

# **Design Consideration and Implementation of the LCX Polarimetric Scatterometer (POLARSCAT)**

**M. Ali Tassoudji  
K. Sarabandi  
F.T. Ulaby**

**June 1989**

**Report 022486-T-1**

Sign  
UHR 1148

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>System Design</b>	<b>2</b>
2.1	Design Goals . . . . .	2
2.2	Design Considerations of a Network Analyzer Based Scatterometer . . . . .	3
<b>3</b>	<b>Detailed System Description</b>	<b>6</b>
3.1	General . . . . .	6
3.2	Network Analyzer and Control System . . . . .	6
3.3	Amplifier and Pulsing Network . . . . .	8
3.4	Microwave Circuitry . . . . .	11
3.5	Antennas . . . . .	16
3.5.1	Design . . . . .	16
3.5.2	Patterns . . . . .	17
<b>4</b>	<b>Calibration and Measurement Accuracy</b>	<b>22</b>
4.1	Introduction . . . . .	22
4.2	System Transfer Function . . . . .	25
4.3	Calibration . . . . .	26
4.4	Measurement Accuracy . . . . .	29
<b>5</b>	<b>Measurement Modes</b>	<b>35</b>
5.1	Point Target . . . . .	35
5.2	Surface . . . . .	36
5.3	Volume . . . . .	37
5.4	Polarimetry . . . . .	39
<b>A</b>	<b>APPENDIX A. Photographs of the system</b>	<b>42</b>



## List of Figures

1	Network analyzer operation as a scatterometer. . . . .	3
2	Single antenna network analyzer based scatterometer. . . . .	3
3	Block diagram of the LCX scatterometer. . . . .	6
4	Diagram of the amplifier and pulsing network. . . . .	9
5	Time domain response of the system with a sphere in the chamber; (a)without switching, (b) with switching. . . . .	10
6	Switch control pulses for transmit and receive, pulsing period is 225 ns, the on-pulse length is 90 ns. . . . .	11
7	Block diagram of the L-band RF unit. . . . .	13
8	Block diagram of the C-band RF unit. . . . .	13
9	Block diagram of the X-band RF unit. . . . .	14
10	Schematic of a square pyramidal horn antenna. . . . .	16
11	L-band vertical polarization antenna pattern. . . . .	19
12	L-band horizontal polarization antenna pattern. . . . .	19
13	C-band vertical polarization antenna pattern. . . . .	20
14	C-band horizontal polarization antenna pattern. . . . .	20
15	X-band vertical polarization antenna pattern. . . . .	21
16	X-band horizontal polarization antenna pattern. . . . .	21
17	Geometry of scattering of a plane wave from a particle. . . . .	23
18	Simplified block diagram of a dual polarized radar system. . . . .	26
19	Automatic radar cross section measurement setup. . . . .	31
20	Geometry of scattering of a plane wave from a long, thin cylinder. . . . .	31
21	Geometry of a wire-mesh . . . . .	32
22	Radar cross section (hh) versus frequency of a cylinder with L=30.48 cm and D=1.625 mm, (—)measured and (- - -)theory. . . . .	32

23	Radar cross section (vv) versus frequency of a cylinder with L=30.48 cm and D=1.625 mm, (—)measured and (---)theory. . . . .	33
24	Radar cross section (hv) versus frequency of a cylinder with L=30.48 cm and D=1.625 mm, (—)measured and (---)theory. . . . .	33
25	Relative phase $S_{hh}$ (to $S_{vv}$ ) versus frequency of a cylinder with L=30.48 cm and D=1.625 mm, (—)measured and (---)theory. . . . .	34
26	Relative phase $S_{hv}$ (to $S_{vv}$ ) versus frequency of a cylinder with L=30.48 cm and D=1.625 mm, (—)measured and (---)theory. . . . .	34
27	Geometry used in the surface target measurements. . . . .	40
28	Geometry used in the volume target measurements. . . . .	40
29	Photograph of the LCX Antennas. . . . .	43
30	Photograph of the C-band RF Unit. . . . .	44
31	Photograph of the X-band RF Unit. . . . .	44

## List of Tables

1	HP 8753 frequency sweep range for the LCX scatterometer. . . . .	7
2	Amplifier and pulsing network components. . . . .	10
3	L-band RF unit components. . . . .	14
4	C-band RF unit components. . . . .	15
5	X-band RF unit components. . . . .	15
6	LCX square horn antenna dimensions. . . . .	17
7	Characteristic of LCX antennas. . . . .	18
8	Gain and beamwidth variations of LCX antennas with frequency. . . . .	18

## **1 Introduction**

The LCX polarimetric scatterometer (POLARSCAT) is a radar system designed to measure the backscattering characteristics of extended and discrete targets, over the frequency bands of L, C, and X. It measures the magnitude and phase of the scattered electric field for all linear polarization combinations.

This system is mainly designed to operate from a truck-mounted boom. A platform at the top of the boom carries the antennas and RF equipment, and the control and processing units are housed in a control room on the bed of the truck. Communication between the RF units mounted on the boom and the rest of the system is accomplished via control and RF cables. The control cable is used to control the antenna positioner, TV camera, and the transmit/receive polarization.

In this report the design and implementation of the LCX scatterometer is discussed. The text is divided into four sections: System Design, Detailed System Description, Calibration and Measurement Accuracy, and Measurement modes. Under the system design section, the design goals and how we can satisfy them with a network analyzer based scatterometer are discussed. The section entitled Detailed System Description gives a complete description of all the units used in the system, the network analyzer, pulsing network, microwave circuitry, and antennas. The Calibration and Measurement Accuracy section explains a new calibration procedure and its accuracy for polarimetric scatterometers. Finally, the measurement routines used by the scatterometer are discussed in the Measurement Modes section.

## **2 System Design**

Before starting to discuss the design of a network analyzer based scatterometer, a brief section about the desired design goals is needed.

### **2.1 Design Goals**

The overall objective of designing the LCX polarimetric scatterometer system was to measure the scattering amplitude of point and distributed targets with great accuracy. This includes measuring the amplitude and phase of the signal backscattered from the target for any of the four linear polarization configurations. The design goals include:

- Coverage of L, C, and X-Band frequencies
- System mobility for indoor and outdoor experiments
- Small antenna size and weight to be mounted on a boom truck
- Absolute measurement of scattering coefficient of distributed targets
- Phase measurement capability
- Adequate dynamic range for targets of interest
- High sensitivity to measure targets with small RCS
- Ranging capability to separate targets by range
- Sufficient spatial resolution
- High speed switching between polarizations
- Automated switching and data acquisition
- Real time processing of the measurements

By using a vector network analyzer as the basic signal processor of the scatterometer, we can satisfy most of the requirements stated above.

## 2.2 Design Considerations of a Network Analyzer Based Scatterometer

In general, network analyzers are designed to make S-parameter measurements in the frequency domain. They can be configured to operate as a scatterometer system (Fig. 1) In the scatterometer configuration the RF signal is coupled to free space by an antenna. The signal reflected from the target is then picked up by the receive antenna, and is compared with a sample of the transmitted signal. This system can also be changed to a single antenna system using a circulator as shown in Fig. 2.

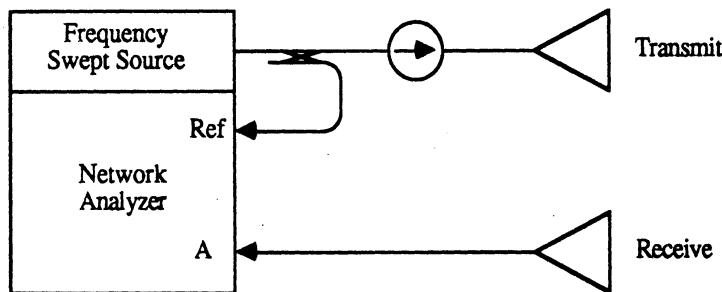


Figure 1: Network analyzer operation as a scatterometer.

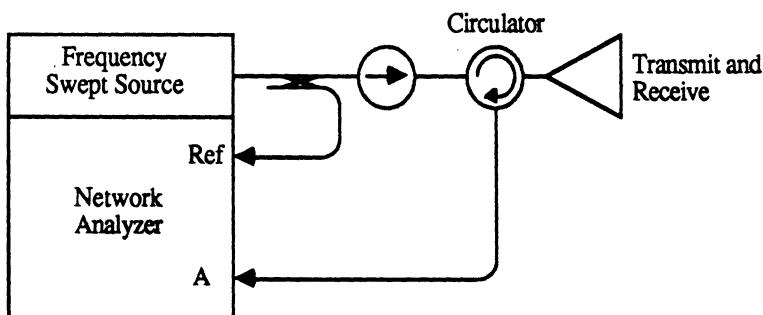


Figure 2: Single antenna network analyzer based scatterometer.

Network analyzer based scatterometers have measurement capabilities that are not provided by traditional scatterometers. With the recent development of more powerful network analyzers,

such as the HP 8753 and HP 8720, the advantages are more impressive. They have the capability of computing the inverse Fourier Transform of the measured data to give the time domain response, which displays reflections from the target as a function of time or distance. The time domain response gives great insight into where in the range dimension the reflections occur. The time domain gating feature can also be used to analyze the measured response and reduce the effects of unwanted signals. In addition, the recent HP network analyzers can be fully automated using an external computer over the HP-IB(Interface Bus). A large amount of data reduction associated with a measurement can be performed using a computer, while the system remains flexible enough to be used for a variety of applications.

One of our design requirements is to be able to measure targets with a wide range of radar cross sections(RCS). In practice the limitation is on measuring the targets with small RCS. This minimum detectable target for a given scatterometer at a certain range is limited by three major factors. The first factor is the thermal noise level, which is the absolute minimum detectable signal and can be determined by the product of the noise figure and bandwidth. The dynamic range of the sampler in the network analyzer is the second factor. This problem arises when a single antenna system is used, or when there is insufficient isolation between the transmit and receive antenna. In such cases part of the transmitted signal returns to the receiver and sets the minimum detectable signal level, which can be obtained from dividing the returned signal by the dynamic range of the sampler. Finally, the third limiting factor is the effect of multiple reflection at the target range. Usually in radar systems there are impedance mismatches at interfaces between different RF components, antenna input ports, and antenna to free space transition.

Since the network analyzer is a coherent system its bandwidth is very narrow, hence thermal noise usually is not a limiting factor. Dynamic range and multiple reflection problems limit the performance of the scatterometer, and they can be improved by increasing the isolation between the transmit and receive paths of the system. A two antenna system can solve these problems, but antenna size limitations and small beamwidth (spatial resolution) requirement force us to use a single antenna system. Deploying a pulsing network and including isolators in the microwave

circuitry can somewhat eliminate the antenna mismatch and multiple reflections. A detailed description of the system is given in the next section of this report.

Another design requirement is that the antennas should be mounted on a platform at the top of a boom truck, with rotatable positioner, where there are limitations in space and weight handling. Thus, the network analyzer and its peripherals must be operated from the base platform on the bed of the truck. In this case the RF transmit and receive signals should pass through long coaxial cables that run from the network analyzer to the antennas. In order to avoid the high losses in cables at C and X-Band frequencies, a lower IF is up-converted to the desired RF band using a single side-band up-converter. Then the received signal is down-converted by the same local oscillator used for up-conversion to maintain phase coherence.

### 3 Detailed System Description

#### 3.1 General

Figure 3 presents an overall system block diagram. It divides the system into 4 subsections: network analyzer and control network, amplifier and switching network, microwave circuitry and antennas. Description of each subsystem is presented in the following sections.

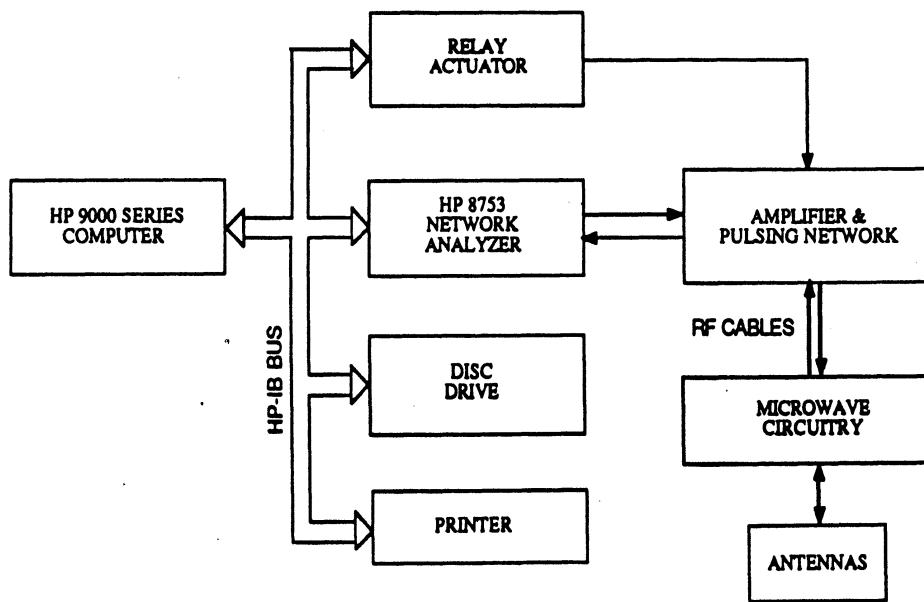


Figure 3: Block diagram of the LCX scatterometer.

#### 3.2 Network Analyzer and Control System

The HP 8753A is a high performance RF vector network analyzer used to measure reflection and transmission parameters. It integrates a high resolution synthesized RF source and a dual channel receiver to measure and display magnitude and phase responses of active and passive RF networks. It has the capability of transforming measured data from the frequency domain to the time domain.

	L-Band	C-Band	X-Band
Center Frequency	1.5 GHz	1.75 GHz	1.5 GHz
Frequency Bandwidth	300 MHz	500 MHz	1.0 GHz

Table 1: HP 8753 frequency sweep range for the LCX scatterometer.

It consists of three receiver inputs R, A and B with identical samplers. The R input is usually used as a reference for phase-locking the source. A directional coupler is used in our system to provide a reference signal for the R input. A minimum of -35 dbm power is required for phase-locking to occur.

The built-in synthesized source generates a swept RF signal in the range of 300 KHz to 3.0 GHz. The frequencies used in the HP 8753A for the LCX system are listed in Table 1. The X-band scatterometer is operated with a bandwidth(BW) of 500 MHz in field experiments because a BW of 1.0 GHz would require twice the number of points to avoid aliasing in the time domain presentation.

The RF output power of the network analyzer ranges from -5 to 20 dBm. This power should be adjusted according to the length of RF cables used in the system. Another factor that determines the amount of transmit power is the use of the pulsing network which will be described later in the report.

The HP 8753A is equipped with a remote programming interface using the HP-IB. This provides a controlling computer to send commands or instructions to and receive data from the network analyzer. Several output modes are available for outputting data. We use a polar format for data transfer that provides us with the magnitude and phase of the signal.

Automatic control of all equipment is accomplished by the HP 9386 computer. The HP 8753A system and peripheral equipment, including a disc drive and printer, are all interfaced with the computer, providing automated measurement, real-time data reduction, hard copy output and data storage. The computer also controls a HP 59306A relay actuator which selects the frequency and polarization of the measurement. The 10 ms switching time on the relays allows us to make

fast polarimetric measurements.

### 3.3 Amplifier and Pulsing Network

To detect small targets or targets that are far away from the antenna, we needed to add an amplifier in the receive path of the system. In addition to the signal reflected from the target, the short-range returns from the antenna and circulator leakage are also amplified, and may result in saturation of the receiver. To reduce the level of these direct returns, a pulsing network is used.

With this network, during the time that the signal is transmitted, the receiver is turned off, and when the return from the target is expected the receiver is turned on and the transmitter is switched off. Because this switching is done at a rate which is much higher than the bandwidth of the receiver in the network analyzer, the analyzer does not sense that the incoming signal is pulsed and measures it as CW.

The block diagram of the components used in our pulsing network is shown in Figure 4 and their characteristics are listed in Table 2. The two high-speed microwave switches are placed in the transmit and receive arms. They are driven by two pulse generators in a way that one is triggered by the other. An isolator and a high pass filter are used to remove switching transients that can create adverse effects on the response of the network analyzer. It should be noted that the pulsing network is not interfaced with the network analyzer and is operated independent of the analyzer operating sequence.

The effect of pulsing was tested with the X-band system when operated inside an anechoic chamber. An 8-inch diameter sphere was placed in the chamber on a styrofoam pedestal about 15 meters away from the antenna. Figure 5 shows the time domain response of the scene as displayed by the network analyzer. Without switching (upper trace) there is a large return at the start of the trace caused by the antenna reflections. The return from the sphere is some 100 nanoseconds later. The noise floor is about 23 dB below the sphere, mostly attributable to thermal(random) noise. Switching is timed in such a way that the antenna reflection is decreased by 50 dB (lower trace), the transmitted power can now be increased without overdriving the receiver. The signal

to noise ratio is now improved to 35 dB.

Figure 6 shows the timing sequence used for the switches. The scale is 20 ns/div. The transmitter is turned on for 90 ns and off for 135 ns, corresponding to 4.4 MHz repetition rate. These parameters were obtained experimentally for optimum response and depend mostly on the spacing between the antenna and the target. For distributed targets these parameters must be chosen more carefully in such a way that no part of the backscattered signal is chopped off.

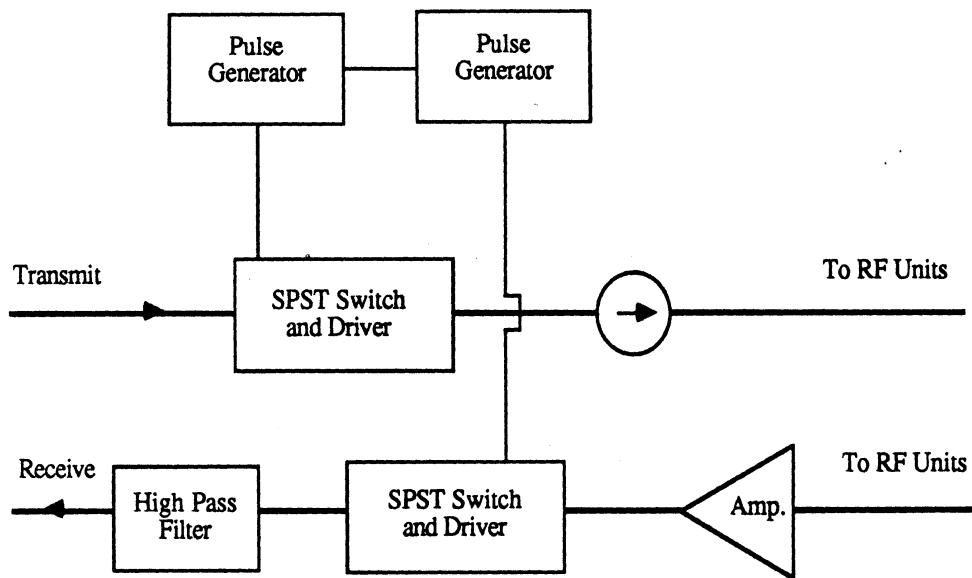


Figure 4: Diagram of the amplifier and pulsing network.

Component	MFG Model No.	Specifications
SPST PIN Diode switch	HP 33132A	Isol. 33dB, Ins. Loss 1dB @ 1-2GHz
Switch driver	HP 33190B	Off 0-0.8V, On 2.5V min.
50 MHz Pulse Generator	HP 8012B	Period 20ns-1s, Trans. time 5ns-0.5s
Amplifier	MITEQ AMMIC 1047	50-2500MHz, 38.7dB, 15V, 150mA power @1dB Comp. 18dBm
High pass filter	Mini-Circuits SHP-1000	1GHz cutoff, -40dB at 0.6GHz
Isolator	UTE CT-2102-OT	Isol>20dB at 1-2GHz, VSWR<1.2

Table 2: Amplifier and pulsing network components.

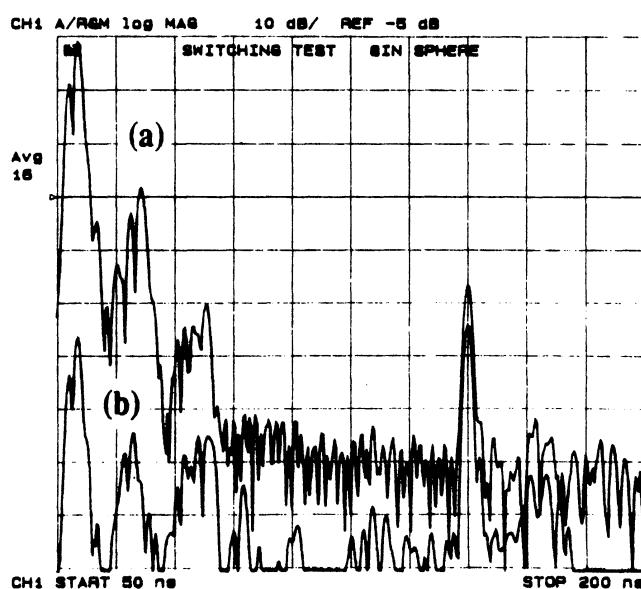


Figure 5: Time domain response of the system with a sphere in the chamber; (a)without switching, (b) with switching.

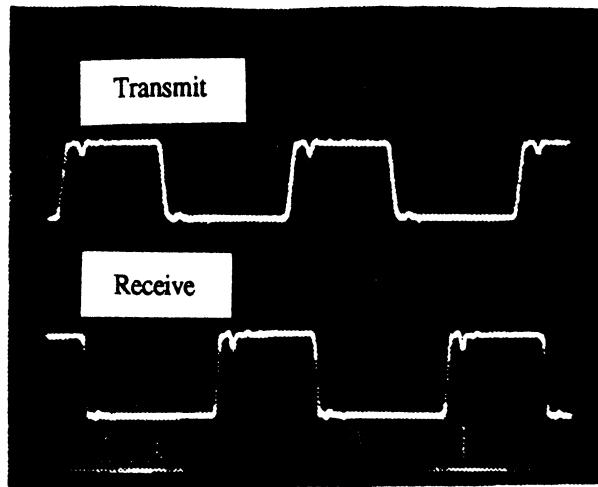


Figure 6: Switch control pulses for transmit and receive, pulsing period is 225 ns, the on-pulse length is 90 ns.

### 3.4 Microwave Circuitry

The microwave circuitry in the LCX scatterometer is designed such that it enables the system to transmit and receive two orthogonal linear polarizations (vertical and horizontal) at the desired frequencies. The diagrams in Figures 7, 8 and 9 illustrate the designs used in the RF units of the three systems.

Because losses in the cables are tolerable at L-band, the base swept frequency of the network analyzer is itself used for transmission. Two switches are placed in the transmit and receive paths to provide us with easy switching between transmit and receive polarizations, which are controlled by the relay actuator in the control unit. The performance specifications of the components used in the L-band RF unit are listed in Table 3.

To avoid high losses at C and X-band frequencies, a lower frequency signal (IF) provided by the network analyzer is up-converted using a local oscillator(LO). This is accomplished using a mixer that produces the sum and difference frequencies ( $f_{RF} = f_{LO} \pm f_{IF}$ ). In the C-band unit ( $f_{LO} - f_{IF}$ ) is used and the sum frequency is rejected with a low pass filter. In the X-band unit

we use a high pass filter that passes ( $f_{LO} + f_{IF}$ ) and rejects the difference frequency. The C and X-band systems are almost identical in design except that one uses the lower side-band frequency and the other the upper side-band.

An amplifier is placed after the mixer to compensate for the conversion loss of the mixer. A power divider is used to split the LO power for use in the down-conversion section. Two separate sets of down-converter sections are used (one for each polarization) to provide the option of receiving both polarizations with two receivers. For the time being a transfer switch located at the end of these two sections selects the receive polarization.

Isolators are distributed throughout the system to help reduce reflections, and hence the system noise level, which arise from the relatively high VSWR of the mixer ports. Because variations in temperature can cause performance changes in the oscillator and the amplifier, a heater with a temperature-controlled unit is placed in both C and X-band RF units to stabilize the temperature.

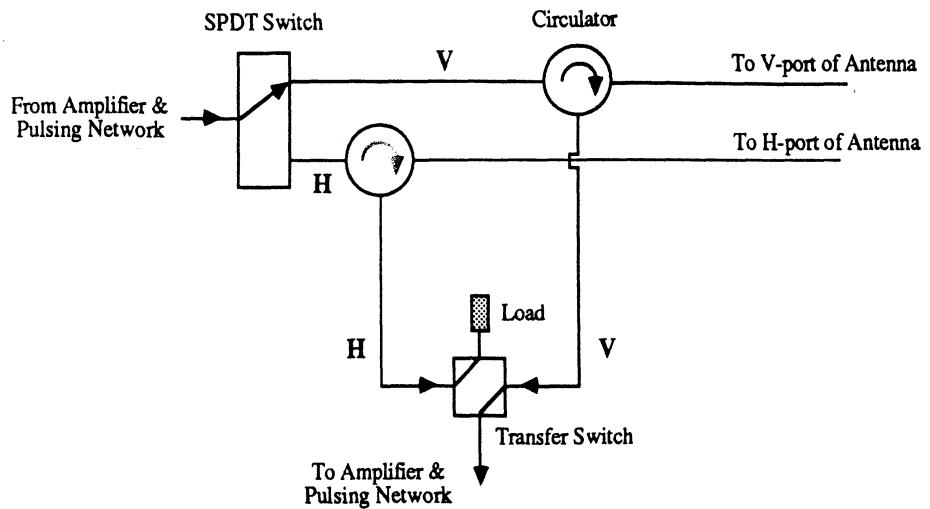


Figure 7: Block diagram of the L-band RF unit.

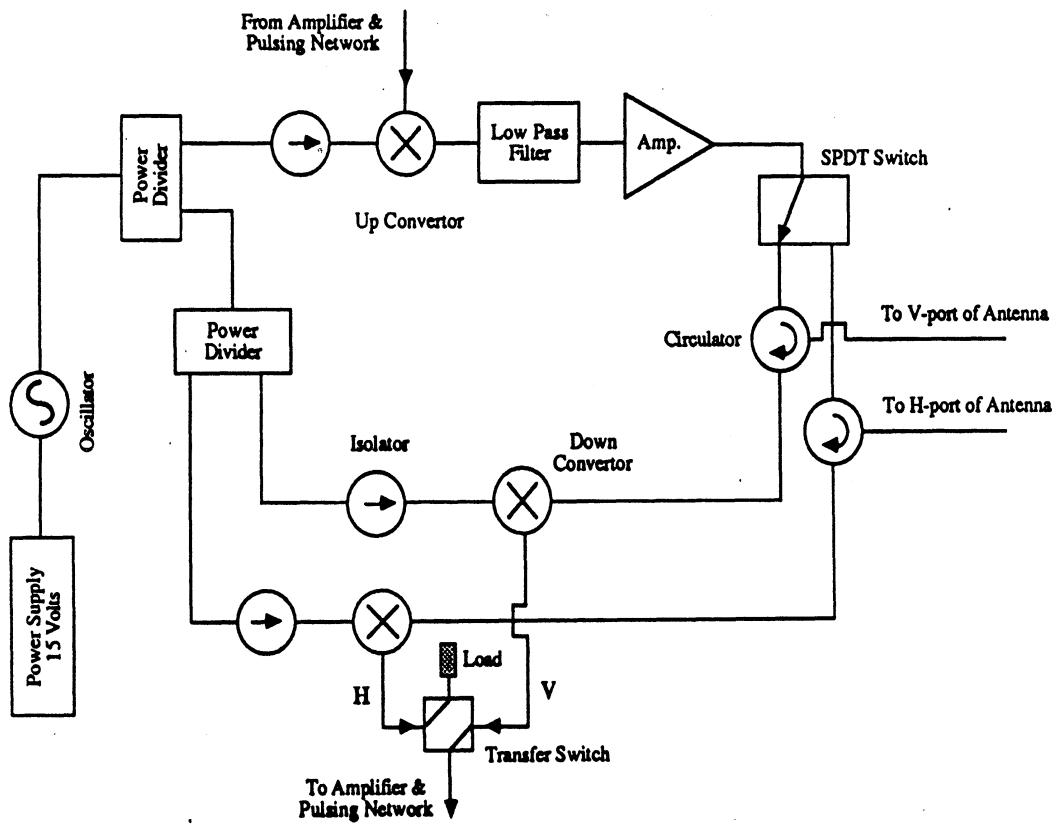


Figure 8: Block diagram of the C-band RF unit.

