RL 867

POLARIMETRIC MEASUREMENT OF POWERLINE CABLES

Kamal Sarabandi
Yisok Oh
Fawwaz T, Ulaby

July, 1991

POLARIMETRIC MEASUREMENT OF POWERLINE CABLES

by

Kamal Sarabandi

Yisok Oh

Fawwaz T. Ulaby

Radiation Laboratory

Department of Electrical Engineering and Computer Science

The University of Michigan

July 8, 1991

Abstract

Polarimetric backscattering measurements of a variety of powerline cables are presented in this report. The objective of this investigation is to study the effect of braiding of the cables on the backscattering at skew incidence. The measurements were performed for four different powerline samples of actual size and a large scaled prototype (made in machine shop) at C-, X-, and Ka-band over a wide range of incidence angles. The prototype cable is six times larger than the smallest cable (167.8 MCM copper cable); the size was chosen such that the radar response at higher frequencies can be simulated. The data were collected over 500 MHz bandwidth at C- and X-band with 1.25 MHz increment and 1 GHz bandwidth at Ka-band with 2.5 MHz increment. Also the effect of non-uniform illumination and measurement in the near field of the cables were studied. Experimental data shows a significant increase in the backscattering for VV-polarization (σ_{vv}) at angles away from normal incidence. This backscatter is proportional to the number and diameter of the strands on the surface of the cables. There are also noticeable increases in the HH and cross-polarized components of the scattering matrices, and their magnitudes, relative to that of the VV component, are proportional to the pitch angle of the helix.

Contents

List of Figures

List of Tables

1 Introduction

Detection and collision avoidance of obstacles of small physical cross section has always been an important problem for low-flying aircrafts. High voltage powerlines and powerline towers are particularly hazardous in this respect.

Many collision warning techniques have been suggested in the past. Among the most promising techniques are laser radar [Savan and Barr, 1988] and millimeter wave radar [Rembold et al., 1982], particularly the latter when used in the synthetic aperture imaging mode. Available methods, however, suffer from a number of shortcomings. A major limitation of laser systems is atmospheric attenuation under fog and cloudy conditions, which would hamper target detection considerably. The problem with microwave and millimeter wave radars lies in the fact that current models used for characterizing the scatter by powerlines are inadequate. The radar uses a linearly polarized wave and transmission lines are modeled as long perfectly conducting circular cylinders [Rembold, 1984]. The description imposes a significant restriction on the ability of radar to detect powerlines. The choice of frequency and polarization have not been examined in previous measurements and models in such a way as to optimize the detection of powerlines by radars. The fact that a high voltage powerline is made up of strands of wires in a helical arrangement can be exploited with regard to backscattering detection of the powerlines. At high frequencies the helical geometry of powerlines becomes an important factor influencing the scattering behavior of electromagnetic waves which can be taken advantage of in detecting powerlines at off-specular directions. The surface of the cables is periodic along the axis of the cables and usually the period is only a fraction of the helical pitch. The effect of the helicity and the periodicity of the surface in backscatter at incidence angles away from normal incidence is investigated in this report.

Detectability of targets with characteristic scattering matrices can be greatly enhanced by using polarization synthesis, which requires the radar system to have polarimetric capabilities. Polarization synthesis techniques are particularly useful for detecting targets in the presence of clutter [Ioannidis and Hammers, 1979].

In this report we present the polarimetric backscatter response of four different types of powerline cables at C-, X-, and Ka-band. Experimental data are collected over a wide range of incidence angles and for two different cable lengths. In one case the cable length is chosen such that the illumination is uniform, and in the other case the cable is long and the illumination is non-uniform. In the case of non-uniform illumination on a cable, the scattering matrix of a smooth cylinder of identical length and diameter is also measured for comparison. First the measurement setup and the radar configurations are discussed briefly and then a polarimetric calibration procedure used in support of these measurements is explained. Finally, the radar cross section measurements and significant results are given. The cable measurements are performed under two conditions: (1) short cables where the illumination is uniform and the radar is at the far field of the cable, (2) long cables where the illumination is tapered and the radar is in the near field.

2 System Configuration and Measurement Setup

The polarimetric measurements of the powerline cables were performed by C- , X-, and Ka-band scatterometers. A simplified block diagram of the scatterometer system is shown in Fig. (1). The scatterometers are HP 8753-based (8753A for C- and X-band and 8753C for Ka-band) systems 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 measurements allows us to remove the unwanted short-range signals from the desired target response (known as software gating). It also allows us to separate the contribution of different scattering points within the target under investigation. The C- and X-band scatterometers are slightly different from the Ka-band system. The C- and X-band system are single antenna radars while the Ka-band is a dual antenna system. The sequence of polarization selection, data collection, and target orientation is performed via an HP 9000 series computer (see Fig. 1). 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 [Liepa et al., 1989]. The pulsing network is not used for the Ka-band system since there is enough isolation between the transmitter and the receiver. 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 continuous wave 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 up- and down-convertors are used to transmit and receive the desired frequencies. The block diagrams of the C- and X-band microwave circuitry are given in Figs. (3) and (4), respectively. The up-convertors for the C- and X-band units are very stable microwave sources operating at 6.5 and 8 GHz respectively. The frequency range of the network analyzer is set to 1.5-2 GHz and 1.25-1.75 GHz, respectively, in order to operate the C- and X-band scatterometer at 4.5-5 and 9.25-9.75GHz. The antenna for each system is comprised of an orthomode transducer (OMT) and a dual-polarized square horn with an overall cross-polarization isolation of 20 dB.

The Ka-band scatterometer can operate in both the coherent and coherent-on-receive modes [Kuga et al., 1991]. In the coherent mode the scattering matrix of the target can be measured and in the coherent-on-receive mode the scatterometer measures the Muller matrix of the target directly. In this study the coherent mode of the scatterometer is used. The dual channel capability of the HP 8753C allows simultaneous measurements of V and H channels in Coupled/Chop mode. Point by point external triggering is used for transmitting V and H sequentially and for synchronizing a polarization control circuit to create different polarizations. The block diagram of the Ka-band system is given in Fig. 5. This scatterometer is a two-antenna system with at least 70 dB isolation between the transmitter and the receiver. The polarization selection in the transmitter is accomplished by a Faraday rotator and in the receiver by an OMT. The antennas in both the transmit and receive channels are lens-corrected corrugated circular horns. The local oscillator is a 32 GHz Gunn diode which is stabilized by an injection locking technique. Table 1 gives the important specifications of the scatterometer systems.

To measure the backscatter of the cables with a very good signal to background ratio for all incidence angles, the cables were mounted on a styrofoam pedestal in an anechoic chamber. The correct position of the cables with respect to the antenna coordinate system was accomplished by an azimuth-over-elevation positioner as depicted in Fig. (6). The azimuth turntable 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 obtain accurate measurements of the scattering matrices of the cables, the measured data must be calibrated. In this study the scatterometers are calibrated using the single-target calibration technique (STCT) [Sarabandi and Ulaby, 1990]. The error in measurement of the scattering matrix using this technique is less than 0.5 dB in magnitude and 5° in phase. With 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 can be shown that the measured scattering matrix of a target with scattering matrix $\mathbf{S}$ 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 $\mathbf{R}$ and $\mathbf{T}$ matrices are the receive and transmit channel imbalances, C is the antenna cross-talk contamination factor, and $\mathbf{N}$ 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 target (cable), respectively, by m_{ij}^o and m_{ij}^u, the unknown scattering matrix elements of the cables can be obtained from

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

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 $a \triangleq \frac{m_{12}^o m_{21}^o}{m_{11}^o m_{22}^o}$ and the branch of the square root is chosen such that $\text{Re}\left[\sqrt{1-a}\right] > 0$. STCT is developed for single antenna radars, thus its application in the Ka-band scatterometer is questionable. However, measurements of targets with known scattering matrices with the Ka-band system show that STCT is capable of improving the cross-polarization isolation of the antenna system.

4 Experimental Results

High voltage transmission line cables are usually constructed from a number of aluminum strands twisted around a central core of one or more steel strands in a helical fashion. The number of layers and diameter of aluminum strands determines the current capacity of the cable. In a distribution network, however, low tension and high current cables are used which are made of copper or aluminum strands. Four different types of powerline cables were acquired for this experiment and their electrical and geometrical specifications are listed in Table 2. Also the geometry of the cross section of the powerline samples are depicted in Fig. 7. A photograph of the powerline samples and the anechoic chamber is shown in Figs. 8-10. The important parameters of the cables, as far as electromagnetic scattering is concerned, are the outer surface geometry such as the cable diameter, diameter of each strand, the pitch of the helices, and the surface period along the axis of the cable as shown in Fig. 11.

The powerline cables are targets of extended length, that is the length of the cables, in all practical situations, extends beyond the footprint of the illumination area. Thus in reality the cables are illuminated with a tapered wave and the radar is in the near field of the target. In this case the radar echo is a function of not only the geometry but also the radiation pattern and the distance. In contrast, the far field measurement reveals both the main scattering features and a unique quantitative value for the radar echo independent of the radiation pattern. In this study we use cables with two different lengths, short and long. For short cables uniform illumination and far field criteria are satisfied and for long cables the illumination is non-uniform and the radar is in the near field of the cables. The measured radiation pattern and radar cross section of the long cable are compared with those measured for smooth cylinders of similar lengths and diameters to study the effect of braiding on the cables. A series of scattering matrix measurements of powerline cables and smooth cylinders is performed at C-, X-, and Ka-band over the frequency ranges 4.5-5 GHz, 9.25-9.75 GHz, and 34-35 GHz, respectively. The sequence of automatic data collection is explained in Appendix A. All the data presented in this report are measurements at the center frequencies, namely 4.75 GHz for C-band, 9.5 GHz for the X-band, and 34.5 GHz for Ka-band.

The scattering matrix measurements were performed in an 18-meter long anechoic chamber at a distance of 13 meters. The targets are mounted on a styrofoam pedestal to get a minimal background contribution. The styrofoam pedestal is practically invisible at C- and X-band but at Ka-band its radar cross section is considerable (about -30 dBsm). To remove the contribution of the background and the effects of multiple reflection within the radar system, the chamber and pedestal are measured in the absence of targets and then subtracted from the target response. A 30.5-cm and a 4.45-cm metallic spheres are used for C-, X-band

and the Ka-band system as calibration targets, respectively. The signal to noise ratio was better than 30 dB in all cases.

The cables are positioned in the horizontal plane (H-plane) of the antenna system and the radiation patterns are measured in the principal plane, that is the axis of rotation is perpendicular to the horizontal plane. The accurate orientation of the cables with respect to the antenna system is achieved by a stepper motor positioner with an accuracy of a fraction of a tenth of a degree. This cable orientation is suitable for radar systems mounted on low-flying aircrafts where the powerlines are in the horizontal plane. In detection of powerlines using synthetic aperture radars, however, the results of this experiment may not be directly applicable. When a powerline cable is not in the principal plane of the radar antenna, the circular symmetry of the powerline cables can be used to obtain its scattering matrix from the scattering matrix of the cable in the principal plane. Figure 12 shows the relative positions of a radar system and a powerline. The plane of incidence, which includes the cable axis and the direction of incidence, intersects the antenna polarization plane, which includes the vertical and horizontal directions of the antenna, along a unit vector $\hat{h}'$. If the antenna coordinate frame is rotated by an angle ψ such that $\hat{h}$ and $\hat{h}'$ coincide, then the horizontal principal plane and the plane of incidence would coincide also for which the scattering matrix is known. For known values of polar direction of the cable axis (ϕ) and incidence angle (θ), the coordinate frame rotation angle (ψ) can be obtained from

$$\psi = \tan^{-1}\left[\cos\theta \tan\phi\right]. \tag{1}$$

The scattering matrix of the cable in this situation **S**, in terms of the scattering matrix of the cable in the principal plane, $\mathbf{S}_p$, can be obtained from

$$\mathbf{S} = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \mathbf{S}_p \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix}. \tag{2}$$

We are now in a position to present the measured data and let us start with the C-band measurements. Figure 13-16 shows the radar cross section of the 30-cm-long cables #1 through #4 . Measured RCS of a 30-cm-long smooth cylinder with a diameter similar to that of cable #1 is also shown in Fig. 17 for comparison. The co-polarized components (VV, HH) of the scattering matrix are more or less similar to those of the smooth cylinders except for cable #4 where σ_{vv} is slightly higher than σ_{hh}. At incidence angles beyond 50 degrees scattering is dominated by the end cap of the cables. There is also a rather significant decrease in σ_{vv}/σ_{hh} of cables, when compared with the same ratio for smooth cylinders at normal incidence (see Table 3 for RCS of the smooth cylinders). The reason for this phenomena is that the surface of the cables shows two different surface impedances due to the corrugations and the surface impedance is higher for vertical polarization (TE case) than for horizontal polarization (TM case) [Sarabandi and Ulaby, 1991]. The decrease in the ratio of the co-polarized terms is inversely proportional to the diameter of the cables. Figures 18-27 give the near field measurements of the cables and their smooth cylinder counterparts. The fifth cable is a 1.2-meter-long prototype cable, made in the machine shop, whose cross sectional dimensions are six times larger than those of cable #1. RCS measurements of this cable at C- and X-band simulate the response of cable #1 at 28.5 GHz and 57 GHz, respectively. The near field effect reduces the RCS at normal incidence, widens the RCS pattern, and enhances the increase in σ_{vv} and σ_{hh}. These effects can be seen better by comparing the backscattering data of cable #4 with three different lengths: 30 cm, 120 cm, and 180 cm, as shown in Figs. 16, 24, and 28, respectively. The RCS response of cable #5 in all polarizations has a peak at approximately 15° from the backscatter direction (see Fig. 26). This peak is one of the Bragg modes in backscatter. For a periodic target the Bragg backscatter occurs for incidence angles according to

$$\theta_n = \sin^{-1}\left(\frac{n\lambda}{2L}\right) \tag{3}$$

where λ is the wavelength and L is the period (Fig. 11). Since most of the scattered energy is in the specular direction, the higher order Bragg modes are very weak at angles away from normal incidence.

Figures 29-32 gives the backscattering cross section of 30-cm-long cables at X-band and Fig. 33 shows the measurement of the cylinder counterpart of cable #1. The near field measurements of the cables and cylinders are also shown in Figs. 34-43. Similar scattering features, but more pronounced than those observed for C-band, exist in the X-band data. For the prototype cable in this case two of the Bragg modes are observable, one around 7° and the other around 15°. The important scattering feature of interest is the increase in σ_{vv} at angles away from normal incidence. This characteristic for cables #3 and #4 are shown more clearly in Figs. 45 and 46 with a finer angular resolution of 0.25°. The increase in σ_{vv}

can be interpreted as follows. The surface of a cable can be viewed as a cylinder with a number of parallel slanted narrow grooves on the surface. The inclination angle of the grooves is equal to the pitch angle of the helices which is around 15°. The backscatter from a metallic groove for TE case (electric field perpendicular to the groove axis) is much stronger than the TM case (electric field parallel to the groove axis) [Senior et al., 1990], because in the TE case the groove is capable of supporting TEM wave and in the TM case the groove is incapable of supporting any waveguide mode. Since the grooves are almost parallel to the axis of the cylinder the backscatter for VV-polarization is much stronger.

Figures 47-50 show the RCS of the 30-cm-long cables. In these cables only one of the Bragg modes is observable and as in the lower frequencies, the backscatter for VV-polarization is significantly higher than the other components. The effect of curvature on cable #2 has also been tested and the result is shown in Fig. 51 for a curved cable with radius of curvature of about 10 meters. The curvature lowers the RCS of the cable at normal incidence and Bragg directions and widens the scattering patterns. Also the near field measurement of cable #4 and its cylinder counterpart are shown in Figs. 52 and 53, respectively.

The backscattering measurements are performed polarimetrically so that the target response to any desired transmit and receive polarization configuration can be synthesized [van Zyl and Ulaby, 1990]. Obviously for detection purposes the desired polarization is the one which maximizes the target response. Since the general scattering feature of all the cables away from normal incidence is similar, polarization synthesis of cable #4 is considered only. Figures 54, 55, and 56 show the co- and cross-polarized signatures of the cable at 4.75 GHz and 19° , 9.5 GHz and 43.4° , and 34.5 GHz and 24.5° respectively. Also Fig. 57 shows the polarization signatures of the cable at the Bragg angle and 34.5 GHz. All the polarization signatures closely resembles the polarization signature of a vertical dipole. Thus the optimum polarization for detection, as expected, is VV-polarization when the cable is in the H-plane of the antenna system. If the cable were not in the H-plane, then the optimum polarization would still be a linear polarization with rotation angle ψ as given by (1).

5 Conclusions

Near field and far field polarimetric backscattering measurements of four powerline cable samples are presented at C-,X-, and Ka-band over a wide range of incidence angles. The near field measurements of smooth cylinders with a length and diameter similar to those of the cables are also performed for comparison. The experimental results from the backscatter measurements of cables and cylinders indicate that:

1. At normal incidence there is a significant decrease in σ_{vv}/σ_{hh} of the cables in comparison with the ratio for the smooth cylinders. This decrease is inversely proportional to the diameter of the cables.

2. At low frequencies and for small diameter cables ($D/\lambda \ll 1$) there is no significant backscatter at angles away from normal incidence.

3. The cross-polarized component of the cable backscatter is considerablely high over the entire range of incidence angles. The cross-polarized RCS level is directly proportional to the cable and the cable strand diameters.

4. When the cable diameter is comparable to the wavelength there is a considerable increase in σ_{vv} in off-specular directions and this increase depends on the diameter of the cable strands.

5. At millimeter wavelengths the Bragg modes are observable at angles close to normal incidence ($\theta^i < 15°$). The VV-polarized backscatter for the Bragg modes is much higher than the other polarizations.

6. Measurement in the near field of the cables reduces the RCS at normal incidence, widens the RCS pattern, and enhances the increase in σ_{vv} at angles away from normal incidence.

7. Curvature in cables lowers the RCS of the cables at normal incidence and Bragg directions and also widens the scattering pattern.

8. The optimum polarization to detect powerline cables is VV-polarization provided the cable is in the H-plane of the antenna. The optimum polarization in other cases is still a linear polarization and the rotation angle of the optimum polarization is a function of the relative orientation of the powerline with respect to the antenna coordinate system.

Acknowledgement

This research was performed under sub-contract to Ressler Associates, Inc. (301) 206-3232, under DARPA/SBIR contract number DAAH01-91-C-R066.

References

[1] Ioannidis, G.A and D.E. Hammers, "Optimum antenna polarization for target discrimination in clutter," *IEEE Trans. Antennas Propagat.*, vol. 27, no. 5, May 1979.

[2] Kuga, Y., K. Sarabandi, A. Nashashibi, F.T. Ulaby, and R. Austin, "Millimeter wave polarimetric scatterometer systems: measurement and calibration techniques," *Proc. AGARD, Electromag. Wave Propag. Panel 48 Symp.*, Montreal, May 1991.

[3] 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.

[4] Rembold, B., H.G. Wippich, M. Bischoff, and W.F.X. Frank, "A MM-wave collision warning sensor for helicopters," *Proc. Military Microwave.*, pp. 344-351, London, 1982.

[5] Rembold, B., "Radar cross section of long wires," *IEEE Trans. Antennas Propagat.*, vol. 32, no. 10, Oct. 1984.

[6] Senior, T.B.A., K. Sarabandi, J.Natzke, "Scattering by a narrow gap", *IEEE Trans. Antennas Propagat.*, vol. 38, no. 7, 1102-1110, July 1990.

[7] Sarabandi, K., and F.T. Ulaby, " High frequency scattering from corrugated stratified cylinders", *IEEE Trans. Antennas Propagat.*, vol. 39, no. 4, 512-520, April 1991.

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

[9] Savan, M. and D. N. Barr, "Reflectance of wires and cables at 10.6 micrometer," Center for Night Vision and Electro-Optics, MSEL-NV-TR-0063, Jan. 1988.

[10] 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.

[11] van Zyl, J. J. and F.T. Ulaby,<u>Radar Polarimetry for Geoscience Applications</u> (ed. F.T. Ulaby and C. Elachi), Artech House, Dedham MA, 1990.

