RCS MEASUREMENT OF POLARIMETRIC ACTIVE RADAR CALIBRATORS

by

Kamal Sarabandi

Yisok Oh

Radiation Laboratory

Department of Electrical Engineering and Computer Science

The University of Michigan

PI: Fawwaz T. Ulaby

JPL Contract 958744

June 19, 1990

ongn

UMKBUZ!

Abstract

Practical aspects of polarimetric measurement of active radar calibrators (PARC) are discussed in this report. A new polarimetric calibration technique (STCT) is employed for polarimetric radar cross section measurement of L- and C-band PARCs. The amplitude and phase of the scattering matrix elements are given over a wide range of incidence angle for azimuth, elevation, 45°, and 135° planes. Tables for scattering matrix elements and the polarization signatures at boresight are also provided.

Contents

List of Figures

List of Tables

1 Introduction

Airborne and space-borne polarimetric imaging SAR has been shown to be an important remote sensing tool for the acquisition of quantitative information about the earth's environment on a global scale. In order to utilize the quantitative data scientifically, external calibration of the radar system is necessary. The external calibration of radar systems is performed by using a target with a specified radar cross section (RCS) known as a calibration target.

Calibration targets, in general, can be categorized into two major groups: 1) passive calibrators and 2) active calibrators [1]. Although passive calibrators are more stable and reliable than the active ones, they are less desirable because of their large physical dimensions. In recent years, polarimetric active radar calibrators (PARC) have been used extensively and are planned to be employed for external calibration of SAR systems [2] in support of future spaceborne missions.

The success of external calibration relies on the knowledge of the scattering matrix of the calibration target(s). Although it may be possible to estimate the elements of the scattering matrix for a calibration target to a reasonable extent, the manufacturing tolerances always leave a great deal of uncertainty in the estimation. Therefore it is imperative to measure the calibrators against a precise calibration target (such as a metallic sphere) under laboratory condition. This also reveals another drawback for passive calibrators with large physical dimensions, namely that the far-field condition and uniform illumination criteria cannot be met in the laboratory.

Until recently, it has been very difficult to measure the scattering matrices of targets over a wide range of incidence angle and frequency with the desired accuracy. Advances in technology and calibration methods have made it possible to measure the elements of the scattering matrix with an accuracy of 0.5 dB in magnitude and 5° in phase [3].

The purpose of this report is to document measurements of JPL's L- and C-band PARCs using the new calibration technique (STCT) given in Appendix A. The measurements were performed in the UM anechoic chamber using a fully automated polarimetric scatterometer[4].

2 System Configuration and Measurement Setup

The polarimetric measurements of the PARCs were performed by L- and C-band scatterometers. A simplified block diagram of the scatterometer system is shown in Fig. (1). The scatterometer is an HP 8753A-based system with both phase and amplitude measurement capability and 100 dB dynamic range. The ability of the network analyzer to generate the time domain response of the frequency measurement allows us to separate the unwanted short-range signals from the desired target response (known as software gating). The sequence of polarization selection, data collection, and target orientation is performed via an HP 9000 series computer. The relay actuator energizes the frequency and polarization switches. The amplifier and pulsing network eliminates the short-range returns from the antenna and circulators to increase the dynamic range for RCS measurements [5].

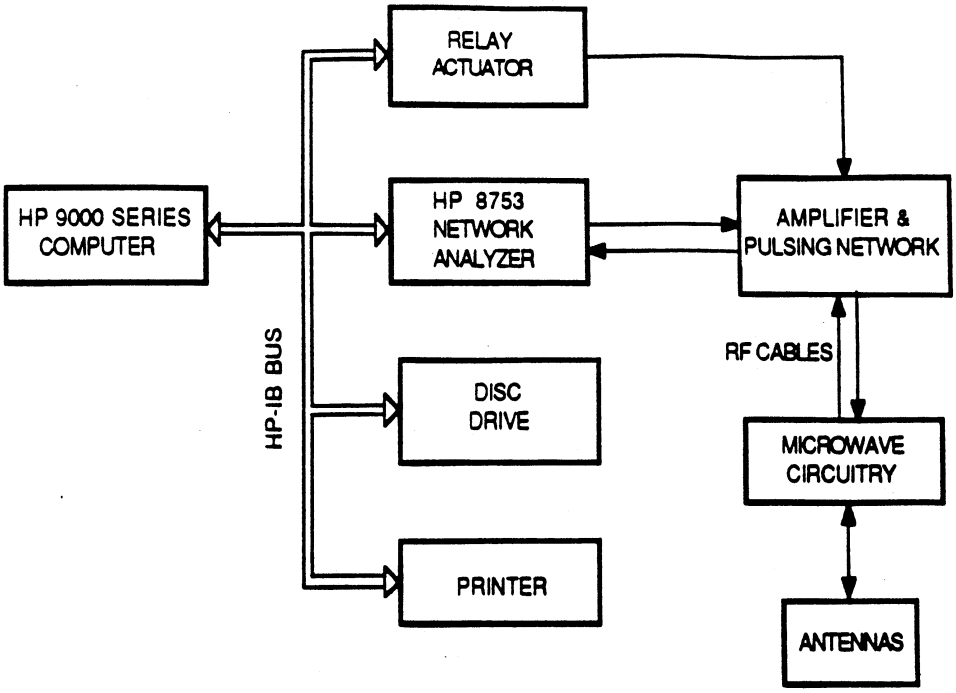

Figure 1: Block diagram of the scatterometer system.

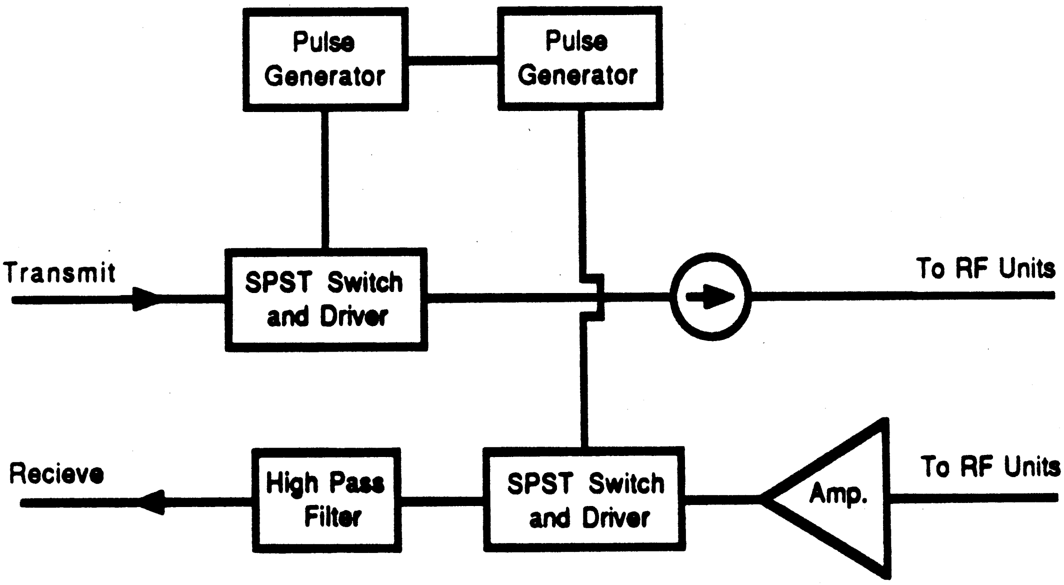

Figure 2: Block diagram of the pulsing network.

In this scheme, the receiver is switched off during transmission and then reconnected when the target return is expected to arrive at the receiver. Since the switching is done at a much higher rate than the receiver's bandwidth the network analyzer does not sense that the incoming signal is pulsed and it is measured as if it were a CW signal. The block diagram of the amplifier and pulsing network is shown in Fig. (2).

The synthesized source of the network analyzer spans the frequency range 300 KHz to 3 GHz and therefore for C-band up- and down-convertors are used. The block diagrams of the L- and C-band microwave circuitry are given in Figs. (3) and (4), respectively. The up-convertor of the C-band unit is a very stable microwave source operating at 6.5 GHz. The frequency range of the network analyzer for C-band measurement must be set to 1-1.5 GHz in order to operate the C-band scatterometer at 5-5.5 GHz. The operating frequency of the L-band system is 1.1-1.4 GHz. A new orthomode transducer (OMT) and dual-polarized antenna were designed for this project at L-band. The overall cross-polarization isolation of the new OMT and antenna is better than 35 dB.

Because PARC has a high gain amplifier, reflection from nearby objects might increase the feedback which would cause oscillation. To avoid this problem and also to have a very good signal to background ratio, the PARCs were mounted on a styrofoam pedestal in an anechoic chamber. The correct position of the PARCs with respect to the antenna coordinate system was accomplished by an azimuth-over-elevation positioner as depicted in Fig. (5). The azimuth turntable

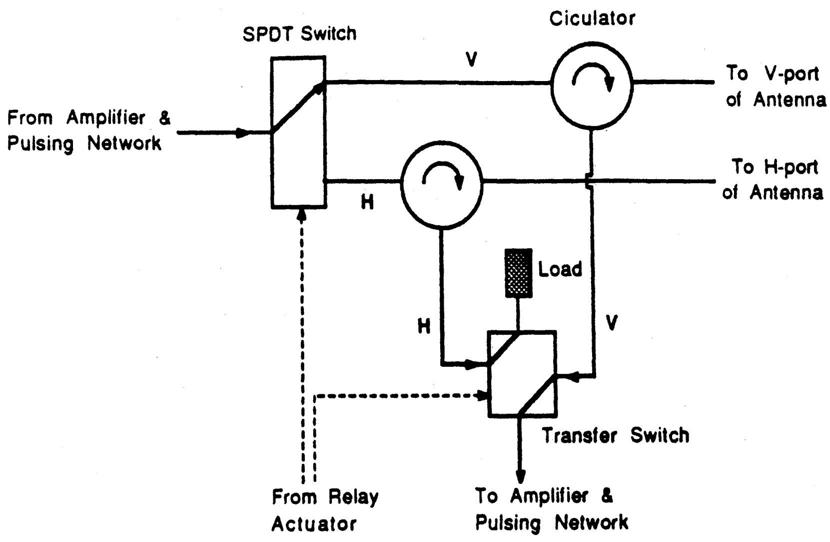

Figure 3: Block diagram of the L-band microwave circuitry.

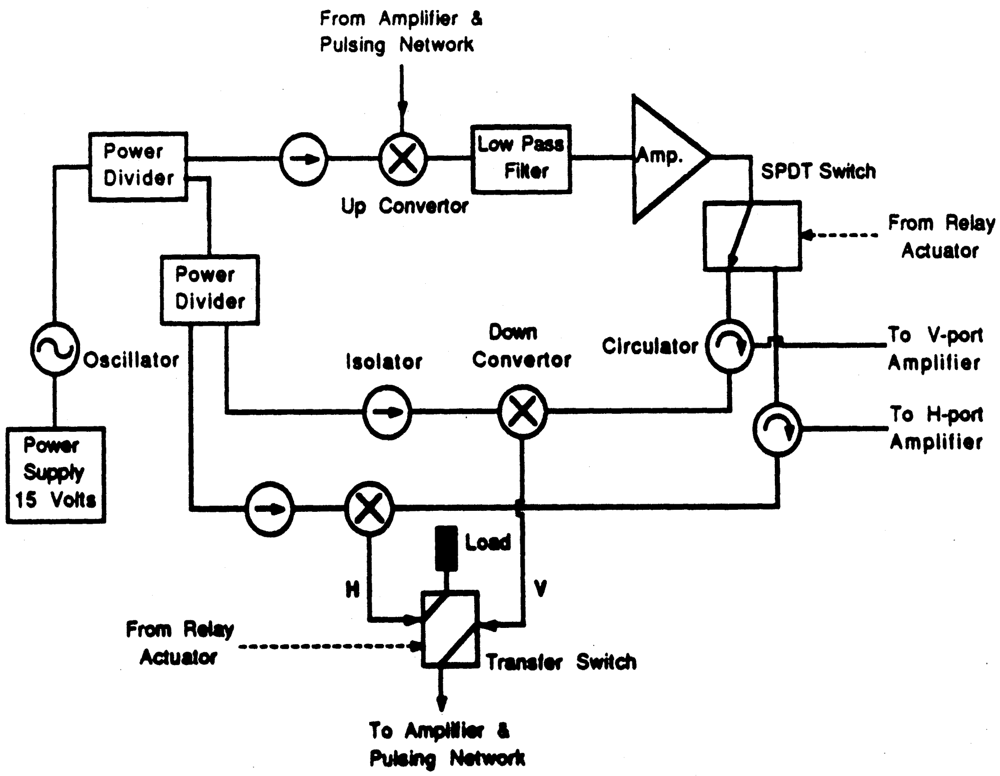

Figure 4: Block diagram of the C-band microwave circuitry.

is a computer controlled stepper motor with an accuracy of a fraction of a tenth of a degree and the elevation controller is a precise analog positioner.

3 Calibration Procedure

To calibrate the PARCs, we used the single-target calibration technique (STCT) described in Appendix A. The error in measurement of scattering matrix using this technique is less than 0.5 dB in amplitude and 5° in phase. With the STCT the antenna cross-talk contamination and channel imbalances are obtained by measuring only a single calibration target, namely a metallic sphere. This technique is immune to errors caused by target alignment with respect to the antenna coordinate system.

Using a four-port network approach it is shown that the measured scattering matrix of a target with scattering matrix **s** is given by

$$\mathbf{M} = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix} \begin{bmatrix} 1 & C \\ C & 1 \end{bmatrix} \mathbf{s} \begin{bmatrix} 1 & C \\ C & 1 \end{bmatrix} \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix} + \mathbf{N}$$

where the **R** and **T** matrices are the receive and transmit channel imbalances, C is the antenna cross-talk contamination factor, and **N** is a matrix representing thermal noise and background reflections. The background contribution can be obtained by measuring the empty chamber and the effect of the thermal noise can be minimized by an integration process.

By denoting the measured scattering matrix elements of the sphere and the PARC, respectively, by m_{ij}^{o} and m_{ij}^{u}, the unknown scattering matrix elements of the PARC can be obtained from

$$
\begin{aligned}
s_{vv} &= \frac{1}{(1-C^2)^2}\left[-2C^2\left(\frac{m^u_{12}}{m^o_{12}}+\frac{m^u_{21}}{m^o_{21}}\right)+(1+C^2)\left(\frac{m^u_{11}}{m^o_{11}}+C^2\frac{m^u_{22}}{m^o_{22}}\right)\right]s^o \\
s_{hh} &= \frac{1}{(1-C^2)^2}\left[-2C^2\left(\frac{m^u_{12}}{m^o_{12}}+\frac{m^u_{21}}{m^o_{21}}\right)+(1+C^2)\left(\frac{m^u_{22}}{m^o_{22}}+C^2\frac{m^u_{11}}{m^o_{11}}\right)\right]s^o \\
s_{vh} &= \frac{C}{(1-C^2)^2}\left[2\frac{m^u_{12}}{m^o_{12}}+2C^2\frac{m^u_{21}}{m^o_{21}}-(1+C^2)\left(\frac{m^u_{11}}{m^o_{11}}+\frac{m^u_{22}}{m^o_{22}}\right)\right]s^o \\
s_{hv} &= \frac{C}{(1-C^2)^2}\left[2\frac{m^u_{21}}{m^o_{21}}+2C^2\frac{m^u_{12}}{m^o_{12}}-(1+C^2)\left(\frac{m^u_{11}}{m^o_{11}}+\frac{m^u_{22}}{m^o_{22}}\right)\right]s^o
\end{aligned}
$$

where s^o is the theoretical value for the diagonal elements of the sphere's scattering matrix. The cross-talk contamination factor is given by

$$C = \pm\frac{1}{\sqrt{a}}\left(1-\sqrt{1-a}\right)$$

where $a \triangleq \frac{m^o_{12}m^o_{21}}{m^o_{11}m^o_{22}}$ and the branch of the square root is chosen such that $\mathrm{Re}\left[\sqrt{1-a}\right] > 0$. The uncertainty on the sign of C can be removed when measuring PARCs because the general trend of phase behavior is known.

4 Experimental Results

The measurements were performed in a 14-meter long anechoic chamber. The PARC under test was mounted on a styrofoam pedestal on an azimuth-over-elevation positioner. To assure that the PARC is not oscillating in the chamber, the energy flow between the receive and transmit antenna was monitored by inserting a 20 dB directional coupler between the amplifier and the transmit antenna. Also to avoid saturating the PARC's amplifier and scatterometer's receiver, the scatterometer transmit power was adjusted such that the PARC output power was around 5 dBm. For the actual RCS measurements the directional coupler was removed.

A 12-inch metallic sphere was used as the calibration target, and the signal to noise ratio was better than 30 dB in all cases. The L- and C-band measurements were performed over the frequency range 1.1-1.4 GHz and 5-5.5 GHz, respectively. All the data presented in this report are measurements at the center frequencies, namely 1.25 GHz for L-band and 5.3 GHz for the C-band. The backscattered radar cross section patterns were measured over the range of incidence angle from -40 to +40 degrees relative to the boresight direction in the azimuth, elevation, 45°, and 135° planes.

The radar cross section of a PARC can be decomposed into two components. The first component is the contribution of the front panel and the antennas and the second component is the contribution of the delay line and amplifier. Using the range-gating capability of the scatterometers, the two responses can be separated. Figs. (6) and (7) represent the time-domain responses of an L-band and a C-band PARC, respectively, where the relative amplitudes of the two components can be compared. In all of the results shown in this report, the contribution of the front panel and the antennas has been gated out.

In this study three L-band and three C-band PARCs were measured. The L-band PARCs are designated according to their serial number and the C-band PARCs designations are according to Fig. (8). In order to measure the RCS patterns in the desired planes, the PARCs were oriented as shown in Fig. (9). Tables 1 and 2 give the measured elements of the scattering matrices for L- and C-band at boresight for quick reference.

The polarization signature of PARCs L1 and C1 are plotted in Figs. (10) - (13). It is shown that co-polarized signatures for the case where the antennas are oriented 45° with respect to vertical direction (45° and 135° for L-band and azimuth and elevation for C-band) are slightly different from the theoretical response due to small errors in the phase measurement. However, for the case where the antennas are parallel and perpendicular to the vertical direction (azimuth and elevation for L-band and 45° and 135° for C-band) the errors in phase measurement do not affect the polarization signatures.

In Figs. (14) - (49) the measured RCS patterns of L- and C-band PARCs are shown. The C-band patterns seem reasonable but the L-band patterns are not quite symmetric, the maximum RCS appears at about 10° from boresight, and the measured phases are very much different from the expected values. These problems are due to the fact that the transmit and receive antennas are very close to each other (touching). The antennas are in the near field of each other and therefore the amplitude and phase patterns are different from the patterns of isolated antennas.

5 Conclusions

The radar cross section patterns of three L-band and three C-band PARCs were measured polarimetrically. The measurements were performed in an anechoic chamber and accurate orientation of the PARCs was achieved by a very precise azimuth-over-elevation positioner.

The measurements show that the RCS is not very sensitive to changes in the azimuth and elevation angles, but it is very sensitive to the rotation angle about the boresight direction. It is found that when the PARCs transmit and receive antennas are parallel and perpendicular to the vertical direction, small errors in the orientation angles or errors in phase measurement have a minor effect on calibration accuracy. It is also found that the close proximity of the L-band antennas has caused significant problems for the patterns and considerable errors in the scattering matrix elements. We recommend that the antennas be separated by at least one wavelength.

References

[1] Brunfeldt, D.R., and F.T. Ulaby, " Active calibrators for radar calibration ", *IEEE Trans. Geosci. Remote Sensing*, vol 22, no 2, 1984.

[2] Freeman, A., Y. Shen, and C.L. Werner, " Polarimetric SAR calibration experiment using active radar calibrators", *IEEE Trans. Geosci. Remote Sensing*, vol. 28, no. 2, 1990.

[3] Sarabandi, K., and F.T. Ulaby, " A convenient technique for polarimetric calibration of radar systems", *IEEE Trans. Geosci. Remote Sensing*, submitted for publication.

[4] Tassoudji, M.A., K. Sarabandi, and F.T. Ulaby, "Design consideration and implementation of the LCX polarimetric scatterometer (POLARSCAT)", *Radiation Laboratory Report No. 022486-T-2*, The University of Michigan, June 1989.

[5] Liepa, V.V., K. Sarabandi, and M.A. Tassoudji, " A pulsed network analyzer based scatterometer", *Proc. of IEEE Geosci. Remote Sens. Symp.*, Vancouver, July 1989.

		vv	vh	hv	hh
L_1	Az	1.9∠0.0	-31.1∠91.3	36.72∠121.4	1.4∠-120.0
	El	-6.1∠0.0	35.0∠14.2	-53.2∠-27.6	-2.5∠-84.6
	45°	31.2∠0.0	30.6∠-13.4	30.6∠197.0	29.9∠184.5
	135°	31.3∠0.0	30.9∠177.8	30.9∠19.4	30.8∠185.2
L_2	Az	-3.4∠0.0	-24.6∠-9.1	35.8∠153.0	-3.3∠-34.5
	El	-3.0∠0.0	36.4∠184.9	-47.6∠-81.2	-2.7∠20.0
	45°	29.9∠0.0	29.5∠9.0	29.5∠175.7	29.2∠184.9
	135°	30.8∠0.0	30.6∠187.8	30.6∠-6.5	30.2∠182.4
L_3	Az	-2.7∠0.0	-57.6∠-74.1	36.6∠173.4	-2.6∠186.6
	El	-17.4∠0.0	38.8∠-56.6	-44.2∠202.4	-1.7∠101.9
	45°	32.2∠0.0	31.6∠8.0	31.8∠179.1	31.9∠182.7
	135°	32.6∠0.0	32.2∠186.7	32.2∠0.3	32.3∠183.3

Table 1: Elements of scattering matrix for L-band PARCs. The entries are in the form $A_{ij}\angle\phi_{ij}$, where A_{ij} is the RCS σ_{ij} in dBsm ($\sigma_{ij} = 4\pi|s_{ij}|^2$) and ϕ_{ij} is the phase of s_{ij} in degrees.

		vv	vh	hv	hh
C_1	Az	27.2∠0.0	27.1∠187.2	27.1∠-1.7	27.1∠185.5
	El	27.2∠0.0	27.1∠-3.1	27.0∠183.7	26.8∠180.8
	45°	-1.9∠0.0	-24.2∠119.0	32.8∠-59.0	-7.1 ∠182.2
	135°	-1.2∠0.0	33.0∠-52.3	-46.0∠122.9	-3.1∠-163.6
C_2	Az	26.7∠0.0	26.7∠-2.7	26.7∠181.6	26.6∠179.3
	El	26.6∠0.0	26.7∠180.1	26.7∠-0.2	26.8∠180.3
	45°	-19.6∠0.0	32.5∠-148.8	-46.6∠52.2	-10.9 ∠145.8
	135°	-40.2∠0.0	-55.2∠6.3	32.4 ∠-55.7	-15.6∠186.5
C_3	Az	26.5∠0.0	26.4∠1.0	26.4∠178.6	26.2∠179.6
	El	26.3∠0.0	26.1∠180.0	26.1∠-1.3	25.9∠178.9
	45°	-5.2∠0.0	32.3∠145.3	-51.0∠-18.6	-8.3∠179.3
	135°	-8.5∠0.0	-44.3∠-141.4	31.5∠46.8	-3.8∠184.4

Table 2: Elements of scattering matrix for L-band PARCs. The entries are in the form $A_{ij}\angle\phi_{ij}$, where A_{ij} is the RCS σ_{ij} in dBsm ($\sigma_{ij} = 4\pi|s_{ij}|^2$) and ϕ_{ij} is the phase of s_{ij} in degrees.

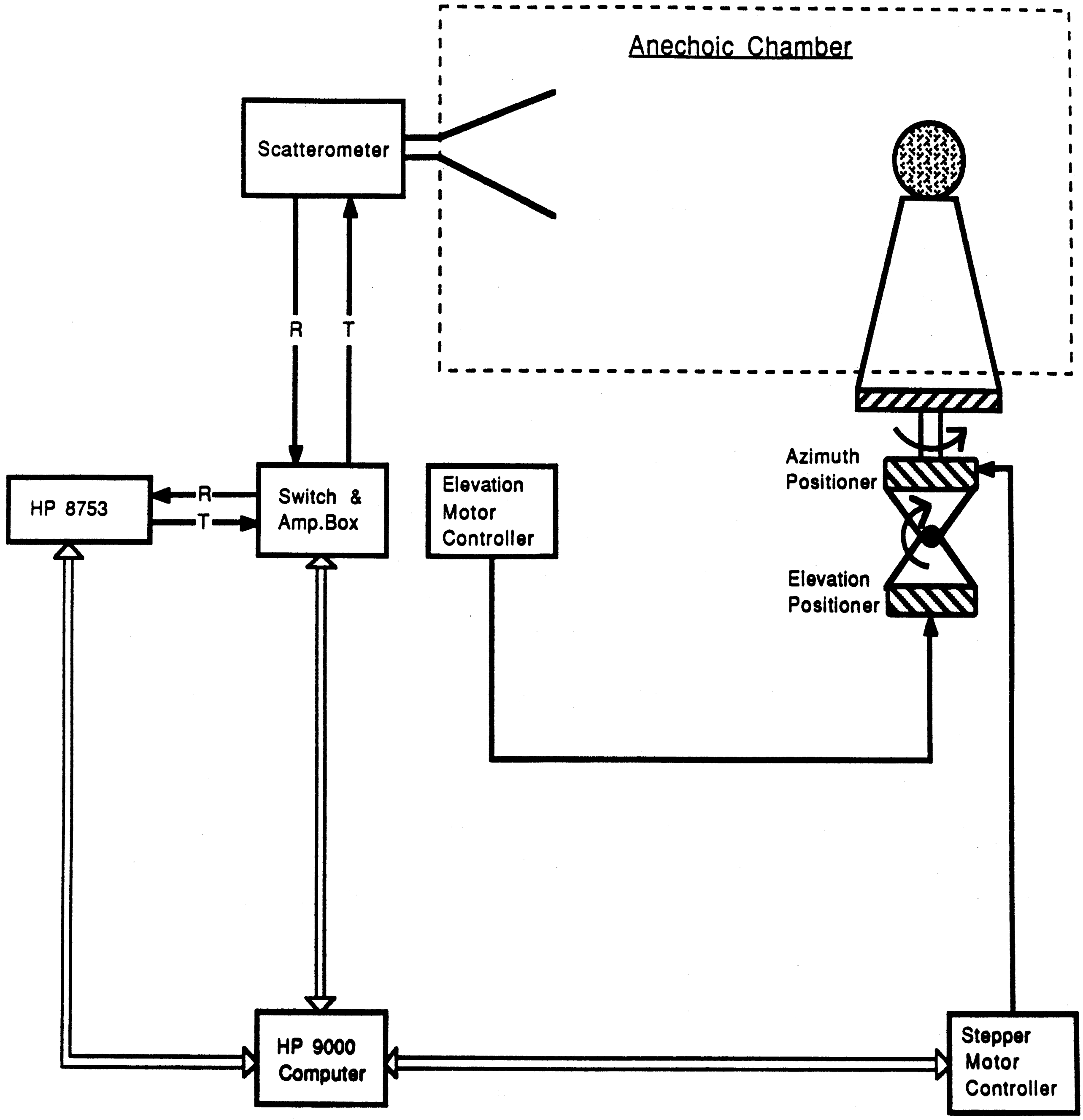

Figure 5: Automatic radar cross section measurement setup.

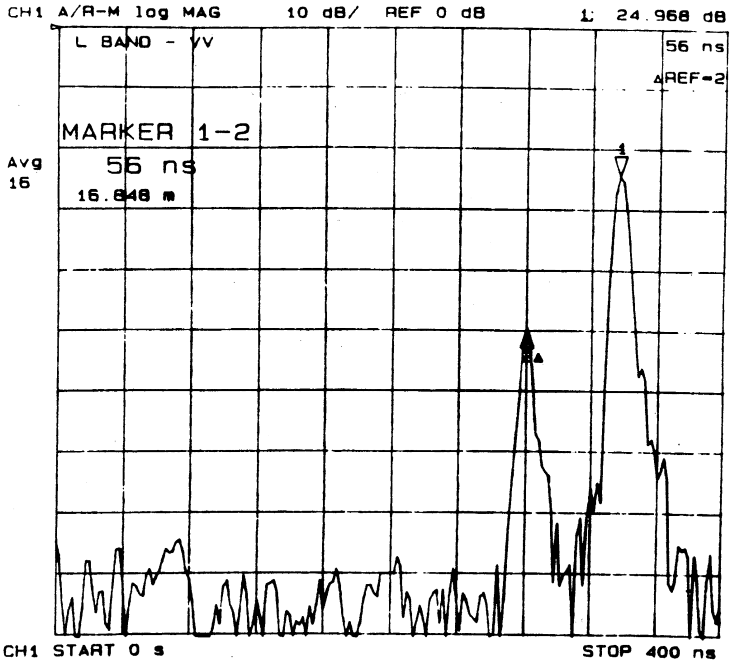

Figure 6: Time domain response of an L-band PARC.

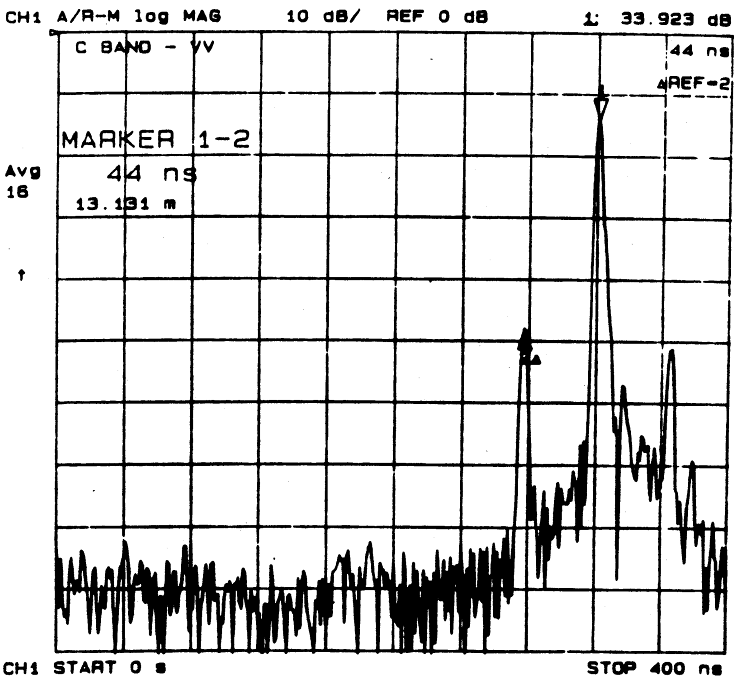

Figure 7: Time domain response of a C-band PARC.

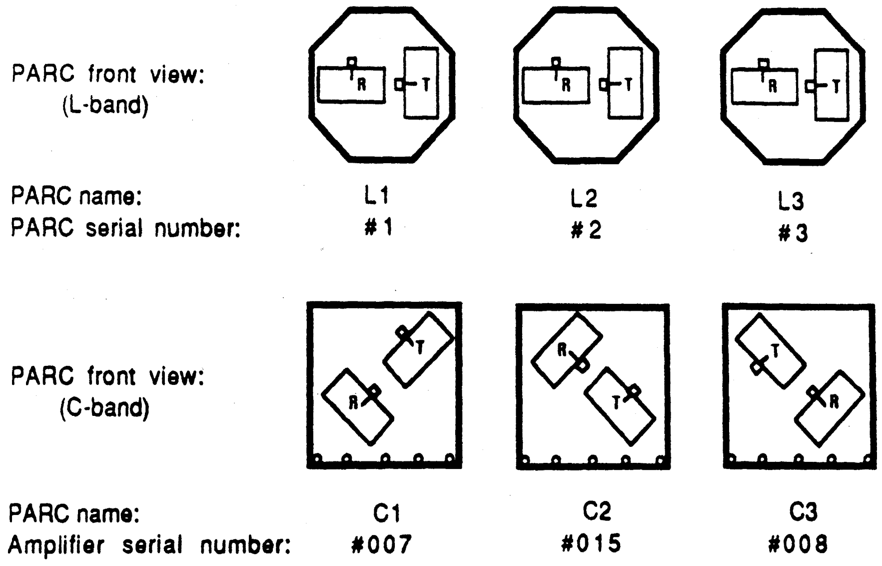

Figure 8: The front panel of L- and C-band PARC's as seen by a radar.

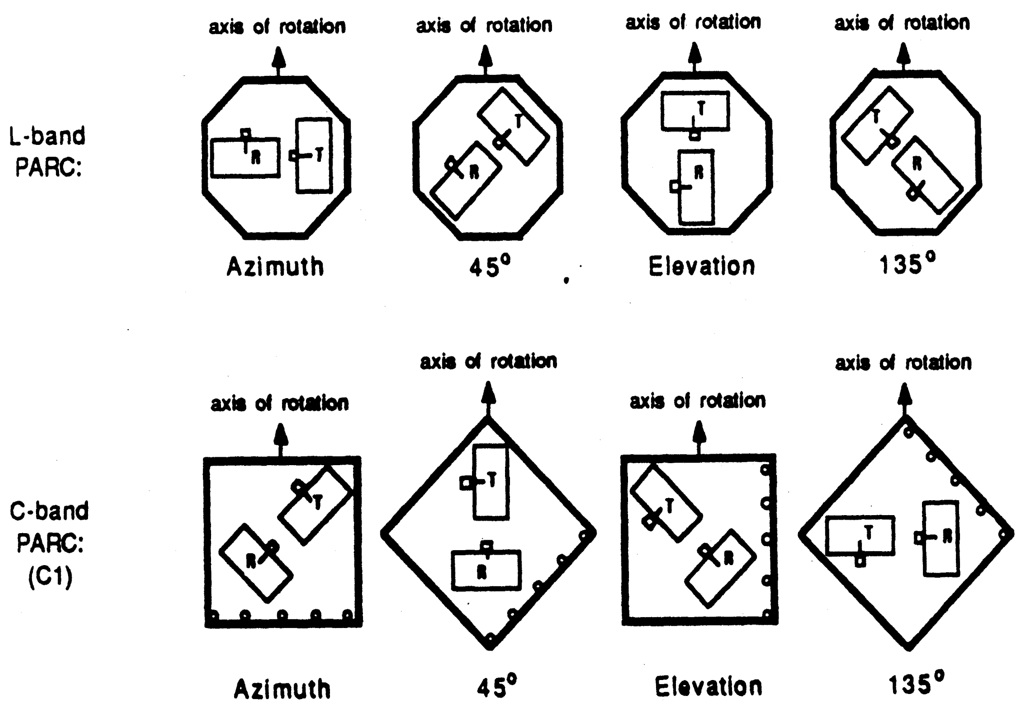

Figure 9: The orientation of L- and C-band PARC's for pattern measurement.

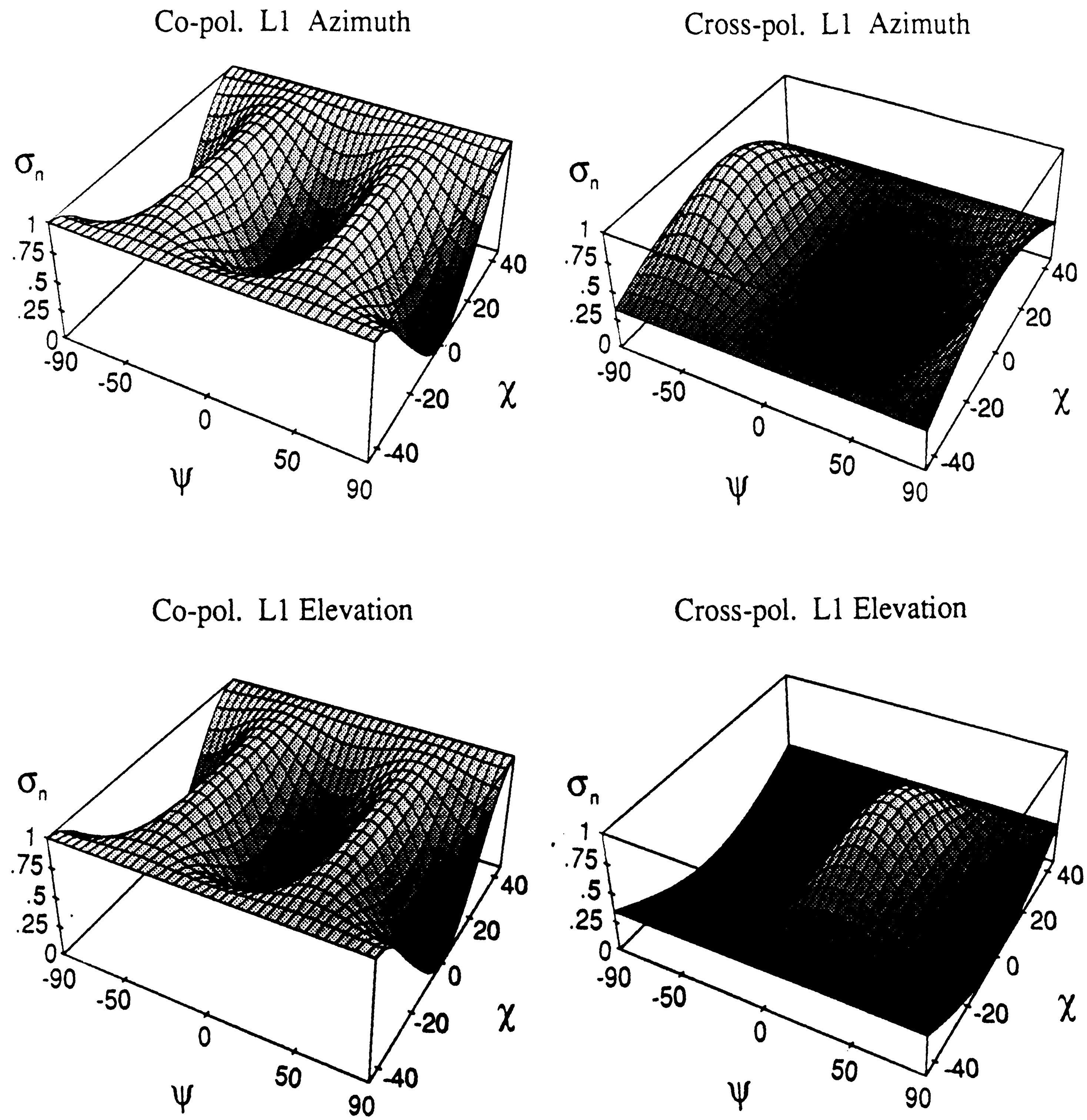

Figure 10: The azimuth and elevation polarization signature for L1 PARC.

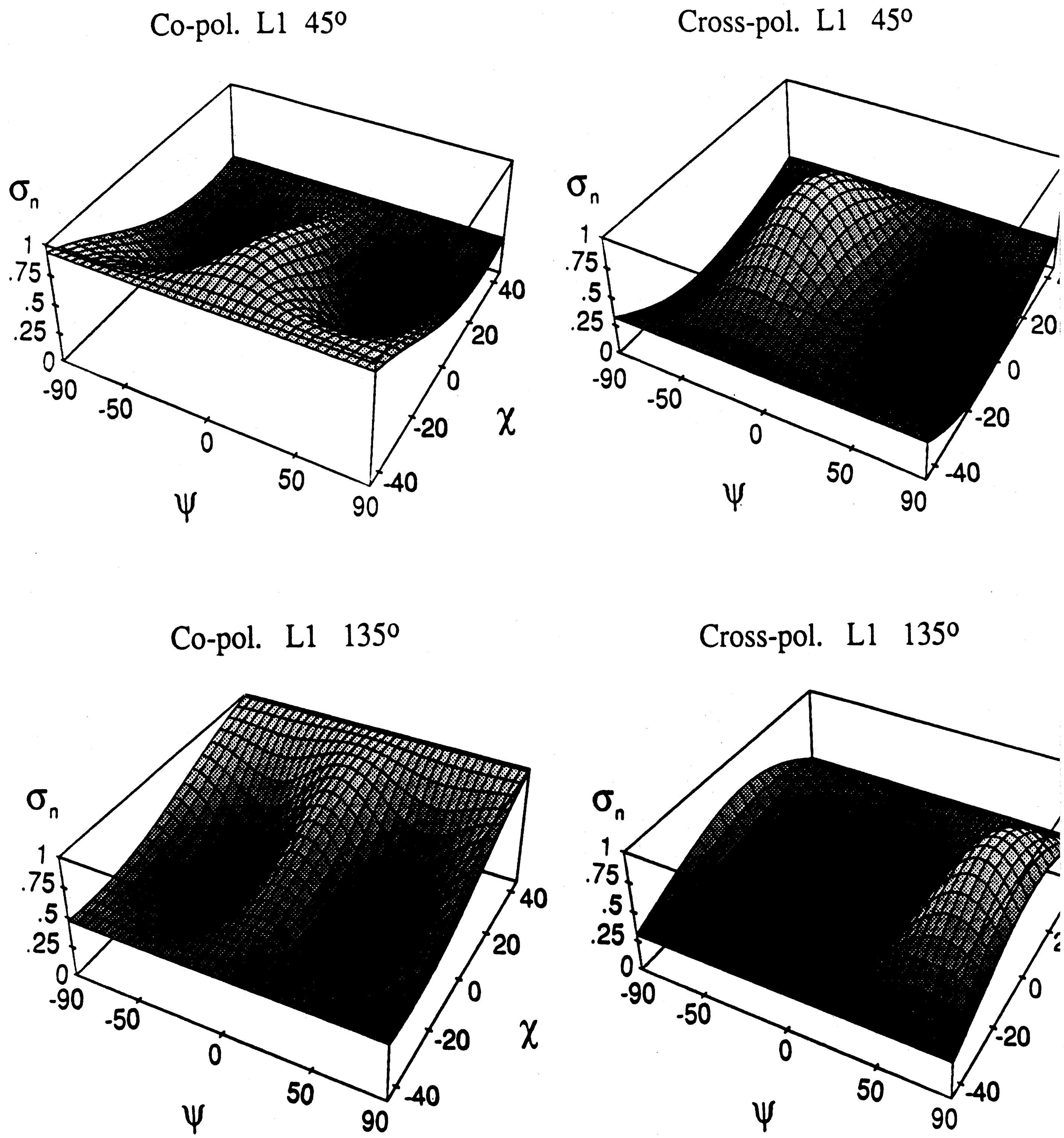

Figure 11: The 45° and 135° polarization signature for L1 PARC.