Component	MFG Model No.	Specifications
Circulator	UTE CT-2104-O	Isol>20dB at 1-2GHz, VSWR<1.2
SPDT switch	Teledyne CS-33S10	Switch time 10ms, 28V, 80mA
Transfer switch	Transco 715C70100	Switch time 20ms, 28V, 120mA

Table 3: L-band RF unit components.

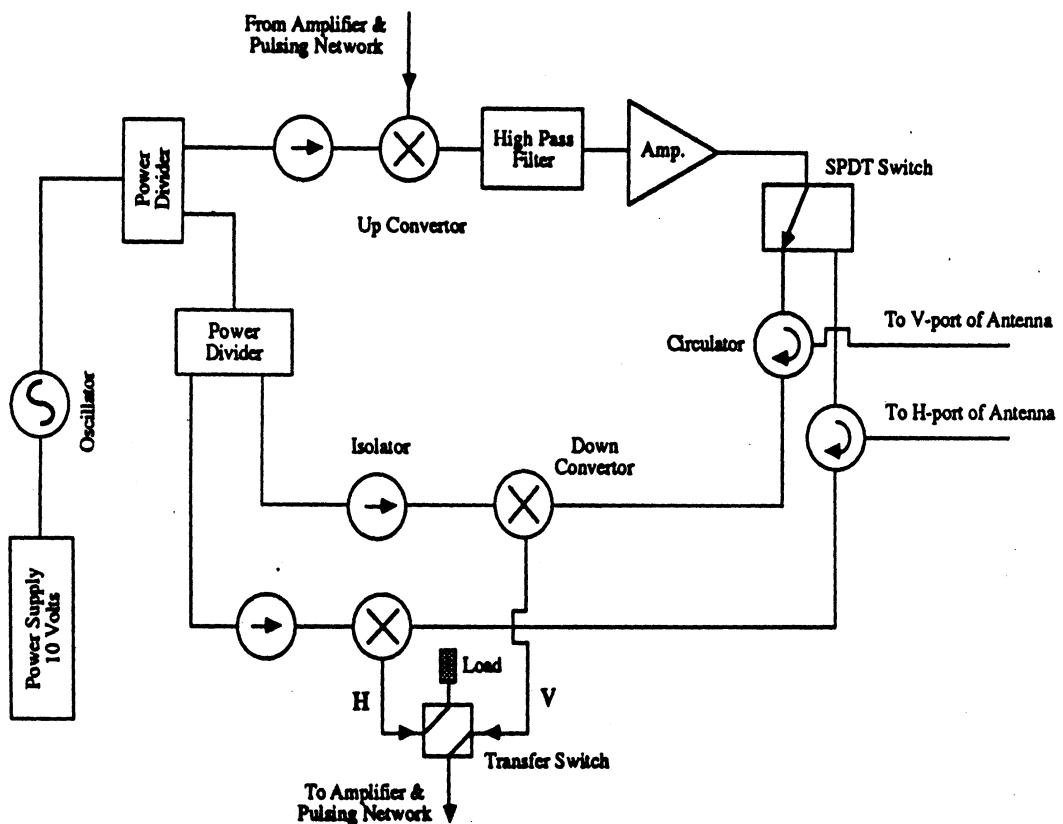


Figure 9: Block diagram of the X-band RF unit.

Component	MFG Model No.	Specifications
Oscillator	EMF Systems 251003	6.5GHz, +17.7dBm, 15V, 250mA
Mixer	Watkins Johnson M14	4-8GHz, Conv loss <7dB
SPDT switch	Teledyne CS-33S10	Switch time 10ms, 28V, 80mA
Transfer switch	Transco 715C70100	Switch time 20ms, 28V, 120mA
Low pass filter	Microlab/FXR LA-60F	6GHz cutoff, -32dB at 7.5GHz
Power divider	ARRA 5200-2	4-8GHz, Iso >18dB, VSWR <1.6
Circulator	UTE CT-4428-O	4-8GHz, Isolation >20dB,
Isolator	UTE CT-4426-OT	Ins Loss <.4dB, VSWR <1.2
Amplifier	Avantek AFT-6263	2-6GHz, 28.7dB, 15V, 282 mA power @1dB Comp. 22.2dBm

Table 4: C-band RF unit components.

Component	MFG Model No.	Specifications
Gunn Oscillator	Custom made	8.0GHz, +16dBm, 10V, 586mA
Mixer	Watkins Johnson M77C	8-12.5GHz, Conv loss <7.5dB
SPDT switch	Teledyne CS-33S10	Switch time 10ms, 28V, 80mA
Transfer switch	Transco 715C70100	Switch time 20ms, 28V, 120mA
High pass filter	Tapered WR-90 WG	8.7GHz cutoff,-30dB at 8GHz
Power divider	ARRA A6200-2	8-12.4GHz, Iso>18dB, VSWR<1.6
Circulator	UTE CT-5158-O	8-12.4GHz, Isolation >20dB,
Isolator	UTE CT-5157-OT	Ins Loss <.4dB, VSWR <1.2
Amplifier	Avantek AFT-12664	6-12GHz, 31dB, 15V, 350 mA power @1dB Comp. 21.5dBm

Table 5: X-band RF unit components.

### 3.5 Antennas

#### 3.5.1 Design

After extensive research, square horn antennas were chosen as the best design that could meet our design goals. Low VSWR constraint and physical size limitation ruled out the use of reflector and array antennas. Also the fact that their cross-pol purity remains constant across the main beam make them particularly suitable for polarimetric measurement of distributed targets.

An orthogonal mode transducer(OMT) feed provides the capability of transmitting and receiving both polarizations. It has two ports (vertical and horizontal) each of which can be used to transmit and receive the signal. The physical dimensions of each antenna were determined by first selecting a certain gain and beamwidth and then optimizing them to get a reasonable size square horn[1]. A schematic of a square pyramidal horn is shown in Figure 10 and the dimensions are given in Table 6.

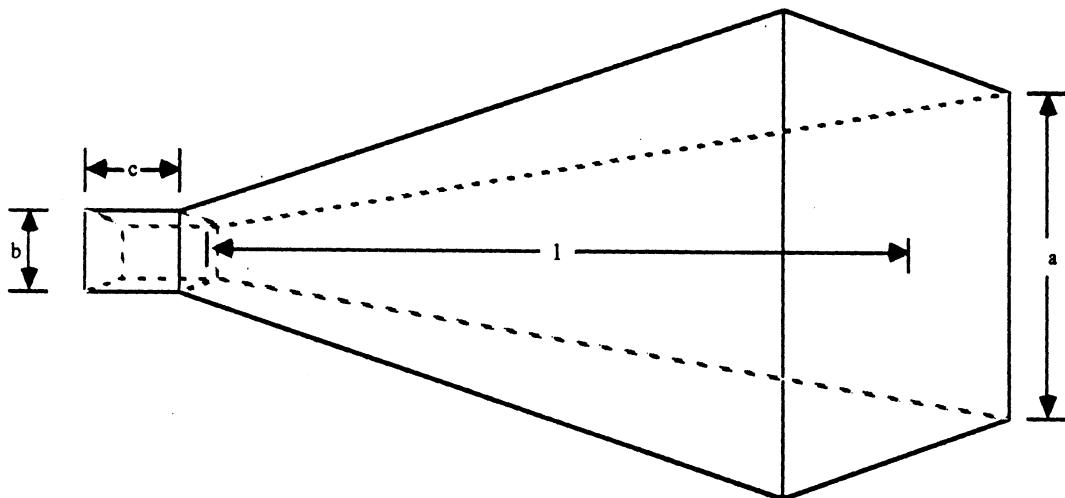


Figure 10: Schematic of a square pyramidal horn antenna.

	L-Band	C-Band	X-Band
Aperture width a(cm)	92.4	43.0	40.6
Waveguide width b(cm)	12.1	4.7	2.03
Waveguide length c(cm)	20.0	5.1	2.5
Antenna length l(cm)	188.7	136.0	142.2

Table 6: LCX square horn antenna dimensions.

### 3.5.2 Patterns

Antenna pattern measurements were performed with the HP 8510 network analyzer in the anechoic chamber. The source antenna was positioned at the neck of the chamber. The test antenna was placed on a wooden pedestal attached to a stepper-motor centered at the same height as the source antenna. The network analyzer and the stepper-motor were controlled by a HP 9836 computer.

The output of the HP 8510 was sent to a source antenna via a low loss cable. An amplifier was required at the source antenna to increase the signal to noise ratio when the X-band antenna was tested. A horn antenna was used as the source antenna due to its large bandwidth. The signal received by the antenna under test was sent back to the network analyzer and the HP 8510 was set up to display the ratio of the received to transmitted signals. Time domain gating was used to eliminate reflections from ground and other directions, such that only the direct signal picked up by the antenna was being measured.

The pattern measurements were performed for vertical and horizontal polarizations in each of the principal planes. The frequencies used were the same as mentioned in Table 7. The data were recorded every 1 degree over the whole pattern and also every 0.1 degree over the main beam, which were used for the illumination integral calculation. The patterns measured for each antenna at their center frequency are shown in Figures 11 through 16. The characteristics of the LCX antennas are given in Table 7. The gain and beamwidth of the antennas are functions of

frequency. Their limits over the bandwidth of operation are listed in Table 8. The azimuth and elevation values are reversed for the horizontal polarization.

	L-Band	C-Band	X-Band
Center Frequency $f_c$ (GHz)	1.5	4.75	9.5
Frequency bandwidth(MHz)	300	500	1000
OMT frequency range(GHz)	1.35-1.72	4.5-5.0	8.5-10.0
Gain at $f_c$ (dB)	22.1	25.3	29.5
Cross-pol isolation at $f_c$ (dB)	22.1	26.3	28.7
V-pol beamwidth El,Az at $f_c$ (deg)	12.0,15.2	8.0,10.4	5.4,6.5
Far field distance at $f_c$ (meter)	8.5	5.8	10.5

Table 7: Characteristic of LCX antennas.

	L-Band	C-Band	X-Band
Gain variation over bandwidth	20.8-23.4	25.5-27.5	26.7-31.5
V-pol elevation beamwidth(deg)	11.1-12.9	7.7-8.4	5.2-5.5
V-pol azimuth beamwidth(deg)	13.8-16.5	10.1-10.8	6.5-6.9

Table 8: Gain and beamwidth variations of LCX antennas with frequency.

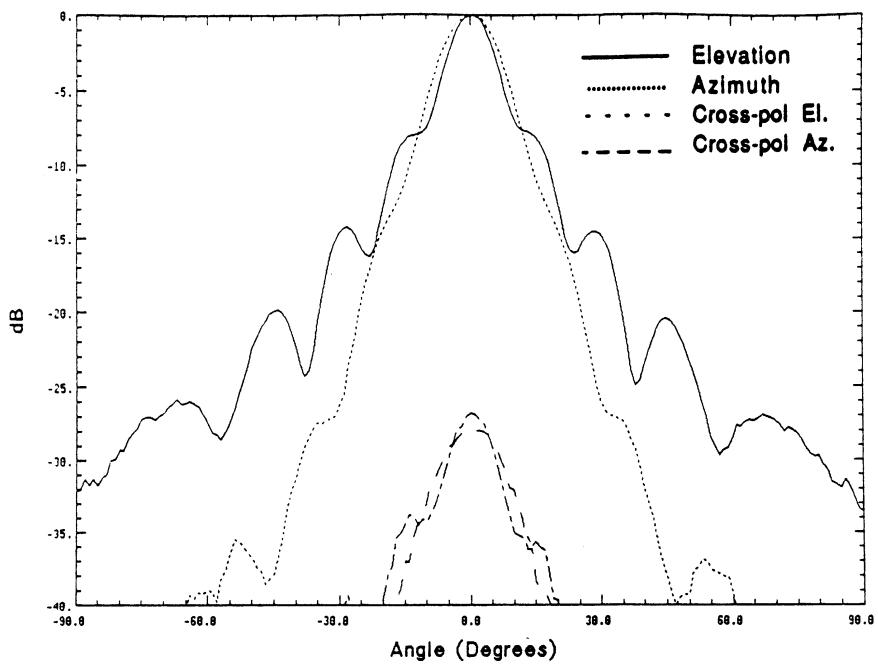


Figure 11: L-band vertical polarization antenna pattern.

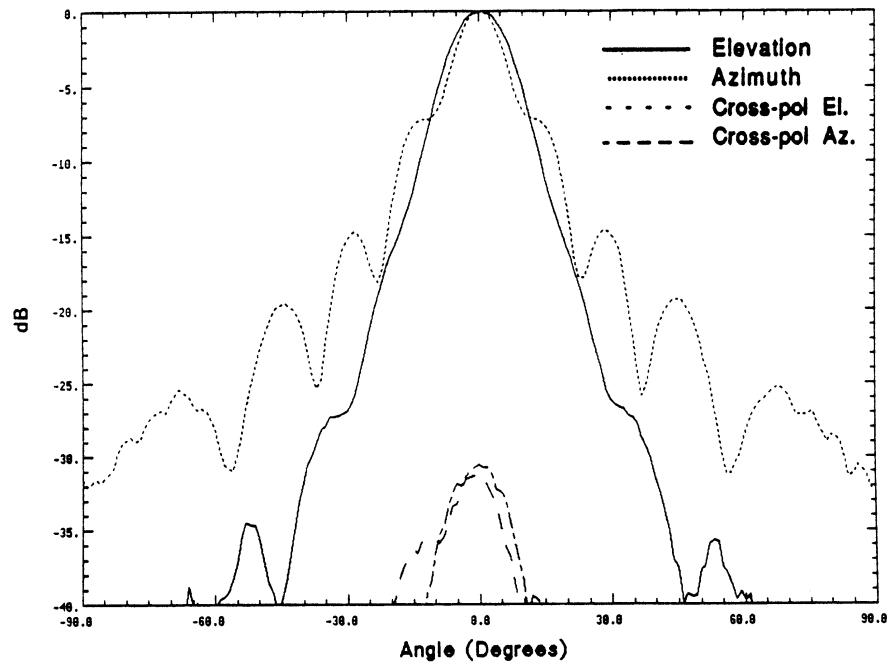


Figure 12: L-band horizontal polarization antenna pattern.

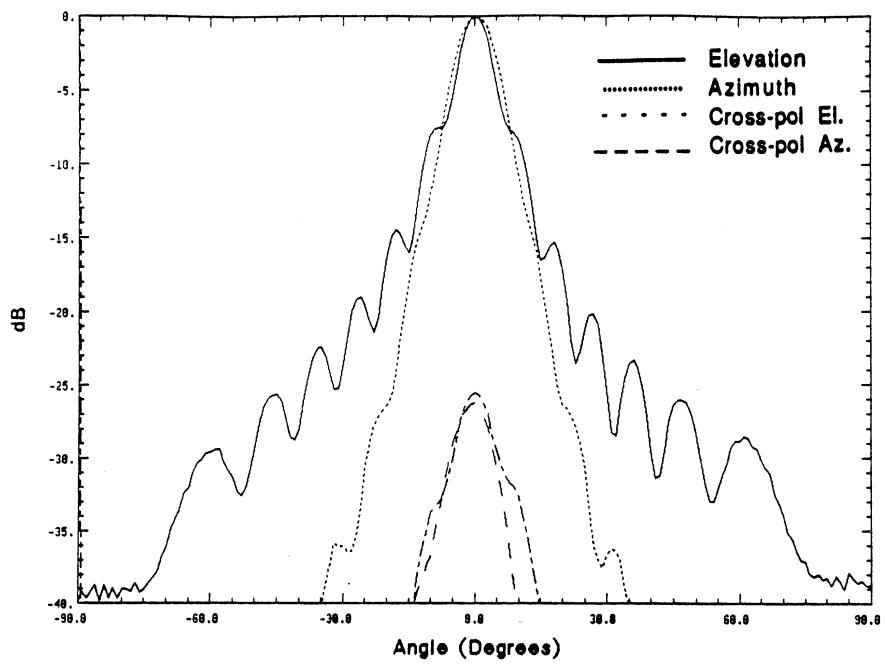


Figure 13: C-band vertical polarization antenna pattern.

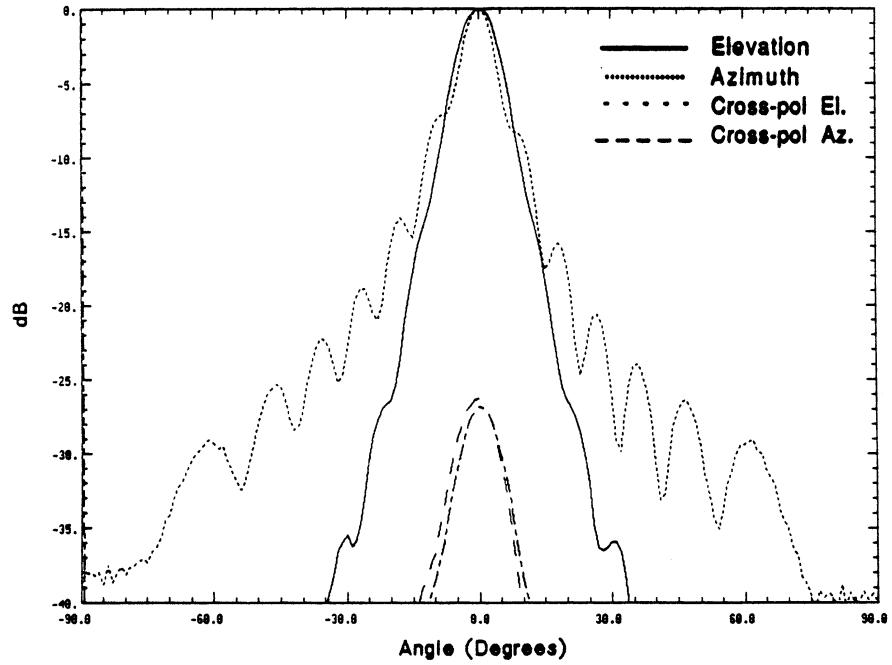


Figure 14: C-band horizontal polarization antenna pattern.

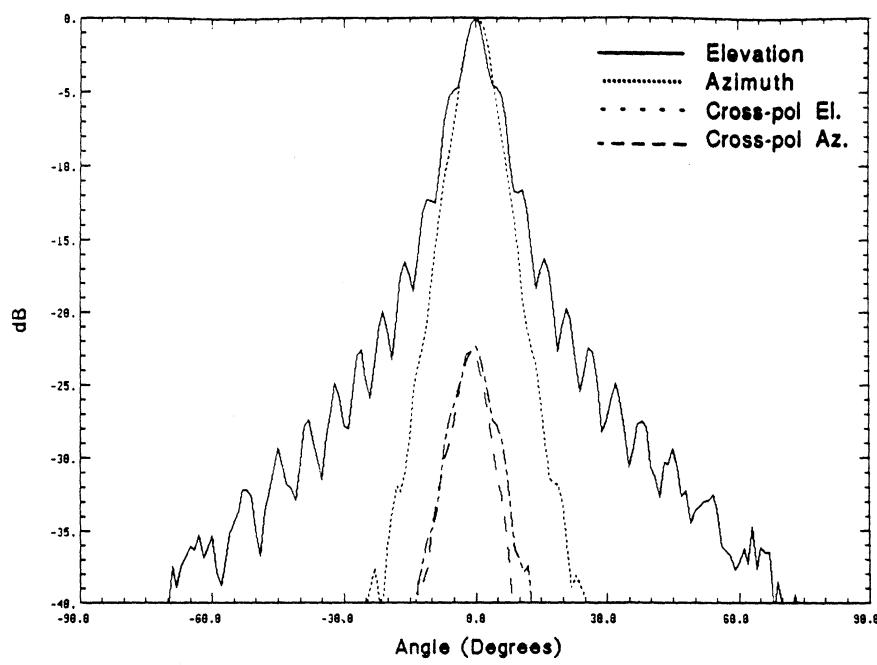


Figure 15: X-band vertical polarization antenna pattern.

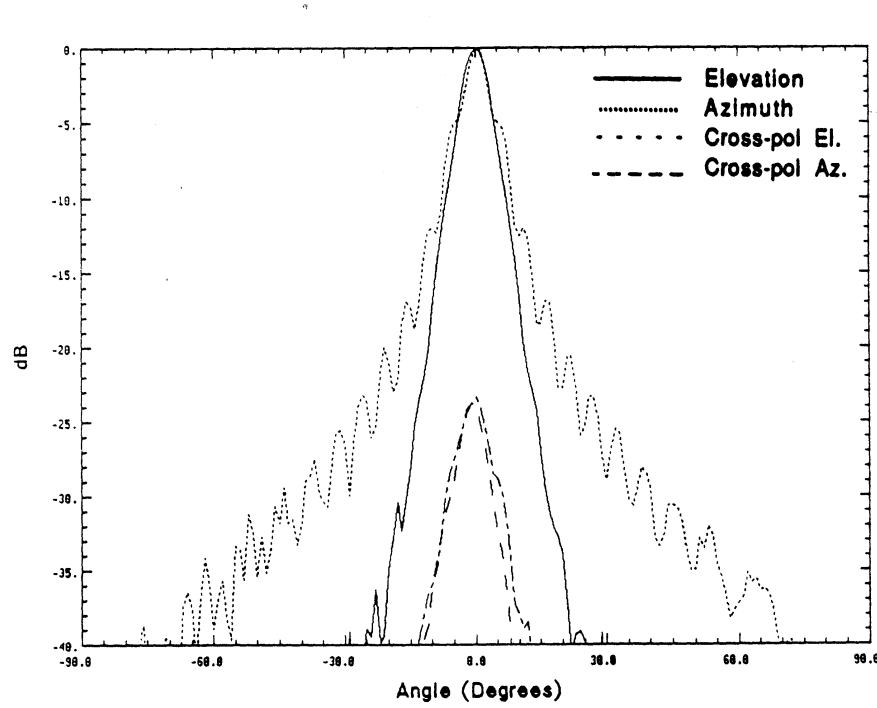


Figure 16: X-band horizontal polarization antenna pattern.

## 4 Calibration and Measurement Accuracy

### 4.1 Introduction

A polarimetric radar is a phase-coherent instrument used to measure the polarization scattering matrix  $\mathbf{S}$  of point or distributed targets. The matrix  $\mathbf{S}$  relates the field  $\mathbf{E}^s$  scattered by the target to the field  $\mathbf{E}^i$  of a plane wave incident upon the target [4, p.1087],

$$\mathbf{E}^s = \frac{e^{-ikr}}{r} \mathbf{S} \mathbf{E}^i, \quad (1)$$

where  $r$  is the distance from the center of the target to the point of observation and  $k$  is the wave number. For a plane wave incident upon the particle in the direction  $\hat{\mathbf{k}}_i$ , its electric field vector may be written in terms of vertical and horizontal polarization components,  $E_v^i$  and  $E_h^i$ , using the coordinate system  $(\hat{\mathbf{v}}_i, \hat{\mathbf{h}}_i, \hat{\mathbf{k}}_i)$  shown in Fig. 17,

$$\mathbf{E}^i = (E_v^i \hat{\mathbf{v}}_i + E_h^i \hat{\mathbf{h}}_i) e^{-ik\hat{\mathbf{k}}_i \cdot \mathbf{r}}, \quad (2)$$

where

$$\hat{\mathbf{v}}_i = \cos\theta_i \cos\phi_i \hat{\mathbf{x}} + \cos\theta_i \sin\phi_i \hat{\mathbf{y}} - \sin\theta_i \hat{\mathbf{z}} \quad (3)$$

$$\hat{\mathbf{h}}_i = -\sin\phi_i \hat{\mathbf{x}} + \cos\phi_i \hat{\mathbf{y}} \quad (4)$$

$$\hat{\mathbf{k}}_i = \sin\theta_i \cos\phi_i \hat{\mathbf{x}} + \sin\theta_i \sin\phi_i \hat{\mathbf{y}} + \sin\theta_i \hat{\mathbf{z}}. \quad (5)$$

In (2), a time dependence of the form  $e^{+i\omega t}$  is assumed and suppressed.

The far-field wave scattered in the direction  $\hat{\mathbf{k}}_s$  is a spherical wave given by

$$\mathbf{E}^s = E_v^s \hat{\mathbf{v}}_s + E_h^s \hat{\mathbf{h}}_s, \quad (6)$$

where  $(\hat{\mathbf{v}}_s, \hat{\mathbf{h}}_s, \hat{\mathbf{k}}_s)$  are defined by the same expressions given in (3) to (5) except for replacing the subscript  $i$  with the subscript  $s$ . For the backscattering case,  $\theta_i + \theta_s = \pi$ ,  $\phi_i + \phi_s = \pi$ ,  $\hat{\mathbf{k}}_s = -\hat{\mathbf{k}}_i$ ,  $\hat{\mathbf{v}}_s = \hat{\mathbf{v}}_i$ , and  $\hat{\mathbf{h}}_s = -\hat{\mathbf{h}}_i$ .

In matrix form, (1) may be rewritten as

$$\begin{bmatrix} E_v^s \\ E_h^s \end{bmatrix} = \frac{e^{-ikr}}{r} \begin{bmatrix} S_{vv} & S_{vh} \\ S_{hv} & S_{hh} \end{bmatrix} \begin{bmatrix} E_v^i \\ E_h^i \end{bmatrix} \quad (7)$$

where

$$\mathbf{S} = \begin{bmatrix} S_{vv} & S_{vh} \\ S_{hv} & S_{hh} \end{bmatrix} \quad (8)$$

is defined in terms of the scattering amplitudes  $S_{mn}$  with  $m$  and  $n$  denoting the polarization ( $v$  or  $h$ ) of the scattered and incident fields, respectively. The scattering amplitude  $S_{mn}$  is, in general, a complex quantity comprised of a magnitude  $s_{mn} = |S_{mn}|$  and a phase angle  $\psi_{mn}$ ,

$$S_{mn} = s_{mn} e^{i\psi_{mn}}; \quad m,n=v \text{ or } h, \quad (9)$$

and it is related to the radar cross section (RCS) of the target,  $\sigma_{mn}$ , by

$$\begin{aligned} \sigma_{mn} &= 4\pi |S_{mn}|^2 \\ &= 4\pi s_{mn}^2; \quad m,n=v \text{ or } h. \end{aligned} \quad (10)$$

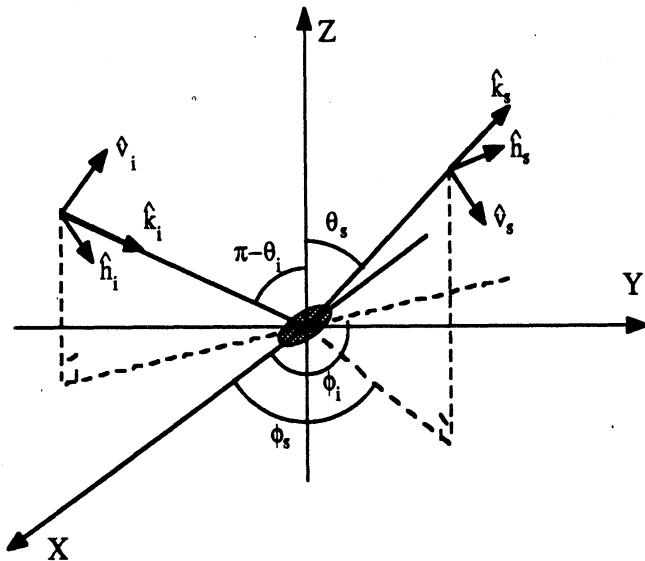


Figure 17: Geometry of scattering of a plane wave from a particle.

