DTIC FILE COPY

(2)

AD-A229 806

TECHNICAL REPORT CR-RD-AS-90-3

EXTENSION AND UPDATING OF THE COMPUTER SIMULATION OF RANGE RELATIVE DOPPLER PROCESSING AND INVARIANT MAPPING FOR MM WAVE SEEKERS

VOLUME III

SIMULATION OF A DOPPLER BEAM SHAPENING RADAR FOR MM WAVE SEEKERS

Robert J. Polge
Bassem R. Mahafza
Jong G. Kim
College of Engineering
University of Alabama in Huntsville
Huntsville, AL 35899

DTIC
ELECTE
DEC 27 1990
S B D

OCTOBER 1990

Prepared for:
Advanced Sensors Directorate
Research, Development, and Engineering Center

Contract No. DAAH01-87-D-0021

U.S. ARMY MISSILE COMMAND
Redstone Arsenal, Alabama 35898-5000

Approved for public release; distribution is unlimited.

90 12 26 005

SMI FORM 1021, 1 AUG 85 PREVIOUS EDITION IS OBSOLETE

DISPOSITION INSTRUCTIONS

DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.

DISCLAIMER

THE FINDINGS IN THIS REPORT ARE NOT TO BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION UNLESS SO DESIGNATED BY OTHER AUTHORIZED DOCUMENTS.

TRADE NAMES

USE OF TRADE NAMES OR MANUFACTURERS IN THIS REPORT DOES NOT CONSTITUTE AN OFFICIAL INDORSEMENT OR APPROVAL OF THE USE OF SUCH COMMERCIAL HARDWARE OR SOFTWARE.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) TR-CR-RD-AS-90-3, Vol III
6a. NAME OF PERFORMING ORGANIZATION College of Engineering 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Advanced Sensors Directorate RD&E Center
6c. ADDRESS (City, State, and ZIP Code) University of Alabama in Huntsville Huntsville, AL 35899	7b. ADDRESS (City, State, and ZIP Code) Commander, U.S. Army Missile Command ATTN: AMSMI-RD-AS Redstone Arsenal, AL 35898-5253
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAAH01-87-D-0021
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification) EXTENSION AND UPDATING OF THE COMPUTER SIMULATION OF RANGE RELATIVE DOPPLER PROCESSING AND INVARIANT MAPPING FOR MM WAVE SEEKERS, VOLUME III, SIMULATION OF A DOPPLER BEAM SHARPENING RADAR FOR MM WAVE SEEKERS

12. PERSONAL AUTHOR(S)
Robert J. Polge, Bassem R. Mahafza, Jong G. Kim

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Interim	FROM 4/3/88 TO 12/31/88	October 1990	78

16. SUPPLEMENTARY NOTATION
Report consists of 3 volumes.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Range Relative Doppler Processing (RRDP)
			MM Wave Seekers
			Invariant Mapping

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

This report extends the application of Range Relative Doppler Processing "RRDP" and of the Invariant Mapping Technique to beam doppler sharpening radars, where the goal is to map a large area on the ground. In this application the antenna follows a horizontal path while scanning a wide azimuth interval. Two programs are developed: DOPFP.FOR and DOPXY.FOR. The first detects targets within the sequence of footprints. The second program uses invariant mapping to generate a file which contains reflectivity data versus absolute coordinates for a selectable x-y window. An example illustrates the detection of (40) forty randomly distributed scatterers. The performance is quite good, except that the azimuth resolution is poor for footprints at very low azimuth angles. All the results are in complete agreement with the theory.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL A. H. Green	22b. TELEPHONE (Include Area Code) 205-876-1728 22c. OFFICE SYMBOL AMSMI-RD-AS

DD Form 1473, JUN 86 — Previous editions are obsolete. SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

PREFACE

The report for the work performed under Contract No. D.O. 0064, DAAH01–87–D–0021 consists of three volumes: (1) "Extension and Updating of the Computer Simulation of Range Relative Doppler Processing and Invariant Mapping for MM Wave Seekers", (2) "User Manual of the Range Relative Doppler Processing Simulation for MM Wave Seekers", and (3) "Computer Simulation of a Doppler Beam Sharpening Radar for MM Wave Seekers". The period of performance is April 12 to December 31, 1988.

The main objective of Volume I is to extend and update the MM wave computer simulation developed for Contract No. DAAH01–87–D–0021, D.O. 18, documented in the final report dated February 1988, and entitled "Increasing Azimuth and Elevation Resolution of MM Wave Seeker Systems Using Coherent or Noncoherent Range Relative Doppler Processing (RRDP) with Constant or Linear Frequency Modulation and Invariant Mapping".

Volume II is a User Manual for the computer simulation documented in Volume I. With this manual, MICOM personnel should be able to: (1) install the Fortran software on an IBM PC/compatible, (2) duplicate the results presented in this report, and (3) run the simulation for different clutter maps and targets.

The main objective of Volume III is to develop Doppler Beam Sharpening for MM wave seekers. In this application the geometry is significantly different. That is, the seeker follows a straight horizontal trajectory while the antenna performs a forward near circular scan perpendicular to the trajectory.

Robert J. Polge

Huntsville, Alabama

March 1989

ACKNOWLEDGEMENT

The Principal Investigator wishes to take this opportunity to acknowledge the cooperation and guidance of Mr. Jim Mullins and Mr. A. H. Green both of the U. S. Army Missile Command. Special Thanks are due to Mrs. Janie Parent for her efforts in typing the manuscript.

Table of Contents

Page

Accession For	
NTIS GRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. Introduction and Background

Two new digital processing techniques [1–4] were developed to extend applications of SAR processing to nonlinear trajectories. They are entitled, "Range Relative Doppler Processing (RRDP)", and "Invariant Mapping".

Volume I [3] presents the most recent update for these techniques. The major contributions are: (1) a more general geometry where the trajectory is not restricted to the vertical plane, (2) an exact simulation for the dish antenna, (3) a more exact signal synthesis that includes the scatterer height, (4) the development of an interactive subroutine which allows online updating of the data file, (5) a much better compensation for antenna gain and range attenuation, (6) a reduction through frequency interpolation of the peak reflectivity variation due to FFT quantization, (7) a new, better, and more efficient invariant mapping algorithm, and (8) online graphics displays for the reflectivity maps (3–D and contours). Two main programs were developed, FRRDP and CRRDP. The first generates the footprint reflectivity map versus azimuth and range. The second program uses our Invariant Mapping Technique to create an absolute x–y reflectivity map. Examples show that the scatterers are detected at the proper locations and that impulse invariance has been achieved over one footprint within ±3%, and over the entire absolute map within ±5.5%.

Volume II [4] is a user manual for the computer simulation documented in Volume I [3]. With this manual MICOM personnel should be able to: (1) install the Fortran software on an IBM PC/compatible, (2) duplicate the results presented in Volume I, and (3) run the simulation for different clutter maps and targets. This manual contains: (1) hardware and software requirements, (2) user overview of the simulation, (3) demonstration diskettes, (4) installation procedure, (5) definition of inputs, (6) display information for FRRDP output, and (7) display information for CRRDP output.

The present report, which is denoted as Volume III, extends the application of RRDP and invariant mapping to Beam Doppler Sharpening Radars, where the goal is to map a wide area on the ground. In this application, the antenna platform follows a horizontal flight path with constant velocity, while the antenna is scanned to cover the desired azimuth interval.

Two main programs are developed: DOPFP.FOR and DOPXY.FOR. The first program detects scatterers in one or in all footprints. It also generates a file DOPFPG.GRD which contains reflectivity data versus relative doppler and range for the selected footprint. Using SURFER [5] on the file DOPFPG.GRD produces 3–D and contour plots of the reflectivity versus relative doppler and range. The second program includes invariant mapping and generates a file DOPXYG.GRD which contains reflectivity data versus absolute x–y coordinates for a selectable x–y window. Using SURFER on the file DOPXYG.GRD produces 3–D and contour plots of the reflectivity versus x–y coordinates for a selected x–y window. Again one can run the simulation for one or all footprints.

While many of the subroutines developed in Volume I are also used in DOPFP and DOPXY, several new complex subroutines were developed and the main programs are completely new. Problem areas include: (1) extending the simulation from one footprint to a sequence of footprints, (2) making the program work for any azimuth within the required interval, (3) defining the sequence of footprints for optimum coverage, (4) integrating the information for a sequence of footprints through invariant mapping, (5) developing subroutines to detect peaks and print for each peak: amplitude, footprint index, azimuth, range, and x–y coordinates, (6) developing low resolution displays for the very large area being mapped, and (7) developing high resolution 3–D or contour displays for the selected x–y window, within the limitation of MS–DOS.

The organization is as follows: (1) the geometry and system parameters for a single footprint are discussed, (2) the footprint simulation is extended to arbitrary azimuth, (3) the generation of a sequence of footprints for optimum scanning is presented, (4) the resolution is analyzed as a function of azimuth angle and time dwell, (5) a block diagram shows an overview of DOPFP, (6) another block diagram shows an overview of DOPXY, (7) the inputs and outputs for DOPFP and DOPXY are explained, (8) examples demonstrate DOPFP, and (9) the same examples demonstrate DOPXY. The last section contains conclusions and recommendations for future work.

2. Geometry and System Parameters for a Single Footprint

In Range Relative Doppler Processing (RRDP) the information of ground reflectivity is acquired sequentially, one footprint at a time. As in the earlier reports, the dwell interval is 20 ms which yields a frequency resolution of 50 Hz. The contour of the footprint is defined as the 3 dB contour at central dwell time. As usual the footprint is first resolved in range by range gating online where the range cell is the inverse of the system bandwidth. Then each range cell is divided into FFT cells on the base of relative doppler frequency, where zero doppler corresponds to the center of the range cell. For a typical antenna azimuth angle there are about 40 non–interpolated azimuth cells within a range bin. Thus, the azimuth resolution is improved by a factor of about 40. The center and boundary of each FFT cell can be mapped onto the x–y absolute coordinates. It follows that a peak amplitude in an FFT cell corresponds to a peak reflectivity for which the x–y coordinates can be computed. It has been shown that the relative azimuth [1–3] (measured with respect to the ground line of sight GLOS) varies nearly linearly with frequency. Therefore the display of reflectivity versus range and FFT index is very useful. Using Invariant Mapping, one can map each azimuth cell onto the x–y plane so as to obtain a reflectivity map versus absolute coordinates. For more details on RRDP for a single footprint one should refer to Volumes I and II. Figure 2.1 shows the geometry for a

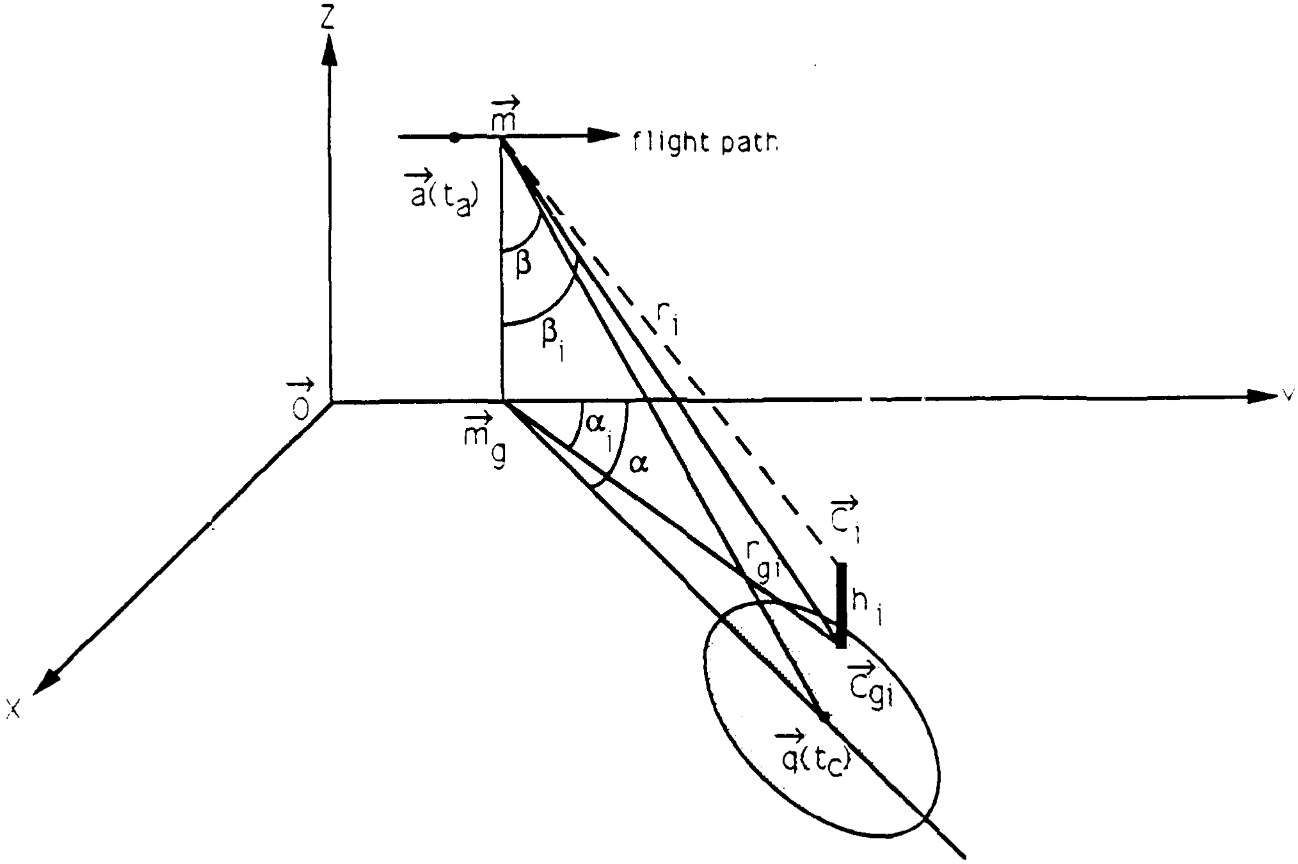


Figure 2.1 Geometry for a single footprint.

single footprint. In our application, the trajectory is an horizontal line in the vertical plane with height (h = 300 m), and constant velocity (v = 170 m/s). The elevation angle β of the antenna, which is the complement of the depression angle, is equal to 75°. The azimuth angle α is within the interval -30° to 30°.

In Fig. 2.1 the position of the antenna at zero absolute time is

$$\vec{a}_0 = (0, 0, h) . \tag{2.1}$$

Then the position of the antenna at central dwell time t_c is

$$\vec{a}(t_c) = \vec{m} = (0, y_m = v\, t_c, h) . \tag{2.2}$$

Given the azimuth and elevation angles (α, β^*) and the central dwell time t_c, one can easily compute the coordinates of the footprint center;

$$x_q = h \tan \beta^* \sin\alpha \tag{2.3a}$$

$$y_q = h \tan \beta^* \cos\alpha + v\, t_c . \tag{2.3b}$$

Appendix A contains all the relevant footprint information. This Appendix is very similar to Appendix A of Volume I [3], except that the maximum range is computed differently because of the small depression angle. The same notation is used as in Volume I: Δr is the range bin width and the footprint center is given by,

$$\vec{q}^* = \vec{q}(t_c) = \{x_q, y_q, 0\} . \tag{2.4}$$

The relevant information for the ith scatterer $\vec{C}_i$ is in Appendix B. The absolute coordinates of the scatterer $\vec{C}_i$ and of its ground projection $\vec{C}_{gi}$ are:

$$\vec{C}_i = \{x_i, y_i, h_i\} \tag{2.5a}$$

$$\vec{C}_{gi} = \{x_i, y_i, 0\} . \tag{2.5b}$$

As in Volume I, the relative azimuth and elevation for $\vec{C}_i$, are denoted as (μ_i, ϵ_i) and they are computed with respect to $\vec{C}_{gi}$. More precisely, the azimuth and elevation, with respect to $\vec{m}$, of a scatterer $\vec{C}_i$ in the jth range bin are,

$$\alpha_i = \alpha + \mu_i \tag{2.6a}$$

$$\beta_i = \beta_j^* + \epsilon_i \tag{2.6b}$$

where β_j^* is the elevation angle for the center of the jth range cell.

The position of the antenna within the observation interval D_{ob}, is given by

$$\vec{a}(t_a) = \vec{a}(t + t_c) = \{0, v\, t_a = v(t+t_c), h\} \tag{2.7a}$$

$$-\frac{D_{ob}}{2} \leq t \leq \frac{D_{ob}}{2} \tag{2.7b}$$

where t_a denotes absolute time, and t denotes relative time.

As explained in Volume I the gain pattern for a circular dish of diameter D is [6,7]

$$G(sD/\lambda) = 2A \frac{J_1(\pi s D/\lambda)}{(\pi s D/\lambda)} \tag{2.8}$$

where J_1 is a Bessel function of the first kind of the first order, and A is the area of the circular dish aperture. The normalized dish pattern is in Fig. 3.1 of Volume I.

The list of system parameters is in Table 2.1. These parameters are different significantly for those of Volume I in three areas: (1) lower height, (2) lower depression angle, and (3) horizontal velocity.

3. Reflectivity Versus Relative Doppler and Range for a Single Footprint: DOPFPS

A main program DOPFP.FOR has been developed to detect scatterers in one or in all footprints. This program also generates a file DOPFPG.GRD which contains reflectivity data versus relative doppler and range for the selected footprint. Using SURFER [5] on the file DOPFPG.GRD produces 3–D and contour plots of the reflectivity versus relative doppler and range. Since the program DOPFP.FOR is quite complex, this section explains only a simplified program operating on a single footprint. This fictitious single footprint program is denoted as DOPFPS.FOR.

The block diagram for DOPFPS.FOR is in Fig. 3.1. This program differs from RRDPF.FOR of Volume I as follows: (1) the simulation is extended to negative azimuth angles, (2) the antenna gain compensation has been improved, (3) the clutter map is now defined for an arbitrary number of footprints, yet the signal simulation for a range cell involves only the relevant scatterers, (4) subroutines were developed to detect and print the peaks and the associated information after processing, and (5) the structure of the

Table 2.1. List of system parameters

.0200000	TIMDWL=observation interval
1.0000000	DELRAN=range bin
20.0000000	XMMNPR=threshold to print XM(KFF)
.7500000	WPAR(1)=rectangular grid increment @XY
.1000000	WPAR(2)=mapping threshold in FP or XY
287.5000000	WPAR(3)=window(xmin-abs.) @XY
1276.2500000	WPAR(4)=window(ymin-abs.) @XY
357.2500000	WPAR(5)=window(xmax-abs.) @XY
1346.0000000	WPAR(6)=window(ymax-abs.) @XY
102.0000000	WPAR(7)=window(xsize) @XY
187.0000000	WPAR(8)=window(ysize) @XY
.0000000	XIFXMT=transmitter phase
4.0000000	SELWIN=define FFT window
.0000000	DETIPR= to print PRXYSC-1 @FP
2.0000000	SELDIS=define dish simulation
1.0000000	WINIPR=(0.,1.) window (WPAR,center FPT)
.0000000	SCANEW=-999. to enter scatterer/s online
300.0000000	HEIREF=ref-height for RANCOF(init-value)
1.3090000	BETREF=ref-beta for RANCOF (init-value)
-999.0000000	SHOSCA=-999. for scatterer-data in FP 3@
1.0000000	BUGLEV:(0. for noPRT);((1.,2.) to debug)
140.0000000	FPMXNB= bound on # footprint processed
1.0000000	SURFIN= 1. to write the file DOPFPG.GRD

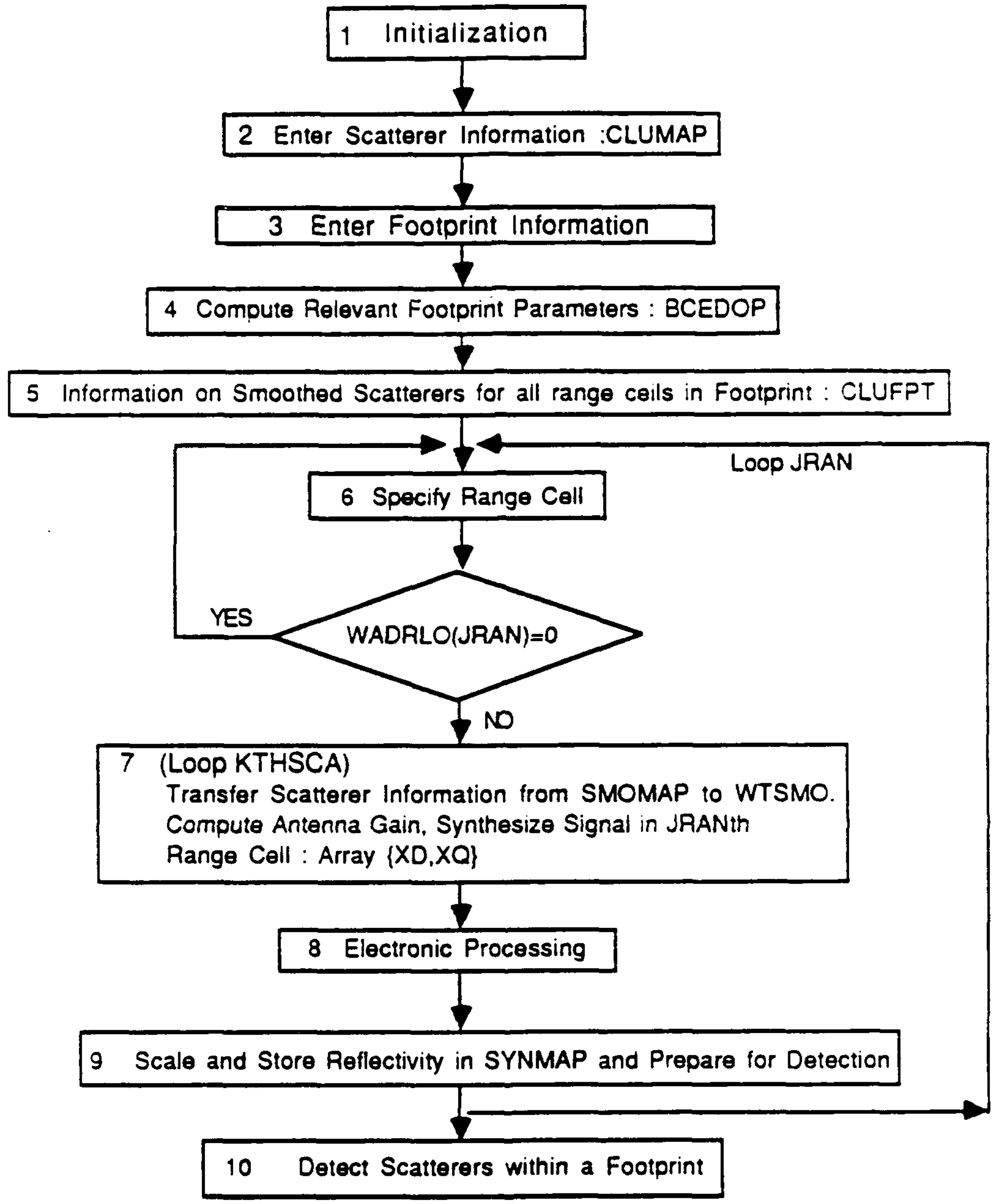


Figure 3.1 Simulation of a single footprint : DOPFPS

program has been greatly improved. The listings of DOPFP and of the new or modified associated subroutines are in Appendix C. For the other subroutines, refer to Volume I.

In the initialization block, the system parameters are entered interactively by reading, displaying on the screen, and updating the system input parameter file SYPAI.INP (see Table 2.1). In SYPAI.INP the parameters that are only for debugging are specified by the symbol "@@". Parameters only used in DOFPF or only in DOPXY are indicated respectively by, "@FP" and "@XY". The subroutines WINPUX is used to enter the parameters interactively. WINPUX is the same as WINPUT, which was explained in Volume I, except6 that the data is displayed on the screen before and after the changes. The arrays that are independent of the footprint, are also computed. These include:

{WTIME ⇔ relative sampling time}

{WFREQ ⇔ relative doppler frequency}

{WIND ⇔ Kaiser window for FFT}

{LNPPP ⇔ *translated* FFT *index*}.

The purpose of {LNPPP} is to use the index KNPPP which varies from 1 to NFFT from minimum to maximum doppler, rather than the FFT index KFF which does not correspond to a continuous doppler variation. The correspondence is : {KNPPP = 1 ⇔ KFF−1 = (NFFT/2 + 1)}, where KFF is modulo NFFT.

The scatterer information is entered in block 2 by reading the clutter data file CLMPI.INP. Table 3.1 shows a clutter file for 40 scatterers. The format is as follows: (1) number of scatterers, and (2) one row of data for each scatterer: {amplitude, phase, x–coordinate, y–coordinate, z–coordinate, scatterer index}. The scatterer data is read, and displayed on the screen.