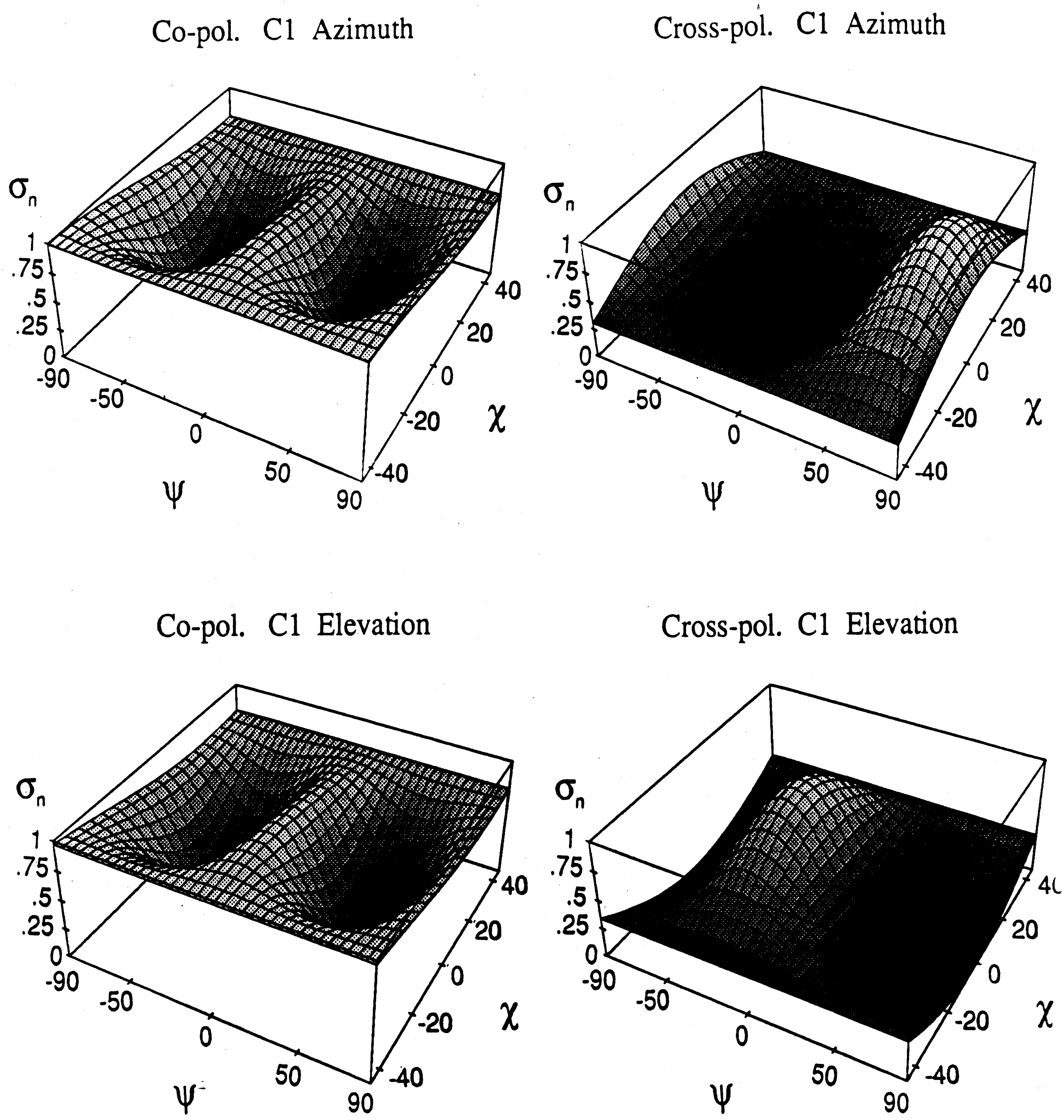

Figure 12: The azimuth and elevation polarization signature for C1 PARC.

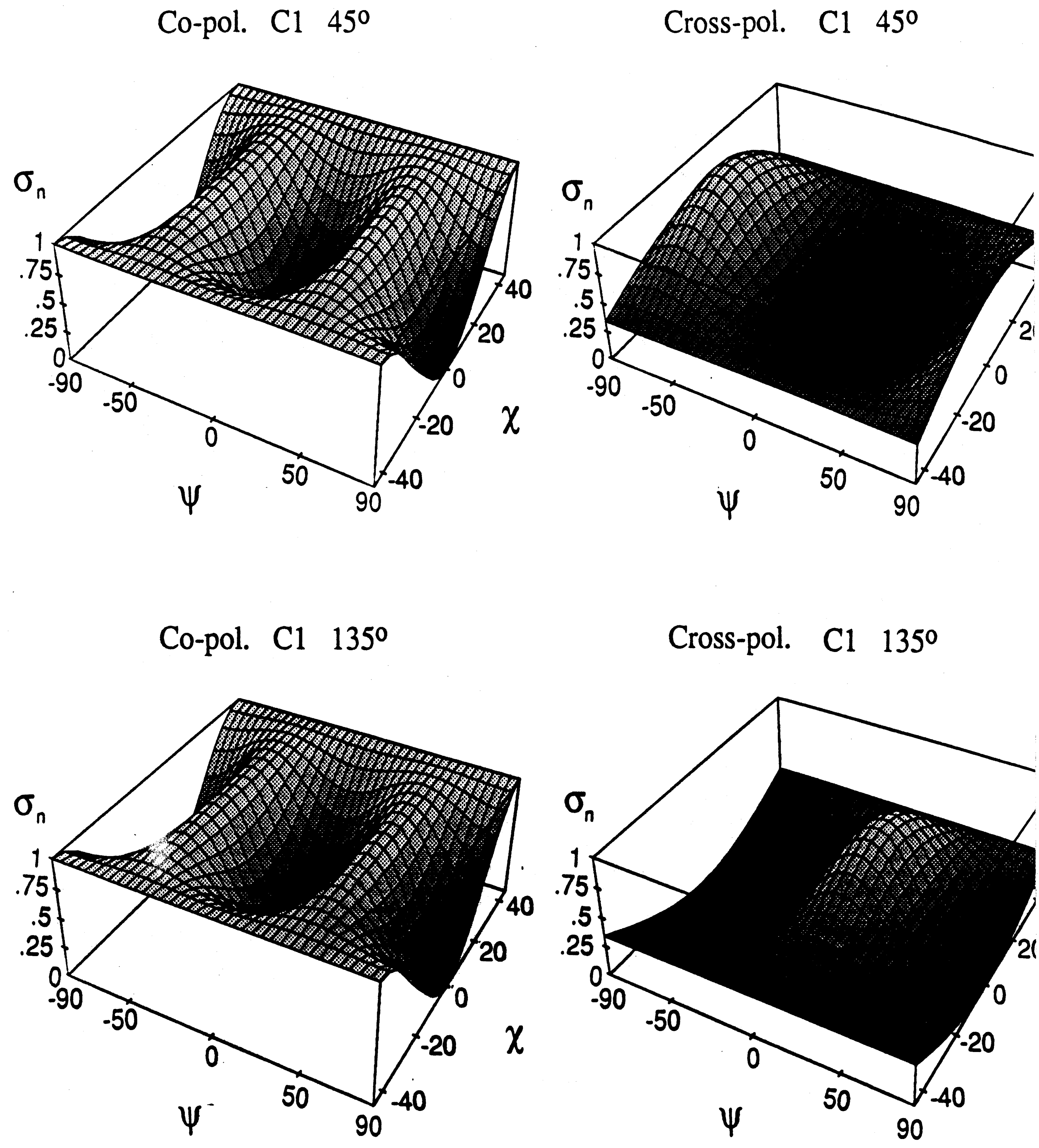

Figure 13: The 45° and 135° polarization signature for C1 PARC.

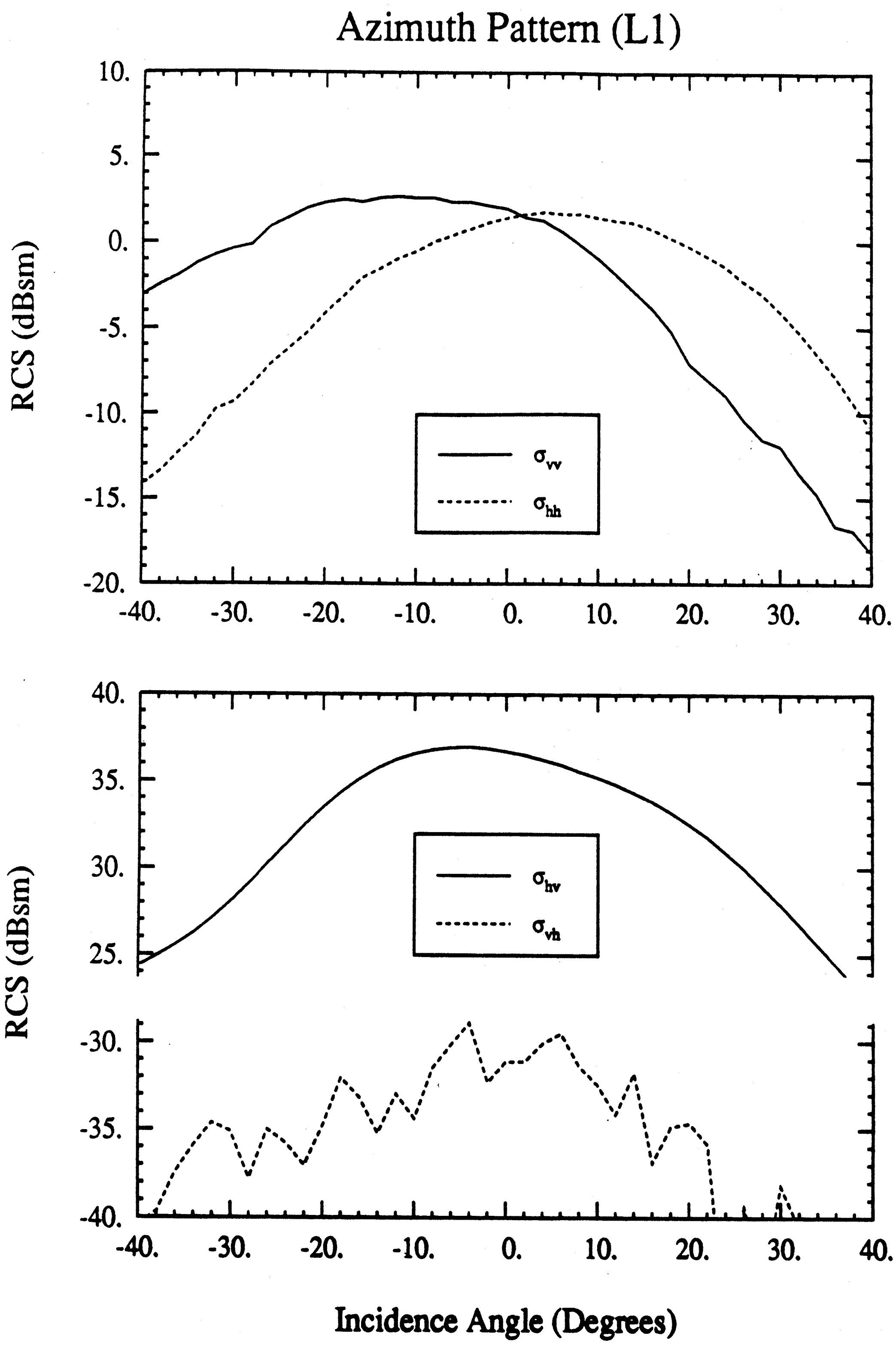

Figure 14: Azimuth pattern of amplitude of scattering matrix elements for L1 PARC.

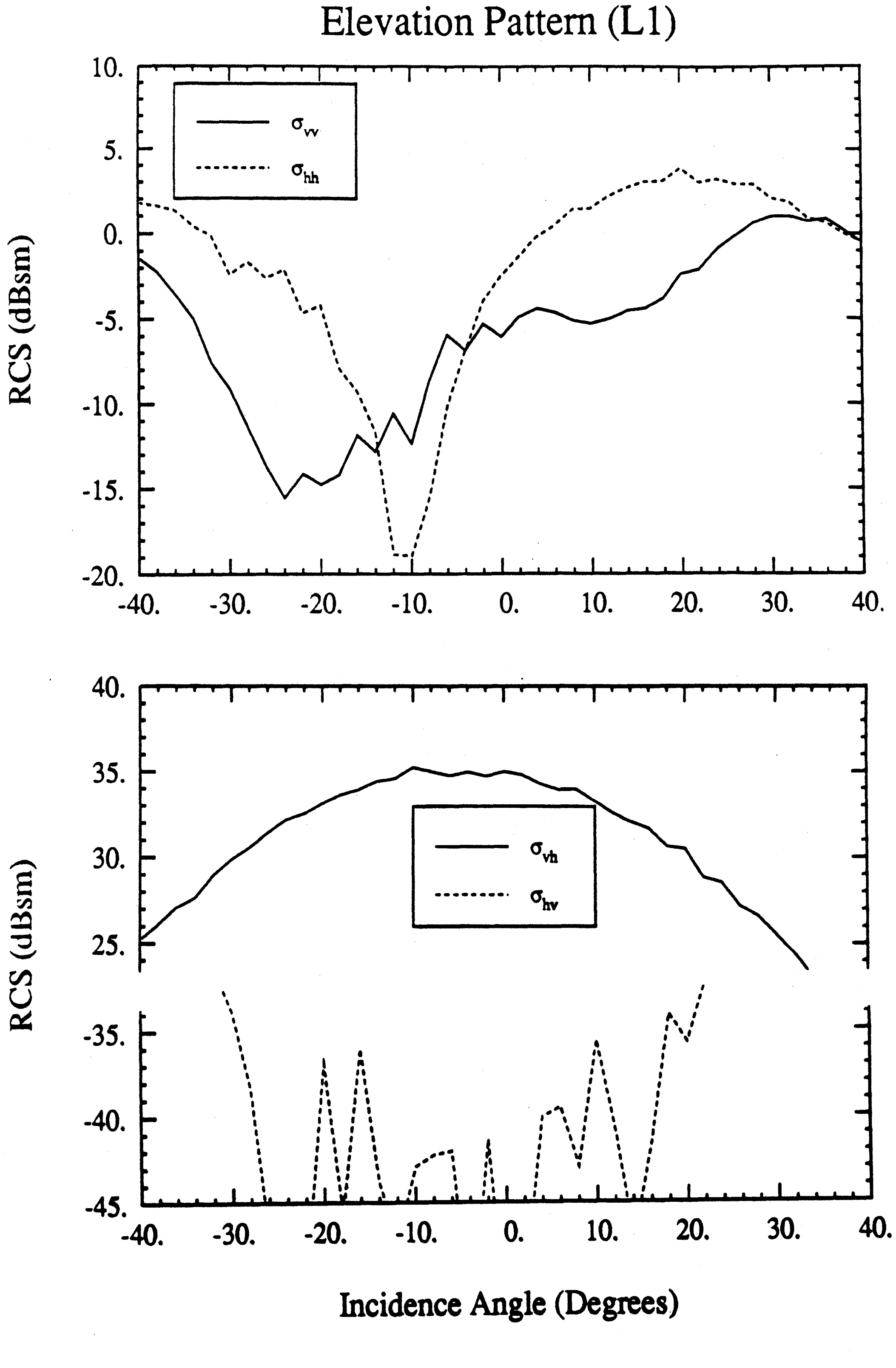

Figure 15: Elevation pattern of amplitude of scattering matrix elements for L1 PARC.

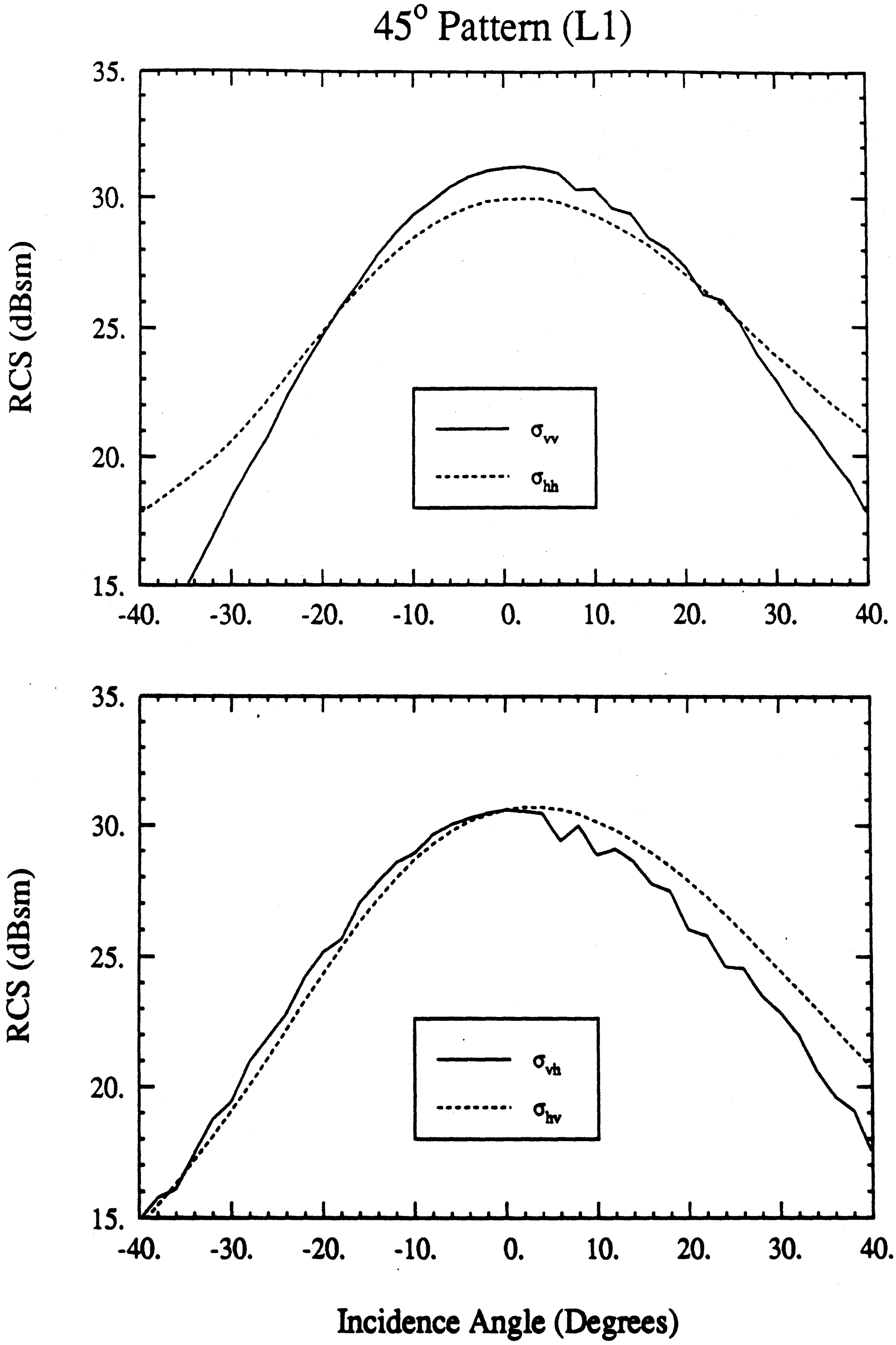

Figure 16: 45° pattern of amplitude of scattering matrix elements for L1 PARC.

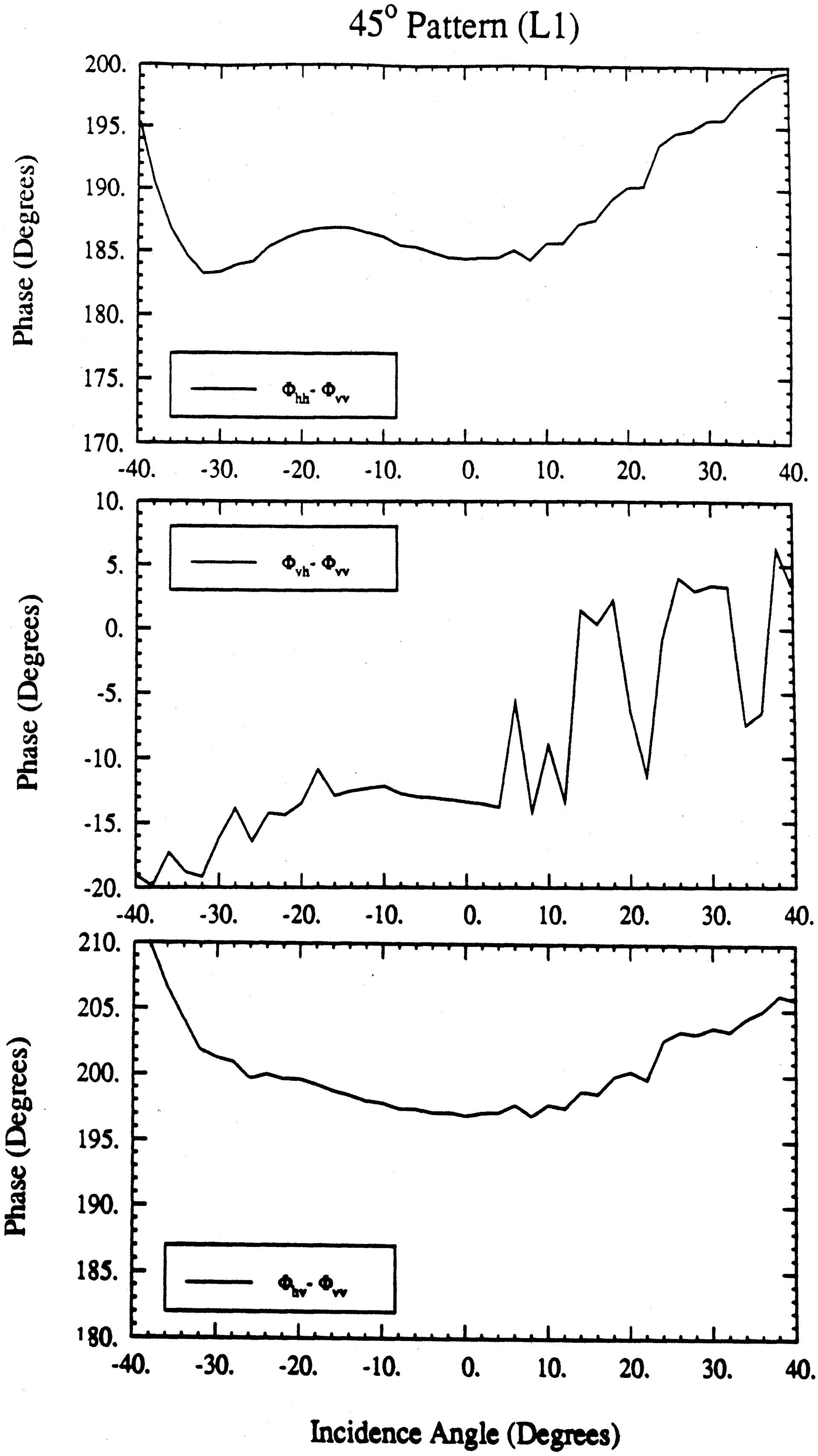

Figure 17: 45° pattern of phase of scattering matrix elements for L1 PARC.

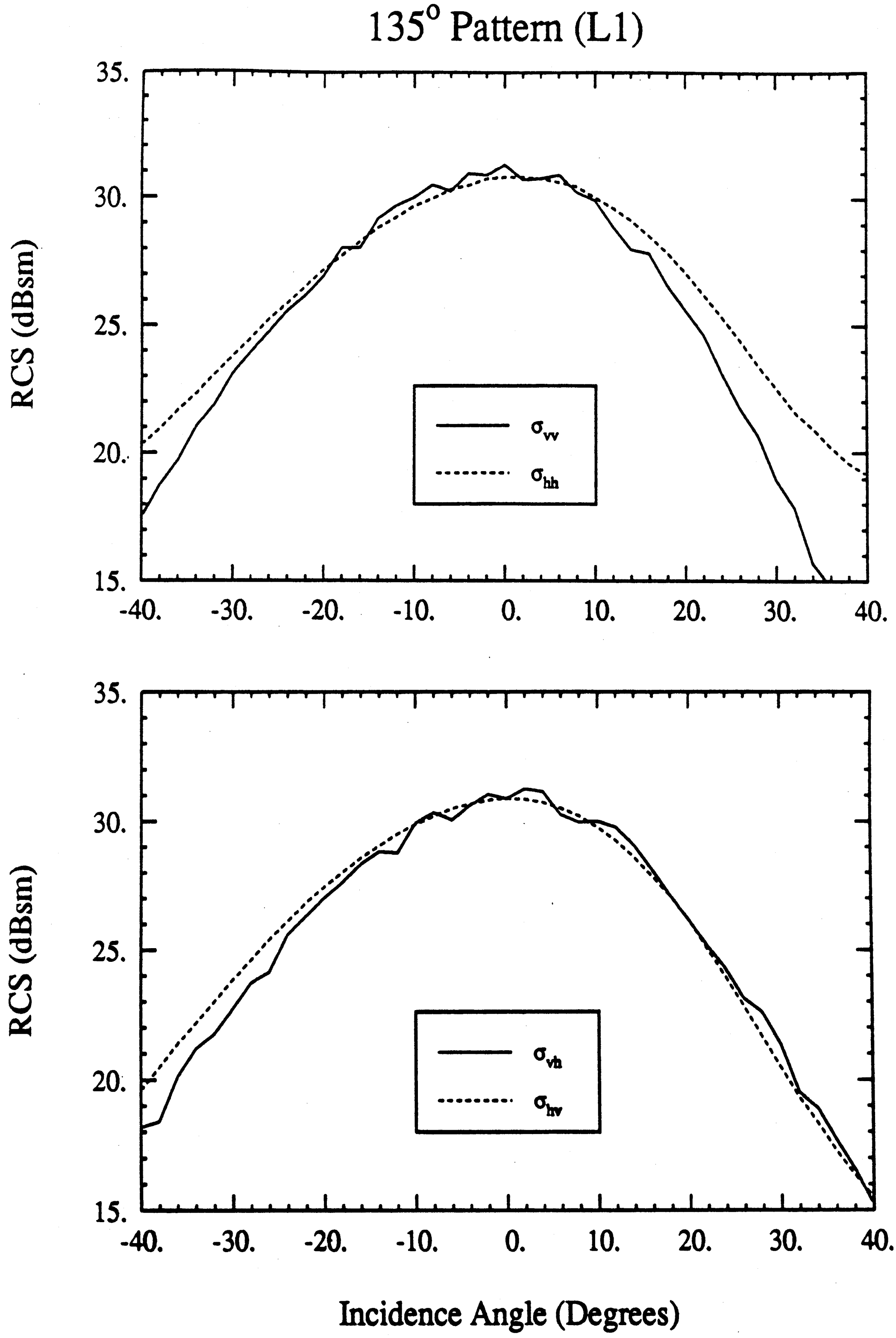

Figure 18: 135° pattern of amplitude of scattering matrix elements for L1 PARC.

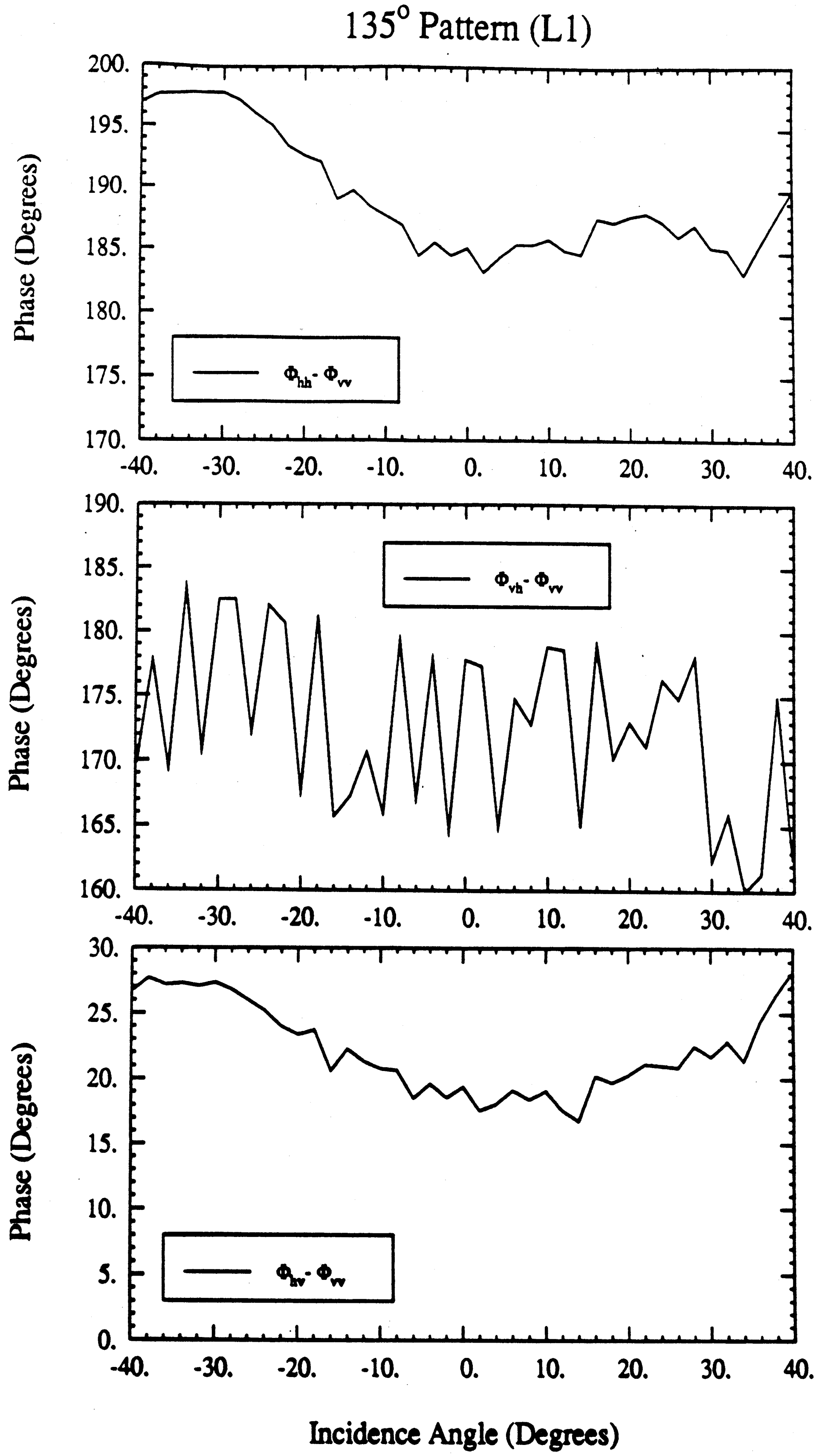

Figure 19: 135° pattern of phase of scattering matrix elements for L1 PARC.

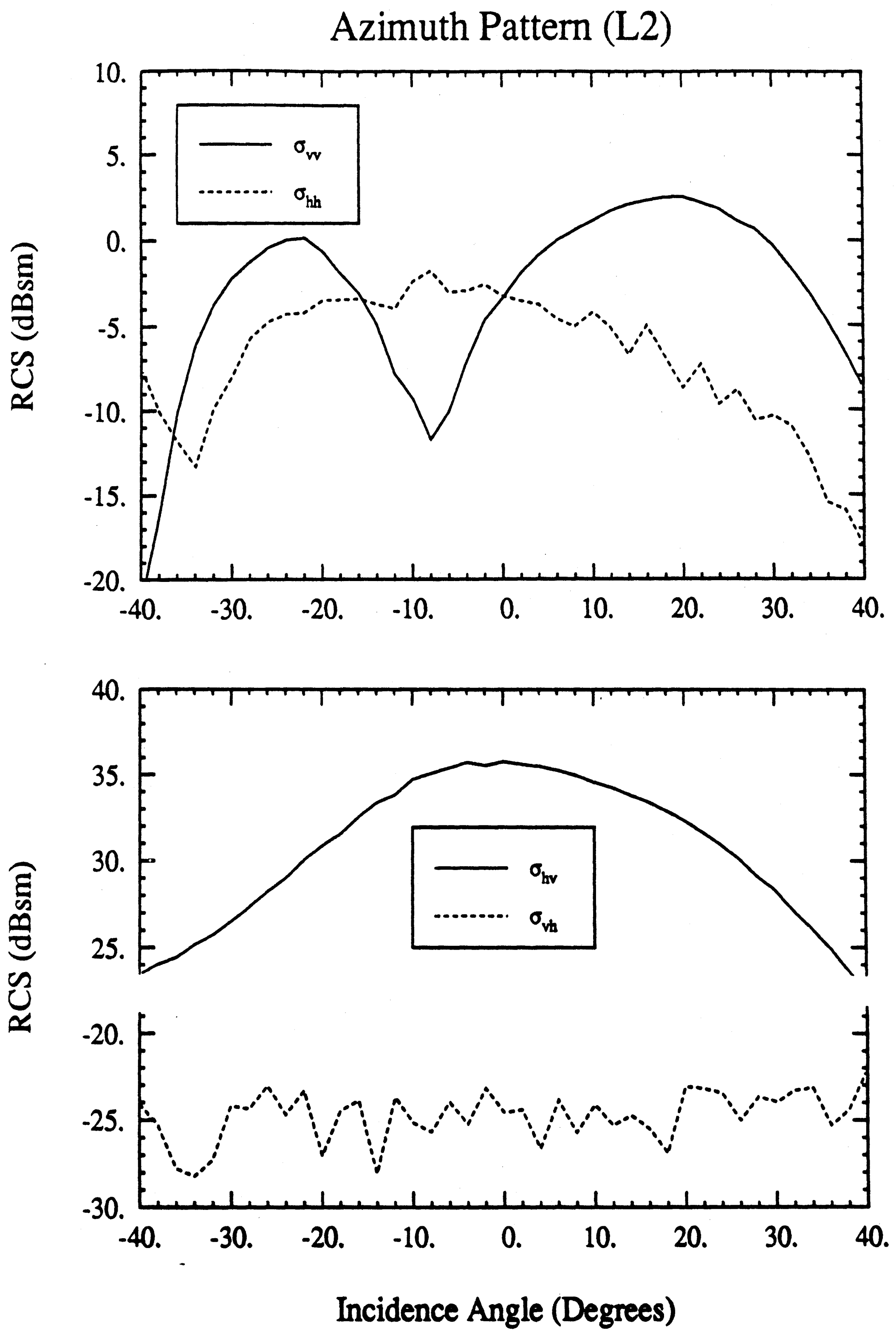

Figure 20: Azimuth pattern of amplitude of scattering matrix elements for L2 PARC.

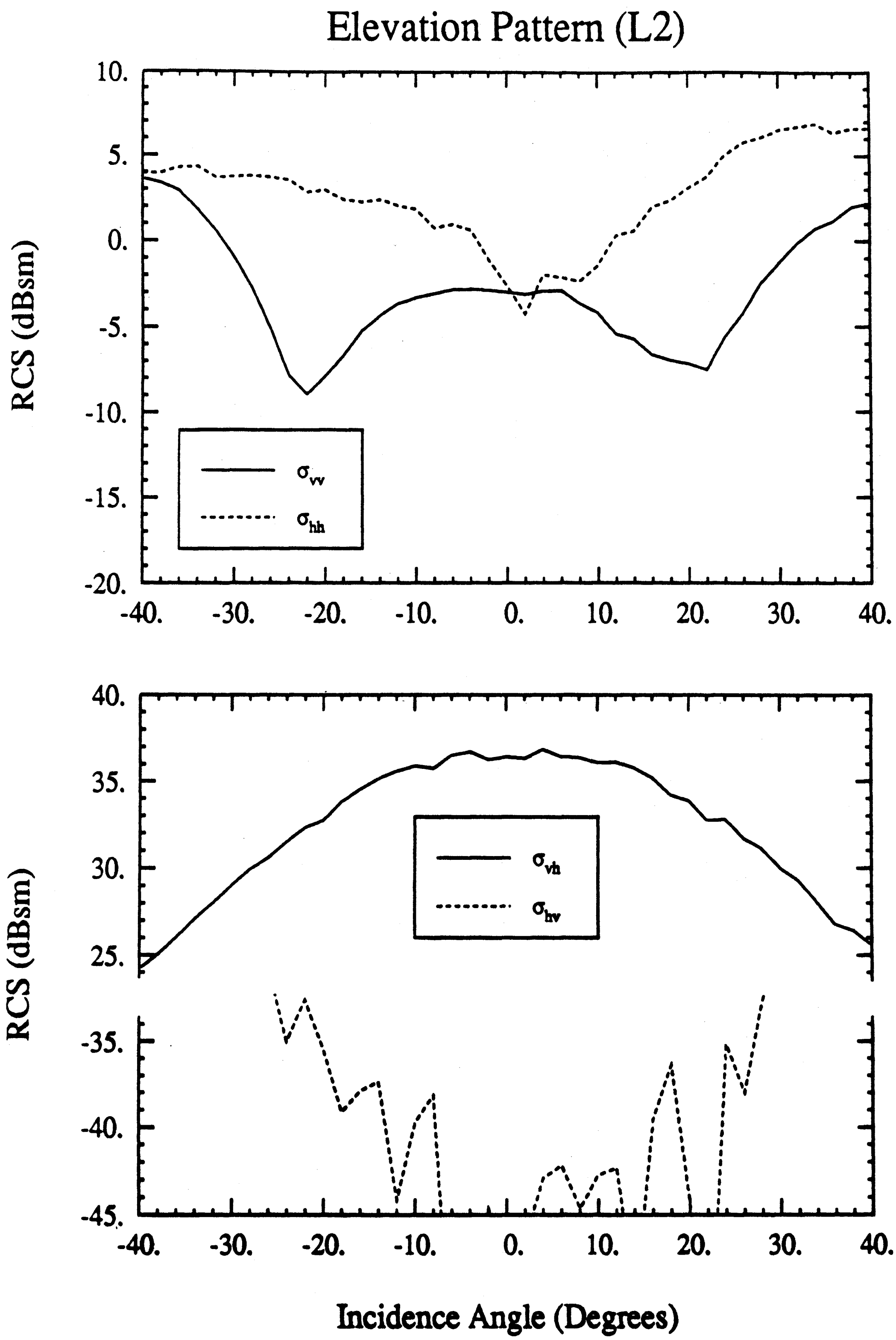

Figure 21: Elevation pattern of amplitude of scattering matrix elements for L2 PARC.

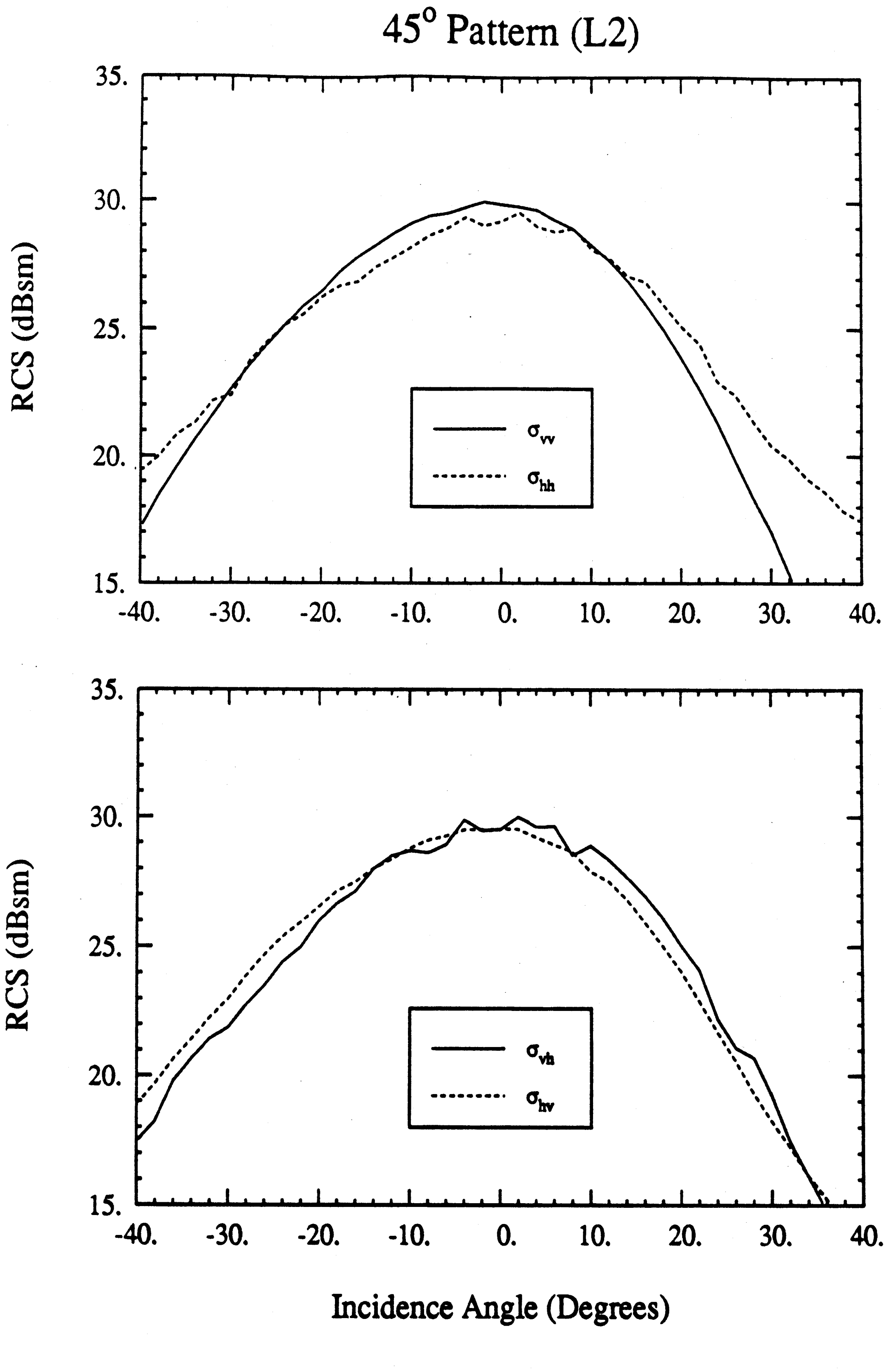

Figure 22: 45° pattern of amplitude of scattering matrix elements for L2 PARC.

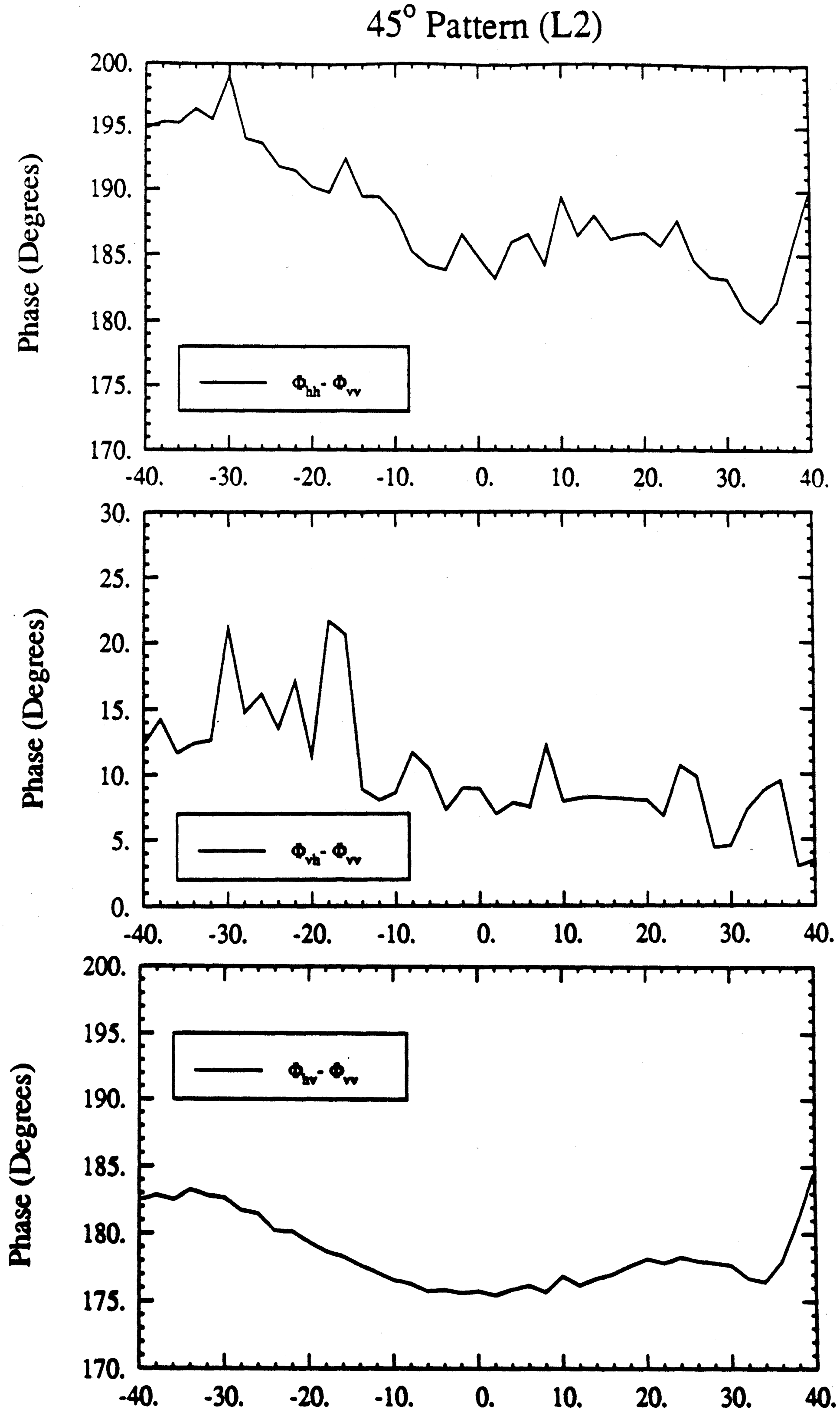

Figure 23: 45° pattern of phase of scattering matrix elements for L2 PARC.