Interest in measuring  $\mathbf{S}$  stems from the fact that if the elements of  $\mathbf{S}$  are known, we can compute the RCS of the target that would be observed by a radar with any specified combination of transmit and receive antenna configuration, including elliptical and circular polarizations [6]. In fact, we do not need to know all four magnitudes and four phases of  $\mathbf{S}$  in order to synthesize

the desired RCS; it is sufficient to know the four magnitudes and any three of the phase angles, measured with respect to the fourth as reference. Thus, if we choose  $\psi_{vv}$  as reference, we can write (8) in the form

$$\begin{aligned} \mathbf{S} &= e^{i\psi_{vv}} \begin{bmatrix} s_{vv} & s_{vh}e^{i\psi'_{vh}} \\ s_{hv}e^{i\psi'_{hv}} & s_{hh}e^{i\psi'_{hh}} \end{bmatrix} \\ &= e^{i\psi_{vv}} \mathbf{S}' \end{aligned} \quad (11)$$

where

$$\psi'_{mn} = \psi_{mn} - \psi_{vv}, \quad m,n=v \text{ or } h. \quad (12)$$

For backscattering, the reciprocity theorem mandates that  $S_{hv} = S_{vh}$ , which further reduces the number of unknown quantities from 7 to 5.

The formulation given above is equally applicable to a distributed target. If the effective area illuminated by the radar antenna is  $A$ , the polarimetric scattering behavior of the distributed target is characterized by the differential scattering matrix  $\mathbf{S}^0 = \mathbf{S}/\sqrt{A}$ .

In principle,  $S_{vv}$  and  $S_{hv}$  can be determined by measuring  $E_v^s$  and  $E_h^s$  with the target illuminated by a pure vertically polarized wave  $\mathbf{E}^i = E_v^i \hat{\mathbf{v}}$ ; and, similarly,  $S_{vh}$  and  $S_{hh}$  can be determined by measuring the same quantities when the target is illuminated by  $\mathbf{E}^i = E_h^i \hat{\mathbf{h}}$ . Such a procedure requires that (1) the transmit and receive antennas of the measurement system each have excellent isolation between its v- and h-ports, and (2) the receive-transmit transfer functions of the measurement system be known for all four polarization combinations (vv, vh, hv, and hh). Design techniques are currently available to achieve antenna polarization isolation on the order of 30 dB. For a radar scatterometer system intended to measure the differential scattering matrices of distributed targets such as ground surfaces and vegetation canopies, such a level of isolation is sufficient to insure good measurement accuracy of the magnitudes and phases of all four scattering amplitudes. The error associated with measuring the like-polarized components  $S_{vv}$  and  $S_{hh}$  is negligibly small, and for  $S_{hv}$  (and  $S_{vh}$ ) the error also is less than 0.85 dB if  $|S_{hv}| / |S_{vv}| \geq 0.31$ , which corresponds to  $\sigma_{hv}/\sigma_{vv} \geq 0.1$  (or -10 dB). For natural targets the like- and cross-polarized components,  $S_{hv}$  and  $S_{vh}$  for example, are uncorrelated and

for  $\sigma_{hv}/\sigma_{vv} \geq 0.01$  the associated error would be less than 0.4 dB.

If the radar antennas do not individually have good polarization isolation between their v- and h-ports, it is necessary to characterize each antenna by a polarization distortion matrix that accounts for the coupling between the two ports, and to use at least two, and preferably three, targets of known scattering matrices in order to calibrate the radar completely [7]. Now, we will focus our attention on the problems associated with measuring the receiver-transmitter transfer function for a radar with reasonably good overall cross-polarization isolation using suitable external calibration targets.

## 4.2 System Transfer Function

Although a radar may use a single antenna to provide both transmit and receive functions and may also use a polarization switching network capable of exciting either v- or h-polarized waves in the antenna, we shall use the block diagram shown in Fig. 18 to represent the general case of a two-pole transmitter and a two-pole receiver. Assuming perfect isolation between antenna ports, the voltage received by the v-polarized receive antenna due to illumination of a target at range  $r$  by a v-polarized wave is given by

$$\begin{aligned} E_{vv} &= \left[ \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 r^4} \right]^{1/2} e^{-i2kr} R_v T_v S_{vv} \\ &= \frac{K}{r^2} e^{-i2kr} R_v T_v S_{vv} \end{aligned} \quad (13)$$

where

$$K = \left[ \frac{P_t G_t G_r \lambda^2}{(4\pi)^2} \right]^{1/2}, \quad (14)$$

$S_{vv}$  is the scattering amplitude of the target,  $P_t$  is the transmitted power, and  $G_r$  and  $G_t$  are the nominal gains of the transmit and receive antennas. The quantities  $R_v$  and  $T_v$  are field transfer functions for the receive and transmit antennas, respectively, which account for the deviation in both amplitude and phase from the nominal condition described by  $G_t G_r$ . Similarly, for any receive-transmit polarization configuration, we have

$$E_{mn} = \frac{K}{r^2} e^{-i2kr} R_m T_n S_{mn}, \quad m,n=v \text{ or } h. \quad (15)$$

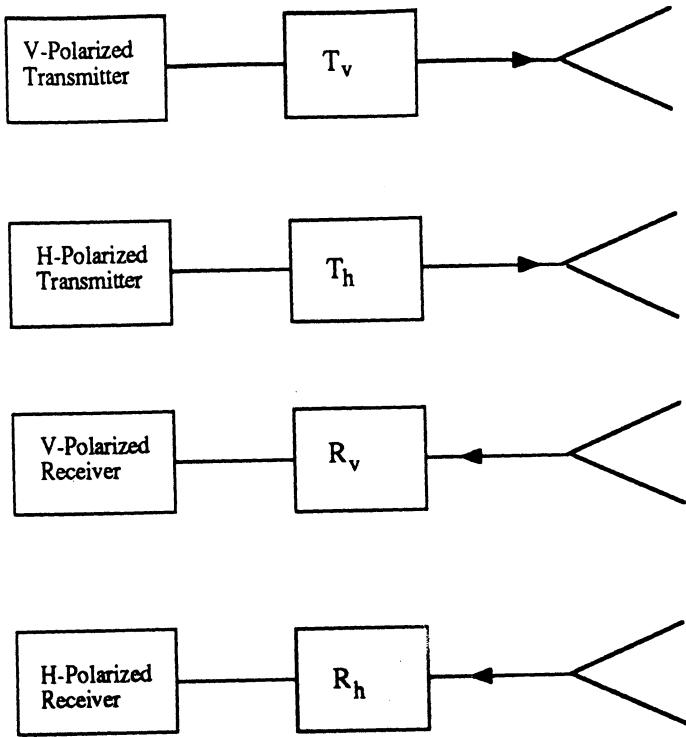


Figure 18: Simplified block diagram of a dual polarized radar system.

### 4.3 Calibration

The standard calibration approach involves the use of one reference target of known scattering matrix. Upon measuring  $E_{mn}$  with  $S_{mn}$  known, the quantity  $(KR_m T_n)$  can be determined in amplitude and phase, with the latter being relative to some reference distance time delay.

In principle, the procedure is simple and straightforward. The problem arises when we need to select a reference target of known scattering matrix. The metal sphere is the easiest target to align and its scattering matrix can be computed exactly [2, p.297]. Unfortunately, it can only be used to calibrate the vv- and hh-channels because its  $S_{hv}^{sp} = S_{vh}^{sp} = 0$ . Targets that exhibit significant cross-polarized scattering include the dihedral corner reflector, tilted cylinders, and others, but scattering from such targets is inherently sensitive to the orientation of the target relative to the  $(\hat{v}_i, \hat{h}_i, \hat{k}_i)$  coordinate system. This, and other factors such as edge scattering, may lead to significant errors between the calculated values of the scattering amplitudes and their actual values. The orientation problem may be reduced down to an acceptable level when

operating in an anechoic chamber under controlled laboratory conditions, but it poses a difficult problem when it is necessary to calibrate a truck-mounted scatterometer, under field conditions.

To solve this problem, we use two reference targets, namely a sphere and any target with strong cross-polarized RCS. As will be shown below, it is not necessary to know the RCS of the second reference target in order to calibrate the radar system.

First, let us use a metal sphere of known size, and place it at a distance  $r_0$  from the radar. The scattering amplitudes of a metal sphere are  $S_{hh} = S_{vv} \equiv S_0$ , and  $S_{hv} = S_{vh} = 0$ . The received fields for vv and hh polarizations are

$$E_{vv}^0 = \frac{K}{r_0^2} e^{-i2kr_0} R_v T_v S_0 \quad (16)$$

$$E_{hh}^0 = \frac{K}{r_0^2} e^{-i2kr_0} R_h T_h S_0 \quad (17)$$

and consequently,

$$K_{vv} = \frac{E_{vv}^0}{S_0} = \frac{K}{r_0^2} e^{-i2kr_0} R_v T_v S_0 \quad (18)$$

$$K_{hh} = \frac{E_{hh}^0}{S_0} = \frac{K}{r_0^2} e^{-i2kr_0} R_h T_h S_0. \quad (19)$$

where the subscript and superscript 0 denote quantities associated with the metal sphere.

Next, let us use any point target that exhibits strong cross-polarized scattering, and let us measure the received field for hv and vh polarizations,

$$E_{hv}^c = \frac{K}{r_c^2} e^{-i2kr_c} R_h T_v S_{hv}^c \quad (20)$$

$$E_{vh}^c = \frac{K}{r_c^2} e^{-i2kr_c} R_v T_h S_{vh}^c \quad (21)$$

where the subscript and the superscript c refers to the cross-polarization calibration target. The reciprocity theorem states that in the backscattering direction, the cross-polarized scattering amplitudes are always equal. Hence,

$$S_{hv}^c = S_{vh}^c \quad (22)$$

and consequently,

$$K_d \equiv \frac{E_{hv}^c}{E_{vh}^c} = \frac{R_h T_v}{R_v T_h}. \quad (23)$$

Now for a test target with unknown scattering matrix  $S^u$ , placed at a distance  $r_u$  from the radar, the received field is

$$E_{vv}^u = \frac{K}{r_u^2} e^{-i2kr_u} R_v T_v S_{vv}^u, \quad (24)$$

$$E_{hh}^u = \frac{K}{r_u^2} e^{-i2kr_u} R_h T_h S_{hh}^u, \quad (25)$$

$$E_{hv}^u = \frac{K}{r_u^2} e^{-i2kr_u} R_h T_v S_{hv}^u, \quad (26)$$

$$E_{vh}^u = \frac{K}{r_u^2} e^{-i2kr_u} R_v T_h S_{vh}^u. \quad (27)$$

Since we are interested in the phase angle  $\psi'$ , we can eliminate the factor  $e^{-i2kr_u}$  in the above equations. Using (18), (19), and (23) in combination with (24)-(27) we obtain the following expressions for the unknown scattering amplitudes

$$S_{vv}^u = \frac{E_{vv}^u}{K_{vv}} \left( \frac{r_u}{r_0} \right)^2 \quad (28)$$

$$S_{hh}^u = \frac{E_{hh}^u}{K_{hh}} \left( \frac{r_u}{r_0} \right)^2 \quad (29)$$

$$S_{hv}^u = \frac{E_{hv}^u}{\sqrt{K_{vv} K_{hh} K_d}} \left( \frac{r_u}{r_0} \right)^2 \quad (30)$$

$$S_{vh}^u = \sqrt{\frac{K_d}{K_{vv} K_{hh}}} E_{vh}^u \left( \frac{r_u}{r_0} \right)^2. \quad (31)$$

Equations (28)-(31) provide expressions for the four scattering amplitudes in terms of (1) the like-polarized received voltages for the metal sphere,  $E_{vv}^0$  and  $E_{hh}^0$ , (2) the ratio of the cross-polarized received voltages for the second calibration target,  $K_d = E_{hv}^c/E_{vh}^c$ , (3) the like-polarized scattering amplitude of the sphere,  $S_0$ , and (4) the ranges to the sphere and the test target,  $r_0$  and  $r_u$ . Note that knowledge of the scattering amplitude of the second calibration target is not required.

#### 4.4 Measurement Accuracy

To verify the validity of the calibration technique summarized by equations (28)-(31), we measured the scattering matrix of a tilted cylinder with the X-band scatterometer. The scatterometer was used in continuous chirped mode operating at a frequency of 9-10 GHz. The scattering measurements were performed in a 13-m long anechoic chamber using the setup diagrammed in Fig. 19.

Although an exact theoretical solution for a finite-length, conducting cylinder does not exist, the solution based on the assumption that the current along the axis of the cylinder is constant provides accurate results in the specular direction, if the length of the cylinder ( $L$ ) is much larger than the wavelength [8]. In order to minimize edge effects caused by scattering by the ends of the cylinder, the diameter of the cylinder ( $D$ ) must also be chosen to be much smaller than the wavelength. Hence, we selected a cylinder with  $L=30.48$  cm and  $D=1.625$  mm.

Correct positioning of the test target with respect to the antenna coordinates is very important. First, the target must be placed at the center of the antenna beam in order to avoid phase variations of the incident field along the axis of the target. This was accomplished using a pair of two laser beams. Another alignment parameter that has to be carefully controlled is the angle  $\phi$  between the incidence direction ( $\hat{k}_i$ ) and the projection of the cylinder axis onto the horizontal plane (Fig. 20). The elements of the scattering matrix are very sensitive to variations in azimuth angle and the rate of change is proportional to the length of the cylinder. This angle was set to  $90^\circ$  with a fine-control stepper motor (steps of a fraction of a tenth of a degree) by maximizing the received power. The 13p ( $\hat{v}_i$ ). Accurate setting of this angle is very difficult. This angle was set to  $50^\circ$  using an inclinometer.

Under the mentioned conditions a signal to noise ratio of 25 dB was achieved for the test cylinder and after background subtraction the signal to noise ratio was improved to 40 dB. To eliminate short-range reflections from the antenna circulators, the returned signal was time-gated, as a result of which the frequency response around the beginning and the end of the frequency band was distorted and discarded.

A 15-cm sphere was used for sphere calibration and a 45° wire-mesh (Fig. 21) was employed as the cross-polarization target. The distances of all the targets from the scatterometer, which were accurately measured using the time-domain feature of the HP 8753, were arranged such that  $r_0 = r_c = r_u$ . The measured amplitudes of the scattering matrix elements of the cylinder are compared with theoretical values in Figures 22- 24. The measured values are within  $\pm 0.3dB$  of the theoretical results. For relative phase, the measured values (Figs. 25 and 26) are within  $\pm 5^\circ$  of theoretical predictions. These deviations are attributed to alignment errors and to the imperfect polarization isolation of the antenna.

The excellent agreement between measurements and theory demonstrates that the calibration technique is an effective approach for calibrating single antenna polarimetric scatterometer systems. This technique is particularly useful for field operations because it does not require accurate alignment of calibration targets or knowledge of the radar cross section of the cross-polarization target.

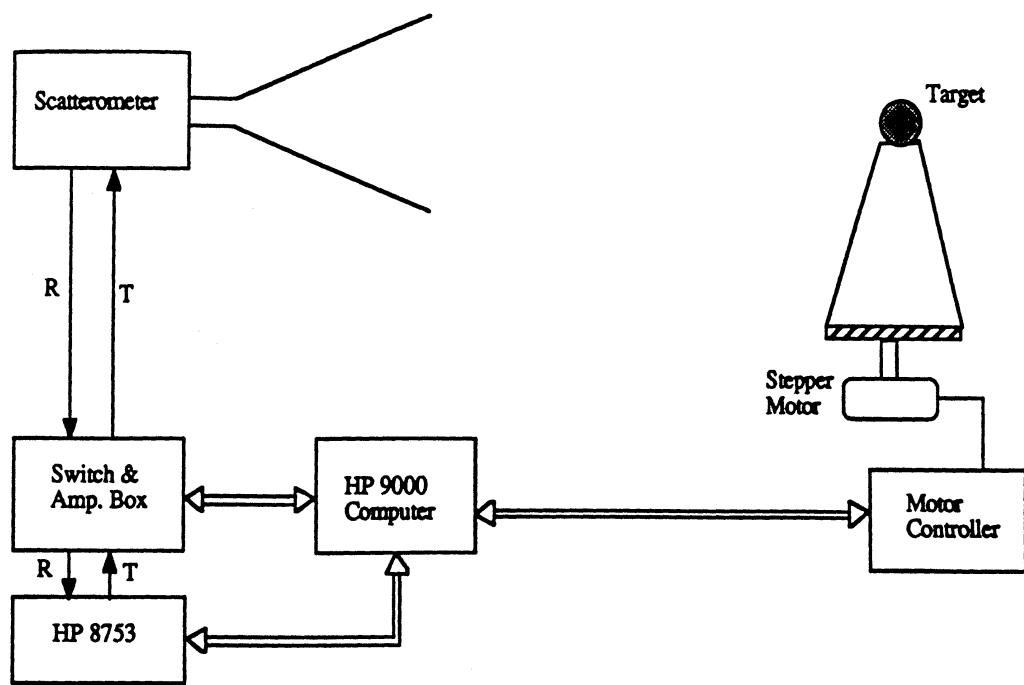


Figure 19: Automatic radar cross section measurement setup.

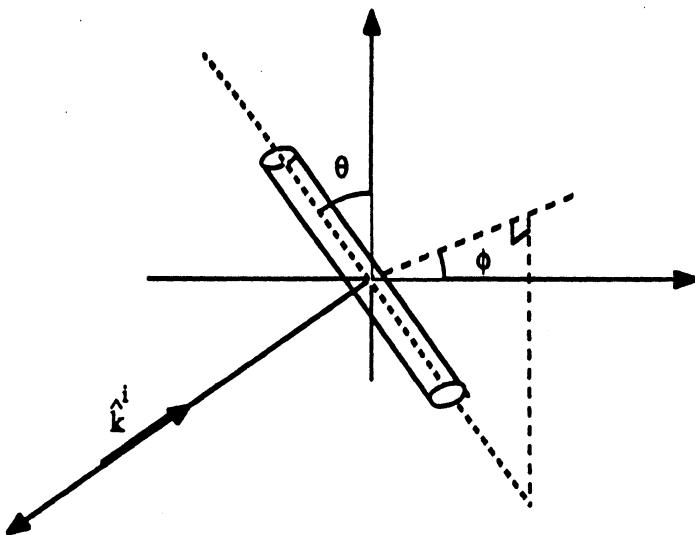


Figure 20: Geometry of scattering of a plane wave from a long, thin cylinder.

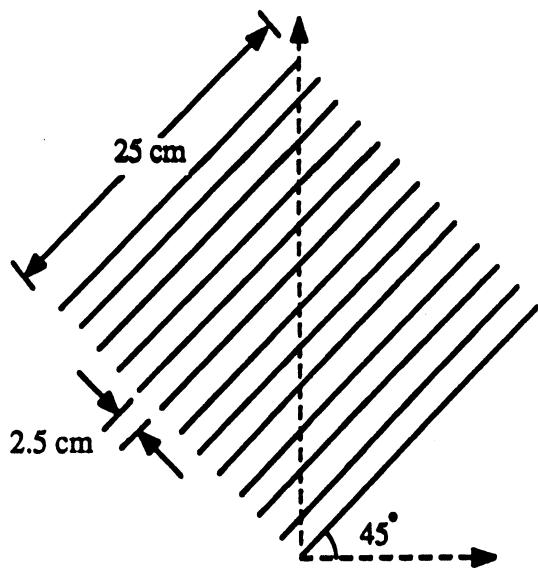


Figure 21: Geometry of a wire-mesh

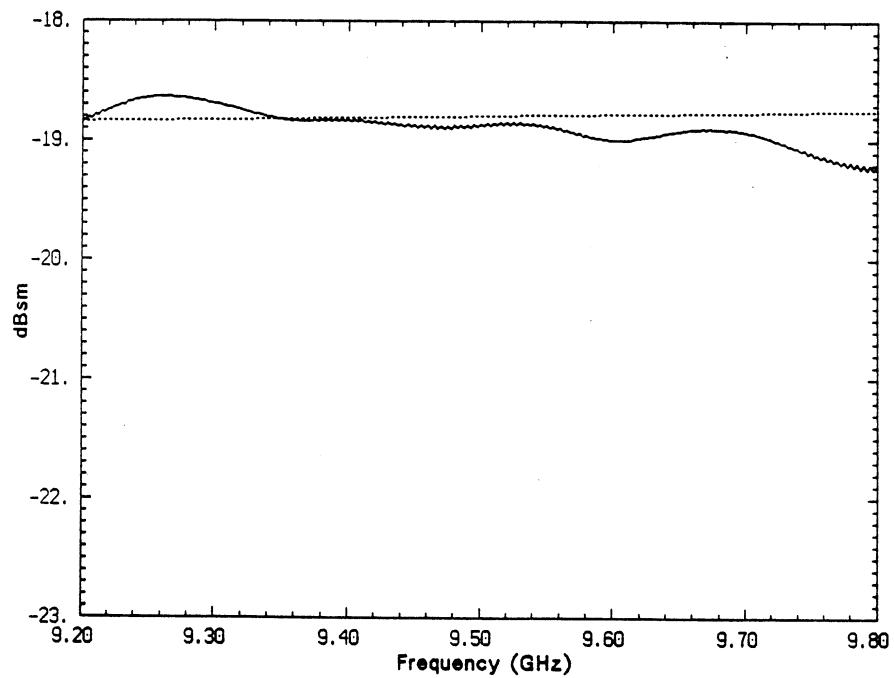


Figure 22: Radar cross section (hh) versus frequency of a cylinder with  $L=30.48$  cm and  $D=1.625$  mm, (—)measured and (- -)theory.

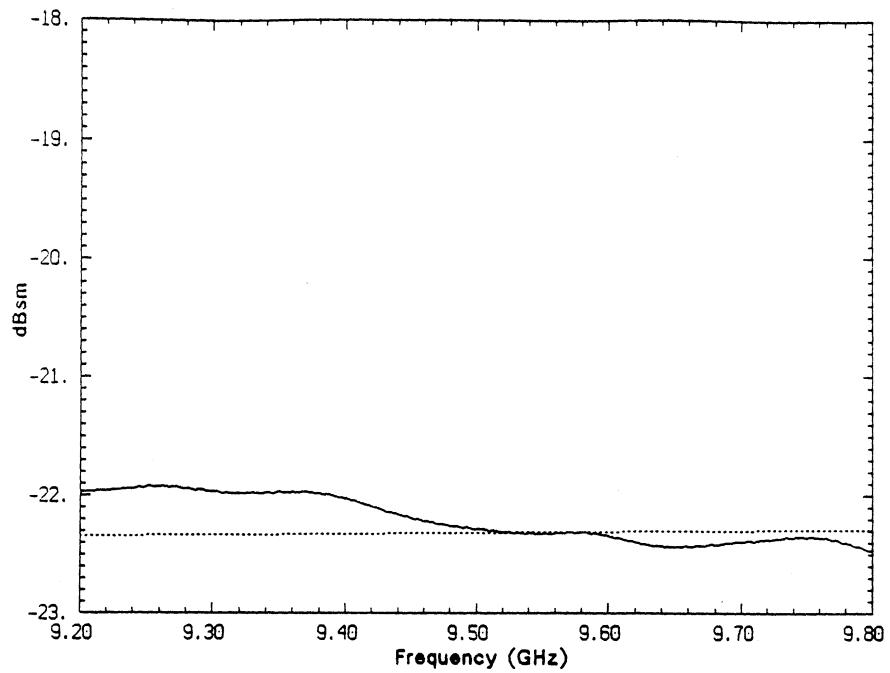


Figure 23: Radar cross section (vv) versus frequency of a cylinder with  $L=30.48$  cm and  $D=1.625$  mm, (—)measured and (- - -)theory.

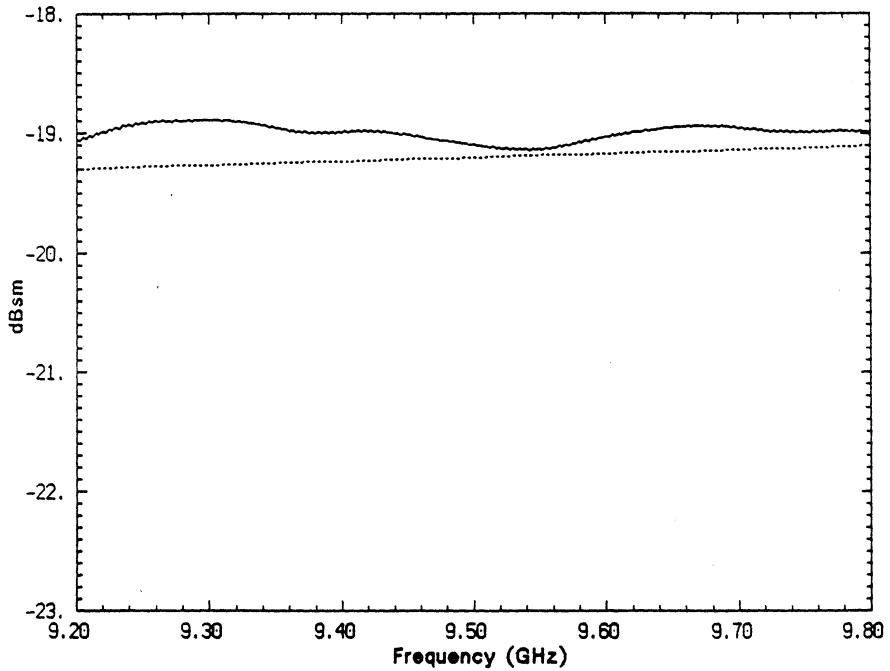


Figure 24: Radar cross section (hv) versus frequency of a cylinder with  $L=30.48$  cm and  $D=1.625$  mm, (—)measured and (- - -)theory.

