTOP,IT,DR,DZ
C      WRITE(*,1)IRUN,TIME,ISTOP,JSTOP,IT,DR,DZ
      WRITE(*,*)'READING WINDFIELD DATA '
      DO 50 J=1,JSTOP
      DO 60 I=1,ISTOP-1,5
   60 READ(9,3)(UC(I+K-1,J),K=1,5)
      READ(9,33)UC(ISTOP,J)
   50 CONTINUE
C     DO 70 J=1,JSTOP
C     DO 80 I=1,ISTOP-1,5
C 80  WRITE(*,3)(UC(I+K-1,J),K=1,5)
C     WRITE(*,33)UC(ISTOP,J)
C 70  CONTINUE
C      READ(10,3)(WC(I+K-1,J),K=1,5)
      DO 99 J=1,JSTOP
      DO 109 I=1,ISTOP-1,5
  109 READ(10,3)(WC(I+K-1,J),K=1,5)
      READ(10,33)WC(ISTOP,J)
   99 CONTINUE
      WRITE(*,*)'READING REFLECTIVITY DATA '
C      READ(11,3)(ZZ(I+K-1,J),K=1,5)
      DO 90 J=1,JSTOP
      DO 100 I=1,ISTOP-1,5
  100 READ(11,3)(ZZ(I+K-1,J),K=1,5)
      READ(11,33)ZZ(ISTOP,J)
   90 CONTINUE
C      DO 110 J=1,JSTOP
C      DO 120 I=1,ISTOP-1,5
C 120  WRITE(*,3)(ZZ(I+K-1,J),K=1,5)
C      WRITE(*,33)ZZ(ISTOP,J)
C 110  CONTINUE
    3 FORMAT(1X,5E15.7)
   33 FORMAT(1X,E15.7)
    1 FORMAT(1X,I5,F10.2,5X,3I5,2F10.4)
C     CALL PRNTOUT
      RETURN
   10 CONTINUE
C
C   CALCULATE U,V, AND W
C
```