	C-band	X-band	Ka-band
Center Frequency (GHz)	4.75	9.5	34.5
Bandwidth (GHz)	500	500	1000
Antenna Gain (dB)	25.3	29.5	32.5
Beamwidth (deg.)	9	6	5
Far field distance (m)	5.8	10.5	5.5
Cross-pol. isolation (dB)	25	20	25
Output power (mw)	100	100	100
Minimum detectable target at 15 m (dBsm)	-35 (co-pol.) -60 (cross-pol.)	-45 (co-pol) -65 (cross-pol.)	-45 (co-pol.) -65 (cross-pol.)

Table 1: System specifications of the scatterometers.

No.	Circular Mils (MCM)	Copper # St.	Copper d (cm)	D (cm)	Current Capacity (Amps)	P (cm)	L (cm)
1	167.8	7	0.400	1.20	420	14.60	1.95

No.	Circular Mils (MCM)	Aluminum # St.	Aluminum d (cm)	Steel # St.	Steel d (cm)	D (cm)	Current Capacity (Amps)	P (cm)	L (cm)
2	556.5	19	0.446	-	-	2.22	730	23.5	1.52
3	954	54	0.337	7	0.337	3.01	1010	35.56	2.00
4	1431	45	0.446	7	0.301	3.52	1300	40.65	2.40

Table 2: Electrical and geometrical specifications of the cables under test.

	4.75 GHz		9.5 GHz		34.5 GHz	
Diam. (cm)	vv	hh	vv	hh	vv	hh
1.27	-2.08	-0.68	-0.73	1.73	6.75	6.72
2.22	0.08	1.27	3.66	3.77	9.17	9.08
3.15	-0.03	2.45	5.25	5.15	10.57	10.58
3.49	1.73	2.81	5.38	5.57	11.03	11.02
7.62	5.05	5.92	8.61	8.84	14.4	14.4

Table 3: Radar cross section (dBsm) of smooth cylinders of 1 meter long.

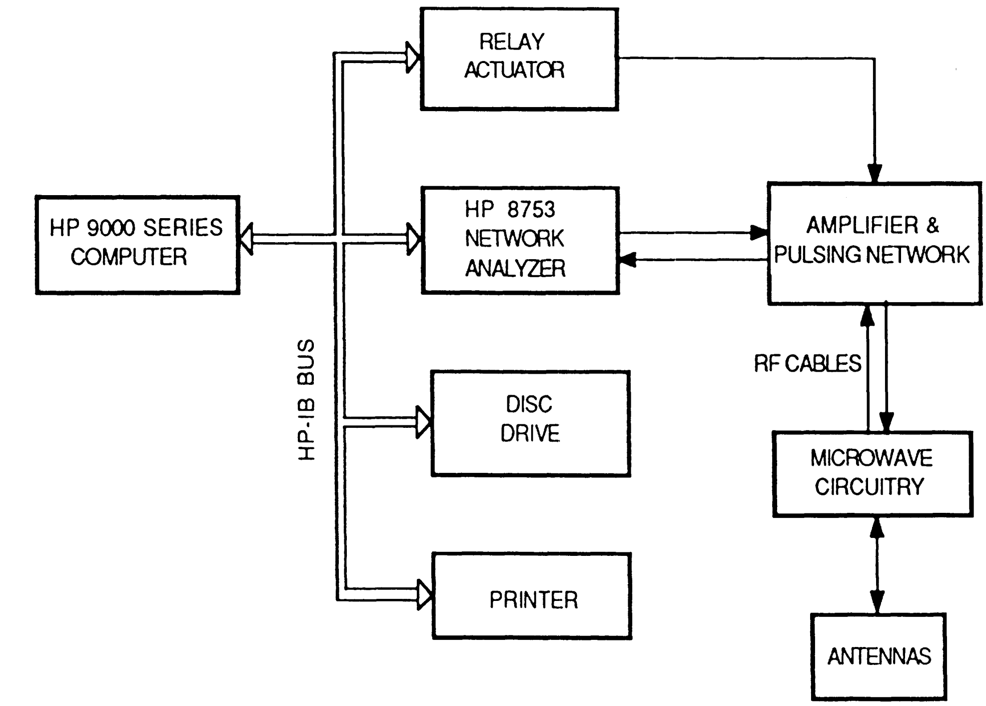


Figure 1: Block diagram of the C- and X-band scatterometer system.

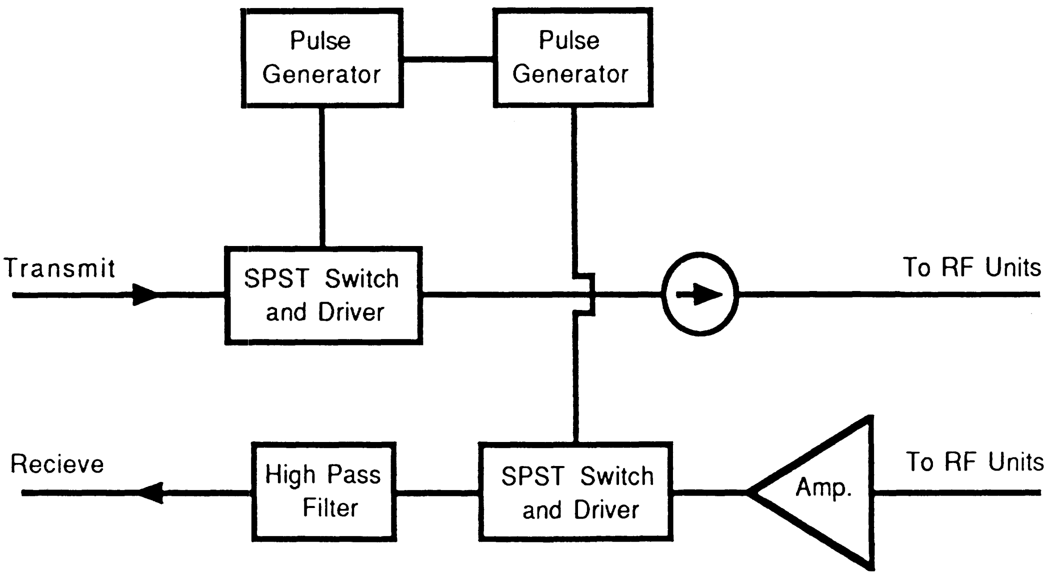


Figure 2: Block diagram of the pulsing network.

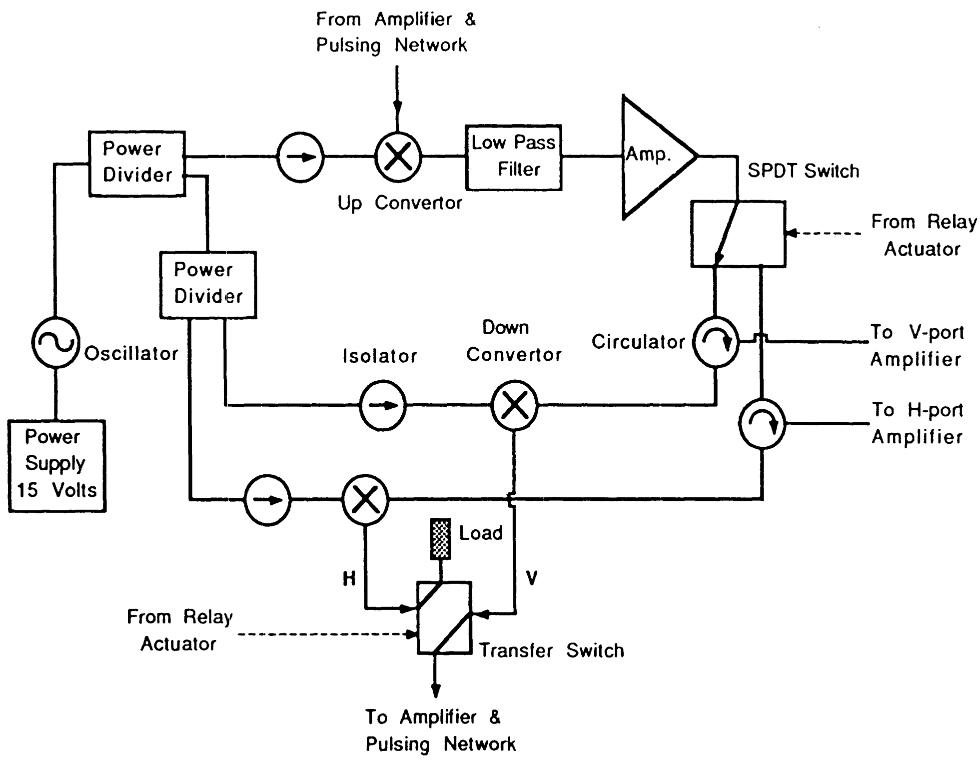


Figure 3: Block diagram of the C-band RF circuitry.

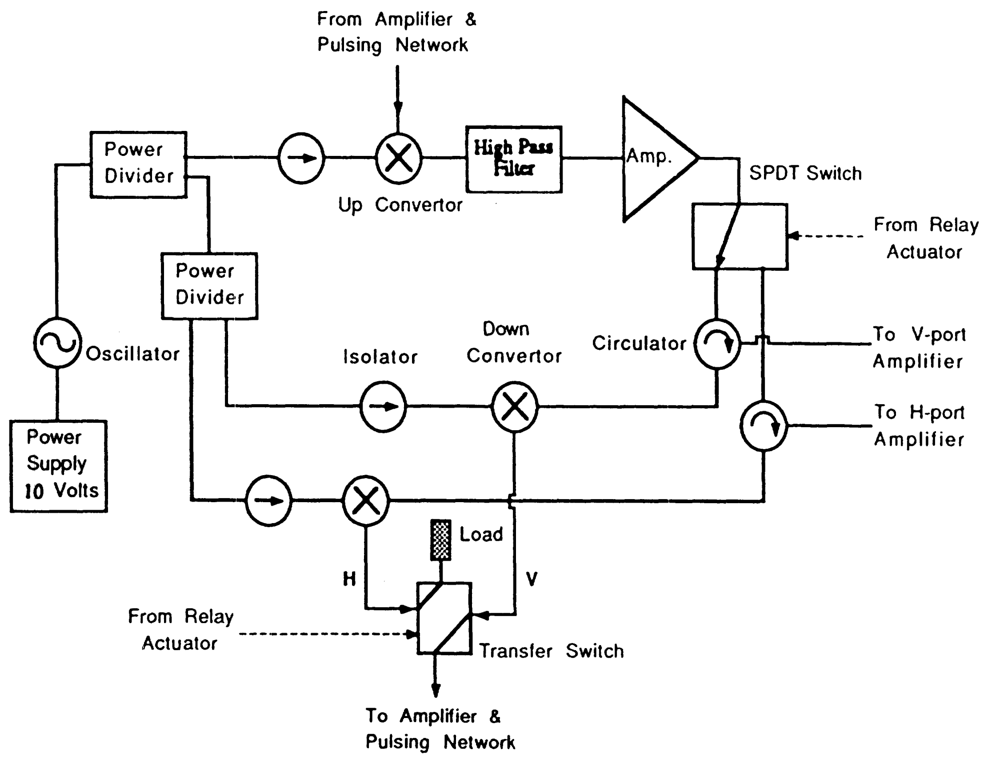


Figure 4: Block diagram of the X-band RF circuitry.

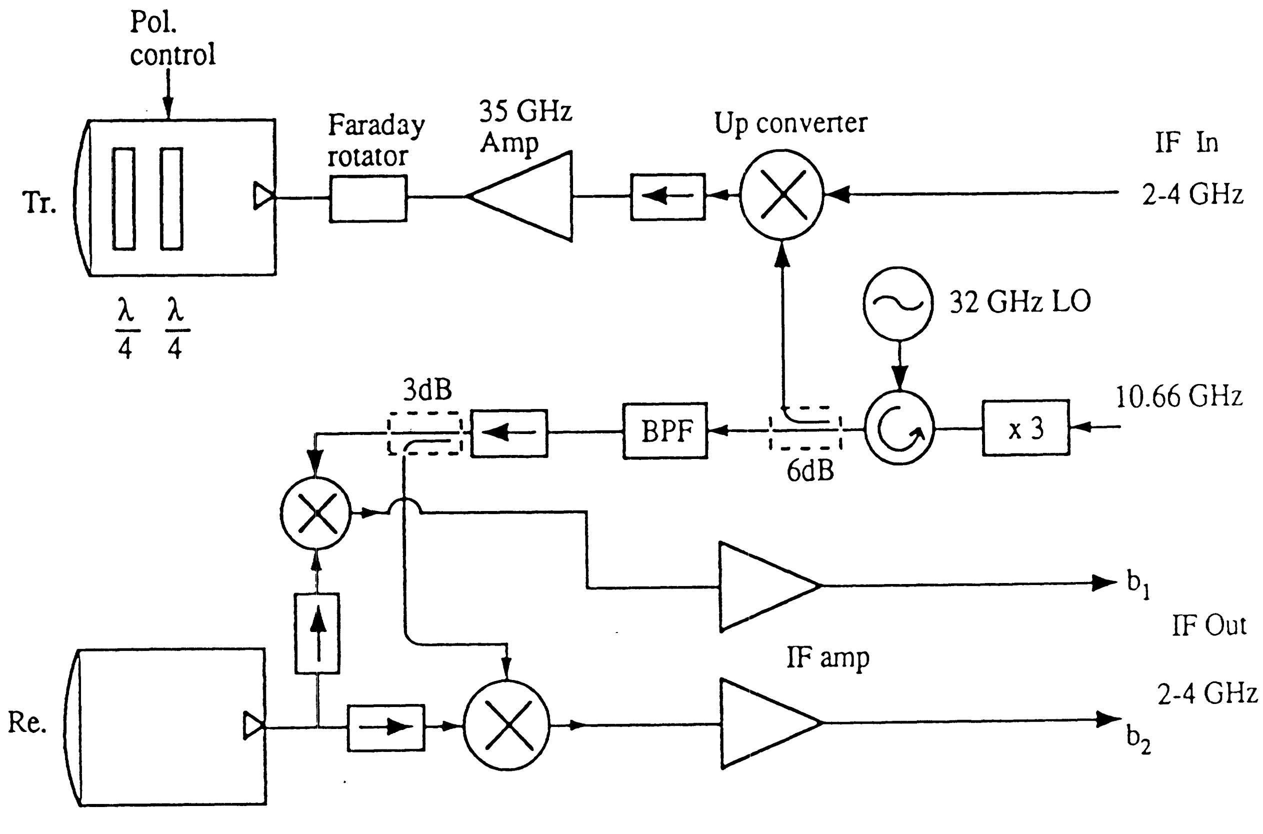


Figure 5: Block diagram of the Ka-band RF circuitry.

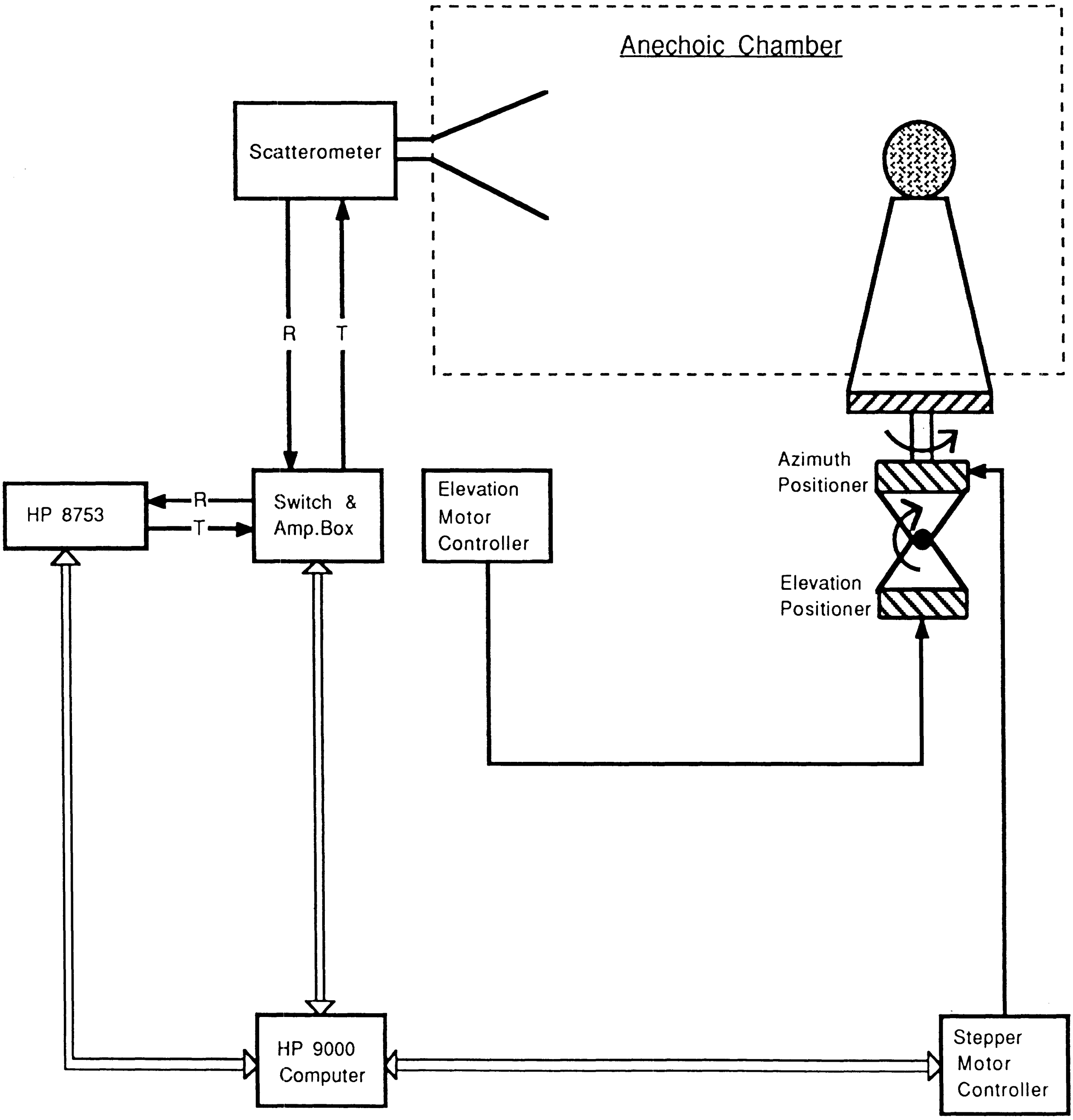


Figure 6: Automatic radar cross section measurement setup.