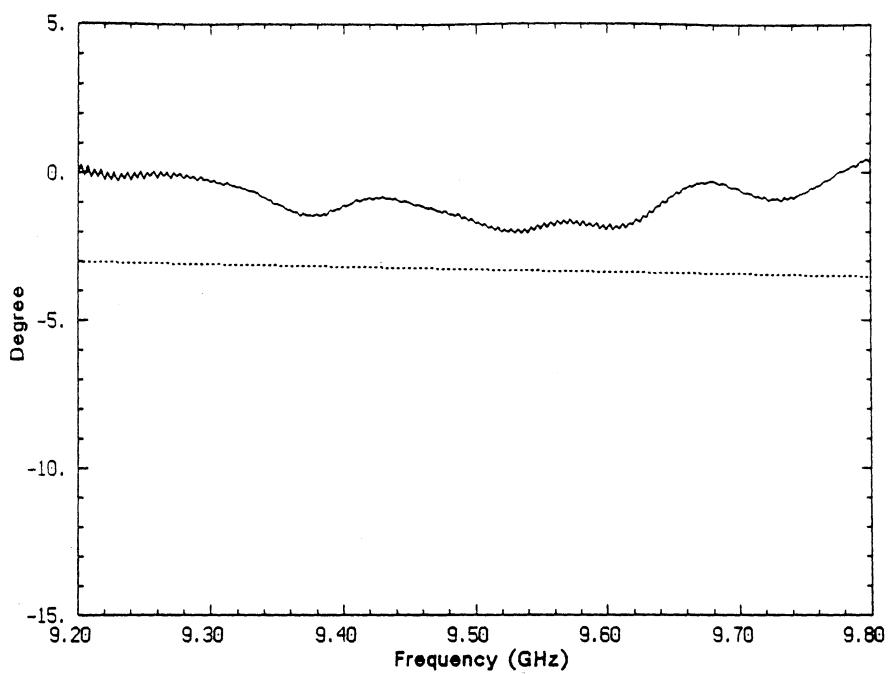


Figure 25: Relative phase  $S_{hh}$  (to  $S_{vv}$ ) versus frequency of a cylinder with  $L=30.48$  cm and  $D=1.625$  mm, (—)measured and (---)theory.

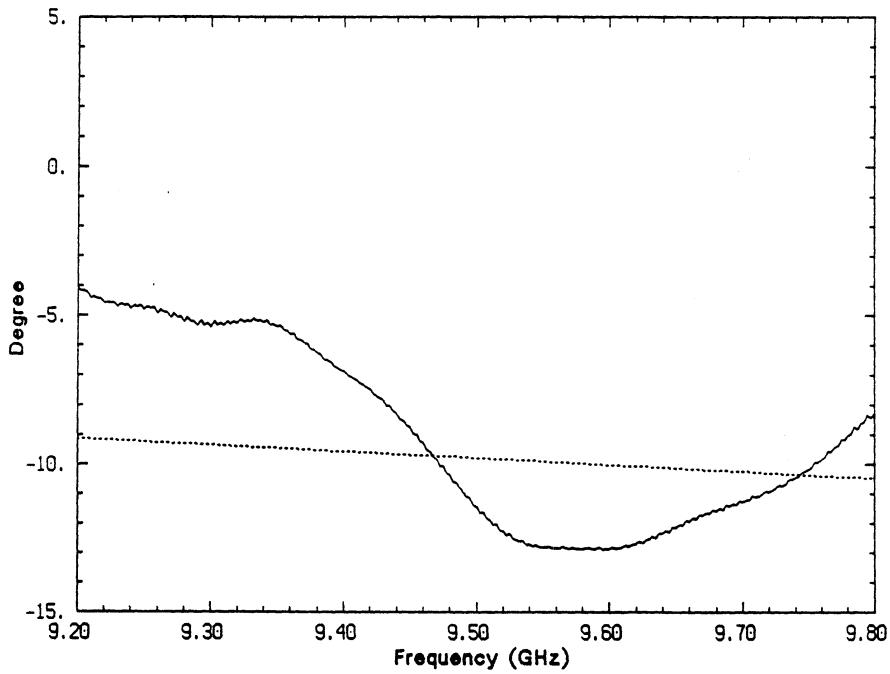


Figure 26: Relative phase  $S_{hv}$  (to  $S_{vv}$ ) versus frequency of a cylinder with  $L=30.48$  cm and  $D=1.625$  mm, (—)measured and (---)theory.

## 5 Measurement Modes

The LCX polarimetric scatterometer can perform measurements in four different modes: (a) point target, (b) surface, (c) volume, and (d) polarimetry, each of which will be discussed in the sections that follows. A software(LCX Version 4.0) was written to facilitate the use of the scatterometer. By selecting from various menus and responding to on-screen prompts, the operator can calibrate and use the scatterometer in the desired mode without having to manually set the parameters on the network analyzer and the relay actuator.

To simplify the equations in the following sections, we redefine the cross polarization calibration factors as follows

$$K_{hv} = \sqrt{K_{vv} K_{hh} K_d} \quad (32)$$

$$K_{vh} = \sqrt{\frac{K_{vv} K_{hh}}{K_d}} \quad (33)$$

such that equations (28)-(31) become

$$S_{mn}^u = \frac{E_{mn}^u}{K_{mn}} \left( \frac{r_u}{r_0} \right)^2. \quad (34)$$

In the next four sections the assumption is that calibration has been performed and the calibration factors( $K_{mn}$ ) and the distance to the calibration target( $r_0$ ) are known at all N discrete frequencies across the bandwidth of operation.

### 5.1 Point Target

The point target routine measures the scattering amplitude of a target with finite size such that the solid angle subtended by the target is smaller than the antenna solid angle. The target should be placed in the far field of the antenna. Point target measurements are done in the frequency domain. Time domain gating is set such that only the target characteristic is being

observed in the frequency response. While setting the gate parameters, the operator is asked to specify the target range( $r_u$ ) on the time domain trace which will be used in calibrating the data.

After setting the gate parameters the program steps through each selected polarization and records N complex values each corresponding to the scattered field( $E_{mn}^u$ ) at that frequency. After removal of the target, a measurement of the background signal( $E_{mn}^b$ ) is obtained with the same gate parameters. Then the program calculates the scattering amplitude of the target at each frequency point using

$$S_{mn}^u = \frac{(E_{mn}^u - E_{mn}^b)}{K_{mn}} \left( \frac{r_u}{r_0} \right)^2. \quad (35)$$

We can calculate the radar cross section of the target using equation (10) which gives

$$\sigma_{mn} = 4\pi \left| \frac{E_{mn}^u - E_{mn}^b}{K_{mn}} \right|^2 \left( \frac{r_u}{r_0} \right)^4. \quad (36)$$

## 5.2 Surface

The surface routine measures the average surface backscattering coefficient( $\sigma^o$ ). The measurements are performed in the frequency domain for N frequency points from each of M independent samples. The gate parameters are set at the beginning of the program and will remain the same for all the M samples.

To derive the equations used in calibration of the distributed target measurements we begin with the radar equation for a point target located at  $r_u$

$$P_r = \frac{P_t G_{t0} G_{r0} \lambda^2 \sigma}{(4\pi)^3 r_u^4}. \quad (37)$$

where  $P_r$  is the received power,  $P_t$  is the transmitted power,  $G_{t0}$  and  $G_{r0}$  are the boresight transmit and receive antenna gains, respectively,  $\lambda$  is the wavelength, and  $\sigma$  is the radar cross section of the target. By comparing equations (36) and (37) and substituting  $P_r/P_t$  for  $|E_{mn}^u - E_{mn}^b|^2$

and  $|K|^2$  for  $|K_{mn}|^2$  we get

$$|K|^2 r_0^4 = \frac{G_{t0} G_{r0} \lambda^2}{(4\pi)^2}. \quad (38)$$

The form of the radar equation used in determining the return power from a surface is given by the expression[3]

$$P_r = \frac{P_t \lambda^2}{(4\pi)^3} \int_{\text{Ill.area}} G_{t0} G_{r0} \frac{\sigma^0}{r^4} dA. \quad (39)$$

Figure 27 illustrates the geometry used. If we assume  $\sigma^0$  to be constant over the illuminated area and writing the gains as  $G_t(\theta, \phi) = G_{t0} g_t(\theta, \phi)$ , the above equation can be rewritten as

$$P_r = \frac{P_t G_{t0} G_{r0} \lambda^2 \sigma^0}{(4\pi)^3} \int_{\text{Ill.area}} g_t(\theta, \phi) g_r(\theta, \phi) \frac{1}{r^4} dA. \quad (40)$$

Using equation (38), we can write

$$\frac{P_r}{P_t} = \frac{|K|^2 r_0^4 \sigma^0}{4\pi} \int_{\text{Ill.area}} g_t(\theta, \phi) g_r(\theta, \phi) \frac{1}{r^4} dA. \quad (41)$$

The integral in this equation is known as the *illumination integral*, I. Values of the illumination integral for a given height and incidence angle have been evaluated using the main-lobe patterns of the antenna.

After rearranging equation (41) to obtain the backscattering coefficient we get

$$\sigma^0 = \frac{4\pi P_r / P_t}{|K|^2 I r_0^4}. \quad (42)$$

The averaging process is performed over N frequency points and M independent samples. If we write  $\sigma_{ij}^0$  for the backscattering coefficient at the  $j^{th}$  point of the  $i^{th}$  trace, then we obtain the overall average of the backscattering coefficient using

$$\sigma^0 = \frac{1}{M} \sum_{i=1}^M \frac{1}{N} \sum_{j=1}^N \sigma_{ij}^0. \quad (43)$$

### 5.3 Volume

The volume measurement mode measures the differential volume backscattering coefficient. The measurements are performed in the time domain mode of the network analyzer and an

integration range over the target return is set by the operator. The program divides this range into  $N$  slices and measures  $M$  independent samples. After averaging the samples, it evaluates the volume backscattering coefficient of the  $k^{th}$  slice attenuated by the volume above that slice.

The development of the formulas is similar to those explained in the previous section. Consider a volume illuminated by the scatterometer as in Figure 28, and divide this volume into  $k$  slices perpendicular to the direction of propagation, each with range extent  $\Delta r_k$ . Similar to the surface case we can write an expression for the power received from each of these slices[3]. The power received from the  $k^{th}$  slice is given by

$$P_{rk} = \frac{P_t G_{t0} G_{r0} \lambda^2}{(4\pi)^3} I_v(h_k, \theta) \frac{\sigma_{vk}}{L(z_k)} \Delta r_k \quad (44)$$

where  $L(z_k)$  is the two way loss through the medium above the  $k^{th}$  slice in the z-direction and  $I_v$  is the volume illumination integral and is given by

$$I_v(h_k, \theta) = \int_{Ill.\text{area}} g_s(\theta, \phi) g_r(\theta, \phi) \frac{1}{r_k^4} dA_{\perp}. \quad (45)$$

where  $A_{\perp} = A \cos \theta$  is the illuminated area perpendicular to the antenna beam. The volume integral equation can be written as

$$I_v(h_k, \theta) = I(h_k, \theta) \cos \theta \quad (46)$$

so that only the surface illumination integral needs to be evaluated.

Since the volume measurement mode is done in the time domain we need to take the average of the calibration constants obtained in the frequency domain and use that as the correction factor. Using equation (38) and (46), we can rearrange equation (44) and get

$$P_{rk}/P_t = \frac{|K|^2 r_0^4}{4\pi} I(h_k, \theta) \frac{\sigma_{vk}}{L(z_k)} \Delta r_k \cos \theta \quad (47)$$

Thus, the volume backscattering coefficient for the  $k^{th}$  slice attenuated by the volume above it can be obtained from

$$\frac{\sigma_{vk}}{L(z_k)} = \frac{4\pi P_{rk}/P_t}{|K|^2 r_0^4 I(h_k, \theta) \Delta r_k \cos \theta}. \quad (48)$$

In many instances it is desirable to have the integrated backscattering coefficient  $\sigma^0$ . It can be written as a summation over N slices as

$$\sigma^0 = \sum_{k=1}^N \frac{\sigma_{vk}}{L(z_k)} \Delta r_k \cos \theta. \quad (49)$$

#### 5.4 Polarimetry

The polarimetric measurement mode measures the calibrated magnitude and phase of the four polarizations of the returned signal from a distributed target. It measures several independent samples and stores them separately. These data can be used to calculate the average phase matrix.

In this mode only the center frequency data is recorded. Time domain gating is set by the operator over the target return. For each sample, all four polarizations are measured sequentially with very short intervals, so that the phase information is conserved.

This routine uses a calibration scheme similar to that of the point target except for scaling the result by the illumination integral instead of the point target range. The equation used is given below

$$S_{mn}^u = \frac{E_{mn}^u}{K_{mn} r_0^2 \sqrt{I}}. \quad (50)$$

where  $I$  is the illumination integral.

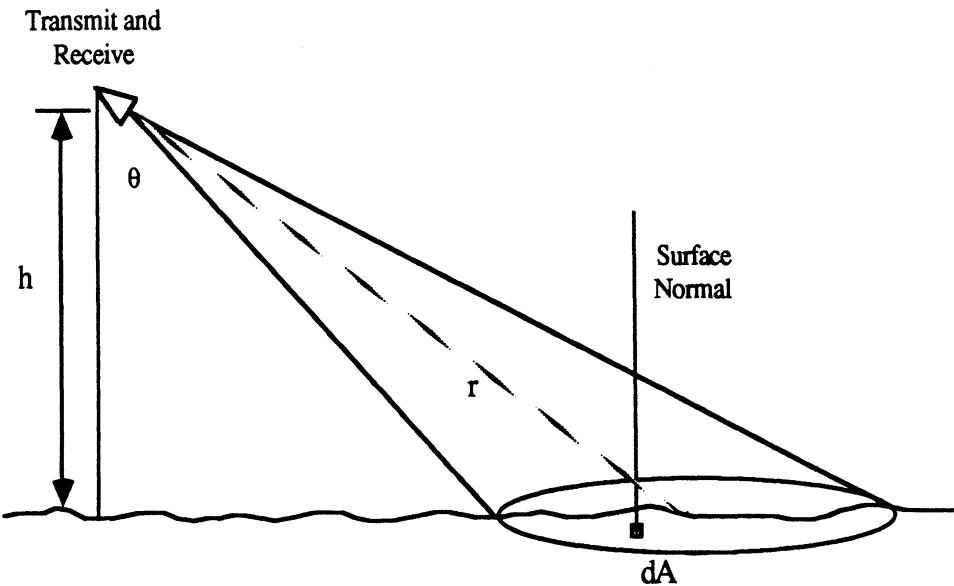


Figure 27: Geometry used in the surface target measurements.

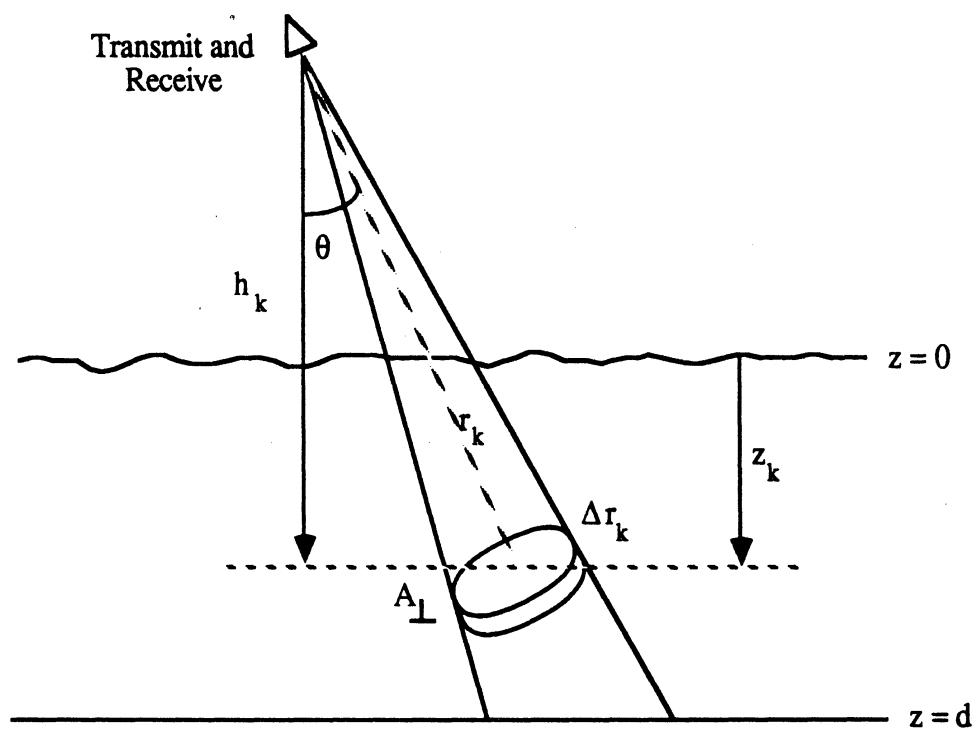


Figure 28: Geometry used in the volume target measurements.

## References

- [1] Braun, E.H., "Some Data for the Design of Electromagnetic Horns", *IEEE Transactions on Antennas and Propagation*, Vol. AP-4, January 1956, pp. 29-31.
- [2] Ulaby, F.T., R.K. Moore, and A.D. Fung, Microwave Remote Sensing: Active and Passive, Vol. I, Reading, MA: Addison Wesley, 1981.
- [3] Ulaby, F.T., R.K. Moore, and A.D. Fung, Microwave Remote Sensing: Active and Passive, Vol. I, Reading, MA: Addison Wesley, 1982.
- [4] Ulaby, F.T., R.K. Moore, and A.D. Fung, Microwave Remote Sensing: Active and Passive, Vol. III, Dedham, MA: Artech House, Inc., 1986.
- [5] Whitt, M.W., F.T. Ulaby, T.F. Haddock, "The Development of a Millimeter-Wave Network Analyzer Based Scatterometer," The University of Michigan, Radiation Lab Technical Report 022872-1-T, January 1987
- [6] Zebker, H.A., J.J. van Zyl, and D.n. Held, "Imaging Radar Polarimetry from Wave Synthesis," *J. Geophys. Res.*, Vol. 92, No. 81, January 1987, pp. 683-701.
- [7] Barnes, R.M., "Polarimetric Calibration Using In-scene Reflectors," MIT Lincoln Lab Technical Report TT-65, September 1986.
- [8] Ruck, G.T., D.E. Barrick, W.D. Stuart, and C.K. Krichbaum, Radar Cross Section Handbook, Vol. 1, New York, NY: Plenum Press, 1970.

## **A Photographs of the system**

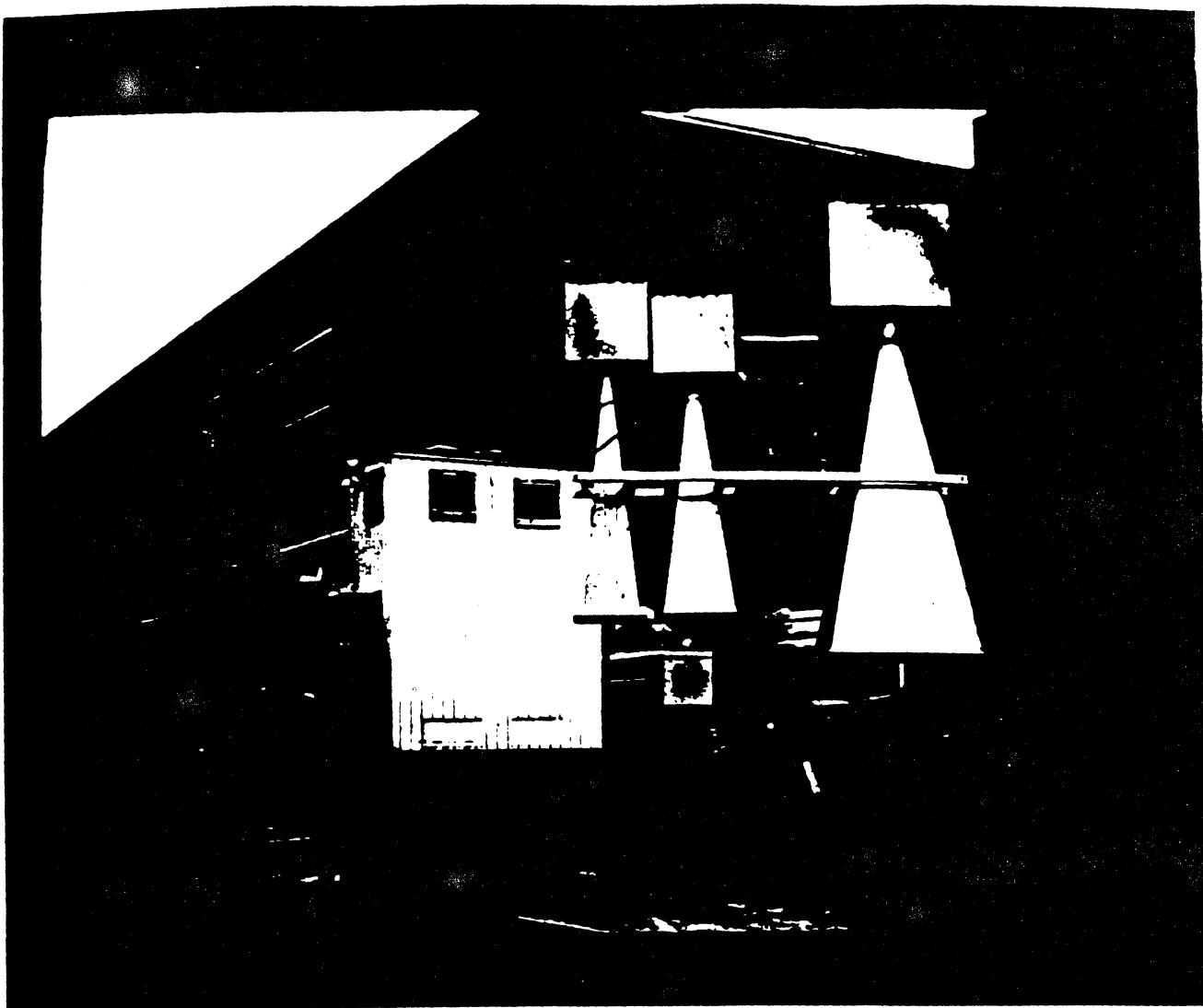


Figure 29: Photograph of the LCX Antennas.

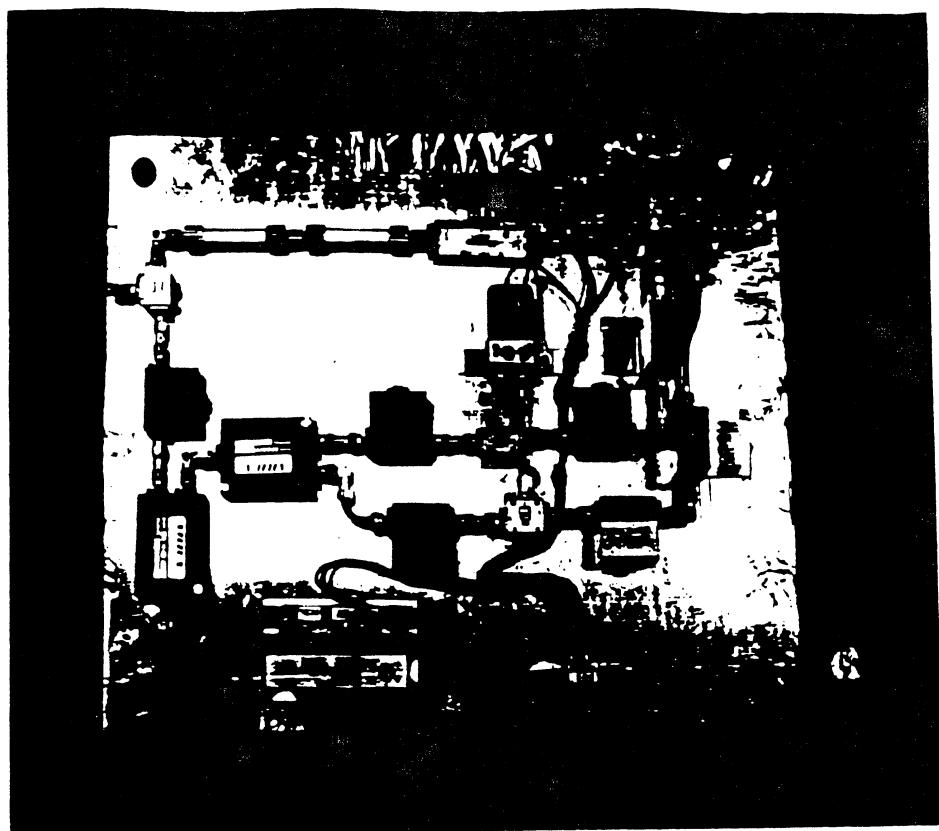


Figure 30: Photograph of the C-band RF Unit.

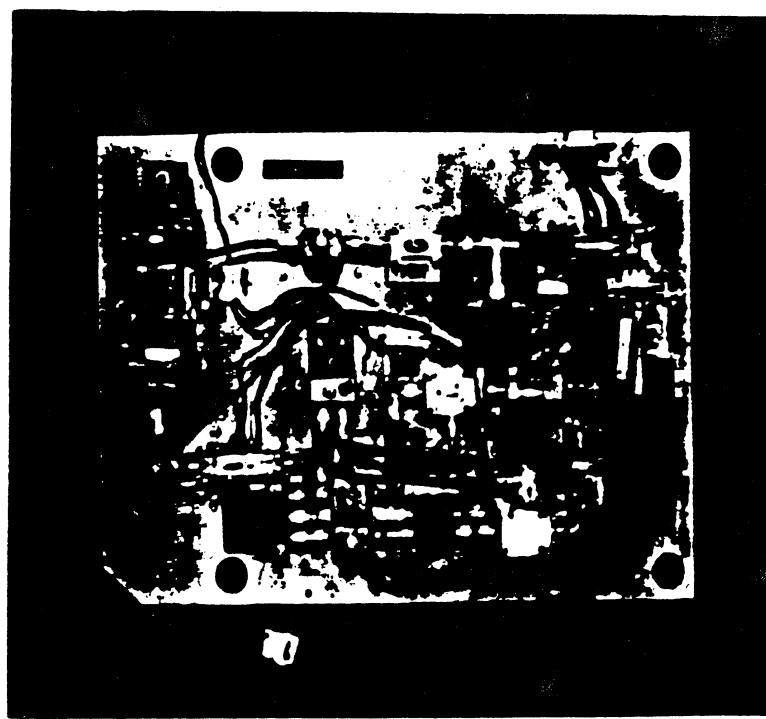


Figure 31: Photograph of the X-band RF Unit.