```
      R1=AX-BX
      R2=AY-BY
      R=SQRT(R1*R1+R2*R2)
      E1=0.
      E2=0.
      IF(R.NE.0.) E1=R1/R
      IF(R.NE.0.) E2=R2/R
      I=INT(R/DR)+1
      J=INT(AZ/DZ)+1
      IF(((I+1).GT.ISTOP).OR.((J+1).GT.JSTOP)) GO TO 11
      RX=R-INT(R/DR)*DR
      ZX=AZ-INT(AZ/DZ)*DZ
      WCI1=WC(I,J)
      WCI2=WC(I+1,J)
      WCI3=WC(I+1,J+1)
      WCI4=WC(I,J+1)
      IF(N .NE. 3)GO TO 101
      WCI1=10.**(ZZ(I,J)/10.)
      WCI2=10.**(ZZ(I+1,J)/10.)
      WCI3=10.**(ZZ(I+1,J+1)/10.)
      WCI4=10.**(ZZ(I,J+1)/10.)
 101  UT=(UC(I,J)*(DR-RX)*(DZ-ZX)+UC(I+1,J)*RX*(DZ-ZX)+
     #     UC(I+1,J+1)*RX*ZX+UC(I,J+1)*(DR-RX)*ZX)/(DR*DZ)
      W=(WCI1*(DR-RX)*(DZ-ZX)+WCI2*RX*(DZ-ZX)+
     #     WCI3*RX*ZX+WCI4*(DR-RX)*ZX)/(DR*DZ)
      U=E1*UT
      V=E2*UT
      IF(N .EQ. 3 .AND. W .LT. 1.E-5)W=1.E-5
      IF(N .EQ. 3)W=10.*ALOG10(W)
C     WRITE(66,67)U,V,W
 67     FORMAT(1X,3E15.7)
 72     CONTINUE
      GO TO 45
   11 U=0.
      V=0.
      W=0.
      IF(N .EQ. 3) W=-50.
 45   CONTINUE
 40   CONTINUE
      GO TO 990
C
 8    FORMAT(A4)
 9    FORMAT(5E15.7)
 999  CONTINUE
C       PRINT*,'ERROR'
 990  CONTINUE
      RETURN
      END
C
       SUBROUTINE ANTPAT(A,B,ANG,GAIN)
C       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
C
C      CALCULATES ANTENNA GAIN AT ANGLE ANG
C      ANG IS IN DEGREES - P=1
C      REQUIRES SCI.LIB FOR BESSEL SUBROUTINES
C
       COMMON/CONST/PI,DTR,RTD,CC,FREQ,XLAM
       COMMON/ANT/C1,C2,XK,C12,GAINO
       U=ABS(XK*A*SIN(ANG*DTR))
       CON=4./(1.+B)
       GAIN=GAINO
       IF(U .LT. 1.E-12) U=1.E-12
       CALL BESSN(U,1.,XJ1)
       CALL BESSN(U,2.,XJ2)
c       XJ1=besj1(U)
c       XJ2=besj2(U)
       FX=(B*XJ1/U  + (1.-B)*2.*XJ2/(U*U))*CON
       GAIN=FX*FX*GAINO
       RETURN
       END
C
       SUBROUTINE NORMAL(XNRN)
       DATA RN/.237/
       SUM=0.
       DO 20 I=1,12
       CALL STRNUM(RN)
       RNZM=RN-.5
   20  SUM=SUM+RNZM
       XNRN=SUM
       RETURN
       END
C
C
      SUBROUTINE NORNG(R,P)
C
C     THIS SUBROUTINE GENERATES A SEQUENCE OF NUMBERS NORMALLY
C     AND RANDOMLY DISTRIBUTED OVER THE INTERVAL -3 TO 3 FROM
C     UNIFORMLY DISTRIBUTED RANDOM NUMBERS BY THE METHOD OF LINEAR
C     APPROXIMATION TO THE INVERSE OF THE ACCUMULATIVE
C     NORMAL DISTRIBUTION FUNCTION.
C
      DIMENSION Y(6), X(6), S(5)
      DATA Y/0.,.0228,.0668,.1357,.2743,.5/,
     &X/-3.01,-2.0,-1.5,-1.0,-.6,0./,
     &S/43.8596,11.3636,7.25689,2.891352,2.65887/
      CALL STRNUM(R)
      P = R
      I = 1
      IF (P.GT.0.5) P = 1.0-R
2     IF (P.LT.Y(I + 1)) GO TO 8
      I = I + 1
      GO TO 2
```

```
8     P = ((P-Y(I))*S(I) + X(I))
      IF (R.GE.0.5) P = -P
      RETURN
      END
C
C
      SUBROUTINE CROSS(A,B,R)
C     TAKES THE CROSS PRODUCT A X B = R
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(3),B(3),R(3)
      R(1)=A(2)*B(3)-A(3)*B(2)
      R(2)=A(3)*B(1)-A(1)*B(3)
      R(3)=A(1)*B(2)-A(2)*B(1)
      RETURN
      END
C
      SUBROUTINE MEG(PHI,OM,R)
C     CREATES MATRIX FOR CONVERSION FROM EARTH CENTERED. TO
C     GEOGRAPHIC COORD.
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION R(3,3)
      DTR=2.*3.141592654/360.
      X=PHI*DTR
      Y=OM*DTR
      SX=SIN(X)
      CX=COS(X)
      SY=SIN(Y)
      CY=COS(Y)
      R(1,1)=-SX*CY
      R(2,1)=-SY
      R(3,1)=-CX*CY
      R(1,2)=-SX*SY
      R(2,2)=CY
      R(3,2)=-CX*SY
      R(1,3)=CX
      R(2,3)=0.
      R(3,3)=-SX
      RETURN
      END
C
      SUBROUTINE SPHER(A,R)
C     RECTANGULAR TO SPHERICAL CONVERSION OF VECTOR
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(3),R(3)
      X=A(1)
      Y=A(2)
      Z=A(3)
      RTD=360./(2.*3.141592654)
      AM=SQRT(X*X+Y*Y+Z*Z)
      CONE=(ACOS(Z/AM))*RTD
      CLOCK=(ATAN(Y/X))*RTD
```