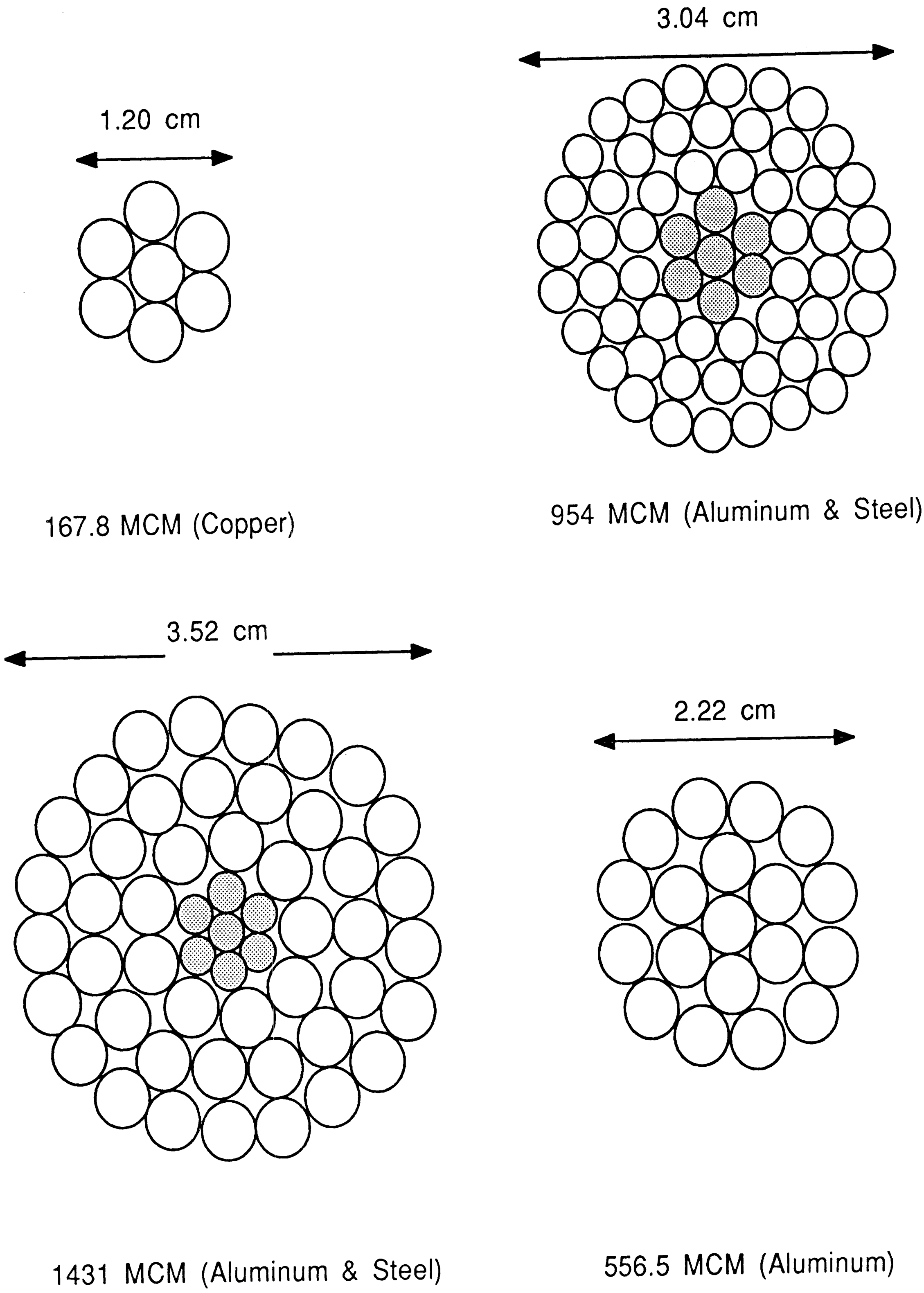


Figure 7: Cross section of powerline samples.

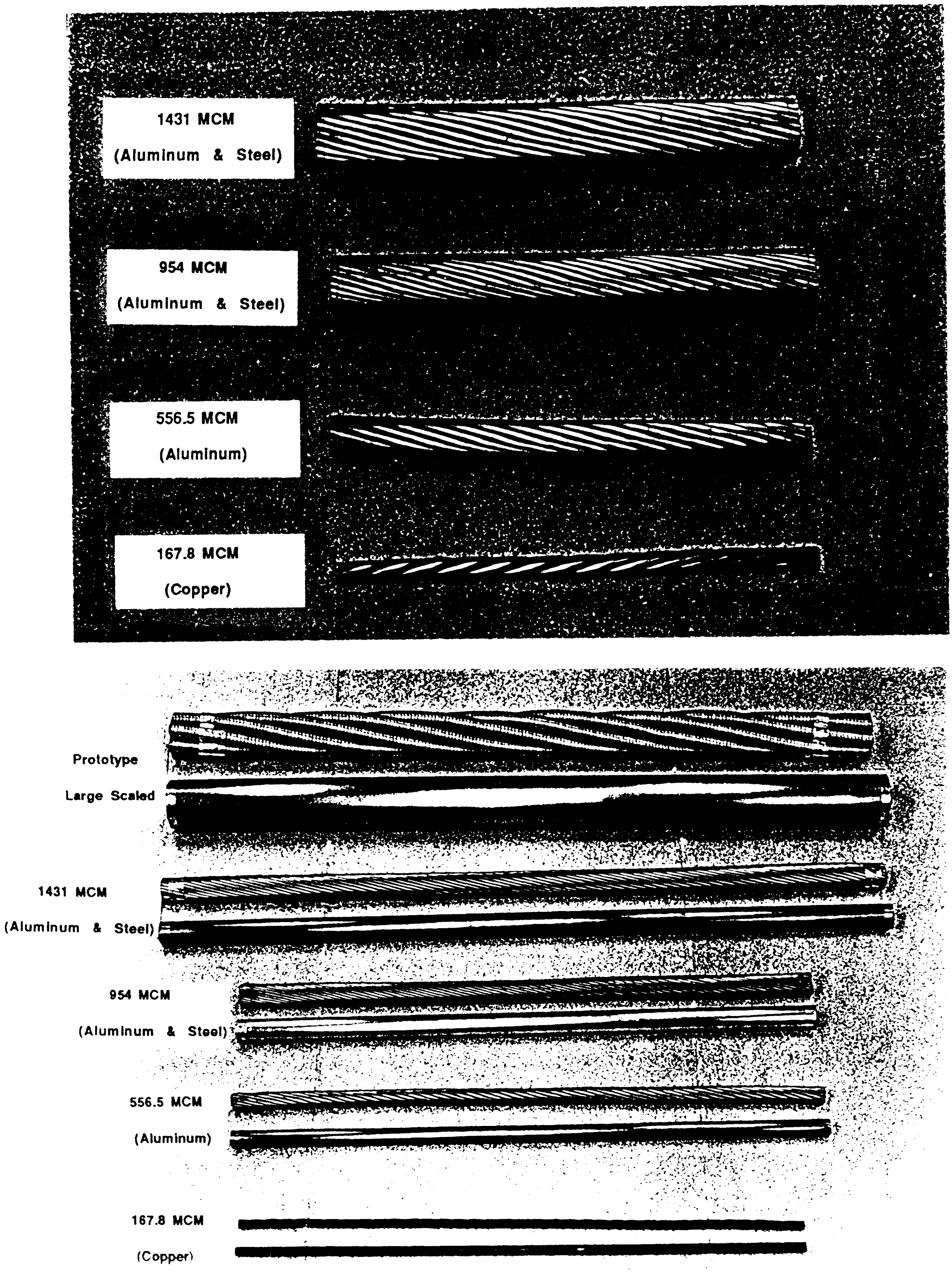


Figure 8: Photograph of powerline samples and their smooth cylinder counterpart.

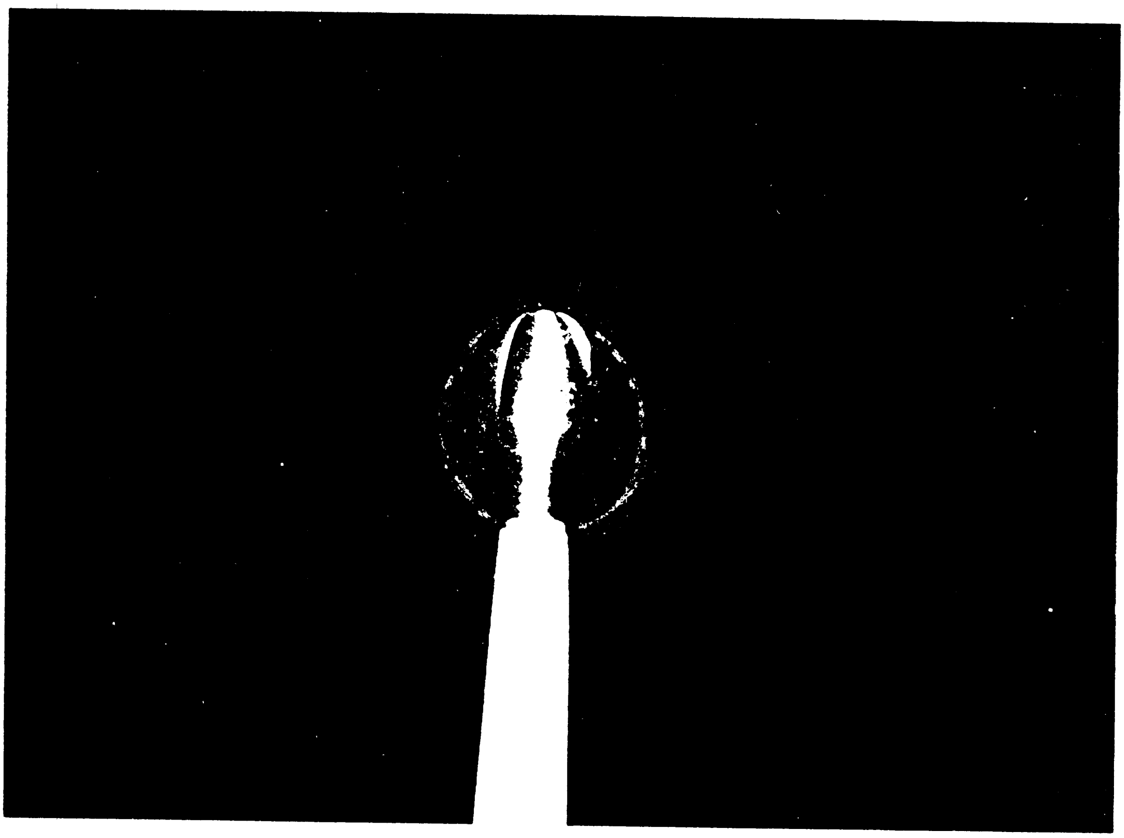


Figure 9: Photograph of calibration sphere in the chamber.

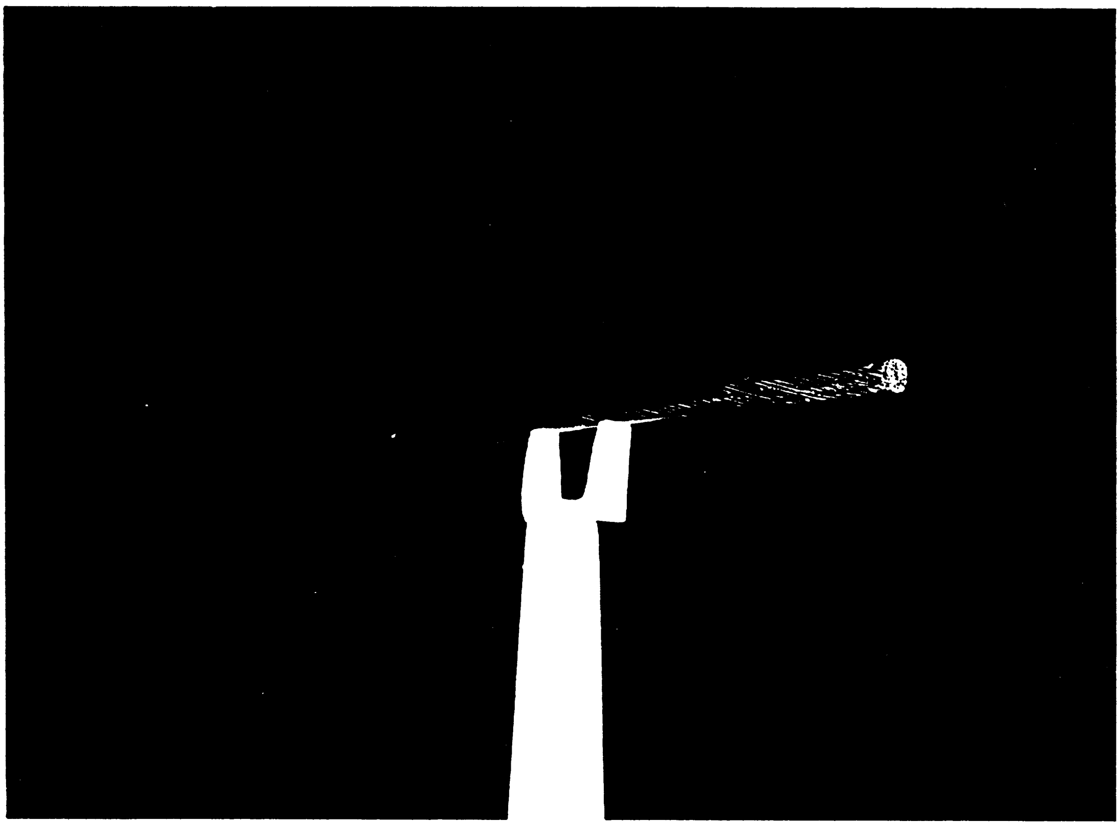


Figure 10: Photograph of cable samples in the chamber.

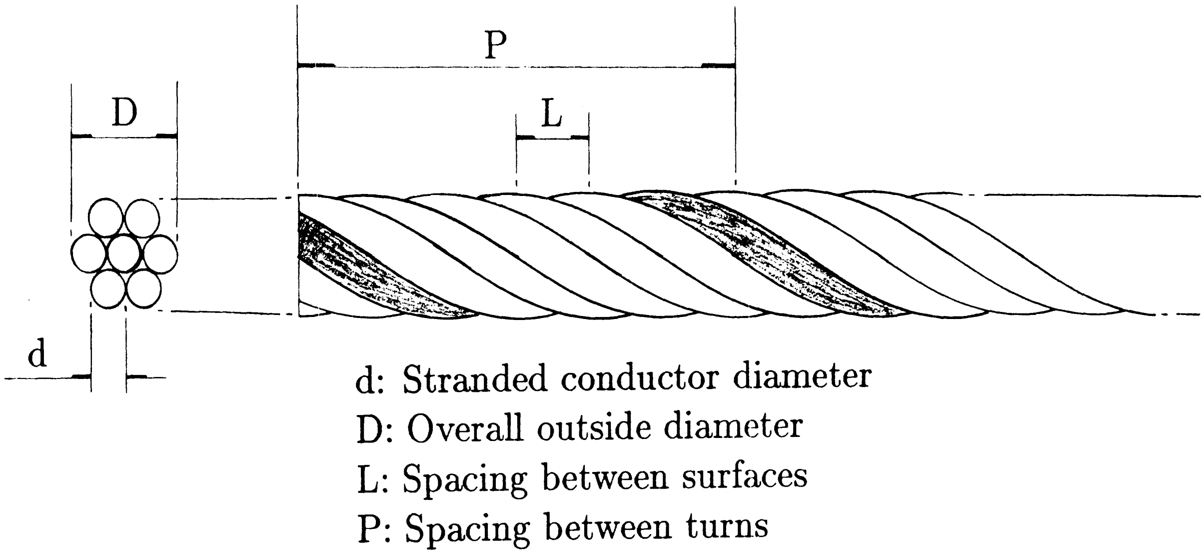


Figure 11: Geometry of a powerline cable.

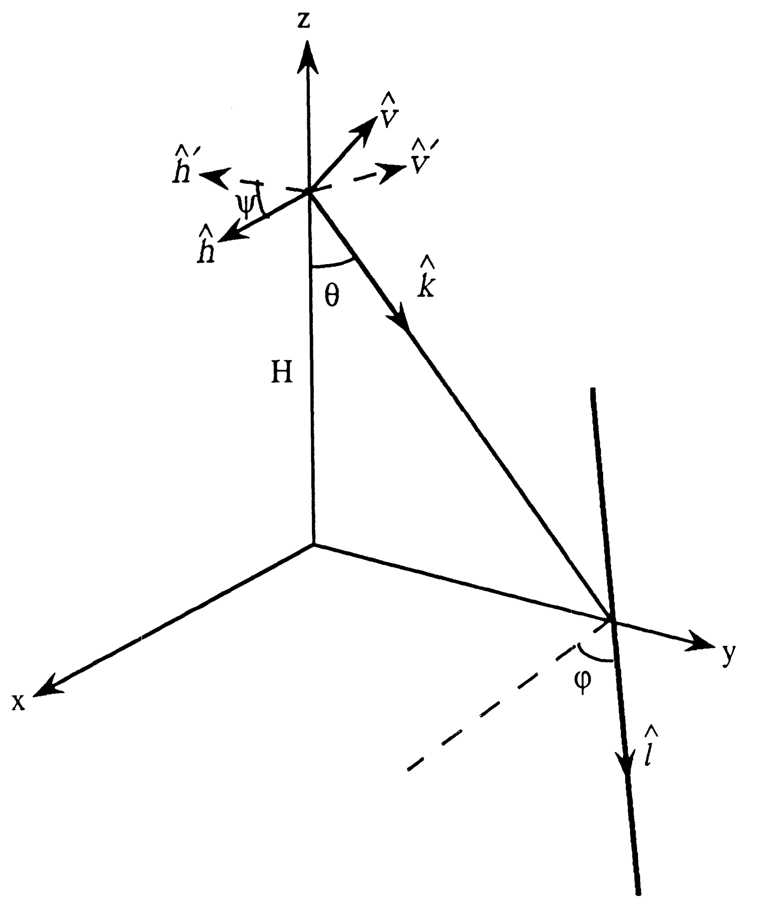


Figure 12: Geometry of a radar system above a powerline cable.

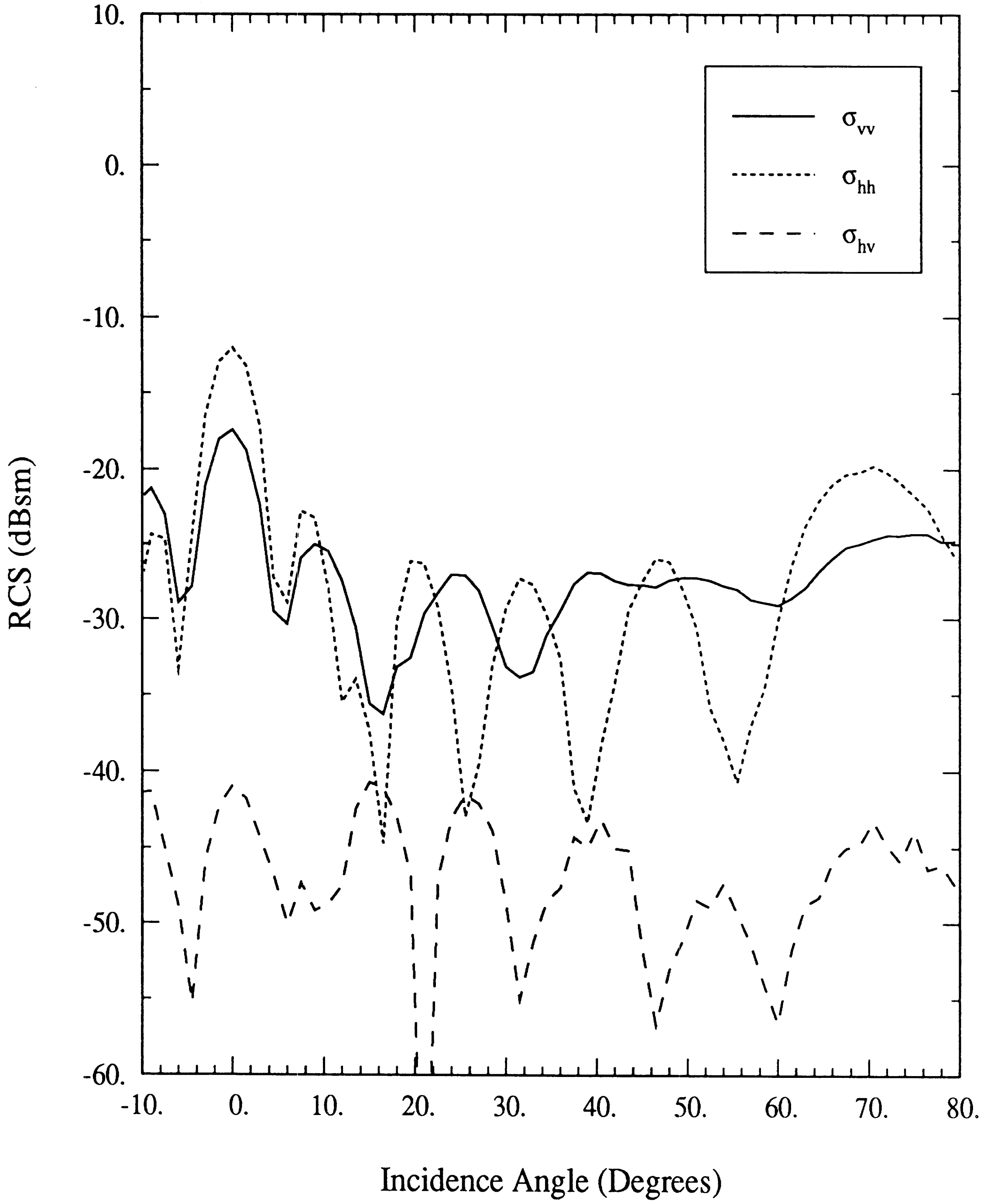


Figure 13: The radar backscatter cross section of 30-cm-long cable #1 at 4.75 GHz versus incidence angle.