## **B Measurement Software Listing**

```

100 !*****
110 ! L/C/X POLARIMETER MEASUREMENT PROGRAM
120 ! FILE: LCX      VERSION 4.0
130 !*****
140 ! LAST EDIT: JUNE 1989
150
160
170 ! RE-WRITE OF PROGRAM TO CALIBRATE ANTENNAS USING A SPHERE AND A
180 ! CROSS-POL TARGET. DATE FEB 4, 1989. THE SUBROUTINE RADAR MENU
190 ! HAS BEEN LEFT ESSENTIALLY INTACT, BUT MOST OF THE OTHER ROUTINES
200 ! HAVE BEEN MODIFIED TO REDUCE COMPLEXITY.
210
220 ! NAME      DATE      VER.      CHANGE
230
240
250 !*****
260 OPTION BASE 1
270 COM /Paths/ @Nwa,@Nwa_data,@Hpib,@Relay
280 COM /Constants/ Vel,Zero(3)
290 COM /Sys_1/ Freq$(3)[3],Freq_cent(3),Freq_span(3)
300 COM /Sys_2/ PolS(4)[2],Polsw$(3,4)[8]
310 COM /Sys_3/ INTEGER F disp,P disp
320 COM /Sys_4/ Drive_a$(15),Drive_b$(15),Drive_c$(15),INTEGER Preamble,Bytes
330 COM /Sys_5/ Target$(30),Version$(12),Mode$(7),Out_type$(10),Sounds$(3),Bells$(1),Debug$(3)
340 COM /Sys_6/ Angle,Angle$(10),Bin_rng(3),Beam(3),INTEGER Npts,Ntrace
350 COM /Sys_7/ INTEGER Cal_flag(2,4),Meas_flag(2,4)
360 COM /Cal7/ Cal_keep$(3)[10],Rcal(3),Donecal(3)
370 COM /Illum/ C(3,4,13)
380 DATA "L","C","X"          ! FREQUENCY
390 DATA "VV","HH","HV","VH"   ! POLARIZATION
400 DATA 1.5,1.75,1.5        ! FREQ_CENT
410 DATA 3.,5.,5              ! FREQ_SPAN
420 DATA 12.5,9.0,6.2        ! BEAMWIDTH
430 DATA 1.0,0.6,0.6          ! BIN_RNG
440 DATA "?*B3456","?*A56B34","?*A6B345","?*A5B346"  ! L
450 DATA "?*B3456","?*A34B56","?*A4B356","?*A3B456"  ! C
460 DATA "?*A34B56","?*B3456","?*A3B456","?*A4B356"  ! X
470 DATA ":",700,0,":",700,1,":MEMORY,0,7"           ! DRIVE_A,B,C
480 READ Freq$(*)
490 READ PolS(*)
500 READ Freq_cent(*)
510 READ Freq_span(*)
520 READ Beam(*)
530 READ Bin_rng(*)
540 READ Polsw$(*)
550 READ Drive_a$,Drive_b$,Drive_c$
560
570 ! Set up error handling routine.
580
590 ON ERROR CALL Fix_error
600
610 ! Read illumination integral coefficients from disc.
620 ! Also define look-up table for FORM1 to FORM3 conversion.
630
640 ASSIGN @Disc TO FNFileloc$("ILLUM", "/LCX/ILL"&Drive_a$)
650 ENTER @Disc;C(*)
660 ASSIGN @Disc TO *
670 CALL Table_def
680
690 ! Initialize important parameters.
700
710 DEG
720 MAT Donecal= (0)
730 MAT Cal_flag= (0)
740 MAT Meas_flag= (0)
750 Mode$="MATRIX "
760 F disp=1
770 P_disp=1
780 Out_type$="PRINT/DISC"
790 Span_time=3.00E-7
800
810 ! Zero(*) is the 2-way time domain return from the antenna
820 ! aperture. They have to be changed according to cable lengths.
830
840 ! L-Band aperture is 13.50 ns away from the vv-pol max return.
850 ! C-Band aperture is 12.25 ns away from the vv-pol max return.
860 ! X-Band aperture is 13.75 ns away from the vv-pol max return.
870
880 Zero(1)=(1.8700E-7)+(1.350E-8)
890 Zero(2)=(1.8075E-7)+(1.225E-8)
900 Zero(3)=(1.7800E-7)+(1.375E-8)
910 Vel=2.99792458E+8

```

```

920 Ntrace=3
930 Npts=201
940 AngleS="0"
950 Angle=0
960 Targets=""
970 Sounds="ON"
980 Debugs="OFF"
990 Bells=""
1000 Version$="Version 4.0"
1010 PRINTER IS PRT
1020 PRINT CHR$(27)&"&L1L" ! Set Page Breaks
1030 PRINTER IS CRT
1040 Clear_crt
1050 PRINT
1060 PRINT
1070 PRINT "*****"
1080 PRINT "*"
1090 PRINT "*"
1100 PRINT "* LCX **"
1110 PRINT "* UNIVERSITY OF MICHIGAN RADIATION LAB **"
1120 PRINT "* L/C/X MEASUREMENT PROGRAM **"
1130 PRINT "* (VERSION 4.0) **"
1140 PRINT "* OCTOBER 23, 1988 **"
1150 PRINT "*"
1160 PRINT "* *****"
1170 IF FNAsk("INITIALIZE RAM DISK?") THEN
1180   INITIALIZE Drive_c$,0
1190   INITIALIZE Drive_c$,1000
1200 END IF
1210 HP8753_init(@Nwa,@Nwa_data,@Hplib,@Relay,Debug$)
1220 Series_init
1230 OUTPUT-KBD;"SCRATCH KEY"&CHR$(255)&CHR$(88);
1240 Start_loop: !
1250 RadaF menu(Ends)
1260 IF Ends=="Y" THEN GOTO Program_end
1270 IF Mode$=="POINT" THEN CALL Point_target
1280 IF Mode$=="SURFACE" THEN CALL Surface
1290 IF Mode$=="VOLUME" THEN CALL Volume
1300 IF Mode$=="MATRIX" THEN CALL Polarimetry
1310 GOTO Start_loop
1320 RETURN
1330 Program_end: !
1340 DISP "PROGRAM EXIT"
1350 LOAD KEY "EDITKEY:MEMORY,0,1"
1360 STOP
1370 END
1380 !*****
1390 !SUB Freq_set(INTEGER Ifreq)
1400 !
1410 ! This subroutine sets the transmit frequency for the HP8753.
1420 !
1430 COM /Paths/ @Nwa,@Nwa_data,@Hplib,@Relay
1440 COM /Sys 1/ Freq$(*),Freq_cent(*),Freq_span(*)
1450 OUTPUT @Nwa;"TIMDTRANOFF"
1460 OUTPUT @Nwa;"CENT "&VALS(Freq_cent(Ifreq))&" GHZ"
1470 OUTPUT @Nwa;"SPAN "&VALS(Freq_span(Ifreq))&" GHZ"
1480 SUBEND ! Freq_set
1490 !*****
1500 SUB Pol_sw(INTEGER Ifreq,Ipol)
1510 !
1520 ! This subroutine sets the transmit and receive polarization by
1530 ! sending the proper command over the HPIB to the polarization
1540 ! relays. It also displays the freq and pol on HP8753.
1550 !
1560 COM /Paths/ @Nwa,@Nwa_data,@Hplib,@Relay
1570 COM /Sys 1/ Freq$(*),Freq_cent(*),Freq_span(*)
1580 COM /Sys 2/ Pol$(*),Polsw$(*)
1590 OUTPUT @Relay;Polsw$(Ifreq,Ipol)
1600 OUTPUT @Nwa;"TITL "" "&Freq$(Ifreq)&" BAND - "&Pol$(Ipol)&""
1610 WAIT .1
1620 SUBEND ! Pol_sw
1630 !*****
1640 SUB Angle_set(Angle)
1650 !
1660 ! This subroutine tells the user to set the antenna positioner angle.
1670 !
1680 COM /Sys 5/ Targets$,Version$,Mode$,Out type$,Sounds,Bell$,Debug$
1690 Angle$=&VALS(Angle)&CHR$(179)&" " !degree sign
1700 DISP "SET ANTENNA ANGLE TO ";Angle$;" <CONTINUE>";Bell$
1710 PAUSE
1720 DISP "WORKING"
1730 SUBEND ! Angle_set
1740 !*****
1750 SUB Series_init
1760 !
1770 ! This subroutine prints a header for the printout and sets the system
1780 ! date and time.
1790 !

```

```

1800 DIM Input$[80]
1810 PRINTER IS PRT
1820 PRINT CHR$(12)
1830 Set_clock
1840 LINPUT "ENTER MEASUREMENT SERIES TITLE",Input$
1850 Preface$=**&RPTS(" ",9)
1860 PRINT RPTS(**,70)
1870 PRINT Preface$\&Input$
1880 LINPUT "ENTER OPERATOR NAME",Input$
1890 PRINT Preface$\&Input$
1900 PRINTER IS CRT
1910 PRINT
1920 PRINT
1930 PRINTER IS PRT
1940 PRINT Preface$\&"MEASUREMENT SERIES STARTED AT "&TIME$(TIMEDATE)
1950 PRINTER IS CRT
1960 WAIT 1.0
1970 Comments
1980 SUBEND ! Series_init
1990 !*****SUBROUTINE TO PRINT COMMENTS*****
2000 SUB Comments
2010 !
2020 ! This subroutine prints comments on the printout.
2030 !
2040 DIM Input$[80]
2050 Clear_crt
2060 PRINT "ENTER COMMENTS. WHEN FINISHED, BEGIN THE NEXT LINE"
2070 PRINT "WITH A / THEN <ENTER>."
2080 PRINT RPTS(**,70)
2090 PRINTER IS PRT
2100 PRINT RPTS(**,70)
2110 Loop_start: !
2120 LINPUT "",Input$
2130 IF Input$[1,1]="/" THEN Loop_end
2140 PRINTER IS CRT
2150 PRINT Input$
2160 PRINTER IS PRT
2170 PRINT Input$
2180 GOTO Loop_start
2190 Loop_end: !
2200 PRINTER IS PRT
2210 PRINT RPTS(**,70)
2220 PRINTER IS CRT
2230 Clear_crt
2240 SUBEND ! Comments
2250 !*****SUBROUTINE TO CONFIGURE HP-IB BUS*****
2260 SUB Hp8753_init (@Nwa,@Nwa_data,@Hpib,@Relay,Debug$)
2270 !
2280 ! This subroutine configures the HP-IB bus and presets the HP8753.
2290 ! It also sets the frequency and pol switches to the default values &
2300 ! HP8753 in the time domain mode.
2310 !
2320 COM /Sys_1/ Freq$(*),Freq_cent(*),Freq_span(*)
2330 COM /Sys_2/ Pol$(*),Polsw$(*)
2340 COM /Sys_3/ INTEGER F_disp,P_disp
2350 ASSIGN @Hpib TO 7
2360 ASSIGN @Nwa TO 716
2370 ASSIGN @Nwa_data TO 716;FORMAT OFF
2380 ASSIGN @Relay TO 710
2390 REMOTE @Hpib
2400 ABORT @Hpib
2410 CLEAR @Nwa
2420 IF Debug$="ON" THEN OUTPUT @Nwa;"DEBUON"
2430 OUTPUT @Nwa;"USEPASC"
2440 IF Debug$="OFF" THEN
2450   OUTPUT @Nwa;"DEBUOFF"
2460   OUTPUT @Nwa;"TITLE ""&Freq$(1)&" BAND - "&Pol$(1)&"""
2470 END IF
2480 CALL Freq_set(F_disp)
2490 CALL Freq_sw(F_disp)
2500 CALL Pol_sw(F_disp,P_disp)
2510 OUTPUT @Nwa;"TIMDTRANON;STAR 100 NS;STOP 400 NS"
2520 SUBEND ! Hp8753_init
2530 !*****SUBROUTINE TO CALL RADAR MENU*****
2540 SUB Radar_menu(End$)
2550 !
2560 ! This is the main subroutine. In this subroutine, measurement parameters
2570 ! are set, and calibration routine is called. This subroutine is exited
2580 ! when beginning a measurement sequence or ending the program.
2590 !
2600 COM /Paths/ @Nwa,@Nwa_data,@Hpib,@Relay
2610 COM /Constants/ Vel,Zero(*)
2620 COM /Sys_1/ Freq$(*),Freq_cent(*),Freq_span(*)
2630 COM /Sys_2/ Pol$(*),Polsw$(*)
2640 COM /Sys_3/ INTEGER F_disp,P_disp
2650 COM /Sys_4/ Drive_a$,Drive_b$,Drive_c$,INTEGER Preamble,Bytes
2660 COM /Sys_5/ Target$,Version$,Mode$,Out_type$,Sound$,Bell$,Debug$
2670 COM /Sys_6/ Angle,Angle$,Bin_rng(*),Beam(*),INTEGER Npts,Ntrace

```

```

2680 COM /Sys 7/ INTEGER Cal_flag(*),Meas_flag(*)
2690 COM /Cal7 Cal_keep$(*),Rcal(*),Donecal(*)
2700 INTEGER F_disp_old,P_disp_old
2710 INTEGER F,P
2720 !
2730 ! This is the main menu subroutine.
2740 !
2750 ALLOCATE INTEGER Cal_flag_old(2,4)
2760 ALLOCATE INTEGER Meas_flag_old(2,4)
2770 End$="N"
2780 Menu clear: !
2790 OFF KEY
2800 Clear_crt
2810 Menu: !
2820 OUTPUT KBD;CHR$(255)&CHR$(84); !home display
2830 PRINT TABXY(1,1)
2840 PRINT
2850 PRINT
2860 PRINT " L/C/X"
2870 PRINT " PARAMETER SELECTION MENU"
2880 PRINT
2890 PRINT
2900 PRINT
2910 PRINT " PARAMETER CURRENT VALUE"
2920 PRINT
2930 PRINT " FREQUENCY ";
2940 FOR F=1 TO 3
2950 IF Meas_flag(1,F) THEN PRINT Freq$(F)&" ";
2960 NEXT F
2970 PRINT "
2980 PRINT " POLARIZATION ";
2990 FOR P=1 TO 4
3000 IF Meas_flag(2,P) THEN PRINT Pol$(P)&" ";
3010 NEXT P
3020 PRINT "
3030 PRINT " ANTENNA ANGLE "&Angle$
3040 PRINT " TARGET TYPE "&Target$
3050 PRINT " MEASUREMENT MODE "&Mode$ 
3060 PRINT " CURRENT DISPLAY "&Freq$(F_disp)&" "&Pol$(P_disp);
3070 Menu 1: !
3080 OFF KEY
3090 ON KEY 0 LABEL " FREQ/POLARIZ " GOTO Fred_pol
3100 ON KEY 1 LABEL " ANGLE " GOTO Angle
3110 ON KEY 2 LABEL " TARGET " GOTO Target
3120 ON KEY 3 LABEL " MODE " GOTO Mode
3130 ON KEY 4 LABEL " BEGIN " GOTO Begin
3140 ON KEY 5 LABEL " MORE " GOTO Menu_2
3150 ON KEY 6 LABEL " COMMENTS " GOTO Comment_print
3160 ON KEY 7 LABEL " DISPLAY " GOTO Display
3170 ON KEY 8 LABEL " CALIBRATE " GOTO Calibrate
3180 ON KEY 9 LABEL " QUIT " GOTO Quit
3190 Spin:GOTO Spin ! WAIT FOR SOFTKEY INTERRUPT
3200 Menu 2: !
3210 OFF KEY
3220 ON KEY 1 LABEL " PARAMETERS " GOTO Parameters
3230 ON KEY 3 LABEL " SPHERE RESP " CALL Sphere_response
3240 ON KEY 9 LABEL " MAIN MENU " GOTO Menu_clear
3250 GOTO Spin
3260 !-----+
3270 Fred_pol: !
3280 OFF KEY
3290 MAT Meas_flag_old= Meas_flag
3300 MAT Meas_flag= (0)
3310 ON KEY 0 LABEL " L BAND " GOTO Set_l
3320 ON KEY 1 LABEL " C BAND " GOTO Set_c
3330 ON KEY 2 LABEL " X BAND " GOTO Set_x
3340 ON KEY 4 LABEL " STORE " GOTO Store
3350 ON KEY 5 LABEL " VV " GOTO Vv_set
3360 ON KEY 6 LABEL " HH " GOTO Hh_set
3370 ON KEY 7 LABEL " HV " GOTO Hv_set
3380 ON KEY 8 LABEL " VH " GOTO Vh_set
3390 ON KEY 9 LABEL " CANCEL " GOTO Can
3400 GOTO Spin
3410 Set_l: !
3420 OFF KEY 0
3430 Meas_flag(1,1)=1
3440 GOTO Spin
3450 Set_c: !
3460 OFF KEY 1
3470 Meas_flag(1,2)=1
3480 GOTO Spin
3490 Set_x: !
3500 OFF KEY 2
3510 Meas_flag(1,3)=1
3520 GOTO Spin
3530 Vv_set: !
3540 OFF KEY 5
3550 Meas_flag(2,1)=1

```

```

3560 GOTO Spin
3570 Hh_set: !
3580 OFF KEY 6
3590 Meas_flag(2,2)=1
3600 GOTO Spin
3610 Hv_set: !
3620 OFF KEY 7
3630 Meas_flag(2,3)=1
3640 GOTO Spin
3650 Vh_set: !
3660 OFF KEY 8
3670 Meas_flag(2,4)=1
3680 GOTO Spin
3690 Can: !
3700 OFF KEY
3710 MAT Meas_flag= Meas_flag_old
3720 GOTO Menu
3730 Store:!
3740 OFF KEY
3750 GOTO Menu
3760 !-----3770 Angle: !
3780 OFF KEY
3790 INPUT "ENTER MEASUREMENT ANGLE",Angle
3800 Angle$=VALS$(Angle)&CHR$(179)&" " !degree sign
3810 GOTO Menu
3820 !-----3830 Target: !
3840 OFF KEY
3850 LINPUT "ENTER TARGET TYPE",Target$
3860 Target$=TRIMS$(Target$)
3870 Target$=Target$&RPTS$(" ",30-LEN(Target$))
3880 GOTO Menu
3890 !-----3900 Comment print:
3910 OFF KEY
3920 Comments
3930 GOTO Menu
3940 !-----3950 Quit: !
3960 OFF KEY
3970 Clear crt
3980 Ends="Y"
3990 GOTO Sub_end
4000 !-----4010 Parameters:!
4020 OFF KEY
4030 Clear_crt
4040 PRINT
4050 PRINT
4060 PRINT " PARAMETER CURRENT VALUE"
4070 PRINT
4080 PRINT "# OF TRACES/SET ";Ntrace
4090 PRINT "# OF POINTS ";Npts
4100 PRINT " VOLUME BIN SIZE (meters) ";Bin_rng(1);Bin_rng(2);Bin_rng(3)
4110 PRINT " OUTPUT TYPE ";Out_types
4120 PRINT " SOUND ";Sounds
4130 PRINT " DEBUG ";Debug$"
4140 ON KEY 0 LABEL "# OF TRACES" GOTO Trace
4150 ON KEY 1 LABEL "# OF POINTS" GOTO Points
4160 ON KEY 2 LABEL " BIN SIZE" GOTO Bin_rng_set
4170 ON KEY 3 LABEL " OUTPUT TYPE" GOTO Output_set
4180 ON KEY 4 LABEL " SOUND" GOTO Sound_set
4190 ON KEY 5 LABEL " DEBUG MODE" GOTO Debug_set
4200 ON KEY 9 LABEL " MAIN MENU" GOTO Menu_Clear
4210 GOTO Spin
4220 Trace: !
4230 OFF KEY
4240 INPUT "INPUT THE # OF TRACES DESIRED (integer)",Ntrace
4250 GOTO Parameters
4260 Points:!
4270 OFF KEY
4280 INPUT "INPUT THE # OF SAMPLE POINTS DESIRED (integer 51,101,201,401)",Npts
4290 Bytes=Npts*16
4300 GOTO Parameters
4310 Bin_rng_set: !
4320 OFF KEY
4330 ON KEY 0 LABEL " L BAND" GOTO Bin_rng_l
4340 ON KEY 1 LABEL " C BAND" GOTO Bin_rng_c
4350 ON KEY 2 LABEL " X BAND" GOTO Bin_rng_x
4360 GOTO Spin
4370 Bin_rng l: !
4380 OFF KEY
4390 INPUT "ENTER THE DESIRED BIN SIZE (ONE-WAY IN METERS)",Bin_rng(1)
4400 GOTO Parameters
4410 Bin_rng c:!
4420 OFF KEY
4430 INPUT "ENTER THE DESIRED BIN SIZE (ONE-WAY IN METERS)",Bin_rng(2)

```

```

4440 GOTO Parameters
4450 Bin_rng x:!
4460 OFF KEY
4470 INPUT "ENTER THE DESIRED BIN SIZE (ONE-WAY IN METERS)",Bin_rng(3)
4480 GOTO Parameters
4490 Output_set: !
4500 OFF KEY
4510 ON KEY 0 LABEL " PRINT      " GOTO Out_pr
4520 ON KEY 1 LABEL " DISC      " GOTO Out_d
4530 ON KEY 2 LABEL " PRINT/DISC " GOTO Out_pr_d
4540 GOTO Spin
4550 Out_pr: !
4560 OFF KEY
4570 Out type$="PRINT      "
4580 GOTO Parameters
4590 Out d: !
4600 OFF KEY
4610 Out type$="DISC      "
4620 GOTO Parameters
4630 Out pr d: !
4640 OFF KEY
4650 Out type$="PRINT/DISC"
4660 GOTO Parameters
4670 Sound set:!
4680 OFF KEY
4690 IF Sound$="ON " THEN
4700   Sounds="OFF"
4710   Bell$=" "
4720 ELSE
4730   Sounds="ON "
4740   Bell$=" "
4750 END IF
4760 GOTO Parameters
4770 Debug set:!
4780 OFF KEY
4790 IF Debug$="ON " THEN
4800   Debug$="OFF"
4810   OUTPUT @Nwa;"DEBUOFF"
4820 ELSE
4830   Debug$="ON "
4840   OUTPUT @Nwa;"DEBUON"
4850 END IF
4860 GOTO Parameters
4870 !-----.
4880 Calibrate: !
4890 OFF KEY
4900 MAT Cal_flag old= Cal_flag
4910 MAT Cal_flag= (0)
4920 ON KEY 0 LABEL " L BAND      " GOTO Cal_1
4930 ON KEY 1 LABEL " C BAND      " GOTO Cal_c
4940 ON KEY 2 LABEL " X BAND      " GOTO Cal_x
4950 ON KEY 4 LABEL "BEGIN FULL CAL" GOTO Begin_cal
4960 ON KEY 5 LABEL " VV          " GOTO Cal_vv
4970 ON KEY 6 LABEL " HH          " GOTO Cal_hh
4980 ON KEY 7 LABEL " HV-VH       " GOTO Cal_cross
4990 ON KEY 9 LABEL " CANCEL      " GOTO Cancel
5000 GOTO Spin
5010 Cal 1: !
5020 OFF KEY 0
5030 Cal_flag(1,1)=1
5040 GOTO Spin
5050 Cal c: !
5060 OFF KEY 1
5070 Cal_flag(1,2)=1
5080 GOTO Spin
5090 Cal x: !
5100 OFF KEY 2
5110 Cal_flag(1,3)=1
5120 GOTO Spin
5130 Cal vv: !
5140 OFF KEY 5
5150 Cal_flag(2,1)=1
5160 GOTO Spin
5170 Cal hh: !
5180 OFF KEY 6
5190 Cal_flag(2,2)=1
5200 GOTO Spin
5210 Cal_cross:!
5220 OFF KEY 7
5230 Cal_flag(2,3)=1
5240 Cal_flag(2,4)=1
5250 GOTO Spin
5260 Begin cal: !
5270 OFF KEY
5280 Clear_crt
5290 Cal(Cal_flag(*),Npts)
5300 GOTO Menu_clear
5310 Cancel: !

```

```

5320 OFF KEY
5330 MAT Cal_flag= Cal_flag_old
5340 GOTO Menu
5350 !-----.
5360 Mode:!
5370 OFF KEY
5380 ON KEY 0 LABEL " POINT      " GOTO Point
5390 ON KEY 1 LABEL " SURFACE    " GOTO Surface
5400 ON KEY 2 LABEL " VOLUME     " GOTO Volume
5410 ON KEY 3 LABEL " MATRIX     " GOTO Matrix
5420 GOTO Spin
5430 Point:!
5440 OFF KEY
5450 Modes="POINT"
5460 GOTO Menu
5470 Surface:!
5480 OFF KEY
5490 Modes="SURFACE"
5500 GOTO Menu
5510 Volume:!
5520 OFF KEY
5530 Modes="VOLUME"
5540 GOTO Menu
5550 Matrix:!
5560 OFF KEY
5570 Modes="MATRIX"
5580 GOTO Menu
5590 !-----.
5600 Display:!
5610 OFF KEY
5620 Clear_crt
5630 F_disp_old=F_disp
5640 P_disp_old=P_disp
5650 F_disp=0
5660 P_disp=0
5670 ON KEY 0 LABEL " L      " GOTO L_disp
5680 ON KEY 1 LABEL " C      " GOTO C_disp
5690 ON KEY 2 LABEL " X      " GOTO X_disp
5700 ON KEY 4 LABEL " DISPLAY " GOTO Disp
5710 ON KEY 5 LABEL " VV     " GOTO Vv_disp
5720 ON KEY 6 LABEL " HH     " GOTO Hh_disp
5730 ON KEY 7 LABEL " HV     " GOTO Hv_disp
5740 ON KEY 8 LABEL " VH     " GOTO Vh_disp
5750 ON KEY 9 LABEL " CANCEL " GOTO Can_disp
5760 GOTO Spin
5770 L_disp:!
5780 IF F_disp=0 THEN F_disp=1
5790 IF F_disp<>1 THEN GOTO Mess_disp
5800 OFF KEY 0
5810 GOTO Spin
5820 C_disp:!
5830 IF F_disp=0 THEN F_disp=2
5840 IF F_disp<>2 THEN GOTO Mess_disp
5850 OFF KEY 1
5860 GOTO Spin
5870 X_disp:!
5880 IF F_disp=0 THEN F_disp=3
5890 IF F_disp<>3 THEN GOTO Mess_disp
5900 OFF KEY 2
5910 GOTO Spin
5920 Vv_disp:!
5930 IF P_disp=0 THEN P_disp=1
5940 IF P_disp<>1 THEN GOTO Mess_disp
5950 OFF KEY 5
5960 GOTO Spin
5970 Hh_disp:!
5980 IF P_disp=0 THEN P_disp=2
5990 IF P_disp<>2 THEN GOTO Mess_disp
6000 OFF KEY 6
6010 GOTO Spin
6020 Hv_disp:!
6030 IF P_disp=0 THEN P_disp=3
6040 IF P_disp<>3 THEN GOTO Mess_disp
6050 OFF KEY 7
6060 GOTO Spin
6070 Vh_disp:!
6080 IF P_disp=0 THEN P_disp=4
6090 IF P_disp<>4 THEN GOTO Mess_disp
6100 OFF KEY 8
6110 GOTO Spin
6120 Disp:!
6130 IF F_disp=0 OR P_disp=0 THEN
6140   DISP "YOU MUST SET BOTH A FREQUENCY AND A POLARIZATION TO DISPLAY"
6150   GOTO Spin
6160 END IF
6170 OFF KEY
6180 DISP "WORKING"
6190 OUTPUT @Nwa;"USEPASC"