```
      IF(X) 5,5,6
5     IF(Y) 7,7,8
6     IF(Y) 10,11,11
7     CLOCK=180.+CLOCK
      GO TO 12
8     CLOCK=180.+CLOCK
      GO TO 12
10    CLOCK=360.+CLOCK
      GO TO 12
11    CLOCK=CLOCK
12    CONTINUE
      R(1)=AM
      R(2)=CLOCK
      R(3)=CONE
      RETURN
      END
C
      SUBROUTINE GMPRD(A,B,R,N,M,L)
C     THIS SUBROUTINE MULTIPLIES TWO MATRICES A X B = R
C     N=ROWS OF A,M=COLUMNS OF A,ROWS OF B. L=COL. OF B
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(*),B(*),R(*)
      IR=0
      IK=-M
      DO 10 K=1,L
      IK=IK+M
      DO 10 J=1,N
      IR=IR+1
      JI=J-N
      IB=IK
      R(IR)=0.
      DO 10 I=1,M
      JI=JI+N
      IB=IB+1
10    R(IR)=R(IR)+A(JI)*B(IB)
      RETURN
      END
C
      SUBROUTINE GMTRA(A,R,N,M)
C     TRANSPOSES MATRIX A TO GIVE R. N=ROWS,M=COLS.
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(*),R(*)
      IR=0
      DO 10 I=1,N
      IJ=I-N
      DO 10 J=1,M
      IJ=IJ+N
      IR=IR+1
10    R(IR)=A(IJ)
      RETURN
      END
```

```
C
      SUBROUTINE ROT(IROT,ANG,R)
C     CREATES ROTATION MATRIX. IROT SPECIFIES AXIS OF ROTATION
C     1=X AXIS, 2=Y AXIS, 3=Z AXIS
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION R(3,3)
      X=ANG*(2.*3.141592654)/360.
      C=COS(X)
      S=SIN(X)
      DO 10 I=1,3
      DO 20 J=1,3
      R(I,J)=0.
 20   CONTINUE
 10   CONTINUE
      IF (IROT .NE. 1) GO TO 200
      R(2,2)=C
      R(3,2)=-S
      R(2,3)=S
      R(3,3)=C
      R(1,1)=1.
      RETURN
 200  IF (IROT .NE. 2) GO TO 300
      R(1,1)=C
      R(3,1)=S
      R(1,3)=-S
      R(3,3)=C
      R(2,2)=1.
      RETURN
 300  R(1,1)=C
      R(2,1)=-S
      R(1,2)=S
      R(2,2)=C
      R(3,3)=1.
      RETURN
      END
C
      SUBROUTINE EMAT(P,T,PS,R)
C
C     GENERATES EULER MATRIX. P=PHI,T=THETA,PS=PSI
C
      DIMENSION R(3,3)
      DTR=3.141592654/180.
      X=P*DTR
      Y=T*DTR
      Z=PS*DTR
      CP=COS(X)
      SP=SIN(X)
      CT=COS(Y)
      ST=SIN(Y)
      CPS=COS(Z)
      SPS=SIN(Z)
```

```
          R(1,1)=CT*CPS
          R(2,1)=-CP*SPS + SP*ST*CPS
          R(3,1)=SP*SPS + CP*ST*CPS
          R(1,2)=CT*SPS
          R(2,2)=CP*CPS + SP*ST*SPS
          R(3,2)=-SP*CPS + CP*ST*SPS
          R(1,3)=-ST
          R(2,3)=SP*CT
          R(3,3)=CP*CT
          RETURN
          END
      SUBROUTINE BESSN(X1,B1,Y1)
C
C     SUBROUTINE TO FIND THE BESSEL FUNCTION OF INTEGER OR
C     FRACTIONAL ORDER FOR ANY X GREATER THAN ZERO.
C
      DIMENSION C(6), D(6), S(26)
      DATA S/-3.375,54.4921875,-272.953125,354.375,-2.0625,13.7578125,
     &-19.6875,-1.0625,1.875,-.375,.0030381944,-.4861111111,10.286458
     &33333,-58.,78.75,.0005580357,-.4241071428,3.6026785714,
     &-5.625,.0125,-.35,.75,.0416666666,.25,3.1415926535,1.57
     &07963267/
      Y1 = 0.0
      IF(B1.LT.0.0) RETURN
    2 IF(X1.GE.14.9) GO TO 51
      C1 = B1 + 1.0
      CALL GAMMA(C1,Z1)
      TERM1 = (X1*0.5)** B1/Z1
      P1 = 1.0
      Y1 = TERM1
   10 TERM1 = -TERM1*X1*X1/(4.0*(B1 + P1)*P1)
      Y1 = Y1 + TERM1
      P1 = P1 + 1.0
      IF(ABS(Y1)*1.E-11.LT.ABS(TERM1)) GO TO 10
      RETURN
C 51   IF(B1.GT.1.0) GO TO 53
   51 CONTINUE
      IFLAG = 1
      X = X1
      B = B1
      Y = Y1
      GO TO 1
   15 RETURN
   53 IB1 = B1
      A1 = B1 - FLOAT(IB1)
      IFLAG = 2
      X = X1
      B = A1
      XX1 = 0.
      Y = XX1
      GO TO 1
```