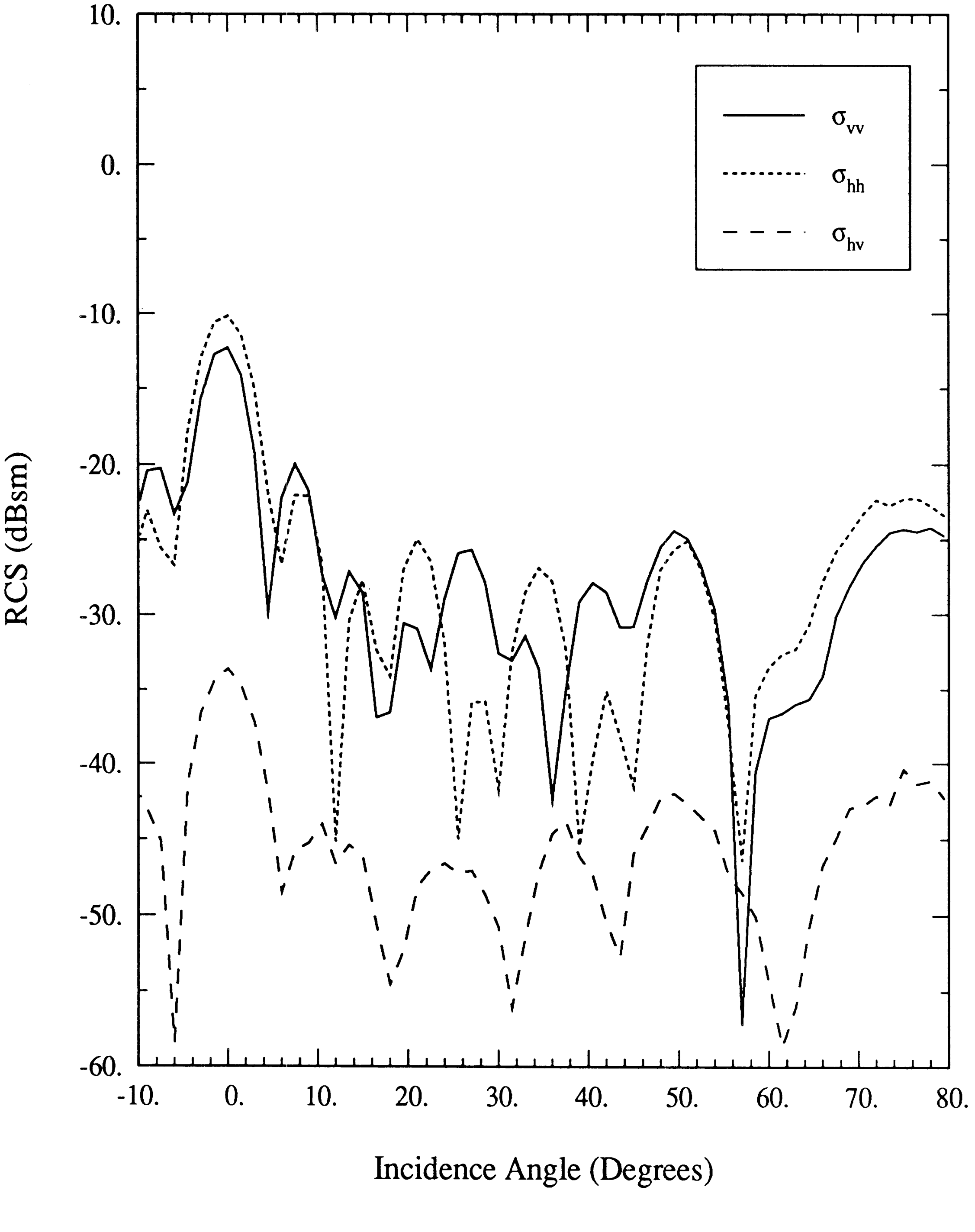


Figure 14: The radar backscatter cross section of 30-cm-long cable #2 at 4.75 GHz versus incidence angle.

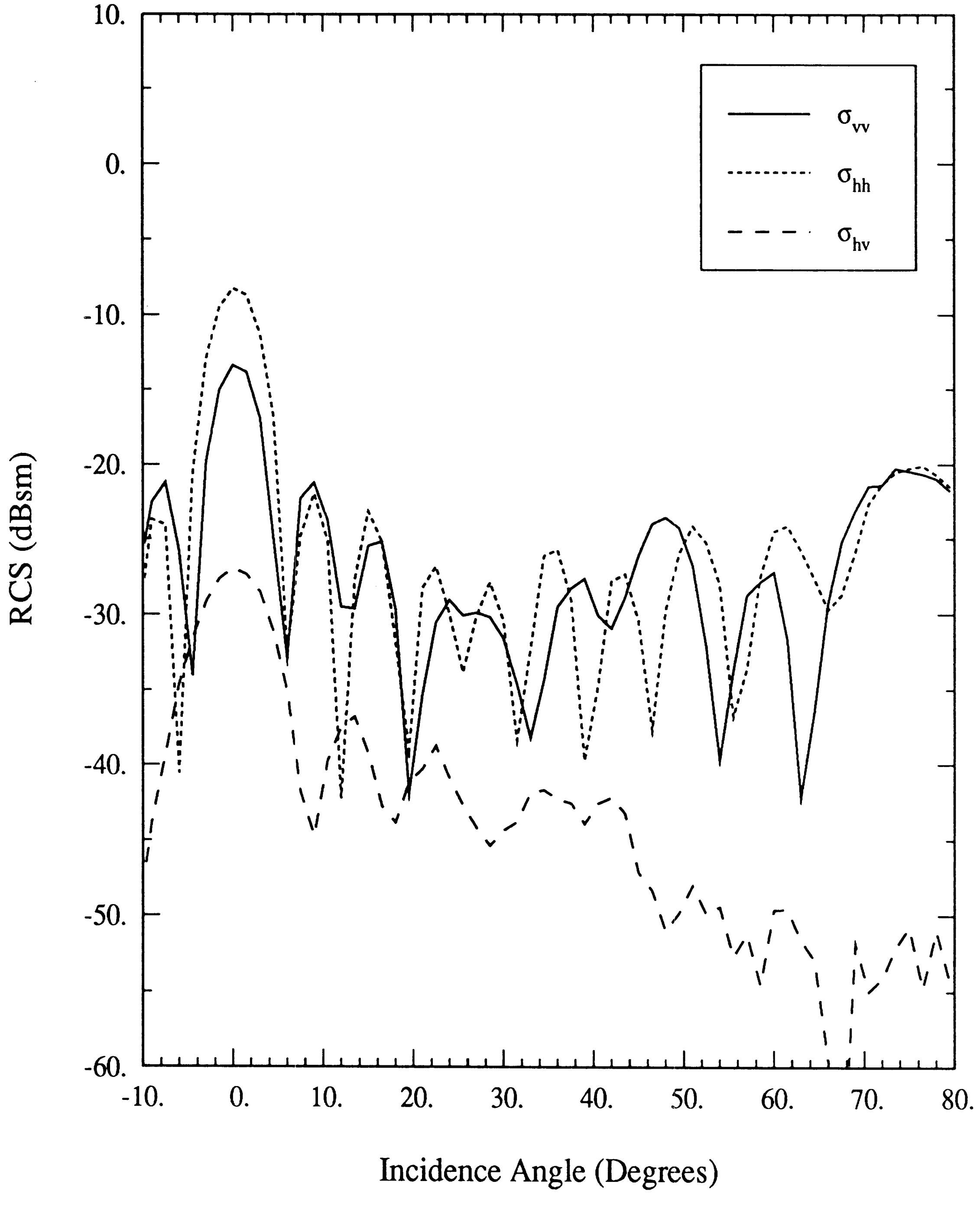


Figure 15: The radar backscatter cross section of 30-cm-long cable #3 at 4.75 GHz versus incidence angle.

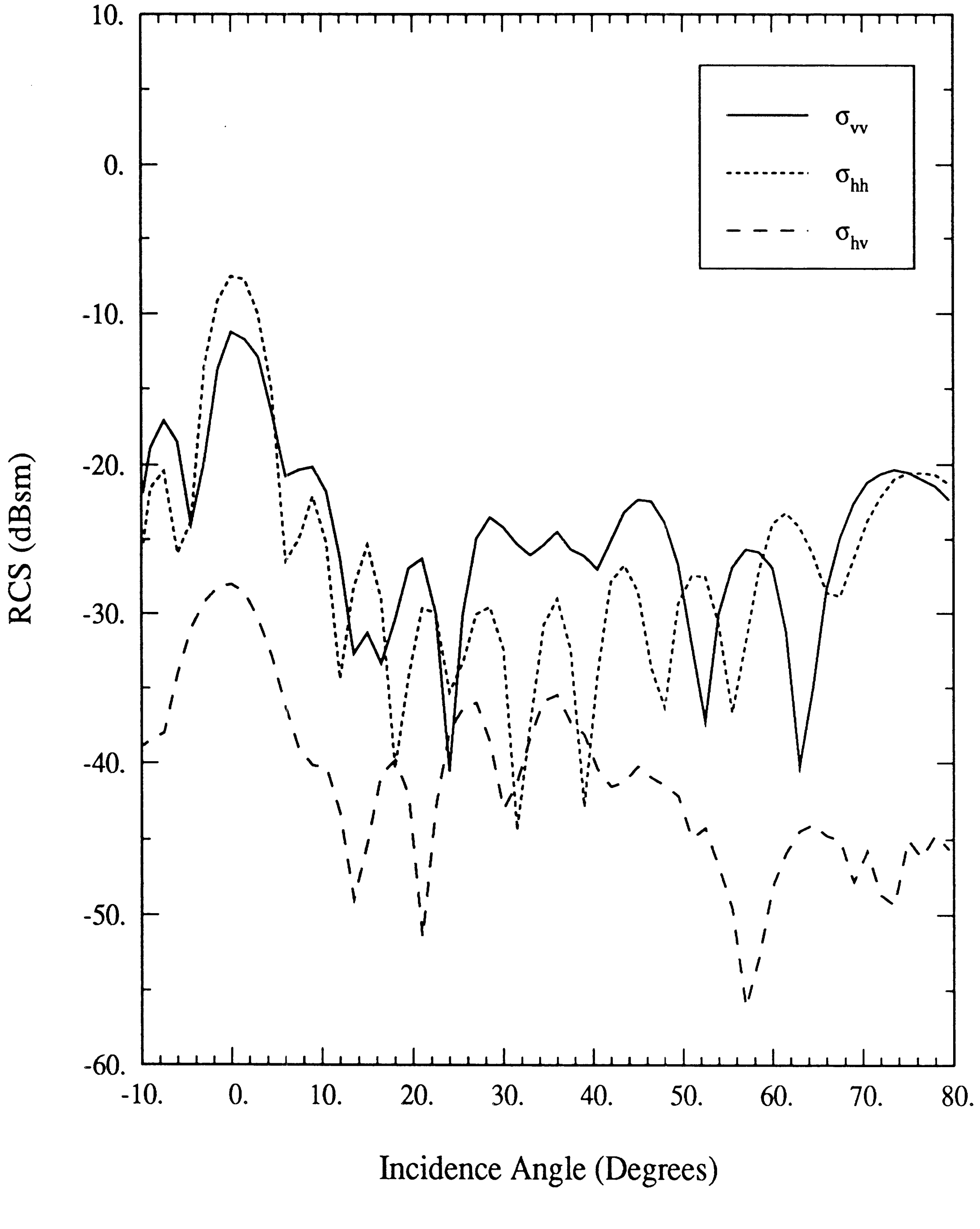


Figure 16: The radar backscatter cross section of 30-cm-long cable #4 at 4.75 GHz versus incidence angle.

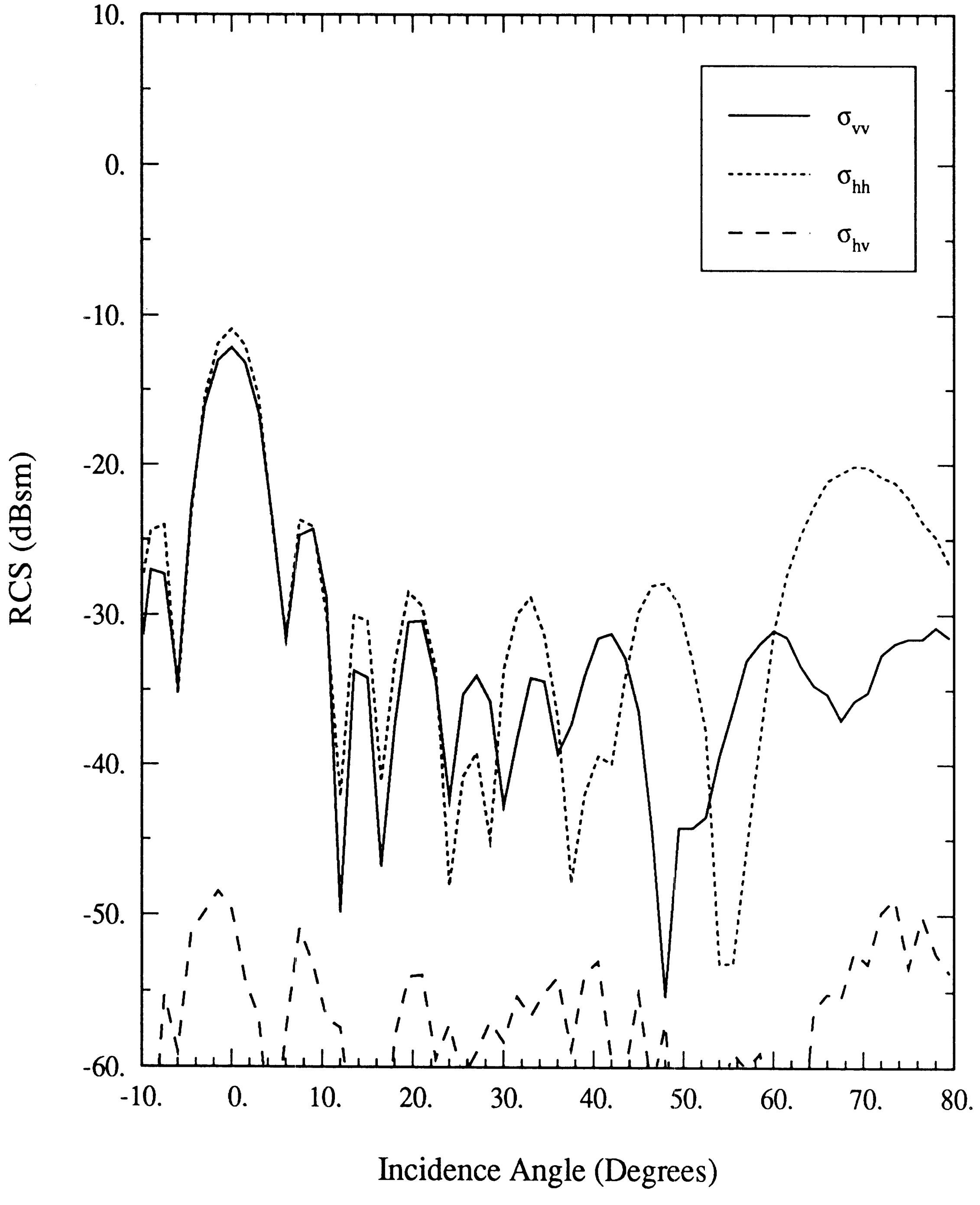


Figure 17: The radar backscatter cross section of 30-cm-long smooth cylinder of diameter 1.27 cm at 4.75 GHz versus incidence angle.

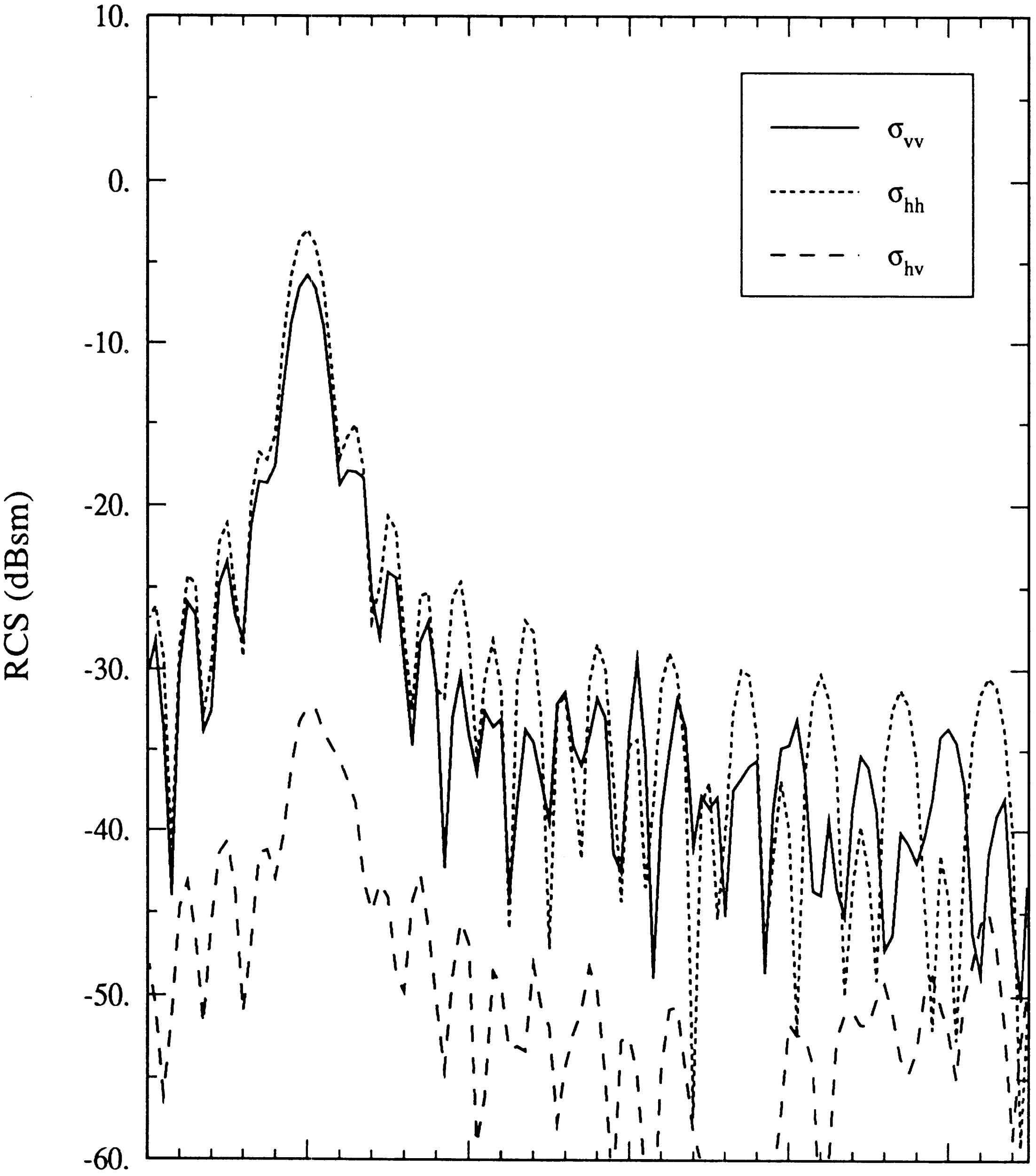


Figure 18: The radar backscatter cross section of 84-cm-long cable #1 at 4.75 GHz versus incidence angle.