```

```

6200 CALL Freq_set(F_disp)
6210 OUTPUT @Nwa;"TIMDTRANON; LOGM; GATEOFF; CONT"
6220 IF Debug$="OFF" THEN
6230   OUTPUT @Nwa;"DEBUOFF"
6240   OUTPUT @Nwa;"TITL """&Freq$(F_disp)&" BAND - "&Pol$(P_disp)&"""
6250 ELSE
6260   OUTPUT @Nwa;"DEBUON"
6270 END IF
6280 CALL Freq_sw(F_disp)
6290 CALL Pol_sw(F_disp,P_disp)
6300 DISP ""
6310 GOTO Menu
6320 Can_disp:!
6330 OFF KEY
6340 F_disp=F_disp_old
6350 P_disp=P_disp_old
6360 DISP ""
6370 GOTO Menu
6380 Mess_disp:!
6390 DISP "ONLY ONE FREQ./POL. CAN BE DISPLAYED AT ONCE."
6400 GOTO Spin
6410 !-----
6420 Begin: !
6430 OFF KEY
6440 Clear_crt
6450 Sub_end: !
6460 Message$=**
6470 SUBEND ! Radar menu
6480 !*****SUBROUTINE Radar menu*****
6490 SUB Clear_crt
6500   OUTPUT KBD;CHRS(255)&CHRS(75);
6510 SUBEND
6520 !*****SUBROUTINE Clear_crt*****
6530 SUB Date_string(Brief_date$,Realtime)
6540 !
6550 ! This subroutine converts the timedata as produced by the computer
6560 ! to a 10-digit number YYMMDDHHMM used for various purposes.
6570 !
6580 INTEGER Month
6590 Realdate$=DATE$(Realtime)
6600 Brief_date$=Realdate$[10,11]
6610 SELECT Realdate$[4,6]
6620 CASE "Jan"
6630   Month=1
6640 CASE "Feb"
6650   Month=2
6660 CASE "Mar"
6670   Month=3
6680 CASE "Apr"
6690   Month=4
6700 CASE "May"
6710   Month=5
6720 CASE "Jun"
6730   Month=6
6740 CASE "Jul"
6750   Month=7
6760 CASE "Aug"
6770   Month=8
6780 CASE "Sep"
6790   Month=9
6800 CASE "Oct"
6810   Month=10
6820 CASE "Nov"
6830   Month=11
6840 CASE "Dec"
6850   Month=12
6860 END SELECT
6870 Month$=VALS(Month)
6880 IF LEN(Month$)=1 THEN Month$="0"&Month$
6890 Day$=VALS(VAL(Realdate$[1,2]))
6900 IF LEN(Day$)=1 THEN Day$="0"&Day$
6910 Brief_date$=Brief_date$&Month$&Day$
6920 Realtime$=TIME$(Realtime)
6930 Brief_date$=Brief_date$&Realtime$[1,2]&Realtime$[4,5]
6940 SUBEND ! Date_string
6950 !*****SUBROUTINE Date_string*****
6960 SUB Gate_set(Gate_cent,Gate_span,Angle,Range,Height,Beam,INTEGER F)
6970 !
6980 ! This subroutine sets the time gate and returns the range distance.
6990 !
7000 COM /Paths/ @Nwa,@Nwa_data,@Hpib,@Relay
7010 COM /Constants/ Vel,Zero(*)
7020 COM /Sys 1/ Freq$(*),Freq_cent(*),Freq_span(*)
7030 COM /Sys 5/ Target$,Version$,Mode$,Out_types$,Sounds$,Bell$,Debug$
7040 Gate_cent=1.50E-7
7050 Gate_span=2.00E-8
7060 OUTPUT @Nwa;"TIMDTRANON; LOGM;"
7070 OUTPUT @Nwa;"GATSNORM;GATECENT "&VAL$(Gate_cent)&" S;GATESPAN "&VAL$(Gate_span)&" S;GATECENT"

```

```

7080     OUTPUT @Nwa;"OPC?;WAIT"
7090     ENTER @Nwa;Reply
7100     LOCAL 716
7110     DISP "SET CENTER OF TIME GATE WITH KNOB OR KEYPAD <CONTINUE>",Bell$
7120     PAUSE
7130     REMOTE 716
7140     DISP "WORKING"
7150     OUTPUT @Nwa;"OUTPACTI;"
7160     ENTER @Nwa;Gate_cent
7170 !
7180     OUTPUT @Nwa;"MARK1";Gate_cent
7190     LOCAL 716
7200     DISP "FIND RANGE WITH MARKER"
7210     PAUSE
7220     REMOTE 716
7230     OUTPUT @Nwa;"OUTPACTI;"
7240     ENTER @Nwa;Range
7250     Range=(Range-Zero(F))*Vel/2.0
7260     Height=Range*COS(Angle)
7270     Gate_span=2*(Height/Vel)*(1/COS(Angle+Beam/2)-1/COS(Angle-Beam/2))
7280     Test=1.5*(1.92/Freq_span(F))/1.E+9
7290     IF Gate span<Test THEN Gate span=Test
7300     OUTPUT @Nwa;"GATESPAN"&VAL$ (Gate_span) &" S"
7310     OUTPUT @Nwa;"GATESPAN"
7320     OUTPUT @Nwa;"OPC?;WAIT"
7330     ENTER @Nwa;Reply
7340     LOCAL 716
7350     DISP "MODIFY GATE SPAN IF DESIRED <CONTINUE>",Bell$
7360     PAUSE
7370     REMOTE 716
7380     DISP "WORKING"
7390     OUTPUT @Nwa;"OUTPACTI;"
7400     ENTER @Nwa;Gate span
7410     IF Gate span<0 THEN Gate span=-Gate span
7420     OUTPUT @Nwa;"GATESPAN "&VAL$ (Gate_span) &" S"
7430     SUBEND ! Gate_set
7440 !*****SUBROUTINE CALIBRATION*****
7450 SUB Cal(INTEGER Cal_flag(*),Npts)
7460 !
7470 ! This subroutine calibrates all frequencies and polarizations which
7480 ! were selected from main menu after pressing the CALIBRATE softkey
7490 ! (as recorded in the Cal_flag matrix).
7500 OPTION BASE 1
7510 COM /Paths/ @Nwa,@Nwa_data,@Hplib,@Relay
7520 COM /Constants/ Vel,Zero(*)
7530 COM /Sys 1/ Freq$(*).Freq_cen(*).Freq_span(*)
7540 COM /Sys 2/ PolS(*).Polsw$(*)
7550 COM /Sys 3/ INTEGER F disp,P disp
7560 COM /Sys 4/ Drive a$,Drive b$,Drive c$,INTEGER Preamble,Bytes
7570 COM /Sys 5/ Target$,Version$,Mode$,Out type$,Sound$,Bell$,Debug$
7580 COM /Cal7 Cal_keeps(*),Rcal(*),Donecal7*)
7590 INTEGER F,P,K
7600 ALLOCATE COMPLEX Room(3,4,Npts),Refer(3,4,Npts),Room_ref(3,4,Npts)
7610 ALLOCATE Amp(Npts),Phas(Npts),Freq(Npts)
7620 ALLOCATE COMPLEX Sph(Npts),Kvv(Npts),Khh(Npts),Kvh(Npts),Kvh(Npts)
7630 ALLOCATE COMPLEX Kc(Npts),Kt(Npts),Km(Npts),Kd(Npts)
7640 TO-TIMEDATE
7650 Preamble=9025
7660 Bytes=Npts*16
7670 MAT Refer= (CMPLX(0,0))
7680 MAT Room_ref= (CMPLX(0,0))
7690 MAT Room= (CMPLX(0,0))
7700 !
7710 ! Begin stepping through frequency and polarization for
7720 ! calibration.
7730 !
7740 FOR F=1 TO 3
7750     IF Cal_flag(1,F)=0 THEN 7910
7760     CALL Fred_set(F)
7770     CALL Fred_sw(F)
7780     CALL Pol_sw(F,1)
7790     OUTPUT @Nwa;"TIMDTRANON;STAR 100 NS;STOP 400 NS"
7800     DISP "POSITION REFERENCE TARGET FOR "&Freq$(F)&" BAND <CONTINUE>",Bell$
7810     PAUSE
7820     DISP "WORKING"
7830     CALL Measure_cal(Room_ref(*),Cal_flag(*),Npts,F,"ROOM_REF")
7840 !
7850 ! Measure background
7860 !
7870     DISP "REMOVE REFERENCE TARGET (DO NOT REMOVE MOUNT!) <CONTINUE>",Bell$
7880     PAUSE
7890     DISP "WORKING"
7900     CALL Measure_cal(Room(*),Cal_flag(*),Npts,F,"ROOM")
7910     NEXT F
7920     MAT Refer= Room_ref-Room
7930     MAT Room_ref= Room
7940 !
7950 ! JUGGLE ARRAYS AND STORE CALIBRATION.

```

```

7960 !
7970 FOR F=1 TO 3
7980 ! IF Cal_flag(1,F)=0 THEN 8460
7990 DISP "ENTER CALIBRATION TARGET TYPE (SPH8,SPH12) FOR ";Freq$(F);"-BAND";
8000 INPUT File_ref$&Drive_c$&Preamble,Npts
8010 ! Move file to RAM cache.
8020 COPY FNFileloc$(File_ref$,"/LCX/SphTheor"&Drive_a$) TO File_ref$&Drive_c$&Preamble,Npts
8030 ASSIGN @Datafile TO File_ref$&Drive_c$;FORMAT OFF
8040 ENTER @Datafile,1;File_ref$,TS,Bytes,Preamble,Npts
8050 ENTER @Datafile;Freq(*),Amp(*),Phas(*)
8060 ASSIGN @Datafile TO *
8070 !
8080 ! Convert Theoretical Sphere Data to Real & Imaginary
8090 !
8100 FOR K=1 TO Npts
8110 ! Sph_re=Amp(K)*COS(Phas(K))
8120 ! Sph_im=Amp(K)*SIN(Phas(K))
8130 ! Sph(K)=CMPLX(Sph_re,Sph_im)
8140 NEXT K
8150 !
8160 ! Calculating Calibration Constants
8170 !
8180 FOR P=1 TO 3
8190 ! IF Cal_flag(2,P)=0 THEN 8320
8200 ! MAT Kc= Refer(F,P,*)
8210 ! IF P=1 THEN MAT Kvv= Sph/Kc
8220 ! IF P=2 THEN MAT Khh= Sph/Kc
8230 ! IF P=3 THEN
8240 ! MAT Kt= Refer(F,4,*)
8250 ! FOR K=1 TO Npts
8260 ! ! Km(K)=SQRT(Kvv(K)*Khh(K))
8270 ! ! Kd(K)=SQRT(Kc(K)/Kt(K))
8280 ! NEXT K
8290 ! MAT Khv= Km/Kd
8300 ! MAT Kvh= Km . Kd
8310 ! END IF
8320 ! NEXT P
8330 ! Realtime=TIMEDATE
8340 !
8350 ! Storing Calibration Constants
8360 !
8370 CALL Date_string(Cal_keep$(F),Realtime)
8380 ! Cal_keep$(F)=Freq$(F)&Cal_keep$(F)[2,10]
8390 Bytes_per_rec=256+(4*16*Npts)
8400 CREATE BDAT FNFileloc$(Cal_keep$(F),"//LCX/CAL"&Drive_a$),1,Bytes_per_rec
8410 ASSIGN @Keep TO FNFileloc$(Cal_keep$(F),"//LCX/CAL"&Drive_a$)
8420 OUTPUT @Keep;Cal_keep$(F),Npts,Rcal(F)
8430 OUTPUT @Keep;Kvv(*),Khh(*),Khv(*),Kvh(*)
8440 ASSIGN @Keep TO *
8450 Donecal(F)=1
8460 NEXT F
8470 CALL Freq_sw(F_disp)
8480 CALL Pol_sw(F_disp,P_disp)
8490 T1=TIMEDATE
8500 T=(T1-T0)/60
8510 DISP "CALIBRATION TOOK ";DROUND(T,5);" MINUTES"
8520 WAIT 5
8530 SUBEND ! Cal
8540 !*****SUBROUTINE MEASURE_CAL*****
8550 SUB Measure_cal(COMPLEX Array(*),INTEGER Flag(*),Npts,F,Title$)
8560 !
8570 ! This subroutine measures the cal target and background.
8580 ! It is called from Cal subroutine.
8590 !
8600 OPTION BASE 1
8610 COM /Paths/ @Nwa,@Nwa_data,@HPib,@Relay
8620 COM /Sys 1/ Freq$(*),Freq_cent(*),Freq_span(*)
8630 COM /Sys 2/ PolS(*),PolwS(*)
8640 COM /Cal/7 Cal_keep$(*),Rcal(*),Donecal(*)
8650 INTEGER P
8660 ALLOCATE COMPLEX Dummy(Npts)
8670 FOR P=1 TO 4
8680 ! IF Flag(2,P)=0 THEN 8870
8690 CALL Pol_sw(F,P)
8700 IF Title$=="ROOM_REF" THEN !Do the co-pol's first
8710 ! IF P=3 THEN
8720 ! DISP "POSITION CROSS-POL TARGET FOR "&Freq$(F)&" BAND <CONTINUE>";Bell$&PAUSE
8730 ! END IF
8740 CALL Gate_set(Gate_cent(F,P),Gate_span(F,P),Angle,Range(F),Height,Beam(F),F)
8750 Rcal(F)=Range(F)
8770 PRINT "RANGE=",Rcal(F)
8780 OUTPUT @Nwa;" TIMDTRANOFF;"
8790 OUTPUT @Nwa;"SAVE"&VALS(P)&; WAIT;"
8800 END IF
8810 OUTPUT @Nwa;"RECA"&VALS(P)&; WAIT;"
8820 OUTPUT @Nwa;"POLA; AUTO; AVERFACT 17;"
8830 OUTPUT @Nwa;"AVEROON; NUMG 17;"
```

```

8840     CALL Get_form3(Dummy(*),Npts,F)
8850     MAT Array(F,P,*)= Dummy
8860     OUTPUT @Nwa;"AVEROFF; LOGM; TIMDTRANON; WAIT; CONT"
8870     NEXT P
8880 SUBEND ! Measure_cal
8890 !*****SUBROUTINE Point_target*****
10650 SUB Point_target
10660 !
10670 ! This subroutine measures the rcs of a point target.
10680 ! The range of the target and cal target should be the same,
10690 ! if different, correct it using the (Rcal/Range)^2.
10700 !
10710 OPTION BASE 1
10720 COM /Paths/ @Nwa,@Nwa_data,@Hplib,@Relay
10730 COM /Constants/ Vel,Zero(*)
10740 COM /Sys 1/ Freq$(*),Freq cent(*),Freq span(*)
10750 COM /Sys 2/ PolS(*),Polsw$(*)
10760 COM /Sys 3/ INTEGER F disp,P disp
10770 COM /Sys 4/ Drive_a$,Drive_b$,Drive_c$,INTEGER Preamble,Bytes
10780 COM /Sys 5/ Target$,Version$,Mode$,Out_type$,Sound$,Bell$,Debug$
10790 COM /Sys 6/ Angle,Angles,Bin rng(*),Beam(*),INTEGER Npts,Ntrace
10800 COM /Sys 7/ INTEGER Cal flag(*),Meas flag(*)
10810 COM /Cal7 Cal keep$(*),Rcal(*),Donecal(*)
10820 DIM Outfil$[10]
10830 INTEGER F,P,K,Flg
10840 !
10850 ! Allocate memory space.
10860 !
10870 ALLOCATE COMPLEX Room(3,4,Npts),Refer(3,4,Npts),Room ref(3,4,Npts)
10880 ALLOCATE COMPLEX Kcal(3,4,Npts),Subkcal(Npts),Dummy(Npts)
10890 ALLOCATE COMPLEX Data1(3,4,Npts)
10900 ALLOCATE Magdata(3,4,Npts),Phasedata(3,4,Npts)
10910 ALLOCATE Range(3)
10920 ALLOCATE Gate cent(3),Gate span(3)
10930 ALLOCATE Tim(3,4)
10940 MAT Refer= (CMPLX(0,0))
10950 MAT Room ref= (CMPLX(0,0))
10960 MAT Room= (CMPLX(0,0))
10970 CALL Load cal(Kcal(*),Flg)
10980 IF Flg=1 THEN SUBEXIT
10990 !
11000 ! Begin target measurement (step thru frequency & polarization).
11010 !
11020 FOR F=1 TO 3
11030   IF Meas_flag(1,F)=0 THEN 11450
11040   CALL Freq_set(F)
11050   CALL Freq_sw(F)
11060   Flg=0
11070   FOR P=1 TO 4
11080     IF Meas_flag(2,P)=0 THEN 11250
11090     CALL Pol_sw(F,P)
11100     IF Flg=0 THEN
11110       DISP "POSITION TEST TARGET FOR "&Freq$(F)&" BAND <CONTINUE>";Bells
11120       PAUSE
11130     END IF
11140     CALL Gate_set(Gate cent(F),Gate span(F),Angle,Range(F),Height,Beam(F),F)
11150     Rcal(F)=Range(F)
11160     PRINT "RANGE=",Rcal(F)
11170     OUTPUT @Nwa;" TIMDTRANOFF;"
11180     OUTPUT @Nwa;"SAVE"&VALS(P)&; WAIT;
11190     OUTPUT @Nwa;"POLA; AUTO; AVERFACT 17;"
11200     OUTPUT @Nwa;"AVEROON; NUMG 17;"
11210     CALL Get_form3(Dummy(*),Npts,F)
11220     MAT Room_ref(F,P,*)= Dummy
11230     Flg=1
11240     OUTPUT @Nwa;"AVEROFF; LOGM; TIMDTRANON; WAIT; CONT"
11250 NEXT P
11260 !
11270 ! Background measurement
11280 !
11290 Flg=0
11300 FOR P=1 TO 4
11310   IF Meas_flag(2,P)=0 THEN 11440
11320   CALL Pol_sw(F,P)
11330   IF Flg=0 THEN
11340     DISP "REMOVE TEST TARGET (DO NOT REMOVE MOUNT!) <CONTINUE>";Bells
11350     PAUSE
11360   END IF
11370   OUTPUT @Nwa;"RECA"&VALS(P)&; WAIT;
11380   OUTPUT @Nwa;"POLA; AUTO; AVERFACT 17;"
11390   OUTPUT @Nwa;"AVEROON; NUMG 17;"
11400   CALL Get_form3(Dummy(*),Npts,F)
11410   MAT Room(F,P,*)= Dummy
11420   Flg=1
11430   OUTPUT @Nwa;"AVEROFF; LOGM; TIMDTRANON; WAIT; CONT"
11440 NEXT P
11450 NEXT F
11460 !

```

```

11470 !     Data correction      Magdata is in voltage
11480 !
11490 !     MAT Refer= Room_ref-Room    ! Subtract Background
11500 !     MAT Room_ref= RRoom
11510 !     MAT DataI= KcalRefer      ! Scale data according to Cal constants
11520 !     MAT Magdata= ABS(DataI)   ! Store the magnitude in Magdata
11530 !     MAT Phasedata= ARG(DataI) ! Store the phase in Phasedata
11540 !
11550 !     Realtime=TIMEDATE
11560 !     CALL Date_string(Outfil$,Realtime)
11570 !     CALL Catalog(Outfil$,Mode$,Targets$,Angle)
11580 !
11590 !     Write output to printer if this output type is set.
11600 !
11610 IF TRIMS(Out_type$)="PRINT" OR TRIMS(Out_type$)="PRINT/DISC" THEN
11620     PRINTER IS~PRT
11630     PRINT
11640     PRINT "VERSION = ";Version$; "# OF POINTS = ";Npts
11650     PRINT
11660     PRINT
11670     PRINT TAB(4); "FREQ"; TAB(12); "GATE CENT"; TAB(25); "GATE SPAN";
11680     PRINT TAB(38); "R_CAL"; TAB(47); "RANGE"; TAB(56); "CAL_FILE"
11690     PRINT
11700 FOR F=1 TO 3
11710     IF Meas_flag(1,F)=0 THEN 11750
11720     PRINT TAB(6); Freq$(F); TAB(12); DROUND(Gate_cent(F),5); TAB(25); DROUND(Gate_span(F),5);
11730     PRINT TAB(37); DROUND(Rcal(F),4); TAB(46); DROUND(Range(F),4);
11740     PRINT TAB(55); Cal_Keep$(F)
11750 NEXT F
11760     PRINT
11770     PRINT
11780     PRINT TAB(4); "FREQ"; TAB(11); "POL"; TAB(20); "MAGNITUDE"; TAB(34); " PHASE "
11790     PRINT
11800 FOR F=1 TO 3
11810     IF Meas_flag(1,F)=0 THEN 11960
11820     FOR P=1 TO 4
11830         IF Meas_flag(2,P)=0 THEN 11940
11840         FOR K=1 TO 3
11850             PRINT TAB(4); Freq$(F) &" beg"; TAB(11); Pol$(P); TAB(19);
11860             PRINT DROUND(Magdata(F,P,1),5); TAB(30); DROUND(Phasedata(F,P,1),5)
11870             !
11880             PRINT TAB(4); Freq$(F) &" mid"; TAB(11); Pol$(P); TAB(19);
11890             PRINT DROUND(Magdata(F,P,(Npts-1)/2),5); TAB(30); DROUND(Phasedata(F,P,(Npts-1)/2),5)
11900             !
11910             PRINT TAB(4); Freq$(F) &" end"; TAB(11); Pol$(P); TAB(19);
11920             PRINT DROUND(Magdata(F,P,Npts),5); TAB(30); DROUND(Phasedata(F,P,Npts),5)
11930     NEXT K
11940     NEXT P
11950     PRINT
11960     NEXT F
11970     PRINTER IS CRT
11980 END IF
11990 !
12000 ! Create an output filename based on the current date and time.
12010 ! Then output important processing parameters and sigma zero data.
12020 !
12030 IF TRIMS(Out_type$)="DISC" OR TRIMS(Out_type$)="PRINT/DISC" THEN
12040     Bytes_per_Fec=64+(10+12+7+30+3*10+4*8+8*2+7*5)+(8*8)+(Npts*2*8)
12050     CREATE BDAT FNFileloc$(Outfil$,"/LCK/DATA"&Drive_a$),1,Bytes_per_rec
12060     ASSIGN @Disc TO FNFileloc$(Outfil$,"/LCK/DATA"&Drive_a$)
12070     OUTPUT @Disc;Outfil$,Version$,Mode$,Targets$,Cal_Keep$(*),Angle,Range(*),Meas_flag(*)
12080     OUTPUT @Disc;Gate_cent(*),Gate_span(*),Npts,Ntrace
12090     OUTPUT @Disc;Magdata(*),Phasedata(*)
12100     ASSIGN @Disc TO *
12110 END IF
12120 !
12130 ! Place radar in default mode.
12140 !
12150 CALL Freq_sw(F_disp)
12160 CALL Pol_sw(F_disp,P_disp)
12170 SUBEND ! Point Target
12180 !*****SUB Surface*****
12190 SUB Surface
12200 !
12210 ! This subroutine treats all targets as surfaces.
12220 ! It measures the sigma-zero and std dev of it.
12230 ! It averages each frequency trace and then averages
12240 ! over Ntrace traces.
12250 !
12260 OPTION BASE 1
12270 COM /Paths/ @Nwa,@Nwa_data,@Hplib,@Relay
12280 COM /Constants/ Vel,Zero(*)
12290 COM /Sys 1/ Freq$(*),Freq_cen(*),Freq_span(*)
12300 COM /Sys_2/ Pol$(*),Polsw$(*)
12310 COM /Sys_3/ INTEGER F_disp,P_disp
12320 COM /Sys_4/ Drive_a$,Drive_b$,Drive_c$,INTEGER Preamble,Bytes
12330 COM /Sys_5/ Targets$,Version$,Mode$,Out_type$,Sound$,Bell$,Debug$
12340 COM /Sys_6/ Angle,Angle_rng(*),Beam(*),INTEGER Npts,Ntrace

```

```

12350 COM /Sys 7/ INTEGER Cal_flag(*),Meas_flag(*)
12360 COM /Cal7 Cal_keeps(*),Rcal(*),Donecal(*)
12370 DIM Outfiles[10]
12380 INTEGER F,P,K,Flg,Skyflg
12390 ! Allocate memory space.
12400 !
12410 !
12420 ALLOCATE COMPLEX Sky(3,4,Npts),Subsky(Npts),Data1(Npts)
12430 ALLOCATE COMPLEX Kcal(3,4,Npts),Subkcal(Npts),Dummy(Npts)
12440 ALLOCATE Pwr_mag(Ntrace),Dummy5(Npts),Dummy6(Npts)
12450 ALLOCATE Sig_zro(3,4),Sig_zro_sd(3,4)
12460 ALLOCATE Range(3)
12470 ALLOCATE Gate_cent(3),Gate_span(3)
12480 ALLOCATE Tim(3,4)
12490 MAT Sky= (CMPLX(0,0))
12500 Skyflg=0
12510 !
12520 ! Load Calibration Constants
12530 !
12540 CALL Load_cal(Kcal(*),Flg)
12550 IF Flg=1 THEN SUBEXIT
12560 CALL Angle_set(Angle)
12570 IF FNAsk("DO SKY MEASUREMENT?") THEN
12580 FOR F=1 TO 3
12590   CALL Freq_set(F)
12600   CALL Freq_sw(F)
12610   CALL Pol_sw(F,1)
12620   CALL Gate_set(Gate_cent(F),Gate_span(F),Angle,Range(F),Height,Beam(F),F)
12630   OUTPUT @Nwa;"GATECENT "&VALS(Gate_cent(F))&" S;"
12640   OUTPUT @Nwa;"GATESPAN "&VALS(Gate_span(F))&" S;"
12650   CALL Measure_sky(Sky(*),Meas_flag(*),Npts,F)
12660   Skyflg=1
12670 NEXT F
12680 END IF
12690 !
12700 ! Begin measurement sequence (step thru frequency and polarization).
12710 !
12720 FOR F=1 TO 3
12730   IF Meas_flag(1,F)=0 THEN 13350
12740   CALL Freq_set(F)
12750   CALL Freq_sw(F)
12760   IF Skyflg=0 THEN
12770     CALL Gate_set(Gate_cent(F),Gate_span(F),Angle,Range(F),Height,Beam(F),F)
12780   END IF
12790   OUTPUT @Nwa;"GATECENT "&VALS(Gate_cent(F))&" S"
12800   OUTPUT @Nwa;"GATESPAN "&VALS(Gate_span(F))&" S"
12810   OUTPUT @Nwa;"TIMDTRANOFF;"
12820   OUTPUT @Nwa;"POLA;ENTO;" ! Polar form for Re/Im pairs.
12830   OUTPUT @Nwa;"AUTO;OPC?;WAIT"
12840   ENTER @Nwa;Reply
12850 FOR P=1 TO 4
12860   IF Meas_flag(2,P)=0 THEN 13330
12870   MAT Subsky= Sky(F,P,*)
12880   MAT Subkcal= Kcal(F,P,*)
12890   CALL Pol_sw(F,P)
12900   DISP "READY FOR MEASUREMENT? <CONTINUE>";Bells
12910   PAUSE
12920   DISP "WORKING"
12930 !
12940   ! Average frequency trace points across band.
12950 !
12960 Tim(F,P)-TIMEDATE ! time of measurement.
12970 ! Display number of traces remaining in this set.
12980 PRINT TABXY(1,5),"Traces remaining = ";
12990 FOR K=1 TO Ntrace ! Loop through Ntrace traces
13000   ! Get a trace, consisting of Npts complex values, from the
13010   ! HP8753 using form3. Form1 can be used, if faster data
13020   ! transfer is required.
13030   CALL Get form3(Dummy(*),Npts,F)
13040   MAT Data1= Dummy-Subsky ! Subtract Sky
13050   MAT Dummy= Data1 . Subkcal ! Multiply by cal constant
13060   MAT Dummy5= ABS(Dummy)
13070   MAT Dummy6= Dummy5 . Dummy5 ! Convert to power
13080   ! Sum the power over Npts points and divide by Npts to get
13090   ! average power.
13100   Pwr_mag(K)=SUM(Dummy6)/Npts
13110   WAIT 2
13120   PRINT TABXY(20,50);
13130   PRINT USING "DDD";Ntrace-K
13140 NEXT K
13150 !
13160 ! Scale the data according to the calibration and illum integral.
13170 ! Then generate stats on the sigma zero measurement.
13180 !
13190 Sum=0
13200 Sumsq=0
13210 FK=1/(FNILL(Height,Angle,F,P)*Rcal(F)^4) ! Calc.the ill intg ratio
13220 FOR K=1 TO Ntrace