```
16    P1 = A1 - 1.0
      IFLAG = 3
      X = X1
      B = P1
      YY1 = 0
      Y = YY1
      GO TO 1
60    Y1 = 2.0*A1**X1/X1 - YY1
      A1 = A1 + 1.0
      IF(A1.EQ.B1)RETURN
      YY1 = XX1
      XX1 = Y1
      GO TO 60
1     CONTINUE
      A = B*B - 0.25
      T = 1.0/X
      T2 = T*T
      C(1) = (((S(1)*A+S(2))*A+S(3))*A+S(4))*A
      C(2) = ((S(5)*A+S(6))*A+S(7))*A
      C(3) = (S(8)*A+S(9))*A
      C(4) = S(10)*A
      BBO = (((C(1)*T2+C(2))*T2+C(3))*T2+C(4))*T2*T2+1.0
      BB = BBO*SQRT(2.*T/(S(25)*SQRT(1.-A*T2)))
      D(1) = ((((S(11)*A+S(12))*A+S(13))*A+S(14))*A+S(15))*A
      D(2) = (((S(16)*A+S(17))*A+S(18))*A+S(19))*A
      D(3) = ((S(20)*A+S(21))*A + S(22))*A
      D(4) = (S(23)*A + S(24))*A
      D(5) = .5*A
      AA = ((((D(1)*T2 + D(2))*T2+D(3))*T2+D(4))*T2 +D(5))*T2+1.0
      AA = X-(B + .5)*S(26)
      BB= SQRT(2./(S(25)*X))
      Y1 = BB*COS(AA)
      GO TO (15,16,60),IFLAG
      END
      SUBROUTINE GAMMA(X,GAM)
C
C     SUBROUTINE TO FIND THE GAMMA FUNCTION FOR
C     ANY VARIABLE X GREATER THAN ZERO.
C
      DIMENSION B(9)
      DATAB/.035868343,-.193527818,.482199394,-.756704078,
     &.918206857,-.897056937,.988205891,-.577191652,1.00/
      N = 0
      GAM1 = X
      X1 = X
      IF(X.LE.0.0) GO TO 5
      IF(X.LE.1.0) GO TO 4
      IF(X.LE.2.0) N = 1
1     X1 = X1 - 1.0
      IF(X1.LT.2.0) GO TO 2
      GAM1 = GAM1*X1
```

```
      GO TO 1
2     IF(X1.GT.1.0) X1 = X1 - 1.00
3     ND = 8
      CALL PLYEVL (B,ND,X1,GAM)
      IF(N.EQ.1) RETURN
      GAM = GAM*GAM1
      RETURN
4     GAM1 = 1.0/X1
      GO TO 3
5     GAM = 9999999.9
      RETURN
      END
        SUBROUTINE PLYEVL(A,ND,X,ANS)
C
C       THIS SUBROUTINE EVALUATES ANY SINGLE VARIABLE POLYNOMIAL
C       OF THE FORM A(1)*X**ND+A(2)*X**(ND-1)+... +A(ND)*X+A(ND+1)
C
        DIMENSION A(1)
        ANS=A(1)
        IF (ND.EQ.0) RETURN
        DO 20 I=1,ND
   20   ANS=ANS*X+A(I+1)
        RETURN
        END