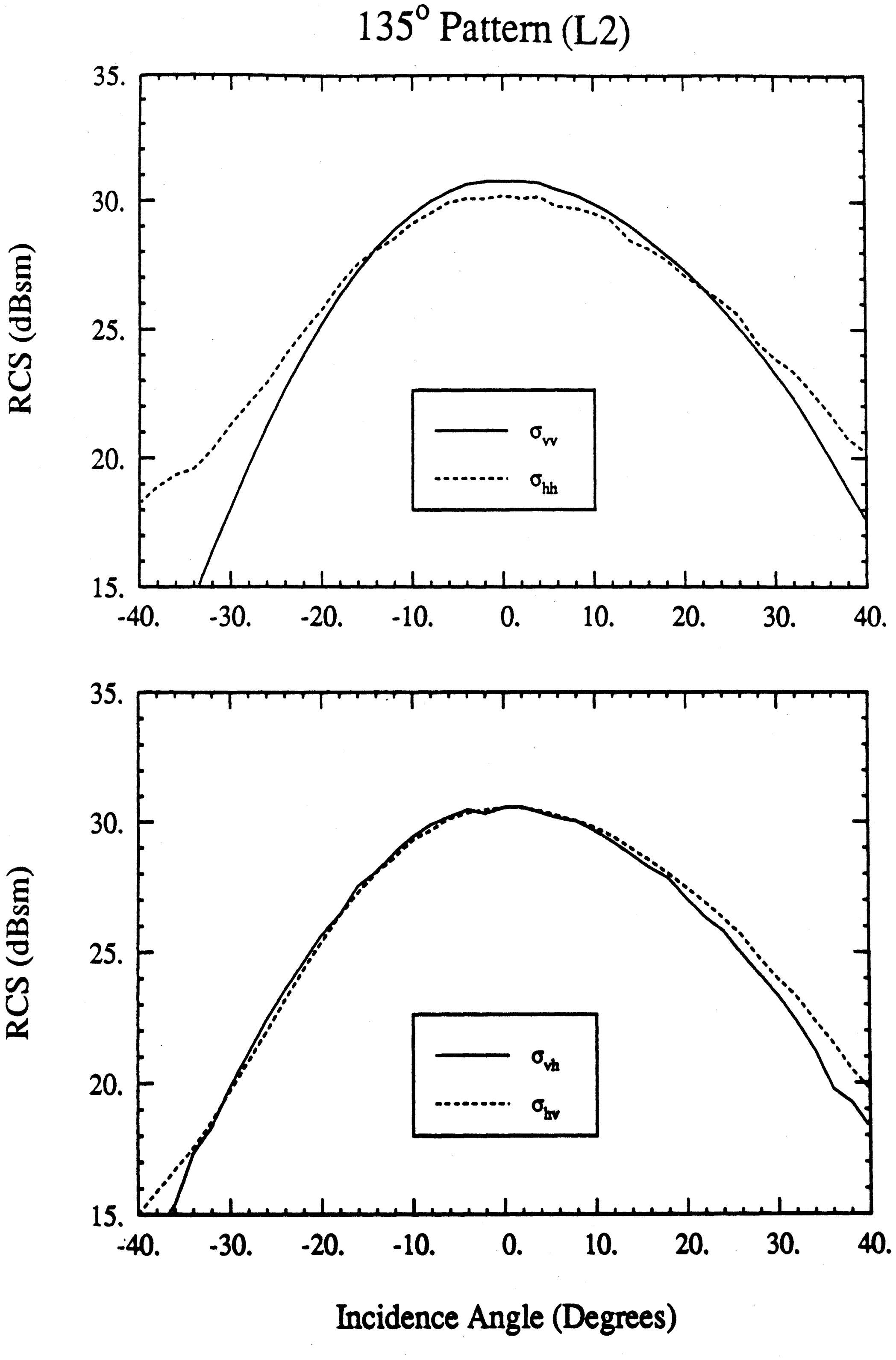

Figure 24: 135° pattern of amplitude of scattering matrix elements for L2 PARC.

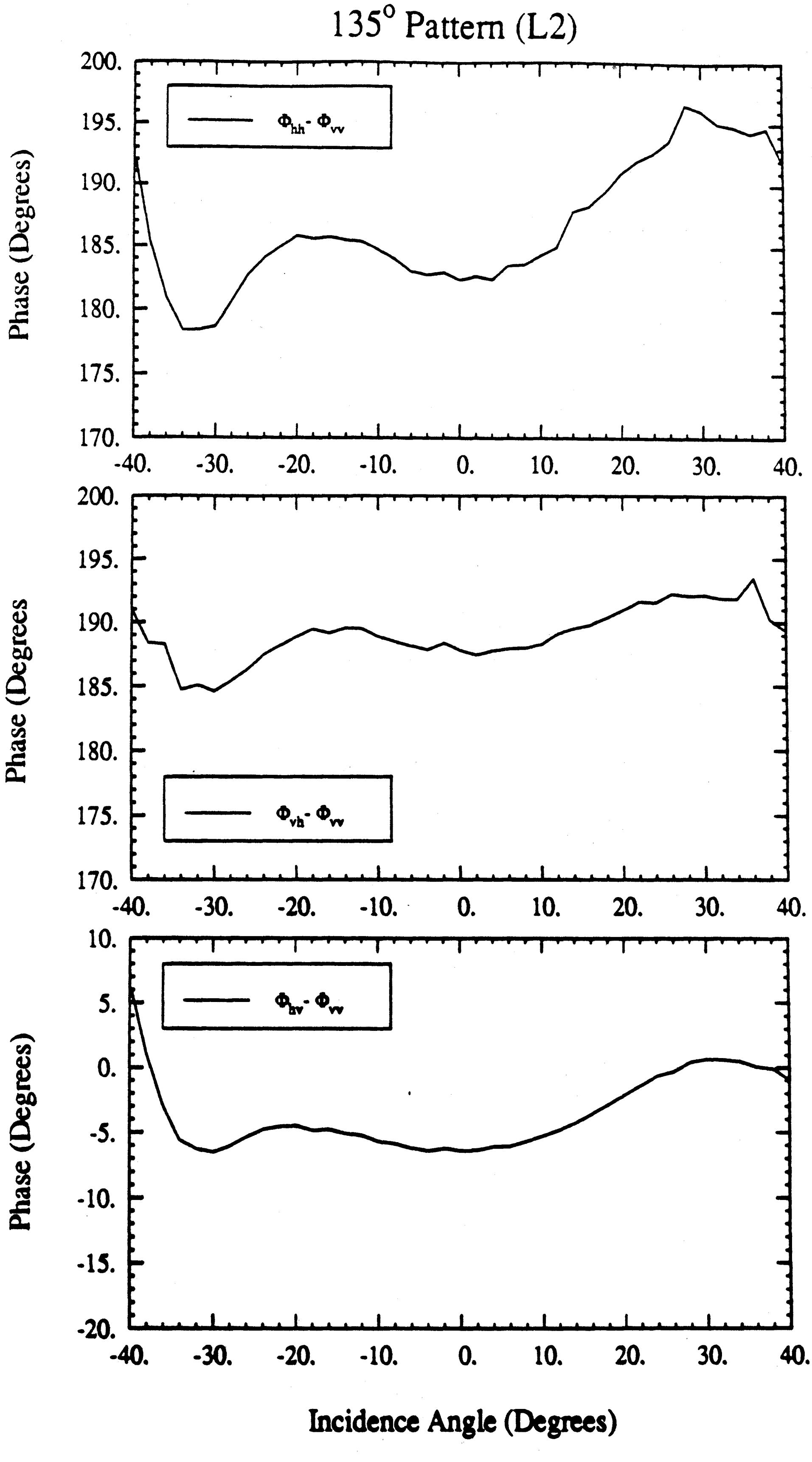

Figure 25: 135° pattern of phase of scattering matrix elements for L2 PARC.

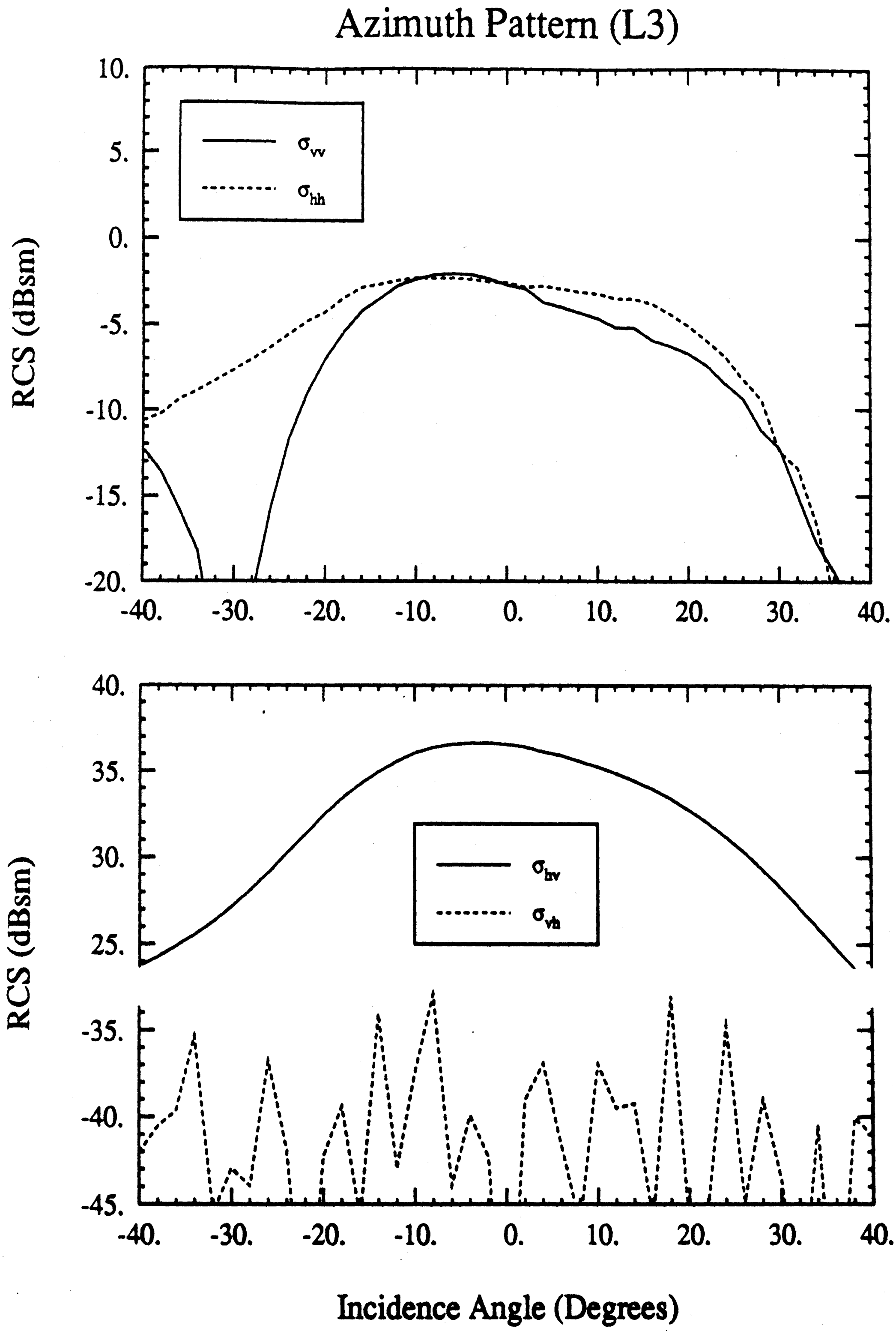

Figure 26: Azimuth pattern of amplitude of scattering matrix elements for L3 PARC.

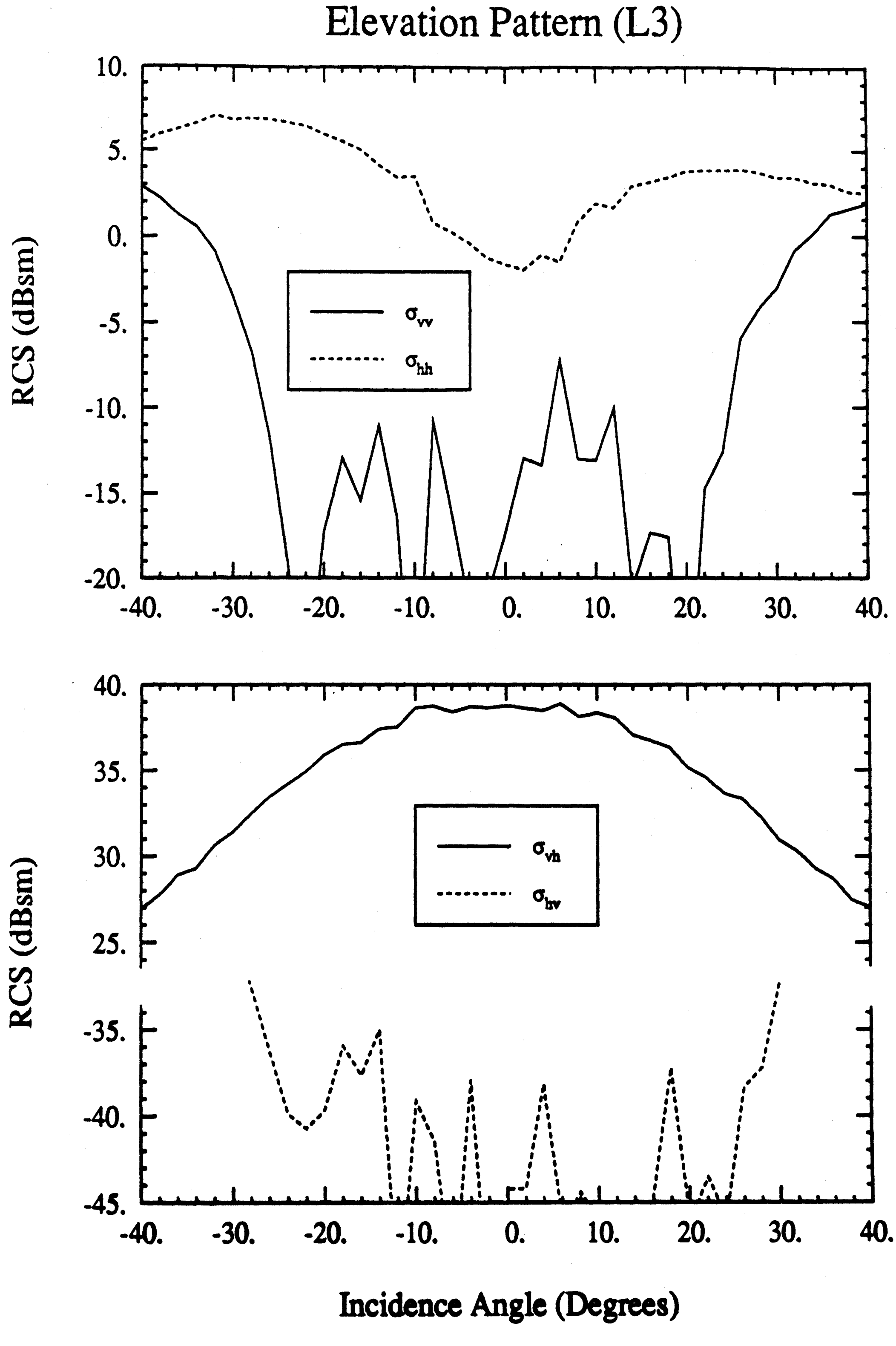

Figure 27: Elevation pattern of amplitude of scattering matrix elements for L3 PARC.

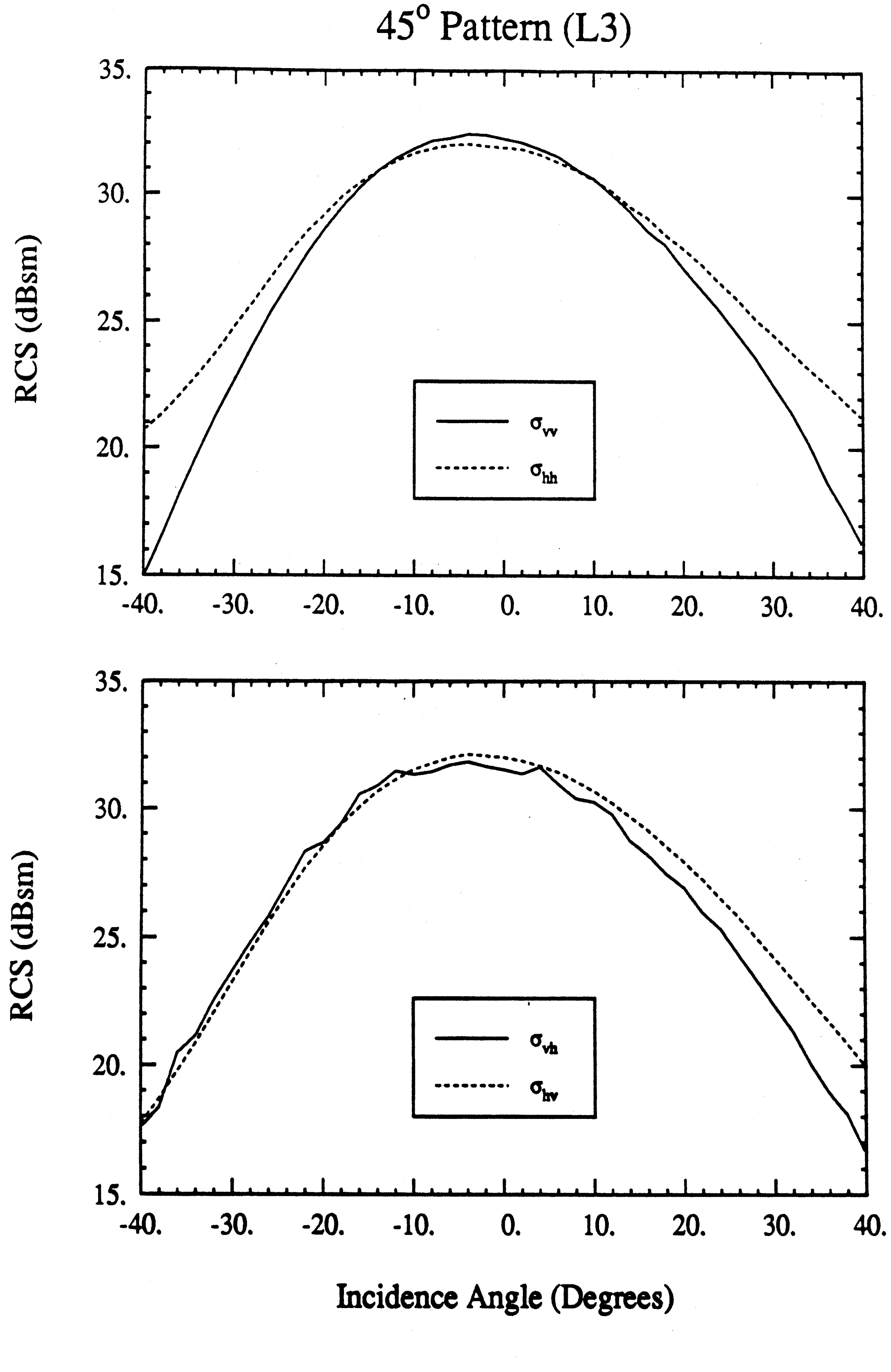

Figure 28: 45° pattern of amplitude of scattering matrix elements for L3 PARC.

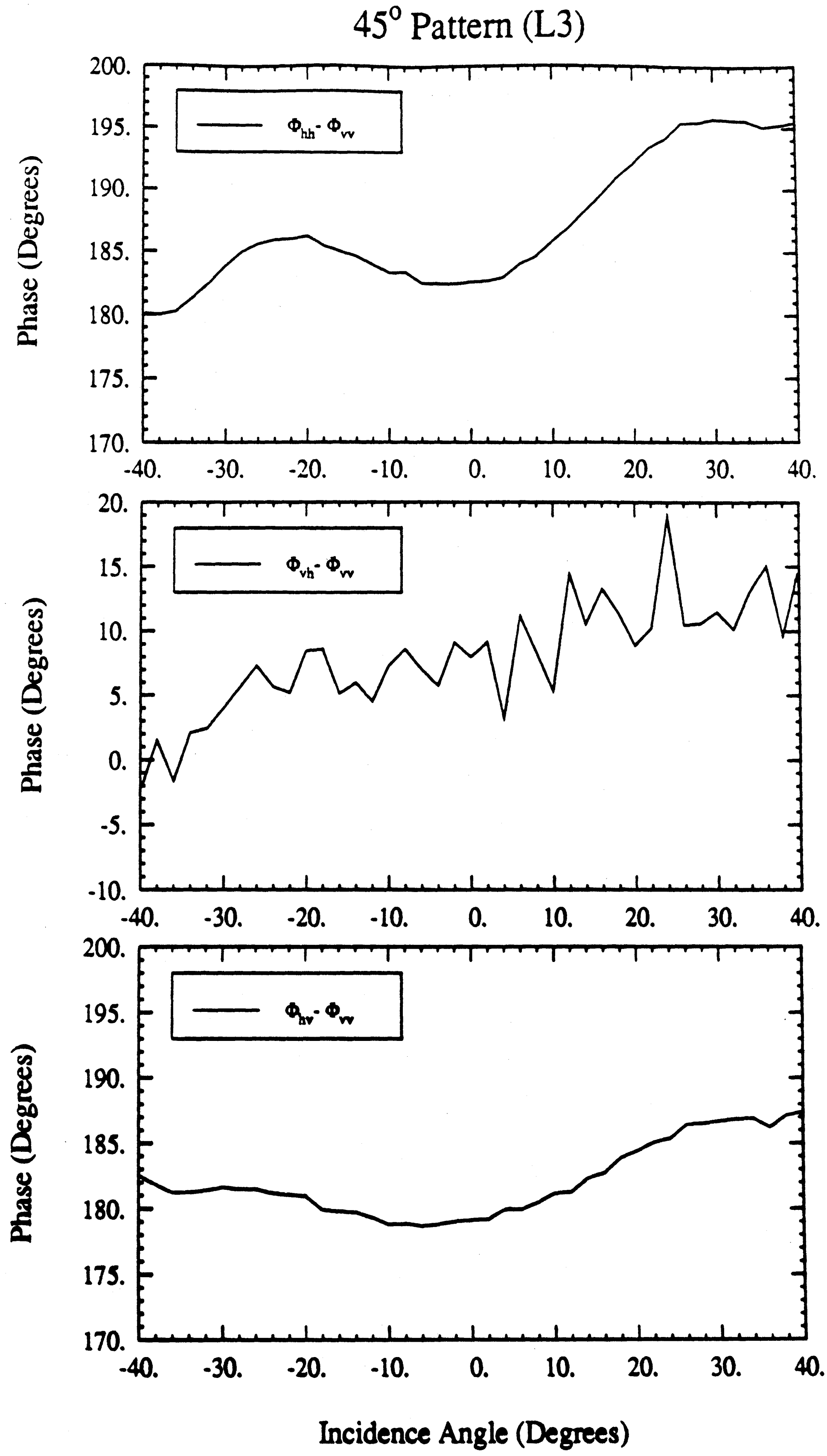

Figure 29: 45° pattern of phase of scattering matrix elements for L3 PARC.

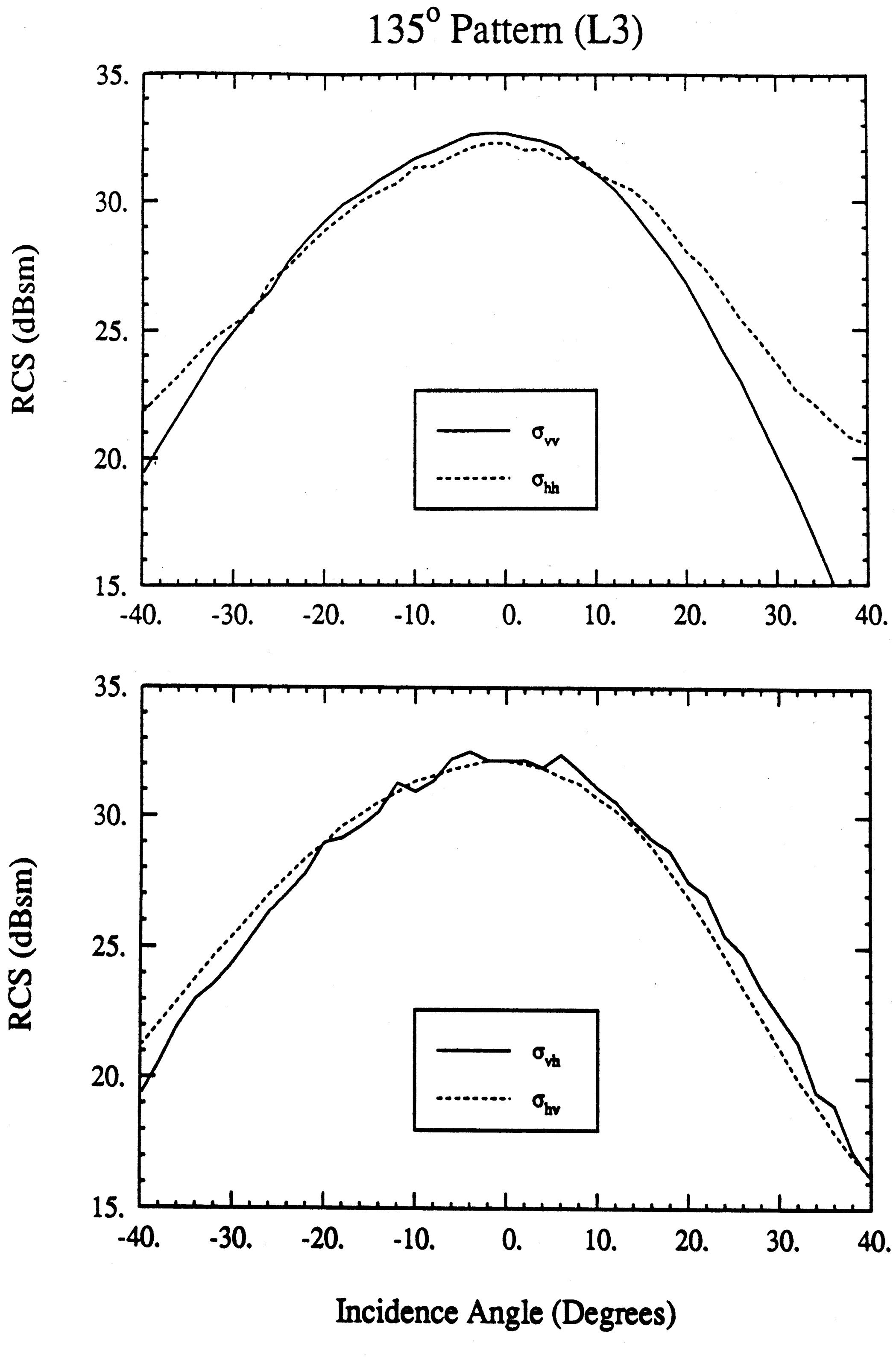

Figure 30: 135° pattern of amplitude of scattering matrix elements for L3 PARC.

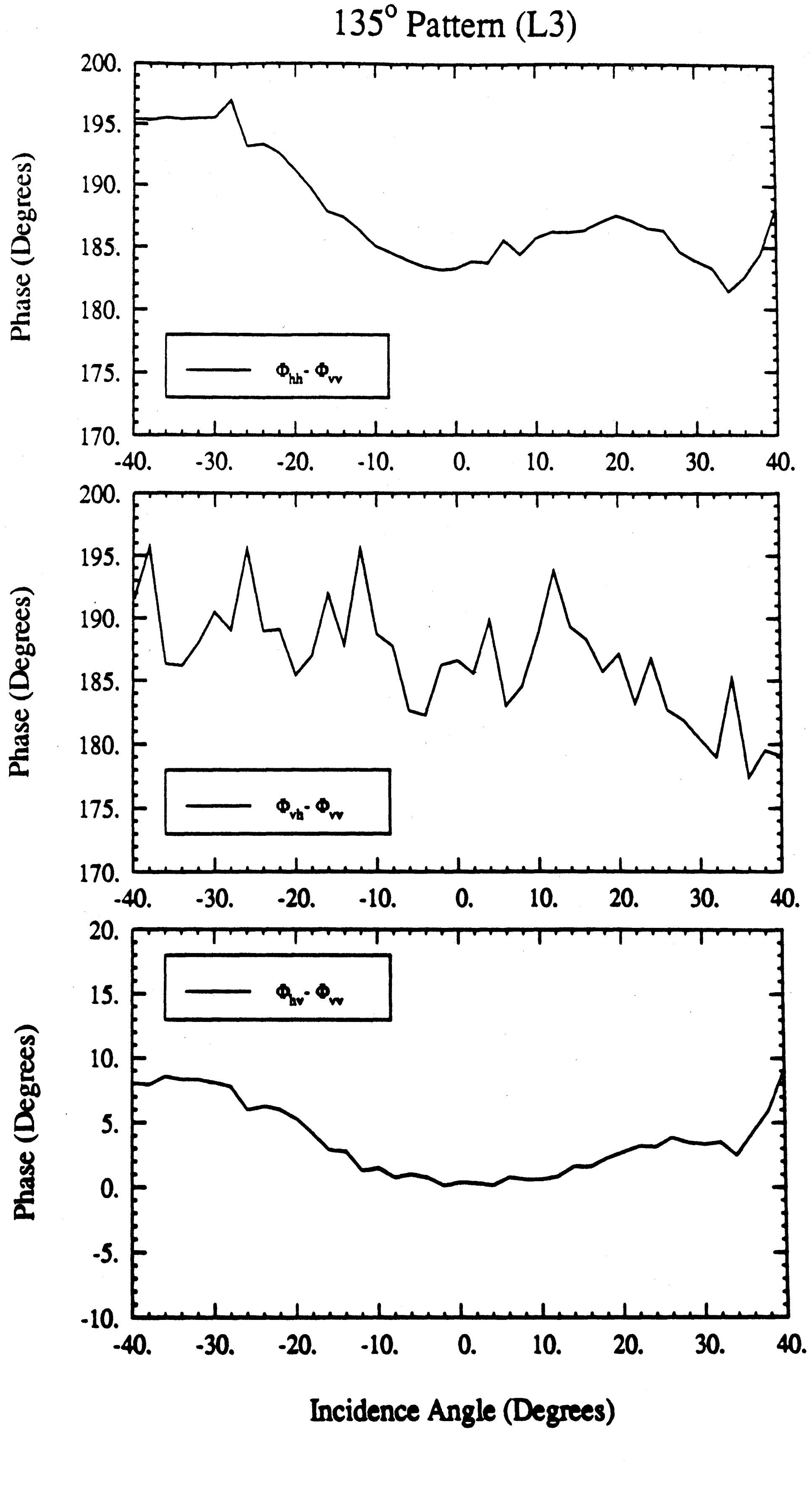

Figure 31: 135° pattern of phase of scattering matrix elements for L3 PARC.

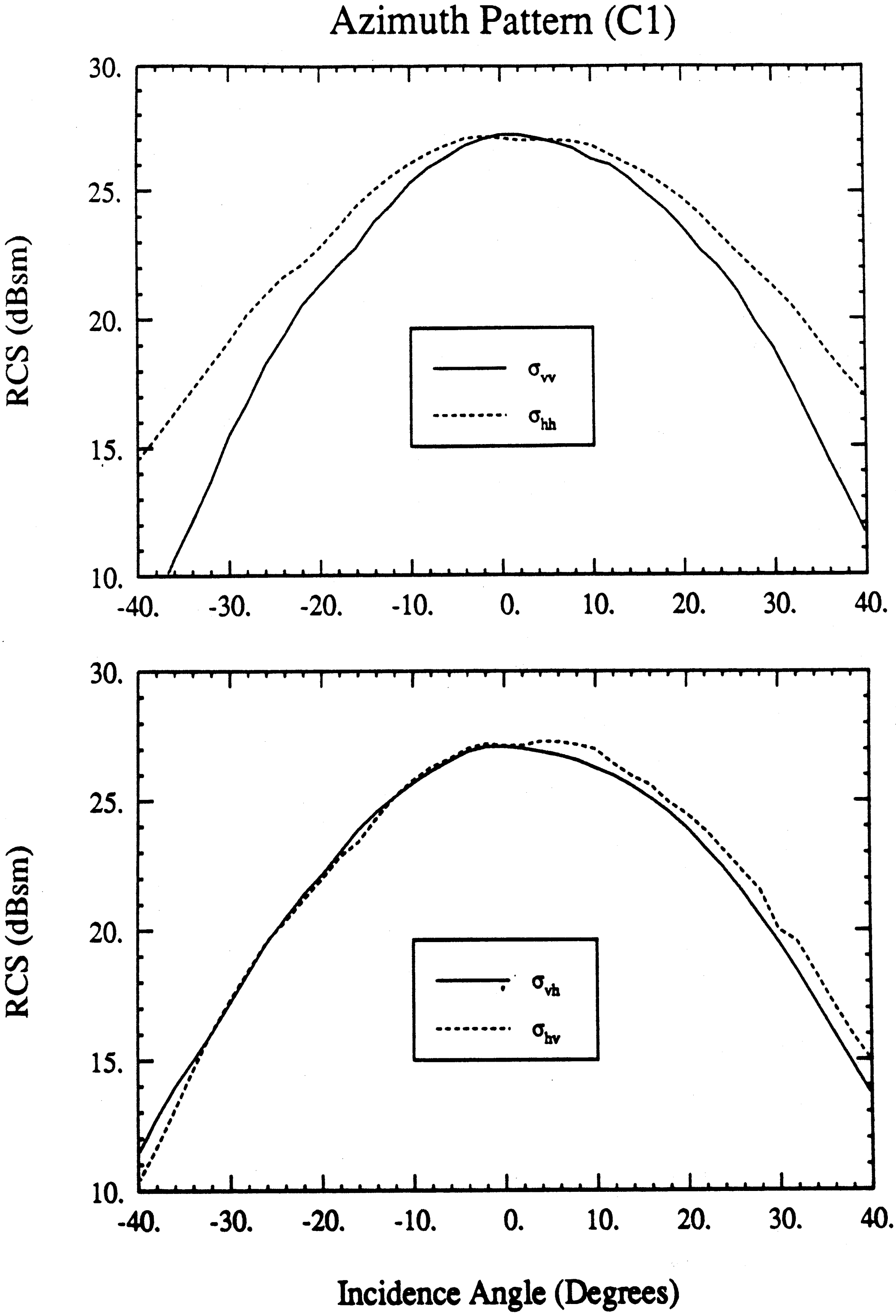

Figure 32: Azimuth pattern of amplitude of scattering matrix elements for C1 PARC.

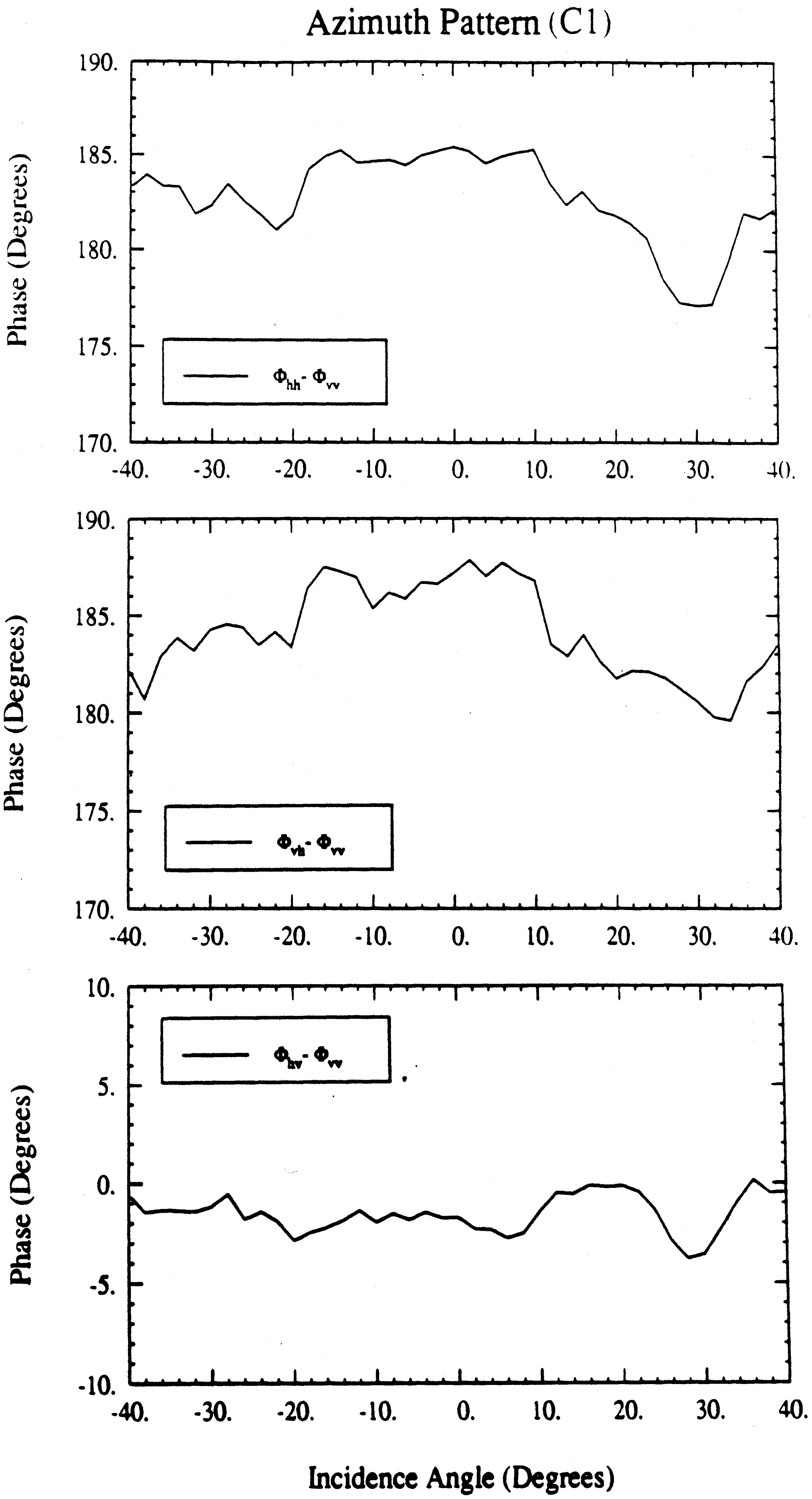

Figure 33: Azimuth pattern of phase of scattering matrix elements for C1 PARC.

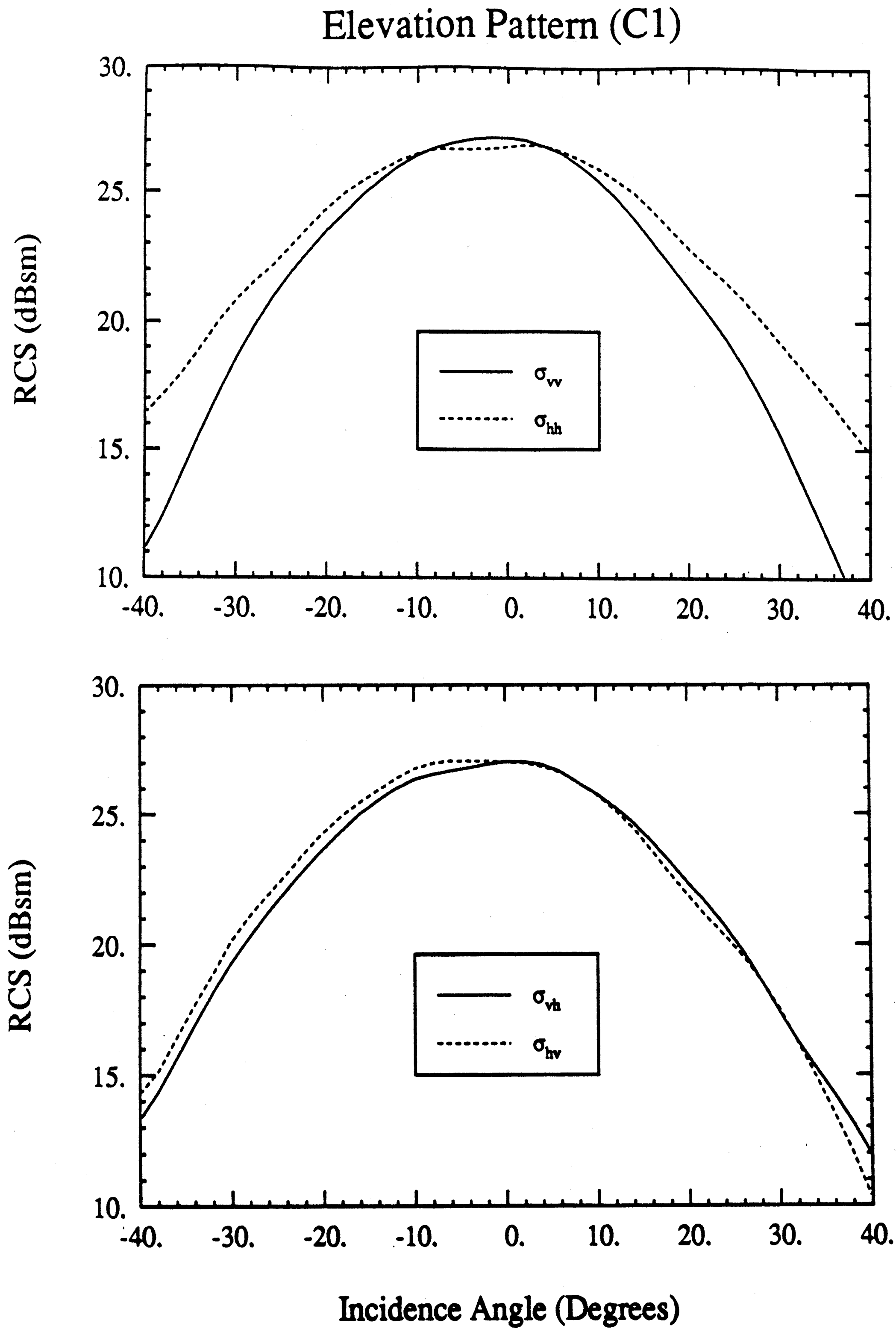

Figure 34: Elevation pattern of amplitude of scattering matrix elements for C1 PARC.