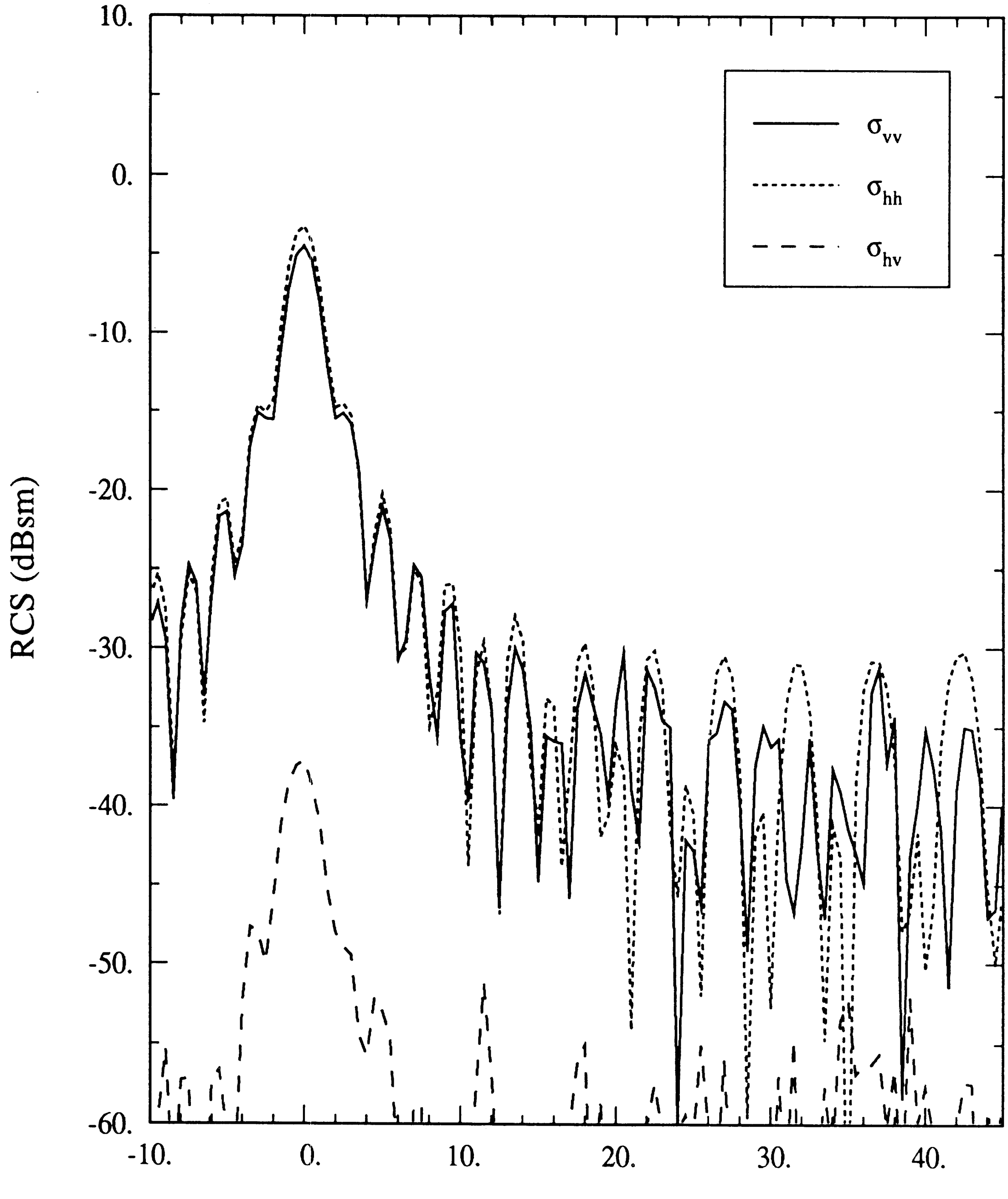


Figure 19: The near field radar backscatter cross section of 84-cm-long smooth cylinder of diameter 1.27 cm at 4.75 GHz versus incidence angle.

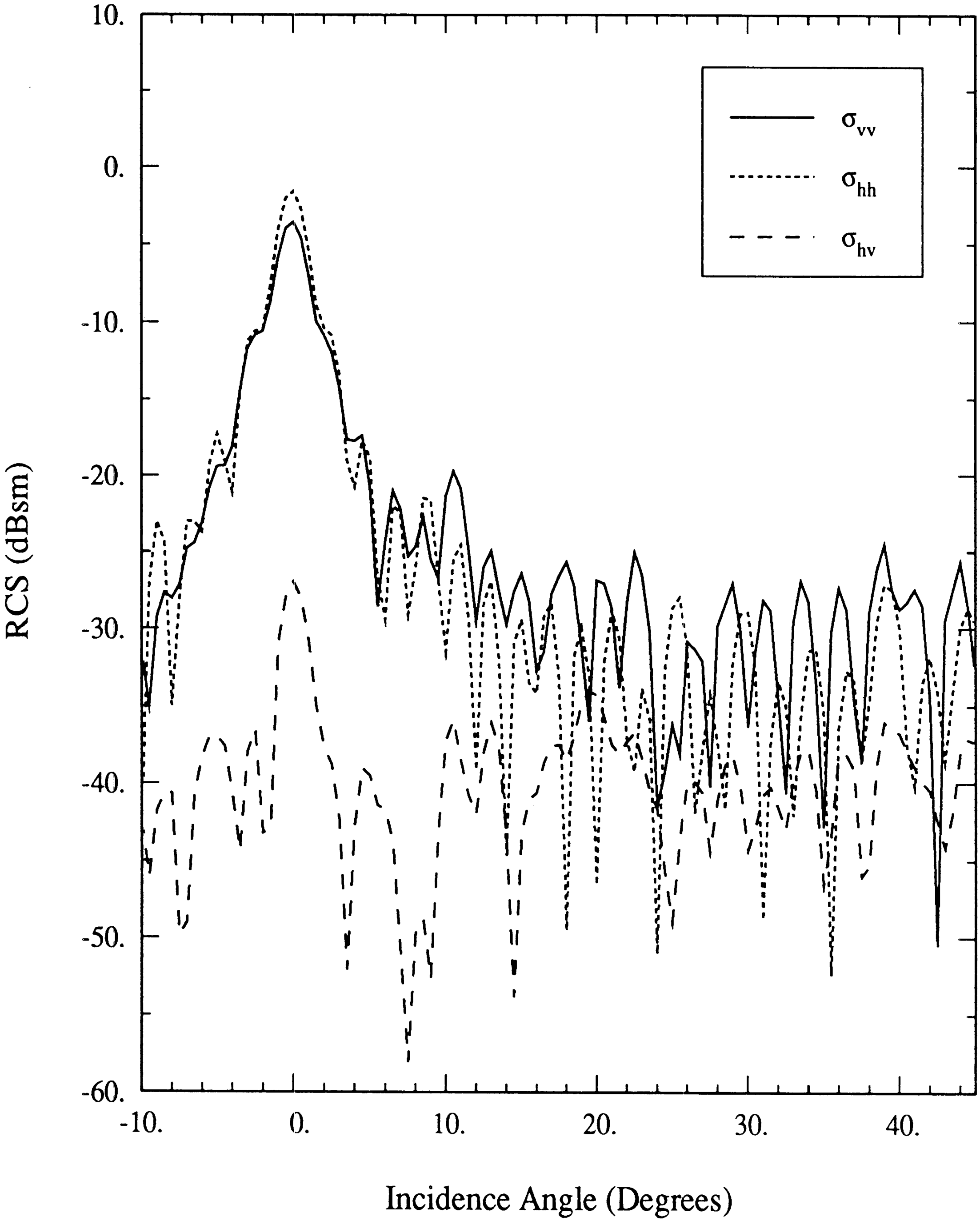


Figure 20: The near field radar backscatter cross section of 90-cm-long cable #2 at 4.75 GHz versus incidence angle.

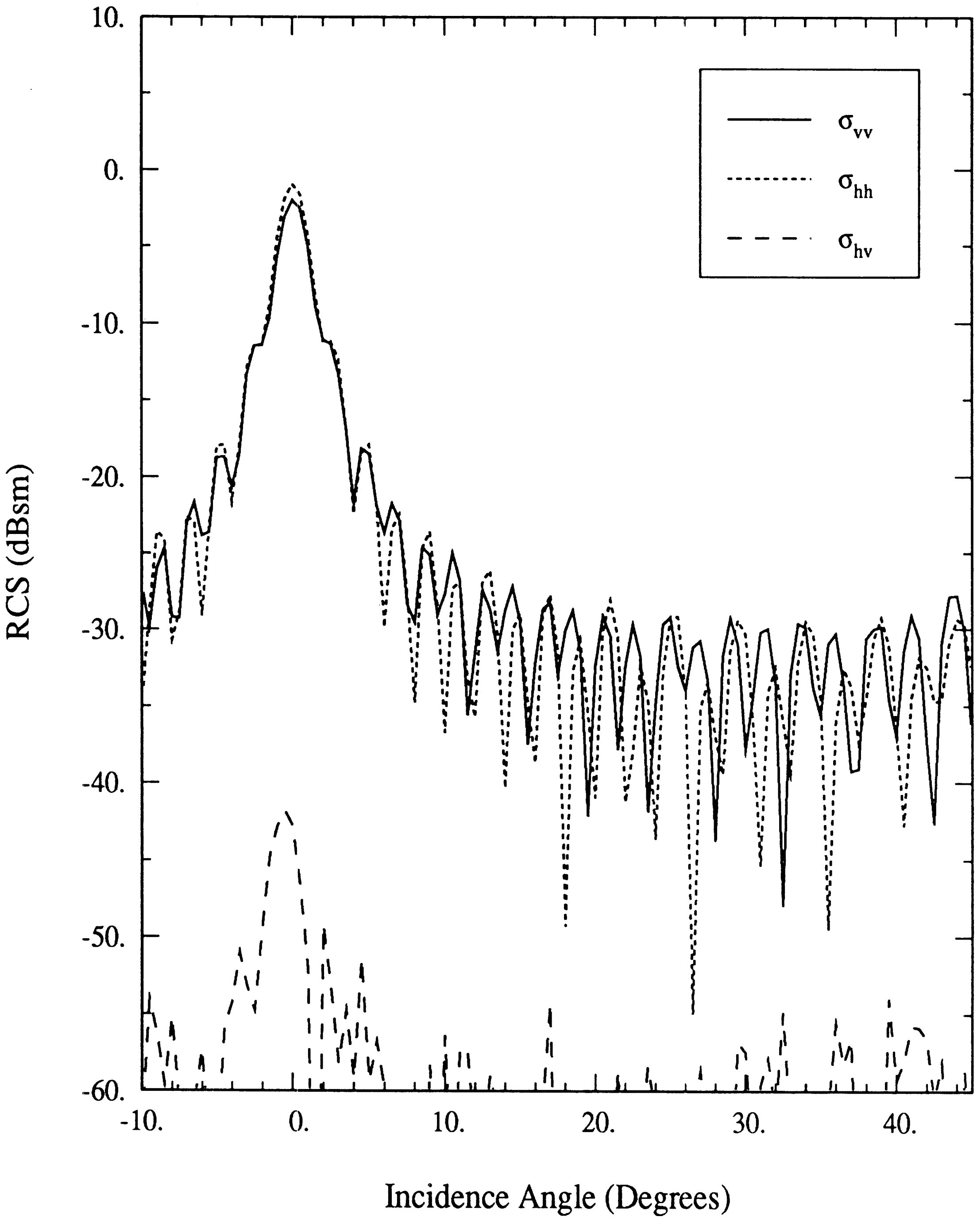


Figure 21: The near field radar backscatter cross section of 90-cm-long smooth cylinder of diameter 2.22 cm at 4.75 GHz versus incidence angle.

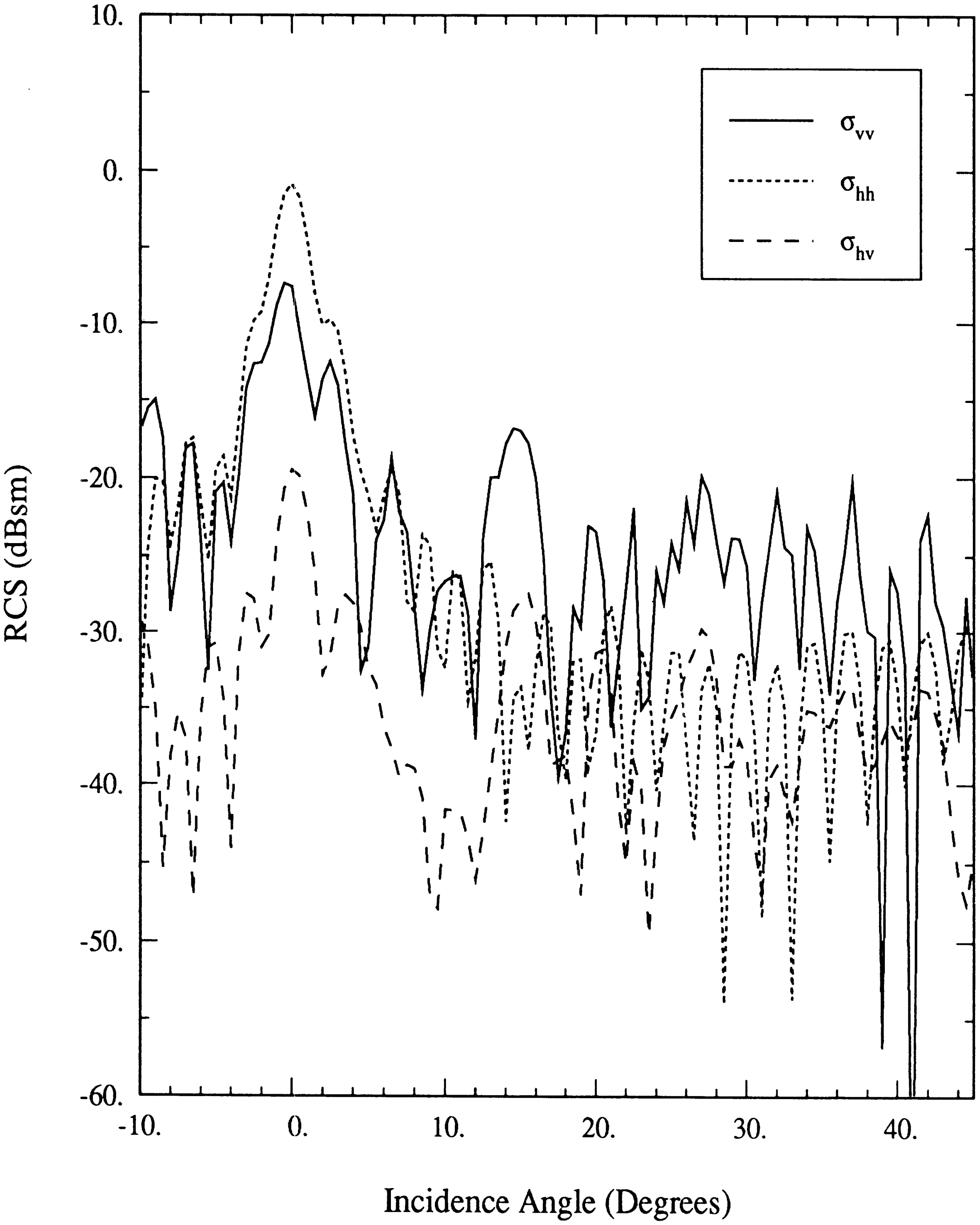


Figure 22: The near field radar backscatter cross section of 90-cm-long cable #3 at 4.75 GHz versus incidence angle.

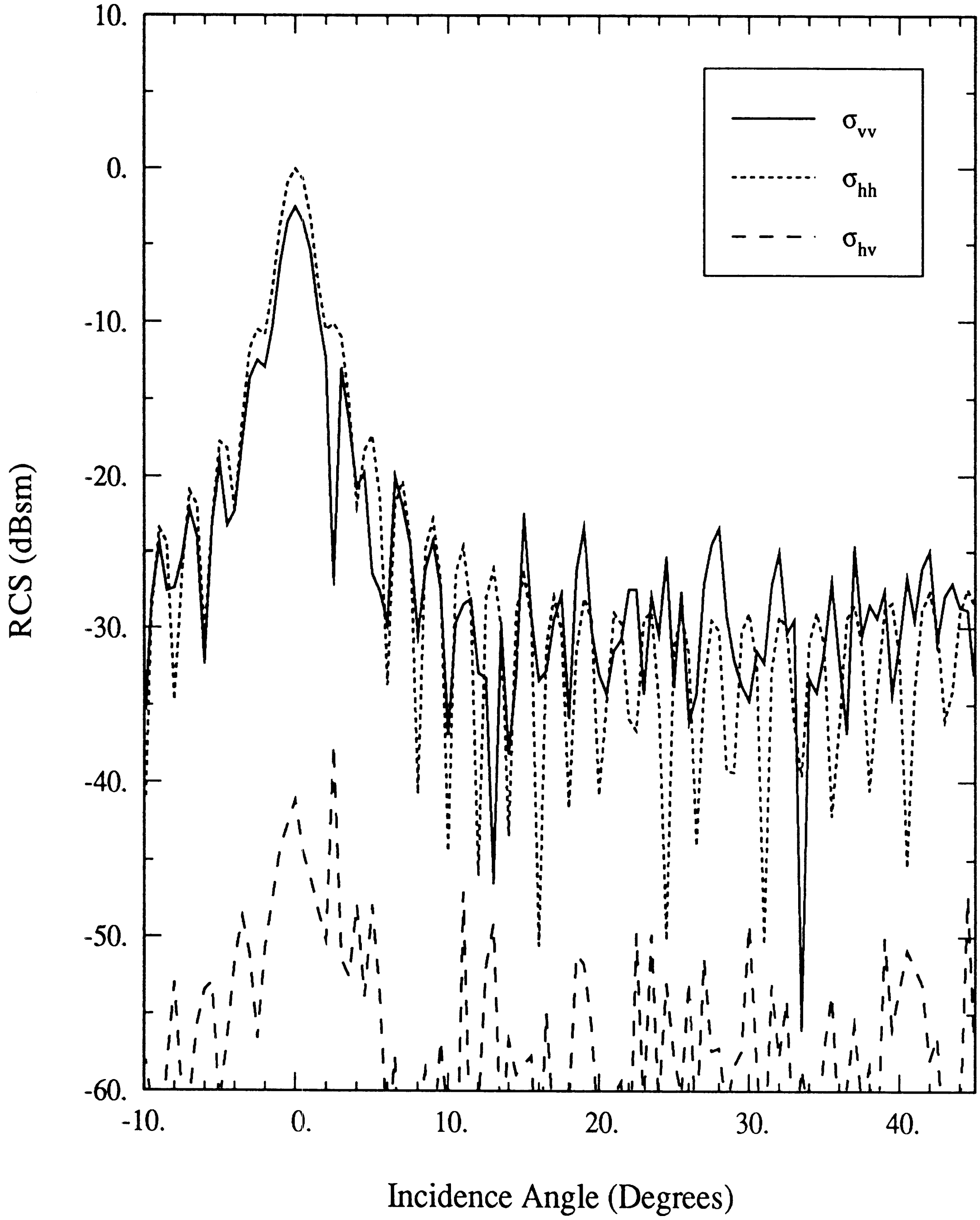


Figure 23: The near field radar backscatter cross section of 90-cm-long smooth cylinder of diameter 3.15 cm at 4.75 GHz versus incidence angle.