In block 3 the footprint information is entered by reading one row of the file FPSQI,INP. The eight parameters defining the footprint are:

Table 3.1. Clutter data file CLMPI.INP for 40 scatterers

40.000					
1.000	.000	-447.409	1083.000	.000	1.000
1.000	.000	-511.018	1204.927	.000	2.000
1.000	.000	-460.390	1121.403	.000	3.000
1.000	.000	-530.195	1228.854	.000	4.000
1.000	.000	-450.643	1092.814	.000	5.000
1.000	.000	-349.354	1194.242	.000	6.000
1.000	.000	-320.981	1214.717	.000	7.000
1.000	.000	-355.527	1276.781	.000	8.000
1.000	.000	-372.186	1320.675	.000	9.000
1.000	.000	-347.018	1267.020	.000	10.000
1.000	.000	-199.092	1322.478	.000	11.000
1.000	.000	-203.157	1243.043	.000	12.000
1.000	.000	-184.739	1312.798	.000	13.000
1.000	.000	-220.562	1300.950	.000	14.000
1.000	.000	-229.101	1346.910	.000	15.000
1.000	.000	-61.271	1246.853	.000	16.000
1.000	.000	-62.473	1266.286	.000	17.000
1.000	.000	-61.264	1430.723	.000	18.000
1.000	.000	-43.413	1369.523	.000	19.000
1.000	.000	-50.957	1262.079	.000	20.000
1.000	.000	77.345	1411.096	.000	21.000
1.000	.000	72.384	1418.710	.000	22.000
1.000	.000	45.316	1275.881	.000	23.000
1.000	.000	70.514	1300.239	.000	24.000
1.000	.000	62.073	1446.496	.000	25.000
1.000	.000	226.499	1434.587	.000	26.000
1.000	.000	193.773	1324.643	.000	27.000
1.000	.000	190.213	1270.189	.000	28.000
1.000	.000	192.693	1307.827	.000	29.000
1.000	.000	215.186	1324.218	.000	30.000
1.000	.000	331.811	1277.989	.000	31.000
1.000	.000	353.076	1333.173	.000	32.000
1.000	.000	318.932	1292.184	.000	33.000
1.000	.000	367.597	1349.559	.000	34.000
1.000	.000	314.725	1250.479	.000	35.000
1.000	.000	514.790	1283.882	.000	36.000
1.000	.000	499.218	1298.021	.000	37.000
1.000	.000	505.042	1274.220	.000	38.000
1.000	.000	502.415	1350.866	.000	39.000
1.000	.000	450.037	1250.946	.000	40.000

VELOM ⇔ Missile velocity

ALFM ⇔ Azimuth angle for $\vec{q}(t_c)$

BETM ⇔ Elevation angle for $\vec{q}(t_c)$

XM ⇔ x–coordinate for missile at central time

YM ⇔ y–coordinate for missile at central time

ZM ⇔ z–coordinate for missile at central time = h

XFP ⇔ x–coordinate for footprint center

YFP ⇔ y–coordinate for footprint center

Block 4 is the computation of all the relevant footprint parameters. This is accomplished by subroutine BCEDOP, which implements the formulas given in Appendix A.

In block 5 the information for the scatterers that fall within the footprint is computed using subroutine CLUFPT. As well known, a point scatterer affects three consecutive range cells because of the finite bandwidth of the system. This effect which is denoted as range smoothing is simulated using subroutine RANGSMOO, which splits each scatterer into three equivalent scatterers. All the necessary scatterer information is passed to the main program through the arrays SMOMAP, WADRLO, WADRUP, and WADSMO.

The heart of DOPFPS in "loop JRAN" which corresponds to "DO 35" in the program. In this loop the range cells are processed sequentially: JRAN = 1, JRANMX.

In block 6 a check for the presence of smoothed scatterers within the current JRANth cell is performed. If WADRLO(JRAN) =0 there are no scatterers and "loop JRAN" has been completed.

Block 7 represents "loop KTHSCA" which corresponds to "DO 30" in the program. Its purpose is to synthesize the signal {XD(KDQ), XQ(KDQ), KDQ = 1, NFFT} in the

JRANth range cell. The contributing scatterers are defined by {WADRLO(JRAN) < KTHSCA < WADRUP(JRAN)}. The information for the KTHSCA scatterer is transferred from SMOMAP into the array WTSMO. The antenna gain and the contribution to the signal due to KTHSCA scatterer are computed using subroutines SCAGAN and TSIGNAL.

The electronic processing of the JRANth cell is in block 8. It consists of the following steps: (1) deramping, (2) windowing, and (3) fast Fourier transformation (FFT). The subroutines DERAMP, WINDOWS and FFT are used for this purpose. Note that the FFT is performed on the extended arrays {XD(k), XQ(k), k = 1, 2*NFFT} which are padded with NFFT zeros. This corresponds to the implementation of frequency interpolation which was discussed in section 6 of Volume I.

Block 9 is concerned with the scaling of the FFT cells, the synthesis of the reflectivity array SYNMAP, and the preparation for detection. First, the array WAGNCL which contains the antenna gains at the center of the FFT cells is computed, ("DO 39" in the program). Then, both the scaling and the computation of SYNMAP are performed in "DO 41". There are two important subroutines: SURFMP and DETFPT, which are listed in Appendix C. The subroutine SURFMP uses the Common /SURFP/, and has three entry points. The first entry point, in block 1, is for initialization of the array SYNMAP. The second entry point, in block 9, is for storing reflectivity data in SYNMAP. The subroutine DETFPT includes the Common /DETECT/ and has also three entry points. The first entry point is used in block 9 to store the peak values above the specified threshold within the current range cell.

Block 10 is the last step in the processing of a single footprint. Subroutine SURFMP is called in the third entry mode to generate DODFPG.GRD from SYNMAP in the format specified by SURFER [5]. DETFPT in the second entry mode selects the peaks within the entire footprint.

4. Reflectivity Versus Absolute x–y Coordinates for a Single Footprint: DOPXYS

A main program DOPXY.FOR has been developed to detect scatterers in one or in all footprints. This program includes invariant mapping and generates a file DOPXYG.GRD which contain reflectivity data versus absolute x–y coordinates for a selectable x–y window. Using SURFER on the file DOPXYG.GRD produces 3–D and contour plots. Again one can run the simulation for one or all footprints. In the case of overlapping footprints, the reflectivity information is accumulated and smoothed through invariant mapping. The listings of DOPXY, MAPRGD, MAPAZD are in Appendix D.

Since the program DOPXY.FOR is quite complex, this section explains only a simplified program operating on a single footprint. This fictitious single footprint proggram is denoted DOPXYS.FOR.

The block diagram for DOPXYS.FOR is in Fig. 4.1. The first eight blocks of Fig. 4.1 are the same as in DOPFPS.FOR and will not be discussed further. Both DOPFPS and DOPXYS use the same input files.

In DOPXY the x–y window can be selected in two ways: (1) manual specification, and (2) automatic specification. In the first case the window parameters in the input file SYPAI.INP are set as follows:

WINIWR = 0.

WPAR(3) = x–for minimum corner

WPAR(4) = y–for minimum corner

WPAR(5) is computed

WPAR(6) is computed

WPAR(7) = size of window along x–coordinate

WPAR(8) = size of window along y–coordinate.

In the second case, WINIWR is set to equal (1.), and the window of size (WPAR(7), WPAR(8)) is automatically centered on the center of the selected footprint. The window

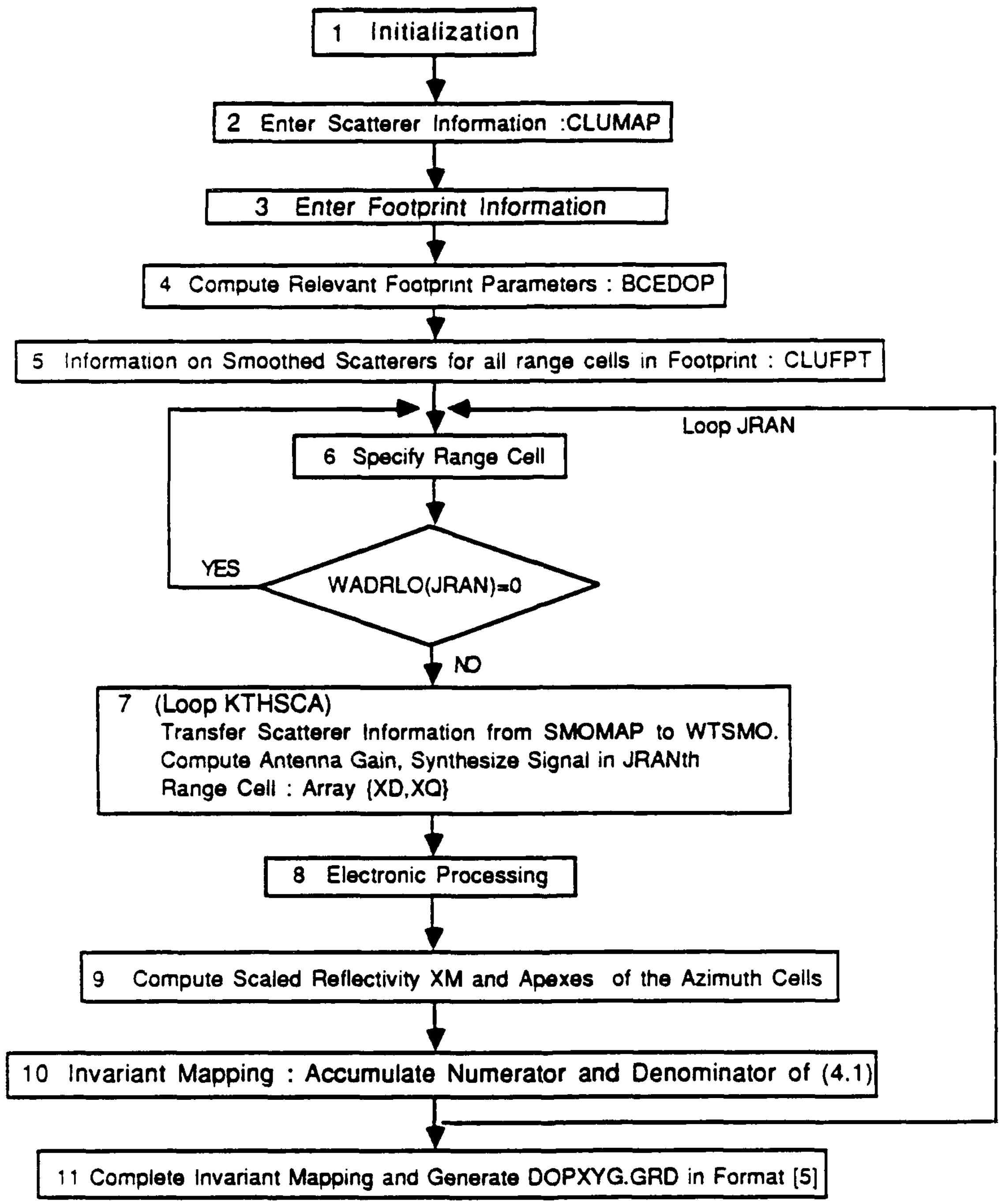


Figure 4.1 Simulation of a single footprint : DOPXYS

is rounded up to integer values of DELMAP = WPAR(1) using the subroutine MAPWIN with entry 1. The automatic window is set by MAPWIN with entry 4. The subroutine MAPWIN is listed in Appendix D.

Block 9 is concerned with scaling of the FFT cells and computation of trapezoid apexes coordinates. First, the array WAGNCL which contains the antenna gains at the center of the FFT cells is computed, ("DO 39") in the program). Then, scaling for range attenuation is performed in "DO 41". Also, subroutine APEXTRAP, which was discussed in Volume I, computes the apex coordinates for the trapezoidal FFT cells in the current range cell.

As explained in Volume I the invariant mapping algorithm implements the following formula

$$RFR(k) = \frac{\sum_{m} OVL(m,k) RFT(m)}{\sum_{m} OVL(m,k)} \tag{4.1}$$

where: m defines all the trapezoids which overlap the kth rectangle, and

TRP(m) $\Leftrightarrow$ mth trapezoidal cell

RFT(m) $\Leftrightarrow$ reflectivity of the mth trapezoidal cell

REC(k) $\Leftrightarrow$ kth absolute rectangular cell

RFR(k) $\Leftrightarrow$ estimated reflectivity of the kth rectangular cell

OVL(m,k) $\Leftrightarrow$ overlap area between mth trapezoid and kth rectangle.

Block 10 uses subroutine MAPRGD to compute the array SYNMAP and SUMARA which represent respectively, the numerator and denominator of (4.1). The accumulation in SYNMAP and SUMARA is performed by subroutine MAPAZD which is called by MAPRGD. Communication between DOPXYS and MAPAZD is through common /GRDMAP/.

Block 11 corresponds to the two "DO 60" in the program. First, the reflectivity map SYNMAP is computed according to (4.1). Then, the reflectivity map versus absolute x and y coordinates is written on the file DOPXYG.GRD using the format recommended by [5].

5. Sequence of Footprints for Scanning

In our application the antenna platform moves at a constant speed along a horizontal path as explained in section 2. The absolute coordinates of the platform at absolute time t_a are,

$$\vec{a}(t_a) = \{0, v\, t_a, h\}. \tag{5.1}$$

The elevation angle β, is constant and equal to 75°. For each azimuth angle $-30^\circ < \alpha < 30^\circ$ one can compute the coordinates of the footprint center (x_q, y_q) as explained in Appendix A.

Scanning can be defined as selecting a sequence of azimuth values $\{\alpha\}$ which will provide a complete ground coverage including some overlap. Subroutine FPIDOP, which is listed in Appendix D, is used to compute the sequence of angles α, and to write the footprint information for each α value. The assumptions are as follows: (1) the initial value of α is about -30°, (2) each dwell interval for every footprint is 20 msec, (3) the time interval between consecutive footprints is greater than 20 msec to allow step scanning, (4) mapping is performed on increasing α from about -30° to $+30^\circ$ with fast return from -30° to $+30^\circ$, and (5) there should be sufficient overlap between consecutive footprints and scans in order to guarantee detection.

The inputs of FPIDOP are read in the file GEODOPI.INP. The parameters used in our examples are in Table 5.1.

The subroutine FPIDOP writes the file FPSQI.INP which contains the footprint information. More precisely, one row in FPSQI.INP defines for each dwell time: (1) the velocity vector, (2) the antenna position, and (3) the center of the footprint. There are

Table 5.1. Geometry parameters for scanning: GEODOPI.INP.

170.0000000	VELOM= seeker velocity
.0000000	ALFM= seeker azimuth angle
1.5707960	BETM= seeker elevation angle
.0000000	WBRUN(1)=x-begining of run
.0000000	WBRUN(2)=y-begining of run
300.0000000	WBRUN(3)=z-begining of run
1.3089970	BETFP= footprint elevation angle
1.0472000	ALFCY= cycle azimuth coverage
1.4300000	OVRLAP=footprint overlap (1,1.5)
4.0000000	CYNB= # cycles in run
.0100000	TIMGAP= time gap between cycles (sec)
-.5235988	ALFSTR=initial ALFFP for a run

nine components per row: VELOM, ALFM, BETM, XM, YM, ZM, XFP, YFP, and ALFFP. The last component ALFFP is used only for debugging. The main formulas in FPIDOP are

$$\text{NBCY} = \#\text{ of cycles} = \text{input}$$

$$\text{ANGCY} = \text{angle per cycle from } \vec{m}$$

$$= \cos^{-1}\{\sin^2(\text{BETFP}) + \cos(\text{ALFCY}) + \cos^2(\text{BETFP})\}$$

$$\text{NBFPCY} = \text{footprints per cycle} = 2\ \text{INT}\left[\frac{\text{ANGCY*OURLAP*0.5}}{\theta_{3\text{dB}}} + \epsilon\right]$$

$$\text{DLALF} = \alpha \text{ increment in FP sequence}$$

$$= \frac{\text{ALFCY}}{\text{NBFPCY}}$$

$$\text{FPLNG} = \text{FP length} = h\,[\tan(\text{BETFP}+\theta_H) - \tan(\text{BETFP}-\theta_H)]$$

$$\text{PERCY} = \text{period per cycle} = \frac{\text{FPLNG}}{V_y\ \text{OVRLAP}}$$

$$\text{DLTISQ} = \text{time increment in FP sequence} = \frac{\text{PERCY} - \text{TIMGAP}}{\text{NBFPCY}} .$$

Figure 5.1 shows the sequence of footprint centers for the first four cycles. The unit for x and y are meters. Figure 5.2 shows that the footprint overlap in both azimuth and range to guarantee complete coverage.

6. Predicted Resolution: azimuth and range

The relative doppler in the jth range cell for a horizontal flight path is given by

$$\text{rdop} = -\sin\beta_j^* \cdot v(\cos(\alpha^* + \mu) - \cos\alpha^*) \tag{6.1}$$

where β_j^* is the elevation angle of the jth range cell and μ defines the position of the scatterer with respect to the ground line of sight (GLOS). This formula is in complete

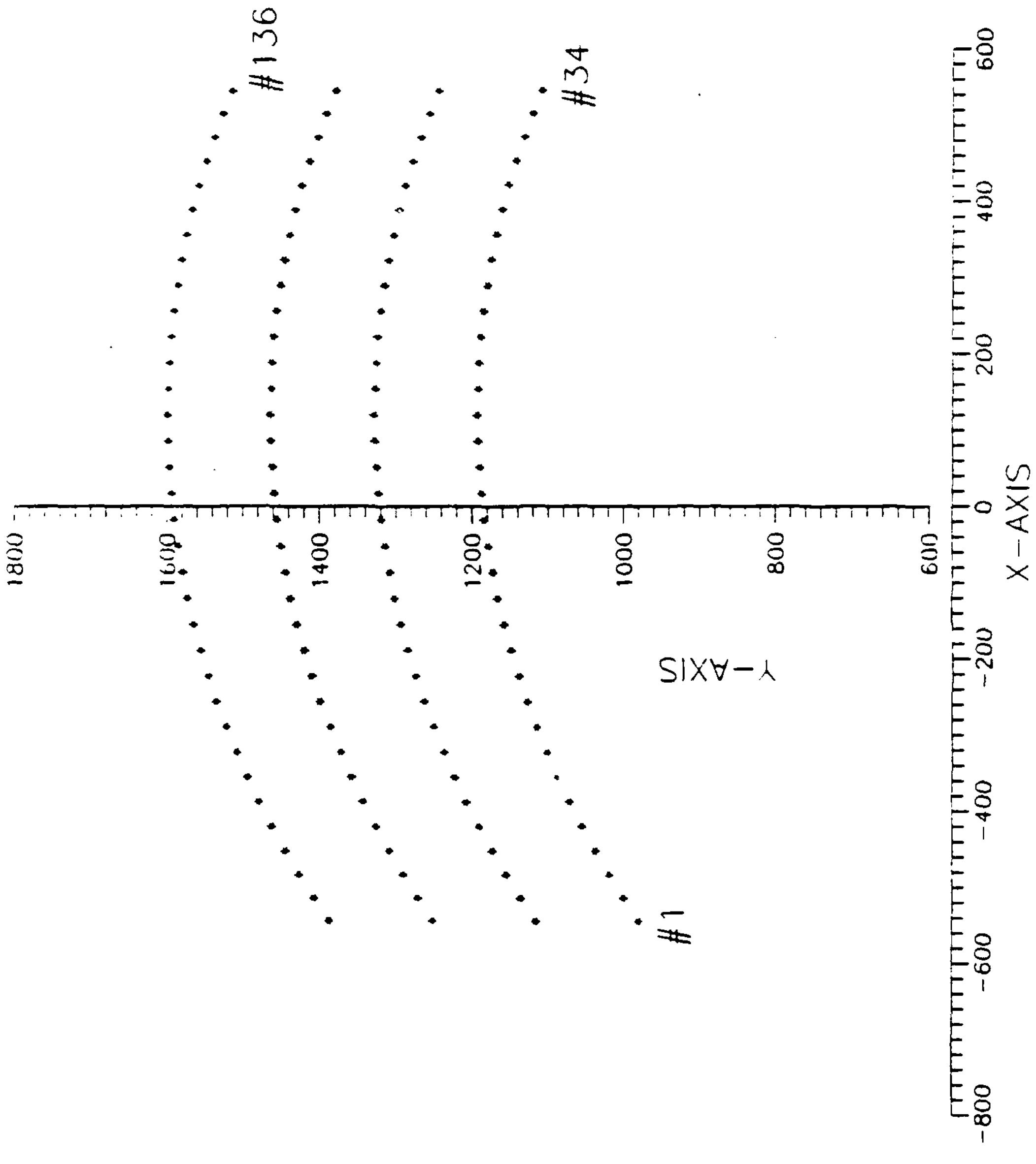


Figure 5.1. Sequnce of footprint centers for four cycles.

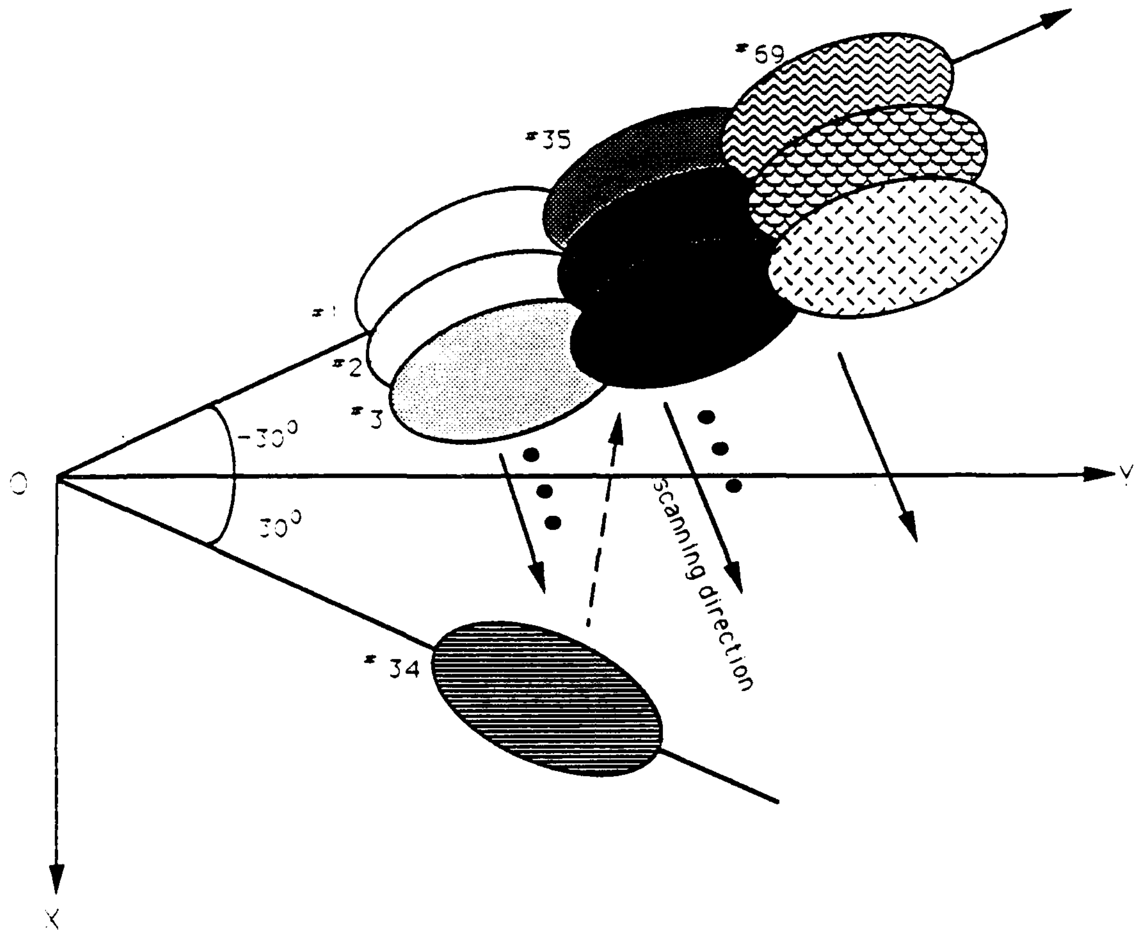


Figure 5.2 Illustration of footprint overlaps.

agreement with the relative doppler computed from the Taylor expansion or by direct dot product. During the sequence of footprints, the dwell time per footprint remains constant and it is equal to 20 ms. It follows that the doppler resolution in any range cell or footprint is equal to $\Delta f = 50$ Hz. However, the azimuth resolution is not constant. The azimuth resolution at specified (α^*, β_j^*) can be computed from (6.1). More precisely, the angular azimuth resolution is given by

$$\mu = \cos^{-1}\left[\frac{\lambda \ \mathrm{rdop}}{2v \ \sin \beta_j^*} + \cos \alpha^*\right] - \alpha^* , \tag{6.2}$$

where rdop = 50 Hz. Figure 6.1 shows the variation of the azimuth resolution for the central range cell $\beta_j^* = 75^\circ$, when the absolute value of α varies from 0° to 45°. One observes that the azimuth resolution is 10 times better at $\alpha^* = 30^\circ$ than at $\alpha^* \cong 2.5^\circ$. On the other hand, the azimuth resolution is quite good for $\alpha > 8^\circ$. We will show that all the targets can be detected, even for very small α^*.