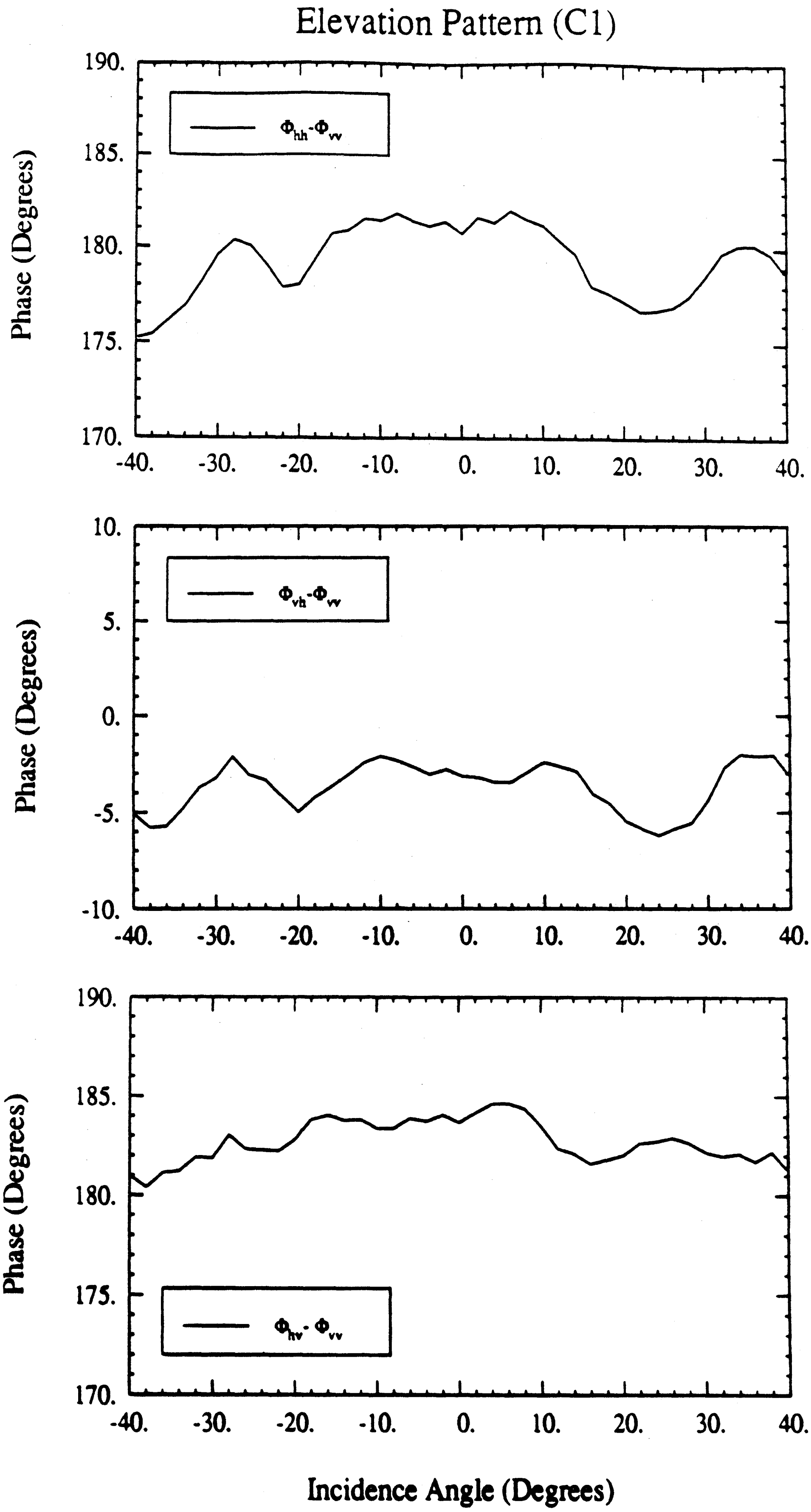

Figure 35: Elevation pattern of phase of scattering matrix elements for C1 PARC.

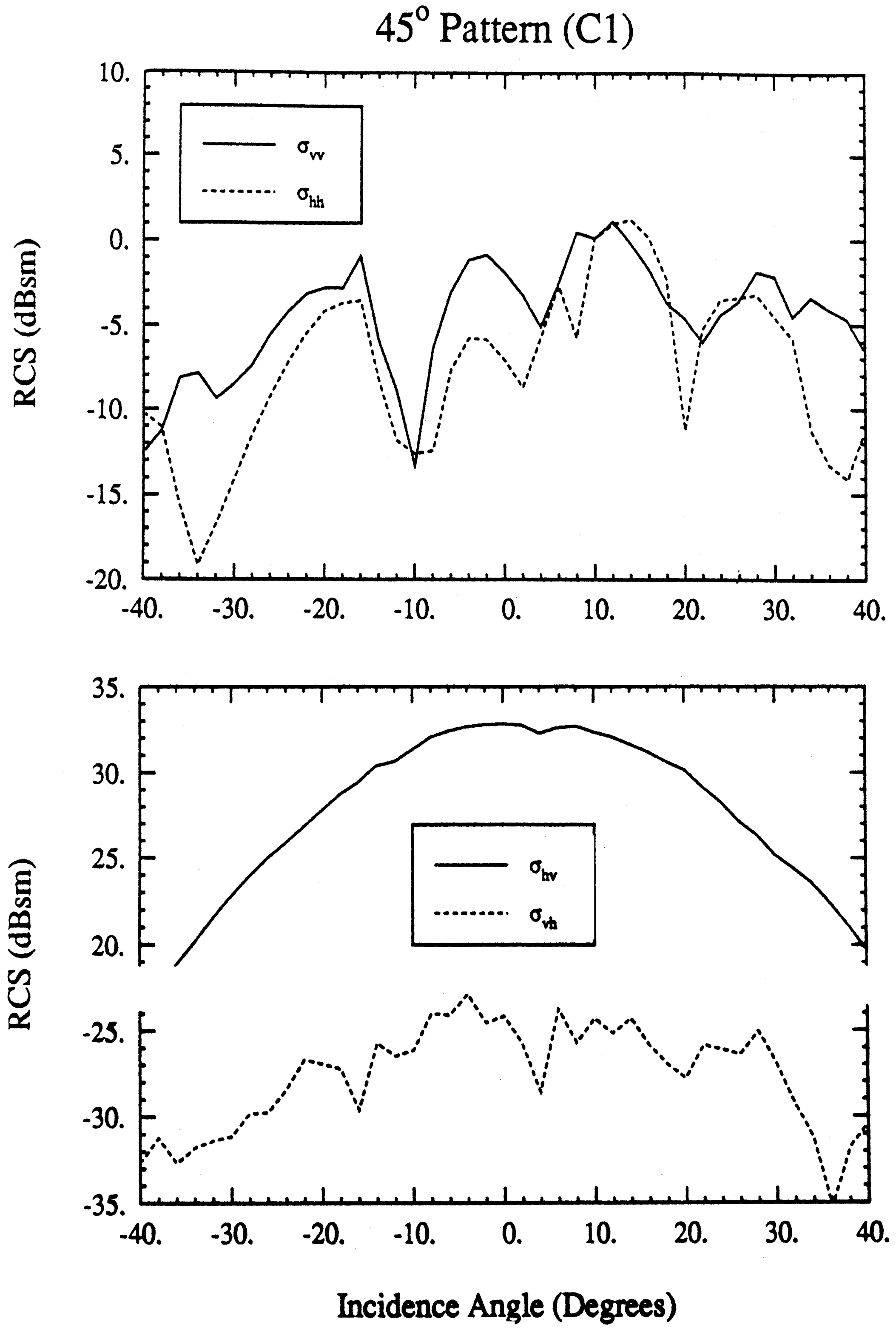

Figure 36: 45° pattern of amplitude of scattering matrix elements for C1 PARC.

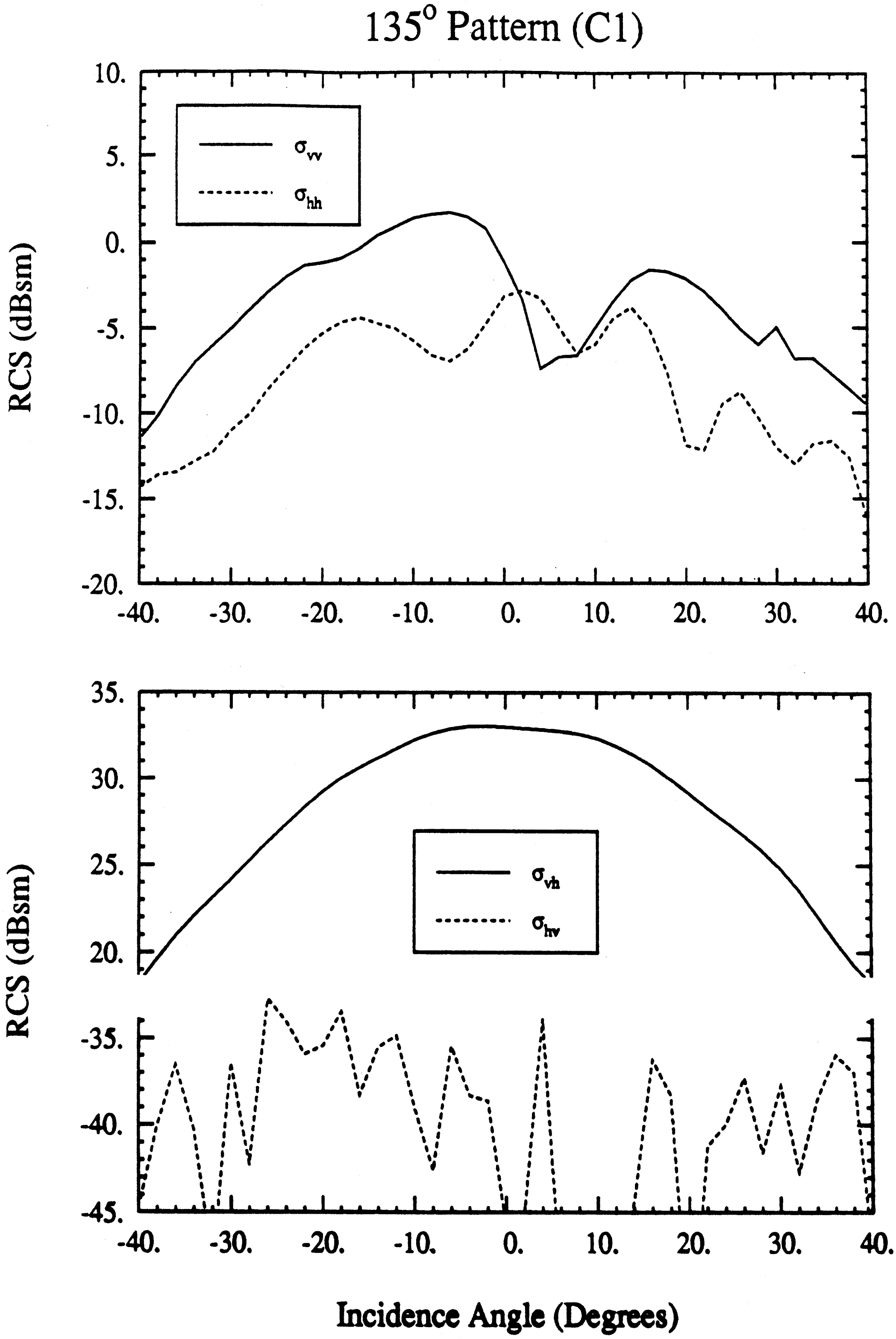

Figure 37: 135° pattern of amplitude of scattering matrix elements for C1 PARC.

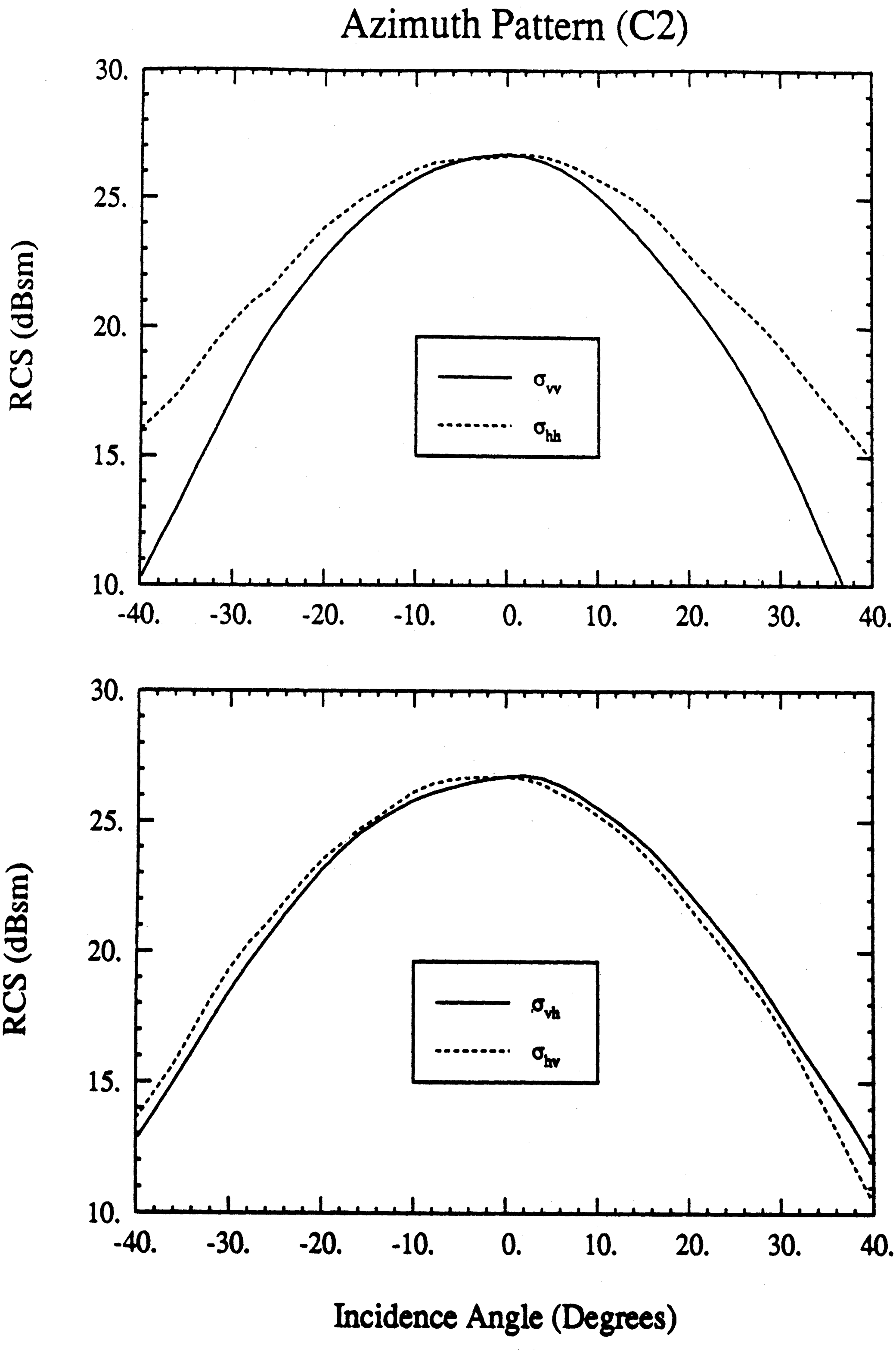

Figure 38: Azimuth pattern of amplitude of scattering matrix elements for C2 PARC.

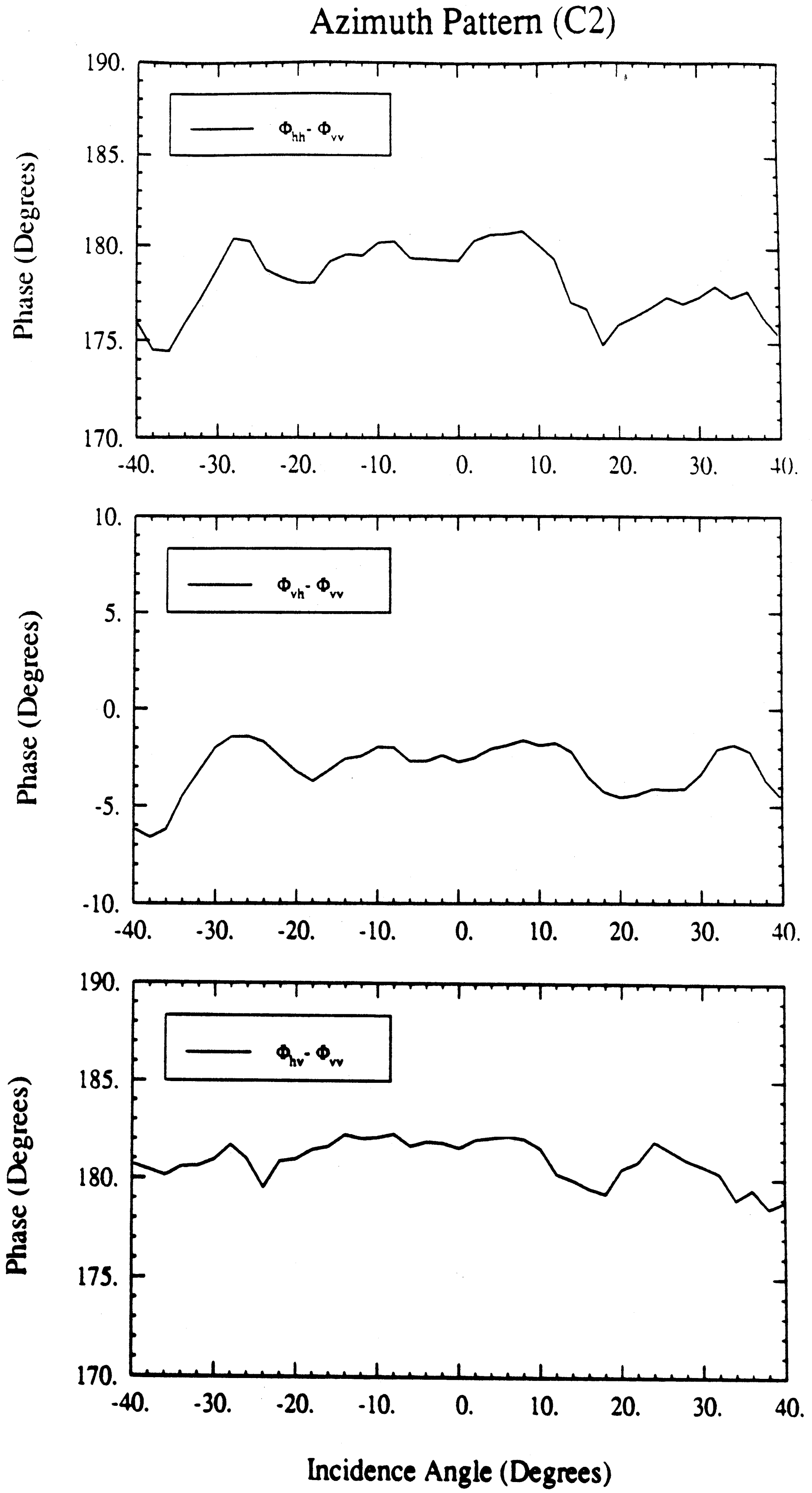

Figure 39: Azimuth pattern of phase of scattering matrix elements for C2 PARC.

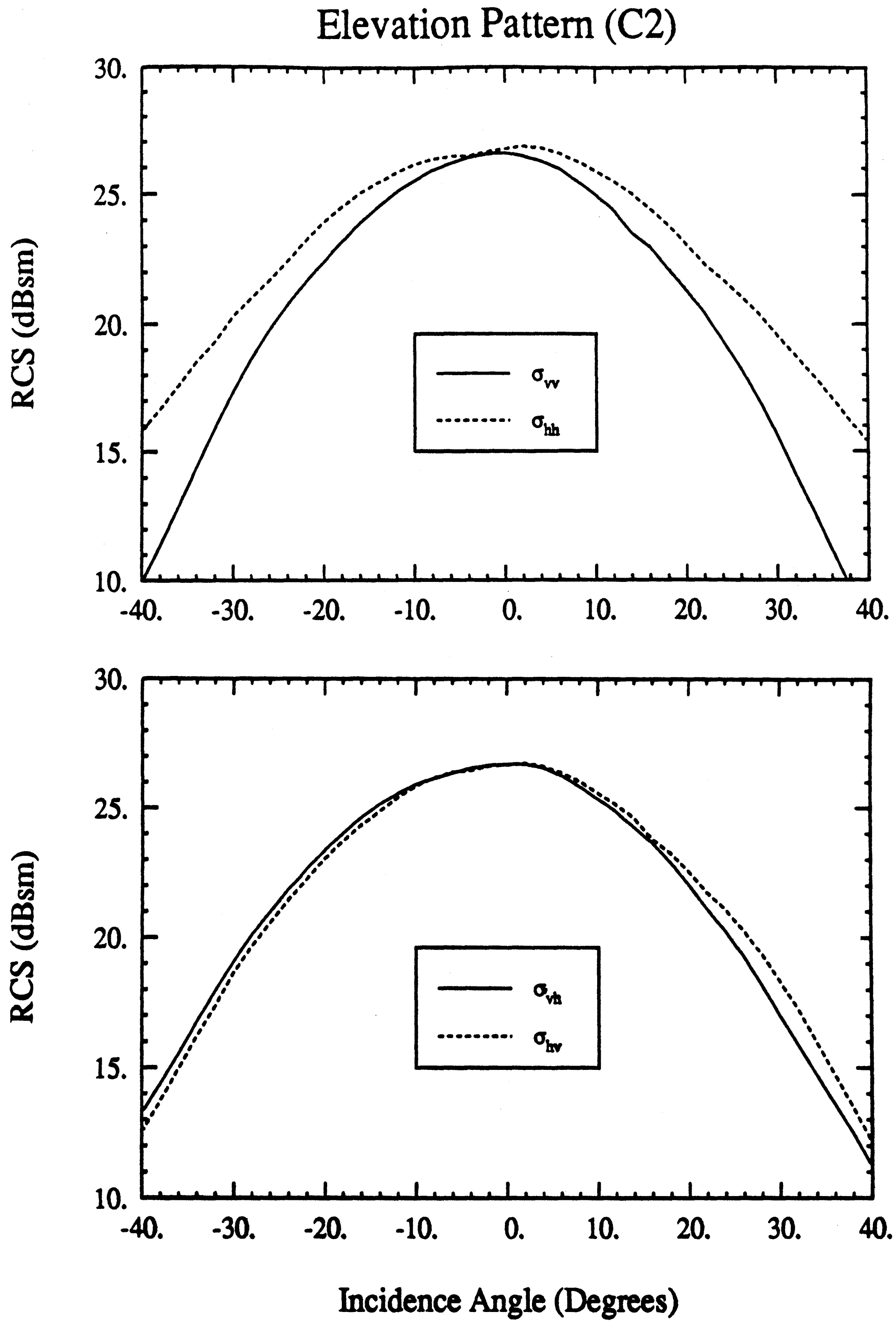

Figure 40: Elevation pattern of amplitude of scattering matrix elements for C2 PARC.

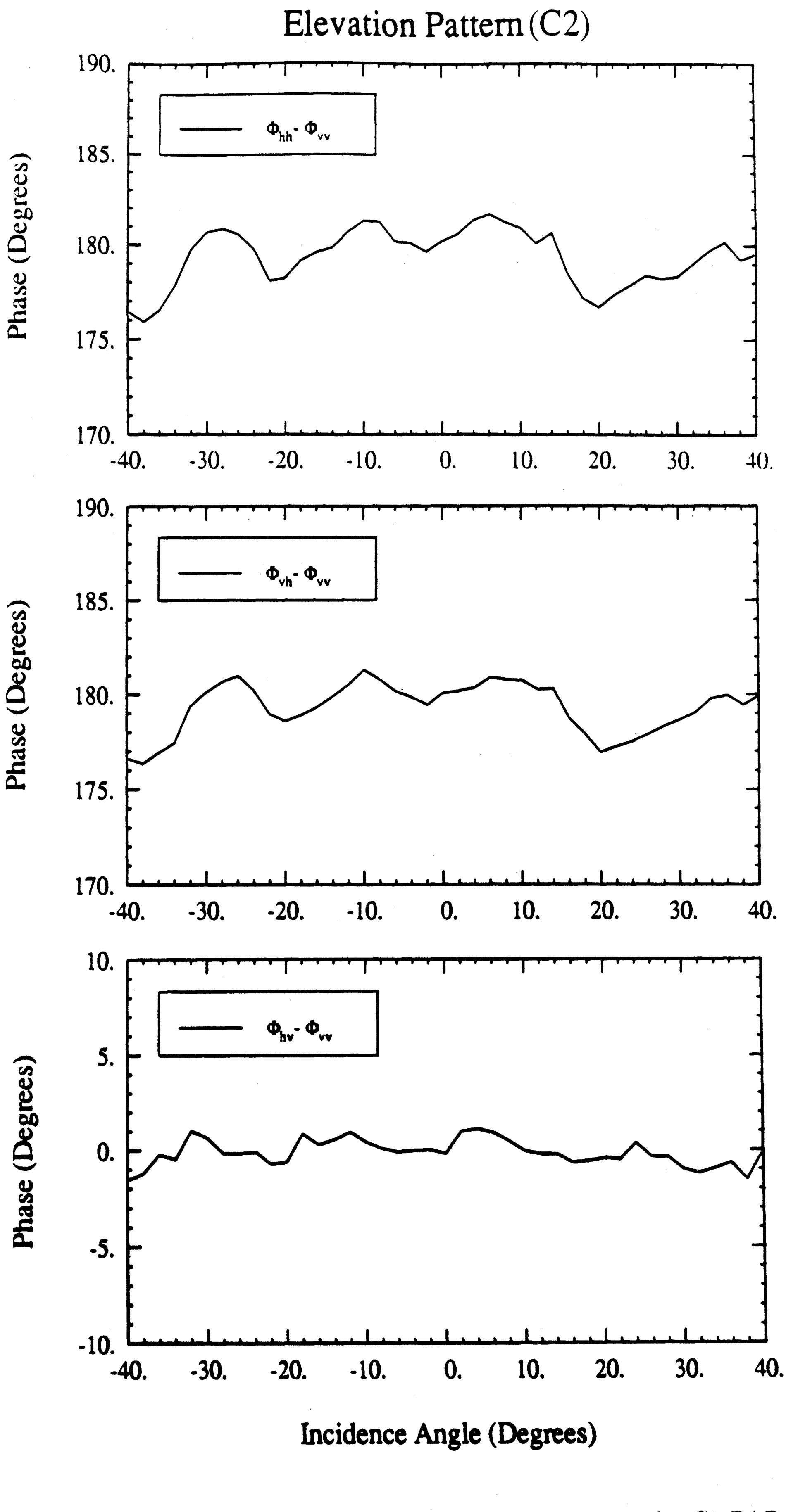

Figure 41: Elevation pattern of phase of scattering matrix elements for C2 PARC.

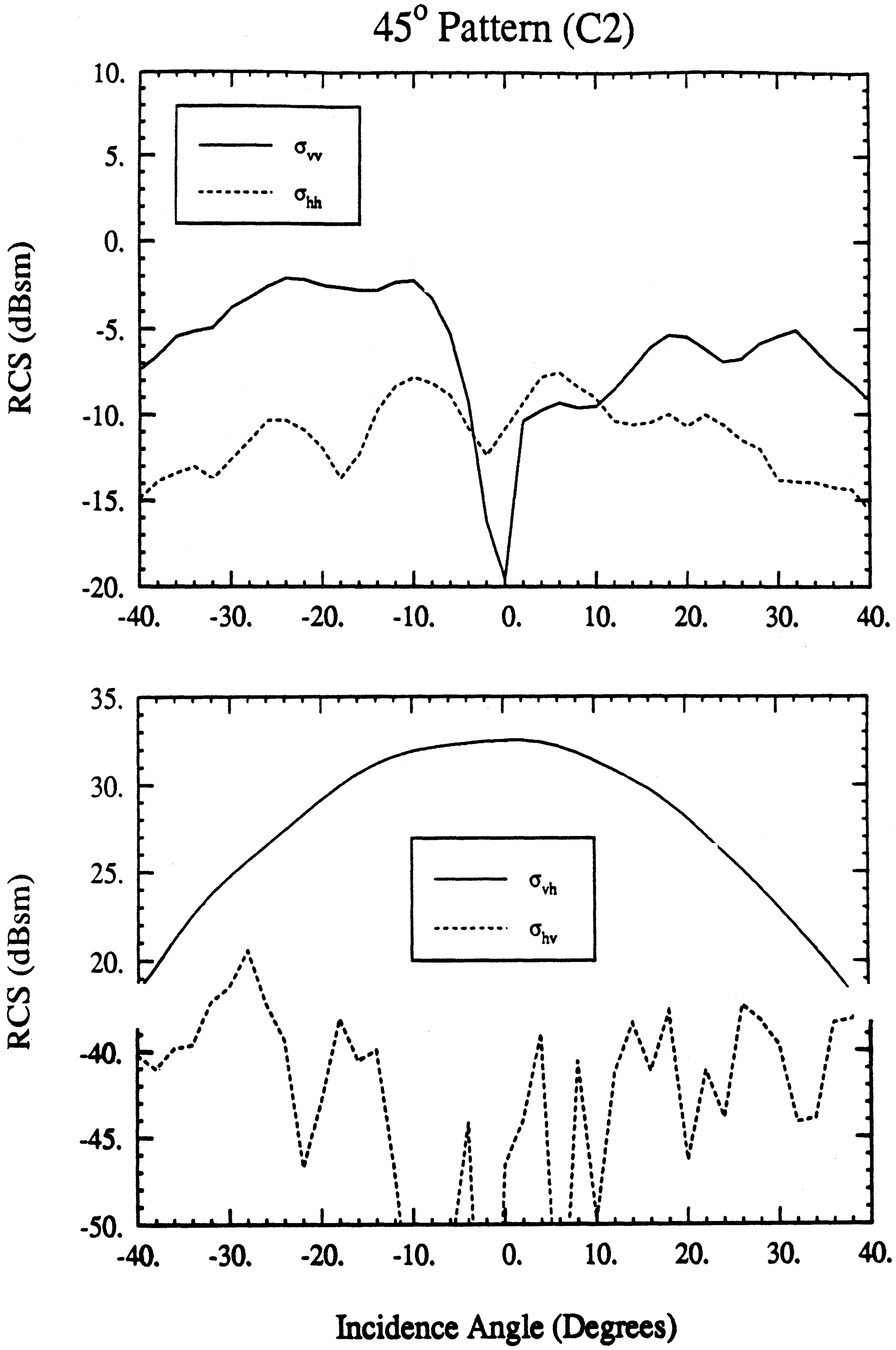

Figure 42: 45° pattern of amplitude of scattering matrix elements for C2 PARC.

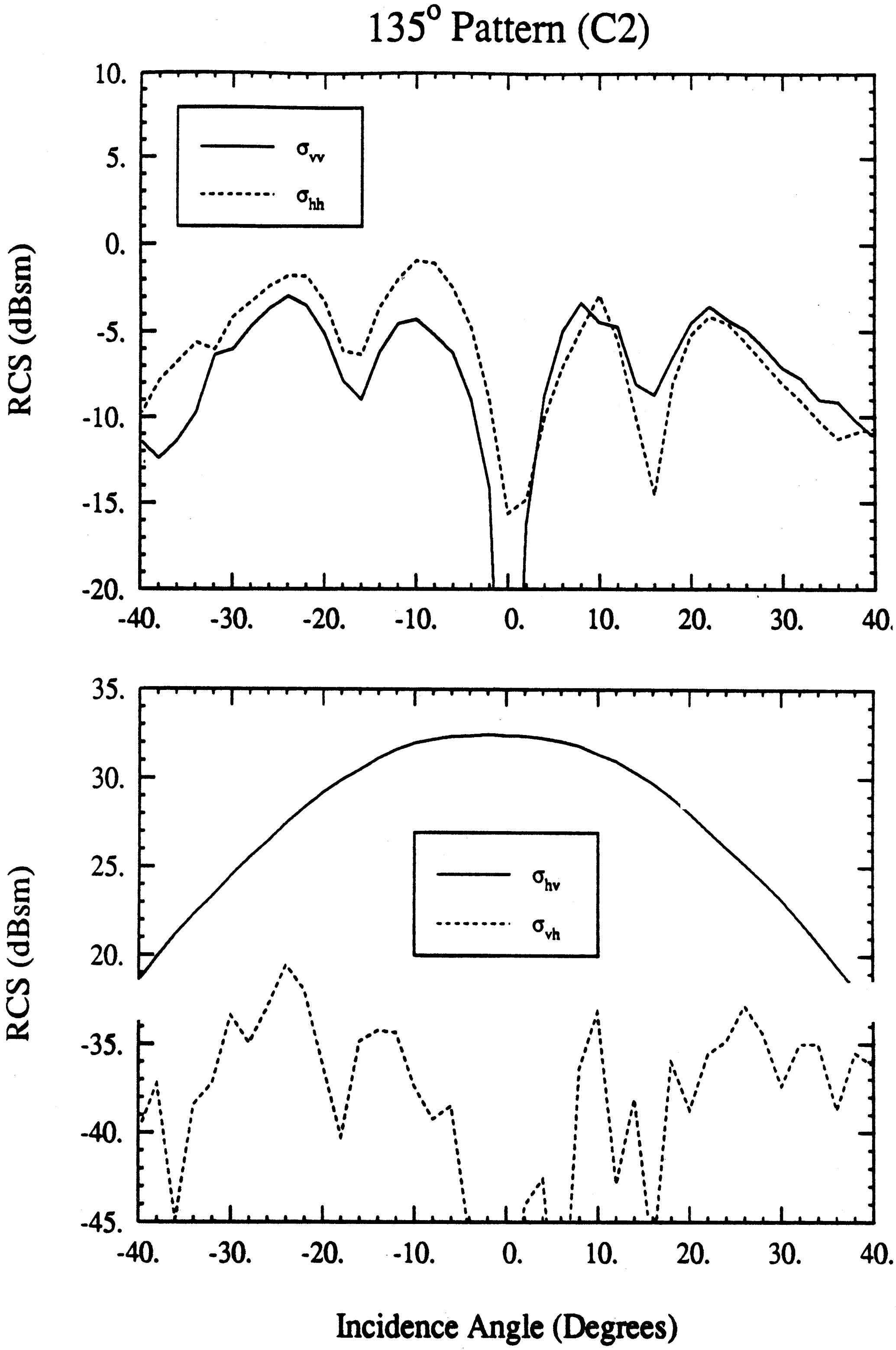

Figure 43: 135° pattern of amplitude of scattering matrix elements for C2 PARC.

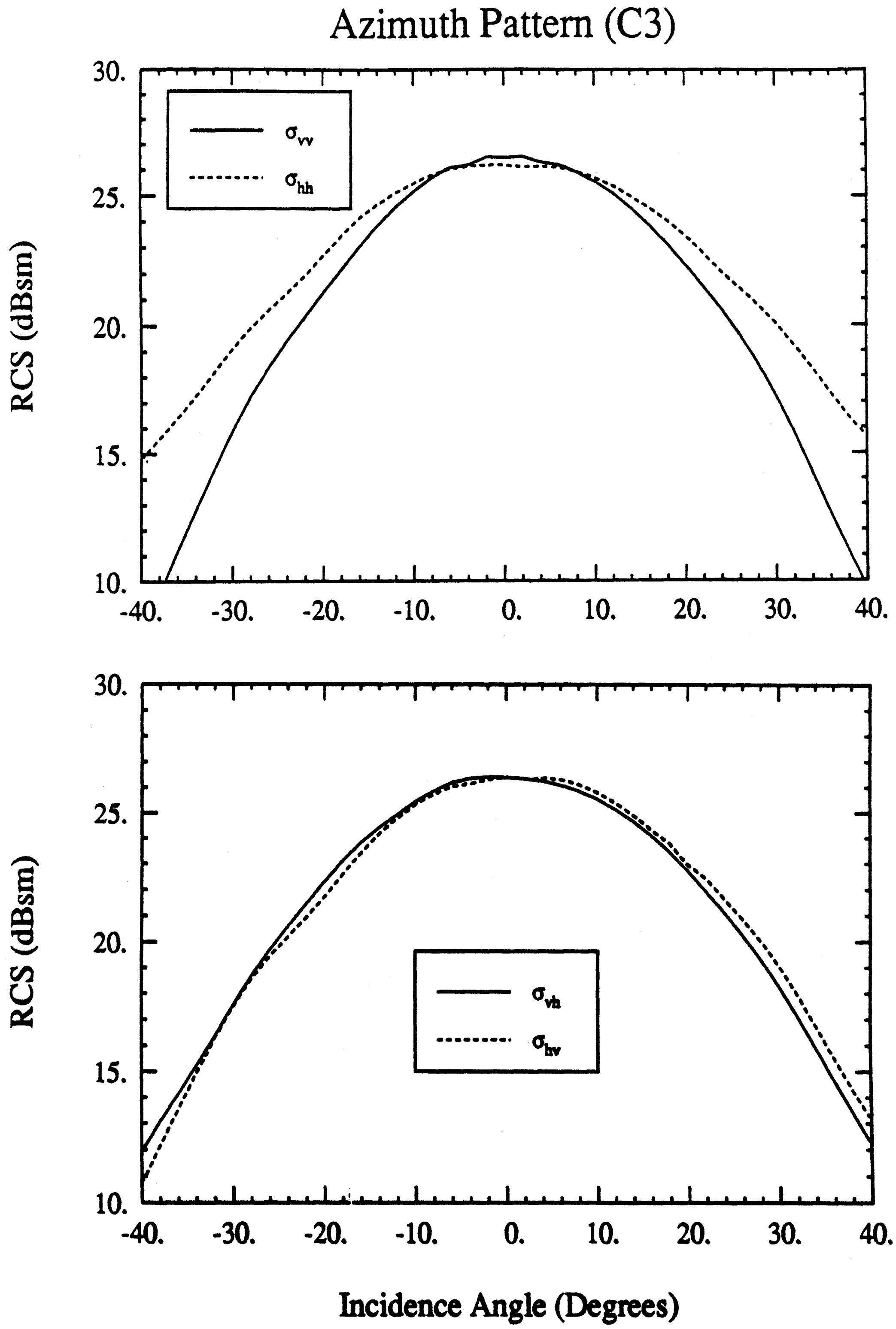

Figure 44: Azimuth pattern of amplitude of scattering matrix elements for C3 PARC.

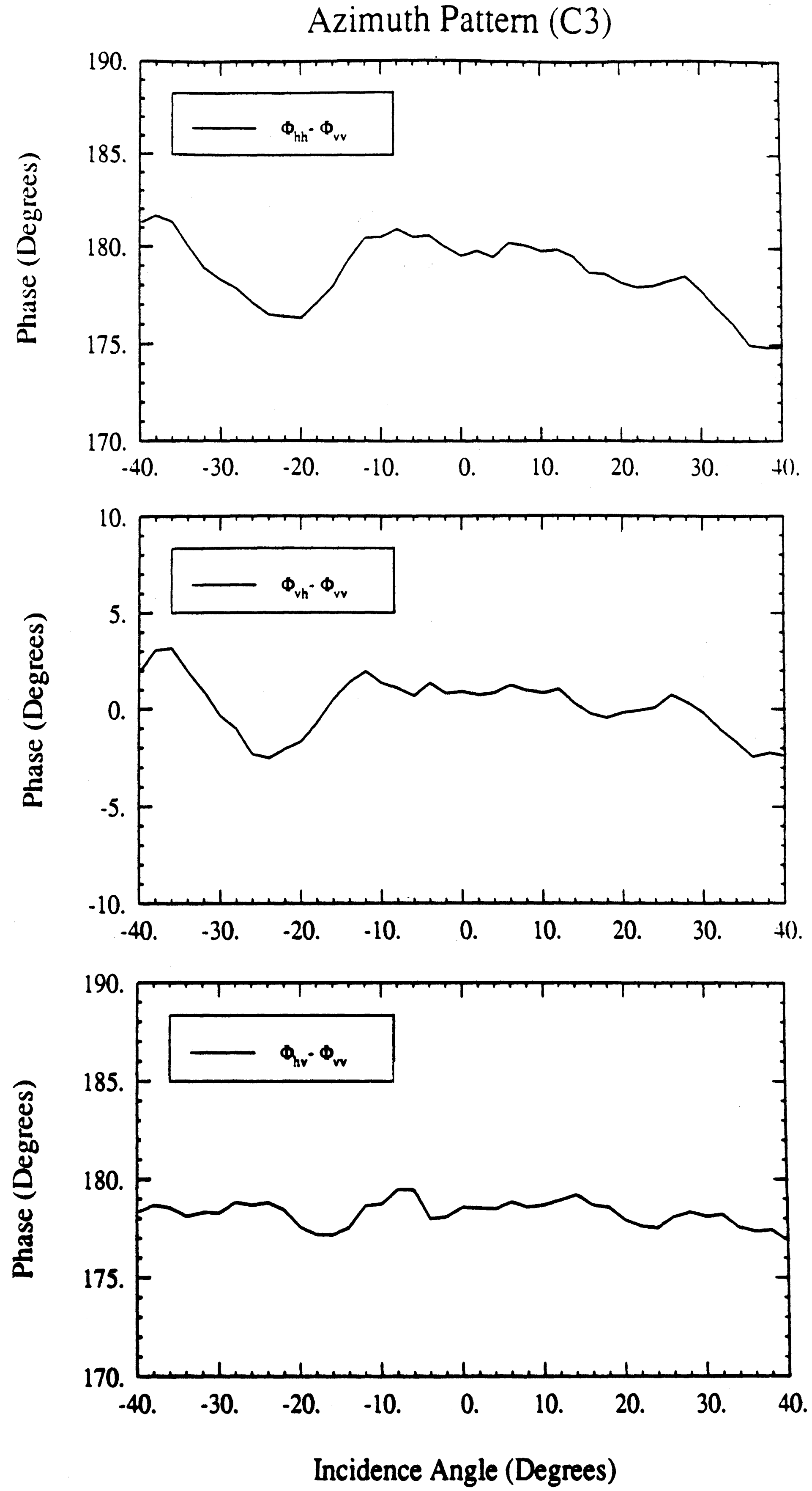

Figure 45: Azimuth pattern of phase of scattering matrix elements for C3 PARC.