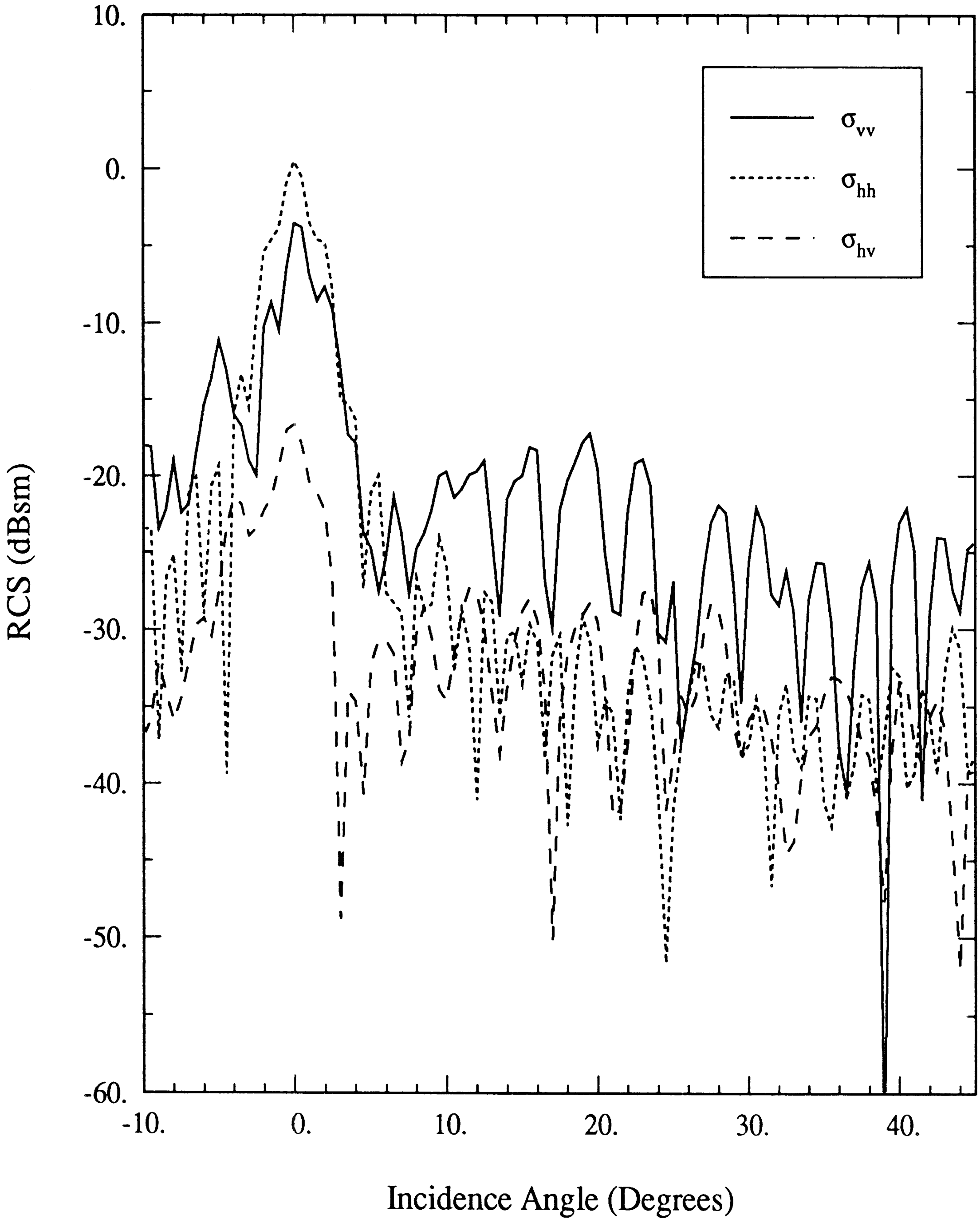


Figure 24: The near field radar backscatter cross section of 120-cm-long cable #4 at 4.75 GHz versus incidence angle.

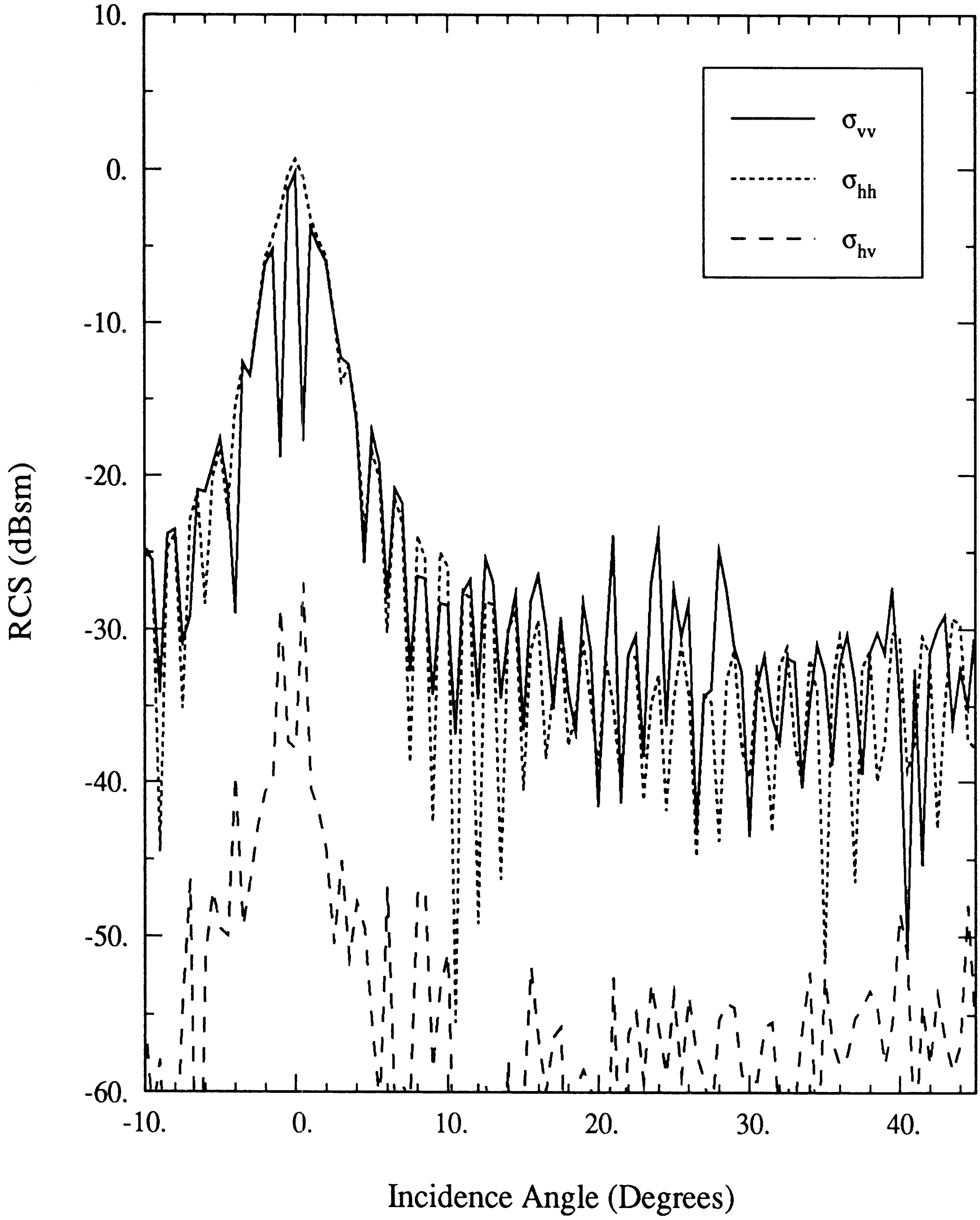


Figure 25: The near field radar backscatter cross section of 120-cm-long smooth cylinder of diameter 3.49 cm at 4.75 GHz versus incidence angle.

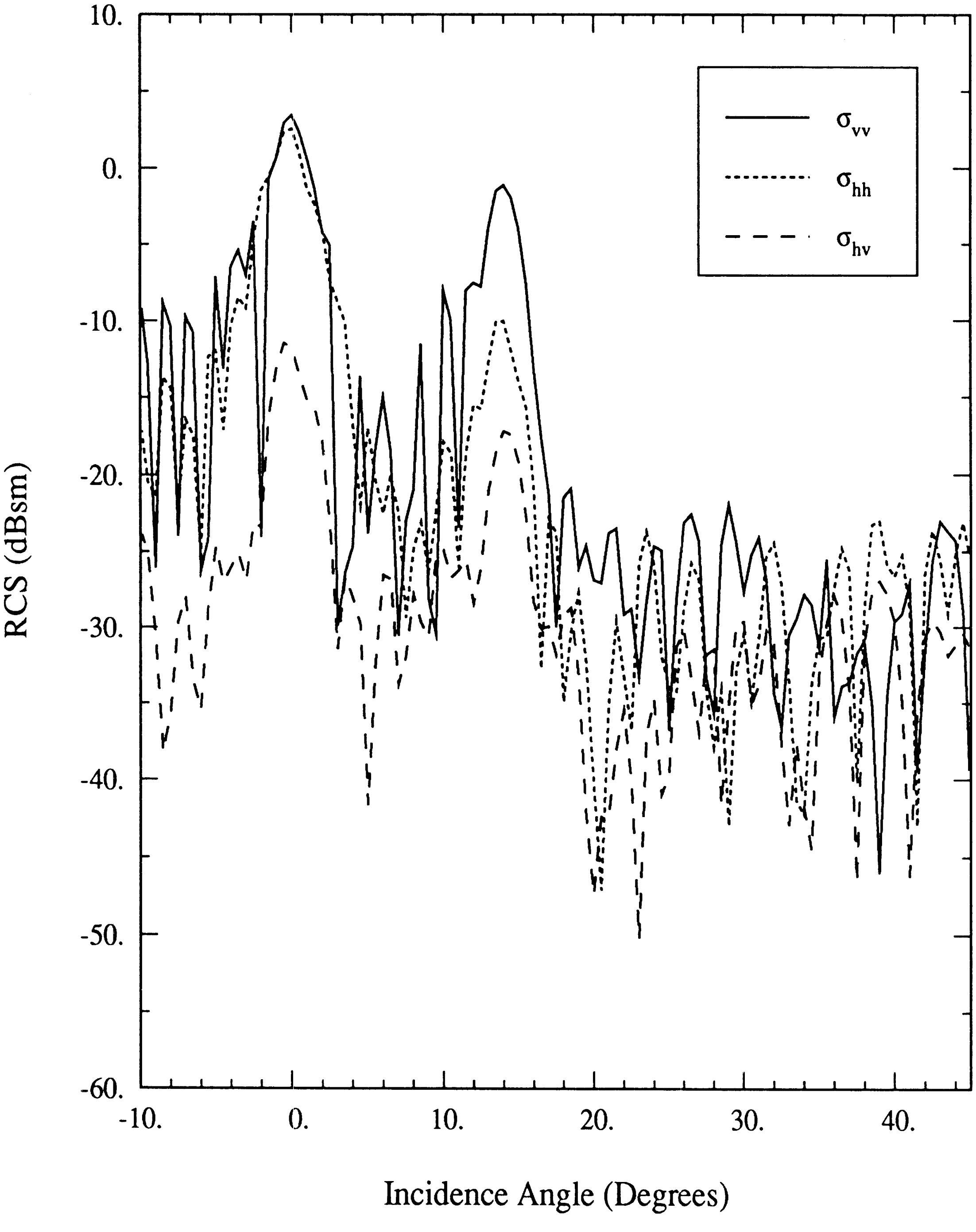


Figure 26: The near field radar backscatter cross section of 120-cm-long prototype cable at 4.75 GHz versus incidence angle.

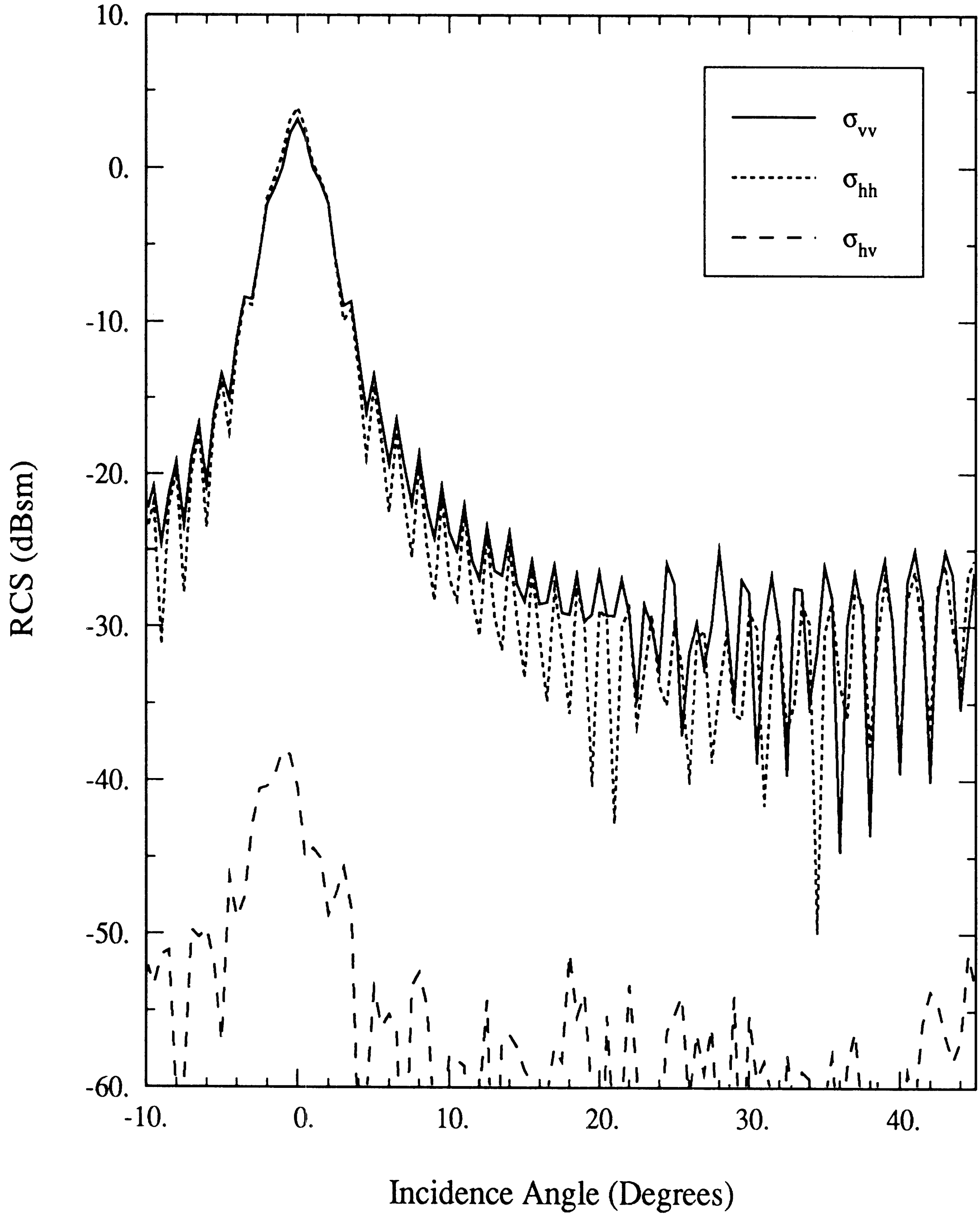


Figure 27: The near field radar backscatter cross section of 120-cm-long smooth cylinder of diameter 7.62 cm at 4.75 GHz versus incidence angle.

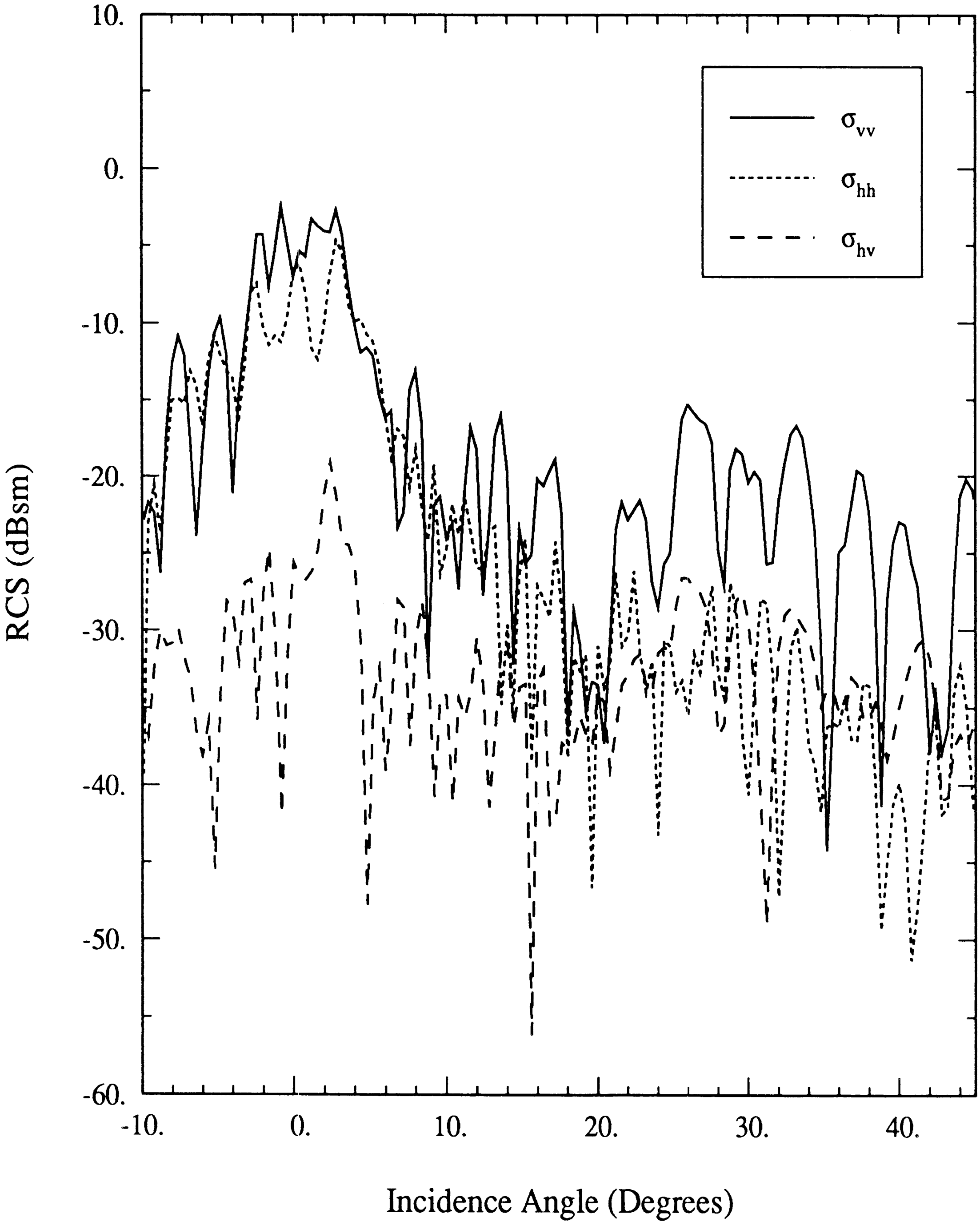


Figure 28: The near field radar backscatter cross section of 180-cm-long cable #4 at 4.75 GHz versus incidence angle.

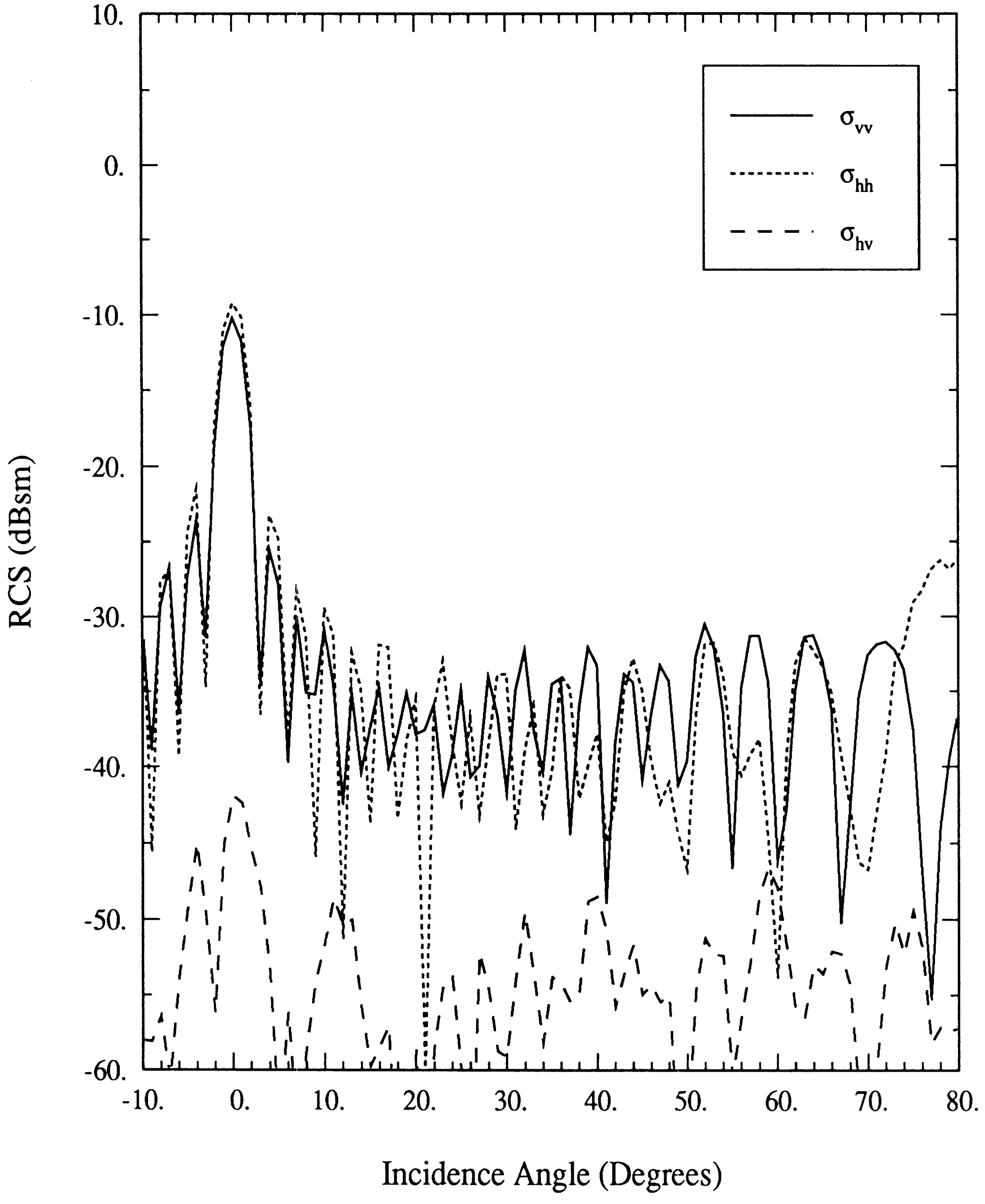


Figure 29: The radar backscatter cross section of 30-cm-long cable #1 at 9.5 GHz versus incidence angle.