Range resolution, which is the inverse of the system bandwidth, is assumed constant and equal to 1 m.

7. Overview of the Simulation of a Sequence of Footprints: DOPFP

The purpose of DOPFP is to simulate the detection of targets within an azimuth interval from -30° to $+30^\circ$, using RRDP to increase the azimuth resolution. A sequence of footprints is generated to insure a complete ground coverge, including a sufficient azimuth and range overlap. The processing of a single footprint has been explained in section 3 using the fictitious program DOPFPS.

Figure 7.1 gives an overview of DOPFP. Since DOFPF is an extension of DOPFPS only the additional features need to be explained. The initialization block in DOPFP includes an online prompt which is self explanatory:

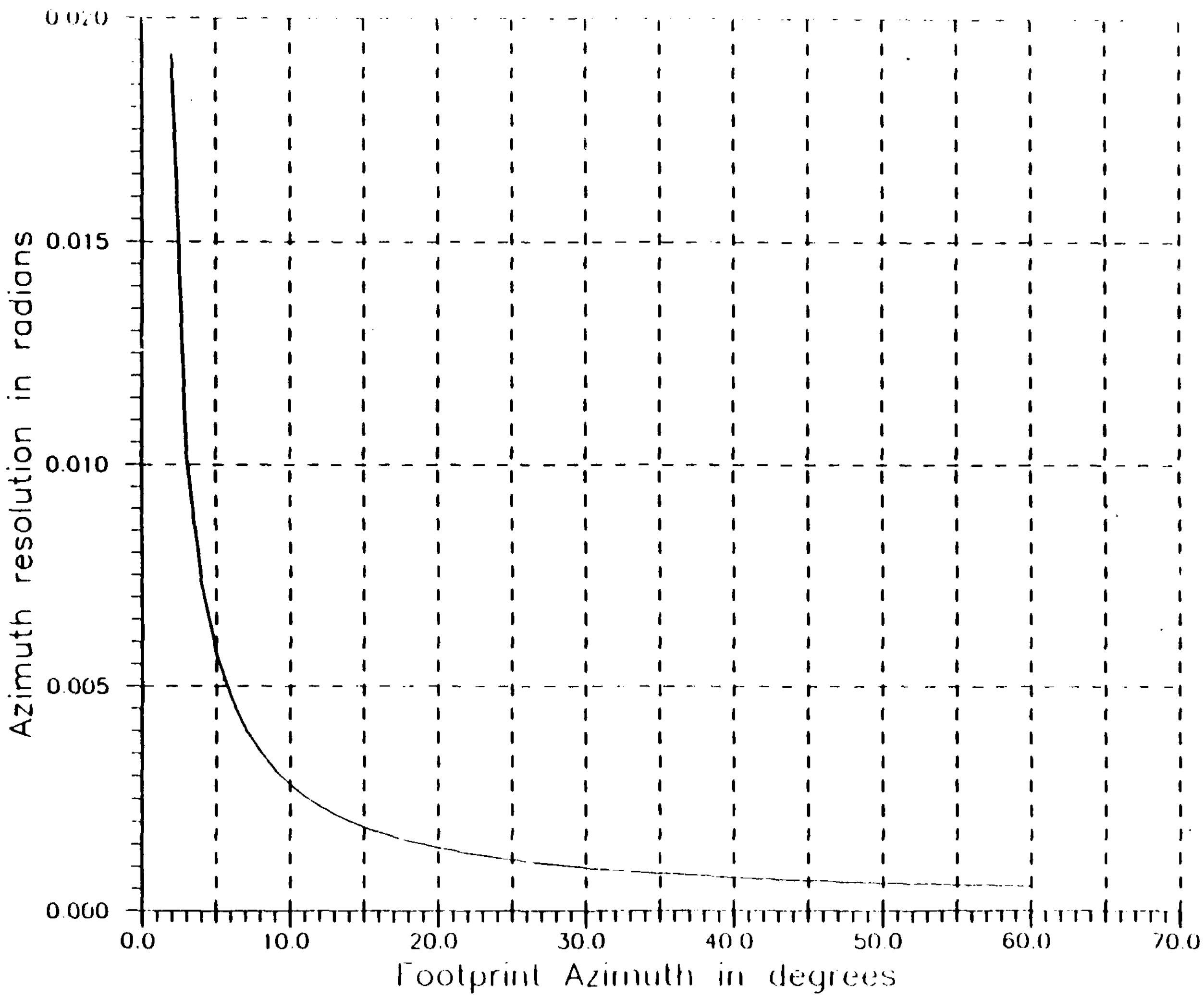


Figure 6.1. Azimuth resolution versus footprint azimuth.

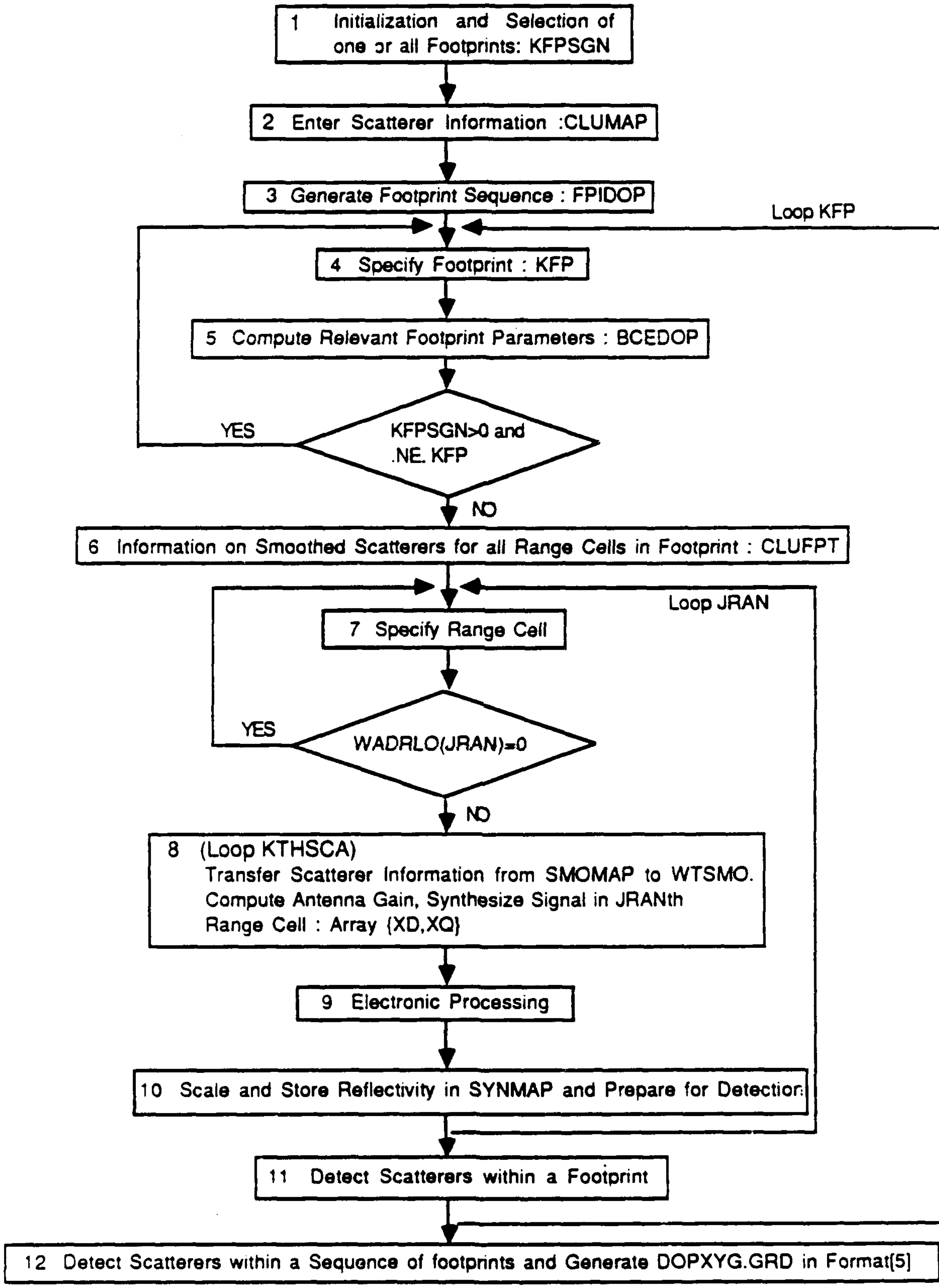


Figure 7.1 Simulation of a sequence of footprints : DOPFPG.GRD contains reflectivity data versus relative doppler and range.

"SELECT KFPSGN: (1) –999 = Abort, (2) positive to process and display only footprint #KFPSGN, and (3) negative to process all footprints and display footprint #|KFPSGN|"

"K = ?"

While the ABORT command is instantaneous, the actual selection of one or all footprints is implemented within Loop KFP ("DO 1" in the program), just after computing all the footprint parameters with BCEDOP. For example if "16" is entered, then only the 16th footprint is processed, all the others being skipped. Targets within the 16th footprint are detected. At the same time DOPFP generates the file DOPFPG.GRD which contains reflectivity data versus relative doppler and range, for later display by SURFER. On the other hand if "–16" is entered, then all the footprints are processed. In this case all the ground targets should be detected. Again DOPFPG.GRD contains only the reflectivity information for the 16th footprint.

Target detection is implemented using subroutine DETFPT which was introduced in Section 3. As earlier the first two entries points are used: (1) to store the peak values above the specified threshold within the current range cell (block 9), (2) to select the peaks within a footprint (block 10). The third entry point is used for the final peak selection (block 11).

The generation of the sequence of footprints for the selected scanning scheme is performed by subroutine FPIDOP (block 3) which was discussed in Section 5.

8. Overview of the Simulation with Invariant Mapping: DOPXY

The objective of DOPXY is to generate an x–y reflectivity map for the selected sequence of footprints using both RRDP and invariant mapping. It follows that DOPXY is very similar to DOPFP. The main difference is in the output files: DOPFPG.GRD contains the reflectivity data versus relative doppler and range for the |KFPSGN|th footprint, while DOPXYG.GRD contains the reflectivity data versus absolute x–y coordinates for the KFPSGNth footprint if KFPSGN>0 or for all footprints if KFPSGN<0. Another difference is that the exact determination of reflectivity peaks using subroutine DETECT is implemented in DOPFP but not in DOPXY, since this would be a duplication. Figure 8.1 gives an overview of DOPXY.

The two programs DOPFP and DOPXY use the same input files and provide complementary information. The main output of DOPFP is the list of detected scatterers while the main output of DOPXY is the file DOPXYG.GRD which contains the reflectivity data versus x–y. Invariant mapping [1–3] performs an incoherent integration on the footprint information. This can be demonstrated by selecting an x–y window and running DOPXY two ways: (1) for a single footprint within the window, and (2) for all the footprints. One can see the smoothing effect due to integration. Also, additional scatterers detected within other footprints appear in the x–y window.

9. Inputs and Outputs for DOPFP and DOPXY

DOPFP and DOPXY use the same input files:

SYPAI.INP ⇔ System parameters file

CLMPI.INP ⇔ Clutter data file

GEODOPI.INP ⇔ Geometry parameter file for scanning.

An example of SYPAI.INP is listed in Table 2.1 and discussed in section 3. An example of CLMPI.INP is in Table 3.1 and is discussed in section 3. The file GEODOPI.INP is discussed in section 5 and an example is in Table 5.1.

The simulation DOPFP has two output files:

DOPFPP.PRT ⇔ printout file used for debugging, also contains the final list of detected scatterers.

DOPFPG.GRD ⇔ reflectivity data versus relative doppler and range using SURFER format.

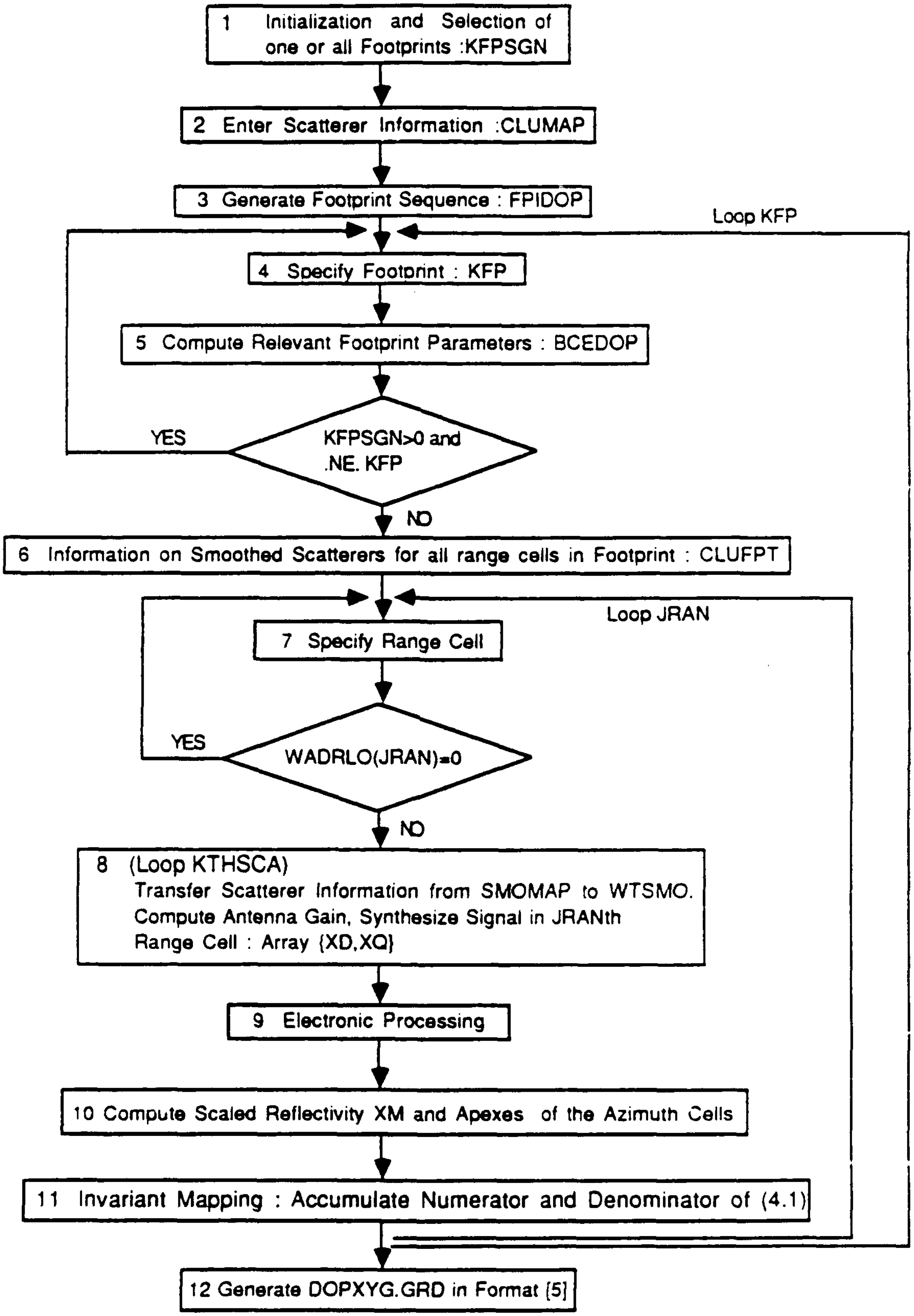


Figure 8.1 Simulation of a sequence of footprints : DOPXYG.GRD contains reflectivity data versus absolute x-y coordinates.

Table 9.1 shows the final list of detected scatterers obtained when using the clutter input file from Table 3.1. The columns of Table 9.1 are as follows: {peak index, debugging index, footprint, range bin, doppler index, peak amplitude, x–coordinate, y–coordinate}.

The simulation DOPXY also has two output files:

DOPXYP.PRT $\Leftrightarrow$ printout file used for debugging purposes.

DOPXYG.GRD $\Leftrightarrow$ reflectivity data versus absolute x–y coordinates in SURFER format.

10. Demonstration of the DOPFP Simulation

The performance of DOPFP is demonstrated by detecting (40) forty randomly distributed scatterers on the ground being scanned. The input files for this example are: (1) SYPAI.INP in Table 2.1, (2) CLMPI.INP in Table 3.1, and (3) GEODOPI.INP in Table 5.1. The HP 7475A plotter was used to obtain hard copies.

Running DOPFP with a negative value for KFPSGN will generate the list of detected targets given in Table 9.1. Comparison of the position of the detected peaks to the actual scatterer location in Table 3.1 shows that the detection error is quite small (within ± 0.5 m in each coordinate). The detected peaks should theoretically be equal. However, the detected peak amplitude varies within $\pm 9\%$ of the mean. Running DOPFP with a positive value for KFPSGN will generate the list of the scatterers detected within the (KFPSGN)th footprint.

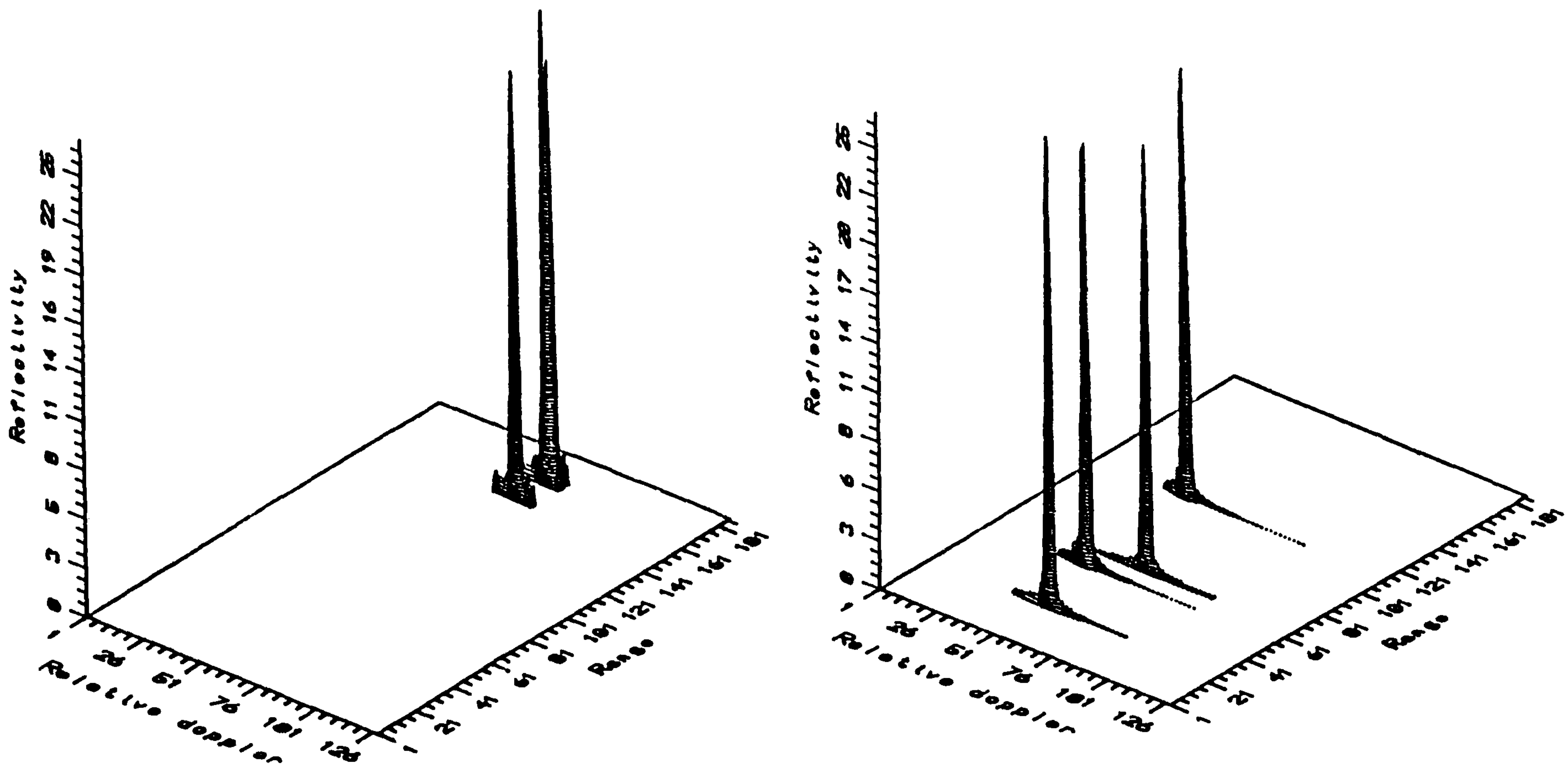
In addition to the list of detected peaks, DOPFP generates the file DOPFPG.GRD which contains the reflectivity data versus relative doppler and range for the |KFPSGN|th footprint. In this file the range index varies from 1 to JRANMX, and the relative doppler index varies from 1 to 128, where 65 corresponds to zero relative doppler (center of range cell).

Figures 10.1a and 10.1b shows a 3–D plot of reflectivity for the 16th and 61th footprints, respectively. Figures 10.2a and 10.2b show the corresponding contour maps.

Table 9.1. Final list of detected scatterers.

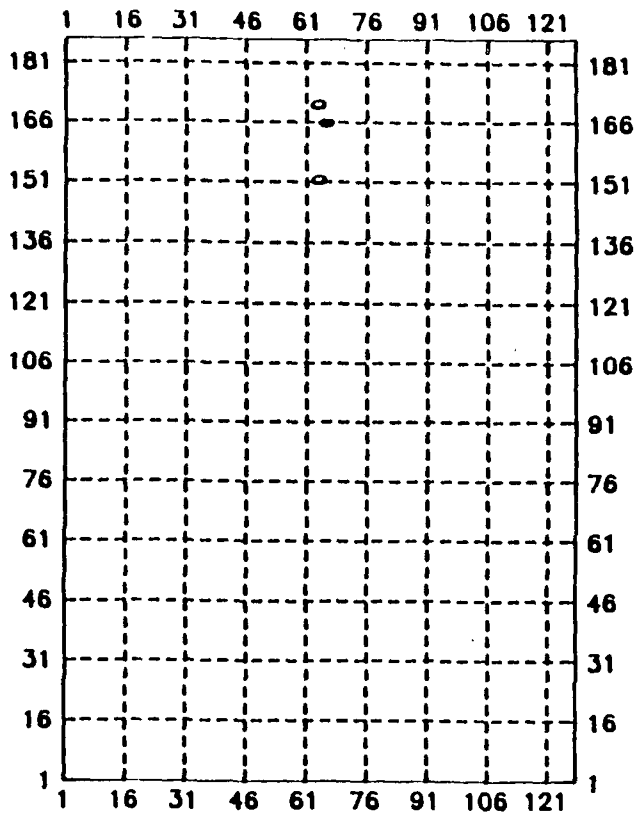
DETFPT+LSTPIK: KPIK,KAD,KFP,JRAN,KNPPP,XM,XSCA,YSCA

DETFPT-!	1	36	54	165	69	27.4051	75.8491	1411.1520
DETFPT-!	2	23	45	111	70	27.3461	-220.8387	1300.8890
DETFPT-!	3	20	42	132	44	27.2710	-355.7807	1276.7160
DETFPT-!	4	42	61	27	54	27.1940	314.5651	1250.4850
DETFPT-!	5	61	83	71	71	27.1860	-60.5324	1430.6290
DETFPT-!	6	30	50	35	65	26.8870	-49.2655	1262.1770
DETFPT-!	7	54	67	96	91	26.8115	505.1594	1274.2760
DETFPT-!	8	26	46	124	63	26.7918	-198.6389	1322.5990
DETFPT-!	9	68	92	62	58	26.7390	226.7033	1434.6640
DETFPT-!	10	12	37	47	75	26.6955	-460.5746	1121.3750
DETFPT-!	11	13	37	141	60	26.6356	-511.0810	1205.0560
DETFPT-!	12	46	62	130	68	26.5043	367.7256	1349.3880
DETFPT-!	13	31	50	21	62	26.4431	-63.0861	1246.8710
DETFPT-!	14	1	4	124	97	26.4358	-447.1980	1082.9330
DETFPT-!	15	50	66	112	30	26.3119	515.0131	1283.9090
DETFPT-!	16	14	37	18	70	26.2371	-450.4071	1092.8520
DETFPT-!	17	17	41	124	90	26.0349	-346.6365	1267.0080
DETFPT-!	18	69	101	44	49	25.9598	502.4506	1350.6440
DETFPT-!	19	15	37	170	50	25.9259	-530.0738	1229.0770
DETFPT-!	20	18	41	58	61	25.8600	-349.7366	1194.1560
DETFPT-!	21	64	87	71	63	25.8340	61.5157	1446.3220
DETFPT-!	22	21	42	65	60	25.6760	-321.0694	1214.9520
DETFPT-!	23	5	12	178	66	25.6440	-203.0483	1242.7950
DETFPT-!	24	38	57	84	62	25.3274	193.1102	1324.8800
DETFPT-!	25	65	88	41	68	25.2679	71.6483	1418.4600
DETFPT-!	26	60	78	32	78	24.9477	-228.7078	1347.2560
DETFPT-!	27	51	66	56	101	24.8499	449.6770	1250.8830
DETFPT-!	28	10	23	161	72	24.7814	189.5856	1270.5690
DETFPT-!	29	47	62	111	78	24.6996	353.0634	1333.5400
DETFPT-!	30	39	57	68	60	24.5755	193.2785	1308.0180
DETFPT-!	31	63	84	8	66	24.5306	-42.1248	1369.9460
DETFPT-!	32	44	61	67	63	24.4731	319.0387	1292.5420
DETFPT-!	33	34	53	37	66	24.2831	43.3615	1276.3230
DETFPT-!	34	45	61	57	45	24.1248	331.3734	1277.8450
DETFPT-!	35	33	50	40	62	23.9599	-64.2520	1266.5800
DETFPT-!	36	35	53	61	61	23.7105	69.8309	1299.8780
DETFPT-!	37	28	46	112	70	23.6840	-184.9778	1312.3290
DETFPT-!	38	41	58	84	70	23.1410	215.4423	1324.6640
DETFPT-!	39	53	66	117	62	22.8162	499.1581	1297.4650
DETFPT-!	40	59	74	60	75	22.4777	-371.7148	1320.2830

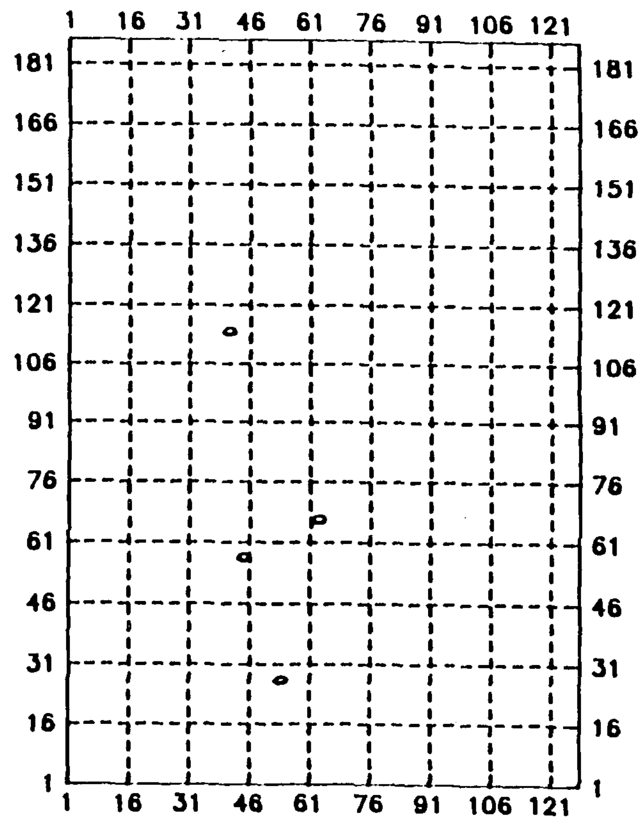


(a) (b)

Figure 10.1. 3-D reflectivity plots versus relative doppler and range:
(a) 16th footprint, (b) 61 st footprint.