C************************************************************************
      SUBROUTINE FFT (Z,M,IWK)
C     E2.4
*************************************************************************
C
C
C   FUNCTION                - COMPUTE THE FAST FOURIER TRANSFORM, GIVEN A
C                               COMPLEX VECTOR OF LENGTH EQUAL TO A POWER
C                               OF TWO
C   USAGE                   - CALL FFT (Z,M,IWK)
C   PARAMETERS  Z           - COMPLEX VECTOR OF LENGTH N=2**M
C                               WHICH CONTAINS ON INPUT THE
C                               DATA TO BE TRANSFORMED. ON
C                               OUTPUT,A CONTAINS THE FOURIER
C                               COEFFICIENTS.
C               M           - N = 2**M IS THE NUMBER OF DATA POINTS.
C                             M= +N FFT WILL PERFORM FOURIER
C                                 TRANSFORM.
C                             M= -N FFT WILL PERFORM INVERSE
C                                 TRANSFORM.
C               IWK         - WORK AREA VECTOR OF LENGTH M+1.
C   PRECISION               - SINGLE
C   LANGUAGE                - FORTRAN
C   LATEST REVISION         - APRIL 16, 1980
```

```
C-----------------------------------------------------------------------
C
      DIMENSION IWK(*),Z(*),Z0(2),Z1(2),Z2(2),Z3(2)
      COMPLEX Z,ZA0,ZA1,ZA2,ZA3,AK2
      EQUIVALENCE          (ZA0,Z0(1)),(ZA1,Z1(1)),(ZA2,Z2(1)),(ZA3,Z3(1))
     *,                                (A0,Z0(1)),(B0,Z0(2)),(A1,Z1(1)),(B
     *1,Z1(2)),                        (A2,Z2(1)),(B2,Z2(2)),(A3,Z3(1)),(B
     *3,Z3(2))
      DATA                SQ,SK,CK /.70710678118655,.38268343236509,
     *                              .92387953251129/
      DATA                TWOPI/6.2831853071796/,ZERO/0.0/,ONE/1.0/
C                                   SQ=SQRT2/2,SK=SIN(PI/8),CK=COS(PI/8)
C                                   TWOPI=2*PI
      SIGN = 1.0
      IF (M .LT. 0) SIGN = -1.0
      M= IABS(M)
      MP = M+1
      N = 2**M
      IF (SIGN .EQ. 1.0) GO TO 4
      DO 2 I=1,N
         Z(I) = CONJG(Z(I))
    2 CONTINUE
    4 IWK(1)=1
      MM = (M/2)*2
      KN = N+1
C                                  INITIALIZE WORK VECTOR
      DO 5 I=2,MP
         IWK(I) = IWK(I-1)+IWK(I-1)
    5 CONTINUE
      RAD=TWOPI/N
      MK = M - 4
      KB = 1
      IF (MM .EQ. M) GO TO 15
      K2 = KN
      K0 = IWK(MM+1) + KB
   10 K2 = K2 - 1
      K0 = K0 - 1
      AK2 = Z(K2)
      Z(K2) = Z(K0)- AK2
      Z(K0) = Z(K0)+ AK2
      IF (K0 .GT. KB) GO TO 10
   15 C1 = ONE
      S1 = ZERO
      JJ = 0
      K = MM - 1
      J = 4
      IF (K .GE. 1) GO TO 30
      GO TO 9005
   20 IF (IWK(J) .GT. JJ) GO TO 25
      JJ = JJ - IWK(J)
      J = J-1
```