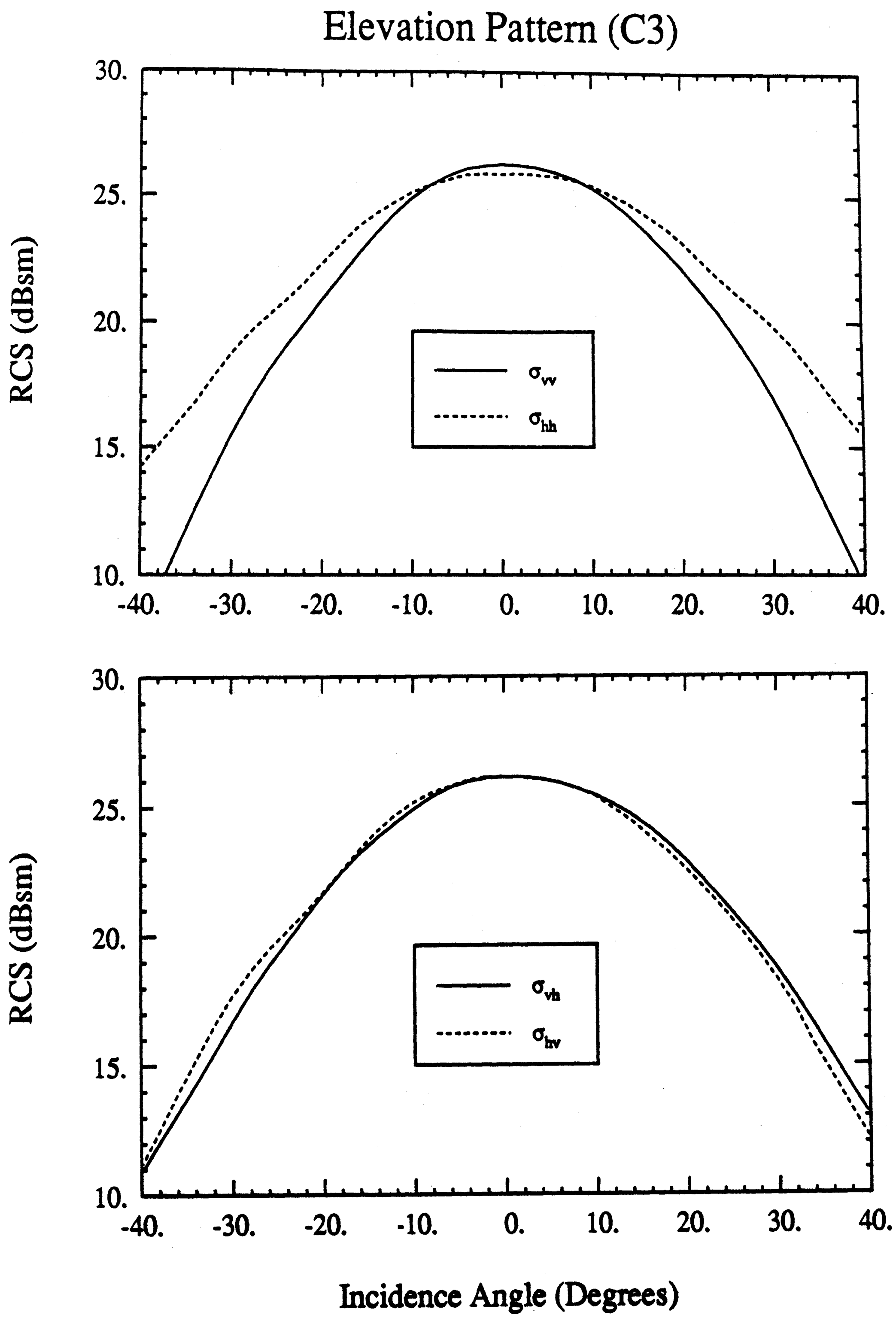

Figure 46: Elevation pattern of amplitude of scattering matrix elements for C3 PARC.

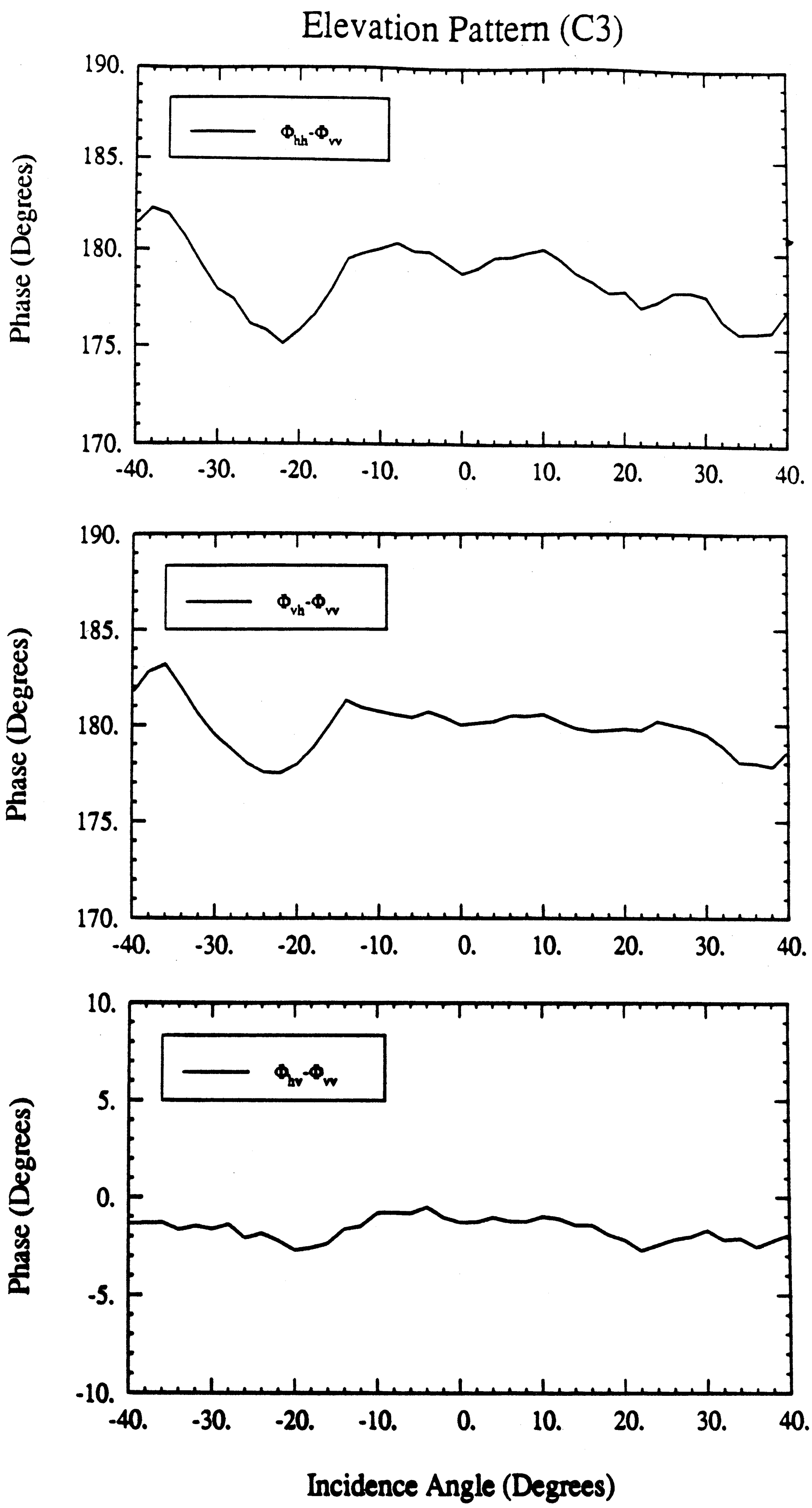

Figure 47: Elevation pattern of phase of scattering matrix elements for C3 PARC.

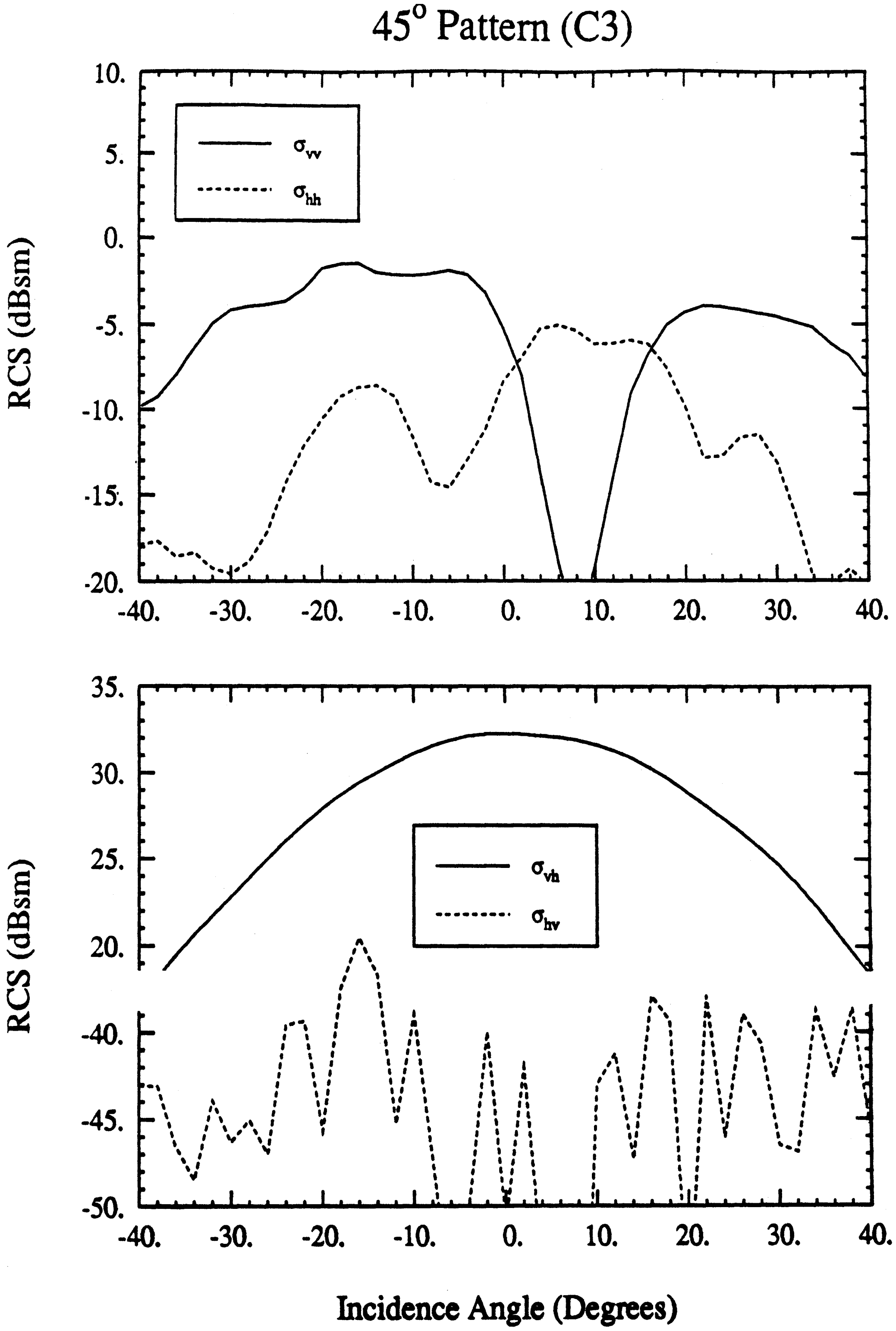

Figure 48: 45° pattern of amplitude of scattering matrix elements for C3 PARC.

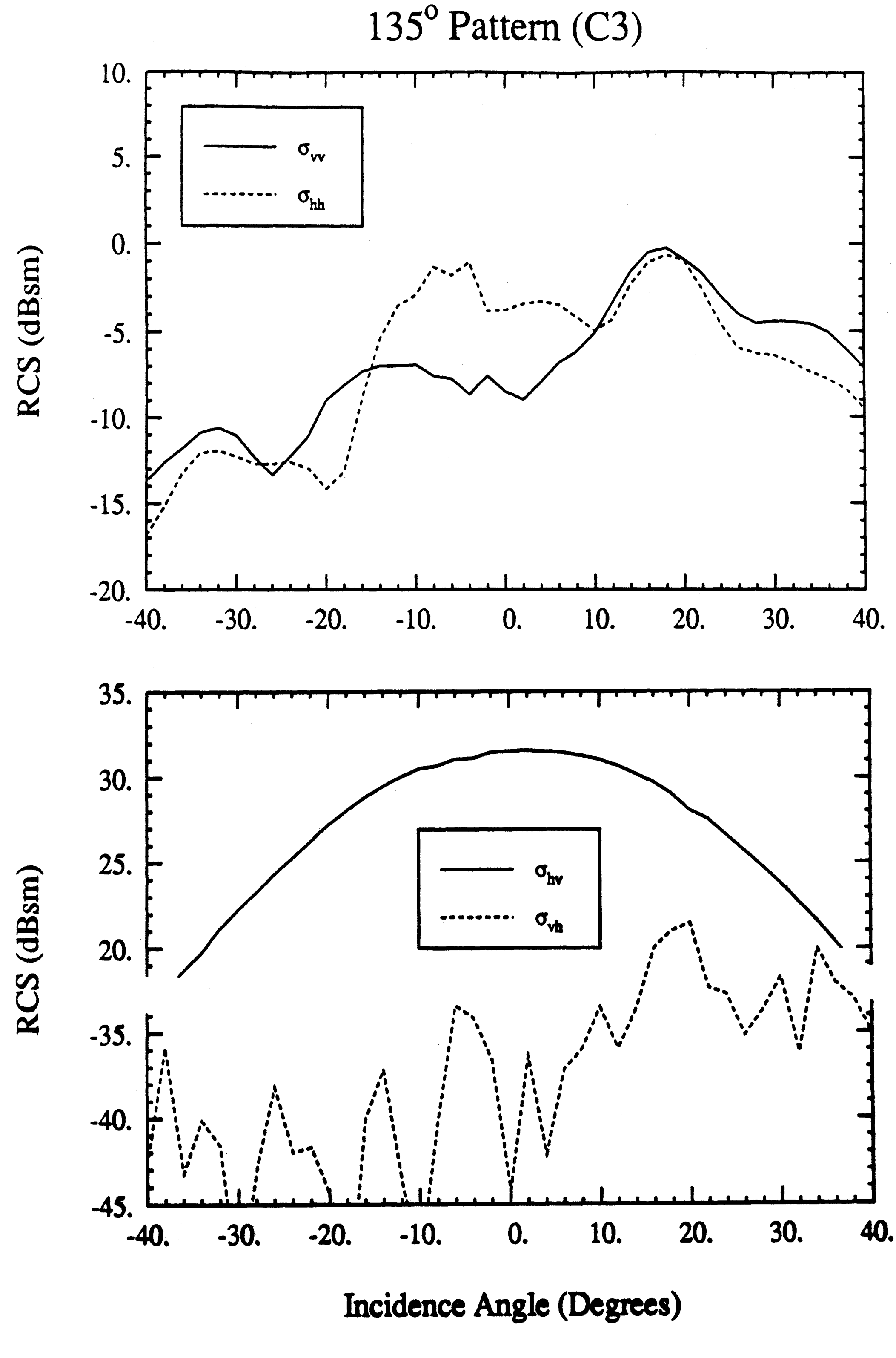

Figure 49: 135° pattern of amplitude of scattering matrix elements for C3 PARC.

APPENDIX A

A CONVENIENT TECHNIQUE FOR POLARIMETRIC CALIBRATION OF SINGLE-ANTENNA RADAR SYSTEMS

K. Sarabandi and F.T.Ulaby

Abstract- This paper introduces a practical technique for calibrating single-antenna polarimetric radar systems. With this technique, only a single calibration target, such as a conducting sphere or a trihedral corner reflector, is needed to calibrate the radar system, both in amplitude and phase, for all linear polarization configurations. By using a metal sphere, which is orientation independent, error in calibration measurement is minimized, while simultaneously calibrating the cross-polarization channels.

The antenna system and two orthogonal channels (in free space) are modeled as a four-port passive network. Upon using the reciprocity relations for the passive network, and assuming the cross-coupling terms of the antenna to be equal the cross-talk factors of the antenna system and the transmit and receive channel imbalances can be obtained from measurement of the backscatter from a metal sphere. For an X-band radar system with cross-polarization isolation of 25 dB, comparison of values measured for a sphere and a cylinder with theoretical values shows agreement within 0.4 dB in magnitude and 5° in phase. Also an effective polarization isolation of 50 dB is achieved using this calibration technique.

A1 Introduction

Accurate knowledge of the scattering matrix of a target is an important ingredient towards extracting biophysical information about the target. The scattering matrix of a target can be measured by using a set of orthogonal polarization. In practice, however, it is very difficult, if not impossible, to design an antenna system with perfect isolation between the orthogonal polarization channels, which leads to contamination of the measurements.

In recent years, considerable effort has been devoted to the development of techniques for calibrating polarimetric radar systems. Calibration techniques available in the literature can be categorized into two major groups: 1) calibration techniques for imaging radars, and 2) calibration techniques for point-target measurement systems, which may also be appropriate for imaging radars. In the first group, the scattering properties of clutter are usually employed to simplify the calibration problem [Sheen and Kasischke, 1989]. van Zyl [1989] and Klein [1989] developed a method for estimating the cross-talk contamination of the antenna by assuming that the like- and cross-polarized responses of natural targets with azimuthal symmetry are uncorrelated. Among the point-target calibration techniques, the generalized calibration technique (GCT) by Whitt and Ulaby [1989] characterizes the distortion matrices (channel imbalances and antenna cross-talk) of the receive and transmit antenna by using three calibration targets. An eigenvalue approach is employed to solve for the distortion matrices. In a similar technique by Barnes [1986] the distortion matrices are obtained by using targets with specific scattering matrices. This technique is referred to by Ulaby et al. [1989] as the constrained calibration technique (CCT). Although, in principle, GCT and CCT can fully characterize the distortion matrices, they are very sensitive to target alignment and to the knowledge of the theoretical values of the scattering matrices of the calibration targets. A third calibration technique for point targets by Sarabandi, et al. [1989] uses a sphere and any other depolarizing calibration target (scattering matrix of this target need not to be known), and is therefore immune to errors caused by target orientation and lack of precise knowledge of the theoretical values of the calibration targets' scattering matrices. However, the drawback of this method, which is called the isolated-antenna calibration technique (IACT), is that it does not account for cross-talk contamination in the antenna. The isolated antenna assumption can lead to significant errors in the cross-polarized terms when the ratio of cross- to like-polarized terms is small and/or cross-talk contamination is large.

To remove the drawback of the IACT while maintaining insensitivity to orientation of the calibration targets, we introduce in this paper a technique for calibrating single-antenna radar systems using a four-port network approach. The antenna system and two orthogonal directions in free space are modeled as a four-port network, and channel imbalances as well as the antenna cross-talk contamination are determined by measuring the backscatter from a single calibration target, namely a conducting sphere. This technique will henceforth be referred to as STCT, or single-target calibration technique. Like IACT, STCT is is insensitive to target orientation, but it also accounts for the antenna cross-talk contamination. If the antenna cross-talk contamination is very small ($\approx$ 0), the STCT is not appropriate and the IACT should be used instead.

The validity and accuracy of this technique were tested using X-band and L-band scatterometers, both in an anechoic chamber and under field conditions. Cylinders and spheres are used as test targets. Excellent agreement was obtained between the measured and theoretical values of the test target. Also the effective cross-polarization isolation obtained in this method was in the order of 50 dB.

A2 Theoretical Formulation

By defining a set of orthogonal directions in a plane perpendicular to the direction of propagation, the field components of the wave scattered by a given point target can be related to the components of the incident plane wave though the scattering matrix of the target $\mathbf{s}$. The antenna structure of a polarimetric radar system must be designed in such a way that the transmit and receive polarizations are parallel to the specified orthogonal directions. In practice, however, it is not possible to construct antennas that are totally free of polarization contaminations, i.e., coupling between the orthogonal polarization ports of the antenna. Polarization contamination (antenna cross-talk) takes place in the orthogonal mode transducer (OMT) and in the antenna structure itself.

Suppose the two orthogonal directions in free space are viewed as two ports of a four-port passive device that includes the OMT and the antenna structure (see Fig. 1). This four-port network can be characterized by a scattering matrix $\boldsymbol{\mathcal{S}}$

which relates the incident wave vector $\mathbf{V}^+$ to the reflected wave vector $\mathbf{V}^-$,

$$\mathbf{V}^- = \mathcal{S}\mathbf{V}^+$$

where

$$\mathcal{S} = \begin{bmatrix} \mathcal{S}_{11} & \mathcal{S}_{12} & \mathcal{S}_{13} & \mathcal{S}_{14} \\ \mathcal{S}_{21} & \mathcal{S}_{22} & \mathcal{S}_{23} & \mathcal{S}_{24} \\ \mathcal{S}_{31} & \mathcal{S}_{32} & \mathcal{S}_{33} & \mathcal{S}_{34} \\ \mathcal{S}_{41} & \mathcal{S}_{42} & \mathcal{S}_{43} & \mathcal{S}_{44} \end{bmatrix} \tag{A1}$$

Since the four-port device is passive its scattering matrix must be symmetric. Thus,

$$\mathcal{S}_{ij} = \mathcal{S}_{ji} \qquad i,j \in \{1,2,3,4\}$$

If the reference plane of the n^{th} port is translated outward by distance ℓ_n, the new scattering matrix for the device becomes [Collins, pp. 172, 1966]

$$\mathcal{S}' = \Theta\mathcal{S}\Theta \tag{A2}$$

where the translation matrix Θ is given by

$$\Theta = \begin{bmatrix} e^{-i\beta_1\ell_1} & 0 & 0 & 0 \\ 0 & e^{-i\beta_2\ell_2} & 0 & 0 \\ 0 & 0 & e^{-i\beta_3\ell_3} & 0 \\ 0 & 0 & 0 & e^{-i\beta_4\ell_4} \end{bmatrix} \tag{A3}$$

In (3) β_n is the propagation constant of the n^{th} port transmission line. In this case, since ports 3 and 4 are two ports in free space, $\beta_3 = \beta_4 = k_0$ and the translation matrix must be modified to account for spherical propagation. If the target is located at a distance r from the radar system and the reference planes at ports 3 and 4 are translated to the target location, then the translation matrix becomes

$$\Theta = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{e^{-ik_0 r}}{r} & 0 \\ 0 & 0 & 0 & \frac{e^{-ik_0 r}}{r} \end{bmatrix}$$

The scattering matrix, after translation of the reference planes of ports 3 and 4 by distance r, takes the following form

$$\mathcal{S}' = \begin{bmatrix} \mathcal{S}_{11} & \mathcal{S}_{12} & \mathcal{S}_{13}\frac{e^{-ik_0 r}}{r} & \mathcal{S}_{14}\frac{e^{-ik_0 r}}{r} \\ \mathcal{S}_{21} & \mathcal{S}_{22} & \mathcal{S}_{23}\frac{e^{-ik_0 r}}{r} & \mathcal{S}_{24}\frac{e^{-ik_0 r}}{r} \\ \mathcal{S}_{31}\frac{e^{-ik_0 r}}{r} & \mathcal{S}_{32}\frac{e^{-ik_0 r}}{r} & \mathcal{S}_{33}\frac{e^{-2ik_0 r}}{r^2} & \mathcal{S}_{34}\frac{e^{-2ik_0 r}}{r^2} \\ \mathcal{S}_{41}\frac{e^{-ik_0 r}}{r} & \mathcal{S}_{42}\frac{e^{-ik_0 r}}{r} & \mathcal{S}_{43}\frac{e^{-2ik_0 r}}{r^2} & \mathcal{S}_{44}\frac{e^{-2ik_0 r}}{r^2} \end{bmatrix} \qquad (A4)$$

Note that here we have ignored the gain and effective areas of the transmit and receive antennas which will be included in the channel imbalances. The signal flow-chart of the antenna system and free space ports is shown in Fig. 2. Suppose the radar is equipped with a space discriminating filter (range gating filter) which is tuned at r. The filtered scattering matrix ($\mathcal{S}''$) is then given by

$$\mathcal{S}'' = \begin{bmatrix} 0 & 0 & \mathcal{S}_{13} & \mathcal{S}_{14} \\ 0 & 0 & \mathcal{S}_{23} & \mathcal{S}_{24} \\ \mathcal{S}_{31} & \mathcal{S}_{32} & 0 & 0 \\ \mathcal{S}_{41} & \mathcal{S}_{42} & 0 & 0 \end{bmatrix} \frac{e^{-ik_0 r}}{r}$$

Basically, short-range reflections from the antenna system and multiple bounces between the antenna and the target have been gated out. The incident and reflected waves at each port can now be represented by two uncoupled matrix equations as follows

$$\begin{bmatrix} V_1^- \\ V_2^- \end{bmatrix} = \frac{e^{-ik_0 r}}{r} \begin{bmatrix} S_{13} & S_{14} \\ S_{23} & S_{24} \end{bmatrix} \begin{bmatrix} V_3^+ \\ V_4^+ \end{bmatrix} \tag{A5}$$

$$\begin{bmatrix} V_3^- \\ V_4^- \end{bmatrix} = \frac{e^{-ik_0 r}}{r} \begin{bmatrix} S_{31} & S_{32} \\ S_{41} & S_{42} \end{bmatrix} \begin{bmatrix} V_1^+ \\ V_2^+ \end{bmatrix} \tag{A6}$$

On the other hand, the incident and reflected waves at ports 3 and 4 are the scattered and incident waves, respectively, of the target, which is represented by a two-port network and thus related to each other by the scattering matrix of the target s. That is,

$$\begin{bmatrix} V_3^+ \\ V_4^+ \end{bmatrix} = \begin{bmatrix} s_{vv} & s_{vh} \\ s_{hv} & s_{hh} \end{bmatrix} \begin{bmatrix} V_3^- \\ V_4^- \end{bmatrix} \tag{A7}$$

Note that the scattering matrix used here is defined in the backscattering alignment convention since the orthogonal directions are specified independent of the incident and scattering directions. After rearranging (5)-(7) in order to relate the reflected waves to the incident waves at ports 1 and 2, in addition to employing the reciprocity property of the four-port network, we get

$$\begin{bmatrix} V_1^- \\ V_2^- \end{bmatrix} = \frac{e^{-2ik_0 r}}{r^2} \begin{bmatrix} S_{13} & S_{14} \\ S_{23} & S_{24} \end{bmatrix} \begin{bmatrix} s_{vv} & s_{vh} \\ s_{hv} & s_{hh} \end{bmatrix} \begin{bmatrix} S_{13} & S_{23} \\ S_{14} & S_{24} \end{bmatrix} \begin{bmatrix} V_1^+ \\ V_2^+ \end{bmatrix} \tag{A8}$$

Upon normalizing with respect to the like-polarized channels (8) becomes

$$\begin{bmatrix} V_1^- \\ V_2^- \end{bmatrix} = \frac{e^{-2ik_0 r}}{r^2} \begin{bmatrix} \mathcal{S}_{13} & 0 \\ 0 & \mathcal{S}_{24} \end{bmatrix} \begin{bmatrix} 1 & C_1 \\ C_2 & 1 \end{bmatrix} \begin{bmatrix} s_{vv} & s_{vh} \\ s_{hv} & s_{hh} \end{bmatrix} \begin{bmatrix} 1 & C_2 \\ C_1 & 1 \end{bmatrix} \begin{bmatrix} \mathcal{S}_{13} & 0 \\ 0 & \mathcal{S}_{24} \end{bmatrix} \begin{bmatrix} V_1^+ \\ V_2^+ \end{bmatrix} \tag{A9}$$

where $C_1 = \mathcal{S}_{14}/\mathcal{S}_{13}$ and $C_2 = \mathcal{S}_{23}/\mathcal{S}_{24}$ are the antenna cross-talk factors.

So far we have modeled the antenna system, free space channel, and the target by a two-port network. To account for the effects of active circuits on the performance of the overall radar system, let us consider the block diagram depicted in Fig. 3. The transmit (V_{tv}, V_{th}) and receive (V_{rv}, V_{rh}) voltages are the quantities measured by the radar. The channel imbalance quantities (T_v, T_h, R_v, R_h), which relate the transmit/receive voltages to the incident/reflected waves at port 1 and 2, account for variations (in both amplitude and phase) of the active circuits and the antenna gains. The transmit and receive channels of the radar system are separated by a transmit-receive switch (TR switch) or a circulator. These components can be assumed ideal because any leakage that may occur will not be sampled by the range gating process. Therefore, the transmit and receive voltages can be related to the incident and reflected voltages of ports 1 and 2 by

$$\begin{bmatrix} V_{rv} \\ V_{rh} \end{bmatrix} = \begin{bmatrix} R_v & 0 \\ 0 & R_h \end{bmatrix} \begin{bmatrix} V_1^- \\ V_2^- \end{bmatrix} \tag{A10}$$

$$\begin{bmatrix} V_1^+ \\ V_2^+ \end{bmatrix} = \begin{bmatrix} T_v & 0 \\ 0 & T_h \end{bmatrix} \begin{bmatrix} V_{tv} \\ V_{th} \end{bmatrix} \tag{A11}$$

Using (9) in (10) and (11) results in

$$\begin{bmatrix} V_{rv} \\ V_{rh} \end{bmatrix} = \frac{e^{-2ik_0 r}}{r^2} \begin{bmatrix} R_v S_{13} & 0 \\ 0 & R_h S_{24} \end{bmatrix} \begin{bmatrix} 1 & C_1 \\ C_2 & 1 \end{bmatrix} \begin{bmatrix} s_{vv} & s_{vh} \\ s_{hv} & s_{hh} \end{bmatrix} \begin{bmatrix} 1 & C_2 \\ C_1 & 1 \end{bmatrix} \begin{bmatrix} T_v S_{13} & 0 \\ 0 & T_h S_{24} \end{bmatrix} \begin{bmatrix} V_{tv} \\ V_{th} \end{bmatrix} \tag{A12}$$

which may be written in matrix notation as

$$\mathbf{V}_r = \frac{e^{-2ik_0 r}}{r^2} \mathbf{RCsC}^T\mathbf{TV}_t \tag{A13}$$

The matrix $\mathbf{M} = \mathbf{RCsC}^T\mathbf{T}$ represents the measured (uncalibrated) scattering matrix of the target under observation. If the matrices **C**, **R**, and **T** are known, the actual scattering matrix **s** can then be obtained. To determine **C**, **R**, and **T** we note that these matrices depend on the choice of the orthogonal channels in free space. So far we have made no assumption on the direction of the orthogonal channels (v and h) except that they are perpendicular to the direction of propagation. Once the v and h directions are specified, the scattering matrix of the target can, in principle, be determined. A radar system with linear polarization configurations usually is oriented such that for a given polarization most of the transmitted energy falls into the desired channel, i.e. an orientation for which the antennas cross-talk factors (C_1 and C_2) are minimal. With available design techniques, it is easy to achieve the conditions $| C_1 | \leq 0.1$ and $| C_2 | \leq 0.1$, which correspond to a polarization isolation level of 20 dB, but achieving much greater isolation level is difficult. For accurate polarimetric measurements, the effective isolation level should be on the order of 40 dB. Hence the factors C_1 and C_2 may not be ignored, but should instead be determined by the calibration technique.

To demonstrate how the choice of the coordinate frame affects the antennas' cross-talk factors, we obtain a relationship for the cross-talk factors when the coordinate frame is rotated by an angle ψ. Suppose $\mathbf{s}$ represents the scattering matrix of a target for a particular coordinate frame and $\mathbf{s}'$ denotes the scattering matrix of the same target when the coordinate frame is rotated around the incidence direction by an angle ψ. It is a trivial matter to show that

$$\mathbf{s}' = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \mathbf{s} \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \tag{A14}$$

Let us indicate the imbalance and cross-talk matrices in the rotated coordinate system by a prime sign. Since the relative orientation of the antenna system and target have not been changed, the measured scattering matrices for two coordinate systems are identical. That is, $\mathbf{M}' = \mathbf{M}$. Therefore,

$$\mathbf{M} = \mathbf{RCsC}^T\mathbf{T} = \mathbf{R}'\mathbf{C}'\mathbf{s}'\mathbf{C}'^T\mathbf{T}' \tag{A15}$$

Using (14) in (15) results in

$$\mathbf{M} = \mathbf{R}'\mathbf{PsP}^T\mathbf{T}' \tag{A16}$$

where

$$\mathbf{P} = \begin{bmatrix} \cos\psi + C_1'\sin\psi & -\sin\psi + C_1'\cos\psi \\ \sin\psi + C_2'\cos\psi & \cos\psi - C_2'\sin\psi \end{bmatrix} \tag{A17}$$

Again, by normalizing the diagonal elements of the matrix **P** to 1 and then comparing the resultant matrices to **C**, **R**, and **T** it can easily be inferred that

$$C_1 = \frac{-\sin\psi + C_1' \cos\psi}{\cos\psi + C_1' \sin\psi} \tag{A18}$$

$$C_2 = \frac{\sin\psi + C_2' \cos\psi}{\cos\psi - C_2' \sin\psi} \tag{A19}$$

$$\mathcal{S}_{13} = (\cos\psi + C_1' \sin\psi)\mathcal{S}_{13}'$$

$$\mathcal{S}_{24} = (\cos\psi - C_2' \sin\psi)\mathcal{S}_{24}'$$

If it is required that maximum energy falls into the v channel when port 1 (v port of the OMT) is energized, the condition $C_1 = 0$ should be enforced which is equivalent to setting $\psi = \arctan C_1'$. This condition, however, does not maximize the energy transfer into the h channel when the h port of the OMT is energized. Moreover the cross-talk terms usually are complex quantities and the above condition may not be achievable. In order to maximize the energy transfer for both channels and simplify the calibration procedure we look for a rotation angle such that $C_1 = C_2 \ll 1$, i.e.

$$\frac{-\sin\psi + C_1' \cos\psi}{\cos\psi + C_1' \sin\psi} = \frac{\sin\psi + C_2' \cos\psi}{\cos\psi + C_2' \sin\psi}$$

which requires that

$$\psi = \frac{1}{2} \arctan \frac{C_1' - C_2'}{1 + C_1' C_2'}. \tag{A20}$$

This can be accomplished if both C_1' and C_2' are real quantities. It is relatively easy to design the antenna such that $C_1' \approx C_2'$ to begin with, and then by adjusting the rotation angle using (20) the cross-talk factors can be made even more similar. Therefore from here on, we shall assume that the antennas' cross-talk factors are identical and the error associated with this assumption is on the order of the difference in the imaginary parts of the cross-talk factors. In view of this approximation, the measured scattering matrix is given by

$$\mathbf{M} = \mathbf{R}\begin{bmatrix} 1 & C \\ C & 1 \end{bmatrix}\mathbf{s}\begin{bmatrix} 1 & C \\ C & 1 \end{bmatrix}\mathbf{T}. \tag{A21}$$

A3 Calibration Procedure

The relationship between the measured scattering matrix **M** and the actual scattering matrix of an unknown target is given by equation (21). If the elements of the imbalance and cross-talk matrices are known, the scattering matrix of the target can be obtained from

$$\mathbf{s} = \mathbf{C}^{-1}\mathbf{R}^{-1}\mathbf{M}\mathbf{T}^{-1}\mathbf{C}^{-1}.$$

The matrices **R** and **T** are diagonal and **C** is symmetric with known diagonal elements; therefore, there is a total of five unknowns that need to be determined. The standard approach is to measure targets with known scattering matrices to establish a set of equations for the unknown elements of the **R**, **T**, and **C** matrices. By measuring each calibration target, four nonlinear equations are obtained, so it seems that at least two targets are needed to find all the five unknowns. But as will be shown, a sphere or a target with similar scattering matrix (such as a trihedral) is sufficient to characterize the scattering matrix of the unknown target. In fact, it is not required to find all the five unknowns to obtain the scattering matrix of the unknown target.

Upon expanding (21) and noting that for backscattering, the scattering matrix is symmetric ($s_{12} = s_{21}$) we get

$$\mathbf{M} = \mathbf{R} \begin{bmatrix} s_{vv} + 2Cs_{vh} + C^2 s_{hh} & (1+C^2)s_{vh} + C(s_{vv} + s_{hh}) \\ (1+C^2)s_{vh} + C(s_{vv} + s_{hh}) & s_{hh} + 2Cs_{vh} + C^2 s_{vv} \end{bmatrix} \mathbf{T} \, . \quad \text{(A22)}$$

For the sake of simplicity, the diagonal elements of $\mathbf{R}$ and $\mathbf{T}$ will be denoted by R_i and T_i ($i = 1, 2$) respectively. Measuring a sphere with radar cross section $\sigma^\circ = |s^\circ|^2$, (22) provides the following set of equations

$$R_1 T_1 (1 + C^2) = \frac{m_{11}^\circ}{s^\circ} \quad \text{(A23)}$$

$$2R_1 T_2 C = \frac{m_{12}^\circ}{s^\circ} \quad \text{(A24)}$$

$$2R_2 T_1 C = \frac{m_{12}^\circ}{s^\circ} \quad \text{(A25)}$$

$$R_2 T_2 (1 + C^2) = \frac{m_{22}^\circ}{s^\circ} \quad \text{(A26)}$$

where m_{ij}° denotes the ij^{th} elements of the measured scattering matrix of the sphere. The cross-talk term can be obtained by multiplying (23) by (26) and (24) by (25) and then eliminating the term $R_1 T_1 R_2 T_2$ from the resultant equations. Thus

$$\frac{4C^2}{(1+C^2)^2} = \frac{m_{12}^\circ m_{21}^\circ}{m_{11}^\circ m_{22}^\circ} \triangleq a$$

which is a biquadratic equation with four possible solutions given by

$$C = \pm \frac{1}{\sqrt{a}} \pm \sqrt{\frac{1}{a} - 1}.$$

Requiring C to be a small number, two of these solutions can be discarded (note that $|a| \ll 1$), therefore

$$C = \pm\frac{1}{\sqrt{a}}(1 - \sqrt{1-a}) \tag{A27}$$

To meet the condition $|C| < 1$, the branch cut for $\sqrt{1-a}$ is chosen such that $\mathrm{Re}[\sqrt{1-a}] > 0$. Therefore C is determined from the sphere measurement within a $\pm$ sign.

By denoting the measured scattering matrix elements of the unknown target by m_{ij}^u and using (23)-(26) to find the products of R_iT_j, we obtain

$$s_{vv} + C(s_{vh} + s_{hv}) + C^2 s_{hh} = \frac{m_{11}^u}{m_{11}^o}(1 + C^2)s^o \tag{A28}$$

$$C^2 s_{vv} + C(s_{vh} + s_{hv}) + s_{hh} = \frac{m_{22}^u}{m_{22}^o}(1 + C^2)s^o \tag{A29}$$

$$C(s_{vv} + s_{hh}) + s_{vh} + C^2 s_{hv} = \frac{m_{12}^u}{m_{12}^o}(2C)s^o \tag{A30}$$

$$C(s_{vv} + s_{hh}) + s_{hv} + C^2 s_{vh} = \frac{m_{21}^u}{m_{21}^o}(2C)s^o \tag{A31}$$

Solving these equations simultaneously, the unknown scattering matrix elements can be obtained and are given by

$$s_{vv} = \frac{1}{(1-C^2)^2}\left[-2C^2\left(\frac{m_{12}^u}{m_{12}^o} + \frac{m_{21}^u}{m_{21}^o}\right) + (1 + C^2)\left(\frac{m_{11}^u}{m_{11}^o} + C^2\frac{m_{22}^u}{m_{22}^o}\right)\right] s^o \tag{A32}$$

$$s_{hh} = \frac{1}{(1-C^2)^2}\left[-2C^2\left(\frac{m_{12}^u}{m_{12}^o} + \frac{m_{21}^u}{m_{21}^o}\right) + (1 + C^2)\left(\frac{m_{22}^u}{m_{22}^o} + C^2\frac{m_{11}^u}{m_{11}^o}\right)\right] s^o \tag{A33}$$

$$s_{vh} = \frac{C}{(1-C^2)^2}\left[2\frac{m_{12}^u}{m_{12}^o} + 2C^2\frac{m_{21}^u}{m_{21}^o}) - (1 + C^2)\left(\frac{m_{11}^u}{m_{11}^o} + \frac{m_{22}^u}{m_{22}^o}\right)\right] s^o \tag{A34}$$

$$s_{hv} = \frac{C}{(1-C^2)^2}\left[2\frac{m_{21}^u}{m_{21}^o} + 2C^2\frac{m_{12}^u}{m_{12}^o}) - (1 + C^2)\left(\frac{m_{11}^u}{m_{11}^o} + \frac{m_{22}^u}{m_{22}^o}\right)\right] s^o \tag{A35}$$

It should be pointed out that there is no ambiguity in s_{11} and s_{22} since the the branch of $\sqrt{1-a}$ is defined, but there is a 180° phase ambiguity in s_{12} and s_{21}.

Expressions (32)-(35) give the elements of the scattering matrix when the calibration and the unknown targets are at the same range from the radar. If the range of the calibration target (r_0) is different from the range of the unknown target (r_u), (32)-(35) must be modified by a multiplying factor $(\frac{r_u}{r_0})^2 e^{-2ik_0(r_0-r_u)}$.

The complex quantity C is an inherent characteristic of the antenna system and does not change with variations in the performance of the active devices in the radar system, and therefore is less affected by environmental changes. The ambiguity in the sign of C for an antenna system may be easily resolved once by measuring a target with a known phase relationship between the elements of its scattering matrix (such as a tilted cylinder).

To investigate the accuracy of the measurement of C, we use the fact that $|a| \ll 1$ and therefore (27) becomes $C \approx \frac{1}{2}\sqrt{a}$. If the uncertainty in measurement of a is represented by Δ and $|\Delta| \ll |a|$, then

$$C + \delta C = \frac{1}{2}\sqrt{a+\Delta} \approx \frac{1}{2}\sqrt{a}(1 + \frac{\delta}{a})$$

from which we get

$$\frac{\delta C}{C} = \frac{1}{2}\frac{\Delta}{a}\,.$$

It is concluded that the uncertainty in C is about 50% of the uncertainty in measuring a.

Using a sphere as the calibration target not only simplifies calculation of the unknowns significantly, but also offers two more advantages. One advantage is that the scattering matrix of a sphere is insensitive to orientation and therefore no error will be incurred because of target orientation. The second advantage stems from the fact that spheres are the only three-dimensional structures for which an exact theoretical scattering matrix is known.

A4 Comparison with Measured Data

The validity of the STCT is now examined by measuring scattering matrices of cylinders and spheres as test targets employing a polarimetric X-band scatterometer. The results based on the IACT also is included for comparison. The measurements were performed in a 14-m long anechoic chamber and the target orientation was facilitated by a very fine-tune azimuth over elevation stepper motor positioner. Detailed description of the scatterometer and measurement setup is given in [3].

The analysis given in section 2 does not take into account the effect of noise and disturbances. In reality the measured scattering matrix includes an additive noise factor and therefore (21) becomes

$$\mathbf{M} = \mathbf{RCsCT} + \mathbf{N} \tag{A36}$$

where $\mathbf{N}$ is a matrix representation of disturbances. In order to measure $\mathbf{s}$ accurately, all the elements of $\mathbf{N}$ must be much smaller than the elements of $\mathbf{M}$. The disturbances, for a typical radar system, may include thermal and background noise. Thermal noise is a zero-mean random process with power proportional to the product of the system bandwidth and noise temperature. This effect can be minimized using an averaging process. The background noise includes the signal returns from objects at ranges comparable to that of the test target or the short-range multiple reflections within the system. This problem can be eliminated using background subtraction from the target and background response. Another source of error in the measurement of **s** is the interaction of the target with its support structure (pedestal). This interaction is not linear and hence cannot be subtracted out.

A 12-inch diameter sphere was used as the calibration target, and test targets included a 6-inch sphere, a 8-inch sphere, and a conducting cylinder with a diameter of 0.8 cm and length of 27.2 cm, observed at three different orientations (vertical, horizontal, and 45°). The calibrated elements of the scattering matrix were then compared with the theoretical values computed using the exact Mie-series solution for the spheres and using a semi-exact solution for the cylinder. The semi-exact solution is based on the assumption that the current induced on the surface of the cylinder is identical with that of an infinite cylinder with the same radius. This solution is accurate in the specular direction and when the cylinder length is much larger than the wavelength.

Using averaging and background subtraction a signal to noise ratio of better than 40 dB was achieved in measuring the elements of **M**. For targets with $s_{12} = 0$ (such as sphere and vertical cylinder), the signal to noise ratio for the off-diagonal elements of **M** was better than 25 dB.

Figure 4-6 compare the theoretical and measured scattering matrix elements of the 6-inch sphere. The error in the like-polarized terms is less than 0.3 dB and the results based on the IACT and STCT are exactly identical. It is also shown that the error in the phase is less than 2 degrees. Figure 6 shows the cross-polarized component of the sphere (theoretical value $=-\infty$ in dB scale) where there is a significant disagreement between STCT and IACT. An effective polarization isolation of 50 dB is obtained using STCT. It should be pointed out that the minimum noise-equivalent cross section of the radar system is -65 dBsm. Therefore the cross polarization isolation is limited by the system noise in this case. Similar results were also obtained for the 8-inch sphere. Figures 7-10 depict the results for the vertical cylinder and excellent agreement between the measured data and theory is achieved. As shown in Fig. 9 the measured effective polarization isolation for the vertical cylinder is -50 dB. Results for the 45° tilted cylinder are shown in Figs. 11-15 where the accuracy of the STCT is within ±0.4 dB in magnitude and ±5° in phase. These plots demonstrate the superiority of the STCT over the IACT.