(a)



(b)

Figure 10.2. Contour maps of reflectivity versus realtive doppler and range:
(a) 16th footprint, (b) 61 st footprint.

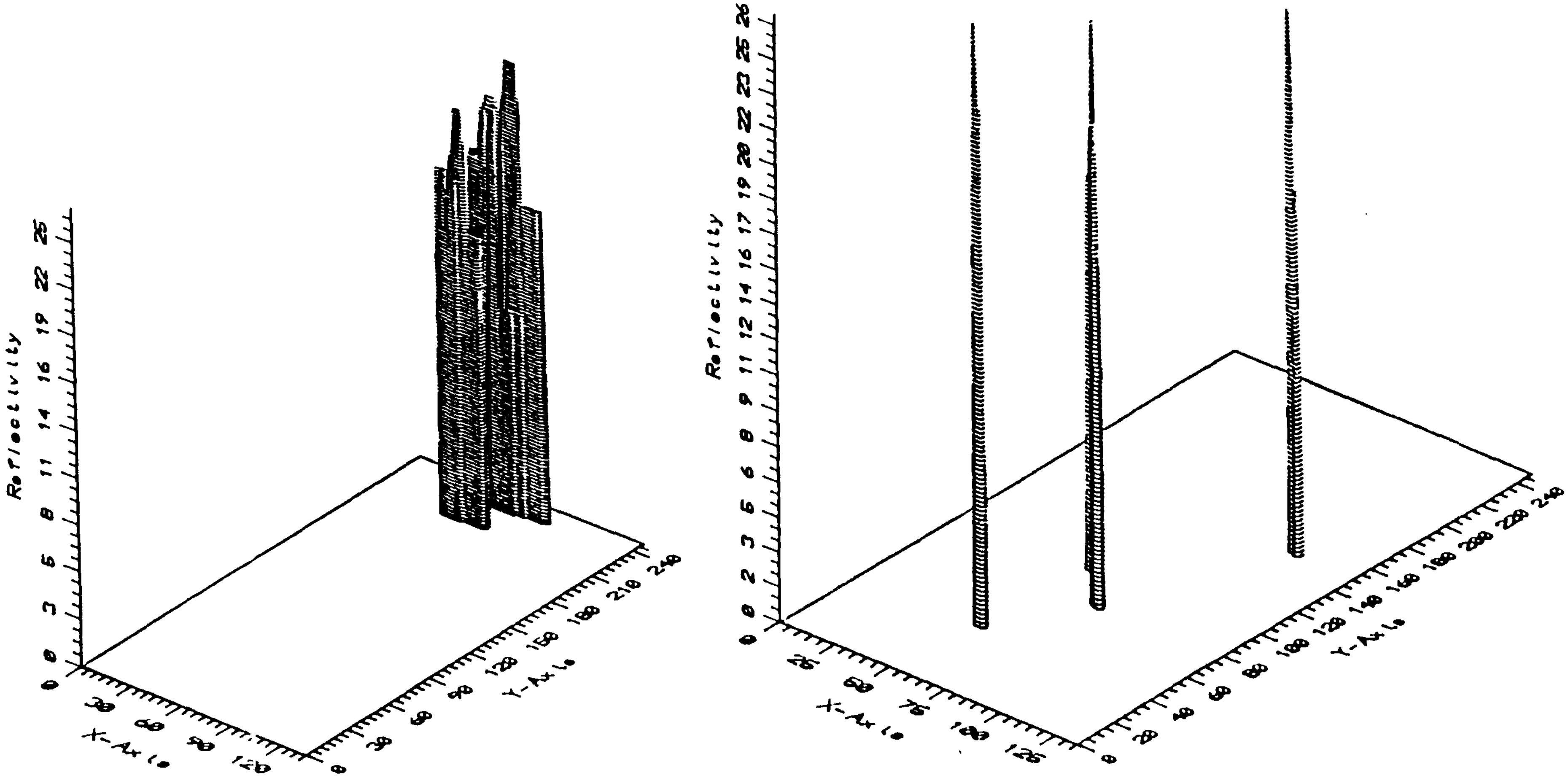
One can observe that impulse invariance has been practically achieved when plotting reflectivity versus doppler and range, even though footprint #16 corresponds to $\alpha^{*} \approx 2.5^{\circ}$. Note however, that the azimuth resolution does depend upon α as indicated in Fig. 6.1.

11. Demonstration of the DOPXY Simulation

The performance of DOPXY is demonstrated by displaying the output x–y reflectivity map for the same input files as in section 10. The HP 7475A plotter was used to obtain hard copies.

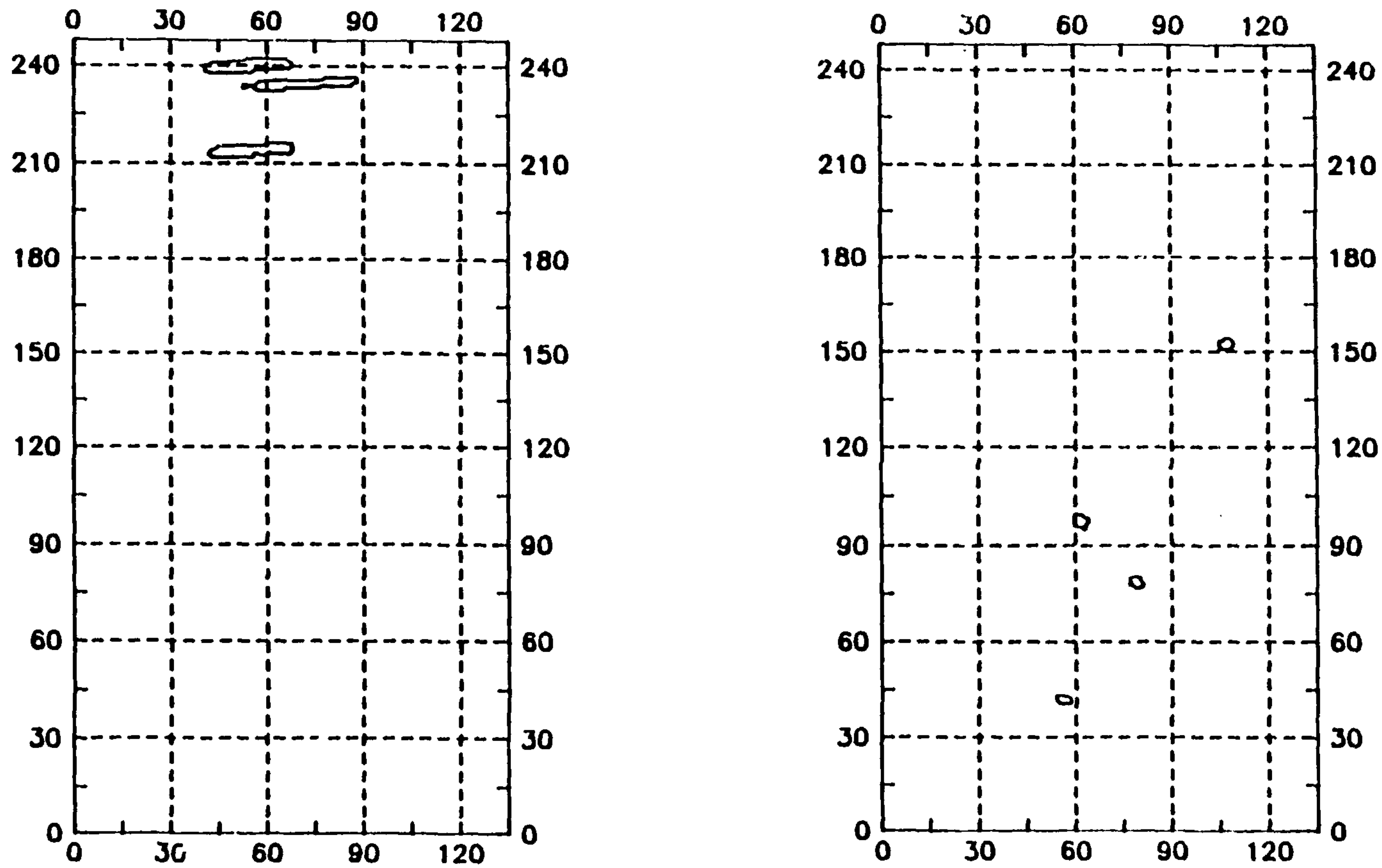
Running DOPXY with a positive value for KFPSGN will generate the file DOPXYG.GRD which contains the reflectivity data versus x–y coordinate for the selected footprint if the x–y window is correctly centered. There are two ways to define the x–y window, a manual setting and an automatic setting. For manual setting enter WINIWR = 0, and define the minimum corner coordinates and the window size in SYPAI.INP. For automatic setting enter WINIWR = 1, then the window is automatically centered on the center of the footprint by subroutine MAPWIN with entry level 4. Automatic setting is recommended because non overlapping window and footprint will produce an empty map. Of course, the reflectivity map will also be empty if there are no scatterers within the selected footprint. The maximum peak within the window is displayed on the screen. If there is one or more targets within the window, the peak should be much greater than zero. A peak equal to zero means tht the window is empty, and the use of SURFER results into a warning "Plannar Grid File". Figures 11.1a and 11.1b show the 3–D reflectivity for the 16th and 61th footprints, respectively. Figures 11.2a and 11.2b show the corresponding contour maps. It is very obvious for both the 3–D plots and contour maps that the azimuth resolution is much better at large α. Yet we are able to detect targets anywhere within the scanned strip.

Running DOPXY with a negative value for KFPSGN will generate the file DOPXYG.GRD which contains the reflectivity data versus absolute x–y coordintes for all the footprints within the x–y window centered on the |KFPSGN|th footprint. Running



(a) (b)

Figure II.1. 3-D reflectivity versus absolute x-y: (a) 16th footprint, (b) 61st footprint.



(a) (b)

Figure II.2. Reflectivity contour maps versus x-y coordinates: (a) 16th footprint, (b) 61st footprint.

DOPXY for KFPSGN = –61 and displaying with SURFER produces Fig. 11.3a and 11.3b. Comparison of Fig. 11.3 with Figs. 11.2a and 11.2b show the appearance of new targets detected in footprints other than 61. Also, the scatterers that are both in 61 and some other footprints have a smoother shape due to the invariant mapping incoherent integration. Running DOPXY for KFPSGN = –16 yields the contour map shown in Fig. 11.4. Comparison to Fig. 11.2a shows again the presence of new detected scatterers and the incoherent integration smoothing effect. As already noted for the single footprint detection, azimuth resolution is poor when α is very small.

We were limited by "DOS" (the execution file is about 550 kByte) to a map of size: x = 102 and y = 187 in DELMAP units. Since DELMAP = 0.75m then the size of the window in meters is x = 76.5m y = 140.25m. A rectangular window was selected to better match the shape of the footprint. Therefore, our approach is to provide an overall low resolution display, and to use DOPXYG.GRD for high resolution display of a selected window. Figure 11.5a shows a low resolution display for the actual clutter map, where the data is from file CLMPI.INP. Figure 11.5b shows a low resolution display for the detected clutter map where the data is as in Table 9.1. One can visualize the very close agreement between the actual and the detected clutter maps. One can also, check that the center of gravity for the high resolution contour maps corresponds closely to the actual scatterer loction.

12. Summary, Conclusions and Recommendations for Future Work

This report extends the application of RRDP and invariant mapping to beam doppler sharpening radar, where the goal is to map a wide area on the ground. The antenna platform follows a horizontal flight path while the antenna scans the specified azimuth interval. Two main programs were developed: DOPFP.FOR and DOPXY.FOR. The first program detects scatterers in one or in all footprints, and also generates a file which contains the reflectivity data versus relative doppler and range for one footprint. The second program includes invariant mapping and generates a file which contains

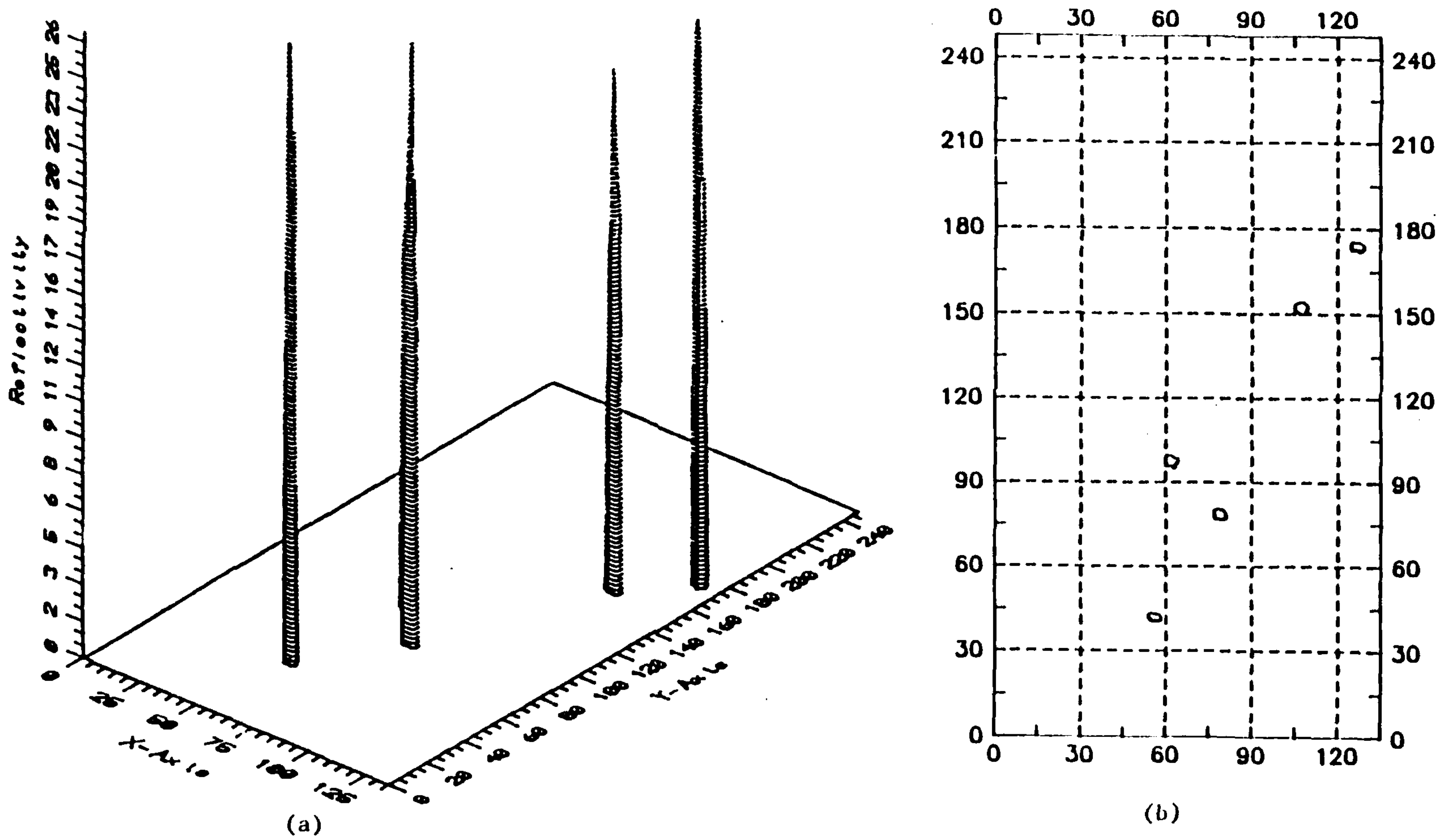


Figure 11.3. Reflectivity versus absolute x-y: (a) 3-D plot, (b) contour map.

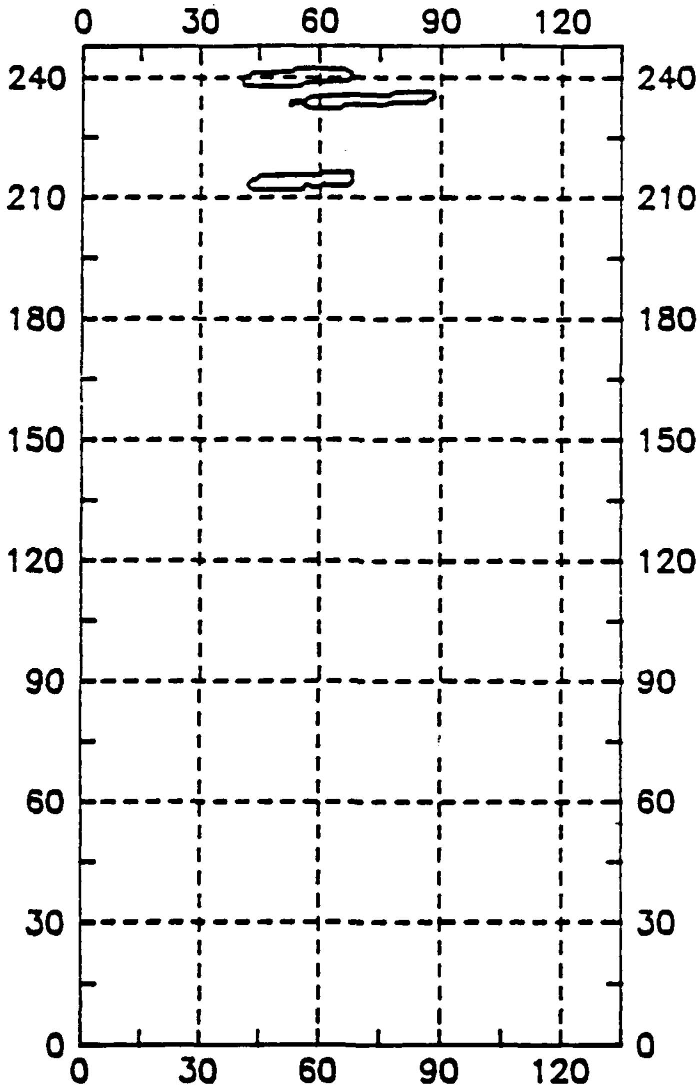


Figure 11.4. Contour plot of the reflectivity versus absolute x-y for the 16th footprint.

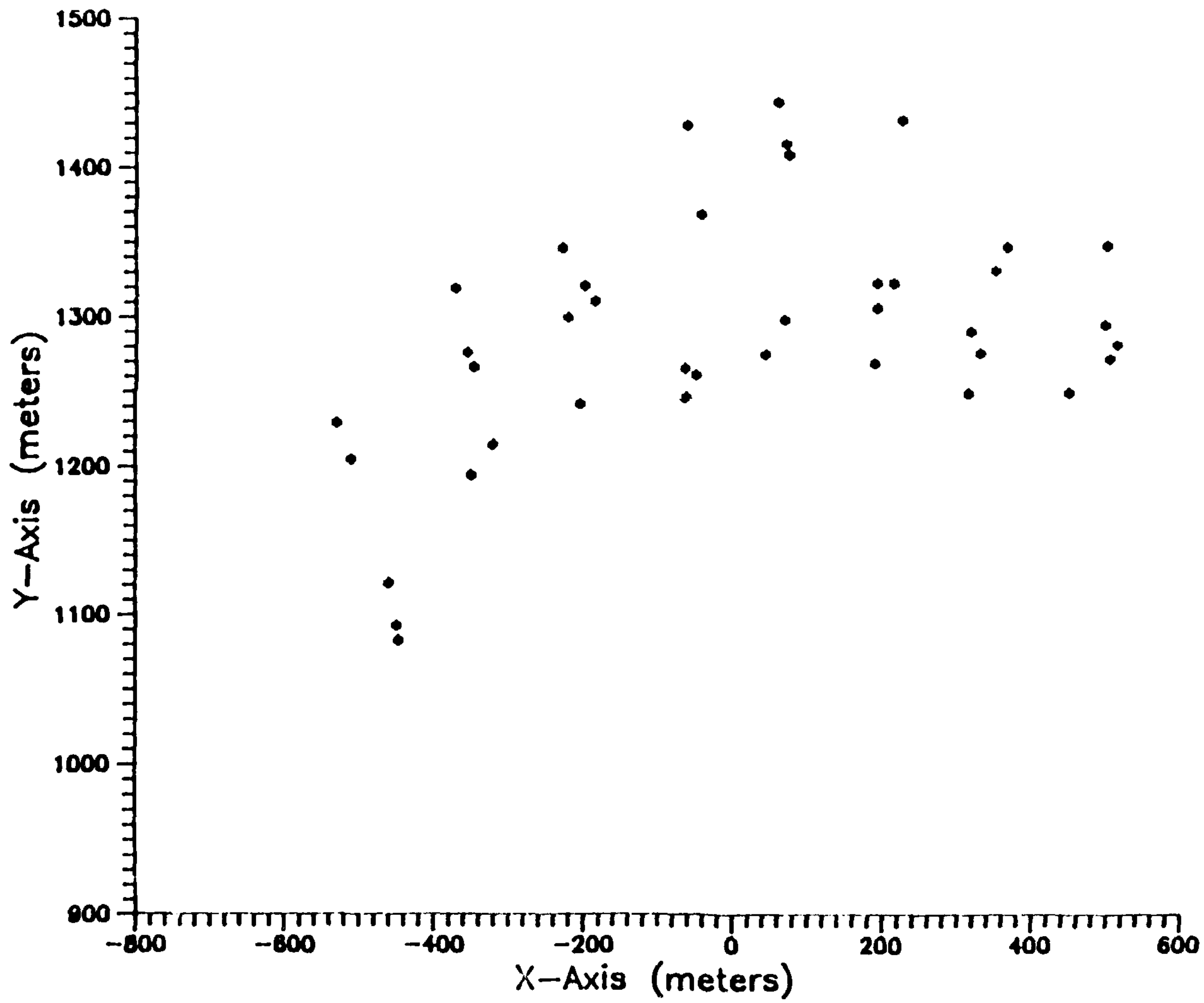


Figure II.5a. Actual x-y coordinates for the 40 scatterers.