```
      IF (IWK(J) .GT. JJ) GO TO 25
      JJ = JJ - IWK(J)
      J = J - 1
      K = K + 2
      GO TO 20
   25 JJ = IWK(J) + JJ
      J = 4
   30 ISP = IWK(K)
      IF (JJ .EQ. 0) GO TO 40
C                                    RESET TRIGONOMETRIC PARAMETERS
      C2 = JJ * ISP * RAD
      C1 = COS(C2)
      S1 = SIN(C2)
   35 C2 = C1 * C1 - S1 * S1
      S2 = C1 * (S1 + S1)
      C3 = C2 * C1 - S2 * S1
      S3 = C2 * S1 + S2 * C1
   40 JSP = ISP + KB
C                                    DETERMINE FOURIER COEFFICIENTS
C                                      IN GROUPS OF 4
      DO 50 I=1,ISP
         K0 = JSP - I
         K1 = K0 + ISP
         K2 = K1 + ISP
         K3 = K2 + ISP
         ZA0 = Z(K0)
         ZA1 = Z(K1)
         ZA2 = Z(K2)
         ZA3 = Z(K3)
         IF (S1 .EQ. ZERO) GO TO 45
         TEMP = A1
         A1 = A1 * C1 - B1 * S1
         B1 = TEMP * S1 + B1 * C1
         TEMP = A2
         A2 = A2 * C2 - B2 * S2
         B2 = TEMP * S2 + B2 * C2
         TEMP = A3
         A3 = A3 * C3 - B3 * S3
         B3 = TEMP * S3 + B3 * C3
   45    TEMP = A0 + A2
         A2 = A0 - A2
         A0 = TEMP
         TEMP = A1 + A3
         A3 = A1 - A3
         A1 = TEMP
         TEMP = B0 + B2
         B2 = B0 - B2
         B0 = TEMP
         TEMP = B1 + B3
         B3 = B1 - B3
         B1 = TEMP
```

```
          Z(K0) = CMPLX(A0+A1,B0+B1)
          Z(K1) = CMPLX(A0-A1,B0-B1)
          Z(K2) = CMPLX(A2-B3,B2+A3)
          Z(K3) = CMPLX(A2+B3,B2-A3)
   50 CONTINUE
      IF (K .LE. 1) GO TO 55
      K = K - 2
      GO TO 30
   55 KB = K3 + ISP
C                                     CHECK FOR COMPLETION OF FINAL
C                                       ITERATION
      IF (KN .LE. KB) GO TO 9005
      IF (J .NE. 1) GO TO 60
      K = 3
      J = MK
      GO TO 20
   60 J = J - 1
      C2 = C1
      IF (J .NE. 2) GO TO 65
      C1 = C1 * CK + S1 * SK
      S1 = S1 * CK - C2 * SK
      GO TO 35
   65 C1 = (C1 - S1) * SQ
      S1 = (C2 + S1) * SQ
      GO TO 35
 9005 CONTINUE
      IF (SIGN .EQ. 1.0) GO TO 75
      XN = N
      DO 70 I=1,N
      Z(I)=CONJG(Z(I))/XN
   70 CONTINUE
   75 CALL FFRDR2(Z,M,IWK)
      IF(SIGN.EQ.-1.0) M = -M
      RETURN
      END
      SUBROUTINE FFRDR2 (Z,M,IWK)
C-FFRDR2--------S-------LIBRARY 3----------------------------------------
C
************************************************************************
C
C   FUNCTION            - THIS SUBROUTINE PERMUTES A COMPLEX DATA VECTOR
C                           IN REVERSE BINARY ORDER TO NORMAL ORDER. THE
C                           ROUTINE CAN ALSO BE USED TO PERMUTE A COM-
C                           PLEX DATA VECTOR IN NORMAL ORDER TO REVERSE
C                           BINARY ORDER SINCE THE PERMUTATION IS SYM-
C                           METRIC.
C   USAGE               - CALL FFRDR2(Z,M,IWK)
C   PARAMETERS   Z      - COMPLEX VECTOR OF LENGTH N=2**M WHICH
C                           CONTAINS ON INPUT THE DATA TO BE
C                           PERMUTED. ON OUTPUT, Z CONTAINS THE
C                           PERMUTED DATA VECTOR.
```