A5 Concluding Remarks

A convenient calibration technique for single-antenna polarimetric radar system with range-gating capability has been developed. The radar cross-talk contamination and channel imbalances are obtained by measuring the backscatter from a single calibration target whose scattering matrix is diagonal with equal diagonal entries. The insensitivity to alignment of calibration target offered by this technique makes it particularly useful for field operation.

Using a four-port network approach, it is shown that the cross-talk contamination factor is a feature of the antenna system only, and hence is not affected by instability of active devices in the radar system. Excellent agreement between measurements and theory was obtained when a sphere was used as the calibration target and cylinders and other spheres were used as test targets. A minimum effective polarization isolation of 50 dB was achieved using this technique.

Acknowledgement This work was supported by JPL SIR-C calibration project.

References

[1] Barnes, R.M., "Polarimetric calibration using in-scene reflectors," Rep. TT.65, MIT, Lincoln Laboratory, Lexington, MA, Sept. 1986.

[2] Klein, J. D.,"Calibration of quadpolarization SAR data using backscattering statistics," Proc. 1989, Intern. Geosci. Remote Sens. Symp., Vancouver, Canada, July 1989.

[3] Sarabandi, K., F.T. Ulaby, and M.A. Tassoudji, "Calibration of polarimetric radar systems with good polarization isolation," accepted for publication in IEEE Trans. Geosci. and Remote Sens., Sept., 1989.

[4] Sheen, D.R. and E.S. Kasischke,"Comparison of SAR polarimetric calibration technique using clutter," Proc. 1989, Intern. Geosci. Remote Sens. Symp., Vancouver, Canada, July 1989.

[5] Ulaby, F.T., P.F. Polatin, M.W. Whitt, V.V. Liepa, "A general polarimetric radar calibration technique: Theory and experiment, Part I- Theory," submitted to IEEE Trans. Antennas Propagat., July 1989.

[6] van Zyl, J. J., " Calibration of polarimetric radar images using only image parameters and trihedral corner reflector responses," submitted to IEEE Trans. Geosci. and Remote Sens., Jan., 1989.

[7] Whitt, M.W. and F.T. Ulaby, "A general polarimetric radar calibration technique: Theory and experiment, Part I- Theory," submitted to IEEE Trans. Antennas Propagat., July 1989.

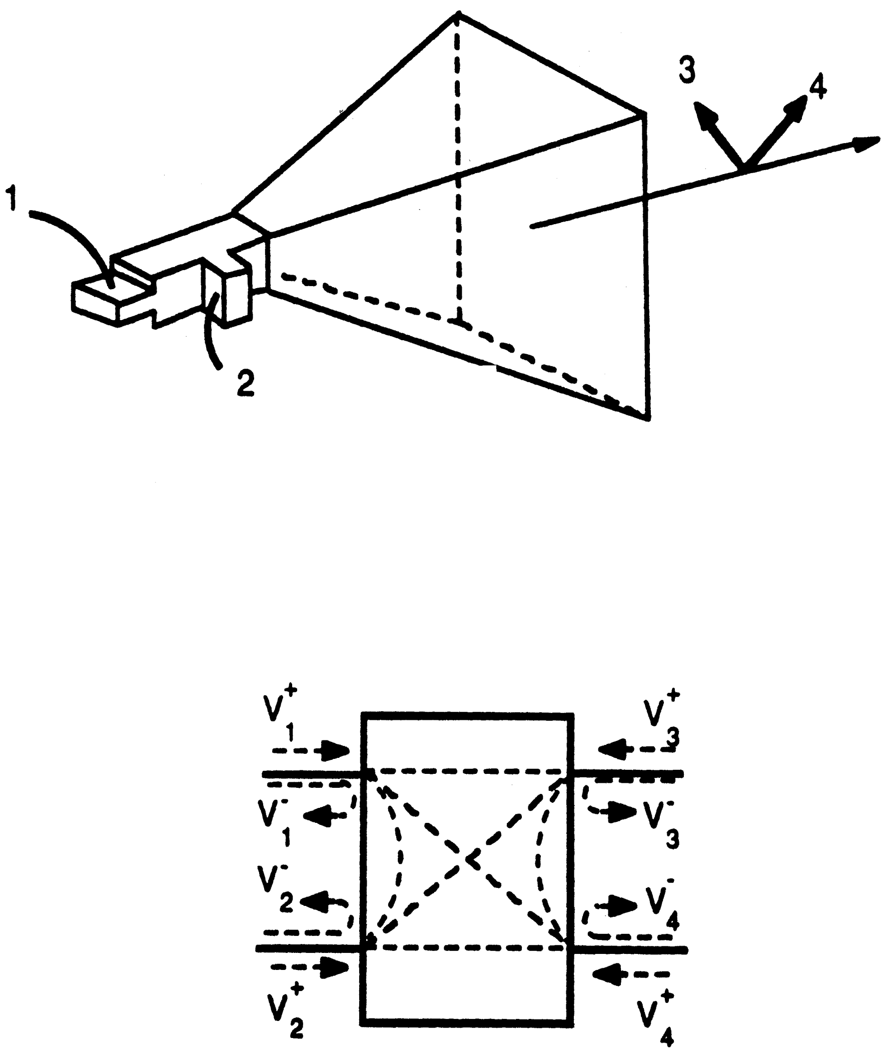

Figure A-1: Antenna system and its equivalent circuit four-port representation.

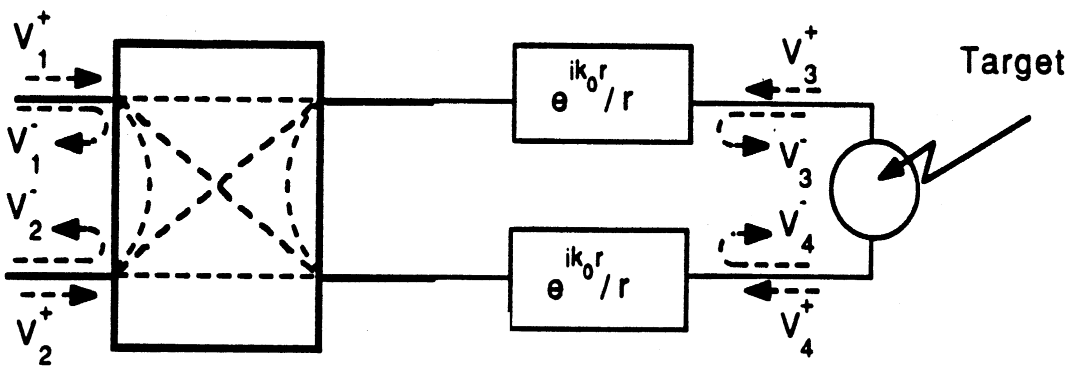

Figure A-2: Signal flow-chart of the antenna system, free space, and the target.

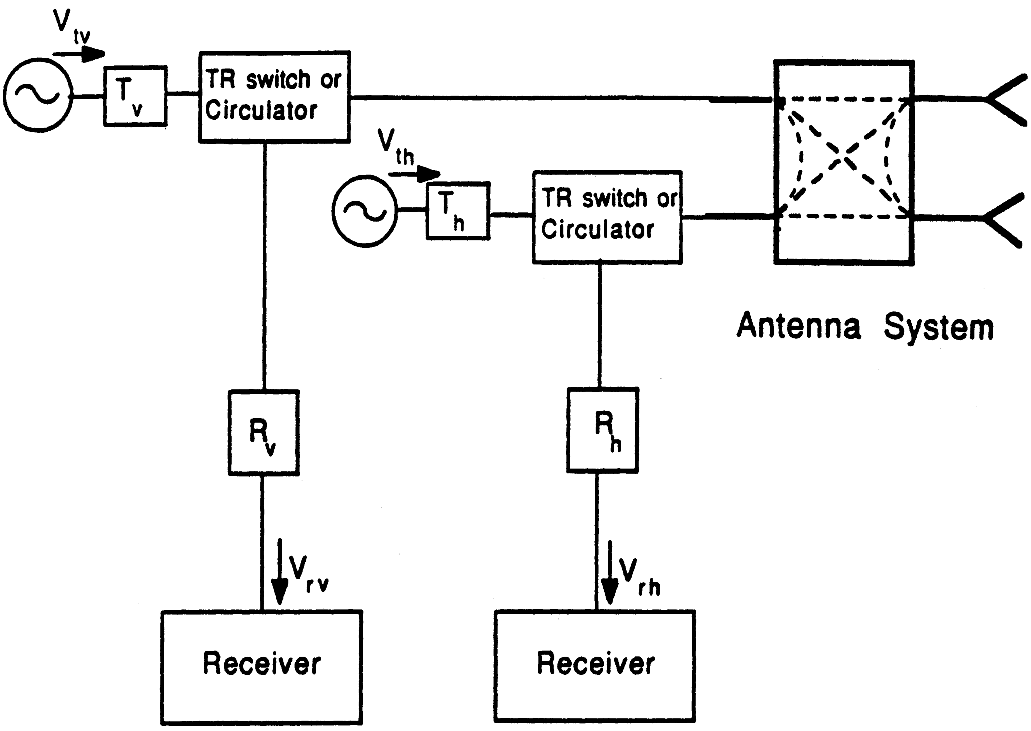

Figure A-3: Simplified block diagram of a typical polarimetric radar.

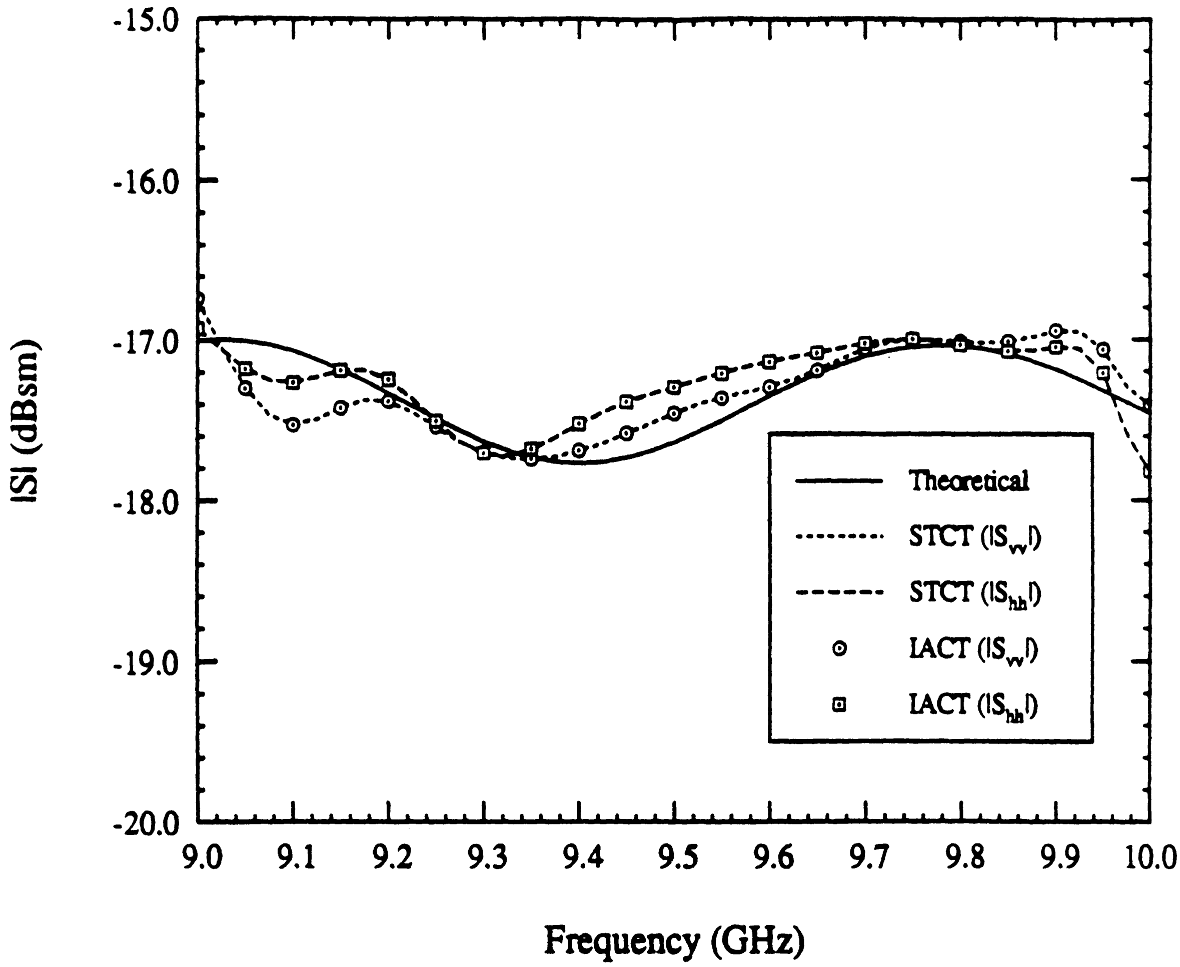

Figure A-4: Magnitude of the diagonal elements of the scattering matrix of a 6-inch sphere.

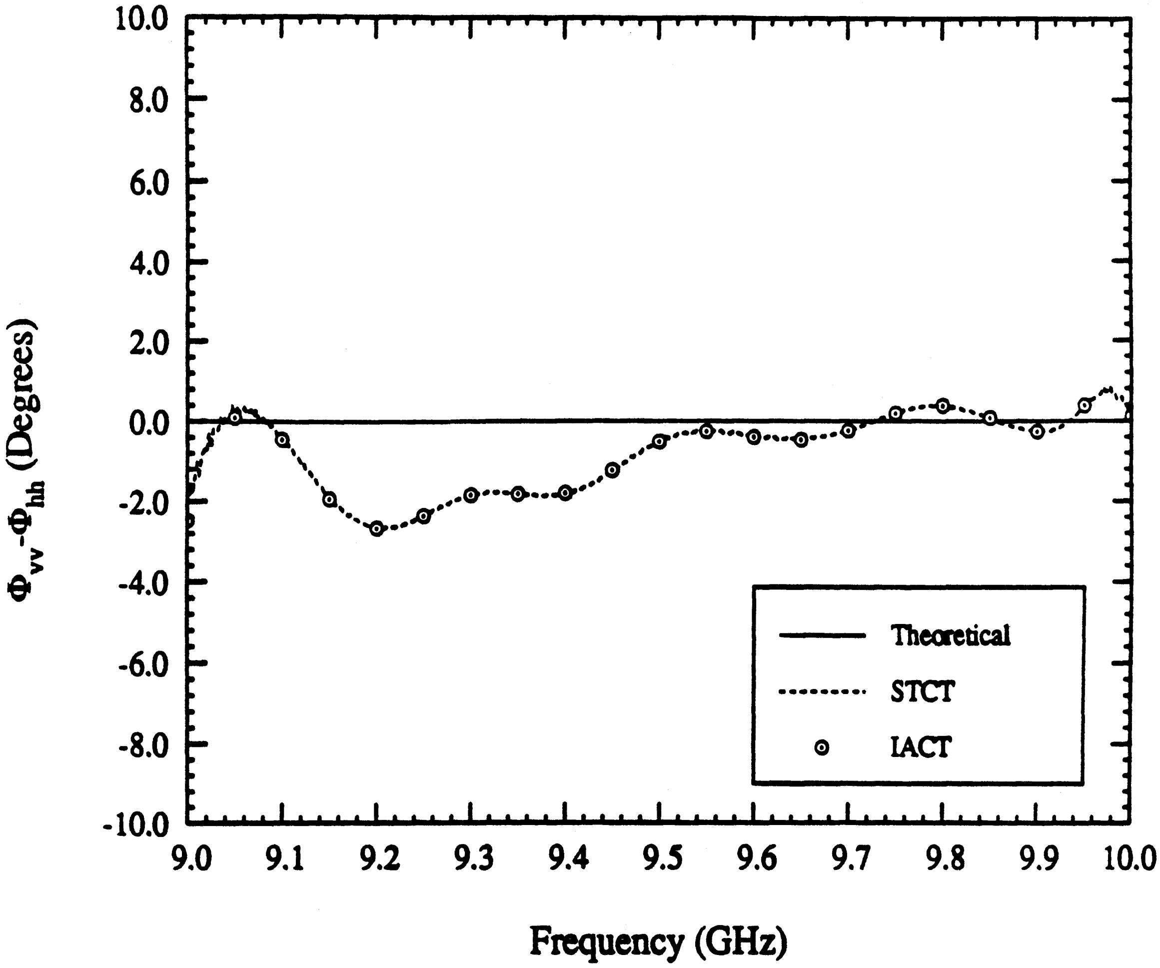

Figure A-5: Phase difference between the diagonal elements of the scattering matrix of a 6-inch sphere.

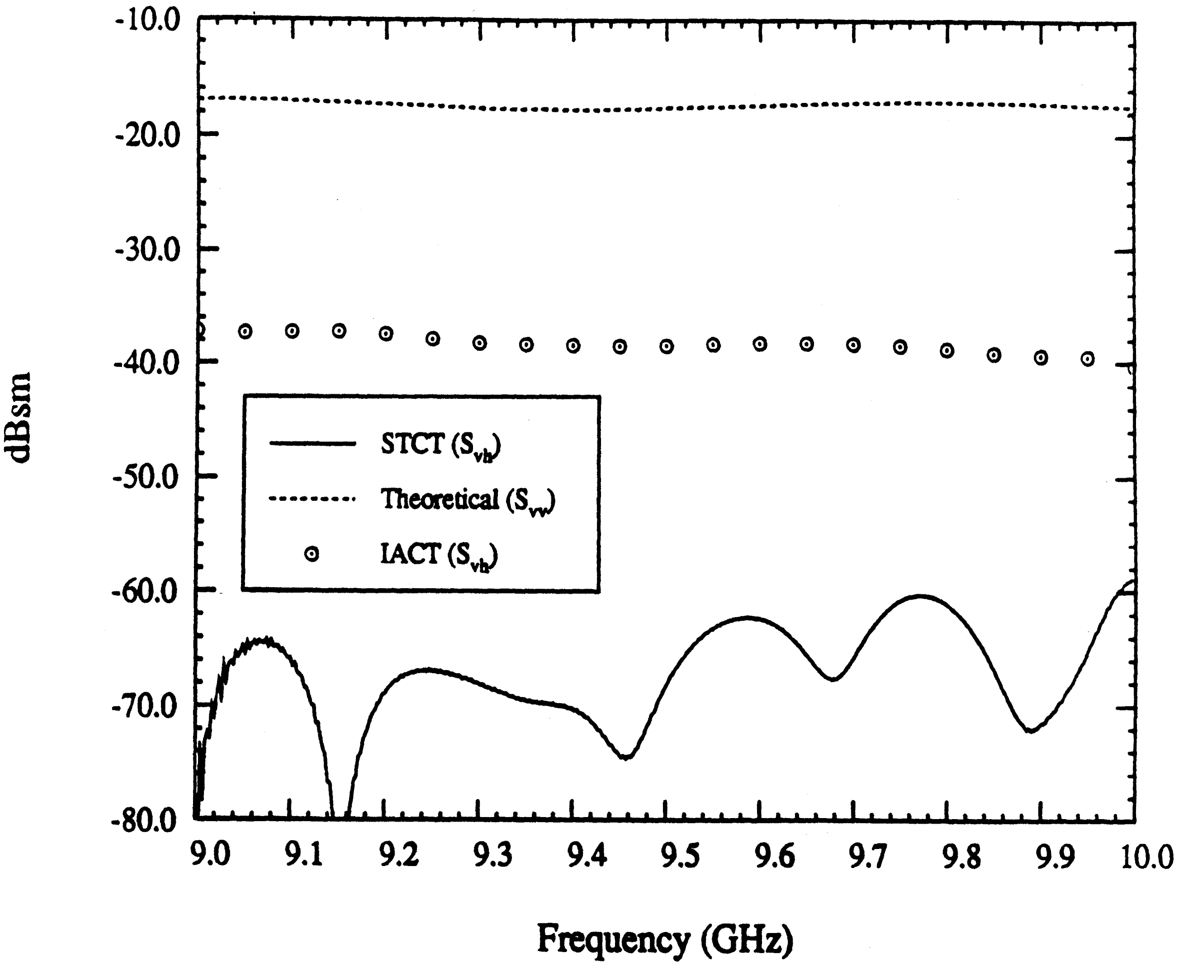

Figure A-6: Magnitude of the off-diagonal element of the scattering matrix of a 6-inch sphere compared with one of the diagonal elements.

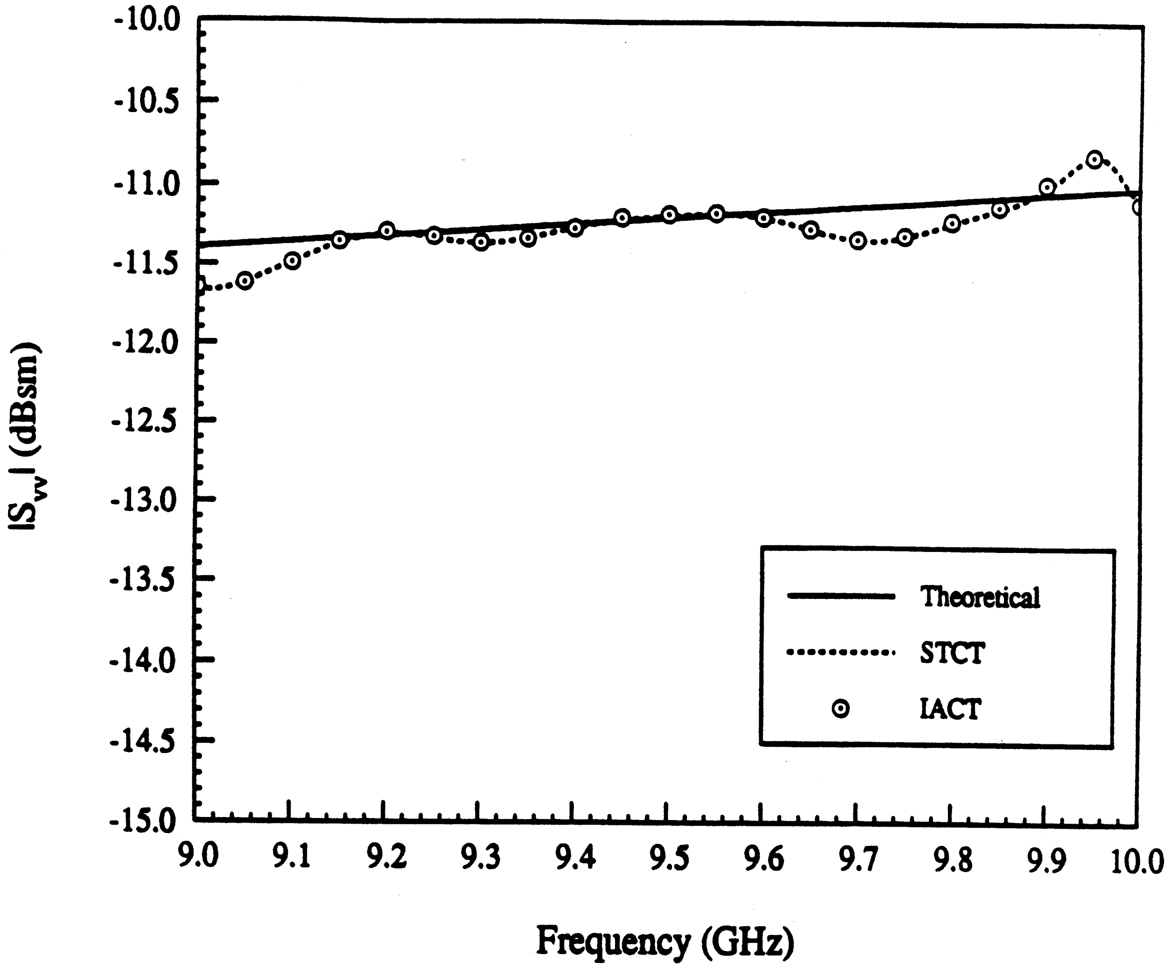

Figure A-7: Magnitude of the diagonal element (s_{vv}) of the scattering matrix of the vertical cylinder.

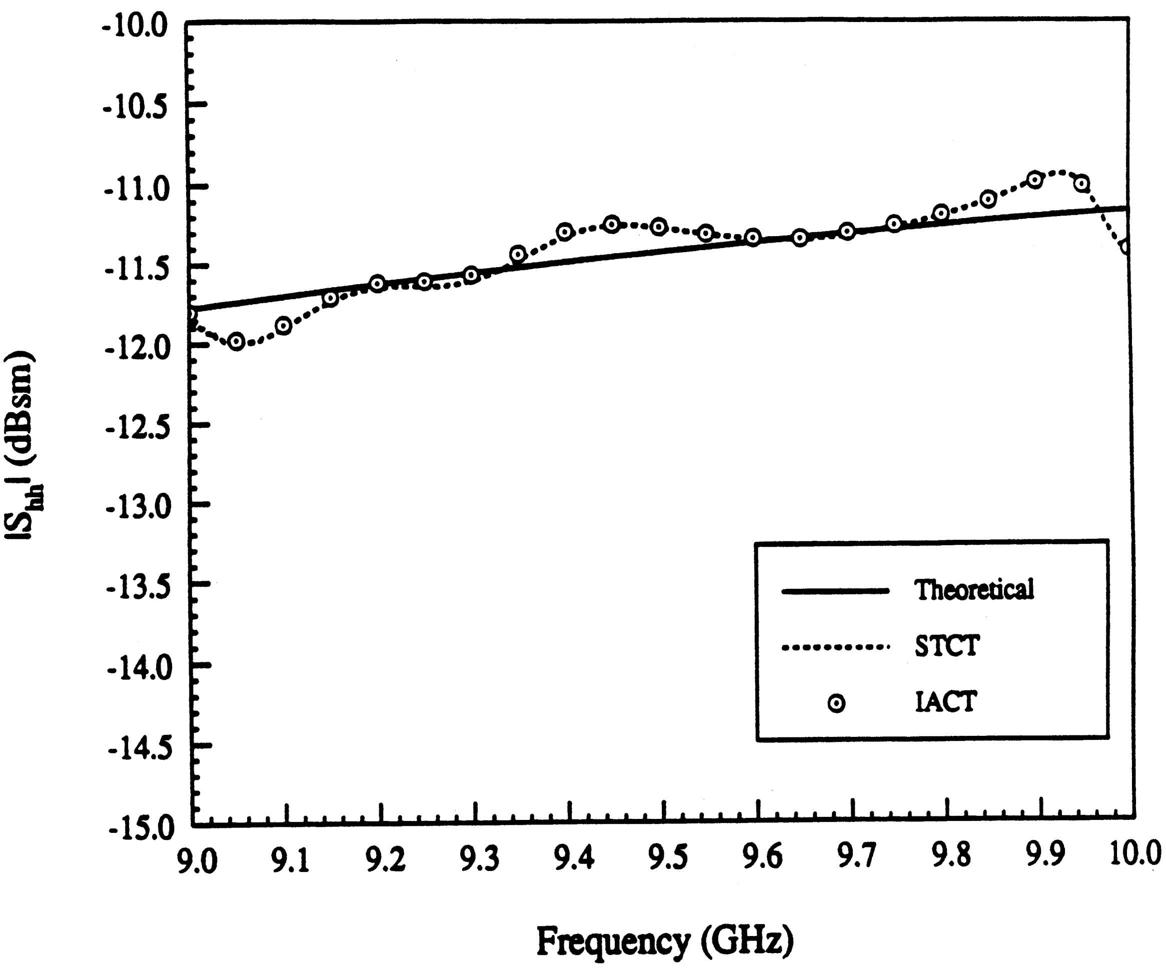

Figure A-8: Magnitude of the diagonal element (s_{hh}) of the scattering matrix of the vertical cylinder.

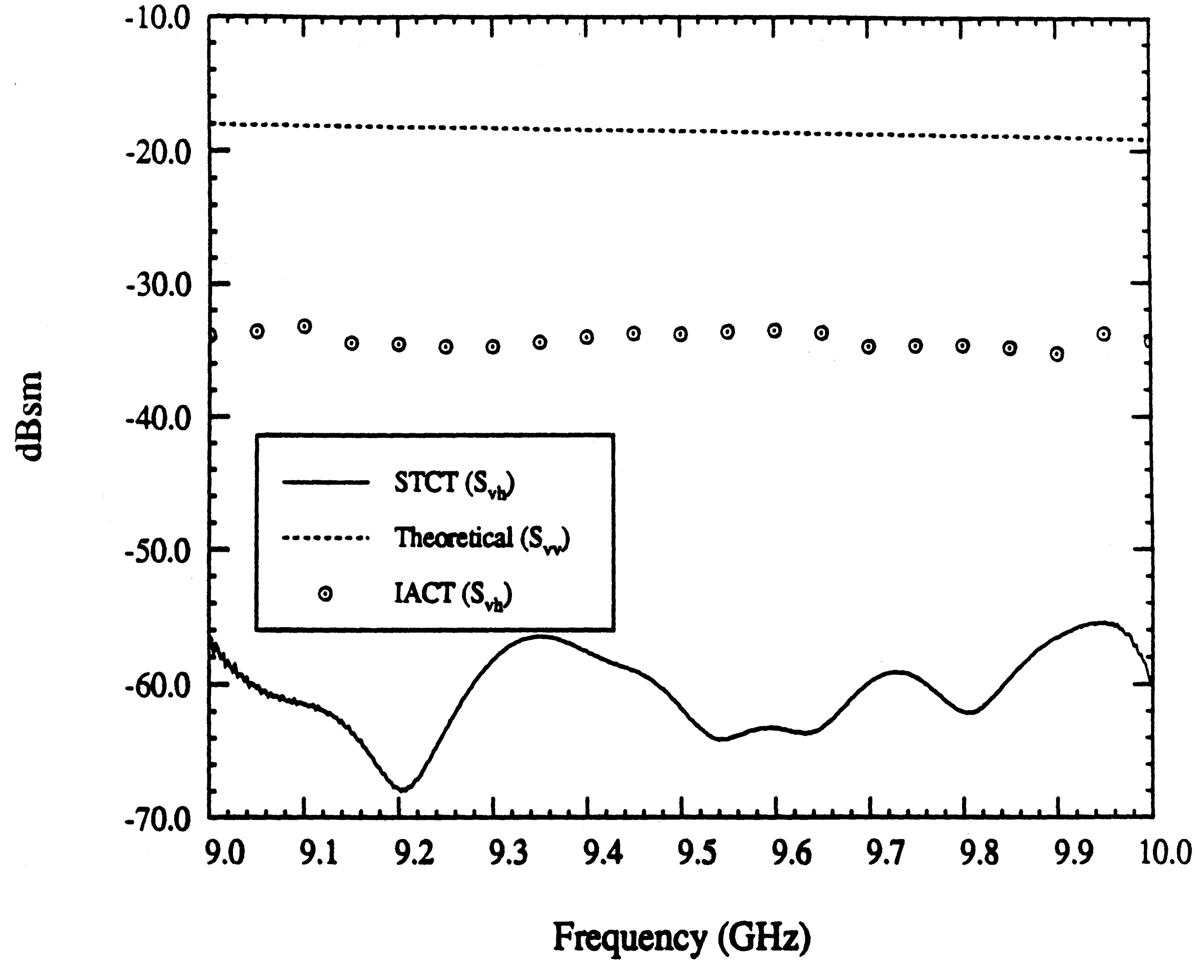

Figure A-9: Magnitude of the off-diagonal element (s_{vh}) of the scattering matrix of the vertical cylinder compared with s_{vv}.

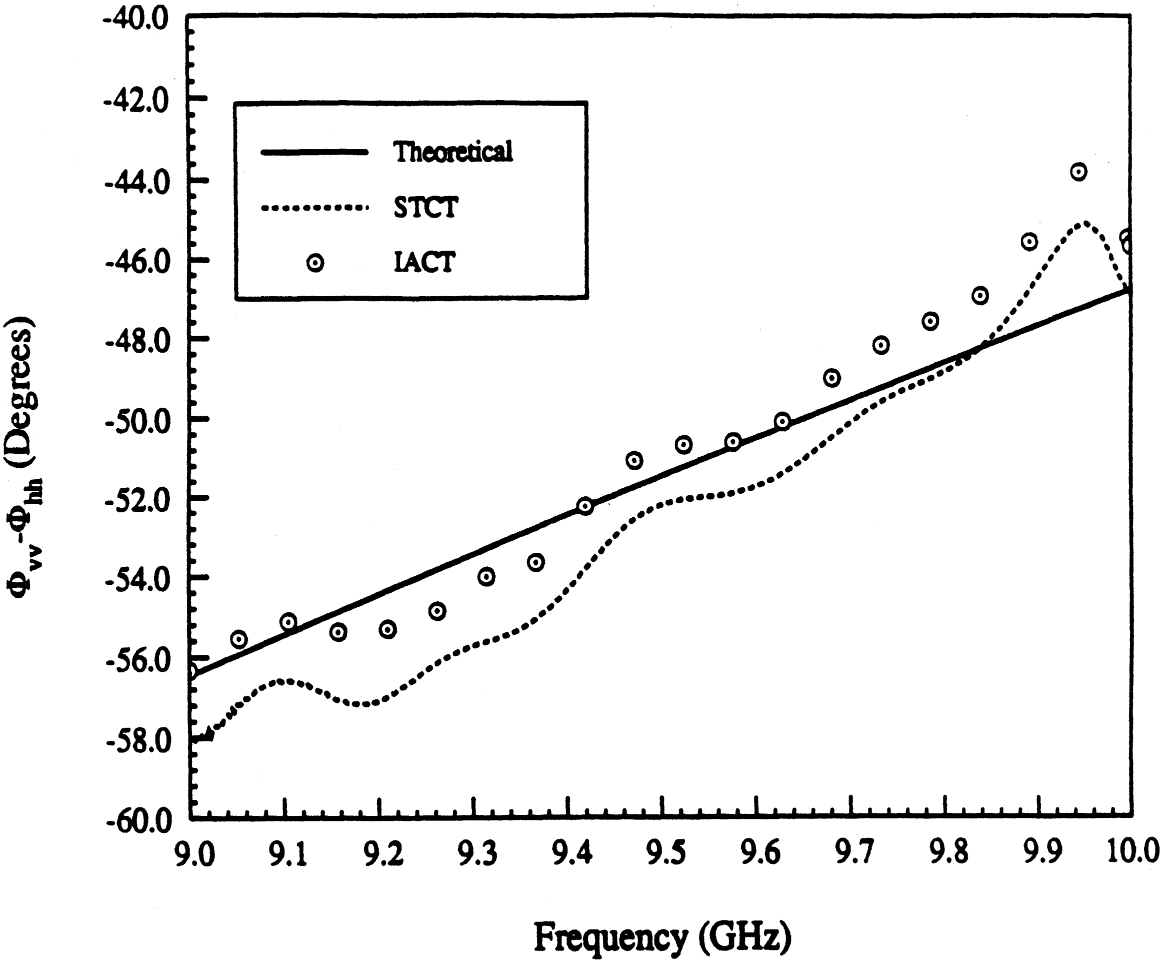

Figure A-10: Phase difference between the diagonal elements of the scattering matrix of the vertical cylinder.

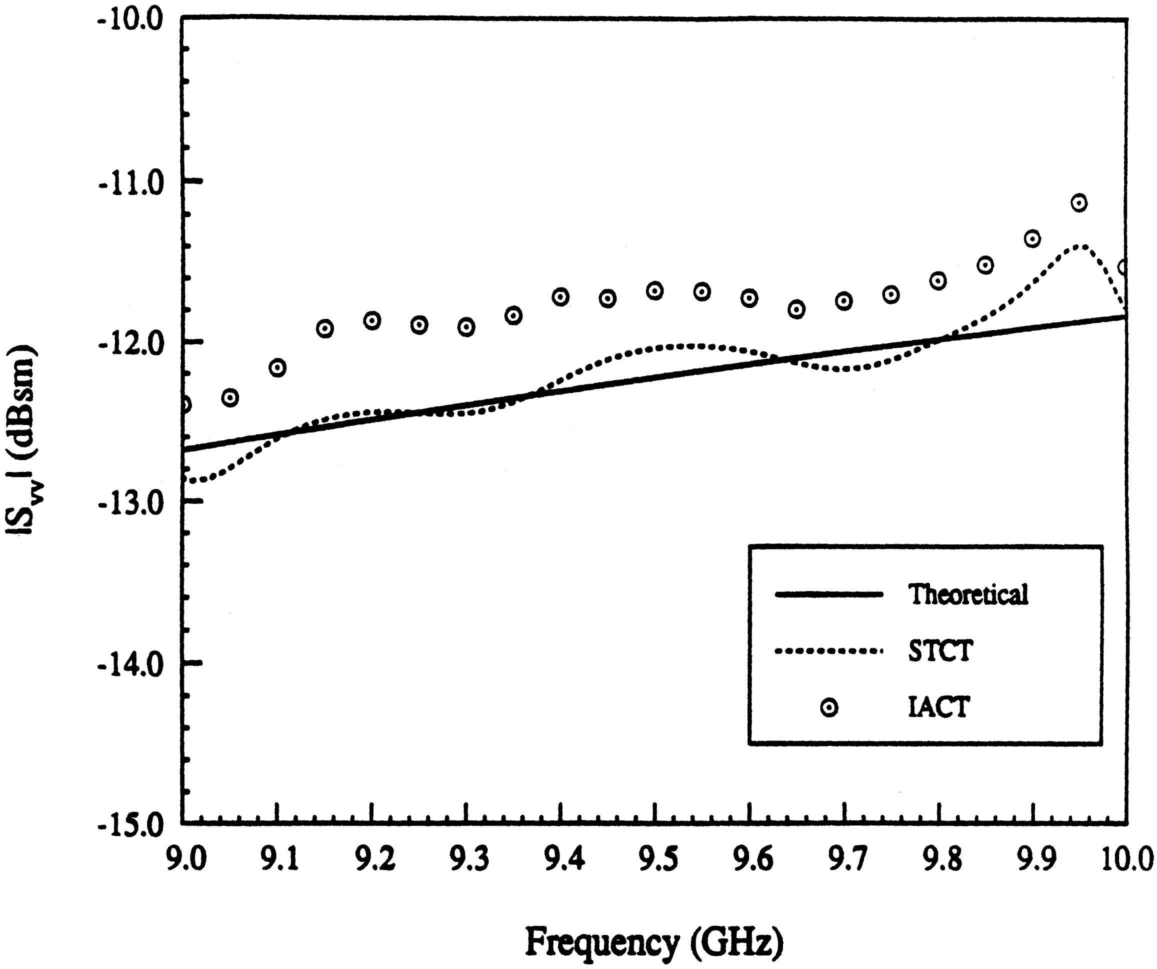

Figure A-11: Magnitude of the diagonal element (s_{vv}) of the scattering matrix of the 45° tilted cylinder.

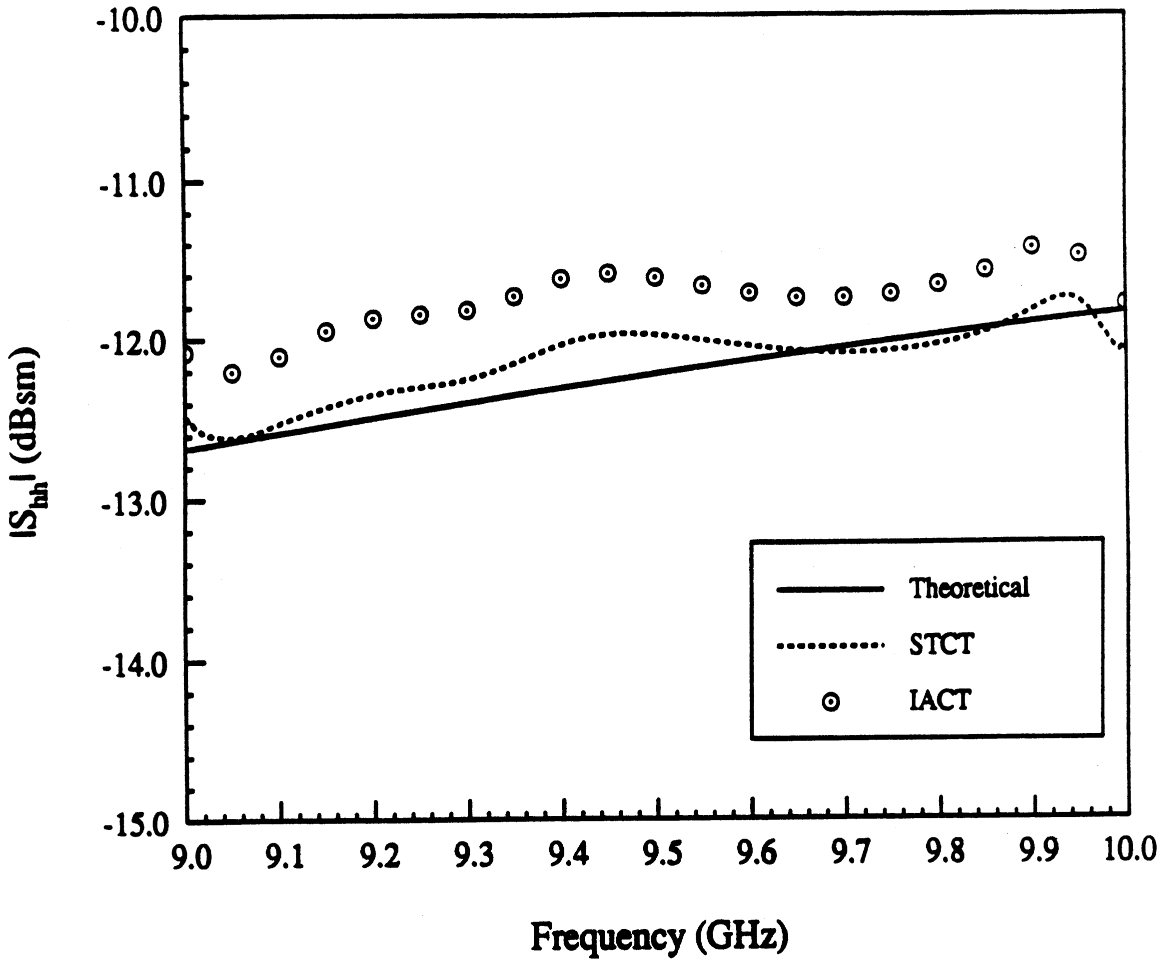

Figure A-12: Magnitude of the diagonal element (s_{hh}) of the scattering matrix of the 45° tilted cylinder.