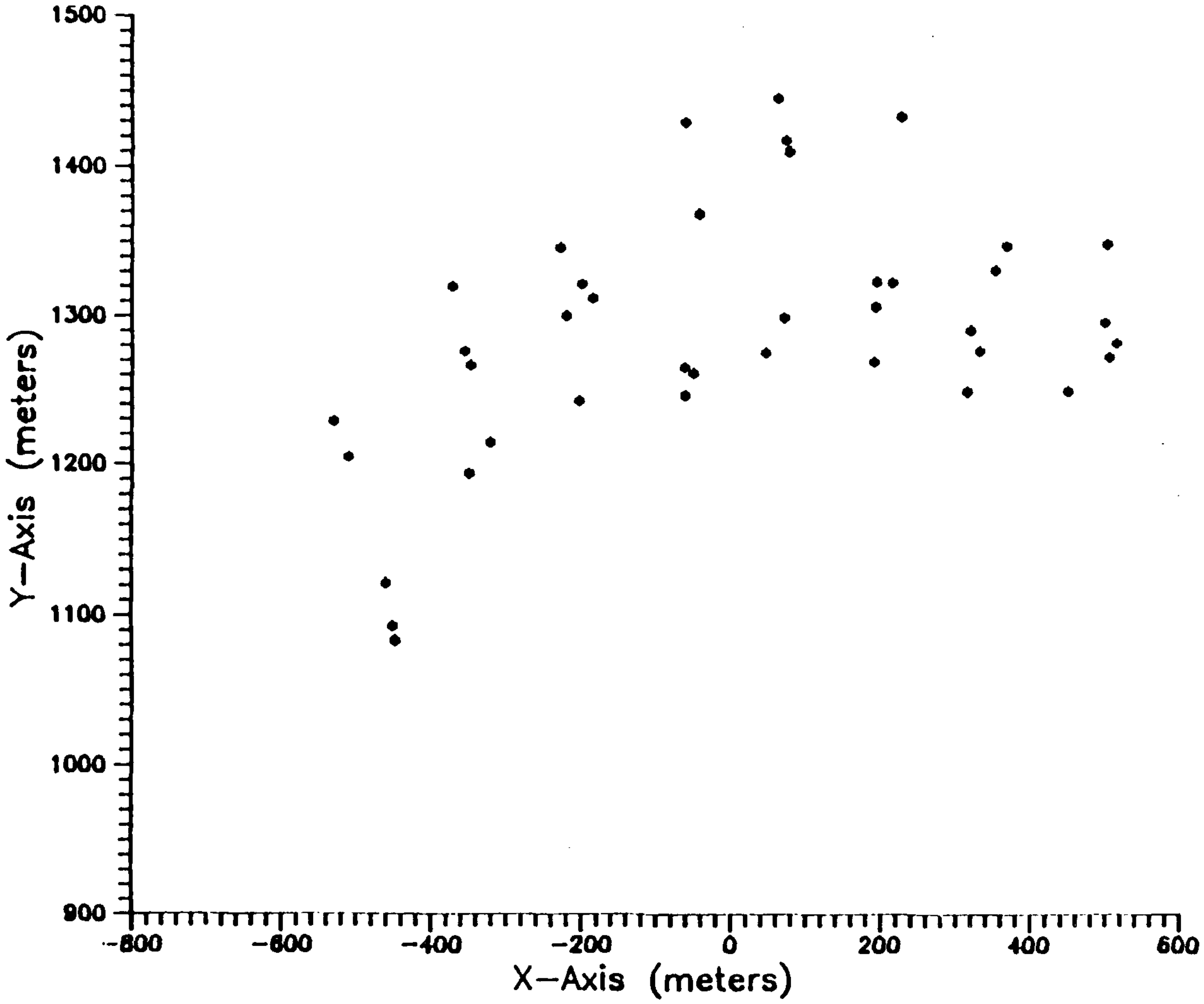


Figure II.5b. Detected x-y coordintes for the 40 scatterers.

reflectivity data versus absolute x–y coordinates for a selectable window. Our examples illustrate the detection of (40) forty scatterers randomly distributed on the ground.

The main contributions of this report are: (1) extending the simulation to a sequence of footprints, (2) making the programs work for positive/negative and very small azimuth, (3) generating a sequence of footprints which guarantees complete ground coverage, (4) developing subroutines to detect peaks, first within a single footprint and then within a sequence of footprints, (5) integrating the information for a sequence of footprints through invariant mapping, (6) implementing an automatic window selection to facilitate the use of the program, and (7) presenting both low resolution and high resolution displays to illustrate the results.

The main conclusions are:

(1) all the (40) forty scatterers where detected at the proper location, with a peak amplitude variation of about ±9%, around the mean,

(2) the range resolution and the relative doppler resolution remain constant,

(3) a formula which expresses the azimuth resolution as a function of the footprint azimuth α^*, and of the elevation β_j for the range cell is presented,

(4) the azimuth resolution is a nonlinear function of the footprint azimuth α^*. In our application it is quite good for $|\alpha^*| > 8^\circ$, but much worse around $|\alpha^*| \approx 2^\circ$,

(5) the invariant mapping technique performs very well for any single footprint within the azimuth coverage. Very little resolution is lost through this mapping.

(6) invariant mapping performs a noncoherent integration over the sequence of footprints as expected.

Recommendations for future work include: (1) to generate a sequence of footprints which takes into account the mechanical limitations of the seeker, (2) to avoid as much as possible small azimuth angles, where the azimuth resolution is much lower (for example, it will be better to scan from (5° to 65°) rather than (-30° to $+30^\circ$)), (3) to extend doppler beam sharpening to nonlinear trajectories, and (4) to extend the doppler sharpening technique to linear arrays for a 3–D scatterer detection.

References

[1] Polge, R. J., Mahafza, B. R., and Kim, J. G., "Computer Simulation for a MM WAve Seeker Including SAR Processing and Invariant Mapping", prepared for Georgia Institute of Technology under subcontract # A–4521–S1 in support of U. S. Army Missile Command Prime Contract #DAAH01–84–DA029, D.O. 0088 UAH Technical Report, May 1987. SECRET.

[2] Polge, R. J., Mahafza, B. R., and Kim, J. G., "Increasing Azimuth and Elevation Resolution of MM Wave Seeker Systems Using Coherent or Noncoherent Range Relative Doppler Processing (RRDP) with Constant or Linear Frequency Modulation and Invariant Mapping", prepared by the University of Alabama in Huntsville for the U. S. Army Missile Command under Contract DAAH01–87–D–0021, D.O., 18 February 1988.

[3] Polge, R. J., Mahafza, B. R., and Kim, J. G., "Extension and Updating of the Computer Simulation of Range Relative Doppler Processing for MM Wave Seekers", prepared by the University of Alabama in Huntsville for the U. S. Army Missile Command under Contract DAAH01–87–D–0021, D. O., January 1989.

[4] Polge. R. J., Mahafza, B. R., and Kim, J. G., "User Manual of the Range RElative Doppler Processing Simulation for MM Wave Seekers", prepared by the University of Alabama in Huntsville for the U. S. Army Missile Command under Contract DAAH01–87–D–0021, D.O., January 1989.

[5] SURFER Reference Manual, by Golden Software Inc., 807 14th Street, P. O. Box 281, Golden, Colorado, 80402.

[6] Rulf, Benjamin and Robertshaw, Gregory A., "Understanding Antennas for Radar, Communications, and Avionics", Van Nostrand, Reinhold, N.Y., 1987.

[7] Balanis, Constantine A., "Antenna Theory Analysis and Design," Harper and Row, N.Y., 1982.

APPENDIX A

Relevant Footprint Information For $\vec{m} = \{d_x, d_y, h)$

A.1 Footprint center $\vec{q}^* = \vec{q}(t_c)$, see Fig. A.1.

Rectangular coordinates of footprint center $= \{x_q, y_q, 0\}$

range: $r^* = \sqrt{}(h^2 + (x_q - d_x)^2 + (y_q - d_y)^2)$

azimuth angle: $\alpha^* = \tan^{-1}[(x_q - d_x)/(y_q - d_y)]$ (A.1)

elevation angle $= \beta^* = \cos^{-1}\left[\frac{h}{r^*}\right]$

A.2 Range bounds and indexing, see Fig. A.1.

Minimum range: $\text{Ranmn} = \dfrac{h}{\cos(\beta^* - \theta_H)}$

Maximum range: $\text{Ranmx} = \dfrac{h}{\cos(\beta^* + \theta_H)}$ (A.2)

θ_H = half beamwidth

Range index for $\vec{q}^*$: $j^* = \text{INT}[(r^* - \text{Ranmn} + \Delta r/2)/\Delta r]$

Maximum Range index: $\text{JRANMX} = \text{INT}((\text{Ranmx} - \text{Ranmn} + \frac{\Delta r}{2})/\Delta r)$

A.3 Information for the jth range cell

median range: $r_j^* = r^* + (j - j^*)\ \Delta r$

elevation angle: $\beta_j^* = \cos^{-1}\left[\frac{h}{r_j^*}\right]$ (A.3)

maximum relative elevation (see Fig. A.2): $\epsilon mx = \beta j^* - \cos^{-1}\left[\dfrac{h}{r_j^* - \Delta r/2}\right]$

maximum relative azimuth (see Fig. A.3): $\mu mx = \cos^{-1}\left[\dfrac{\cos\theta_H - \cos\beta_j^* \cos\beta^*}{\sin\beta_j^* \sin\beta^*}\right]$

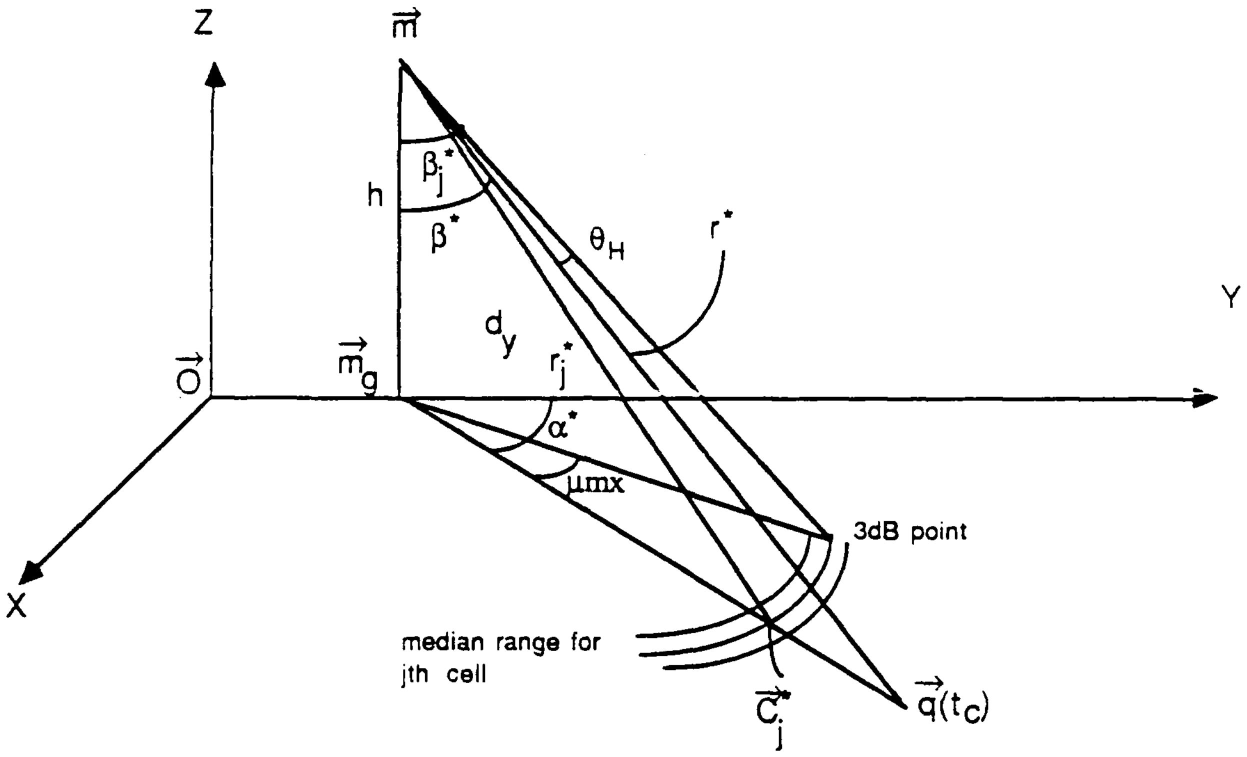


Figure A.1 Footprint center,range bounds,and jth range cell.

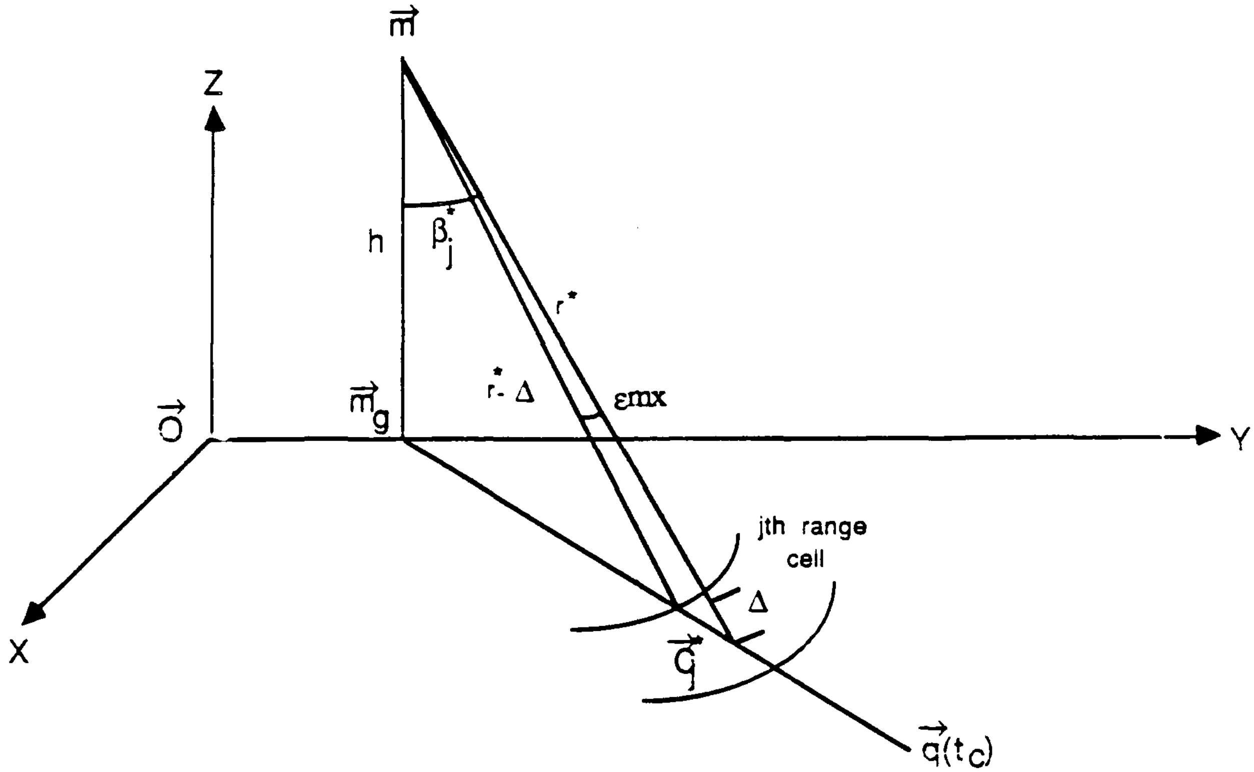


Figure A.2 Maximum relative elevation εmx ($\Delta = \Delta r/2$).

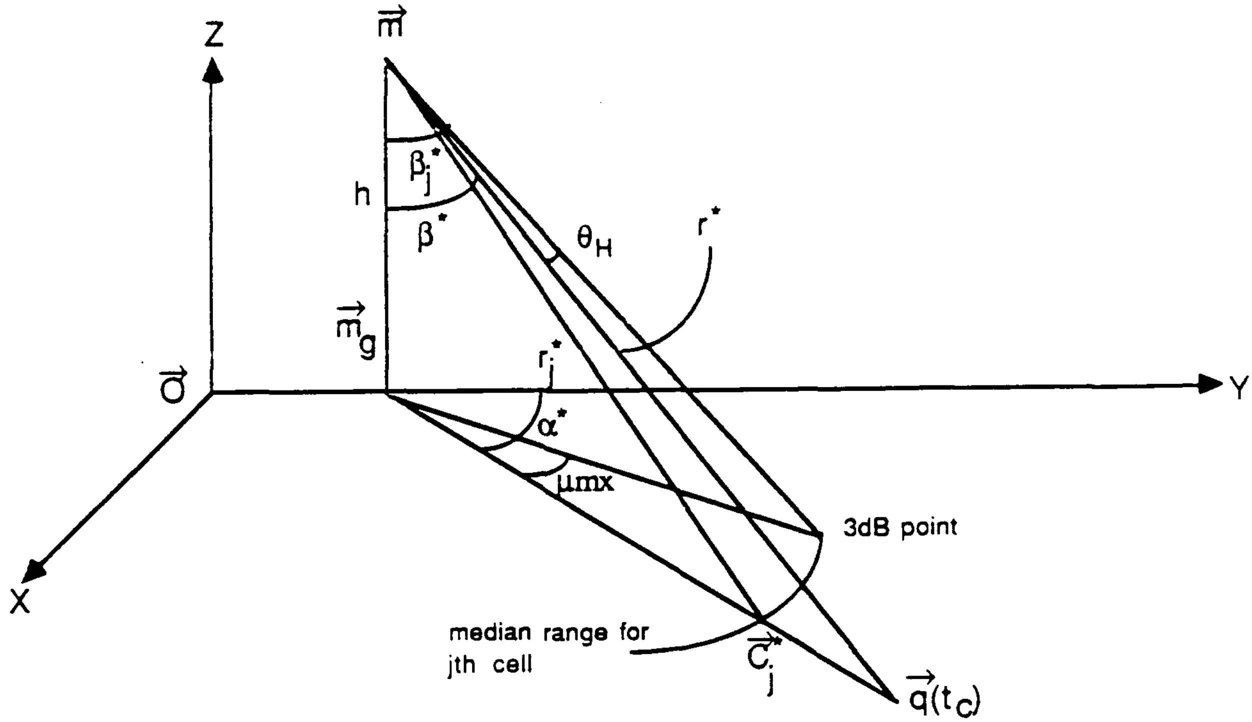


Figure A.3 Ma::imum relative azimuth angle $\mu mx, \theta_H$ =half beamwidth.

APPENDIX B

Relevant Information for a Scatterer $\vec{C}_i$ (see Fig. B.1).

B.1 Computation of $\{r_{gi}, r_i, \alpha_i, \beta_i, \mu_i\}$ from $\{\vec{m}, \vec{C}_i, \alpha^*\}$

$(\vec{m}_g, \vec{C}_{gi})$ are the vertical projections of $(\vec{m}, \vec{C}_i)$

ground range: $GR = |\vec{C}_{gi} - \vec{m}_g| = \sqrt{\quad} [(x_i - d_x)^2 + (y_i - d_y)^2]$

range to $\vec{C}_{gi} = r_{gi} = |\vec{C}_{gi} - \vec{m}| = \sqrt{\quad} [GR^2 + h^2\}$

slant range to $\vec{C}_i$: $r_i = |\vec{C}_i - \vec{m}| = \sqrt{\quad} [GR^2 + (h - h_i)^2]$

range cell index: $j = INT[(r_i - r^* + \Delta r/2)/\Delta r] + j^*$ (B.1)

elevation angle: $\beta_i = \tan^{-1}\left[\frac{GR}{h}\right]$

azimuth angle: $\alpha_i = \tan^{-1}[(x_i - d_x)/(y_i - d_y)]$

relative azimuth increment: $\mu_i = \alpha_i - \alpha^*$

B.2 Antenna Gain for $\vec{C}_i$ at time t.

Antenna angle to $\vec{C}_i$: Ang= Angle$(\vec{q}^* - \vec{a}(t), \vec{C}_i - \vec{a}(t))$

let $w1 = \{a_x(t)\ ,\ a_y(t)\ ,\ a_z(t)\}$

$w2 = \{q_x^*, q_y^*, q_z^*\}$

and $w3 = \{C_{ix}, C_{iy}, C_{iz}\}$

$s = \sin(Ang) = \sqrt{\quad}$ (1 –DOTCOS(w1,w2,w3)**2) (B.2a)

where the function DOTCOS is listed in Appendix E.

One–way Antenna gain = DISH(s,2,DOVL) (B.2b)

where the function DISH is listed in Appendix C, and DOVL is the dish diameter in wavelength units.

The gain is computed at three times

$$\{t = -\frac{D_{ob}}{2}\ ,\ t = 0\ ,\ t = \frac{D_{ob}}{2}\} \qquad (B.2c)$$

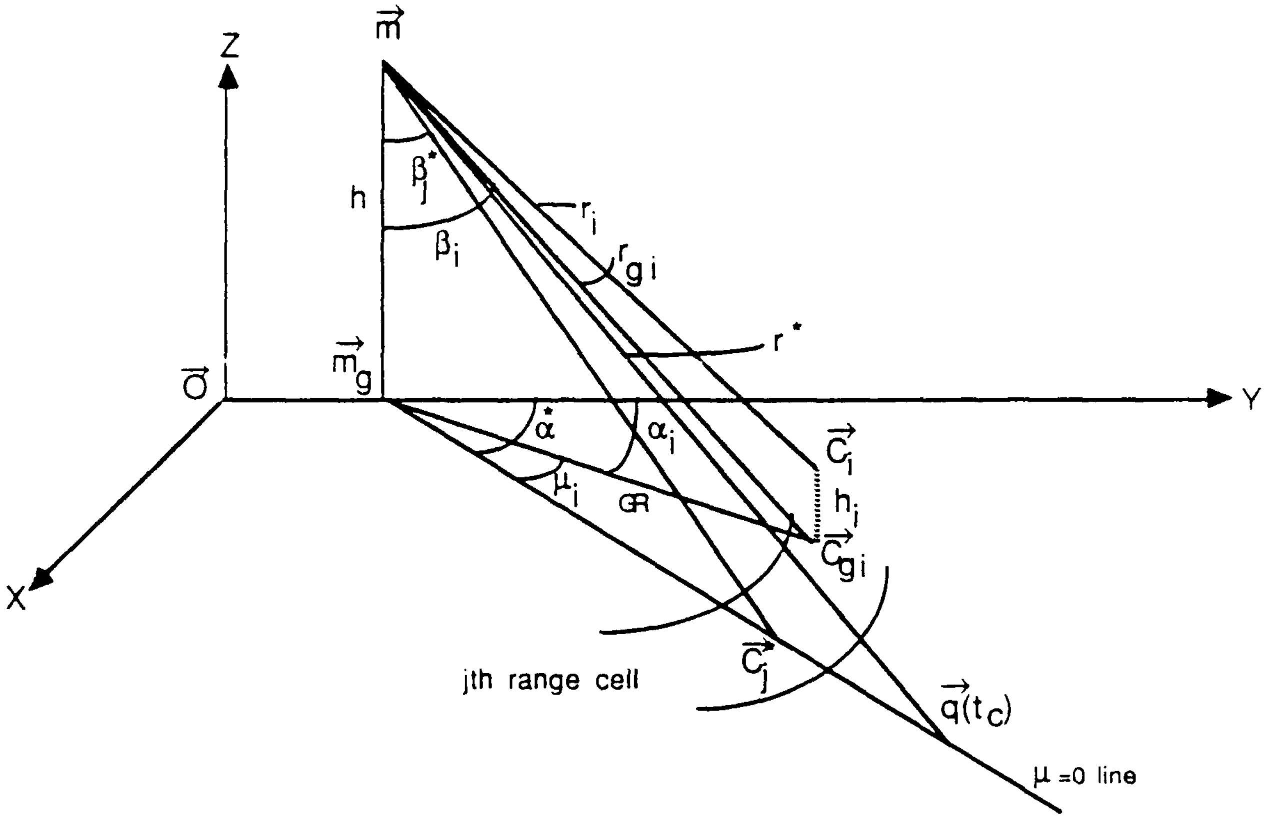


Figure B.1 Fictitious scatterer $\vec{C}_j^*$ and ith scatterer $\vec{C}_i$ in the jth range cell: $\alpha_i = \alpha^* + \mu_i$.