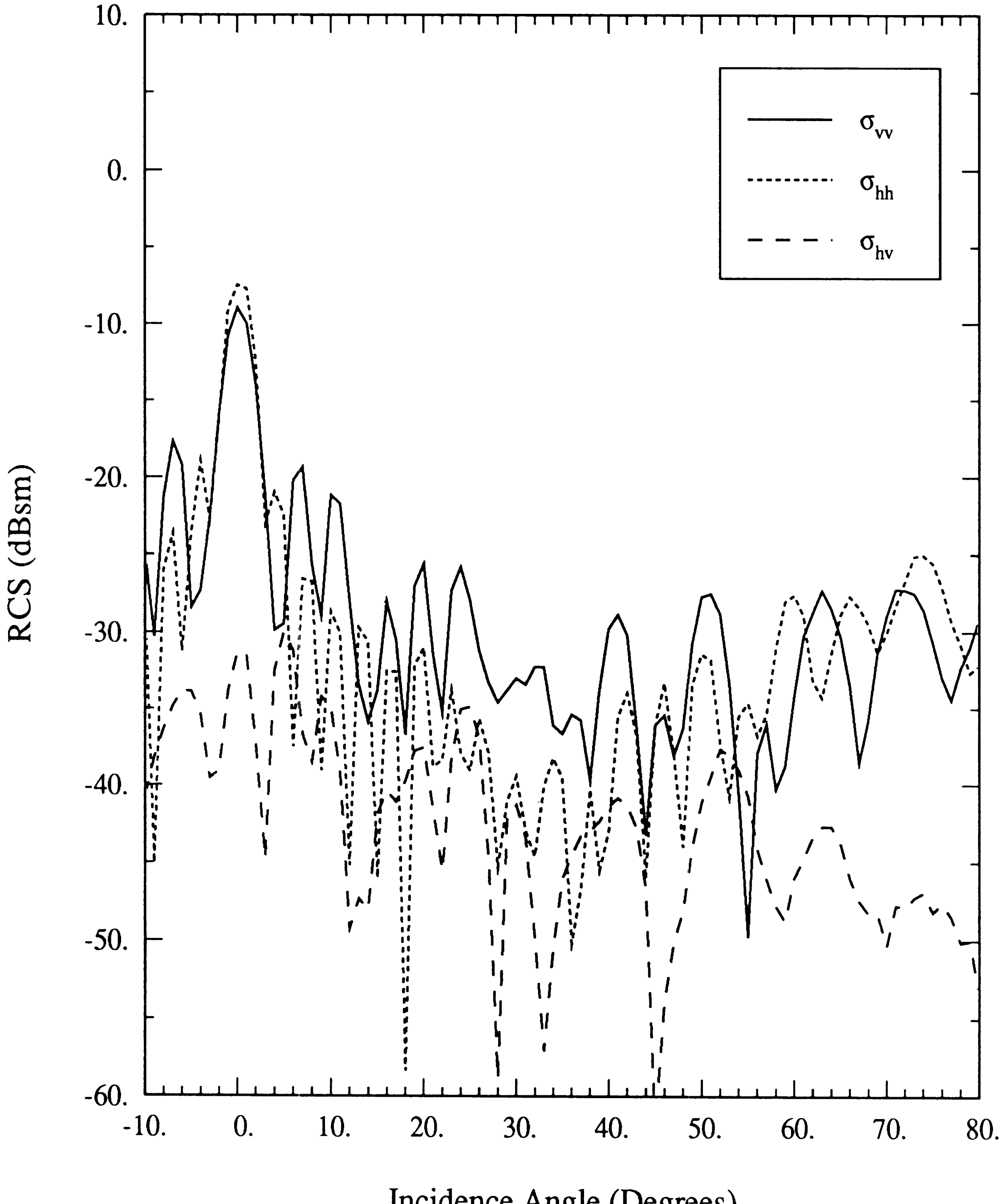


Figure 30: The radar backscatter cross section of 30-cm-long cable #2 at 9.5 GHz versus incidence angle.

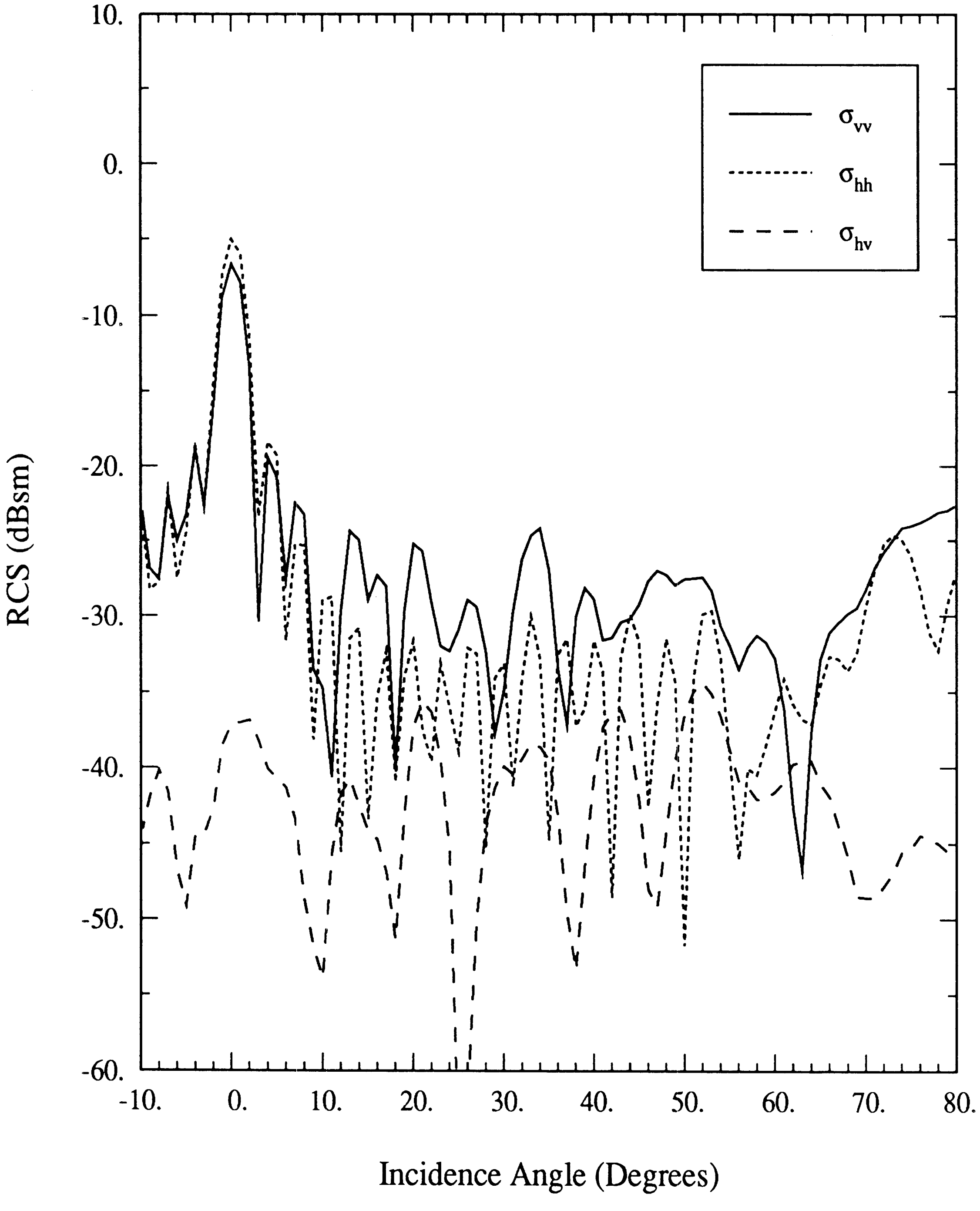


Figure 31: The radar backscatter cross section of 30-cm-long cable #3 at 9.5 GHz versus incidence angle.

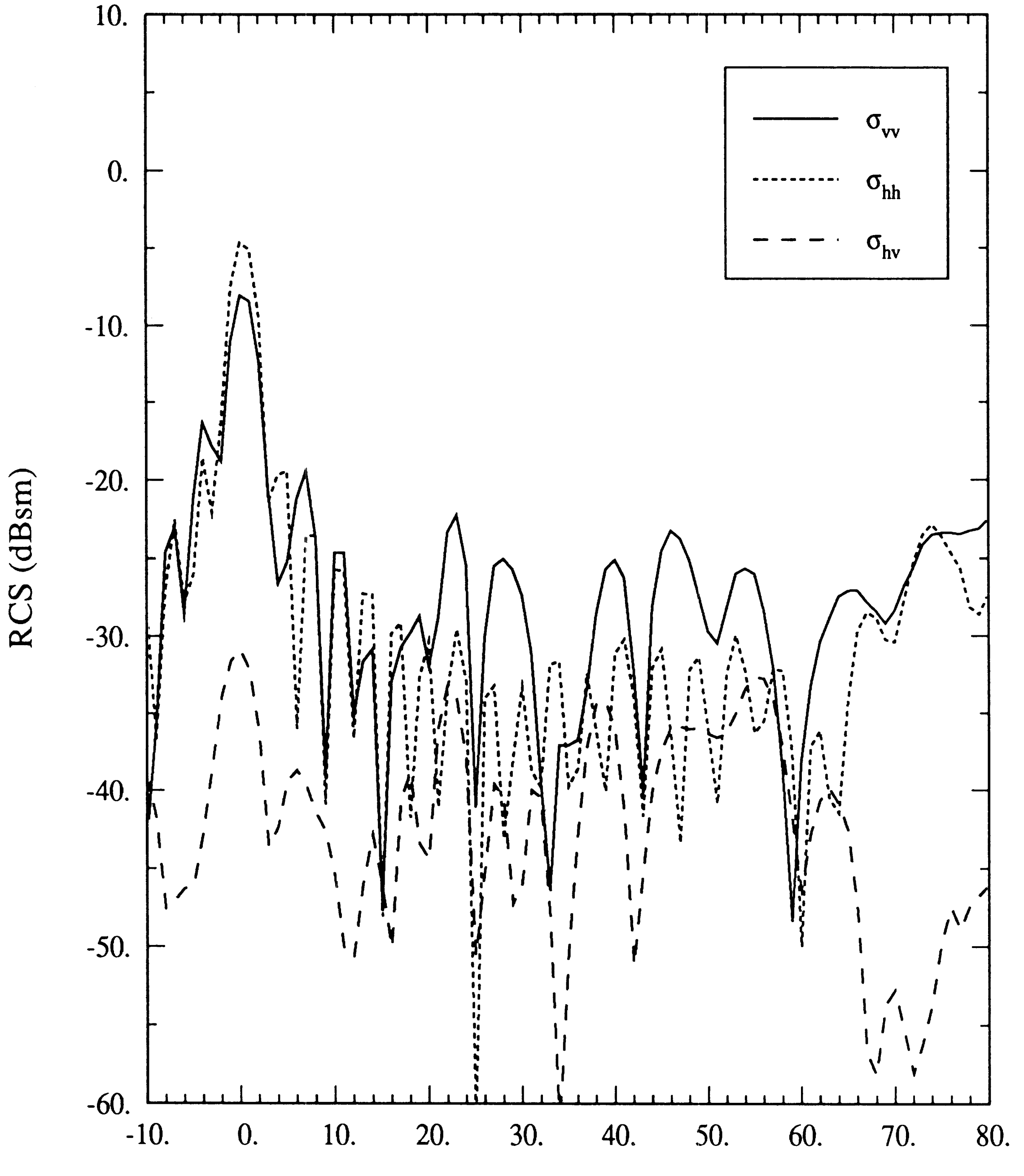


Figure 32: The radar backscatter cross section of 30-cm-long cable #4 at 9.5 GHz versus incidence angle.

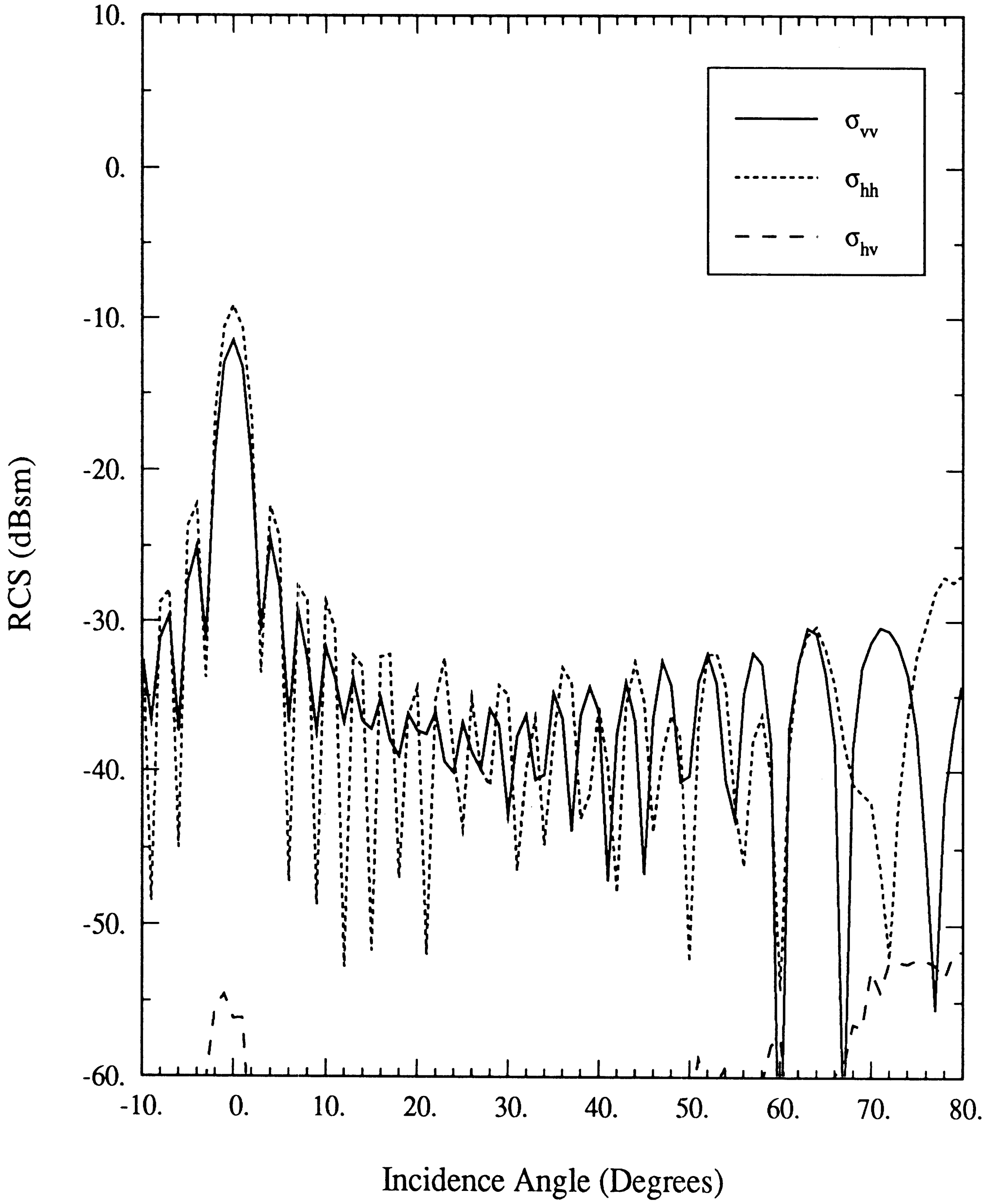


Figure 33: The radar backscatter cross section of 30-cm-long smooth cylinder of diameter 1.27 cm at 9.5 GHz versus incidence angle.

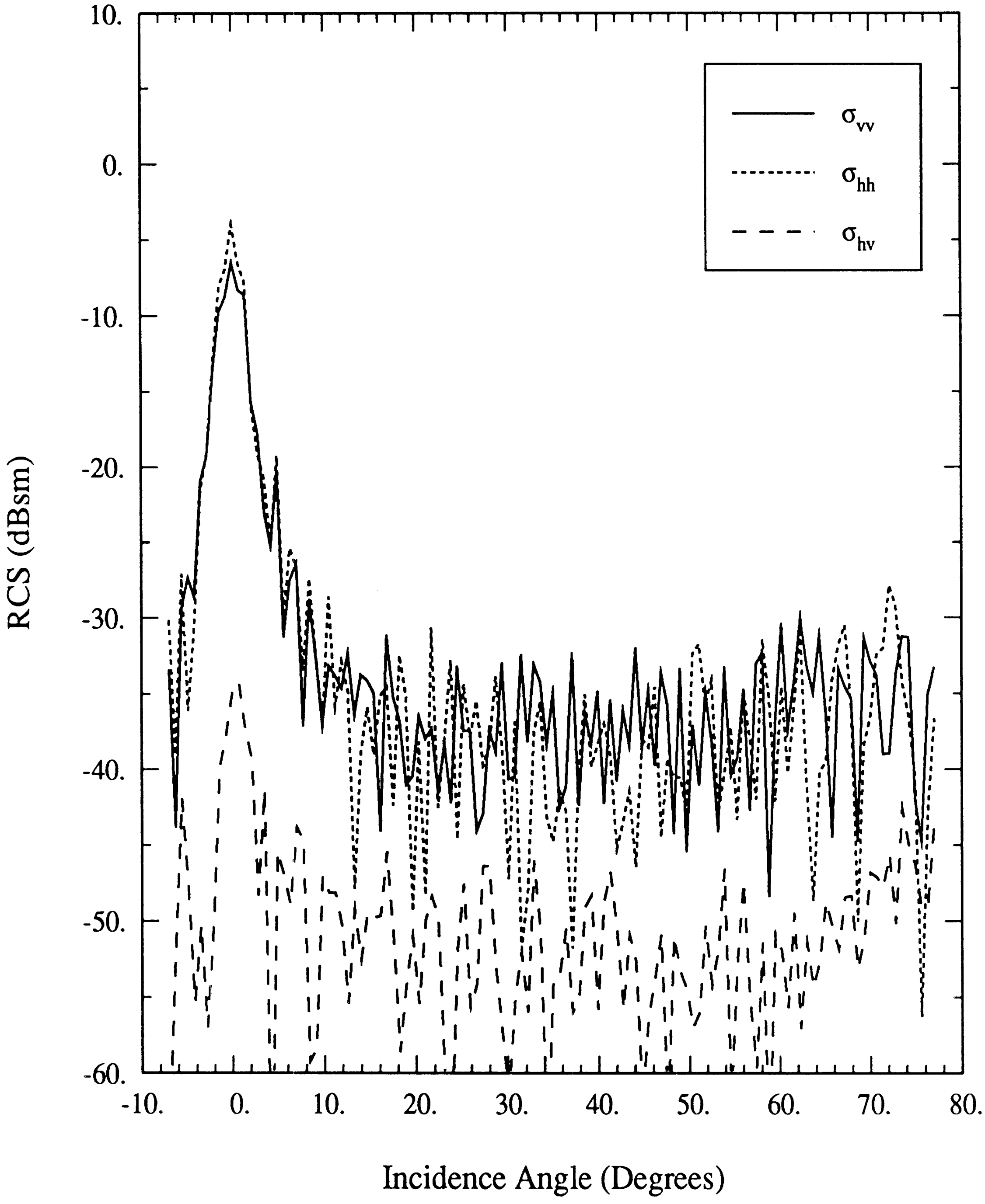


Figure 34: The near field radar backscatter cross section of 84-cm-long cable #1 at 9.5 GHz versus incidence angle.