```

```

13230 ! Scale each trace by the illumination integral and sum.
13240 Sum=Sum+Pwr_mag(K)*Fk
13250 ! Also sum the squares.
13260 Sumsq=Sumsq+Pwr_mag(K)*Pwr_mag(K)*Fk*Fk
13270 NEXT K
13280 ! Sigma-zero is the average scaled power.
13290 Sig_zro(F,P)=Sum/Ntrace
13300 ! Also compute the std dev of the Ntrace valuesod sig-zro.
13310 Sig_zro_sd(F,P)=SQR((Ntrace*Sumsq-Sum*Sum)/(Ntrace*(Ntrace-1)))
13320 ! Sig_zro_sd(F,P)=Sig_zro_sd/SQR(Ntrace)
13330 NEXT P-
13340 OUTPUT @Nwa;" TIMDTRANON; WAIT;" 
13350 NEXT F
13360 Realtime=TIMEDATE ! Put current time in Realtime variable.
13370 ! Create a 10 character filename basedon the current time and date.
13380 CALL Date_string(Outfils,Realtime)
13390 CALL Catalog(Outfils,Mode$,Target$,Angle)
13400 !
13410 ! Write output to printer if this output type is set.
13420 !
13430 IF TRIMS(Out_type$)="PRINT" OR TRIMS(Out_type$)="PRINT/DISC" THEN
13440   PRINTER IS PRT
13450   PRINT
13460   PRINT "VERSION = ";Version$;" # OF POINTS = ";Npts;" # OF TRACES = ";Ntrace
13470   PRINT
13480   PRINT
13490   PRINT TAB(4); "FREQ"; TAB(12); "GATE CENT"; TAB(25); "GATE SPAN"; TAB(38);
13500   PRINT TAB(38); "R_CAL"; TAB(47); "RANGE"; TAB(56); "CAL_FILE"
13510   PRINT
13520   FOR F=1 TO 3
13530     IF Meas_flag(1,F)=0 THEN 13570
13540     PRINT TAB(6);Freq$(F);TAB(12);DROUND(Gate_cent(F),5);TAB(25);DROUND(Gate_span(F),5);
13550     PRINT TAB(37);DROUND(Rcal(F),4);TAB(46);DROUND(Range(F),4);
13560     PRINT TAB(55);Cal_keeps$(F)
13570   NEXT F
13580   PRINT
13590   PRINT
13600   PRINT TAB(4); "FREQ"; TAB(11); "POL"; TAB(20); "SIG_ZRO"; TAB(34); "STD. DEV."; TAB(51); "TIME"
13610   PRINT
13620   FOR F=1 TO 3
13630     IF Meas_flag(1,F)=0 THEN 13700
13640     FOR P=1 TO 4
13650       IF Meas_flag(2,P)=0 THEN 13680
13660       PRINT TAB(6);Freq$(F);TAB(11);Pol$(P);TAB(19);DROUND(FNDb(Sig_zro(F,P)),5);
13670       PRINT TAB(33);DROUND(Sig_zro_sd(F,P),5);TAB(49);TIME$(Tim(F,P));
13680     NEXT P
13690     PRINT
13700   NEXT F
13710   PRINTER IS CRT
13720 END IF
13730 !
13740 ! Create an output filename based on the current date and time.
13750 ! Data will be stored in Directory: /LCX/DATA .
13760 ! Then output important processing parameters and sigma zero data.
13770 !
13780 IF TRIMS(Out_type$)="DISC" OR TRIMS(Out_type$)="PRINT/DISC" THEN
13790   Bytes_per_Fec=64+(10+12+7+30+3*10+4*8+8*2+7*4)+(8*8)+(24*8)
13800   CREATE BDAT FNFileloc$(Outfils,"/LCX/DATA"\&Drive_a$),1,Bytes_per_rec
13810   ASSIGN @Disc TO FNFileloc$(Outfils,"/LCX/DATA"\&Drive_a$)
13820   OUTPUT @Disc;Outfils,Version$,Mode$,Target$,Cal_keeps$(*),Angle,Range(*),Meas_flag(*)
13830   OUTPUT @Disc;Sig_zro(*),Sig_zro_sd(*)
13840   OUTPUT @Disc;Sig_zro(*),Sig_zro_sd(*)
13850   ASSIGN @Disc TO F
13860 END IF
13870 !
13880 ! Place radar in default mode.
13890 !
13900   CALL Freq_sw(F disp)
13910   CALL Pol_sw(F disp,P disp)
13920 SUBEND ! Surface
13930 !*****SUBROUTINE TO COMPUTE VOLUME*****
13940 SUB Volume
13950 !
13960 ! This is the volume subroutine designed to measure volume
13970 ! backscattering coefficients for extended targets such as trees.
13980 !
13990 OPTION BASE 1
14000 COM /Paths/ @Nwa,@Nwa_data,@Hplib,@Relay
14010 COM /Constants/ Vel,Zero(*)
14020 COM /Sys 1/ Freq$(*),Freq_cen$(*),Freq_spa$(*)
14030 COM /Sys 2/ Pol$(*),Polsw$(*)
14040 COM /Sys 3/ INTEGER F disp,P disp
14050 COM /Sys 5/ Target$,Version$,Mode$,Out_type$,Sound$,Bells$,Debug$
14060 COM /Sys 6/ Angle,Angle$,Bin_rng(*),Beam(*),INTEGER Npts,Ntrace
14070 COM /Sys 7/ INTEGER Cal_flag(*),Meas_flag(*)
14080 COM /Cal/ Cal_keeps$(*),Rcal(*),Donecal(*)
14090 REAL Delm(3),Tim(3,4),Start_int(3),Stop_int(3)
14100 REAL Start_time(3),Stop_time(3),No_bytes

```

```

14110  REAL Bin_rng_old(3)
14120  INTEGER F,P,K,L,Flg,Ichk,Istep,Nbins(3),Ibeg,Iend,N,Nbins_max
14130 !
14140 ! Allocate memory space.
14150 !
14160  MAT Bin_rng_old= Bin_rng
14170  ALLOCATE COMPLEX Kcalm(3,4),Kcal(3,4,Npts)
14180  ALLOCATE COMPLEX Block(Npts)
14190  ALLOCATE INTEGER Binary(Npts,0:2),Bin(Ntrace,Npts,0:2)
14200  ALLOCATE Sigv_att(3,4,Npts)
14210  ALLOCATE Sigv_att_sd(3,4,Npts)
14220  ALLOCATE Sig_zro(3,4)
14230  ALLOCATE Sig_zro_sd(3,4)
14240  ALLOCATE Pwr_mag(Ntrace,Npts)
14250  ALLOCATE Sum(3,4,Npts)
14260  ALLOCATE Sumsq(3,4,Npts)
14270  ALLOCATE Range(3)
14280  ALLOCATE Range_s1(3,4,Npts)
14290  MAT Kcalm= (CMPLX(0,0))
14300  MAT Sigv_att= (0)
14310  MAT Sigv_att_sd= (0)
14320  MAT Sig_zro= (0)
14330  MAT Sig_zro_sd= (0)
14340  MAT Pwr_mag= (0)
14350  MAT Sum= (0)
14360  MAT Sumsq= (0)
14370 !
14380 ! Read in calibration data.
14390 !
14400  CALL Load_cal(Kcal(*),Flg)
14410  IF Flg=1 THEN SUBEXIT
14420 !
14430 ! Begin measurement sequence (step thru frequency and polarization).
14440 !
14450  FOR F=1 TO 3
14460    IF Meas_flag(1,F)=0 THEN 15460
14470    CALL Freq_sw(F)
14480    Ichk=0
14490    FOR P=1 TO 4
14500      IF Meas_flag(2,P)=0 THEN 15450
14510      CALL Pol_sw(F,P)
14520      ! Average the Calibration constants.
14530      FOR N=1 TO Npts
14540        Kcalm(F,P)=Kcalm(F,P)+Kcal(F,P,N)
14550      NEXT N
14560      MAT Kcalm= Kcalm/(Npts)
14570      OUTPUT @Nwa;"MARKDISC"
14580 !
14590 ! Set start and stop integration times (and other necessary
14600 ! parameters) for each new frequency.
14610 ! Prompt user for readiness.
14620 !
14630 IF Ichk=0 THEN
14640   CALL Integ_set(Start_int(F),Stop_int(F))
14650   Start_time(F)=Start_int(F)
14660   Delm(F)=(Stop_int(F)-Start_time(F))/(Npts-1)
14670   Istep=PROUND((72*Bin_rng(F)*Vel)/Delm(F),0)
14680   Bin_time=Istep*Delm(F)
14690   Nbins(F)=INT(-(Stop_int(F)-Start_int(F))/Bin_time)
14700   Stop_time(F)=Start_int(F)+Nbins(F)*Bin_time-Delm(F)
14710 Loop: Delm(F)=(Stop_time(F)-Start_time(F))/(Npts-1)
14720   Istep=PROUND((72*Bin_rng(F)*Vel)/Delm(F),0)
14730   Bin_time=Istep*Delm(F)
14740   Nbins(F)=INT(-(Stop_int(F)-Start_int(F))/Bin_time)
14750   IF Nbins(F)*Istep>NpE8 THEN
14760     Stop_time(F)=Stop_time(F)+Bin_time
14770     DISP "READJUSTING STOP_TIME"
14780     GOTO Loop
14790 END IF
14800 Stop_int(F)=Start_int(F)+Nbins(F)*Bin_time-Delm(F)
14810 Bin_rng(F)=Bin_time*.5*Vel
14820 Ichk=1
14830 END IF
14840 OUTPUT @Nwa;"STAR `@VALS(Start_time(F))` S"
14850 OUTPUT @Nwa;"STOP `@VALS(Stop_time(F))` S"
14860 OUTPUT @Nwa;"POLA;AUTO;ENTO" ! Smith chart form for Re/Im pairs.
14870 OUTPUT @Nwa;"OPC?;WAIT"
14880 ENTER @Nwa;Reply
14890 DISP "READY FOR MEASUREMENT? <CONTINUE>";Bell$
14900 PAUSE
14910 DISP "WORKING"
14920 !
14930 ! Transfer data using the binary form1 and convert to form3.
14940 ! Integrate the power in each bin for each of the (Ntrace)
14950 ! number of traces.
14960 !
14970 Tim(F,P)=TIMEDATE ! time of measurement.
14980 FOR K=1 TO Ntrace

```

```

14990     CALL Get_form1(Binary(*))
15000     MAT Bin(K,*,*)= Binary
15010     PRINT K
15020 NEXT K
15030 FOR K=1 TO Ntrace
15040     MAT Binary= Bin(K,*,*)
15050     CALL Convert(Block(*),Binary(*),Npts)
15060 ! Creating the sum & sum*sum of the raw trace to store on disc
15070 FOR N=1 TO Npts
15080     Pwr=ABS(Block(N))^2
15090     Sum(F,P,N)=Pwr
15100     Sumsq(F,P,N)=Pwr*Pwr
15110 NEXT N
15120 ! Determining the average power of each bin
15130 FOR L=1 TO Nbins(F)
15140     Ibeg=L*(L-1)*Istep
15150     Iend=L*Istep
15160     FOR M=Ibeg TO Iend
15170         Pwr_mag(K,L)=Pwr_mag(K,L)+ABS(Block(M)*Kcalm(F,P))^2
15180     NEXT M
15190     Pwr_mag(K,L)=Pwr_mag(K,L)/Istep
15200 NEXT L
15210 NEXT K
15220 !
15230 ! Scale the data according to the calibration and illum integral.
15240 ! Then generate stats on the sigma zero (volume attenuated) for
15250 ! each range bin and the volume as a whole.
15260
15270 Var_tot=0
15280 FOR L=1 TO Nbins(F)
15290     Sum_sl=0
15300     Sumsq_sl=0
15310     Range_sl(F,P,L)=.5*Vel*(Start_int(F)-Zero(F)+.5*(Istep-1)*Delm(F)+(L-1)*Istep*Delm(F))
15320     Height_sl=Range_sl(F,P,L)*COS(Angle)
15330     Fk=1/(Bin_rng(F)*COS(Angle))*FNILL(Height_sl,Angle,F,P)*Rcal(F)^4
15340     FOR K=1 TO Ntrace
15350         Sum_sl=Sum_sl+Pwr_mag(K,L)*Fk
15360         Sumsq_sl=Sumsq_sl+Pwr_mag(K,L)*Pwr_mag(K,L)*Fk*Fk
15370     NEXT K
15380     Var_sl=(Ntrace*Sumsq_sl-Sum_sl*Sum_sl)/(Ntrace*(Ntrace-1))
15390     Sigv_att(F,P,L)=Sum_sl/Ntrace
15400     Sigv_att_sd(F,P,L)=SQR(Var_sl)
15410     Sig_zro(F,P)=Sig_zro(F,P)+Bin_rng(F)*COS(Angle)*Sigv_att(F,P,L)
15420     Var_tot=Var_tot+(Bin_rng(F)*COS(Angle))^2*Var_sl
15430 NEXT L
15440     Sig_zro_sd(F,P)=SQR(Var_tot)
15450 NEXT F
15460 NEXT F
15470 Realtime=TIMEDATE
15480 CALL Date_string(Outfil$,Realtime)
15490 CALL Catalog(Outfil$,Mode$,Target$,Angle)
15500 !
15510 ! Write output to printer if this output type is set.
15520 !
15530 IF TRIM$(Out_type$)="PRINT" OR TRIM$(Out_type$)="PRINT/DISC" THEN
15540     PRINTER IS PRT
15550     PRINT
15560     PRINT "VERSION = ";Version$;" # OF POINTS = ";Npts;" # OF TRACES = ";Ntrace
15570     Sp$=" "
15580     FOR F=1 TO 3
15590         IF Meas_flag(1,F)=0 THEN 15800
15600         PRINT
15610         PRINT Freq$(F)&" BAND"
15620         PRINT Sp$;"START TIME: ";DROUND(Start_time(F),5);" STOP TIME: ";DROUND(Stop_time(F),5)
15630         PRINT Sp$;"START INT: ";DROUND(Start_int(F),5);" STOP INT: ";DROUND(Stop_int(F),5)
15640         PRINT Sp$;"DELM: ";DROUND(Delm(F),5);" RCAL: ";DROUND(Rcal(F),5);" BIN SIZE: "
15650         ";DROUND(Bin_rng(F),5)
15660         FOR P=1 TO 4
15670             IF Meas_flag(2,P)=0 THEN 15790
15680             PRINT
15690             PRINT
15700             PRINT Pol$(P)&" POLARIZATION"
15710             PRINT Sp$;" TIME OF MEAS: ";TIMES(Tim(F,P))
15720             PRINT Sp$;"BIN NO.:";TAB(21);"RANGE:";TAB(41);"SIGV_ATT:";TAB(61);"STD. DEV."
15730             FOR N=1 TO Nbins(F)
15740                 PRINT TAB(7);N;TAB(20);DROUND(Range_sl(F,P,N),5);TAB(40);DROUND(FNDb(Sigv_att(F,P,N)),5);TAB(60);DROUND(Sigv_att_sd(F,P,N),5)
15750             NEXT N
15760             PRINT
15770             PRINT Sp$;"SIG_ZRO: ";DROUND(FNDb(Sig_zro(F,P)),5)
15780             PRINT Sp$;"STD. DEV.: ";DROUND(Sig_zro_sd(F,P),5)
15790             NEXT P
15800             NEXT F
15810             PRINTER IS CRT
15820             END IF
15830 !

```

```

15840 ! Create an output filename based on the current date and time.
15850 ! Then output important processing parameters and sigma zero data.
15860 !
15870 IF TRIMS(Out_type$)="DISC" OR TRIMS(Out_type$)="PRINT/DISC" THEN
15880   Nbins_max=MAX(Nbins(*))
15890   No_bytes=919.+Nbins_max*4*24.+Npts*4*16.+10.
15900   CREATE BDAT Outfil$":,700,0,1",No bytes,1
15910   ASSIGN @Disc TO Outfil$":,700,0,1"
15920   OUTPUT @Disc;Outfil$,Version$,Mode$,Targets$,Angle,Range(*),Meas_flag(*);Cal_keep$(*)
15930   OUTPUT @Disc;Tim(*),Rcal(*)
15940   OUTPUT @Disc;Start_time(*),Stop_time(*)
15950   OUTPUT @Disc;Delm(*),Npts,Ntrace
15960   OUTPUT @Disc;Start_int(*),Stop_int(*)
15970   OUTPUT @Disc;Nbins(*),Bin_rng(*)
15980   OUTPUT @Disc;Sig_zro(*),Sig_zro_sd(*)
15990   CALL Array_pr(Range_s1(*),@Disc,Meas_flag(*),Nbins_max)
16000   CALL Array_pr(Sigv_att(*),@Disc,Meas_flag(*),Nbins_max)
16010   CALL Array_pr(Sigv_att_sd(*),@Disc,Meas_flag(*),NbIns_max)
16020   CALL Array_pr(Sum(*),@Disc,Meas_flag(*),Npts)
16030   CALL Array_pr(Sumsq(*),@Disc,Meas_flag(*),Npts)
16040 !
16050 ! MODIFIED TO PRINT Sig_zro's IN DISC MODE
16060 !
16070   PRINTER IS PRT
16080   FOR K=1 TO 3
16090     PRINT
16100     IF Meas_flag(1,K)=1 THEN PRINT Freq$(K),DROUND(FNDb(Sig_zro(K,1)),5),
16110       IF Meas_flag(1,K)=1 THEN PRINT DROUND(FNDb(Sig_zro(K,2)),5),DROUND(FNDb(Sig_zro(K,3)),5),
16120       NEXT K
16130     PRINTER IS CRT
16140   END IF
16150 !
16160 ! Place radar in default mode.
16170 !
16180 MAT Bin_rng= Bin_rng_old
16190 CALL Freq_sw(FDisp)
16200 CALL Pol_sw(F_DisP,P_DisP)
16210 SUBEND ! Volume
16220 !*****SUB Polarimetry
16230 !
16240 !
16250 ! This subroutine measures all 4 polarizations backscattering
16260 ! coefficients. It only stores the center frequency data.
16270 ! If more than one point is required, make sure that for C-Band
16280 ! the order of the data is reversed.
16290 !
16300 OPTION BASE 1
16310 COM /Baths/ @Nwa,@Nwa_data,@Hplib,@Relay
16320 COM /Constants/ Vel_Zero(*)
16330 COM /Sys_1/ Freq$(*),Freq_cen(*),Freq_span(*)
16340 COM /Sys_2/ Pol$(*),Polsw$(*)
16350 COM /Sys_4/ Drive_a$,Drive_b$,Drive_c$,INTEGER Preamble,Bytes
16360 COM /Sys_5/ Target$,Version$,Mode$,Out_type$,Sound$,Bell$,Debug$
16370 COM /Sys_6/ Angle,Angles,Bin_rng(*),Beam(*),INTEGER Npts,Ntrace
16380 COM /Sys_7/ INTEGER Cal_flag(*),Meas_flag(*)
16390 COM /Cal7/ Cal_keep$(*),Rcal(*),Donecal(*)
16400 INTEGER F,P,K,Flg,Ickk,Skyflg
16410 !
16420 ! Allocate memory space.
16430 !
16440 ALLOCATE INTEGER Binvv(Npts,0:2)
16450 ALLOCATE INTEGER Binhv(Npts,0:2)
16460 ALLOCATE INTEGER Binvh(Npts,0:2)
16470 ALLOCATE INTEGER Binhh(Npts,0:2)
16480 ALLOCATE INTEGER Binary(4,Ntrace,Npts,0:2)
16490 ALLOCATE INTEGER Dummy(Npts,0:2)
16500 ALLOCATE Height(3)
16510 ALLOCATE Range(3)
16520 ALLOCATE FK(4)
16530 ALLOCATE Sceig0(3,4,Ntrace)
16540 ALLOCATE Relphas(3,4,Ntrace)
16550 ALLOCATE Tim(3,Ntrace)
16560 ALLOCATE Gate_cen(3)
16570 ALLOCATE Gate_span(3)
16580 ALLOCATE COMPLEX Sky(3,4,Npts),Subsky(3,4)
16590 ALLOCATE COMPLEX Data1(3,4,Ntrace)
16600 ALLOCATE COMPLEX Kcalm(3,4),Kcal(3,4,Npts)
16610 MAT Sky= (CMPLX(0,0))
16620 Skyflg=0
16630 !
16640 ! Read in calibration data.
16650 !
16660 CALL Load cal(Kcal(*),Flg)
16670 IF Flg=1 THEN SUBEXIT
16680 PRINT Npts,Rcal(*)
16690 Midindex=(Npts-1)/2
16700 Kcalm(F,1)=Kcal(F,1,Midindex)

```

```

16710 Kcalm(F,2)=Kcal(F,2,Midindex)
16720 Kcalm(F,3)=Kcal(F,3,Midindex)
16730 Kcalm(F,4)=Kcal(F,4,Midindex)
16740 DEALLOCATE Kcal(*)
16750 !
16760 CALL Angle_set(Angle)
16770 FOR F=1 TO 3
16780   IF Meas_flag(1,F)=0 THEN 17860
16790   OUTPUT @Nwa;"GATEOFF"
16800   OUTPUT @Nwa;"LOGM"
16810   CALL Fred_set(F)
16820   CALL Fred_sw(F)
16830   Ichk=0
16840 !
16850 ! Take Ntrace measurements of all 4 polarizations for frequency F.
16860 ! Store individual measurements in a larger array.
16870 !
16880 OUTPUT @Nwa;"OPC?;WAIT"
16890 ENTER @Nwa;Reply
16900 FOR K=1 TO Ntrace
16910   OUTPUT @Nwa;"FORM1"
16920   IF F=1 THEN
16930     OUTPUT @Relay;"B56"
16940     DISP "READY FOR MEASUREMENT? <CONTINUE>;Bell$"
16950     PAUSE
16960     OUTPUT @Nwa;"SING;OUTPRAW1"
16970     ENTER @Nwa_data;Preamble,Bytes,Binvv(*)
16980     OUTPUT @Relay;"A5"
16990     WAIT .07
17000     OUTPUT @Nwa;"SING;OUTPRAW1"
17010     ENTER @Nwa_data;Preamble,Bytes,Binvv(*)
17020     OUTPUT @Relay;"A6"
17030     WAIT .07
17040     OUTPUT @Nwa;"SING;OUTPRAW1"
17050     ENTER @Nwa_data;Preamble,Bytes,Binhh(*)
17060     OUTPUT @Relay;"B5"
17070     WAIT .07
17080     OUTPUT @Nwa;"SING;OUTPRAW1"
17090     ENTER @Nwa_data;Preamble,Bytes,Binhh(*)
17100     Tim(F,K)=TIMEDATE
17110 END IF
17120 IF F=2 THEN
17130   OUTPUT @Relay;"B34"
17140   DISP "READY FOR MEASUREMENT? <CONTINUE>;Bell$"
17150   PAUSE
17160   OUTPUT @Nwa;"SING;OUTPRAW1"
17170   ENTER @Nwa_data;Preamble,Bytes,Binvv(*)
17180   OUTPUT @Relay;"A3"
17190   WAIT .07
17200   OUTPUT @Nwa;"SING;OUTPRAW1"
17210   ENTER @Nwa_data;Preamble,Bytes,Binvv(*)
17220   OUTPUT @Relay;"A4"
17230   WAIT .07
17240   OUTPUT @Nwa;"SING;OUTPRAW1"
17250   ENTER @Nwa_data;Preamble,Bytes,Binhh(*)
17260   OUTPUT @Relay;"B3"
17270   WAIT .07
17280   OUTPUT @Nwa;"SING;OUTPRAW1"
17290   ENTER @Nwa_data;Preamble,Bytes,Binhh(*)
17300   Tim(F,K)=TIMEDATE
17310 END IF
17320 IF F=3 THEN
17330   OUTPUT @Relay;"A34"
17340   DISP "READY FOR MEASUREMENT? <CONTINUE>;Bell$"
17350   PAUSE
17360   OUTPUT @Nwa;"SING;OUTPRAW1"
17370   ENTER @Nwa_data;Preamble,Bytes,Binvv(*)
17380   OUTPUT @Relay;"B3"
17390   WAIT .07
17400   OUTPUT @Nwa;"SING;OUTPRAW1"
17410   ENTER @Nwa_data;Preamble,Bytes,Binvv(*)
17420   OUTPUT @Relay;"B4"
17430   WAIT .07
17440   OUTPUT @Nwa;"SING;OUTPRAW1"
17450   ENTER @Nwa_data;Preamble,Bytes,Binhh(*)
17460   OUTPUT @Relay;"A3"
17470   WAIT .07
17480   OUTPUT @Nwa;"SING;OUTPRAW1"
17490   ENTER @Nwa_data;Preamble,Bytes,Binhv(*)
17500   Tim(F,K)=TIMEDATE
17510 END IF
17520 MAT Binary(1,K,*,*)=Binvv
17530 MAT Binary(2,K,*,*)=Binhh
17540 MAT Binary(3,K,*,*)=Binhv
17550 MAT Binary(4,K,*,*)=Binhv
17560 PRINT K
17570 NEXT K
17580 !