```
C                    M        - N=2**M IS THE NUMBER OF DATA POINTS.
C                    IWK      - WORK AREA VECTOR OF LENGTH M+1
C   PRECISION                 - SINGLE
C   LANGUAGE                  - FORTRAN
C   LATEST REVISION           - MARCH 16, 1973
C     ------------------------------------------------------------------
C
      DIMENSION IWK(*),Z(*)
      COMPLEX Z,TEMP
C
      IF(M .LE. 1) GO TO 9005
      MP = M+1
      JJ = 1
C                                        INITIALIZE WORK VECTOR
      IWK(1) = 1
      DO 5  I = 2,MP
         IWK(I) = IWK(I-1) * 2
    5 CONTINUE
      N4 = IWK(MP-2)
      IF (M .GT. 2) N8 = IWK(MP-3)
      N2 = IWK(MP-1)
      LM = N2
      NN = IWK(MP)+1
      MP = MP-4
C                                        DETERMINE INDICES AND SWITCH A*S
      J = 2
   10 JK = JJ + N2
      TEMP= Z(J)
      Z(J)=Z(JK)
      Z(JK)= TEMP
      J = J+1
      IF (JJ .GT. N4) GO TO 15
      JJ = JJ + N4
      GO TO 35
   15 JJ = JJ - N4
      IF (JJ .GT. N8) GO TO 20
      JJ = JJ + N8
      GO TO 35
   20 JJ = JJ - N8
      K = MP
   25 IF (IWK(K) .GE. JJ) GO TO 30
      JJ = JJ - IWK(K)
      K = K - 1
      GO TO 25
   30 JJ = IWK(K) + JJ
   35 IF (JJ .LE. J) GO TO 40
      K = NN - J
      JK = NN - JJ
      TEMP= Z(J)
      Z(J) = Z(JJ)
      Z(JJ) = TEMP
```

```
      TEMP = Z(K)
      Z(K)  = Z(JK)
      Z(JK)= TEMP
   40 J = J + 1
C                                        CYCLE REPEATED UNTIL LIMITING NUMBER
C                                          OF CHANGES IS ACHIEVED
      IF (J .LE. LM) GO TO 10
 9005 RETURN
C**  THIS PROGRAM VALID ON FTN4 AND FTN5 **
      END
```

8.0 REFERENCES

[1] Doviak, R. J.; and Zrnic, D. S.: "Doppler Radar and Weather Observations." Academic Press Inc. 1984.

[2] Proctor, F. H.: "The Terminal Area Simulation System Vol. I: Theoretical Formulation." NASA CR 4046: DOT/FAA/FM-86/50, I. April 1987.

[3] Proctor, F. H.: "The Terminal Area Simulation System Vol. II: Verification Cases." NASA CR 4047: DOT/FAA/PM-86/50, II. April 1987.

[4] Ruck, G.; Barrick, D.; Stuart, W.; Krichbaum, C.: "Radar Cross Section Handbook Vol. II." New York: Plenum, 1970.

[5] Bracalente, E. M.; Britt, C. L.; Jones, W. R.; "Airborne Doppler Radar Detection of Low-Altitude Wind Shear." Journal of Aircraft, Vol. 27, No. 2; February, 1990, pp. 151-157.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA CR-182025 DOT/FAA/DS-90/7	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Users Guide for an Airborne Windshear Doppler Radar Simulation (AWDRS) Program		5. Report Date June 1990
		6. Performing Organization Code
7. Author(s) Charles L. Britt		8. Performing Organization Report No. RTI/3042/18-01F
		10. Work Unit No. 505-66-11-54
9. Performing Organization Name and Address Research Triangle Institute P. O. Box 12194 Research Triangle Park, NC 27709		11. Contract or Grant No. NAS1-17639
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225		13. Type of Report and Period Covered Contractor Report (Final Report)
		14. Sponsoring Agency Code

15. Supplementary Notes

Langley Contract Monitor: Lyle C. Schroeder
Final Report Task 18
Langley Technical Task Monitor: Emedio M. Bracalente

16. Abstract

This report provides a description of the Airborne Windshear Doppler Radar Simulation (AWDRS) program developed for NASA-Langley Research Center by the Research Triangle Institute. The radar simulation program is a comprehensive calculation of the signal characteristics and expected outputs of an airborne coherent pulsed Doppler radar system viewing a low-level microburst along or near the approach path of the aircraft.

The detailed nature of the simulation permits the quick evaluation of proposed trade-offs in radar system parameters and the evaluation of the performance of proposed configurations in various microburst/clutter environments. The simulation also provides a test bed for various proposed signal processing techniques for minimizing the effects of noise, phase jitter, and ground clutter and maximizing the useful information derived for avoidance of microburst windshear by aircraft.

17. Key Words (Suggested by Author(s)) Weather Radar, Airborne, Doppler, Windshear, Microburst		18. Distribution Statement Unclassified - Unlimited; Subject Category 03	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 92	22. Price A05

NASA FORM 1626 OCT 86