Appendix C

Listings for DOPFP.FOR and Associated Subroutines

Program DOPFP.FOR

```
$LARGE
   CHARACTER*240 TITLE
   REAL W(23),TIMCNT,TIMDWL,DELRAN,XMMNPR,WPAR(8),XIFXMT,SELWIN,
  &DETIPR,SELDIS,BITINT,WSQ(12),SHOSCA,BUGLEV
   REAL SMOMAP(450),TXD(64),TXQ(64),WANGN9(265),WAGNCL(128),
  &WBET9(265),WDUM(3),WDOVL(3),WEPSMX(265),WIND(64)
   REAL VELOM,ALFM,BETM,WMPOSB(3),WMPOSC(6),WMPOSE(3),WQPOSB(3),
  &WQPOSC(6),WQPOSE(3),WFREQ(128),WMVELO(6)
   REAL WMU3DB(265),WMUMDR(266),WRAN9(265),WTAU(25),WTIME(64),
  &WTSMO(5),XD(128),XM(128),XQ(128),WADRLO(265),WADRUP(265),
  &WADSMO(90),WAGAIN(3),HEIREF,BETREF,RANREF
   INTEGER LNPPP(128),NBSCA,KFPDIS,NW,LEVBUG,NBFP,NBSCIN,JRSCMN,
  &JRSCMX,IWZMX,JWZMX,KFPSGN,ISURF,IPRDET,NBINCU,NBINOL,NPINFP
   REAL DISH,DOTCOS,FRQOFMU,FREQ,COSANT,GRDMAP,TMU3DB,WINIWR,
  &FPMXNB,RELDOP,BET,SURFIN
   REAL CLIGHT,DELFIN,DELFUT,DELTIM,DOVL,EPS,FMU,FREQO,GDISH2,
  &GDISH3,GRC,HDELRA,HTHETA,HTIMDW,RANCOF,OMEGO,SCAFAC,
  &SINANG,THETA,TMPBET,TPI,V1,V2,V3,X,XSCA,YSCA,Y,Z,FZMX,SCANEW
   INTEGER IDDISH,IFILE,INDX,INTCOF,IW,JRAN,JRANQ9,JRANMX,JW,K,
  &KADSMO,KDQ,KI,KJ,KFF,KNPPP,KSCALO,KSCAUP,KT,KTIM,KTHSCA,
  &NBITFF,NBITFX,NFFT,NFFTH,NFFX,NFFXH,NPPPMN,NPPPMX,NSCAFP,
  &NBFPMX,KFP
   EXTERNAL YMU3DB,FRTOMU
   INTEGER MNIW,MNJW,MXIW,MXJW,NBSCDT,IWFPT,IWRAN,IWFRQ,JWFPT,
  &JWRAN,JWFRQ
   REAL ZMX,SYNMAP,WSCDTX,WSCDTY,WSCDTM,VX,VY,VM
   COMMON /DETECT/NBSCDT,WSCDTX(400),WSCDTY(400),WSCDTM(400),
  &IWFPT(400),IWRAN(400),IWFRQ(400),NBINCU,VX(129),VY(129)
  &,VM(129),JWRAN(129),JWFRQ(129),JWFPT(129),NBINOL
   COMMON /SURFP/SYNMAP(128,200),MNIW,MNJW,MXIW,MXJW,ZMX,NPINFP
   EQUIVALENCE (W(1),TIMDWL),(W(2),DELRAN),(W(3),XMMNPR),(W(4),
  &WPAR(1)),(W(12),XIFXMT),(W(13),SELWIN),(W(14),DETIPR),(W(15),
  &SELDIS),(W(16),WINIWR),(W(17),SCANEW),(W(18),HEIREF),
  &(W(19),BETREF),(W(20),SHOSCA),(W(21),BUGLEV),(W(22),FPMXNB),
  &(W(23),SURFIN)
C   OUTPUT PARAMETERS IN WPAR=(DELMAP,THRESH,XMNMAP,YMNMAP
C   ,XMXMAP,YMXMAP,XSIZE,YSIZE) ****************************************
C    WMVELO=(XVELO,YVELO,ZVELO,VELOM,{2/WAVLNGH},DELFUT) **************
C    WMPOSC=( XM, YM, ZM, HTHETA, RANMN, RANMX)*************************
C    WQPOSC=(XFP=XQ9,YFP=YQ9,ZFP=ZQ9,ALFQ9,BETQ9,RANQ9) ***************
C    DIMENSION OF SMOMAP = NBSCA*15 (SEE SUB. CLUFPT) *****************
   FRQOFMU(V1,FMU,V2,V3)=-V1*(V2*FMU+(V3/2.)*FMU**2)
C    @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C    I/O  FILES : U9=SYPAI.INP(main);U3=DOPFPP.PRT(main);U11=CLMPI.INP
```

```
C     (clumap);U10=FPSQI.INP(made in fpidop,used in bcedop);U12=GEODOPI.
C    &INP(fpidop);U7=DOPFPG.GRD(main) @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C     HALF 3db BEAMWIDTH=.0215 gives DOVL=14.83343168=DISH DIAMETER IN
C    &LAMBDA UNITS ( 2-WAY GAIN = -3.00 DB means DISH=0.84139514)*******
      DATA THETA,FREQO,CLIGHT,TPI,EPS,WDOVL,NW,BITINT/.043,94.E9,3.E8,
     &6.28318531,.0001,17.15094,17.15094,14.80884,23,1./
      DATA WQPOSB(3),WQPOSC(3),WQPOSE(3)/0.,0.,0./
      NBSCDT=0
      SINANG=SIN(0.0215)
      GDISH2=DISH(SINANG,2,WDOVL(2))
      GDISH3=DISH(SINANG,3,WDOVL(3))
      OPEN(UNIT=3,FILE='DOPFPP.PRT',STATUS='OLD')
      WRITE(3,*)'GDISH2,GDISH3',GDISH2,GDISH3
      PRINT*,'WPAR not used in DOPFP except for WPAR(2)=SURF-threshold'
      TITLE='TIMDWL=1;DELRAN=2;XMMNPR=3;WPAR1=4=DELMAP;WPAR2=THRESH.WPAR
     &3=XMN;WPAR4=YMN;WPAR5=XMX/WPAR6=YMX;WPAR7=XSIZE;WPAR8=YSIZE:XIFX
     &=12;SELWIN=13;DETIPR=14;SELDIS=15;WINIWR=16/SCANEW=17;HEIREF=18;BE
     &TREF=19;SHOSCA=20;BUGLEV=21;FPMXNB=22;SURFIN=23'
      OPEN(UNIT=9,FILE='SYPAI.INP',STATUS='OLD')
      CALL WINPUX(NW,W,9,TITLE)
      IDDISH=INT(SELDIS+.00001)
      LEVBUG=INT(BUGLEV+.00001)
      ISURF=INT(SURFIN+EPS)
      IPRDET=INT(DETIPR+EPS)
      NBFPMX=INT(FPMXNB+EPS)
      DOVL=WDOVL(IDDISH)
      OMEGO=TPI*FREQO
      HDELRA=DELRAN*0.5
      HTHETA=THETA*0.5
      WMPOSC(4)=HTHETA
      NBITFF=6
      NBITFX=NBITFF+INT(BITINT+EPS)
      NFFT=2**NBITFF
      NFFTH=NFFT/2
      NFFX=2**NBITFX
      NFFXH=NFFX/2
      DELFIN=1./TIMDWL
      INTCOF=2**INT(BITINT+EPS)
      DELFUT=DELFIN/FLOAT(INTCOF)
      WMVELO(6)=DELFUT
      DELTIM=TIMDWL/NFFT
      HTIMDW=TIMDWL*.5
C     COMPUTE RANGE REFERENCE TO USE IN RANCOF ************************
      RANREF=HEIREF/COS(BETREF)
      CALL CLUMAP(HTHETA,LEVBUG)
      OPEN(UNIT=10,FILE='FPSQI.INP',STATUS='OLD')
      PRINT*,'**********************************************************'
      PRINT*,'SELECT FP TO DISPLAY: KFPSGN, ABORT=-999; ALL FP=-KFPSGN'
      CALL KINPUT(KFPSGN)
      IF(KFPSGN.EQ.-999) STOP 'ABORT'
      KFPDIS=IABS(KFPSGN)
      CALL FPIDOP(NBFPMX,NBFP,HTHETA,TIMDWL,KFPDIS)
      PRINT*,'**** KFPDIS=; NBFPMX=; NBFP=;',KFPDIS,NBFPMX,NBFP, '*****'
```

```
C   COMPUTE ARRAY OF RELATIVE TIME WTIME *****************************
    DO 2 KTIM=1,NFFT
 2  WTIME(KTIM)=DELTIM*(KTIM-NFFTH-0.5)
C   INDEX KNPPP=1,NFFX FOR FREQ=[-NFFXH*DELFUT,(NFFXH-1)*DELFUT] *****
C   COMPUTE ARRAY LNPPP,THEN KFF=LNPPP(KNPPP) & WFREQ(KNPPP) *********
    DO 9 KNPPP=1,NFFX
    WFREQ(KNPPP)=DELFUT*FLOAT(KNPPP-NFFXH-1)/FREQO
    INDX=KNPPP+NFFXH
    IF(INDX.GT.NFFX) INDX=INDX-NFFX
 9  LNPPP(KNPPP)=INDX
    CALL WINDOWS(NFFT,4,3.14159,WIND)
    IF(ISURF.EQ.1) CALL SURFMP(KFF,JRAN,Z,JRANMX,1)
    DO 1 KFP=1,NBFP
    CALL BCEDOP(W,NFFT,WTIME,VELOM,ALFM,BETM,WMPOSB,WMPOSC,WMPOSE,
   &WQPOSB,WQPOSC,WQPOSE,WMVELO,JRANQ9,JRANMX,KFPDIS,KFP)
    IF(KFP.NE.KFPDIS.AND.KFPSGN.GT.0) GO TO 1
    WRITE(*,*)'VELOM,  ALFM,  BETM,  XM,  YM,  ZM,  XFP,  YFP'
    WRITE(*,*) VELOM,ALFM,BETM,(WMPOSC(K),K=1,3),(WQPOSC(IW),IW=1,2)
    WMVELO(5)=2.*FREQO/CLIGHT
C   STORE RANGE,BET,EPSMX,AND MU-3DB FOR FICTITIOUS SCATTERERS *******
    DO 7 JRAN=5,JRANMX-9
    WRAN9(JRAN)=WQPOSC(6)+(JRAN-JRANQ9)*DELRAN
    WBET9(JRAN)=ACOS(WMPOSC(3)/WRAN9(JRAN))
    TMPBET=ACOS(WMPOSC(3)/(WRAN9(JRAN)-HDELRA))
    WEPSMX(JRAN)=WBET9(JRAN)-TMPBET
 7  WMU3DB(JRAN)=YMU3DB(HTHETA,WQPOSC(5),WBET9(JRAN),0)
C   ENTER CLUTTER INFORMATION FOR FOOTPRINT AT WQPOSC & MAP PARAMETERS
    CALL CLUFPT(DELRAN,JRANQ9,JRANMX,WQPOSC,WMPOSC,NSCAFP,SMOMAP,
   &WADRLO,WADRUP,WADSMO)
C   PROCESS RANGE CELLS SEQUENTIALLY AND COMPUTE REFLECTIVITY MAP ****
    IF(IPRDET.EQ.1) PRINT*,'DOPFP-DETFPT: KFP,  JRAN,  KNPPP,  XM,  XS
   &CA,  YSCA '
    DO 35 JRAN=5,JRANMX-9
    KSCALO=WADRLO(JRAN)+EPS
    IF(KSCALO.EQ.0) GO TO 35
C   INITIALIZING QUADRATURE ARRAYS=(XD,XQ) ***************************
    DO 11 K=1,NFFX
    XD(K)=0.
11  XQ(K)=0.
    WDUM(3)=0.
C   COMPUTE TAYLOR COEFFICIENTS WTAU=(TAU,TAUt,TAUe,TAUtt,TAUee,TAUtm,
C   TAUte,TAUttt,TAUeee,TAUtmm,TAUtee,TAUtme,TAUttm,TAUtte) **********
    CALL TAYLOR(VELOM,BETM,ALFM,WQPOSC(4),WQPOSC(5),WMPOSC(3),HTHETA,
   &DELRAN,JRAN,WTAU)
    WRITE(3,*)'ARRAY WTAU'
    CALL PRTXY(WTAU,1,25,119,3)
C   COMPUTE 3DB FOOTPRINT LIMITS :(NPPPMN,NPPPMX) ********************
C   TO LIMIT # CELLS BEING PROCESSED TO KNPPP=NPPPMN,NPPPMX **********
    TMU3DB=SIGN(WMU3DB(JRAN),WQPOSC(4))
    NPPPMN=FRQOFMU(FREQO,TMU3DB,WTAU(6),WTAU(10))/DELFUT+NFFXH-5.
    NPPPMX=FRQOFMU(FREQO,-TMU3DB,WTAU(6),WTAU(10))/DELFUT+NFFXH+5.
    IF(ABS(WQPOSC(4)).LT.WMU3DB(JRAN)) NPPPMX=FRQOFMU(FREQO,-WQPOSC(4)
```

```
     &,WTAU(6),WTAU(10))/DELFUT+NFFXH+5.
C    COMPUTE MU AT CENTER OF CELLS FOR INCREASING DOPPLER *************
     DO 17 K=NPPPMN-1,NPPPMX+1
17   WMUMDR(K)=SIGN(1.,WQPOSC(4))*FRTOMU(WTAU,WFREQ(K))
C    COEFFICIENT FOR RANGE NORMALIZATION: RANCOF **********************
     RANCOF=(WRAN9(JRAN)/RANREF)**4
     KSCAUP=WADRUP(JRAN)+EPS
     DO 30 KTHSCA=KSCALO,KSCAUP
     KADSMO=WADSMO(KTHSCA)+EPS
C    INFORMATION ON SCATTERER,WTSMO=(AMPSCA,FASSCA,MUSCA,JRAN-SCA,ZSCA)
     DO 31 KT=1,5
31   WTSMO(KT)=SMOMAP(KADSMO-4+KT)
C    ANTGAIN(t)=WAGAIN(1)+WAGAIN(2)*t+WAGAIN(3)*t^2 *******************
C    WAGAIN IS OUTPUT OF SCAGAN,INPUTS ARE DOVL,HTIMDW,WMPOS=(BEG,
C    &CENT,END),WQPOSC,WTSMO=(AMP,FAS,MU,JRAN,ZSCA),JRANQ9,DELRAN ******
     CALL SCAGAN(IDDISH,DOVL,HTIMDW,WMPOSB,WMPOSC,WMPOSE,WQPOSB,WQPOSC,
    &WQPOSE,WTSMO,JRANQ9,DELRAN,WAGAIN)
     CALL TSIGNAL(OMEGO,NFFT,RANCOF,XIFXMT,WTSMO,WTAU,WTIME,WAGAIN,TXD,
    &TXQ)
     DO 33 KDQ=1,NFFT
     XD(KDQ)=XD(KDQ)+TXD(KDQ)
33   XQ(KDQ)=XQ(KDQ)+TXQ(KDQ)
30   CONTINUE
     CALL DERAMP(OMEGO,WTAU(4),NFFT,XD,XQ,WTIME)
     DO 37 K=1,NFFT
     XD(K)=XD(K)*WIND(K)
37   XQ(K)=XQ(K)*WIND(K)
     CALL FFT(-NBITFX,1,1,1,XD,XQ)
C    COMPUTE ARRAY OF ANTENNA GAIN AT CENTER OF CELLS *****************
     DO 39 KNPPP=NPPPMN,NPPPMX
     SINANG=DSQRT(1.+EPS-(DSIN(WQPOSC(5))*DSIN(WBET9(JRAN))*DCOS(WMUMDR
    &(KNPPP))+DCOS(WQPOSC(5))*DCOS(WBET9(JRAN)))**2)
39   WAGNCL(KNPPP)=DISH(SINANG,IDDISH,DOVL)**2
C    (XD,XQ) NEED TO BE SCALED FOR RANGE AND ANTENNA GAIN AT KNPPP ****
     DO 41 KNPPP=NPPPMN,NPPPMX
     KFF=LNPPP(KNPPP)
     SCAFAC=1./(RANCOF*WAGNCL(KNPPP))
     XD(KFF)=XD(KFF)*SCAFAC
     XQ(KFF)=XQ(KFF)*SCAFAC
     XM(KFF)=SQRT(XD(KFF)**2+XQ(KFF)**2)
     IF(XM(KFF).GE.XMMNPR) CALL DETFPT(KFP,JRAN,KNPPP,WMUMDR(KNPPP),
    &XM(KFF),JRANQ9,DELRAN,WMPOSC,WQPOSC,1,IPRDET)
     IF(ISURF.EQ.1.AND.KFPDIS.EQ.KFP.AND.XM(KFF).GE.WPAR(2)) CALL
    &SURFMP(KNPPP,JRAN,XM(KFF),JRANMX,2)
41   CONTINUE
35   CONTINUE
     IF(NBINCU.GT.0) CALL DETFPT(KFP,JRAN,KNPPP,WDUM(1),WDUM(2),
    &JRANQ9,DELRAN,WMPOSC,WQPOSC,2,0)
1    CONTINUE
     CALL DETFPT(KFP,JRAN,KNPPP,WDUM(1),WDUM(2),JRANQ9,DELRAN,
    &WMPOSC,WQPOSC,3,0)
     CLOSE (3)
```

```
      CLOSE (10)
      PRINT*,'SURFMP:KFP,NPINFP,ZMX ',KFPDIS,NPINFP,ZMX
      IF(ISURF.EQ.1) CALL SURFMP(KFF,JRAN,Z,JRANMX,3)
      STOP
      END
```

Subroutine BCEDOP.FOR

```
      SUBROUTINE BCEDOP(W,NFFT,WTIME,VELOM,ALFM,BETM,WMPOSB,WMPOSC,
     &WMPOSE,WQPOSB,WQPOSC,WQPOSE,WMVELO,JRANQ9,JRANMX,KFPDIS,KFP)
      INTEGER K,K1,K2,JRANQ9,JRANMX,KFPDIS,KFP,NFFT
      REAL W(1),VELOM,ALFM,BETM,WMPOSB(1),WMPOSC(1),WMPOSE(1),WQPOSB(1),
     &WQPOSC(1),WQPOSE(1),WMVELO(1),WTIME(1),GRC,RANMN,RANMX,ALFQ9,RANQ9
C     WQPOSC=(XFP,YFP,ZFP,ALFQ9,BETQ9,RANQ9) ****************************
C     UPDATING FOOTPRINT INFO. BY READING FILE FPSQI.INP ***************
      IF(KFP.EQ.KFPDIS.AND.KFPDIS.NE.999)PRINT*,'@@@ SUB-BCEDOP reads KF
     &PDIS=',KFPDIS,' row of FPSQI.INP & writes WMPOS(B,C,E),WQPOS(B,C,
     &E),WMVELO(x,y,z) @@@'
      IF(KFPDIS.EQ.999) PRINT*,'@@@  SUB-BCEDOP IN FP=  ',KFP
      READ(10,18) VELOM,ALFM,BETM,(WMPOSC(K1),K1=1,3),(WQPOSC(K2),
     &K2=1,2),ALFQ9
C     COMPUTE MISSILE INFORMATION ***************************************
      WMVELO(1)=VELOM*SIN(BETM)*SIN(ALFM)
      WMVELO(2)=VELOM*SIN(BETM)*COS(ALFM)
      WMVELO(3)=-VELOM*COS(BETM)
      WMVELO(4)=VELOM
      DO 21 K=1,3
      WMPOSB(K)=WMPOSC(K)+WTIME(1)*WMVELO(K)
   21 WMPOSE(K)=WMPOSC(K)+WTIME(NFFT)*WMVELO(K)
C     COMPUTE FOOTPRINT INFORMATION AT WQPOSC **************************
      GRC=SQRT((WQPOSC(1)-WMPOSC(1))**2+(WQPOSC(2)-WMPOSC(2))**2)
      WQPOSC(4)=DATAN2((WQPOSC(1)-WMPOSC(1)),(WQPOSC(2)-WMPOSC(2)))
      WQPOSC(5)=DATAN2(GRC,WMPOSC(3))
      WQPOSC(6)=DSQRT(GRC**2+WMPOSC(3)**2)
      WQPOSB(1)=WMPOSB(3)*DTAN(WQPOSC(5))*DSIN(WQPOSC(4))+WMPOSB(1)
      WQPOSB(2)=WMPOSB(3)*DTAN(WQPOSC(5))*DCOS(WQPOSC(4))+WMPOSB(2)
      WQPOSE(1)=WMPOSE(3)*DTAN(WQPOSC(5))*DSIN(WQPOSC(4))+WMPOSE(1)
      WQPOSE(2)=WMPOSE(3)*DTAN(WQPOSC(5))*DCOS(WQPOSC(4))+WMPOSE(2)
      RANMN=WMPOSC(3)/DCOS(WQPOSC(5)-WMPOSC(4))
      RANMX=WMPOSC(3)/DCOS(WQPOSC(5)+WMPOSC(4))
      RANQ9=WMPOSC(3)/DCOS(WQPOSC(5))
      WMPOSC(5)=RANMN
      WMPOSC(6)=RANMX
      JRANQ9=INT((WQPOSC(6)-RANMN+0.5*W(2))/W(2))
      JRANMX=INT((RANMX-RANMN+0.5*W(2))/W(2))
   18 FORMAT(F10.3,2F10.6,5F10.3,F10.6)
      IF(KFP.NE.KFPDIS) RETURN
      PRINT*,'VELOM,ALFM,BETM,(WMPOSC(K1),K1=1,3),(WQPOSC(K2),K2=1,2)'
      PRINT*,VELOM,ALFM,BETM,(WMPOSC(K1),K1=1,3),(WQPOSC(K2),K2=1,2)
      PRINT*,' YM,ALFQ9,WQPOSC4,RANQ9,RANMN,RANMX,JRANQ9,JRANMX'
```

```
      PRINT*,WMPOSC(2),ALFQ9,WQPOSC(4),RANQ9,RANMN,RANMX,JRANQ9,JRANMX
      PRINT*,'@@@@@@ EXIT SUB- BCEDOP @@@@@@'
      RETURN
      END
```

Subroutine CLUFPT.FOR

```
      SUBROUTINE CLUFPT(DELRAN,JRANQ9,JRANMX,WQ9,WMPOSC,NSCAFP,SMOMAP.
     &WADRLO,WADRUP,WADSMO)
C     CREATED ON 1/24/1989, 1300 hr ************************************
      REAL WMPOSC(1),WSCA(7),WQ9(1),WMSCA(3),SMOMAP(1),
     &GRDMAP,WADRLO(1),WADRUP(1),WJRAN(200),WADSMO(1),W(34)
      REAL COSANT,HDELRA,DOTCOS,EPS
      INTEGER NBSCA,KSMO,KSCA,KX,JRANON,KSCAT,KINC,K,J
      COMMON /GRDCLU/NBSCA,GRDMAP(501),COSANT
      PRINT*,'@@@@@@ SUB-CLUFPT: uses GRDCLU to compute SMOMAP @@@@@'
C     WQ9=(XQ,YQ,ZQ,ALFQ9,BETQ9,RANQ9) ***********************************
C     SMOOTHED OUTPUT FOR CURRENT FOOTPRINT IS SMOMAP, ONE REAL
C     &SCATTERER GIVES 15-COMPONENTS ************************************
C     SMOMAP=(EARLY,SCAFAS,MU,JRANON-1,ZSCA,ON,SCAFAS,MU,JRANON,ZSCA,LAT
C     &E,SCAFAS,MU,JRANON+1,ZSCA) ***************************************
      KSMO=0
      HDELRA=DELRAN/2.
C     WSCA=(X-SCA,Y-SCA,Z-SCA,AMP-SCA,FAS-SCA) *************************
      DO 5 KSCA=1,NBSCA
      KX=(KSCA-1)*5+3
      WSCA(1)=GRDMAP(KX)
      WSCA(2)=GRDMAP(KX+1)
      WSCA(3)=GRDMAP(KX+2)
      WSCA(4)=GRDMAP(KX-2)
      WSCA(5)=GRDMAP(KX-1)
      IF(WSCA(4).LT.0.001) GO TO 5
      IF(DOTCOS(WMPOSC,WQ9,WSCA).LT.COSANT) GO TO 5
C     REPLACE (X,Y) BY (MU,RANSCA) *************************************
      WSCA(6)=WSCA(1)
      WSCA(7)=WSCA(2)
      WSCA(1)=ATAN2(WSCA(6)-WMPOSC(1),WSCA(7)-WMPOSC(2))-WQ9(4)
      WSCA(2)=SQRT((WSCA(6)-WMPOSC(1))**2+(WSCA(7)-WMPOSC(2))**2+
     &(WSCA(3)-WMPOSC(3))**2)
C     RANGE SMOOTHING:1-SCATTERER GIVES WMSCA=(EARLY,ON,LATE) **********
      JRANON=((WSCA(2)-WQ9(6)+HDELRA)/DELRAN)+JRANQ9
      WRITE(3,*)'JRANON=',JRANON,'  JRANQ9=',JRANQ9
      CALL RANGSMOO(DELRAN,JRANON,JRANQ9,WQ9(6),WMPOSC,WSCA,WMSCA)
      KSCAT=0
      DO 7 KINC=2,12,5
      SMOMAP(KSMO+KINC)=WSCA(5)
      SMOMAP(KSMO+KINC+1)=WSCA(1)
      KSCAT=KSCAT+1
```

```
      SMOMAP(KSMO+KINC-1)=WMSCA(KSCAT)
      SMOMAP(KSMO+KINC+2)=JRANON-2+KSCAT
 7    SMOMAP(KSMO+KINC+3)=WSCA(3)
      KSMO=KSMO+15
 5    CONTINUE
      EPS=0.00001
      NSCAFP=0
      DO 10 K=4,KSMO,5
      NSCAFP=NSCAFP+1
      WADSMO(NSCAFP)=K
 10   WJRAN(NSCAFP)=SMOMAP(K)
      CALL SORTRI(WJRAN,WADSMO,-NSCAFP)
      DO 12 J=1,JRANMX
      WADRLO(J)=0.
      WADRUP(J)=0.
      DO 14 K=NSCAFP,1,-1
 14   IF(ABS(WJRAN(K)-FLOAT(J)).LE.EPS) WADRLO(J)=FLOAT(K)
      DO 16 K=1,NSCAFP
 16   IF(ABS(WJRAN(K)-FLOAT(J)).LE.EPS) WADRUP(J)=FLOAT(K)
 12   CONTINUE
      PRINT*,'@@@@@@ EXIT SUB-CLUFPT @@@@@@'
      RETURN
      END
```