```

```

17590 ! Load calibration for the specified polarization and reload
17600 ! (one-by-one) the Ntrace measurements. Gate the first trace
17610 ! and output the voltage mag & phase at the center frequency.
17620 !
17630 ! NOTE: The gate changes for each new frequency but remains the
17640 ! same regardless of polarization for all Ntrace measurements.
17650 !
17660 FOR P=1 TO 4
17670   FOR K=1 TO Ntrace
17680     MAT Dummy= Binary(P,K,*,*)
17690     OUTPUT @Nwa;"FORM1;INPURAW1"
17700     OUTPUT @Nwa data;Preamble,Bytes,Dummy(*)
17710     IF Ichk=0 THEN
17720       CALL Gate_set(Gate_cent(F),Gate_span(F),Angle,Range(F),Height(F),Beam(F),F)
17730       Ichk=1
17740     ELSE
17750       OUTPUT @Nwa;"GATECENT "&VALS(Gate_cent(F))&" S"
17760       OUTPUT @Nwa;"GATESPAN "&VALS(Gate_span(F))&" S"
17770     END IF
17780     IF K=1 THEN OUTPUT @Nwa;"TIMDTRANOFF;GATEON"
17790     OUTPUT @Nwa;"POLA;AUTO;MARK1 "&VALS(Freq_cent(F))&" GHZ"
17800     OUTPUT @Nwa;"POLMRI;OUTPMARK"
17810     ENTER @Nwa;Data1(F,P,K)
17820     OUTPUT @Nwa;"OPC?;WAIT"
17830     ENTER @Nwa;Reply
17840   NEXT K
17850   NEXT P
17860 NEXT F
17870 !
17880 ! Prompt the user to do sky subtraction.
17890 IF FNAsk("DO SKY MEASUREMENT?") THEN
17900   FOR F=1 TO 3
17910     IF Meas_flag(1,F)=0 THEN 18000
17920     CALL Freq_set(F)
17930     CALL Freq_sw(F)
17940     CALL Pol_sw(F,1)
17950     OUTPUT @Nwa;"GATECENT "&VALS(Gate_cent(F))&" S;"
17960     OUTPUT @Nwa;"GATESPAN "&VALS(Gate_span(F))&" S;"
17980     CALL Measure_sky(Sky(*),Meas_flag(*),Npts,F)
17990     MAT Subsky_Sky(*,*,Midindex)
18000   NEXT F
18010 END IF
18020 !
18030 ! Convert to power and scale data to give square root of sigma zero
18040 ! and relative phase.
18050 !
18060 FOR F=1 TO 3
18070   IF Meas_flag(1,F)=0 THEN 18190
18080   FOR P=1 TO 4
18090     FK(P)=SQR(1/(FN111(Height(F),Angle,F,P)*Rcal(F)^4))
18100     FOR K=1 TO Ntrace
18110       Data1(F,P,K)=(Data1(F,P,K)-Subsky(F,P))*Kcalm(F,P)
18120       Sgsig0(F,P,K)=ABS(Data1(F,P,K))*FK(P)
18130       IF P>1 THEN
18140         Relphas(F,P,K)=ARG(Data1(F,P,K))-ARG(Data1(F,1,K))
18150         Relphas(F,P,K)=FNPhase_norm(Relphas(F,P,K))
18160       END IF
18170     NEXT K
18180   NEXT P
18190 NEXT F
18200 OUTPUT @Nwa;"FORM3"
18210 !
18220 Realtime=TIMEDATE
18230 CALL Date_string(Outfil$,Realtime)
18240 CALL Catalog(Outfil$,Mode$,Target$,Angle)
18250 !
18260 ! Write output to printer if this output type is set.
18270 IF TRIM$(Out_type$)="PRINT" OR TRIM$(Out_type$)="PRINT/DISC" THEN
18290   PRINTER IS PRT
18300   PRINT
18310   PRINT "VERSION = ";Version$; "# OF POINTS = ";Npts; "# OF TRACES = ";Ntrace
18320   PRINT
18330   PRINT
18340   PRINT
18350   TAB(4); "FREQ"; TAB(12); "GATE_CENT"; TAB(25); "GATE_SPAN"; TAB(38); "R_CAL"; TAB(47); "RANGE"; TAB(56); "CAL_FIE"
18360   PRINT
18370   FOR F=1 TO 3
18380     IF Meas_flag(1,f)=0 THEN 18410
18380     PRINT TAB(6);FreqS(F);TAB(12);DROUND(Gate_cent(F),5);TAB(25);DROUND(Gate_span(F),5);
18390     PRINT TAB(37);DROUND(Rcal(F),4);TAB(46);DROUND(Range(F),4);
18400     PRINT TAB(55);Cal_KeepS(F)
18410   NEXT F
18420   PRINT
18430   PRINT
18440   FOR F=1 TO 3

```

```

18450 IF Meas_flag(1,F)=0 THEN 18650
18460 PRINT
18470 PRINT
18480 PRINT Freq$(F); " BAND";TAB(12); "MAGNITUDE IS THE SQ ROOT OF SIGMA ZERO (REAL #)"
18490 PRINT TAB(12); "PHASE IS THE RELATIVE PHASE WITH RESPECT TO VV POL."
18500 PRINT
18510 PRINT TAB(12); "MAGVV";TAB(27); "MAGHH";TAB(42); "MAGHV";TAB(57); "MAGVH"
18520 PRINT
18530 FOR K=1 TO Ntrace
18540   PRINT TAB(11);DROUND(Sqsig0(F,1,K),5);TAB(26);DROUND(Sqsig0(F,2,K),5);
18550   PRINT TAB(41);DROUND(Sqsig0(F,3,K),5);TAB(56);DROUND(Sqsig0(F,4,K),5)
18560 NEXT K
18570 PRINT
18580 PRINT
18590 PRINT TAB(12); "PH_HH";TAB(27); "PH_HV";TAB(42); "PH_VH";TAB(58); "TIME"
18600 PRINT
18610 FOR K=1 TO Ntrace
18620   PRINT TAB(11);DROUND(Relphas(F,2,K),5);TAB(26);DROUND(Relphas(F,3,K),5);
18630   PRINT TAB(41);DROUND(Relphas(F,4,K),5);TAB(56);TIMES(Tim(F,K))
18640 NEXT K
18650 NEXT F
18660 PRINTER IS CRT
18670 END IF
18680 !
18690 ! Create an output filename based on the current date and time.
18700 ! Then output important processing parameters and scat. matrix
18710 ! data.
18720 !
18730 IF TRIMS(Out_type$)="DISC" OR TRIMS(Out_type$)="PRINT/DISC" THEN
18740 Bytes_per_rec=64+(10+12+7+30+3*10+4*8+2+7*4)+(8*8)+(15*8)+(3*7*Ntrace*8)
18750 CREATE BDAT FNfileloc$(Outfil$, "/LCX/DATA"&Drive$,1,Bytes_per_rec
18760 ASSIGN @Disc TO FNfileloc$(Outfil$, "/LCX/DATA"&Drive$,1)
18770 OUTPUT @Disc;Outfil$,Versions$,Modes$,Targets$,Cal_Keep$(*),Angle,Range(*),Meas_flag(*)
18780 OUTPUT @Disc;Gate cent(*),Gate_span(*),Npts,Ntrace
18790 OUTPUT @Disc;Tim(*),Rcal(*)
18800 CALL Array_prm(Sqsig0(*),@Disc,Meas_flag(*),Ntrace,0)
18810 CALL Array_prm(Relphas(*),@Disc,Meas_flag(*),Ntrace,1)
18820 ASSIGN @Disc TO *
18830 END IF
18840 !
18850 ! Place radar in default mode.
18860 !
18870 CALL Freq_sw(F_disp)
18880 CALL Pol_sw(F_disp,P_disp)
18890 SUBEND ! Polarimetry
18900 !*****SUB Measure_sky(COMPLEX Sky(*), INTEGER Flag(*),Npts,F)
18910 SUB Measure_sky(COMPLEX Sky(*), INTEGER Flag(*),Npts,F)
18920 !
18930 ! This subroutine measures sky for frequency F. It checks Flag(2,P)
18940 ! to measure the corresponding polarization. It uses form3 to
18950 ! transfer data and stores it in Sky(*).
18960 !
18970 OPTION BASE 1
18980 COM /Paths/ @Nwa,@Nwa_data,@Hpib,@Relay
18990 COM /Sys 3/ INTEGER F"disp,P disp
19000 COM /Sys 5/ Targets$,Versions$,Modes$,Out_type$,Sound$,Bell$,Debug$
19010 INTEGER F
19020 ALLOCATE COMPLEX Dummy2(Npts)
19030 DISP "POSITION FOR SKY MEASUREMENT <CONTINUE>";Bell$
19040 PAUSE
19050 DISP "WORKING"
19060 OUTPUT @Nwa;"TIMDTRANOFF; POLA; AUTO;"
19070 FOR P=1 TO 4
19080 IF Flag(2,P)=0 THEN 19140
19090 CALL Pol_sw(F,P)
19100 OUTPUT @Nwa;"AVERFACT 17;"
19110 OUTPUT @Nwa;"AVEROON; NUMG 17;"
19120 CALL Get form3(Dummy2(*),Npts,F)
19130 MAT Sky(F,P,*)= Dummy2
19140 NEXT P
19150 OUTPUT @Nwa;"AVEROFF; LOGM; TIMDTRANON; WAIT; CONT"
19160 CALL Pol_sw(F,1)
19170 SUBEND ! Measure_sky
19180 !*****SUB Integ_set(Start_int,Stop_int)
19190 SUB Integ_set(Start_int,Stop_int)
19200 !
19210 ! This subroutine is called by Volume to set the integration start
19220 ! and stop points.
19230 !
19240 COM /Paths/ @Nwa,@Nwa_data,@Hpib,@Relay
19250 COM /Sys 5/ Targets$,Versions$,Modes$,Out_type$,Sound$,Bell$,Debug$
19260 OUTPUT @Nwa;"MARK1"
19270 LOCAL 716
19280 DISP "SET INTEGRATION START POINT WITH KNOB OR KEY PAD <CONTINUE>";Bell$
19290 PAUSE
19300 REMOTE 716
19310 OUTPUT @Nwa;"OUTPACTI"
19320 ENTER @Nwa:Start_int

```

```

19330 LOCAL 716
19340 DISP "SET INTEGRATION STOP POINT WITH KNOB OR KEY PAD <CONTINUE>",Bells
19350 PAUSE
19360 REMOTE 716
19370 OUTPUT @Nwa;"OUTPACTI"
19380 ENTER @Nwa;Stop_int
19390 SUBEND ! Integ_set
19400 !*****
19410 SUB Freq_sw(INTEGER Ifreq)
19420 !
19430 ! This subroutine sets the relay actuator to Ifreq frequency.
19440 !
19450 COM /Paths/ @Nwa,@Nwa_data,@Hplib,@Relay
19460 SELECT Ifreq
19470 CASE 1
19480   OUTPUT @Relay;"?*A1B2"
19490 CASE 2
19500   OUTPUT @Relay;"?*A2B1"
19510 CASE 3
19520   OUTPUT @Relay;"?*B12"
19530 END SELECT
19540 WAIT .1
19550 SUBEND ! Freq_sw
19560 !*****
19570 SUB Table_def
19580 !
19590 ! This subroutine creates a look-up table used in the Convert subroutine,
19600 ! which converts form1 data to form3.
19610 !
19620 COM /Table/ Exp_tbl(0:255)
19630 Exp_tbl(0)=2^(-15)
19640 FOR I=0 TO 126
19650   Exp_tbl(I+1)=2*Exp_tbl(I)
19660 NEXT I
19670 Exp_tbl(128)=2^(-143)
19680 FOR I=128 TO 254
19690   Exp_tbl(I+1)=2*Exp_tbl(I)
19700 NEXT I
19710 SUBEND
19720 !*****
19730 SUB Convert(COMPLEX Block(*),INTEGER Binary(*),Npts)
19740 !
19750 ! This subroutine converts binary (form1) to complex base-10 numbers.
19760 !
19770 COM /Table/ Exp_tbl(*)
19780 FOR I=1 TO Npts
19790   Exp=Exp_tbl(BINAND(Binary(I,2),255))
19800   Block(I)=Binary(I,1)*Exp*CMPLX(1.,0.)+Binary(I,0)*Exp*CMPLX(0.,1.)
19810 NEXT I
19820 SUBEND
19830 !*****
19840 SUB Get_form1(INTEGER Binary(*))
19850 !
19860 ! This subroutine gets a trace of data from HP8753 in form1 and
19870 ! puts it in Binary(*).
19880 ! Make sure the order of frequency points for C-Band radar is
19890 ! not important for the routine which uses this subroutine.
19900 !
19910 COM /Paths/ @Nwa,@Nwa_data,@Hplib,@Relay
19920 INTEGER Preamble,Bytes
19930 Preamble=9025
19940 Bytes=Npts*6
19950 OUTPUT @Nwa;"FORM1;OUTPFORM"
19960 ENTER @Nwa_data;Preamble,Bytes,Binary(*)
19970 SUBEND
19980 !*****
19990 SUB Get_form3(COMPLEX Array1(*),INTEGER Npts,I)
20000 ! This Subroutine uses the form3 of HP8753 to transfer data into
20010 ! Array1(Npts). It also sets the GATE and reverses the order of data
20020 ! if measuring C-Band (I=2).
20030 OPTION BASE 1
20040 COM /Paths/ @Nwa,@Nwa_data,@Hplib,@Relay
20050 INTEGER Preamble,Bytes
20060 ALLOCATE COMPLEX Dummy3(Npts)
20070 OUTPUT @Nwa;"WAIT; FORM3;GATEON; WAIT; OUTPFORM;"
20080 ENTER @Nwa_data;Preamble,Bytes,Dummy3(*)
20090 OUTPUT @Nwa;"GATEOFF; WAIT;"
20100 !
20110 ! Reverse the order if C-Band
20120 !
20130 IF I=2 THEN
20140   FOR W=1 TO Npts
20150     Array1(W)=Dummy3(Npts-W+1)
20160   NEXT W
20170 ELSE
20180   MAT Array1= Dummy3
20190 END IF
20200 SUBEND ! Get_form3

```

```

20210 !*****SUBROUTINE Load_cal*****
20220 SUB Load_cal(Kcal(*),Flg)
20230 !
20240 ! This subroutine loads the calibration constants into Kcal(*).
20250 ! If Calibration has not been done in this run, it prompts the user
20260 ! to use an old calibration file. The file should be located in
20270 ! directory "/LCX/CAL".
20280 !
20290 OPTION BASE 1
20300 COM /Paths/ @Nwa,@Nwa,data,@Hplib,@Relay
20310 COM /Sys 4/ Drive_a$,Drive_b$,Drive_c$,INTEGER Preamble,Bytes
20320 COM /Sys-6/ Angle,Angle$(1U],Bin rng(3),Beam(3),INTEGER Npts,Ntrace
20330 COM /Sys-7/ INTEGER Cal_flag(*),Meas_flag(*)
20340 COM /Cal7 Cal keep$(*),Rcal(*),Donecal(*)
20350 DIM Outfiles[1U]
20360 INTEGER F,P
20370 Flg=0
20380 FOR F=1 TO 3
20390     IF Meas_flag(1,F)=0 THEN 20610
20400     IF Donecal(F)=1 THEN
20410         ASSIGN @Keep TO FNFileloc$(Cal keep$(F),"LCX/CAL"&Drive_a$)
20420         ENTER @Keep;Cal keep$(F),Npts,Rcal(F)
20430         ALLOCATE COMPLEX Subkcall(Npts),Subkcal2(Npts),Subkcal3(Npts),Subkcal4(Npts)
20440         ENTER @Keep;Subkcall(*),Subkcal2(*),Subkcal3(*),Subkcal4(*)
20450         ASSIGN @Keep TO *
20460         MAT Kcal(F,1,*)= Subkcall
20470         MAT Kcal(F,2,*)= Subkcal2
20480         MAT Kcal(F,3,*)= Subkcal3
20490         MAT Kcal(F,4,*)= Subkcal4
20500         DEALLOCATE Subkcall(*),Subkcal2(*),Subkcal3(*),Subkcal4(*)
20510     ELSE
20520         IF FNAsk("USE OLD CALIBRATION FILE?") THEN
20530             INPUT "ENTER CALIBRATION FILENAME?",Cal_keep$(F)
20540             Donecal(F)=1
20550             GOTO 20400
20560         ELSE
20570             Flg=1
20580             SUBEXIT
20590         END IF
20600     END IF
20610     NEXT F
20620 SUBEND ! Load cal
20630 !*****SUBROUTINE Catalog*****
20640 SUB Catalog(Filename$,Mode$,Target$,Angle)
20650 !
20660 ! This subroutine prints the 10-digit filename (containing the
20670 ! date and time of measurement) and the current mode,angle and
20680 ! target.
20690 !
20700 PRINTER IS PRT
20710 PRINT
20720 PRINT
20730 PRINT CHR$(27)&"(s1B"; ! Bold print on
20740 PRINT RPT$(">",70)
20750 PRINT "TIME/DATE: "&Filename$&" MODE: "&Mode$&" ANGLE: "&VALS(Angle)&CHR$(129)
20760 PRINT RPT$(" ",26)&"TARGET: "&Target$
20770 PRINT CHR$(27)&"(s0B"; ! Bold print off
20780 PRINTER IS CRT
20790 SUBEND ! Catalog
20800 !*****SUBROUTINE Array_pr*****
20810 SUB Array_pr(Array(*),@Disc,INTEGER Meas_flag(*),Num)
20820 !
20830 ! This subroutine is used by the Volume subroutine to store
20840 ! Array(*,*,*) on @Disc if corresponding Meas_flag is set.
20850 !
20860 OPTION BASE 1
20870 INTEGER F,P,N
20880 ALLOCATE Dummy(Num)
20890 FOR F=1 TO 3
20900     IF Meas_flag(1,F)=0 THEN 20980
20910     FOR P=1 TO 4
20920         IF Meas_flag(2,P)=0 THEN 20970
20930         FOR N=1 TO Num
20940             Dummy(N)=Array(F,P,N)
20950         NEXT N
20960         OUTPUT @Disc;Dummy(*)
20970     NEXT P
20980     NEXT F
20990 SUBEND ! Array_pr
21000 !*****SUBROUTINE Array_prm*****
21010 SUB Array_prm(Array(*),@Disc,INTEGER Meas_flag(*),Num,Ph)
21020 !
21030 ! This subroutine is used by the Polarimetry subroutine to store
21040 ! Array(F,*,*) on @Disc if Meas_flag(1,F) is set.
21050 ! If Array(F,*,*) contains the phase measurements (Ph=1), it does
21060 ! not store Array(F,1,*).
21070 !
21080 OPTION BASE 1

```

```

21090 INTEGER F
21100 ALLOCATE Dum1 (Num) ,Dum2 (Num) ,Dum3 (Num) ,Dum4 (Num)
21110 FOR F=1 TO 3
21120   IF Meas_flag(1,F)=0 THEN Next_freq
21130     MAT Dum1= Array(F,1,*)
21140     MAT Dum2= Array(F,2,*)
21150     MAT Dum3= Array(F,3,*)
21160     MAT Dum4= Array(F,4,*)
21170   IF Ph=0 THEN
21180     OUTPUT @Disc;Dum1(*)
21190   END IF
21200   OUTPUT @Disc;Dum2(*),Dum3(*),Dum4(*)
21210 Next freq:!
21220 NEXT F
21230 SUBEND ! Array prm
21240 !*****
21250 DEF FN111(H,Ang,INTEGER F,P)
21260   COM /Illum/ C(*)
21270   A=Ang*PI/180
21280   RAD
21290   I11=C(F,P,1)+C(F,P,2)*H+C(F,P,8)*H^(1/3)+C(F,P,9)*SQR(H)
21300   I11=I11+COS(A)*(C(F,P,3)+C(F,P,6)*H)+SIN(A)*(C(F,P,4)+C(F,P,5)*H)
21310   I11=I11+C(F,P,7)*COS(A)^3+C(F,P,13)*H*A^2
21320   I11=I11+C(F,P,10)*A+C(F,P,11)*A^2+C(F,P,12)*A^3
21330   I11=10^(I11/10)
21340   DEG
21350   RETURN I11
21360 FNEND ! I11
21370 !*****
21380 DEF FNDb(Number)
21390 !
21400 ! This function converts a number to "decibels".
21410 !
21420   Db=10*LGT(Number)
21430   RETURN Db
21440 FNEND
21450 !*****
21460 DEF FNPhase norm(Theta)
21470   ! This subroutine normalizes Theta to -180<Theta<180, if out of range.
21480   WHILE ABS(Theta)>180
21490     IF Theta>180 THEN
21500       Theta=Theta-360
21510     ELSE
21520       Theta=Theta+360
21530     END IF
21540 END WHILE
21550 RETURN Theta
21560 FNEND ! Phase norm
21570 !*****
21580 DEF FNAsk(Prompts$)
21590   ! This subroutine displays Prompts the user. If answer is "Y" it returns
21600   ! 1 otherwise it returns 0.
21610 OFF KEY
21620 DISP Prompts;
21630 INPUT "",Yn$ 
21640 Yn$=UPCS(Yn$[1,1])
21650 SELECT Yn$
21660   CASE = "Y"
21670     RETURN 1
21680   CASE = "N", =""
21690     RETURN 0
21700   CASE ELSE
21710     RETURN 0
21720 END SELECT
21730 FNEND ! Ask
21740 !
21750 !*****
21760 DEF FNFileloc$(File$,Dir$)
21770
21780 ! This function locates ":" in Dir$ and inserts File$ string between
21790 ! directory and drive string with a "/" after directory(if not there).
21800 ! For example if Dir$="LCX:,700,0", this function returns
21810 ! "/LCX/File$,700,0".
21820 INTEGER C ! for the location of the ':' in Dir$ (minus 1)
21830 LET C=POS(Dir$,:")-1
21840 IF C<=0 THEN
21850   RETURN TRIMS(File$&Dir$)
21860 ELSE
21870   RETURN Dir$[1,C]&RPTS("/",Dir$[C,C]<>"")&File$&Dir$[C+1,LEN(Dir$)]
21880 END IF
21890 FNEND ! Fileloc
21900 !*****
21910 SUB Sphere_response
21920 !
21930   ! Generate Theoretical calibration files.
21940 !
21950 OPTION BASE 1
21960 COM /Sys_4/ Drive_a$,Drive_b$,Drive_c$,INTEGER Preamble,Bytes

```

```

21970      COMPLEX A,B,C,E
21980      REAL Delta,Inc,Frequency,Radius,Vel,X
21990      REAL Fstart,Fstop,Fdelta
22000      INTEGER I,J,K,T,Npoints,Record,Template_flg
22010      M_sphere!:! Conducting sphere RCS.
22020      !-----
22030      Clear_crt
22040      GOSUB Get_sphere_pars
22050      SUBEXIT
22060      M_sph:   B=CMPLX(1,0)
22070      C=CMPLX(1,0)
22080      E=CMPLX(0,0)
22090      T=0
22100      LOOP
22110      T=T+1
22120      A=B
22130      B=C
22140      C=A+B*CMPLX(0, (1-2*T)/X)
22150      E=E+CMPLX(0,T+.5)/(C*(B+C*CMPLX(0,-T/X)))
22160      EXIT IF ABS(REAL(C))>1.E+10
22170      END LOOP
22180      Amp_ref(I)=(ABS(E))^2*4/X^2*PI*Radius*Radius
22190      Amp_ref(I)=SQR(Amp_ref(I))! Square root of sigma for polarimetry.
22200      DISP 20*LGT(Amp_ref(I))
22210      Phase_ref(I)=ARG(E)
22220      RETURN
22230      Get_sphere_pars!:! Gets relevant sphere parameters.
22240      !
22250      !
22260      Vel=2.998E+8
22270      INPUT "Starting frequency in GHz?",Fstart
22280      INPUT "Stopping frequency in GHz?",Fstop
22290      INPUT "Number of points?",Npoints
22300      INPUT "Sphere diameter in inches?",Radius
22310      Radius=Radius/2
22320      Radius=Radius*25.4/1000! Now in meters
22330      ALLOCATE Freq_ref(Npoints),Amp_ref(Npoints),Phase_ref(Npoints)
22340      !
22350      Fdelta=(Fstop-Fstart)/(Npoints-1)
22360      I=0
22370      DISABLE
22380      FOR J=I+1 TO Npoints STEP 1
22390      Frequency=Fstart+I*Fdelta
22400      I=I+1
22410      X=2*PI*Frequency*10^9/Vel*Radius
22420      GOSUB M_sph
22430      Freq_ref(I)=Frequency
22440      PRINT I,Freq_ref(I),Amp_ref(I),Phase_ref(I)
22450      NEXT J
22460      BEEP
22470      ENABLE
22480      New_file_flg=1
22490      INPUT "Store metal sphere RCS data as (eg.SPH12C)",File_ref$ 
22500      ALLOCATE TS[80]
22510      LINPUT "Enter file title:",TS
22520      Bytes_per_rec=256+3*8*N
22530      CREATE BDAT File_ref$&Drive_c$,1,Bytes_per_rec
22540      ASSIGN @Datafile TO File_ref$&Drive_c$;FORMAT OFF
22550      OUTPUT @Datafile,1;File_Ref$,TS,Bytes,Preamble,Npoints
22560      OUTPUT @Datafile;Freq_ref(*),Amp_ref(*),Phase_ref(*)
22570      ASSIGN @Datafile TO *
22580      COPY File_ref$&Drive_c$ TO FNFileLoc$(File_ref$,"/LCX/SphTheor"&Drive_a$)
22590      PURGE File_ref$&Drive_c$
22600      DEALLOCATE TS,Freq_ref(*),Amp_ref(*),Phase_ref(*)
22610      Clear_crt
22620      RETURN
22630      Quit:SUBEXIT
22640      SUBEND ! Sphere response
22650      !*****!
22660      SUB Set_clock
22670      !
22680      ! This subroutine asks the user for date & time and sets the HP clock.
22690      !
22700      OPTION BASE 1
22710      INTEGER I
22720      DIM Chrono$(12),Month$(12)[3]
22730      Exec_key$=CHR$(255)&CHR$(88)
22740      READ Month$(*)
22750      DATA "JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP","OCT","NOV","DEC"
22760      OUTPUT KBD;"SCRATCH KEY "&Exec_key$;
22770      Clear_crt
22780      PRINT "-" Current system date: ";DATES(TIMEDATE)
22790      PRINT " Current system time: ";TIMES(TIMEDATE)
22800      PRINT
22810      Ask:LINPUT "Enter date and time (YYMMDDHHMMss) :",Chrono$ 
22820      IF Chrono$="" AND DATES(TIMEDATE)<>" 1 Mar 1900" THEN
22830      Clear_crt
22840      SUBEXIT

```

```

22850    END IF
22860    Year$=VALS(1900+VAL(Chrono$[1,2]))
22870    IF (VAL(Chrono$[3,4])<=0 OR VAL(Chrono$[3,4])>12) THEN
22880        BEEP
22890        PRINT "Incorrect month value."
22900        GOTO Ask
22910    END IF
22920    Year$=Month$(VAL(Chrono$[3,4])) & " " & Year$
22930    Year$=Chrono$[5,6] & " " & Year$
22940    SET TIMEDATE (DATE(Year$))
22950    IF (VAL(Chrono$[7,8]))>23 THEN
22960        BEEP
22970        PRINT "Incorrect hour value."
22980        GOTO Ask
22990    END IF
23000    Day$=Chrono$[7,8] & ":" 
23010    IF VAL(Chrono$[9,10])>59 THEN
23020        BEEP
23030        PRINT "Incorrect minute value."
23040        GOTO Ask
23050    END IF
23060    Day$=Day$ & Chrono$[9,10] & ":" 
23070    IF (LEN(Chrono$)>10 AND LEN(Chrono$)=12) THEN
23080        IF VAL(Chrono$[11,12])>59 THEN
23090            BEEP
23100            PRINT "Incorrect seconds value."
23110            GOTO Ask
23120        END IF
23130        Day$=Day$ & Chrono$[11,12]
23140    ELSE
23150        Day$=Day$ & "00"
23160    END IF
23170    SET TIME TIME(Day$)
23180    Clear_crt
23190    SUBEXIT
23200    SUBEND ! Set_clock
23210 !*****SUBROUTINES*****!
23220 SUB Fix_error
23230 !
23240 ! This is the error correction subroutine. Certain errors, such as
23250 ! a disc full error, can be fixed without stopping the program.
23260 !
23270     SELECT ERRN
23280     CASE ELSE
23290         PRINTER IS CRT
23300         PRINT "ERROR ";ERRN
23310         PRINT ERMS
23320         PRINT "THE PROGRAM IS PAUSED. FIX ERROR, IF POSSIBLE, AND CONTINUE."
23330         PAUSE
23340     END SELECT
23350 SUBEND
23360 !*****

```

WARNING \*\*\*  
nes were too long and were wrapped.  
WARNING \*\*\*

UNIVERSITY OF MICHIGAN



3 9015 02829 9850