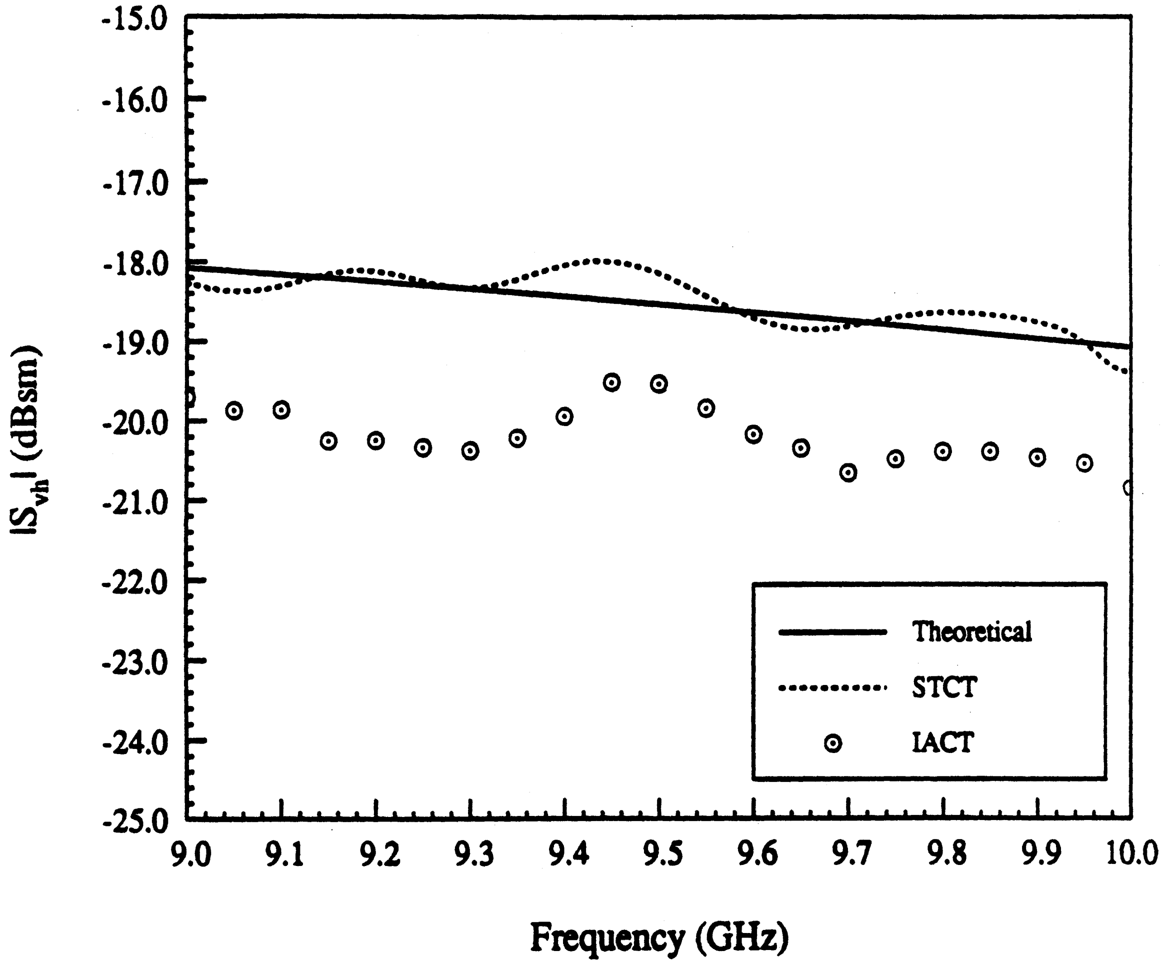

Figure A-13: Magnitude of the off-diagonal element (s_{vh}) of the scattering matrix of the 45° tilted cylinder.

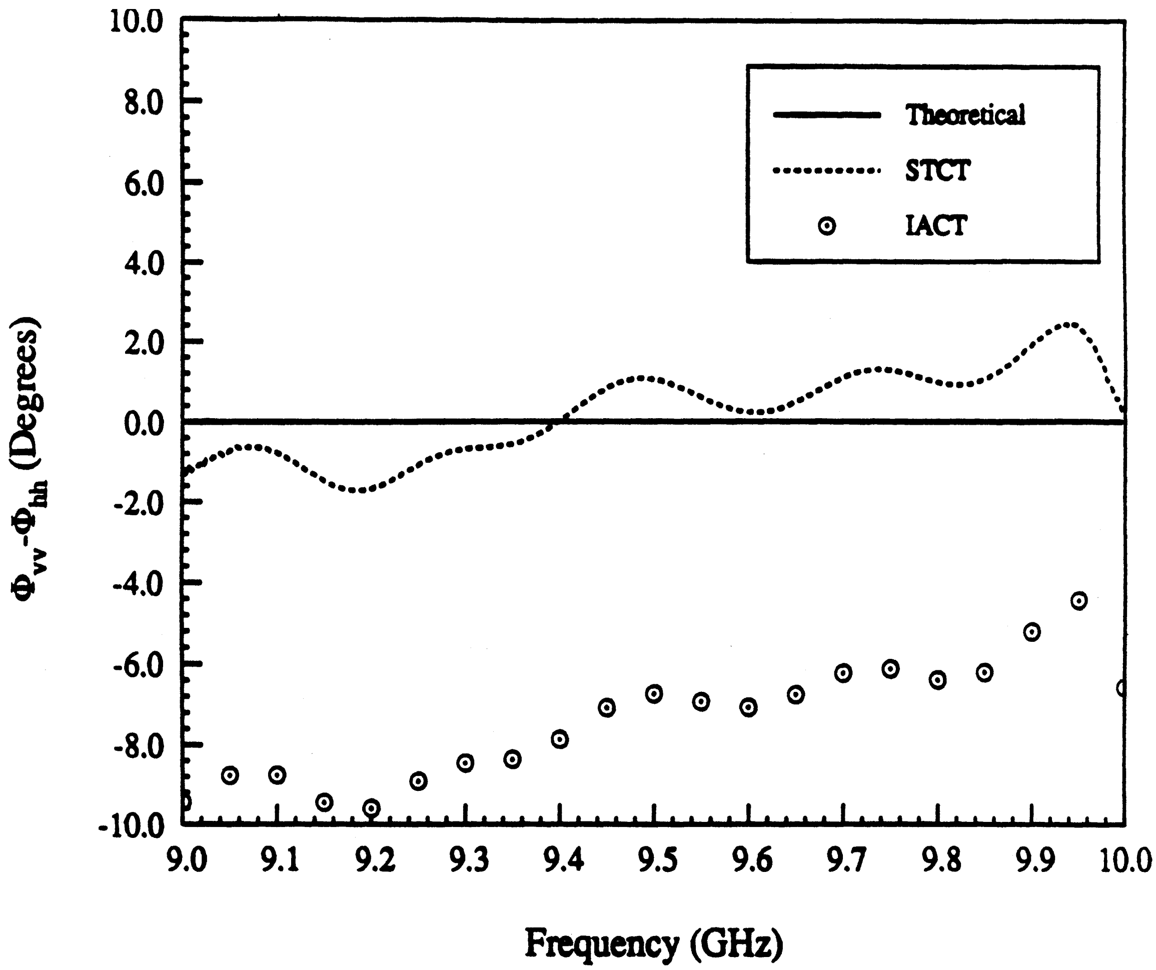

Figure A-14: Phase difference between the diagonal elements of the scattering matrix of the 45° tilted cylinder.

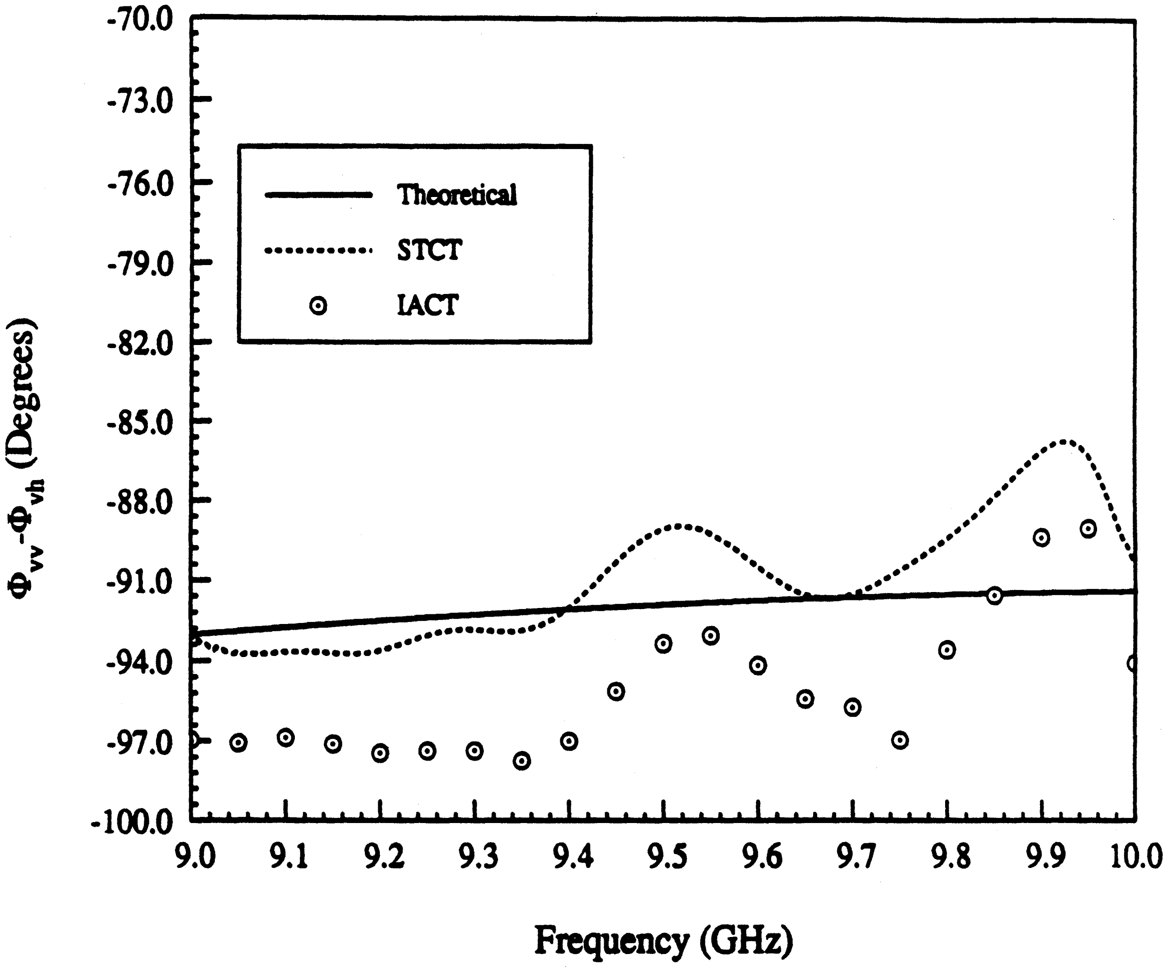

Figure A-15: Phase difference between the diagonal and the off-diagonal elements of the scattering matrix of the 45° tilted cylinder.

APPENDIX B

POLARIMETRIC MEASUREMENT PROGRAM

This program is written in HP BASIC to control the network analyzer based scatterometer (see Fig. (1)) and a stepping motor controller to acquire polarimetric measurement data of the PARCs automatically.

The main program is comprised of two major functions for calibration: 1) target measurement and 2) reference target measurement. For each measurement, the frequency band for L- and C-band is selected and the time gating parameters, i.e., the gate center and the gate span, are selected by the operator prior to data acquisition. The target measurement starts with selection of target rotation parameters such as start angle, increment angle, stop angle and rotation speed by calling the subroutine "Rotate_target". The subroutine "Rotate_target" calls the other subroutine 'comm' to communicate with a stepping motor controller through a serial port of an HP 9000 series computer. The Fourier transform of the time gated data (in magnitude and phase) of the four polarizations (VV, HH, HV, VH) are collected for each incidence angle and for the selected frequency points. The reference measurement for calibration utilizes the averaging technique to reduce the effect of thermal noise and background subtraction to increase S/N.