Subroutine CLUMAP.FOR

```
      SUBROUTINE CLUMAP(HTHETA,LEVBUG)
C     CREATED 02059, 2200 hr *********************************************
      CHARACTER*240 TITLE
      REAL W(34),GRDMAP,HTHETA,COSANT,W1,XI,YI,WGRD(6),SCANB
      INTEGER KSCA,KK,NBSCA,K,LEVBUG
      COMMON /GRDCLU/NBSCA,GRDMAP(501),COSANT
      PRINT*,'@@@@@@  SUBROUTINE CLUMAP  @@@@@@'
      OPEN (UNIT=11,FILE='CLMPI.INP' ,STATUS='OLD')
C     CLUTTER INFO. IS IN CLMPI.INP FORMATTED AS FOLLOWS:FIRST ROW=NBSCA
C     OTHER ROWS {AMP-SCA, FAS-SCA, X-SCA, Y-SCA, Z-SCA} ***************
      READ(11,50) SCANB
      NBSCA=INT(SCANB+.0001)
 40   FORMAT(6(1X,F10.3))
 42   FORMAT(1X,'  GRDMAP(K),K=1,5)   '/5(1XF10.3) )
 50   FORMAT(6F10.3)
      COSANT=COS(HTHETA)
      KK=0
      IF(LEVBUG.GE.1)PRINT*,'AMP-SCA, FAS-SCA, X-SCA, Y-SCA, Z-SCA,SCA#'
      DO 3 KSCA=1,NBSCA
      READ(11,50) (WGRD(K),K=1,6)
      IF(LEVBUG.GE.1) WRITE(*,40) (WGRD(K),K=1,6)
```

```
      DO 5 K=1,5
      KK=KK+1
5     GRDMAP(KK)=WGRD(K)
3     CONTINUE
      CLOSE (11)
      IF(LEVBUG.EQ.2) WRITE(*,42) (GRDMAP(K),K=1,5)
      PRINT*,'@@@@@@   EXIT CLUMAP LEVBUG=',LEVBUG
      RETURN
      END
```

Subroutine DETFPT.FOR

```
      SUBROUTINE DETFPT(KFP,JRAN,KFRQ,FMUCEL,XM,JRANQ9,DELRAN,WMPOSC,
     &WQPOSC,IN123,IPRT1)
      REAL ALFQ9,RANSCA,BETSCA,FMUCEL,WQPOSC(1),WMPOSC(1),XSCA,
     &YSCA,ALFSCA,GRNSCA,RANMN,RANMX,RANQ9,DELRAN,XM
      REAL WSCDTX,WSCDTY,WSCDTM
      INTEGER KFP,JRAN,KFRQ,JRANQ9,IN123,K,NBSCDT,IWFPT,IWRAN,IWFRQ,
     &IPRT1,JWFPT,JWRAN,JWFRQ,NBINCU,NBINOL
      REAL WM(400),WX(400),WY(400),VM,VX,VY
      INTEGER NW,LOCBEG(400),LOCEND(400),NPIK,LOCPIK(200),KAD,ICOMPR
      COMMON /DETECT/NBSCDT,WSCDTX(400),WSCDTY(400),WSCDTM(400),
     &IWFPT(400),IWRAN(400),IWFRQ(400),NBINCU,VX(129),VY(129)
     &,VM(129),JWRAN(129),JWFRQ(129),JWFPT(129),NBINOL
4     FORMAT(1X,'DETFPT-FOR4',3(1XI4),2X,3(1XF11.3))
      GO TO (1,2,3),IN123
1     NBINCU=NBINCU+1
      ALFQ9=WQPOSC(4)
      RANQ9=WQPOSC(6)
      RANMN=WMPOSC(5)
      RANMX=WMPOSC(6)
      RANSCA=RANQ9+DELRAN*FLOAT(JRAN-JRANQ9)
      BETSCA=ACOS(WMPOSC(3)/RANSCA)
      GRNSCA=WMPOSC(3)*TAN(BETSCA)
      ALFSCA=ALFQ9+FMUCEL
      XSCA=GRNSCA*SIN(ALFSCA)+WMPOSC(1)
      YSCA=GRNSCA*COS(ALFSCA)+WMPOSC(2)
      VX(NBINCU)=XSCA
      VY(NBINCU)=YSCA
      VM(NBINCU)=XM
      JWFPT(NBINCU)=KFP
      JWRAN(NBINCU)=JRAN
      JWFRQ(NBINCU)=KFRQ
9     FORMAT('DETFPT-1B: '4(1XI3),1X,3(1XF10.4))
      IF (IPRT1.EQ.1) WRITE(*,9) KFP,JRAN,KFRQ,NBINCU,XM,XSC \,YSCA
      NBINOL=NBINCU
      RETURN
```

```
2   PRINT*,'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    NBINOL=NBINCU
    NBINCU=0
    NW=NBINOL
    DO 17 K=1,NW
    WM(K)=VM(K)
    WX(K)=FLOAT(JWRAN(K))
    WY(K)=FLOAT(JWFRQ(K))
    LOCBEG(K)=K
17  LOCEND(K)=0
    CALL FPTPIK(WM,WX,WY,LOCBEG,LOCEND,NW,2,4,LOCPIK,NPIK,0)
    PRINT*,'DETFPT+FPTPIK: KPIK,KAD,KFP,JRAN,KNPPP,XM,XSCA,YSCA'
    DO 31 K=1,NPIK
    KAD=LOCPIK(K)
    NBSCDT=NBSCDT+1
    WSCDTX(NBSCDT)=VX(KAD)
    WSCDTY(NBSCDT)=VY(KAD)
    WSCDTM(NBSCDT)=VM(KAD)
    IWFPT(NBSCDT)=JWFPT(KAD)
    IWRAN(NBSCDT)=JWRAN(KAD)
    IWFRQ(NBSCDT)=JWFRQ(KAD)
31  WRITE(*,12)K,KAD,JWFPT(K),JWRAN(KAD),JWFRQ(KAD),VM(KAD),
   &VX(KAD),VY(KAD)
    RETURN
3   NW=NBSCDT
    IF(NW.EQ.1) RETURN
    DO 7 K=1,NW
    WM(K)=WSCDTM(K)
    WX(K)=WSCDTX(K)
    WY(K)=WSCDTY(K)
    LOCBEG(K)=K
7   LOCEND(K)=0
    CALL LSTPIK(WM,WX,WY,LOCBEG,LOCEND,NW,6.1,LOCPIK,NPIK,0)
    PRINT*,'DETFPT+LSTPIK: KPIK,KAD,KFP,JRAN,KNPPP,XM,XSCA,YSCA'
    DO 11 K=1,NPIK
    KAD=LOCPIK(K)
11  WRITE(*,12)K,KAD,IWFPT(KAD),IWRAN(KAD),IWFRQ(KAD),WSCDTM(KAD),
   &WSCDTX(KAD),WSCDTY(KAD)
12  FORMAT(1X,'DETFPT-!',5(1XI3),1X,3(1XF11.4))
    RETURN
    END
```

Subroutine FPIDOP.FOR

```
    SUBROUTINE FPIDOP(NBFPMX,NBFP,HTHETA,TIMDWL,KFPDIS)
    INTEGER KFP,NBFP,K,NBCY,NBFPCY,KA,KB,I,NBFPMX
    REAL ANGCY,DLALF,FPLNG,PERCY,DLTISQ,HDLTIS,ALFONE,ALFFP,TIMINC
    REAL XFP,YFP,VELOM,ALFM,BETM,XM,YM,ZM,SALFM,CALFM,SBETM,CBETM,
   &SBETFP,CALFCY,CBETFP,GR
    REAL V(12),WMVELO(3),WMPOSC(3)
```

```
      CHARACTER*240 TITLE
      PRINT*,'***************************************************************'
      PRINT*,'@@@@@@ SUB-FPIDOP; information for FP-sequence @@@@@@'
      OPEN(UNIT=12,FILE='GEODOPI.INP',STATUS='OLD')
      TITLE='VELOM=1; ALFM=2; BETM=3; WMBRUN=4-6; BETFP=7; ALFCY=8 /
     &OVRLAP=9; CYNB=10; TIMGAP=11; ALFSTR=12 '
      CALL WINPUX(12,V,12,TITLE)
      CLOSE (12)
      SALFM=SIN(V(2))
      CALFM=COS(V(2))
      SBETM=SIN(V(3))
      CBETM=COS(V(3))
      SBETFP=SIN(V(7))
      CBETFP=COS(V(7))
      NBCY=INT(V(10)+0.001)
      CALFCY=COS(V(8))
      ANGCY=ACOS(SBETFP**2*CALFCY+CBETFP**2)
      NBFPCY=INT((ANGCY*V(9))/(4.*HTHETA) + 0.5 ) * 2
      DLALF=V(8)/FLOAT(NBFPCY)
      FPLNG=V(6)*(TAN (V(7)+HTHETA)-TAN(V(7)-HTHETA))
      WMVELO(1)=V(1)*SBETM*SALFM
      WMVELO(2)=V(1)*SBETM*CALFM
      WMVELO(3)=-V(1)*CBETM
      PERCY=(FPLNG/V(9))/WMVELO(2)
      DLTISQ=(PERCY-V(11)) / FLOAT(NBFPCY)
      HDLTIS=0.5*DLTISQ
      PRINT*,'NBCY, ANGCY, NBFPCY, DLALF, FPLNG, PERCY, DLTISQ'
      WRITE(*,*) NBCY,ANGCY,NBFPCY,DLALF,FPLNG,PERCY,DLTISQ
      IF(DLTISQ.LT.TIMDWL) STOP 'ERROR-FPIDOP'
      NBFP=NBFPCY*NBCY
      IF(NBFP.GT.NBFPMX) NBFP=NBFPMX
      DO 10 K=1,3
10    WMPOSC(K)=V(K+3)+WMVELO(K)*HDLTIS
      ALFONE=-0.5*(V(8)-DLALF)
      IF(V(12).GT.1000.) ALFONE=V(12)-2000.
      ALFFP=ALFONE
      GR=WMPOSC(3)*TAN(V(7))
      XFP=GR*SIN(ALFFP)+WMPOSC(1)
      YFP=GR*COS(ALFFP)+WMPOSC(2)
      REWIND 10
      WRITE(10,1)(V(KA),KA=1,3),(WMPOSC(KB),KB=1,3),XFP,YFP,ALFFP
1     FORMAT (F10.3,2F10.6,5F10.3,F10.6)
      IF(NBFP.EQ.1) GO TO 6
      DO 12 KFP=2,NBFP
      I=MOD( (KFP-1),NBFPCY)
      DO 14 K=1,3
      TIMINC=DLTISQ
      IF(I.EQ.0) TIMINC=V(11)+DLTISQ
14    WMPOSC(K)=WMPOSC(K)+WMVELO(K)*TIMINC
      ALFFP=ALFFP+DLALF
      IF(I.EQ.0) ALFFP=ALFONE
      GR=WMPOSC(3)*TAN(V(7))
      XFP=GR*SIN(ALFFP)+WMPOSC(1)
```

```
      YFP=GR*COS(ALFFP)+WMPOSC(2)
      WRITE(10,1)(V(KA),KA=1,3),(WMPOSC(KB),KB=1,3),XFP,YFP,ALFFP
12    CONTINUE
6     REWIND 10
      PRINT *,'VELOM  ALFM  BETM  XM  YM  ZM  XFP  YFP  ALFFP'
      DO 16 K=1,NBFP
      READ(10,1) VELOM, ALFM, BETM, XM, YM, ZM, XFP, YFP, ALFFP
      IF(K.EQ.KFPDIS) WRITE(*,2) VELOM, ALFM, BETM, XM, YM, ZM,
     &XFP, YFP, ALFFP
16    CONTINUE
2     FORMAT(9(1X,F8.3))
      REWIND 10
      PRINT*,'@@@@@@ EXIT SUB- FPIDOP @@@@@@'
      RETURN
      END
```

Subroutine FPTPIK.FOR

```
      SUBROUTINE FPTPIK(WM,WX,WY,LOCBEG,LOCEND,NW,JRNDIS,JAZDIS,LOCPIK,
     &NPIK,IPRT)
      INTEGER LOCBEG(1),LOCEND(1),NW,LOCPIK(1),NPIK,K,IDSCA,KS,KK,KID,
     &KFIND,IPRT,JRNDIS,JAZDIS
      REAL WM(1),WX(1),WY(1),XS,YS
      CALL SOR3RL(WM,WX,WY,LOCBEG,NW)
      DO 20 K=1,NW
20    IF(IPRT.EQ.1)PRINT*,'FPTPIK+SORT',K,WM(K),WX(K),WY(K)
      IDSCA=0
      DO 10 K=NW,1,-1
      IF(LOCEND(K).NE.0) GO TO 10
      IDSCA=IDSCA+1
      LOCEND(K)=IDSCA
      KS=K
      XS=WX(K)
      YS=WY(K)
      IF(KS.EQ.1) GO TO 10
      DO 13 KK=KS-1,1,-1
      IF(LOCEND(KK).NE.0) GO TO 13
      IF(ABS(XS-WX(KK)).LT.FLOAT(JRNDIS).AND.ABS(YS-WY(KK)).LT.FLOAT
     &(JAZDIS)) LOCEND(KK)=IDSCA
13    CONTINUE
10    CONTINUE
      NPIK=IDSCA
      DO 25 KID=1,NPIK
      KFIND=0
      DO 35 K=NW,1,-1
      IF(KFIND.EQ.1.OR.LOCEND(K).NE.KID) GO TO 35
      KFIND=1
      LOCPIK(KID)=LOCBEG(K)
35    CONTINUE
```

```
25  CONTINUE
    IF(IPRT.EQ.1)PRINT*,'FPTPIK-END: KPIK, LOCPIK, LOCBEG'
    DO 37 K=1,NPIK
    IF(IPRT.EQ.1)WRITE(*,4) K,LOCPIK(K),LOCBEG(K)
37  CONTINUE
4   FORMAT(1X,'FPTPIK-END',3(2XI5))
    RETURN
    END
```

Function FRTOMU.FOR

```
    FUNCTION FRTOMU(WTAU,FRQCEL)
    REAL FRTOMU,WTAU(1),FRQCEL
    DOUBLE PRECISION TEMP,ATAU6,TMU
    ATAU6=ABS(WTAU(6))
    TEMP=ATAU6**2-2.*WTAU(10)*FRQCEL
    IF(TEMP.GT.1.E-16) TMU=(-ATAU6+DSQRT(TEMP))/WTAU(10)
    IF(TEMP.LE.1.E-16) TMU=-FRQCEL/ATAU6
    FRTOMU=TMU
    RETURN
    END
```

Subroutine LSTPIK.FOR

```
    SUBROUTINE LSTPIK(WM,WX,WY,LOCBEG,LOCEND,NW,DISPIK,LOCPIK,NPIK,
   &IPRT)
    INTEGER LOCBEG(1),LOCEND(1),NW,LOCPIK(1),NPIK,K,IDSCA,KS,KK,KID,
   &KFIND,IPRT
    REAL WM(1),WX(1),WY(1),DISPIK,XS,YS
    CALL SOR3RL(WM,WX,WY,LOCBEG,NW)
    DO 20 K=1,NW
20  IF(IPRT.EQ.1)PRINT*,'LSTPIK+SORT',K,WM(K),WX(K),WY(K)
    IDSCA=0
    DO 10 K=NW,1,-1
    IF(LOCEND(K).NE.0) GO TO 10
    IDSCA=IDSCA+1
    LOCEND(K)=IDSCA
    KS=K
    XS=WX(K)
    YS=WY(K)
    IF(KS.EQ.1) GO TO 10
    DO 13 KK=KS-1,1,-1
    IF(LOCEND(KK).NE.0) GO TO 13
    IF(ABS(XS-WX(KK)).LT.DISPIK.AND.ABS(YS-WY(KK)).LT.DISPIK)
```

```
   &LOCEND(KK)=IDSCA
13  CONTINUE
10  CONTINUE
    NPIK=IDSCA
    DO 25 KID=1,NPIK
    KFIND=0
    DO 35 K=NW,1,-1
    IF(KFIND.EQ.1.OR.LOCEND(K).NE.KID) GO TO 35
    KFIND=1
    LOCPIK(KID)=LOCBEG(K)
35  CONTINUE
25  CONTINUE
    IF(IPRT.EQ.1)PRINT*,'LSTPIK-END: KPIK, LOCPIK, LOCBEG'
    DO 37 K=1,NPIK
    IF(IPRT.EQ.1)WRITE(*,4) K,LOCPIK(K),LOCBEG(K)
37  CONTINUE
4   FORMAT(1X,'LSTPIK-END',3(2XI5))
    RETURN
    END
```

Subroutine MAPWIN.FOR

```
    SUBROUTINE MAPWIN(WPAR,IWIN,WCNWIN,WMNWIN,XFPT,YFPT,MXXMAP,MXYMAP,
   &NXMAP,NYMAP)
    REAL WPAR(8),WCNWIN(2),WMNWIN(2),XFPT,YFPT,XCEN,YCEN
    INTEGER IWIN,K,NXMAP,NYMAP,MXXMAP,MXYMAP
C    IWIN=(1,2,3,4):1=no-reset,2=center WCNWIN,3=min-corner WMNWIN,4=
C    center (XFPT,YFPT); Size of window is:(WPAR7,WPAR8)**************
    GO TO (1,2,3,4),IWIN
1   DO 9 K=3,4
9   WPAR(K)=WPAR(1)*FLOAT(INT(WPAR(K)/WPAR(1)+0.5))
    PRINT*,'MAPWIN-1:readjust WPAR3-4 to int# DELMAP '
    GO TO 14
2   DO 7 K=1,2
7   WPAR(K+2)=WPAR(1)*FLOAT(INT((WCNWIN(K)-WPAR(K+6)*.5)/WPAR(1)+0.5))
    PRINT*,'MAPWIN-2:compute WPAR3-4 from center WCNWIN'
    GO TO 14
3   DO 6 K=1,2
6   WPAR(K+2)=WPAR(1)*FLOAT(INT(WMNWIN(K)/WPAR(1)+0.5))
    PRINT*,'MAPWIN-3:compute WPAR3-4 from min-corner WMNWIN'
    GO TO 14
4   WCNWIN(1)=XFPT
    WCNWIN(2)=YFPT
    DO 11 K=1,2
11  WPAR(K+2)=WPAR(1)*FLOAT(INT((WCNWIN(K)-WPAR(K+6)*.5)/WPAR(1)+0.5))
    PRINT*,'MAPWIN-4:compute WPAR3-4 from center (XFPT,YFPT)'
14  CONTINUE
    NXMAP=INT(0.5+WPAR(7)/WPAR(1))
    NYMAP=INT(0.5+WPAR(8)/WPAR(1))
```

```
      IF(NXMAP.GT.MXXMAP.OR.NYMAP.GT.MXYMAP)STOP 'size window GT dimensi
     &on of SYNMAP & SUMARA'
      XCEN=WPAR(3)+0.5*FLOAT(NXMAP)*WPAR(1)
      YCEN=WPAR(4)+0.5*FLOAT(NYMAP)*WPAR(1)
      WPAR(5)=WPAR(3)+FLOAT(NXMAP)*WPAR(1)
      WPAR(6)=WPAR(4)+FLOAT(NYMAP)*WPAR(1)
      WPAR(7)=WPAR(5)-WPAR(3)
      WPAR(8)=WPAR(6)-WPAR(4)
      PRINT*,'MAPWIN: XMN,YMN,XMX,YMX,XSIZE,YSIZE,XCEN,YCEN,NXMAP,NYMAP'
      WRITE(*,20) (WPAR(K),K=3,8),XCEN,YCEN,NXMAP,NYMAP
 20   FORMAT(8(1XF8.2),2(1XI3))
      RETURN
      END
```

Subrotine SOR3RL.FOR

```
      SUBROUTINE SOR3RL(VAL,VX,VY,LOC,N)
      REAL VAL(1),VX(1),VY(1),VALS,VXS,VYS
      INTEGER LOC(1),N,LOCS,K,M,J,I,II
 7    M=N
 1    CONTINUE
      M=M/2
      IF(M.EQ.0) RETURN
      K=N-M
      J=1
 2    CONTINUE
      I=J
 3    CONTINUE
      II=I+M
      IF(VAL(I).LT.VAL(II)) GO TO 4
      VALS=VAL(I)
      VXS=VX(I)
      VYS=VY(I)
      LOCS=LOC(I)
      VAL(I)=VAL(II)
      VX(I)=VX(II)
      VY(I)=VY(II)
      LOC(I)=LOC(II)
      VAL(II)=VALS
      VX(II)=VXS
      VY(II)=VYS
      LOC(II)=LOCS
      I=I-M
      IF(I.GE.1) GO TO 3
 4    CONTINUE
      J=J+1
      IF(J.GT.K) GO TO 1
      GO TO 2
```

```
      RETURN
      END
```

Subroutine SURFMP.FOR

```
      SUBROUTINE SURFMP(KFRQ,KRAN,Z,JRANMX,IN123)
      INTEGER MNIW,MNJW,MXIW,MXJW,KFRQ,KRAN,IN123,IW,JW,NPINFP,JRANMX
      REAL ZMX,SYNMAP,Z
      COMMON /SURFP/SYNMAP(128,200),MNIW,MNJW,MXIW,MXJW,ZMX,NPINFP
      GO TO (1,2,3), IN123
1     MNIW=999
      MNJW=999
      MXIW=-999
      MXJW=-999
      ZMX=0.
      NPINFP=0
      DO 5 IW=1,128
      DO 5 JW=1,200
5     SYNMAP(IW,JW)=0.
      RETURN
2     NPINFP=NPINFP+1
      MNIW=1
      MNJW=1
      MXIW=128
      MXJW=JRANMX
      ZMX=AMAX1(ZMX,Z)
      SYNMAP(KFRQ,KRAN)=Z
      RETURN
3     OPEN(UNIT=7,FILE='DOPFPG.GRD',STATUS='OLD')
      WRITE(7,'("DSAA")')
      WRITE(7,'(I5,1X,I5)')MXIW-MNIW+1,MXJW-MNJW+1
      WRITE(7,'(E12.5,1X,E12.5)')FLOAT(MNIW),FLOAT(MXIW)
      WRITE(7,'(E12.5,1X,E12.5)')FLOAT(MNJW),FLOAT(MXJW)
      WRITE(7,'(E12.5,1X,E12.5)')0.,ZMX
      DO 69 JW=MNJW,MXJW
      WRITE(7,62) (SYNMAP(IW,JW),IW=MNIW,MXIW)
      WRITE(7,'( )')
69    CONTINUE
62    FORMAT(10(1XF8.3))
      CLOSE(7)
      STOP
      END
```

Subroutine WINPUX.FOR

```
      SUBROUTINE WINPUX(NW,W,IDFILE,TITLE)
      CHARACTER*240 TITLE
      CHARACTER*55 WCHAR(51)
      REAL W(1)
      INTEGER KW,K
      REWIND IDFILE
      DO 7 KW=1,NW
7     READ(IDFILE,11) W(KW),WCHAR(KW)
11    FORMAT(F16.7,A55)
      PRINT*,'ARRAY W BEFORE UPDATE'
      WRITE(*,'(1X,A240)') TITLE
      WRITE(*,2) (W(K),K=1,NW)
2     FORMAT(4(2XF16.7))
      CALL WCHANGE (W)
      REWIND IDFILE
      DO 8 KW=1,NW
8     WRITE(IDFILE,'(F16.7,A55)') W(KW),WCHAR(KW)
      REWIND IDFILE
      DO 17 KW=1,NW
17    READ(IDFILE,11) W(KW),WCHAR(KW)
      PRINT*,'ARRAY W AFTER UPDATE'
      WRITE(*,2) (W(K),K=1,NW)
      RETURN
      END
```

Function YMU3DB.FOR

```
      FUNCTION YMU3DB(HTHETA,BETQ9,BETSCA,IPRT)
      REAL YMU3DB,BETQ9,BETSCA,HTHETA
      DOUBLE PRECISION COSMU,HTHETD,BETQ9D,BETSCD
      HTHETD=HTHETA
      BETQ9D=BETQ9
      BETSCD=BETSCA
      COSMU=(DCOS(HTHETD)-DCOS(BETQ9D)*DCOS(BETSCD))/(DSIN(BETQ9D)*
     &DSIN(BETSCD))
      IF(DABS(COSMU).GT.1.D0) GO TO 2
      YMU3DB=DACOS(COSMU-.0000000001D0)
      IF(IPRT.EQ.0)RETURN
2     PRINT*,'HTHETA, BETQ9, BETSCA, COSMU, YMU3DB'
      PRINT*,HTHETA, BETQ9, BETSCA, COSMU, YMU3DB
      IF(DABS(COSMU).GT.1.D0) STOP 'ABS(COSMU) > 1.'
      RETURN
      END
```

Appendix D

Listings for DOPXY.FOR, MAPAZD.FOR, and MAPRGD.FOR

Program DOPXY.FOR

```
C CREATED  02 26 9 2300hr
    CHARACTER*240 TITLE
    REAL W(23),TIMCNT,TIMDWL,DELRAN,XMMNPR,WPAR(8),XIFXMT,SELWIN,
   &DETIPR,SELDIS,BITINT,WSQ(12),SHOSCA,BUGLEV,DUM1,DUM2,WINIWR
    REAL SMOMAP(450),TXD(64),TXQ(64),WANGN9(265),WAGNCL(128),
   &WBET9(265),WDUM(3),WDOVL(3),WEPSMX(265),WIND(64)
    REAL VELOM,ALFM,BETM,WMPOSB(3),WMPOSC(6),WMPOSE(3),WQPOSB(3),
   &WQPOSC(6),WQPOSE(3),WFREQ(128),WMVELO(6),WWAPEX(128,9)
    REAL WMU3DB(265),WMUMDR(266),WRAN9(265),WTAU(25),WTIME(64),
   &WTSMO(5),XD(128),XM(128),XQ(128),WADRLO(265),WADRUP(265),
   &WADSMO(90),WAGAIN(3),HEIREF,BETREF,RANREF
    INTEGER LNPPP(128),NBSCA,KFPDIS,NW,LEVBUG,NBFP,NBSCIN,JRSCMN,
   &JRSCMX,IWZMX,JWZMX,KFPSGN,ISURF,IPRDET,NBINCU,NBINOL,IWMX,JWMX
    REAL DISH,DOTCOS,FRQOFMU,FREQ,COSANT,GRDMAP,TMU3DB,
   &FPMXNB,RELDOP,BET,SURFIN
    REAL CLIGHT,DELFIN,DELFUT,DELTIM,DOVL,EPS,FMU,FREQO,GDISH2,
   &GDISH3,GRC,HDELRA,HTHETA,HTIMDW,RANCOF,OMEGO,SCAFAC,
   &SINANG,THETA,TMPBET,TPI,V1,V2,V3,X,XSCA,YSCA,Y,Z,FZMX,SCANEW
    INTEGER IDDISH,IFILE,INDX,INTCOF,IW,JRAN,JRANQ9,JRANMX,JW,K,
   &KADSMO,KDQ,KI,KJ,KFF,KNPPP,KSCALO,KSCAUP,KT,KTIM,KTHSCA,IWRWIN,
   &NBITFF,NBITFX,NFFT,NFFTH,NFFX,NFFXH,NPPPMN,NPPPMX,NSCAFP,NXMAP,
   &NYMAP,NBFPMX,KFP
    EXTERNAL YMU3DB,FRTOMU
LARGE SYNMAP,SUMARA
    REAL SYNMAP(0:137,0:250),SUMARA(0:137,0:250)
    COMMON /GRDMAP/SYNMAP,SUMARA
    INTEGER NBSCDT,IWFPT,IWRAN,IWFRQ,JWFPT,JWRAN,JWFRQ
    REAL WSCDTX,WSCDTY,WSCDTM,VX,VY,VM
    COMMON /DETECT/NBSCDT,WSCDTX(400),WSCDTY(400),WSCDTM(400),
   &IWFPT(400),IWRAN(400),IWFRQ(400),NBINCU,VX(129),VY(129)
   &,VM(129),JWRAN(129),JWFRQ(129),JWFPT(129),NBINOL
    EQUIVALENCE (W(1),TIMDWL),(W(2),DELRAN),(W(3),XMMNPR),(W(4),
   &WPAR(1)),(W(12),XIFXMT),(W(13),SELWIN),(W(14),DETIPR),(W(15),
   &SELDIS),(W(16),WINIWR),(W(17),SCANEW),(W(18),HEIREF),
   &(W(19),BETREF),(W(20),SHOSCA),(W(21),BUGLEV),(W(22),FPMXNB),
   &(W(23),SURFIN)
C    PARAMETERS FOR OUTPUT MAP ARE IN WPAR=(DELMAP,THRESH,XMNMAP,YMNMAP
C    ,XMXMAP,YMXMAP,XSIZE,YSIZE) ***************************************
C    WMVELO=(XVELO,YVELO,ZVELO,VELOM,{2/WAVLNGH},DELFUT) **************
C    WMPOSC=( XM, YM, ZM, HTHETA, RANMN, RANMX)************************
C    WQPOSC=(XFP=XQ9,YFP=YQ9,ZFP=ZQ9,ALFQ9,BETQ9,RANQ9) ***************
C    DIMENSION OF SMOMAP = NBSCA*15 (SEE SUB. CLUFPT) *****************
    FRQOFMU(V1,FMU,V2,V3)=-V1*(V2*FMU+(V3/2.)*FMU**2)
```

```
C   @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C   I/O FILES : U9=SYPAI.INP(main);U3=DOPFPP.PRT(main);U11=CLMPI.INP
C   (clumap);U10=FPSQI.INP(made in fpidop,used in bcedop);U12=GEODOPI.
C   &INP(fpidop);U7=DOPFPG.GRD(main) @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C   HALF 3db BEAMWIDTH=.0215 gives DOVL=14.83343168=DISH DIAMETER IN
C   &LAMBDA UNITS ( 2-WAY GAIN = -3.00 DB means DISH=0.84139514)*******
    DATA THETA,FREQO,CLIGHT,TPI,EPS,WDOVL,NW,BITINT/.043,94.E9,3.E8,
   &6.28318531,.0001,17.15094,17.15094,14.80884,23,1./
    DATA WQPOSB(3),WQPOSC(3),WQPOSE(3)/0.,0.,0./
    NBSCDT=0
    SINANG=SIN(0.0215)
    GDISH2=DISH(SINANG,2,WDOVL(2))
    GDISH3=DISH(SINANG,3,WDOVL(3))
    OPEN(UNIT=3,FILE='DOPFPP.PRT',STATUS='OLD')
    WRITE(3,*)'GDISH2,GDISH3',GDISH2,GDISH3
    PRINT*,'WINIWR: 0. window by WPAR, 1. window centered at
   &(XFP,YFP) of footprint KFP=KFPDIS'
    TITLE='TIMDWL=1;DELRAN=2;XMMNPR=3;WPAR1=4=DELMAP;WPAR2=THRESH;WPAR
   &3=XMN;WPAR4=YMN;WPAR5=XMX/WPAR6=YMX;WPAR7=XSIZE;WPAR8=YSIZE;XIFX
   &=12;SELWIN=13;DETIPR=14;SELDIS=15;WINIWR=16/SCANEW=17;HEIREF=18;BE
   &TREF=19;SHOSCA=20;BUGLEV=21;FPMXNB=22;SURFIN=23'
    OPEN(UNIT=9,FILE='SYPAI.INP',STATUS='OLD')
    CALL WINPUX(NW,W,9,TITLE)
    PRINT*,'in WPAR specify (1,2,3,4)&(7,8), then (5,6) are computed'
    PRINT*,'in MAPWIN last 2-arguments are DIM of SYNMAP,DUM=dummy'
    CALL MAPWIN(WPAR,1,WDUM,WDUM,DUM1,DUM2,137,250,NXMAP,NYMAP)
    IWRWIN=INT(SURFIN+EPS)
    IDDISH=INT(SELDIS+.00001)
    LEVBUG=INT(BUGLEV+.00001)
    ISURF=INT(SURFIN+EPS)
    IPRDET=INT(DETIPR+EPS)
    NBFPMX=INT(FPMXNB+EPS)
    DOVL=WDOVL(IDDISH)
    OMEGO=TPI*FREQO
    HDELRA=DELRAN*0.5
    HTHETA=THETA*0.5
    WMPOSC(4)=HTHETA
    NBITFF=6
    NBITFX=NBITFF+INT(BITINT+EPS)
    NFFT=2**NBITFF
    NFFTH=NFFT/2
    NFFX=2**NBITFX
    NFFXH=NFFX/2
    DELFIN=1./TIMDWL
    INTCOF=2**INT(BITINT+EPS)
    DELFUT=DELFIN/FLOAT(INTCOF)
    WMVELO(6)=DELFUT
    DELTIM=TIMDWL/NFFT
    HTIMDW=TIMDWL*.5
C   COMPUTE RANGE REFERENCE TO USE IN RANCOF *************************
    RANREF=HEIREF/COS(BETREF)
    CALL CLUMAP(HTHETA,LEVBUG)
    OPEN(UNIT=10,FILE='FPSQI.INP',STATUS='OLD')
```

```
      PRINT*,'***********************************************************'
      PRINT*,'SELECT FP TO DISPLAY: KFPSGN, ABORT=-999; ALL FP=-KFPSGN'
      CALL KINPUT(KFPSGN)
      IF(KFPSGN.EQ.-999) STOP 'ABORT'
      KFPDIS=IABS(KFPSGN)
      CALL FPIDOP(NBFPMX,NBFP,HTHETA,TIMDWL,KFPDIS)
      PRINT*,'**** KFPDIS=; NBFPMX=; NBFP=;',KFPDIS,NBFPMX,NBFP, '*****'
      DO 13 IW=0,185
      DO 13 JW=0,185
      SUMARA(IW,JW)=0.
 13   SYNMAP(IW,JW)=0.
C     COMPUTE ARRAY OF RELATIVE TIME WTIME ******************************
      DO 2 KTIM=1,NFFT
 2    WTIME(KTIM)=DELTIM*(KTIM-NFFTH-0.5)
C     INDEX KNPPP=1,NFFX FOR FREQ=[-NFFXH*DELFUT,(NFFXH-1)*DELFUT] *****
C     COMPUTE ARRAY LNPPP,THEN KFF=LNPPP(KNPPP) & WFREQ(KNPPP) *********
      DO 9 KNPPP=1,NFFX
      WFREQ(KNPPP)=DELFUT*FLOAT(KNPPP-NFFXH-1)/FREQO
      INDX=KNPPP+NFFXH
      IF(INDX.GT.NFFX) INDX=INDX-NFFX
 9    LNPPP(KNPPP)=INDX
      CALL WINDOWS(NFFT,4,3.14159,WIND)
      IF(ISURF.EQ.1) CALL SURFMP(KFF,JRAN,Z,1)
      DO 1 KFP=1,NBFP
      CALL BCEDOP(W,NFFT,WTIME,VELOM,ALFM,BETM,WMPOSB,WMPOSC,WMPOSE,
     &WQPOSB,WQPOSC,WQPOSE,WMVELO,JRANQ9,JRANMX,KFPDIS,KFP)
      IF(KFP.NE.KFPDIS.AND.KFPSGN.GT.0) GO TO 1
      WRITE(*,*)'VELOM,  ALFM,  BETM,  XM,  YM,  ZM,  XFP,  YFP'
      WRITE(*,*) VELOM,ALFM,BETM,(WMPOSC(K),K=1,3),(WQPOSC(IW),IW=1,2)
      WMVELO(5)=2.*FREQO/CLIGHT
      IF(KFP.EQ.KFPDIS.AND.ABS(WINIWR-1.).LT.EPS)PRINT*,'window centered
     & at (XFP,YFP)'
      IF(KFP.EQ.KFPDIS.AND.ABS(WINIWR-1.).LT.EPS)CALL MAPWIN(WPAR,4,
     &WDUM,WDUM,WQPOSC(1),WQPOSC(2),137,250,NXMAP,NYMAP)
C     STORE RANGE,BET,EPSMX,AND MU-3DB FOR FICTITIOUS SCATTERERS *******
      DO 7 JRAN=5,JRANMX-9
      WRAN9(JRAN)=WQPOSC(6)+(JRAN-JRANQ9)*DELRAN
      WBET9(JRAN)=ACOS(WMPOSC(3)/WRAN9(JRAN))
      TMPBET=ACOS(WMPOSC(3)/(WRAN9(JRAN)-HDELRA))
      WEPSMX(JRAN)=WBET9(JRAN)-TMPBET
 7    WMU3DB(JRAN)=YMU3DB(HTHETA,WQPOSC(5),WBET9(JRAN),0)
C     THIS IN PREPARATION FOR MAPPING **********************************
C     ENTER CLUTTER INFORMATION FOR FOOTPRINT AT WQPOSC & MAP PARAMETERS
      WRITE(3,*)'JRAMNX=',JRANMX
      CALL CLUFPT(DELRAN,JRANQ9,JRANMX,WQPOSC,WMPOSC,NSCAFP,SMOMAP,
     &WADRLO,WADRUP,WADSMO)
C     PROCESS RANGE CELLS SEQUENTIALLY AND COMPUTE REFLECTIVITY MAP ****
      DO 35 JRAN=5,JRANMX-9
      KSCALO=WADRLO(JRAN)+EPS
      IF(KSCALO.EQ.0) GO TO 35
C     INITIALIZING QUADRATURE ARRAYS=(XD,XQ) ***************************
      DO 11 K=1,NFFX
```

```
      XD(K)=0.
 11   XQ(K)=0.
      WDUM(3)=0.
C     COMPUTE TAYLOR COEFFICIENTS WTAU=(TAU,TAUt,TAUe,TAUtt,TAUee,TAUtm,
C     TAUte,TAUttt,TAUeee,TAUtmm,TAUtee,TAUtme,TAUttm,TAUtte) **********
      CALL TAYLOR(VELOM,BETM,ALFM,WQPOSC(4),WQPOSC(5),WMPOSC(3),HTHETA,
     &DELRAN,JRAN,WTAU)
      WRITE(3,*)'ARRAY WTAU'
      CALL PRTXY(WTAU,1,25,119,3)
C     PREPARATION FOR MAPPING: COMPUTE (NPPPMN,NPPPMX) ON 3DB FOOTPRINT
C     TO LIMIT # CELLS BEING MAPPED TO KNPPP=NPPPMN,NPPPMX *************
      TMU3DB=SIGN(WMU3DB(JRAN),WQPOSC(4))
      NPPPMN=FRQOFMU(FREQO,TMU3DB,WTAU(6),WTAU(10))/DELFUT+NFFXH-5.
      NPPPMX=FRQOFMU(FREQO,-TMU3DB,WTAU(6),WTAU(10))/DELFUT+NFFXH+5.
      IF(ABS(WQPOSC(4)).LT.WMU3DB(JRAN)) NPPPMX=FRQOFMU(FREQO,-WQPOSC(4)
     &,WTAU(6),WTAU(10))/DELFUT+NFFXH+5.
C     COMPUTE MU AT CENTER OF CELLS FOR INCREASING DOPPLER *************
      DO 17 K=NPPPMN-1,NPPPMX+1
 17   WMUMDR(K)=SIGN(1.,WQPOSC(4))*FRTOMU(WTAU,WFREQ(K))
C     COEFFICIENT FOR RANGE NORMALIZATION: RANCOF ***********************
      RANCOF=(WRAN9(JRAN)/RANREF)**4
      KSCAUP=WADRUP(JRAN)+EPS
      DO 30 KTHSCA=KSCALO,KSCAUP
      KADSMO=WADSMO(KTHSCA)+EPS
C     INFORMATION ON SCATTERER,WTSMO=(AMPSCA,FASSCA,MUSCA,JRAN-SCA,ZSCA)
      DO 31 KT=1,5
 31   WTSMO(KT)=SMOMAP(KADSMO-4+KT)
C     ANTGAIN(t)=WAGAIN(1)+WAGAIN(2)*t+WAGAIN(3)*t^2 *******************
C     WAGAIN IS OUTPUT OF SCAGAN,INPUTS ARE DOVL,HTIMDW,WMPOS=(BEG.
C     &CENT,END),WQPOSC,WTSMO=(AMP,FAS,MU,JRAN,ZSCA),JRANQ9,DELRAN ******
      CALL SCAGAN(IDDISH,DOVL,HTIMDW,WMPOSB,WMPOSC,WMPOSE,WQPOSB,WQPOSC,
     &WQPOSE,WTSMO,JRANQ9,DELRAN,WAGAIN)
      CALL TSIGNAL(OMEGO,NFFT,RANCOF,XIFXMT,WTSMO,WTAU,WTIME,WAGAIN,TXD,
     & TXQ)
      DO 33 KDQ=1,NFFT
      XD(KDQ)=XD(KDQ)+TXD(KDQ)
 33   XQ(KDQ)=XQ(KDQ)+TXQ(KDQ)
 30   CONTINUE
      CALL DERAMP(OMEGO,WTAU(4),NFFT,XD,XQ,WTIME)
      DO 37 K=1,NFFT
      XD(K)=XD(K)*WIND(K)
 37   XQ(K)=XQ(K)*WIND(K)
      CALL FFT(-NBITFX,1,1,1,XD,XQ)
C     COMPUTE ARRAY OF ANTENNA GAIN AT CENTER OF CELLS *****************
      DO 39 KNPPP=NPPPMN,NPPPMX
      SINANG=DSQRT(1.+EPS-(DSIN(WQPOSC(5))*DSIN(WBET9(JRAN))*DCOS(WMUMDR
     &(KNPPP))+DCOS(WQPOSC(5))*DCOS(WBET9(JRAN)))**2)
 39   WAGNCL(KNPPP)=DISH(SINANG,IDDISH,DOVL)**2
C     (XD,XQ) NEED TO BE SCALED FOR RANGE AND ANTENNA GAIN AT KNPPP ****
C     APEXTRAP COMPUTES WWAPEX AT CENTERS OF CELLS,*********************
C     WWAPEX=(KNPPP;X1,Y1,X2,Y2,X3,Y3,X4,Y4,AREA) **********************
      CALL APEXTRAP(NFFX,WBET9(JRAN),DELRAN,WQPOSC(4),WEPSMX(JRAN),
     &NPPPMN,NPPPMX,WTAU,WMPOSC,WMUMDR,WWAPEX)
```

```
      DO 41 KNPPP=NPPPMN,NPPPMX
      KFF=LNPPP(KNPPP)
      SCAFAC=1./(RANCOF*WAGNCL(KNPPP))
      XD(KFF)=XD(KFF)*SCAFAC
      XQ(KFF)=XQ(KFF)*SCAFAC
41    XM(KFF)=SQRT(XD(KFF)**2+XQ(KFF)**2)
C     FFT CELLS ARE MAPPED EXCEPT (BELOW THRESH.or.OUT FOOTPRT) KNPPP **
C     GOES:(NPPPMN,NPPPMX) ie:(MUMX,MUMN);FOR FFT USE KFF=LNPPP(KNPPP) *
      CALL MAPRGD(NXMAP,NYMAP,NPPPMN,NPPPMX,XM,WPAR,LNPPP,WWAPEX)
35    CONTINUE
1     CONTINUE
      PRINT*,'NXMAP,NYMAP',NXMAP,NYMAP
      FZMX=0.
      DO 60 IW=0,NXMAP-1
      DO 60 JW=0,NYMAP-1
      IF(SUMARA(IW,JW).LT.EPS) GO TO 60
      SYNMAP(IW,JW)=SYNMAP(IW,JW)/SUMARA(IW,JW)
      IF(SYNMAP(IW,JW).LT.FZMX) GO TO 60
      FZMX=SYNMAP(IW,JW)
      IWMX=IW
      JWMX=JW
60    CONTINUE
      PRINT*,'FZMX=;  IWMX=;  JWMX=; ',FZMX,IWMX,JWMX
      OPEN(UNIT=7,FILE='DOPXYG.GRD',STATUS='OLD')
      WRITE(7,'("DSAA")')
      WRITE(7,'(I4,1X,I4)')NXMAP,NYMAP
      WRITE(7,'(I4,1X,I4)')0,NXMAP-1
      WRITE(7,'(I4,1X,I4)')0,NYMAP-1
      WRITE(7,'(I4,1X,I4)')0,INT(FZMX+EPS)
      DO 69 JW=0,NYMAP-1
      WRITE(7,62) (SYNMAP(IW,JW),IW=0,NXMAP-1)
      WRITE(7,'( )')
69    CONTINUE
62    FORMAT(10(1XF8.3))
18    FORMAT(8F10.3)
      CLOSE(3)
      CLOSE(7)
      CLOSE(9)
      CLOSE(10)
      STOP
      END
```

Subroutine MAPAZD.FOR

```
      SUBROUTINE MAPAZD(NXMAP,NYMAP,XMVAL,WAX,WAY,WPAR)
      REAL WAX(1),WAY(1),WPAR(6),WWAREA(0:31,0:31)
      REAL SYNMAP(0:137,0:250),SUMARA(0:137,0:250)
      REAL AREPOL
      INTEGER I,IIO,IFAIL,IMX,IW,J,JJO,JMX,JW
```

```
      COMMON /GRDMAP/SYNMAP,SUMARA
      CALL POLONFRA(4,WAX,WAY,WPAR,IIO,JJO,IMX,JMX,AREPOL,WWAREA,0,
     &IFAIL)
C     3-SYSTEMS OF COORDINATES: WPAR define ABS-location of window ;****
C     (IW,JW)=WINDOW-coordinates ; (I,J)=FRAME-coordinates *************
      DO 43 I=0,IMX-1
      DO 43 J=0,JMX-1
      IF(WWAREA(I,J).LT.0.0002) GO TO 43
      IW=IIO+I
      JW=JJO+J
      IF(IW.LT.0.OR.JW.LT.0) GO TO 43
      IF(IW.GT.NXMAP.OR.JW.GT.NYMAP) GO TO 43
      SYNMAP(IW,JW)=SYNMAP(IW,JW)+WWAREA(I,J)*XMVAL
      SUMARA(IW,JW)=SUMARA(IW,JW)+WWAREA(I,J)
 43   CONTINUE
      RETURN
      END
```

Subroutine MAPRGD.FOR

```
      SUBROUTINE MAPRGD(NXMAP,NYMAP,NPPPMN,NPPPMX,XM,WPAR,LNPPP,WWAPEX)
      REAL WPAR(6),WWAREA(0:15,0:15),WAX(4),WAY(4),WWAPEX(128,9),
     &XM(1),WAXMN,WAXMX,WAYMN,WAYMX
      INTEGER KA,KAPX,KFF,KNPPP,KMNMX,LNPPP(1)
      DO 45 KNPPP=NPPPMN,NPPPMX
      KFF=LNPPP(KNPPP)
      IF(XM(KFF).LT.WPAR(2)) GO TO 45
      KA=0
      DO 40 KAPX=1,7,2
      KA=KA+1
      WAX(KA)=WWAPEX(KNPPP,KAPX)
 40   WAY(KA)=WWAPEX(KNPPP,KAPX+1)
C     CALL POLONFRA ONLY IF TRAPEZOIDS ARE WITHIN THE SPECIFIED WINDOW *
      WAXMN=WAX(1)
      WAYMN=WAY(1)
      WAXMX=WAX(1)
      WAYMX=WAY(1)
      DO 5 KMNMX=1,4
      WAXMN=AMIN1(WAXMN,WAX(KMNMX))
      WAXMX=AMAX1(WAXMX,WAX(KMNMX))
      WAYMN=AMIN1(WAYMN,WAY(KMNMX))
 5    WAYMX=AMAX1(WAYMX,WAY(KMNMX))
      IF(WAXMN.LE.WPAR(3).OR.WAXMX.GE.WPAR(5)) GO TO 45
      IF(WAYMN.LE.WPAR(4).OR.WAYMX.GE.WPAR(6)) GO TO 45
      IF(XM(KFF).GE.10.) CALL MAPAZD(NXMAP,NYMAP,XM(KFF),WAX,WAY,WPAR)
 45   CONTINUE
      RETURN
      END
```

INITIAL DISTRIBUTION

	No. of Copies
U.S. Army Materiel System Analysis Activity ATTN: AMXSY-MP (erbert Cohen) Aberdeen Proving Ground, MD 21005	1
IIT Research Institute ATTN: GACIAC 10 W. 35th Street Chicago, IL 60616	1
AMSMI-RD	1
AMSMI-RD-CS-R	15
AMSMI-RD-CS-T	1
AMSMI-RD-AS	1
AMSMI-RD-AS, A.H. GREEN	4
AMSMI-GC-IP, Mr. Fred Bush	1