At the end of the measurement, the main program calls a subroutine "Store_file" to transfer the collected data to a disk drive with a title of the data file designated by a subroutine "Set_clock" and "Store_file". The subroutines of "Config_and_poll", "Hp_bus_init" and "Series_init" are to configure the system including a HP network analyzer and the HP-IB bus connections. The subroutines of "Freq_set", "Freq_sw" and "Pol_sw" are to control the network analyzer and the relay actuator for the selection of frequencies and polarizations.

```
10     !*******************************************************************
20     !   L/C/X POLARIMETER MEASUREMENT PROGRAM
30     !       FILE: PARC_1
40     !*******************************************************************
50     !   LAST EDIT: May 21, 1990      JPL PARC measurements
60     !
70     !*******************************************************************
80     OPTION BASE 1
90     COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
100    COM /Constants/ Vel,Zero(3),Exec_key$[2]
110    COM /System_config/ INTEGER Printer_flag,Debug_flag,Version$[12],Mode$[10]
                ,Out_type$[10],Sound$[3],Bell$[1],Target$[30],Ref_target$[30]
120    COM /Sys_1/ Freq$(3)[1],Freq_cent(3),Freq_span(3),Gate_cent(3),Gate_span(3)
130    COM /Sys_2/ Pol$(4)[2],Polsw$(3,4)[8]
140    COM /Sys_3/ INTEGER F_disp,P_disp
150    COM /Sys_4/ Drive_a$[15],Drive_b$[15],Drive_c$[15],INTEGER Preamble,Bytes
160    COM /Sys_5/ INTEGER Nskip,Ndata
170    COM /Sys_6/ Ref_angle,Angle,Angle$[10],Beam(3),INTEGER Npts,Ntrace
                ,Average_factor
180    COM /Sys_7/ INTEGER Meas_flag(3)
190    COM /Com4/ INTEGER Rotation_state,REAL Inc_angle,Current_angle,Start_angle,
               Stop_angle,Old_home_angle,INTEGER Sets_per_pos
200    COM /Status/ INTEGER Sc,Connect_flg,E_flg,Debug_flg,Response$[80]
210    !
220    !
230    INTEGER F,I,J,P,T,Meas_flag_old(3),Exit_flag,Nt,Nst,Nskh,Npt
240    DIM Sky_cal_file$(3)[14],Old_target_name$[30]
250    DATA "L","C","X"                                      ! FREQUENCY
260    DATA "VV","HH","HV","VH"                              ! POLARIZATION
270    DATA 1.25,1.25,1.5                                     ! FREQ_CENT
280    DATA .3,.5,.5                                         ! FREQ_SPAN
290    DATA 12.5,9.0,6.2                                     ! BEAMWIDTH
300    DATA "?*B3456","?*A56B34","?*A6B345","?*A5B346"       ! L
310    DATA "?*B3456","?*A34B56","?*A4B356","?*A3B456"       ! C
320    DATA "?*A34B56","?*B3456","?*A3B456","?*A4B356"       ! X
330    DATA ":,700,0",":,700,1",":MEMORY,0,7"                ! DRIVE_A,B,C
340    DATA 400E-9,400E-9,400E-9                             ! GATE CENTERS
350    DATA 10E-9,10E-9,10E-9                                ! GATE SPANS
360    READ Freq$(*)
370    READ Pol$(*)
380    READ Freq_cent(*)
390    READ Freq_span(*)
400    READ Beam(*)
410    READ Polsw$(*)
420    READ Drive_a$,Drive_b$,Drive_c$
430    READ Gate_cent(*),Gate_span(*)
440    !
450    ! Set up error handling routine.
460    !
470    LOAD KEY "NOKEY:MEMORY,0,1"
480    MASS STORAGE IS ":,700,1"
490    !
500    !
510    ! Initialize important parameters.
520    !
530    DEG
540    Rotation_state=-1
550    Current_angle=0.
560    MAT Meas_flag= (1)
570    Mode$="FAST ACQ"
580    F_disp=1
```

```
590   P_disp=1
600   Printer_flag=0
610   !
620   Hp_bus_init
630   IF Printer_flag=1 THEN Out_type$="PRINT/DISC"
640   !
650   Vel=2.99792458E+8
660   Ntrace=1
670   Npts=401
680   Nskip=40
690   Ndata=10
700   Average_factor=1
710   Angle$="0"
720   Angle=0
730   Ref_angle=0
740   Target$=""
750   Sound$="ON "
760   Debug_flag=0
770   Bell$=CHR$(7)
780   Exec_key$=CHR$(255)&CHR$(88)
790   Version$="Version 8.0 "
800   Exit_flag=0
810   Print_banner1
820   !
830   System_memory=VAL(SYSTEM$("AVAILABLE MEMORY"))
840   IF FNAsk("INITIALIZE RAM DISK?") THEN
850      INITIALIZE Drive_c$,0
860      INITIALIZE Drive_c$,INT((System_memory)/512)
870   ELSE
880      ASSIGN @Is_it_there TO Drive_c$;RETURN Outcome
890      IF Outcome=0 THEN
900         CAT Drive_c$;NO HEADER,COUNT Entries
910         IF Entries=0 THEN INITIALIZE Drive_c$,0
920      END IF
930      ASSIGN @Is_it_there TO *
940   END IF
950   !
960   Config_and_poll
970   OUTPUT @Relay;"?*B1256A34"
980   Series_init
990   !
1000  !
1010 Start_loop: !
1020            !
1030     Print_banner4
1040        ON KEY 0 LABEL "   SKY CAL    ",FNTrap_level GOSUB Sky_calibrate
1050        ON KEY 1 LABEL " REFERENCE CAL",FNTrap_level GOSUB Ref_target
1060        ON KEY 2 LABEL "  TARGET RUN  ",FNTrap_level GOSUB Acq_target
1070        ON KEY 3 LABEL " SET FREQUENCY",FNTrap_level GOSUB Freq_set
1080        ON KEY 4 LABEL "    ANGLE     ",FNTrap_level GOSUB Set_angle
1090        ON KEY 5 LABEL " TARGET NAME  ",FNTrap_level GOSUB Set_target
1100        ON KEY 6 LABEL " # OF TRACES  ",FNTrap_level GOSUB Set_traces
1110        ON KEY 7 LABEL " # OF POINTS  ",FNTrap_level GOSUB Set_points
1120        ON KEY 8 LABEL "# OF AVERAGES ",FNTrap_level GOSUB Set_average
1130        ON KEY 9 LABEL "     QUIT     ",FNTrap_level GOTO Quit_fast_acq
1140     GOSUB Allocate_matrix
1150     LOOP
1160        EXIT IF Exit_flag=1
1170     END LOOP
1180     GOSUB Deallocate_mtrx
1190     Exit_flag=0
```

```
1200      GOTO Start_loop
1210      !
1220 Null: RETURN
1230 !
1240 !------------------------------------------------------------------
1250 !
1260 Sky_calibrate:  ! Acquire a sky calibration data set.
1270                 !
1280      OFF KEY
1290      Clear_crt
1300      OUTPUT @Nwa;"TIMDTRANON; LOGM; GATEOFF;"
1310      OUTPUT @Nwa;"STAR 0NS; STOP 400NS;"
1320      PRINT TABXY(1,10);"Please point scatterometer assembly to free space."
1330      PRINT TABXY(1,12);"Press CONTINUE when ready..."
1340      PAUSE
1350      Clear_crt
1360      PRINT TABXY(1,14);"Sky data is being collected ... "
1370      OUTPUT @Nwa;"TIMDTRANOFF; POLA; AVERFACT";VAL$(Average_factor);";"
1380      OUTPUT @Nwa;"GATEOFF; AVEROON;"
1390      !
1400      ! Get a free space (sky) measurement.
1410      !
1420      FOR T=1 TO Ntrace
1430      FOR F=1 TO 3
1440         IF Meas_flag(F)=1 THEN
1450            IF F=1 THEN
1460               OUTPUT @Nwa;"POWE20;"
1470            ELSE
1480               OUTPUT @Nwa;"POWE10;"
1490            END IF
1500            Freq_set(F)
1510            Freq_sw(F)
1520                    OUTPUT @Nwa;"GATECENT";VAL$(Gate_cent(F));"S;"
1530                    OUTPUT @Nwa;"GATESPAN";VAL$(Gate_span(F));"S;"
1540                FOR P=1 TO 4
1550                    Pol_sw(F,P)
1560                    OUTPUT @Nwa;"FORM3; NUMG";VAL$(Average_factor+1);"; WAIT;"
1570                    OUTPUT @Nwa;"WAIT; GATEOON; WAIT; OUTPFORM;"
1580                    ENTER @Nwa_data1;Preamble,Bytes,Trace(*)
1590                    MAT Target_response(F,P,*)= Trace
1600                    OUTPUT @Nwa;" GATEOFF; WAIT;"
1610                NEXT P
1620                          Nskh=Nskip+1
1630                FOR P=1 TO 4
1640                          FOR Nt=Nskh TO Npts STEP Nskip
1650                             Nst=1+INT(Nt/Nskip)
1660                             Target_data(F,T,P,Nst)=Target_response(F,P,Nt)
1670                          NEXT Nt
1680                NEXT P
1690         END IF
1700      NEXT F
1710      NEXT T
1720      Old_target_name$=Target$
1730      Ref_target$="SKY CALIBRATION"
1740      Store_file(Target_data(*),"SKY",FNTime_stamp$,F)
1750      Ref_target$=Old_target_name$
1760      Pol_sw(F_disp,P_disp)
1770      DISP "Sky calibration data saved."
1780      Exit_flag=1
1790 RETURN
1800 !
```

```
1810 !----------------------------------------------------------------------
1820 !
1830 Ref_target: ! Acquire a reference target data set.
1840            !
1850      OFF KEY
1860      Clear_crt
1870      OUTPUT @Nwa;"TIMDTRANON; LOGM; GATEOFF;"
1880      OUTPUT @Nwa;"LOGM; AUTO; STAR 0NS; STOP 400NS;"
1890      PRINT TABXY(1,10);"Please point scatterometer assembly to reference  targ
1900      PRINT TABXY(1,12);"Press CONTINUE when ready..."
1910      PAUSE
1920      GOSUB Set_gates
1930      OUTPUT @Nwa;"TIMDTRANOFF; POLA; AVERFACT";VAL$(Average_factor);";"
1940      OUTPUT @Nwa;"GATEOFF; AVEROON"
1950      INPUT "Enter the reference target angle: ",Ref_angle
1960      !
1970      ! Get the reference target response.
1980      !
1990      FOR T=1 TO Ntrace
2000      FOR F=1 TO 3
2010          IF Meas_flag(F) THEN
2020             IF F=1 THEN
2030                OUTPUT @Nwa;"POWE20;"
2040             ELSE
2050                OUTPUT @Nwa;"POWE10;"
2060             END IF
2070             Freq_set(F)
2080             Freq_sw(F)
2090                 OUTPUT @Nwa;"GATECENT";VAL$(Gate_cent(F));"S;"
2100                 OUTPUT @Nwa;"GATESPAN";VAL$(Gate_span(F));"S;"
2110             FOR P=1 TO 4
2120                 Pol_sw(F,P)
2130                 OUTPUT @Nwa;"FORM3;NUMG";VAL$(Average_factor+1);";WAIT"
2140                 OUTPUT @Nwa;"WAIT; GATEOON; WAIT; OUTPFORM;"
2150                 ENTER @Nwa_data1;Preamble,Bytes,Trace(*)
2160                 MAT Target_response(F,P,*)= Trace
2170                 OUTPUT @Nwa;"GATEOFF;"
2180             NEXT P
2190                    Nskh=Nskip+1
2200             FOR P=1 TO 4
2210                    FOR Nt=Nskh TO Npts STEP Nskip
2220                        Nst=INT(Nt/Nskip)
2230                        Target_data(F,T,P,Nst)=Target_response(F,P,Nt)
2240                    NEXT Nt
2250             NEXT P
2260          END IF
2270      NEXT F
2280      NEXT T
2290      Store_file(Target_data(*),"REF",FNTime_stamp$,F)
2300      !
2310      ! Get the reference target mount response.
2320      !
2330      BEEP
2340      PRINT TABXY(1,10);"Please remove the reference target from its mount."
2350      PRINT TABXY(1,12);"Press CONTINUE when ready..."
2360      PAUSE
2370      Clear_crt
2380      PRINT TABXY(1,14);"Data for the mount is being collected .... "
2390      FOR T=1 TO Ntrace
2400      FOR F=1 TO 3
2410          IF Meas_flag(F) THEN
```

```
2420              IF F=1 THEN
2430                 OUTPUT @Nwa;"POWE20;"
2440              ELSE
2450                 OUTPUT @Nwa;"POWE10;"
2460              END IF
2470              Freq_set(F)
2480              Freq_sw(F)
2490                  OUTPUT @Nwa;"GATECENT";VAL$(Gate_cent(F));"S;"
2500                  OUTPUT @Nwa;"GATESPAN";VAL$(Gate_span(F));"S;"
2510              FOR P=1 TO 4
2520                  Pol_sw(F,P)
2530                  OUTPUT @Nwa;"FORM3;NUMG";VAL$(Average_factor+1);";WAIT"
2540                  OUTPUT @Nwa;"WAIT; GATEOON; WAIT; OUTPFORM;"
2550                  ENTER @Nwa_data1;Preamble,Bytes,Trace(*)
2560                  MAT Target_response(F,P,*)= Trace
2570                  OUTPUT @Nwa;"GATEOFF;"
2580              NEXT P
2590                     Nskh=Nskip+1
2600              FOR P=1 TO 4
2610                     FOR Nt=Nskh TO Npts STEP Nskip
2620                         Nst=INT(Nt/Nskip)
2630                         Target_data(F,T,P,Nst)=Target_response(F,P,Nt)
2640                     NEXT Nt
2650              NEXT P
2660           END IF
2670       NEXT F
2680       NEXT T
2690       Store_file(Target_data(*),"MNT",FNTime_stamp$,F)
2700       Pol_sw(F_disp,P_disp)
2710       DISP "Reference target mount response saved."
2720       Exit_flag=1
2730 RETURN
2740 Acq_target: !
2750              !
2760       OFF KEY
2770       Clear_crt
2780       OUTPUT @Nwa;"TIMDTRANON; LOGM; GATEOFF; AVEROFF;"
2790       OUTPUT @Nwa;"STAR 0NS; STOP 400NS;"
2800       PRINT TABXY(1,10);"Please point scatterometer assembly at surface target."
2810       PRINT TABXY(1,12);"Press CONTINUE when ready..."
2820       PAUSE
2830       GOSUB Set_gates
2840       OUTPUT @Nwa;"TIMDTRANOFF; WAIT; GATEOON; POLA;"
2850       !
2860       ! Get the target response.
2870       !
2880       FOR T=1 TO Ntrace
2890       !
2900       ! Get angles
2910       !
2920       IF T=1 THEN
2930          Rotation_state=-1
2940       ELSE
2950          Rotation_state=2
2960       END IF
2970       SELECT Rotation_state
2980       CASE =0
2990            Clear_crt
3000            PRINT TABXY(1,4);"When ready for measurement, press CONTINUE."
3010            BEEP
3020            PAUSE
```

```
3030            Clear_crt(3,16)
3040            PRINT TABXY(1,4);"Collecting data... "
3050       CASE ELSE
3060            PRINT TABXY(1,4);"Current angle is ";Current_angle;" degrees."
3070            Rotate_target    !
3080            WAIT 1
3090            Clear_crt(3,16)
3100            PRINT TABXY(1,4);"Collecting data ..."
3110       END SELECT
3120           FOR F=1 TO 3
3130               IF Meas_flag(F) THEN
3140                  IF F=1 THEN
3150                     OUTPUT @Nwa;"POWE20;"
3160                  ELSE
3170                     OUTPUT @Nwa;"POWE10;"
3180                  END IF
3190                  Freq_set(F)
3200                  Freq_sw(F)
3210                      OUTPUT @Nwa;"GATECENT";VAL$(Gate_cent(F));"S;"
3220                      OUTPUT @Nwa;"GATESPAN";VAL$(Gate_span(F));"S;"
3230                      OUTPUT @Nwa;"GATEON; WAIT;"
3240                  FOR P=1 TO 4
3250                      Pol_sw(F,P)
3260                      OUTPUT @Nwa;"SING; FORM3; OUTPFORM;"
3270                      ENTER @Nwa_data1;Preamble,Bytes,Trace(*)
3280                      MAT Target_response(F,P,*)= Trace
3290                  NEXT P
3300                         Nskh=Nskip+1
3310                  FOR P=1 TO 4
3320                         FOR Nt=Nskh TO Npts STEP Nskip
3330                             Nst=INT(Nt/Nskip)
3340                             Target_data(F,T,P,Nst)=Target_response(F,P,Nt)
3350                         NEXT Nt
3360                  NEXT P
3370              END IF
3380          NEXT F
3390          PRINT "# OF TRACES LEFT=",Ntrace-T
3400       NEXT T
3410       Store_file(Target_data(*),"GND",FNTime_stamp$,F)
3420       DISP "Surface target data saved."
3430       BEEP
3440       WAIT .1
3450       BEEP
3460       OUTPUT @Nwa;"CONT;"
3470       Exit_flag=1
3480 RETURN
3490 !
3500 !-----------------------------------------------------------------------
3510 !
3520 Freq_set: GOSUB Deallocate_mtrx
3530            OFF KEY
3540           MAT Meas_flag_old= Meas_flag
3550           MAT Meas_flag= (0)
3560           Exit_flag=0
3570              ON KEY 0 LABEL "    L BAND     ",FNTrap_level GOSUB Set_l
3580              ON KEY 1 LABEL "    C BAND     ",FNTrap_level GOSUB Set_c
3590              ON KEY 2 LABEL "    X BAND     ",FNTrap_level GOSUB Set_x
3600              ON KEY 4 LABEL "    STORE      ",FNTrap_level GOTO Store_band
3610              ON KEY 5 LABEL "               ",FNTrap_level GOSUB Null
3620              ON KEY 6 LABEL "               ",FNTrap_level GOSUB Null
3630              ON KEY 7 LABEL "               ",FNTrap_level GOSUB Null
```

```
3640              ON KEY 8 LABEL "                 ",FNTrap_level GOSUB Null
3650              ON KEY 9 LABEL "    CANCEL       ",FNTrap_level GOTO Cancel_band
3660           LOOP
3670              EXIT IF Exit_flag=1
3680           END LOOP
3690 RETURN
3700 Set_l:     OFF KEY 0
3710            Meas_flag(1)=1
3720            F_disp=1
3730 RETURN
3740 Set_c:     OFF KEY 1
3750            Meas_flag(2)=1
3760            F_disp=2
3770 RETURN
3780 Set_x:     OFF KEY 2
3790            Meas_flag(3)=1
3800            F_disp=3
3810 RETURN
3820 Store_band: Print_banner4
3830              Exit_flag=1
3840              GOSUB Allocate_matrix
3850 RETURN
3860 Cancel_band: !
3870               MAT Meas_flag= Meas_flag_old
3880               Exit_flag=1
3890               GOSUB Allocate_matrix
3900 RETURN
3910 !
3920 !-------------------------------------------------------------------
3930 !
3940 Set_angle:  !
3950              INPUT "Enter measurement angle: ",Angle
3960              Angle$=VAL$(Angle)&CHR$(179)&"    "    ! Degree sign.
3970              Print_banner4
3980 RETURN
3990 !
4000 !-------------------------------------------------------------------
4010 !
4020 Set_target: !
4030              LINPUT "Enter target type or name: ",Target$
4040              Target$=TRIM$(Target$)
4050              Target$=Target$&RPT$(" ",30-LEN(Target$))
4060              Print_banner4
4070 RETURN
4080 !
4090 !-------------------------------------------------------------------
4100 !
4110 Set_traces: !
4120            INPUT "Enter the number of traces (or angles) desired(>=3 ): ",Ntrace
4130            GOSUB Deallocate_mtrx
4140            GOSUB Allocate_matrix
4150            Print_banner4
4160 RETURN
4170 !
4180 !-------------------------------------------------------------------
4190 Set_points: !
4200              INPUT "Enter the number of sample points (Npts,201): ",Npts
4210              OUTPUT @Nwa;"POIN "&VAL$(Npts)&";"
4220              INPUT "Enter the data points to be stored (Ndata,10):",Ndata
4230              Nskip=INT(Npts/Ndata)
4240              Bytes=16*Ndata
```

```
4250              Print_banner4
4260              GOSUB Deallocate_mtrx
4270              GOSUB Allocate_matrix
4280 RETURN
4290 !
4300 !-----------------------------------------------------------------------
4310 !
4320 Set_average: !
4330               INPUT "Enter averaging factor: ",Average_factor
4340               Print_banner4
4350 RETURN
4360 !
4370 !-----------------------------------------------------------------------
4380 !
4390 Allocate_matrix: ! Allocate storage space for data.
4400                  !
4410      System_memory=VAL(SYSTEM$("AVAILABLE MEMORY"))
4420      Avail_traces=MIN(Ntrace,INT(System_memory-50000-3*4*16.*Npts)/(3*4*16.*N
4430      IF Avail_traces<Ntrace THEN
4440         BEEP
4450         PRINT TABXY(1,16);"Memory has capacity for only ";Avail_traces;" trace
4460         PRINT "Press CONTINUE key to continue"
4470         PAUSE
4480         Ntrace=Avail_traces
4490      END IF
4500      ALLOCATE COMPLEX Trace(Npts),Target_response(3,4,Npts)
4510      ALLOCATE COMPLEX Target_data(3,Ntrace,4,Ndata)
4520   RETURN
4530 Deallocate_mtrx: ! Return to main program.
4540                 !
4550      DEALLOCATE Target_response(*),Trace(*)
4560      DEALLOCATE Target_data(*)
4570   RETURN
4580 !
4590 !-----------------------------------------------------------------------
4600 !
4610 Set_gates: ! Set gate centers and spans.
4620            !
4630    FOR F=1 TO 3
4640        IF Meas_flag(F) THEN
4650           IF F=1 THEN
4660              OUTPUT @Nwa;"POWE20;"
4670           ELSE
4680              OUTPUT @Nwa;"POWE10;"
4690           END IF
4700           Freq_set(F)
4710           Freq_sw(F)
4720           P=1
4730               Pol_sw(F,P)
4740               OUTPUT @Nwa;"TIMDTRANON; LOGM;"
4750               OUTPUT @Nwa;"STAR 0NS; STOP 400NS; WAIT;"
4760               OUTPUT @Nwa;"MARK1; MARKMAXI;"
4770               OUTPUT @Nwa;"FORM3; OUTPACTI;"
4780               ENTER @Nwa;Gate_cent(F)
4790               OUTPUT @Nwa;"MARKOFF;"
4800               OUTPUT @Nwa;"CONT;"
4810               OUTPUT @Nwa;"GATESPAN";VAL$(Gate_span(F));"S;"
4820               OUTPUT @Nwa;"GATECENT";VAL$(Gate_cent(F));"S;"
4830               OUTPUT @Nwa;"KEY41; KEY59; KEY58; KEY59;"
4840               LOCAL @Nwa
4850               DISP "Adjust gate center to suit, and press CONTINUE."
```

```
4860              PAUSE
4870              OUTPUT @Nwa;"OUTPACTI;"
4880              ENTER @Nwa;Gate_cent(F)
4890              OUTPUT @Nwa;"GATESPAN";VAL$(Gate_span(F));";"
4900              OUTPUT @Nwa;"KEY41; KEY59; KEY58; KEY4;"
4910              LOCAL @Nwa
4920              DISP "Adjust gate span to suit, and press CONTINUE."
4930              PAUSE
4940              OUTPUT @Nwa;"OUTPACTI;"
4950              ENTER @Nwa;Gate_span(F)
4960        END IF
4970    NEXT F
4980    RETURN
4990    !
5000    !----------------------------------------------------------------------
5010    !
5020 Quit_fast_acq: ! End of program
5030  DISP "PROGRAM EXIT"
5040  GOSUB Deallocate_mtrx
5050  LOAD KEY "EDITKEY:MEMORY,0,1"
5060  STOP
5070  END
5080  !
5090  !***********************************************************************
5100  !
5110  DEF FNAsk(Prompt$)
5120    OFF KEY
5130    DISP Prompt$;
5140    INPUT "",Yn$
5150    Yn$=UPC$(Yn$[1,1])
5160    SELECT Yn$
5170           CASE ="Y"
5180                RETURN 1
5190           CASE ="N",=""
5200                RETURN 0
5210           CASE ELSE
5220                RETURN 0
5230    END SELECT
5240  FNEND
5250  !
5260  !***********************************************************************
5270  !
5280  DEF FNFileloc$(File$,Dir$)
5290    INTEGER C  ! for the location of the ':' in Dir$ (minus 1)
5300    LET C=POS(Dir$,":")-1
5310    IF C<=0 THEN
5320      RETURN TRIM$(File$&Dir$)
5330    ELSE
5340      RETURN Dir$[1,C]&RPT$("/",Dir$[C,C]<>"/")&File$&Dir$[C+1,LEN(Dir$)]
5350    END IF
5360  FNEND ! Fileloc
5370  !
5380  !***********************************************************************
5390  !
5400   DEF FNTime_stamp$(OPTIONAL Time_format)
5410       !
5420       DIM Time_digits$[4],Year_digits$[6]
5430       DIM Machine_time$[8],Machine_date$[11]
5440       REAL Timedate_now
5450       !
5460       Timedate_now=TIMEDATE
```

```
5470        Machine_date$=DATE$(Timedate_now)
5480        Machine_time$=TIME$(Timedate_now)
5490        Time_digits$=Machine_time$[1,2]&Machine_time$[4,5]
5500        Year_digits$[1,2]=Machine_date$[10,11]
5510        IF Machine_date$[1,1]=" " THEN Machine_date$[1,1]="0"
5520        !
5530        SELECT Machine_date$[4,6]
5540        CASE ="Jan"
5550             Year_digits$[3,4]="01"
5560        CASE ="Feb"
5570             Year_digits$[3,4]="02"
5580        CASE ="Mar"
5590             Year_digits$[3,4]="03"
5600        CASE ="Apr"
5610             Year_digits$[3,4]="04"
5620        CASE ="May"
5630             Year_digits$[3,4]="05"
5640        CASE ="Jun"
5650             Year_digits$[3,4]="06"
5660        CASE ="Jul"
5670             Year_digits$[3,4]="07"
5680        CASE ="Aug"
5690             Year_digits$[3,4]="08"
5700        CASE ="Sep"
5710             Year_digits$[3,4]="09"
5720        CASE ="Oct"
5730             Year_digits$[3,4]="10"
5740        CASE ="Nov"
5750             Year_digits$[3,4]="11"
5760        CASE ="Dec"
5770             Year_digits$[3,4]="12"
5780        END SELECT
5790        !
5800        Year_digits$[5,6]=Machine_date$[1,2]
5810        SELECT NPAR
5820        CASE =0
5830             RETURN Year_digits$[5,6]&Time_digits$
5840        CASE =1
5850             IF Time_format=1 THEN
5860                RETURN Year_digits$&Time_digits$
5870             END IF
5880             IF Time_format=2 THEN
5890                RETURN Year_digits$[3,6]&Time_digits$
5900             END IF
5910        END SELECT
5920    FNEND
5930    !
5940    !******************************************************************
5950    !
5960    DEF FNTrap_level
5970        RETURN VAL(SYSTEM$("SYSTEM PRIORITY"))+1
5980    FNEND
5990    !
6000    !******************************************************************
6010    !
6020  SUB Config_and_poll
6030      COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
6040      COM /System/ System_memory
6050  !
6060  ! Find out what's out there.
6070  !
```

```
6080      ALLOCATE Device_list$(0:31)[20]
6090      ALPHA PEN 4
6100      KBD LINE PEN 3
6110      KEY LABELS PEN 5
6120      Clear_crt
6130      Netwrk_analyzer=0
6140      ALLOCATE Na_ident$[80]
6150      System_memory=VAL(SYSTEM$("AVAILABLE MEMORY")) ! How much memory for RAM-DI
6160      PRINT "AVAILABLE MEMORY: ";System_memory;" BYTES"
6170      ON TIMEOUT 7,4 GOTO No_na      ! In case there is no network analyzer
6180 Is_na:  OUTPUT @Nwa;"FORM4; OUTPIDEN;"
6190         ENTER @Nwa_data2;Na_ident$
6200         IF POS(Na_ident$,"8510A") THEN Netwrk_analyzer=1
6210         IF POS(Na_ident$,"8510B") THEN Netwrk_analyzer=2
6220         IF POS(Na_ident$,"8720A") THEN Netwrk_analyzer=3
6230         IF POS(Na_ident$,"8720B") THEN Netwrk_analyzer=4
6240         IF POS(Na_ident$,"8753A") THEN Netwrk_analyzer=5
6250         IF POS(Na_ident$,"8753B") THEN Netwrk_analyzer=6
6260         LOCAL @Nwa
6270         PRINT
6280         PRINT Na_ident$
6290         PRINT Netwrk_analyzer
6300      !  Clear_crt
6310      PRINT
6320      PRINT
6330         IF Netwrk_analyzer=0 THEN
6340            !
6350            !
6360 No_na:     BEEP
6370            OFF CYCLE
6380            PRINT TABXY(1,5);"There is no active network analyzer on the HPIB bus
6390            PRINT TABXY(1,6);"Please check connections, and press the RUN key."
6400            PRINT
6410            PRINT TABXY(1,7);"If you DO NOT want to use a network analyzer,
                                    press the CONTINUE key."
6420            PAUSE
6430         END IF
6440         !
6450         !
6460 Check_hpib:  ! Check the rest of the bus
6470      ON TIMEOUT 7,.01 GOTO Nothing
6480      !
6490      FOR Device=700 TO 731
6500          DISP "Checking for device at address: ";Device
6510          Device_list$(Device-700)="NOTHING"
6520          ASSIGN @What_is_it TO Device
6530          Outcome=SPOLL(@What_is_it)
6540          Device_list$(Device-700)="SOMETHING"
6550          PRINT Device,"SOMETHING HERE","spoll: ";Outcome
6560          ASSIGN @What_is_it TO *
6570 Nothing: ! Skip to next device
6580      NEXT Device
6590      !
6600      OFF TIMEOUT 7
6610      ASSIGN @What_is_it TO *
6620      IF Device_list$(1)="SOMETHING" THEN
6630         DISP "Position the printer to Top-Of-Form and press CONTINUE..."
6640         PAUSE
6650         PRINTER IS PRT
6660         PRINT CHR$(27)&"&l1L"; ! Set Page Breaks
6670         Printer_flag=1
```

```
6680        PRINTER IS CRT
6690      END IF
6700      DEALLOCATE Na_ident$
6710      DEALLOCATE Device_list$(*)
6720      ABORT @Hpib
6730  SUBEXIT
6740  SUBEND
6750  !
6760  ! ***********************************************************************
6770  !
6780  SUB Hp_bus_init
6790    COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
6800    COM /Sys_1/ Freq$(*),Freq_cent(*),Freq_span(*),Gate_cent(*),Gate_span(*)
6810    COM /Sys_2/ Pol$(*),Polsw$(*)
6820    COM /System_config/ INTEGER Printer_flag,Debug_flag,Version$,
                            Mode$,Out_type$,Sound$,Bell$,Target$,Ref_target$
6830    !
6840    !  This subroutine configures the HP-IB bus and presets the HP8510.
6850    !
6860    ASSIGN @Hpib TO 7
6870    ASSIGN @Nwa TO 716
6880    ASSIGN @Nwa_data1 TO 716;FORMAT OFF
6890    ASSIGN @Nwa_data2 TO 716;FORMAT ON
6900    ASSIGN @Relay TO 710
6910    REMOTE @Hpib
6920    ABORT @Hpib
6930    CLEAR @Nwa
6940    IF Debug_flag=1 THEN OUTPUT @Nwa;"DEBUON;"
6950    IF Debug_flag=0 THEN
6960      OUTPUT @Nwa;"DEBUOFF;"
6970      OUTPUT @Nwa;"TITL """&Freq$(2)&" BAND   """
6980    END IF
6990  SUBEND
7000  !
7010  !***********************************************************************
7020  !
7030  SUB Series_init
7040  COM /System_config/ INTEGER Printer_flag,Debug_flag,Version$,Mode$,Out_type$
                       ,Sound$,Bell$,Target$,Ref_target$
7050    DIM Input$[80]
7060    !
7070    !  This subroutine prints a header for the printout and sets the system
7080    !  date and time.
7090    !
7100    IF Printer_flag=1 THEN PRINTER IS PRT
7110    PRINT CHR$(12)
7120    Set_clock
7130    LINPUT "ENTER MEASUREMENT SERIES TITLE",Input$
7140    Preface$="*"&RPT$(" ",9)
7150    PRINT RPT$("*",70)
7160    PRINT Preface$&Input$
7170    LINPUT "ENTER OPERATOR NAME",Input$
7180    PRINT Preface$&Input$
7190    PRINTER IS CRT
7200    PRINT
7210    PRINT
7220    PRINT Preface$&"MEASUREMENT SERIES STARTED AT "&TIME$(TIMEDATE)
7230    PRINTER IS CRT
7240  SUBEND
7250  !
7260  !****************************************************************.......
```

```
7270  !
7280  SUB Set_clock
7290      OPTION BASE 1
7300      INTEGER I
7310      DIM Chrono$[12],Month$(12)[3]
7320      Exec_key$=CHR$(255)&CHR$(88)
7330      READ Month$(*)
7340      DATA "JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP","OCT"
              ,"NOV","DEC"
7350      OUTPUT KBD;"SCRATCH KEY "&Exec_key$;
7360      Clear_crt
7370      PRINT "                                Current system date: ";DATE$(TIMEDATE)
7380      PRINT "                                Current system time: ";TIME$(TIMEDATE)
7390      PRINT
7400 Ask: LINPUT "Enter date and time (YYMMDDHHMMss) :",Chrono$
7410      IF Chrono$="" AND DATE$(TIMEDATE)<>" 1 Mar 1900" THEN
7420         Clear_crt
7430         SUBEXIT
7440      END IF
7450      Year$=VAL$(1900+VAL(Chrono$[1,2]))
7460      IF (VAL(Chrono$[3,4])<=0 OR VAL(Chrono$[3,4])>12) THEN
7470         BEEP
7480         PRINT "Incorrect month value."
7490         GOTO Ask
7500      END IF
7510      Year$=Month$(VAL(Chrono$[3,4]))&" "&Year$
7520      Year$=Chrono$[5,6]&" "&Year$
7530      SET TIMEDATE (DATE(Year$))
7540      IF (VAL(Chrono$[7,8]))>23 THEN
7550         BEEP
7560         PRINT "Incorrect hour value."
7570         GOTO Ask
7580      END IF
7590      Day$=Chrono$[7,8]&":"
7600      IF VAL(Chrono$[9,10])>59 THEN
7610         BEEP
7620         PRINT "Incorrect minute value."
7630         GOTO Ask
7640      END IF
7650      Day$=Day$&Chrono$[9,10]&":"
7660      IF (LEN(Chrono$)>10 AND LEN(Chrono$)=12) THEN
7670         IF VAL(Chrono$[11,12])>59 THEN
7680            BEEP
7690            PRINT "Incorrect seconds value."
7700            GOTO Ask
7710         END IF
7720         Day$=Day$&Chrono$[11,12]
7730      ELSE
7740         Day$=Day$&"00"
7750      END IF
7760      SET TIME TIME(Day$)
7770      Clear_crt
7780   SUBEXIT
7790   SUBEND
7800   !
7810   !*********************************************************************
7820   !
7830   SUB Fix_error
7840   SELECT ERRN
7850   CASE ELSE
7860        PRINTER IS CRT
```

```
7870         PRINT "ERROR ";ERRN
7880         PRINT ERRM$
7890         PRINT " PROGRAM IS PAUSED.  FIX ERROR, IF POSSIBLE, AND CONTINUE."
7900         PAUSE
7910   END SELECT
7920   SUBEND
7930   !
7940   !*********************************************************************
7950   !
7960  SUB Clear_crt(OPTIONAL INTEGER Start_line,Num_of_lines)
7970      !
7980      INTEGER I
7990      DIM Clear_line$[80]
8000      Clear_line$="                                   "
8010      IF NPAR=0 THEN
8020         OUTPUT KBD;CHR$(255)&CHR$(75);
8030      ELSE
8040         PRINT TABXY(1,Start_line);"";RPT$(Clear_line$,Num_of_lines)
8050         PRINT TABXY(1,Start_line);"";
8060         SUBEXIT
8070      END IF
8080  SUBEND
8090  !
8100  !*********************************************************************
8110  !
8120  SUB Print_banner1
8130      Clear_crt
8140      PRINT
8150      PRINT
8160      PRINT TABXY(3,16);"**************************************************"
8170      PRINT TABXY(4,16);"*                                                *"
8180      PRINT TABXY(5,16);"*                    LCX                         *"
8190      PRINT TABXY(6,16);"*     UNIVERSITY OF MICHIGAN RADIATION LAB       *"
8200      PRINT TABXY(7,16);"*           L/C/X MEASUREMENT PROGRAM            *"
8210      PRINT TABXY(8,16);"*                 (VERSION 8.0)                  *"
8220      PRINT TABXY(9,16);"*                                                *"
8230      PRINT TABXY(10,16);"*                May 21, 1990                   *"
8240      PRINT TABXY(11,16);"*                                               *"
8250      PRINT TABXY(12,16);"*************************************************"
8260  SUBEXIT
8270  SUBEND
8280  !
8290  !*********************************************************************
8300  !
8310   SUB Print_banner2
8320   PRINT "Don't use Print_banner2."
8330   SUBEND
8340   !
8350   !*********************************************************************
8360   !
8370   SUB Print_banner3
8380   PRINT "Don't use Print_banner3."
8390   SUBEND
8400  !
8410  !*********************************************************************
8420  !
8430   SUB Print_banner4
8440   COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
8450   COM /Constants/ Vel,Zero(*),Exec_key$
8460   COM /System_config/ INTEGER Printer_flag,Debug_flag,Version$,Mode$,
                          Out_type$,Sound$,Bell$,Target$,Ref_target$
```

```
8470  COM /Sys_1/ Freq$(*),Freq_cent(*),Freq_span(*),Gate_cent(*),Gate_span(*)
8480  COM /Sys_2/ Pol$(*),Polsw$(*)
8490  COM /Sys_3/ INTEGER F_disp,P_disp
8500  COM /Sys_4/ Drive_a$,Drive_b$,Drive_c$,INTEGER Preamble,Bytes
8510  COM /Sys_5/ INTEGER Nskip,Ndata
8520  COM /Sys_6/ Ref_angle,Angle,Angle$,Beam(*),INTEGER Npts,Ntrace,
                  Average_factor
8530  COM /Sys_7/ INTEGER Meas_flag(*)
8540      !
8550      !
8560   OFF KEY
8570   Clear_crt
8580   PRINT
8590   PRINT
8600   PRINT "           PARAMETER                       CURRENT VALUE"
8610   PRINT
8620   PRINT "                     FREQUENCY             ";
8630   FOR F=1 TO 3
8640     IF Meas_flag(F) THEN PRINT Freq$(F)&" ";
8650   NEXT F
8660   PRINT "      "
8670   PRINT "        "
8680   PRINT "                     ANTENNA ANGLE         "&Angle$
8690   PRINT "                     TARGET TYPE           "&Target$
8700   PRINT "                     MEASUREMENT MODE      "&Mode$&"       "
8710 !  PRINT "                     CURRENT DISPLAY        "&Freq$(F_disp)&" "&Pol$(
8720   PRINT "        "
8730   PRINT "                     # OF TRACES/SET       ";Ntrace
8740   PRINT "                     # OF SAMPLE POINTS    ";Npts
8750   PRINT "                     # OF DATA POINTS      ";Ndata
8760   PRINT "                      (to be stored)       "
8770   PRINT "                     # OF AVERAGES         ";Average_factor
8780 SUBEXIT
8790 SUBEND
8800 !
8810 !***********************************************************************
8820 !
8830 SUB Store_file(COMPLEX Matrix(*),File_type$,Filename$,INTEGER F)
8840     !
8850     COM /Sys_1/ Freq$(*),Freq_cent(*),Freq_span(*),Gate_cent(*),Gate_span(*)
8860     COM /Sys_2/ Pol$(*),Polsw$(*)
8870     COM /Sys_5/ INTEGER Nskip,Ndata
8880     COM /Sys_6/ Ref_angle,Angle,Angle$,Beam(*),INTEGER Npts,Ntrace,
                     Average_factor
8890     COM /Sys_7/ INTEGER Meas_flag(*)
8900     COM /System_config/ INTEGER Printer_flag,Debug_flag,Version$,Mode$,
                             Out_type$,Sound$,Bell$,Target$,Ref_target$
8910     !
8920     !
8930     INTEGER Records_per_set,T
8940     REAL Bytes_per_set
8950     DIM Suffix$[2]
8960     ALLOCATE COMPLEX Trace(Ndata)
8970     !
8980     !
8990     DISP "Saving file."
9000     SELECT File_type$
9010     CASE ="SKY"  ! Sky data.
9020             Bytes_per_set=16*Ndata
9030             Records_per_set=4*SUM(Meas_flag)*Ntrace
9040           IF SUM(Meas_flag)=3 THEN
```

```
9050           Suffix$="SA"
9060          ELSE
9070           FOR F=1 TO 3
9080             IF Meas_flag(F)=1 THEN
9090               Mf=F
9100             END IF
9110           NEXT F
9120           Suffix$="S"&Freq$(Mf)
9130          END IF
9140            GOSUB Save_traces
9150         !
9160         !
9170     CASE ="REF"
9180         Bytes_per_set=16*Ndata
9190         Records_per_set=4*SUM(Meas_flag)*Ntrace
9200          IF SUM(Meas_flag)=3 THEN
9210            Suffix$="RA"
9220          ELSE
9230           FOR F=1 TO 3
9240             IF Meas_flag(F)=1 THEN
9250               Mf=F
9260             END IF
9270           NEXT F
9280           Suffix$="R"&Freq$(Mf)
9290          END IF
9300            GOSUB Save_traces
9310         !
9320         !
9330     CASE ="MNT"
9340         Bytes_per_set=16*Ndata
9350         Records_per_set=4*SUM(Meas_flag)*Ntrace
9360          IF SUM(Meas_flag)=3 THEN
9370            Suffix$="MA"
9380          ELSE
9390           FOR F=1 TO 3
9400             IF Meas_flag(F)=1 THEN
9410               Mf=F
9420             END IF
9430           NEXT F
9440           Suffix$="M"&Freq$(Mf)
9450          END IF
9460            GOSUB Save_traces
9470         !
9480         !
9490     CASE ="GND"
9500         Bytes_per_set=16*Ndata
9510         Records_per_set=Ntrace*4*SUM(Meas_flag)
9520          IF SUM(Meas_flag)=3 THEN
9530            Suffix$="GA"
9540          ELSE
9550           FOR F=1 TO 3
9560             IF Meas_flag(F)=1 THEN
9570               Mf=F
9580             END IF
9590           NEXT F
9600           Suffix$="G"&Freq$(Mf)
9610          END IF
9620            GOSUB Save_traces
9630     END SELECT
9640     DEALLOCATE Trace(*)
9650     SUBEXIT
```

```
9660          !
9670          !
9680 Save_averaged: !
9690                   ! Save the reference data file.
9700                   !
9710     IF NOT Debug_flag THEN
9720        CREATE BDAT Filename$&Suffix$&Drive_c$,Records_per_set,Bytes_per_set
9730     END IF
9740     Base_record=0
9750     FOR F=1 TO 3
9760     IF Meas_flag(F)=1 THEN
9770     IF Debug_flag THEN
9780        ASSIGN @Disc TO PRT
9790        OUTPUT @Disc;"FILE: ",Filename$,Suffix$
9800        OUTPUT @Disc USING Image_1;Version$,Freq_cent(F),Freq_span(F)
9810        OUTPUT @Disc USING Image_2;Ndata,Average_factor
9820        OUTPUT @Disc USING Image_3;Ref_target$,T
9830        FOR P=1 TO 4
9840            OUTPUT @Disc USING Image_4;Pol$(P),Gate_cent(F),Gate_span(F)
9850            MAT Trace= Matrix(F,1,P,*)
9860            OUTPUT @Disc;Trace(*)
9870        NEXT P
9880     ELSE
9890        ASSIGN @Disc TO Filename$&Suffix$&Drive_c$;FORMAT OFF
9900        OUTPUT @Disc,Base_record+1;Version$,Freq_cent(F),Freq_span(F)
9910        OUTPUT @Disc,Base_record+1;Ndata,Average_factor
9920        OUTPUT @Disc,Base_record+1;Ref_target$,T
9930        FOR P=1 TO 4
9940            OUTPUT @Disc,Base_record+P;Pol$(P),Gate_cent(F),Gate_span(F)
9950            MAT Trace= Matrix(F,1,P,*)
9960            OUTPUT @Disc,Base_record+P;Trace(*)
9970        NEXT P
9980        Base_record=Base_record+4
9990     END IF
10000    END IF
10010    NEXT F
10020    ASSIGN @Disc TO *
10030 RETURN
10040 !
10050 !-----------------------------------------------------------------------------
10060 !
10070 Save_traces:!
10080               ! Save the ground target data file.
10090               !
10100    IF NOT Debug_flag THEN
10110        CREATE BDAT Filename$&Suffix$&Drive_c$,Records_per_set,Bytes_per_set
10120       Base_record=0
10130    END IF
10140    IF Debug_flag THEN
10150       ASSIGN @Disc TO PRT
10160       OUTPUT @Disc;"FILE: ";Filename$;Suffix$
10170       OUTPUT @Disc USING Image_5;Ndata,Ntrace
10180       OUTPUT @Disc USING Image_3;Target$
10190       FOR T=1 TO Ntrace
10200         FOR F=1 TO 3
10210          IF Meas_flag(F)=1 THEN
10220           OUTPUT @Disc USING Image_1;Version$,Freq_cent(F),Freq_span(F)
10230           FOR P=1 TO 4
10240             OUTPUT @Disc USING Image_4;Pol$(P),Gate_cent(F),Gate_span(F)
10250             MAT Trace= Matrix(F,T,P,*)
10260             OUTPUT @Disc;Trace(*)
```

```
10270            NEXT P
10280           END IF
10290          NEXT F
10300         NEXT T
10310     ELSE
10320        ASSIGN @Disc TO Filename$&Suffix$&Drive_c$;FORMAT OFF
10330 !       OUTPUT @Disc,1;Ndata,Ntrace
10340 !       OUTPUT @Disc,1;Target$
10350        FOR T=1 TO Ntrace
10360          FOR F=1 TO 3
10370          IF Meas_flag(F)=1 THEN
10380 !          OUTPUT @Disc,Base_record+1;Version$,Freq_cent(F),Freq_span(F)
10390            FOR P=1 TO 4
10400 !               OUTPUT @Disc,Base_record+P;Pol$(P),Gate_cent(F),Gate_span(F),T
10410                MAT Trace= Matrix(F,T,P,*)
10420                 OUTPUT @Disc,Base_record+P;Trace(*)
10430             NEXT P
10440            Base_record=Base_record+4
10450           END IF
10460          NEXT F
10470        NEXT T
10480     END IF
10490     ASSIGN @Disc TO *
10500 RETURN
10510 !
10520 !------------------------------------------------------------------------
10530 !
10540 Image_1:IMAGE (1X,12A,5X,"FREQ CENTER: ",2D.4D,5X,"FREQ SPAN: ",2D.4D)
10550 Image_2:IMAGE ("NUMBER OF POINTS: ",5D,5X,"NUMBER OF AVERAGES: ",5D)
10560 Image_3:IMAGE ("TARGET: ",30A,"GATING TARGET TYPE: ",2D)
10570 Image_4:IMAGE ("POLARIZATION: ",2A,5X,"GATE CENTER: ",SD.14DE,/,5X,"GATE SPAN
10580 Image_5:IMAGE ("NUMBER OF POINTS: ",5D,5X,"NUMBER OF TRACES: ",5D)
10590 Image_6:IMAGE (5X,SD.14DE,5X,SD.14DE)
10600 SUBEND
10610 !
10620 !************************************************************************
10630 !
10640 SUB Freq_set(INTEGER Ifreq)
10650 COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
10660 COM /Sys_1/ Freq$(*),Freq_cent(*),Freq_span(*),Gate_cent(*),Gate_span(*)
10670   !
10680   !  This subroutine sets the transmit frequency for the HP8753.
10690   !
10700   SELECT Netwrk_analyzer
10710   CASE =3,=4,=5,=6
10720        OUTPUT @Nwa;"TIMDTRANOFF;"
10730   CASE =1,=2
10740        OUTPUT @Nwa;"FREQ;"
10750   END SELECT
10760   OUTPUT @Nwa;"CENT "&VAL$(Freq_cent(Ifreq))&" GHZ;"
10770   OUTPUT @Nwa;"SPAN "&VAL$(Freq_span(Ifreq))&" GHZ;"
10780 SUBEND
10790 !
10800 !************************************************************************
10810 !
10820 SUB Freq_sw(INTEGER Ifreq)
10830 COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
10840   SELECT Ifreq
10850   CASE 1
10860     OUTPUT @Relay;"?*A1B2"
10870   CASE 2
```

```
10880     OUTPUT @Relay;"?*A2B1"
10890   CASE 3
10900     OUTPUT @Relay;"?*B12"
10910   END SELECT
10920   WAIT .1
10930 SUBEND
10940 !
10950 !************************************************************************
10960 !
10970 SUB Pol_sw(INTEGER Ifreq,Ipol)
10980     COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
10990     COM /Sys_1/ Freq$(*),Freq_cent(*),Freq_span(*),Gate_cent(*),Gate_span(*)
11000     COM /Sys_2/ Pol$(*),Polsw$(*)
11010   !
11020   !  This subroutine sets the transmit and receive polarization by
11030   !  sending the proper command over the HPIB to the polarization
11040   !  relays.
11050   !
11060   OUTPUT @Relay;Polsw$(Ifreq,Ipol)
11070   OUTPUT @Nwa;"TITL "" "&Freq$(Ifreq)&" BAND - "&Pol$(Ipol)&""""
11080   WAIT .1
11090 SUBEND
11100 !
11110 !************************************************************************
11120 !
11130 SUB Rotate_target
11140 OPTION BASE 1
11150 COM /Com4/ INTEGER Rotation_state,REAL Inc_angle,Current_angle,Start_angle,
            Stop_angle,Old_home_angle,INTEGER Sets_per_pos
11160 COM /Status/ INTEGER Sc,Connect_flg,E_flg,Debug_flg,Response$[80]
11170 INTEGER Fs_flag,Ss_flag,Speed,Imc_status,Confirm_answer
11180    !
11190    !
11200    Confirm_answer=1
11210    Imc_status=0
11220    Debug_flg=0
11230    Fs_flag=-1
11240    Ss_flag=-1
11250    Clear_crt(3,16)
11260    !
11270    !
11280    SELECT Rotation_state
11290    CASE =-1
11300         IF FNAsk("Do you wish to use the rotator?") THEN
11310            Connect_flg=0
11320            GOSUB Init_imc
11330            GOSUB Init_graph_pos
11340            GOSUB Manual_loop
11350            PRINT "Set Auto Mode Please....."
11360         ELSE
11370            Rotation_state=0
11380            GCLEAR
11390            GRAPHICS OFF
11400         END IF
11410    CASE =0
11420         SUBEXIT
11430    CASE =1
11440         GOSUB Check_position
11450         GOSUB Print_angles
11460         GOSUB Manual_loop
11470    CASE =2
```

```
11480            GOSUB Check_position
11490            GOSUB Auto
11500       CASE =3
11510            GOSUB Check_position
11520            GOSUB Manual_loop
11530            GOSUB Auto
11540       CASE =4
11550            GOSUB Check_position
11560            GOSUB Go_home
11570       CASE =5
11580            GOSUB Check_position
11590            Rotation_state=1  ! Switch to manual mode.
11600       END SELECT
11610       SUBEXIT
11620       !
11630       !
11640  Init_imc: ! Initialize the IMC unit.
11650       GOSUB Check_4_fault
11660       PRINT TABXY(1,3);"INITIALIZING IMC"
11670       Clear_crt(4,15)
11680       Comm("4WB")             ! Set warm boot (clear flags).
11690       PRINT TABXY(1,4);"WB"
11700       Comm("4EB")             ! Clear IMC buffer.
11710       PRINT TABXY(1,4);"EB"
11720       Encoder_ratio=4096  ! 32000
11730       Comm("4ER"&VAL$(Encoder_ratio))        ! Load encoder ratio.
11740       PRINT TABXY(1,4);"ER"&VAL$(Encoder_ratio)
11750       IF FNAsk("Do you wish to set home at the current position?") THEN
11760          Comm("4RS",Confirm_answer)
11770          ENTER Response$;Old_home_angle
11780          Old_home_angle=Old_home_angle/93.3
11790          Comm("4PIZ0")           ! Set IMC at 0.
11800          PRINT TABXY(1,4);"PIZ"&RPT$(" ",LEN(VAL$(Encoder_ratio)))
11810          Comm("4PIA0")           ! Set IMC at 0.
11820          PRINT TABXY(1,4);"PIA"
11830          Current_angle=0
11840       END IF
11850       Comm("4SP100")          ! Set speed to (50pps).
11860       PRINT TABXY(1,4);"SP "&RPT$(" ",LEN(VAL$(Encoder_ratio)))
11870       Comm("4AC500")          ! Set acceleration (500pps^2).
11880       PRINT TABXY(1,4);"AC "
11890       Comm("4DC500")          ! Set deceleration (500pps^2).
11900       PRINT TABXY(1,4);"DC "
11910       GOSUB Check_position
11920       Rotation_state=1
11930       Clear_crt
11940  !
11950  !
11960       PRINT TABXY(1,4);"DONE INITIALIZING IMC"
11970       PRINT TABXY(1,5);"Turntable currently in manual mode."
11980       PRINT TABXY(1,6)
11990  Print_angles:!
12000       PRINT TABXY(1,7);"Current  angle is: ";Current_angle;" degrees."
12010       PRINT TABXY(1,8);"Starting angle is: ";Start_angle;" degrees."
12020       PRINT TABXY(1,9);"Stopping angle is: ";Stop_angle;" degrees."
12030  RETURN
12040  !
12050  !
12060  Manual_loop:! Main activation loop.
12070       LOOP
12080         ON KEY 0 LABEL "FAST SLEW CW  ",FNTrap_level GOSUB Fs_cw
```

```
12090       ON KEY 1 LABEL "FAST SLEW CCW ",FNTrap_level GOSUB Fs_ccw
12100       ON KEY 5 LABEL "SLOW SLEW CW  ",FNTrap_level GOSUB Ss_cw
12110       ON KEY 6 LABEL "SLOW SLEW CCW ",FNTrap_level GOSUB Ss_ccw
12120       ON KEY 2 LABEL "MANUAL CONTROL",FNTrap_level GOSUB Manual
12130       ON KEY 3 LABEL "TARGET GO HOME",FNTrap_level GOSUB Go_home
12140       ON KEY 4 LABEL "STOP ROTATION ",FNTrap_level GOSUB Stop_turn
12150       ON KEY 7 LABEL "SET AUTO MODE ",FNTrap_level GOSUB Set_auto
12160       ON KEY 8 LABEL "SET TARGET HOME",FNTrap_level GOSUB Set_position
12170       ON KEY 9 LABEL "RETURN         ",FNTrap_level GOTO Quit
12180       GOSUB Check_position
12190     END LOOP
12200 !
12210 !-----------------------------------------------------------------------
12220 !
12230 Fs_cw:! Fast slew clockwise.
12240       IF Fs_flag<0 THEN
12250          Comm("4SP500")
12260          Comm("4SFN")
12270          Fs_flag=-1*Fs_flag
12280          Clear_crt(3,15)
12290          PRINT TABXY(1,15);"ROTATING CW (FAST)"
12300       ELSE
12310          Comm("4ST")
12320          Fs_flag=-1*Fs_flag
12330          Clear_crt(3,15)
12340          PRINT TABXY(1,15);"ROTATION STOPPED"
12350          GOSUB Check_position
12360       END IF
12370 RETURN
12380 !
12390 !-----------------------------------------------------------------------
12400 !
12410 Fs_ccw:! Fast slew counterclockwise.
12420       IF Fs_flag<0 THEN
12430          Comm("4ST")
12440          Comm("4SP500")
12450          Comm("4SRN")
12460          Fs_flag=-1*Fs_flag
12470          Clear_crt(3,10)
12480          PRINT TABXY(1,15);"ROTATING CCW (FAST)"
12490       ELSE
12500          Comm("4ST")
12510          Fs_flag=-1*Fs_flag
12520          Clear_crt(3,15)
12530          PRINT TABXY(1,15);"ROTATION STOPPED"
12540          GOSUB Check_position
12550       END IF
12560 RETURN
12570 !
12580 !-----------------------------------------------------------------------
12590 !
12600 Ss_cw:! Slow slew clockwise.
12610       IF Ss_flag<0 THEN
12620          Comm("4ST")
12630          INPUT "Speed?",Sp
12640          Comm("4SP"&VAL$(INT(Sp)))
12650          Comm("4SFN")
12660          Ss_flag=-1*Ss_flag
12670          Clear_crt(3,15)
12680          PRINT TABXY(1,15);"ROTATING CW (SLOW)"
12690       ELSE
```

```
12700          Comm("4ST")
12710          Ss_flag=-1*Ss_flag
12720          Clear_crt(3,15)
12730          PRINT TABXY(1,15);"ROTATION STOPPED"
12740          GOSUB Check_position
12750       END IF
12760 RETURN
12770 !
12780 !--------------------------------------------------------------------
12790 !
12800 Ss_ccw:! Slow slew counterclockwise.
12810       IF Ss_flag<0 THEN
12820       INPUT "Speed?",Sp
12830          Comm("4SP"&VAL$(INT(Sp)))
12840          Comm("4SRN")
12850          Ss_flag=-1*Ss_flag
12860          Clear_crt(3,15)
12870          PRINT TABXY(1,15);"ROTATING CCW (SLOW)"
12880       ELSE
12890          Comm("4ST")
12900          Ss_flag=-1*Ss_flag
12910          Clear_crt(3,15)
12920          PRINT TABXY(1,15);"ROTATION STOPPED"
12930          GOSUB Check_position
12940       END IF
12950 RETURN
12960 !
12970 !--------------------------------------------------------------------
12980 !
12990 Manual: INPUT "ANGLE (IN DEGREES)=?",Inc_angle
13000         INPUT "SPEED? (~100--~500 RECOMMENDED)",Speed
13010         Comm("4SP"&VAL$(Speed))
13020 Auto:   SELECT Rotation_state
13030         CASE =4
13040              GOSUB Go_home
13050              Rotation_state=2
13060              GOTO Auto
13070         CASE ELSE
13080              Angl2=Inc_angle*93.3
13090              Angl1=INT(Angl2)
13100              IF Angl2-Angl1>=.5 THEN Angl1=Angl1+1
13110         !    Current_angle=Current_angle+Inc_angle
13120              Inc_angle$=VAL$(Angl1)
13130              Comm("4IM"&Inc_angle$)
13140              Comm("4RFI")
13150         END SELECT
13160         Imc_status=0
13170         Clear_crt(3,7)
13180         PRINT TABXY(1,14);"ROTATING TARGET, PLEASE WAIT."
13190         !
13200         !
13210         WHILE NOT BIT(Imc_status,0)            ! Wait for motor to stop.
13220               Comm("4RS",Confirm_answer)
13230               ENTER Response$;Imc_status
13240               PRINT TABXY(1,15);DVAL$(Imc_status,2)
13250               GOSUB Check_position
13260               WAIT 1
13270         END WHILE
13280         Imc_status=0
13290         !
13300         !
```

```
13310          Clear_crt(3,16)
13320          PRINT TABXY(1,16);"CURRENT TARGET POSITION IS ";Current_angle;" DEGREES
13330          WAIT 2  ! Wait for target settling.
13340 RETURN
13350 !
13360 !-----------------------------------------------------------------------
13370 !
13380 Stop_turn:Comm("4ST")
13390           WHILE NOT BIT(Imc_status,0)          ! Wait for motor to stop.
13400                 Comm("4RS",Confirm_answer)
13410                 ENTER Response$;Imc_status
13420                 WAIT .1
13430           END WHILE
13440           Clear_crt(3,16)
13450           PRINT TABXY(1,15);"ROTATION STOPPED"
13460           GOSUB Check_position
13470           Imc_status=0
13480 RETURN
13490 !
13500 !-----------------------------------------------------------------------
13510 !
13520 Set_auto: Comm("4SP500")
13530           GOSUB Check_position
13540           Clear_crt(3,16)
13550           PRINT TABXY(1,3);"Current starting  angle: ";Start_angle;" degrees"
13560           PRINT TABXY(1,4);"Current increment angle: ";Inc_angle;" degrees"
13570           PRINT TABXY(1,5);"Current stopping  angle: ";Stop_angle;" degrees"
13580           PRINT TABXY(1,6);"Current rotation  speed: ";Speed
13590           PRINT TABXY(1,7);RPT$(" ",80)
13600           PRINT TABXY(1,8);"Rotator  positioned  at: ";Current_angle;" degrees"
13610           INPUT "Enter starting angle value (degrees): ",Start_angle
13620           INPUT "Enter increment angle (degrees): ",Inc_angle
13630           INPUT "Enter stopping angle (degrees): ",Stop_angle
13640           INPUT "Enter rotation speed of target (~500 recommended): ",Speed
13650           Speed=INT(Speed)
13660           Comm("4SP"&VAL$(Speed))
13670           IF ABS(Start_angle-Current_angle)>.1 THEN
13680              PRINT TABXY(1,9);RPT$(" ",80)
13690              PRINT TABXY(1,10);"Rotating target to starting angle..."
13700              Temp_angle=Inc_angle
13710              Inc_angle=Start_angle-Current_angle
13720              GOSUB Auto
13730              Inc_angle=Temp_angle
13740           END IF
13750           Rotation_state=2
13760           Clear_crt
13770           PRINT TABXY(1,20);"Turntable is in automatic mode. (press the RETURN
13780 RETURN
13790 !
13800 !-----------------------------------------------------------------------
13810 !
13820 Set_position:INPUT "LOCK IN CURRENT TARGET POSITION AS REFERENCE POSITION?",Yn$
13830             IF Yn$="Y" OR Yn$="y" THEN
13840                Comm("4RS",Confirm_answer)
13850                ENTER Response$;Old_home_angle
13860                Old_home_angle=Old_home_angle/93.3
13870                Comm("4PIA0") ! Set absolute position to zero.
13880                Comm("4PIZ0") ! Set incremental postition to zero.
13890                Current_angle=0
13900             ELSE
13910                PRINT "POSITION WAS NOT SET."
```

```
13920               END IF
13930 RETURN
13940 !
13950 !---------------------------------------------------------------------
13960 !
13970 Go_home: IF Speed<200 THEN Speed=200
13980          Comm("4SP"&VAL$(Speed))
13990          Comm("4AM0") ! Move to zero absolute position.
14000          Comm("4RAN") ! Initiate movement.
14010          Comm("4MW")  ! Make sure the move is completed.
14020          Imc_status=0
14030          Clear_crt(3,15)
14040          PRINT TABXY(1,14);"ROTATING TARGET TO HOME POSITION, PLEASE WAIT."
14050          WHILE NOT (BIT(Imc_status,0) AND BIT(Imc_status,5))
14060               GOSUB Check_status
14070               PRINT TABXY(1,15);"CURRENT STATUS: ";DVAL$(Imc_status,2)
14080               GOSUB Check_position
14090               WAIT .1
14100          END WHILE
14110          Clear_crt(3,16)
14120          PRINT TABXY(1,15);"TARGET AT HOME POSITION."
14130          GOSUB Check_position
14140          Imc_status=0
14150 RETURN
14160 !
14170 !---------------------------------------------------------------------
14180 !
14190 Check_status:! Keep an eye on the Whedco controller status.
14200         Comm("4RS",Confirm_answer)
14210         ENTER Response$;Imc_status
14220 RETURN
14230 !
14240 !---------------------------------------------------------------------
14250 !
14260 Check_position:! Get the current turnstile position in degrees.
14270         Comm("4RP",Confirm_answer)
14280         ENTER Response$;Motor_position
14290         Current_angle=Motor_position/93.3
14300    !    Current_angle=Current_angle+Inc_angle
14310         PRINT TABXY(1,16);"CURRENT TARGET POSITION IS ";Current_angle;" DEGREE
14320         GOSUB Draw_positions
14330 RETURN
14340 !
14350 !---------------------------------------------------------------------
14360 !
14370 Check_4_fault: ! Check the IMC for a fault condition and correct or
14380                ! notify the user if necessary.
14390                !
14400     Comm("4FC",Confirm_answer)
14410     ENTER Response$;Fault$
14420     SELECT Fault$
14430     CASE ="Power failure" ! Loss of power
14440          RETURN
14450     CASE ="Force DAC" ! Force DAC command was given
14460          BEEP
14470          PRINT "Force DAC command was given..."
14480          DISP "Press CONTINUE to resume..."
14490          PAUSE
14500     RETURN
14510     CASE ="Over-current" ! Over-current condition exists.
14520          BEEP
```

```
14530           PRINT "An over-current condition has been detected on the IMC."
14540           PRINT
14550           PRINT "Cycle the power to the IMC until the OV-CUR LED goes out"
14560           DISP "Press CONTINUE to reinitialize the IMC"
14570           PAUSE
14580           GOSUB Init_imc
14590      RETURN
14600      END SELECT
14610      RETURN
14620 !
14630 !-------------------------------------------------------------------------
14640 !
14650 Init_graph_pos: ! Creates a graphical depiction of where the target is.
14660                 !
14670     GINIT
14680     GCLEAR
14690     GRAPHICS ON
14700     SHOW 0,100,0,100
14710     PENUP
14720     MOVE 90,70
14730     PEN 1        ! Draw circle
14740     POLYGON 12,360,360
14750     PENUP
14760     MOVE 90,70  ! Draw old home orientation.
14770     PEN 2
14780     DRAW 90+11*COS(Old_home_angle),70-11*SIN(Old_home_angle)
14790     PENUP
14800     MOVE 90,70  ! Draw current home orientation.
14810     PEN 4
14820     DRAW 90,58
14830     PENUP
14840     MOVE 90,70  ! Draw current target orienation.
14850     PEN 3
14860     X_pos=90+11*COS(Current_angle)
14870     Y_pos=70-11*SIN(Current_angle)
14880     DRAW X_pos,Y_pos
14890 RETURN
14900 !
14910 !-------------------------------------------------------------------------
14920 !
14930 Draw_positions:  ! Draws out the angular orienations.
14940     MOVE 90,70   ! Draw old home orientation.
14950     PEN 2
14960     DRAW 90-11*SIN(Old_home_angle),70-11*COS(Old_home_angle)
14970     PENUP
14980     MOVE 90,70  ! Draw current home orientation.
14990     PEN 4
15000     DRAW 90,58
15010     PENUP
15020     DISABLE
15030     MOVE 90,70  ! Draw current target orienation.
15040     PEN -3
15050     DRAW X_pos,Y_pos
15060     MOVE 90,70
15070     PEN 3
15080     X_pos=90-11*SIN(Current_angle)
15090     Y_pos=70-11*COS(Current_angle)
15100     DRAW X_pos,Y_pos
15110     PENUP
15120     ENABLE
15130 RETURN
```

```
15140 !
15150 !-----------------------------------------------------------------------
15160 !
15170 Quit: !
15180       SUBEXIT
15190 SUBEND
15200 !
15210 !***********************************************************************
15220 !
15230 SUB Comm(C$,OPTIONAL INTEGER Confirm_answer)
15240 !
15250 !  PROGRAM MODULE:  Comm
15260 !
15270 !  PURPOSE:         Modified version of the Comm module to be used
15280 !                   for direct two way communication with the WHEDCO
15290 !                   IMC stepping motor controller.
15300 !
15310 !  UPDATE:  3.0     Version 3.0 checks to see if the card being used
15320 !                   is the HP98628A (Datacomm) or the HP98626A (Serial).
15330 !                   Depending on which card is used, the appropriate
15340 !                   registers are selected.
15350 !
15360      OPTION BASE 1
15370      COM /Status/ INTEGER Sc,Connect_flg,E_flg,Debug_flg,Response$
15380      INTEGER Baud_rate,B,Num_chars,Response_flg,Index1
15390      DIM Input$[256],Term$[256],In$[256] BUFFER,From_232$[256]
15400      DIM Num_chars$[6],Num_ltrs$[6],Out$[256] BUFFER
15410      DIM White_print$[1],Crlf$[2]
15420      IF Debug_flg THEN PRINT TABXY(1,1);"ENTERING Comm "
15430      ON ERROR GOSUB Error
15440      !
15450      !
15460      !
15470      IF Connect_flg THEN After_init
15480      Sc=20
15490      ASSIGN @Find_it TO Sc;RETURN Outcome
15500      IF Outcome=0 THEN
15510         ASSIGN @Find_it TO *
15520         CONTROL Sc,0;1                  ! Reset RS-232 interface.
15530         CONTROL Sc,3;1                  ! Async link protocol.
15540         CONTROL Sc,0;1                  ! Set Async toggle.
15550         CONTROL Sc,8;1+2                ! Set RTS and DTR lines.
15560         CONTROL Sc,16;0                 ! Disable connection timeout.
15570         CONTROL Sc,17;0                 ! Disable no activity timeout.
15580         CONTROL Sc,18;0                 ! Disable NO CARRIER timeout.
15590         CONTROL Sc,19;0                 ! Disable transmit timeout.
15600         CONTROL Sc,20;14                ! TX baud speed = 9600
15610         CONTROL Sc,21;14                ! RX baud speed = 9600
15620         CONTROL Sc,22;0                 ! No handshake with Whedco.
15630         CONTROL Sc,23;0                 ! No hardwired handshake.
15640         CONTROL Sc,34;2                 ! 7 bits/character.
15650         CONTROL Sc,35;0                 ! 1 stop bit.
15660         CONTROL Sc,36;1                 ! ODD parity.
15670         Connect_flg=1
15680      ELSE
15690         Sc=8
15700         ASSIGN @Find_it TO *
15710         ASSIGN @Find_it TO Sc;RETURN Outcome
15720         IF Outcome<>0 THEN
15730            PRINT "RS-232 card not installed. Please install and reboot."
15740            ASSIGN @Find_it TO *
```

```
15750              STOP
15760           END IF
15770           ASSIGN @Find_it TO *
15780           RESET Sc
15790           CONTROL Sc,0;1                   ! Reset the RS-232 interface.
15800           CONTROL Sc,3;Baud_rate           ! Set the baud rate.
15810           CONTROL Sc,4;8+2                 ! UART 8 bits/char.  ODD parity.
15820           CONTROL Sc,5;3                   ! UART DTR line active.
15830           CONTROL Sc,12;128+32+16          ! Disable CD,DSR,CTS
15840           STATUS Sc,3;B                    ! Confirm speed to user.
15850           Connect_flg=1
15860        END IF
15870 After_init:!
15880        White_print$=CHR$(136)
15890        Crlf$=CHR$(13)&CHR$(10)
15900        PRINT CHR$(128)&CHR$(136);      ! Set up the screen.
15910        ASSIGN @Screen TO CRT
15920        ASSIGN @Kbd TO KBD
15930        ASSIGN @Rx TO BUFFER In$
15940        ASSIGN @Tx TO BUFFER Out$
15950        ASSIGN @Uart_out TO Sc
15960        ASSIGN @Uart_in TO Sc
15970        Response_flg=0                  ! Reset command acknowledge flag.
15980        Response$=""                    ! Null out response string.
15990        !
16000        !
16010           ENABLE INTR Sc                   ! Enable interrupt on card.
16020           TRANSFER @Tx TO @Uart_out;CONT ! Enable transfer buffers.
16030           TRANSFER @Uart_in TO @Rx
16040           ON INTR Sc,FNTrap_level GOSUB Read_loop ! Process card interrupts.
16050           IF C$<>"" THEN
16060              GOSUB Send_com                ! Send command out to controller.
16070           ELSE
16080              GOTO Quit                     ! If null command, exit quick.
16090           END IF
16100           !
16110           !
16120           !
16130 Wait_for_it:WHILE NOT Response_flg         ! Waiting for acknowledgement.
16140                       GOSUB Read_loop
16150                       IF NPAR=2 THEN       ! We are waiting for data to be
16160                                            ! sent by the Whedco controller.
16170                          LOOP
16180                                GOSUB Read_loop
16190                                IF (POS(Response$,"*")) THEN
16200                                   Response$=Response$[POS(Response$,"*"),LEN(Respo
16210                                   Response_flg=1
16220                                END IF
16230                          EXIT IF ((Response_flg=1) AND (POS(Response$,Crlf$)))
16240                          END LOOP
16250                       ELSE
16260                          WHILE NOT ((POS(Response$,"*")) OR (POS(Response$,"?")))
16270                                GOSUB Read_loop
16280                          END WHILE
16290                          Index1=POS(Response$,"*")
16300                          IF Index1=0 THEN ! Must be a "?" (Whedco command error
16310                             ! Must be a "?" (Whedco command error).
16320                             E_flg=1        ! Notify via error flag.
16330                             Response_flg=1
16340                          ELSE
16350                             ! Normal command interpretation.
```

```
16360                       E_flg=0
16370                       Response_flg=1
16380                    END IF
16390                 END IF
16400           END WHILE
16410       GOTO Quit
16420      !
16430      !
16440 Read_loop:! Read in serial data from Whedco.
16450           !
16460      STATUS @Rx,4;Num_chars                ! Number of characters to
16470      IF Num_chars=0 THEN RETURN            ! receive, if 0 try again.
16480      Num_chars$="#,"&VAL$(Num_chars)&"A"   ! Set up the IMAGE for ENTER.
16490      ENTER @Rx USING Num_chars$;From_232$        ! Transfer contents.
16500      Response$=Response$&From_232$               ! Build up dialogue.
16510      RETURN                                      ! Update pointers.
16520 !
16530 !
16540 Send_com:Term$=Crlf$[1,1]&C$&Crlf$
16550         Num_ltrs$="#,"&VAL$(LEN(Term$))&"A"
16560         OUTPUT @Tx USING Num_ltrs$;Term$
16570         Term$=""
16580         RETURN
16590 !
16600 !
16610 Quit:  OFF ERROR
16620        STATUS @Tx,10;Stat
16630        STATUS @Rx,4;Num_bytes
16640        ABORTIO @Uart_out
16650        ASSIGN @Tx TO *
16660        CONTROL @Rx,8;0
16670        STATUS @Rx,10;Stat
16680        STATUS @Rx,4;Num_bytes
16690        ABORTIO @Uart_in
16700        ASSIGN @Rx TO *
16710        DISABLE INTR Sc
16720        SUBEXIT
16730 !
16740 !
16750 Error:PRINT "HANDLING Comm ERROR"
16760        IF ERRN<>167 THEN Other_error
16770        IF Sc=8 THEN ! Process the simple card.
16780           STATUS Sc,10;Uart_error
16790           IF BIT(Uart_error,1) THEN Overrun
16800           IF BIT(Uart_error,2) THEN Parity
16810           IF BIT(Uart_error,4) THEN Break1
16820           IF BIT(Uart_error,3) THEN Framing
16830           E_flg=1
16840           PAUSE
16850           RETURN
16860         ELSE
16870           PRINT ERRM$
16880           E_flg=1
16890           PAUSE
16900           RETURN
16910         END IF
16920 !
16930 !
16940 Other:  PRINT "UART error status: ";Uart_error
16950         E_flg=1
16960         RETURN
```

```
16970 !
16980 !
16990 Overrun:PRINT "Overrun"
17000          E_flg=1
17010          RETURN
17020 !
17030 !
17040 Parity: PRINT "Parity"
17050          E_flg=1
17060          RETURN
17070 !
17080 !
17090 Break1: PRINT "Break"
17100          E_flg=1
17110          RETURN
17120 !
17130 !
17140 Framing:PRINT "Framing"
17150          E_flg=1
17160          RETURN
17170 !
17180 !
17190 Other_error:PRINT "Error message: ";ERRM$
17200               PAUSE
17210               E_flg=1
17220               SUBEXIT
17230 !
17240 !
17250 SUBEND
17260 !
17270 !**************************************************************************
17280 !
```