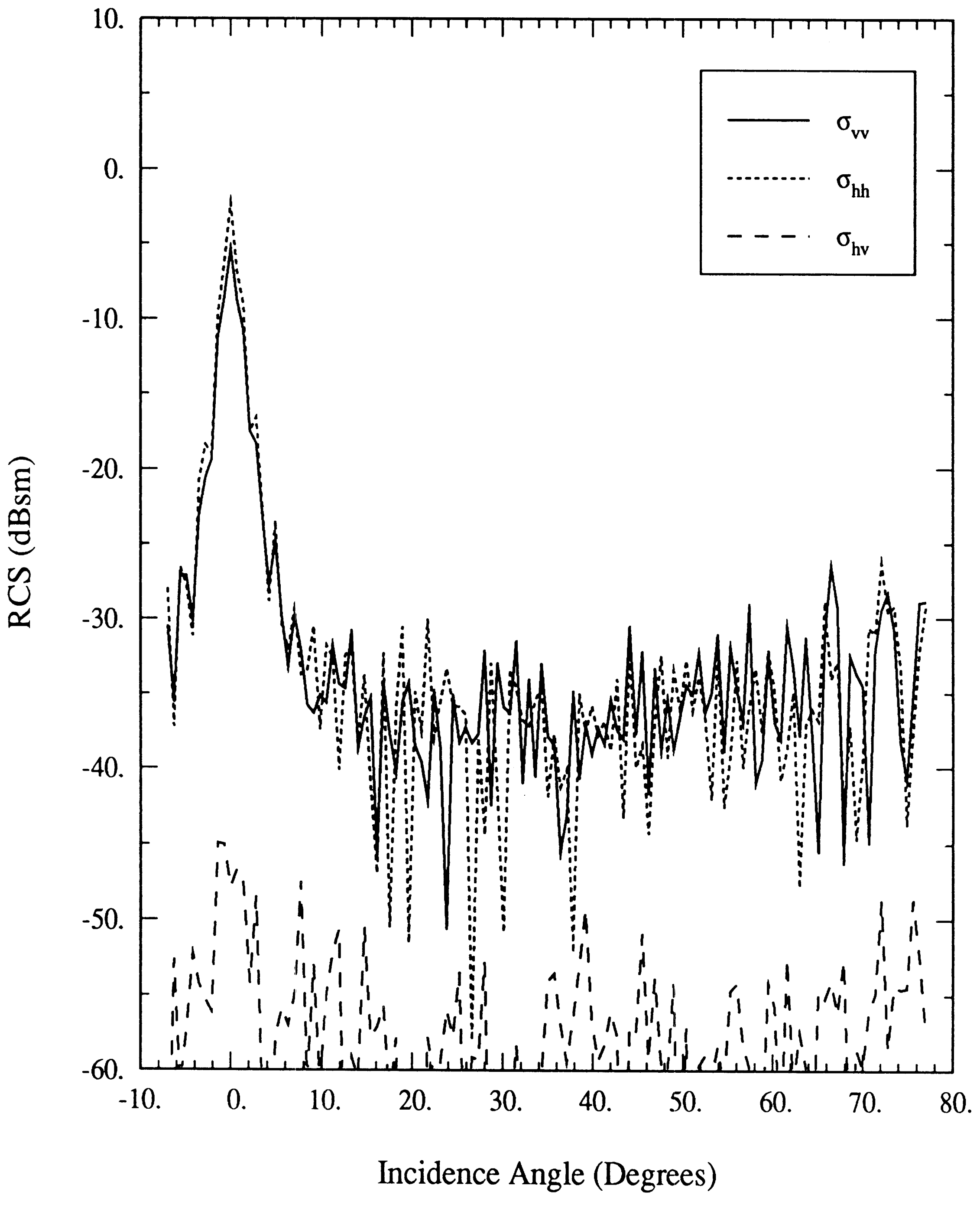


Figure 35: The near field radar backscatter cross section of 84-cm-long smooth cylinder of diameter 1.27 cm at 9.5 GHz versus incidence angle.

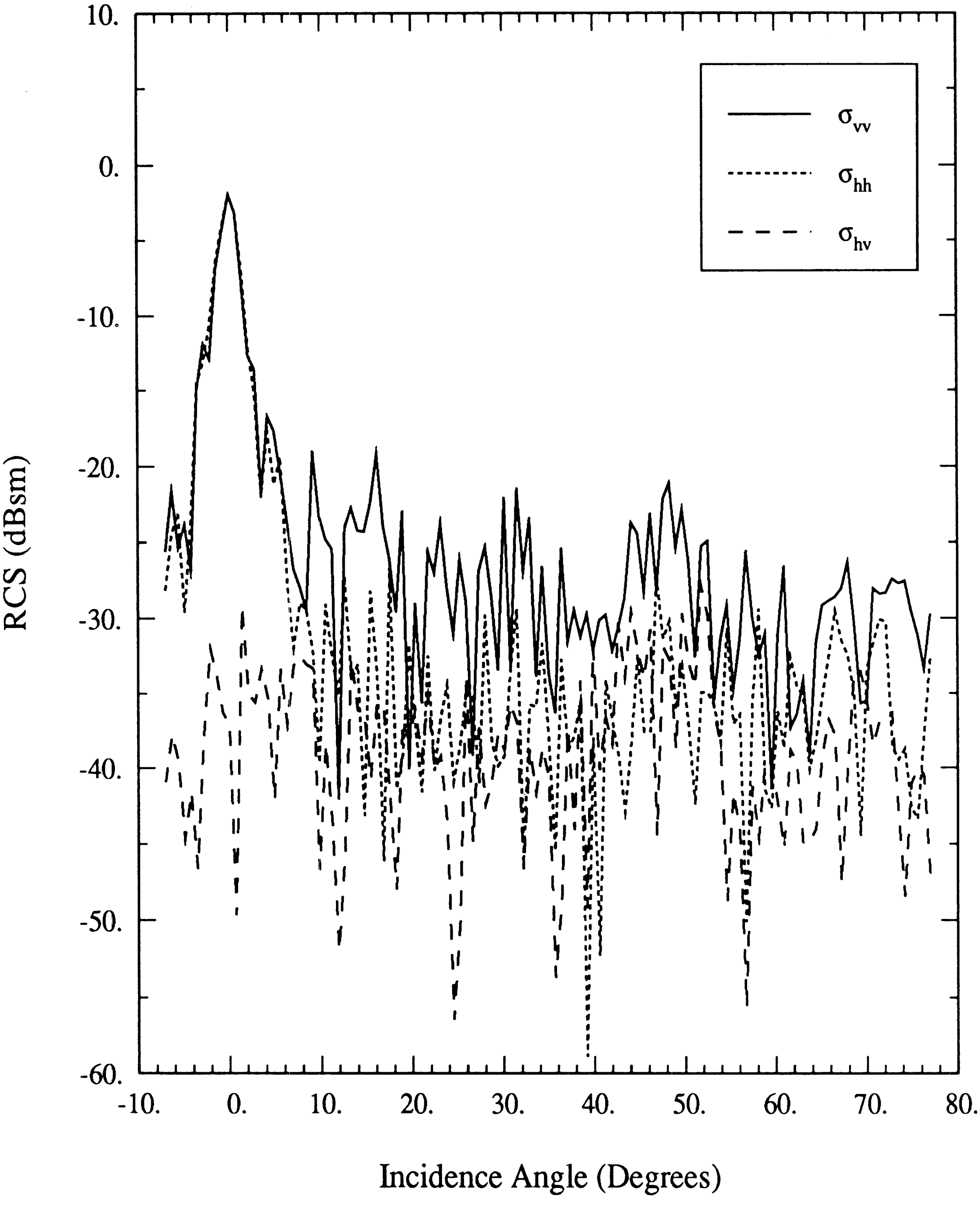


Figure 36: The near field radar backscatter cross section of 90-cm-long cable #2 at 9.5 GHz versus incidence angle.

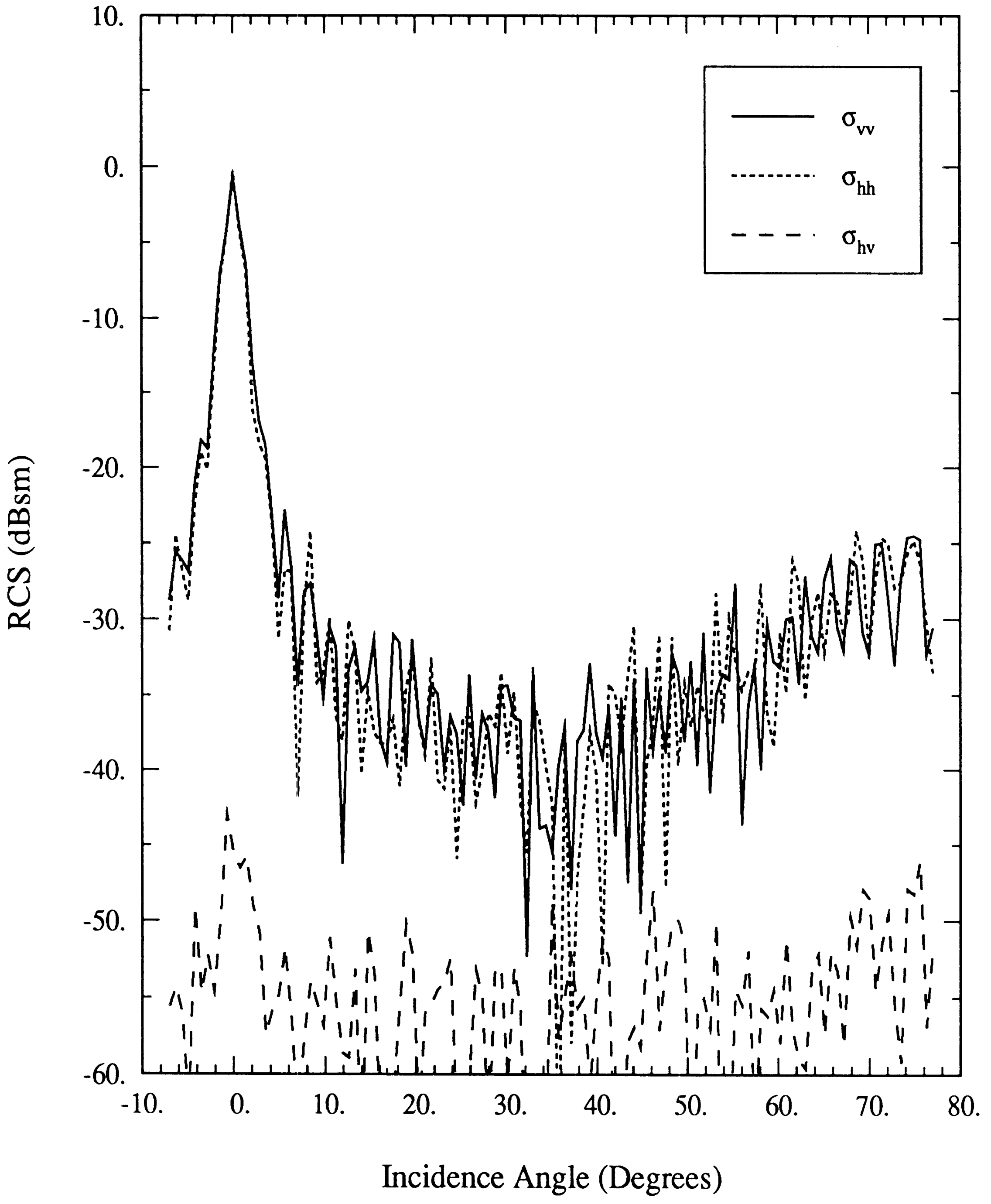


Figure 37: The near field radar backscatter cross section of 90-cm-long smooth cylinder of diameter 2.22 cm at 9.5 GHz versus incidence angle.

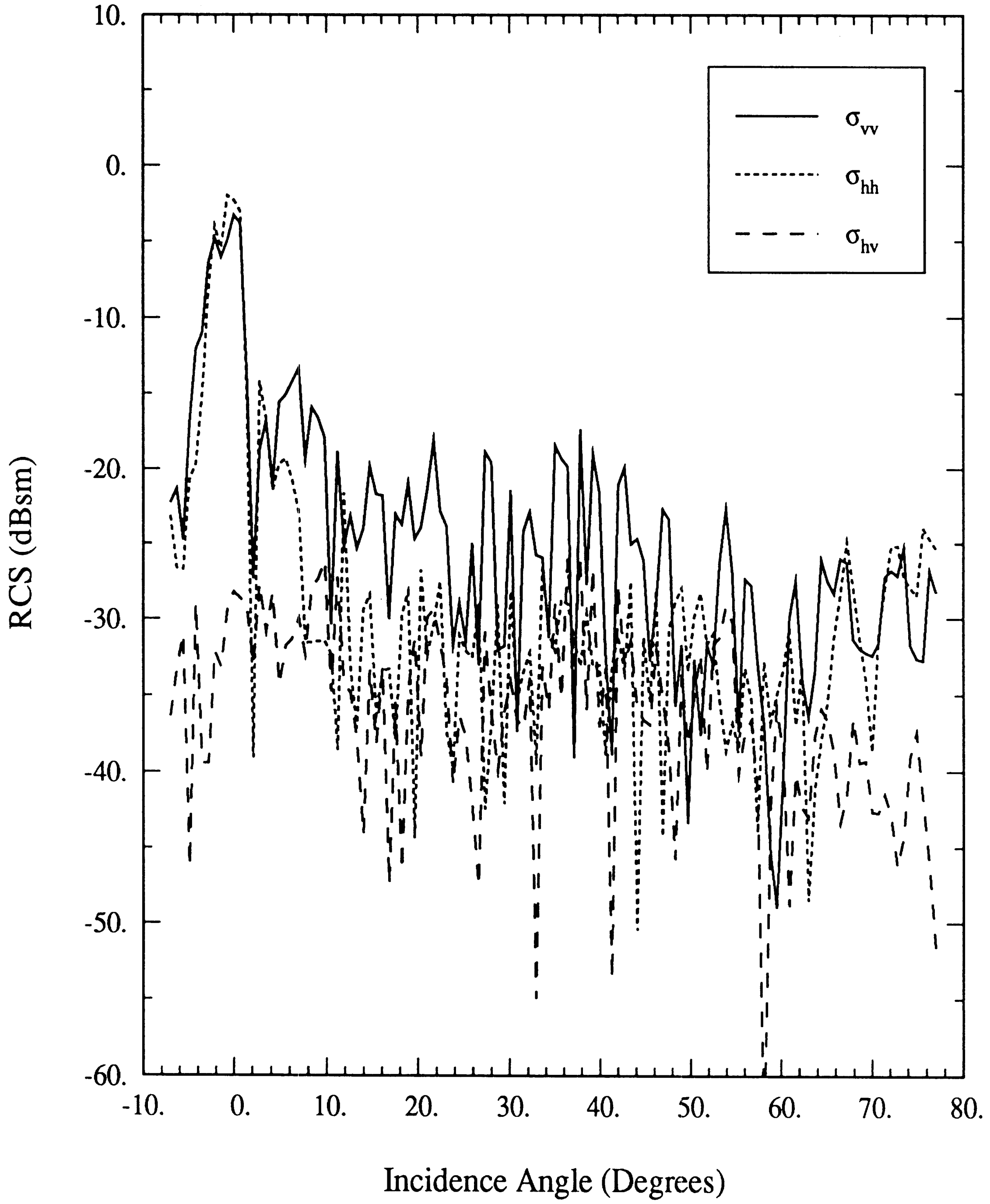


Figure 38: The near field radar backscatter cross section of 90-cm-long cable #3 at 9.5 GHz versus incidence angle.

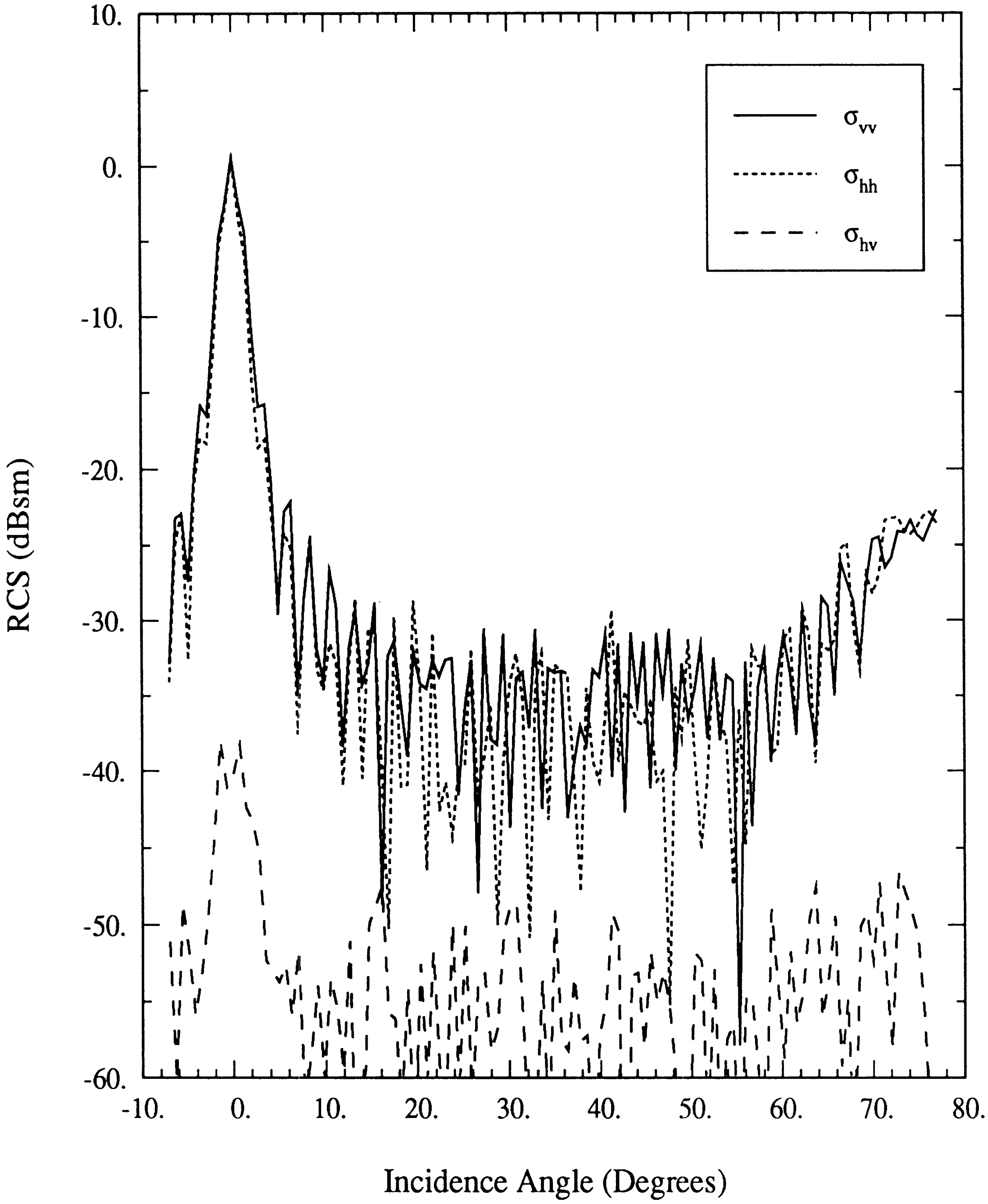


Figure 39: The near field radar backscatter cross section of 90-cm-long smooth cylinder of diameter 3.15 cm at 9.5 GHz versus incidence angle.

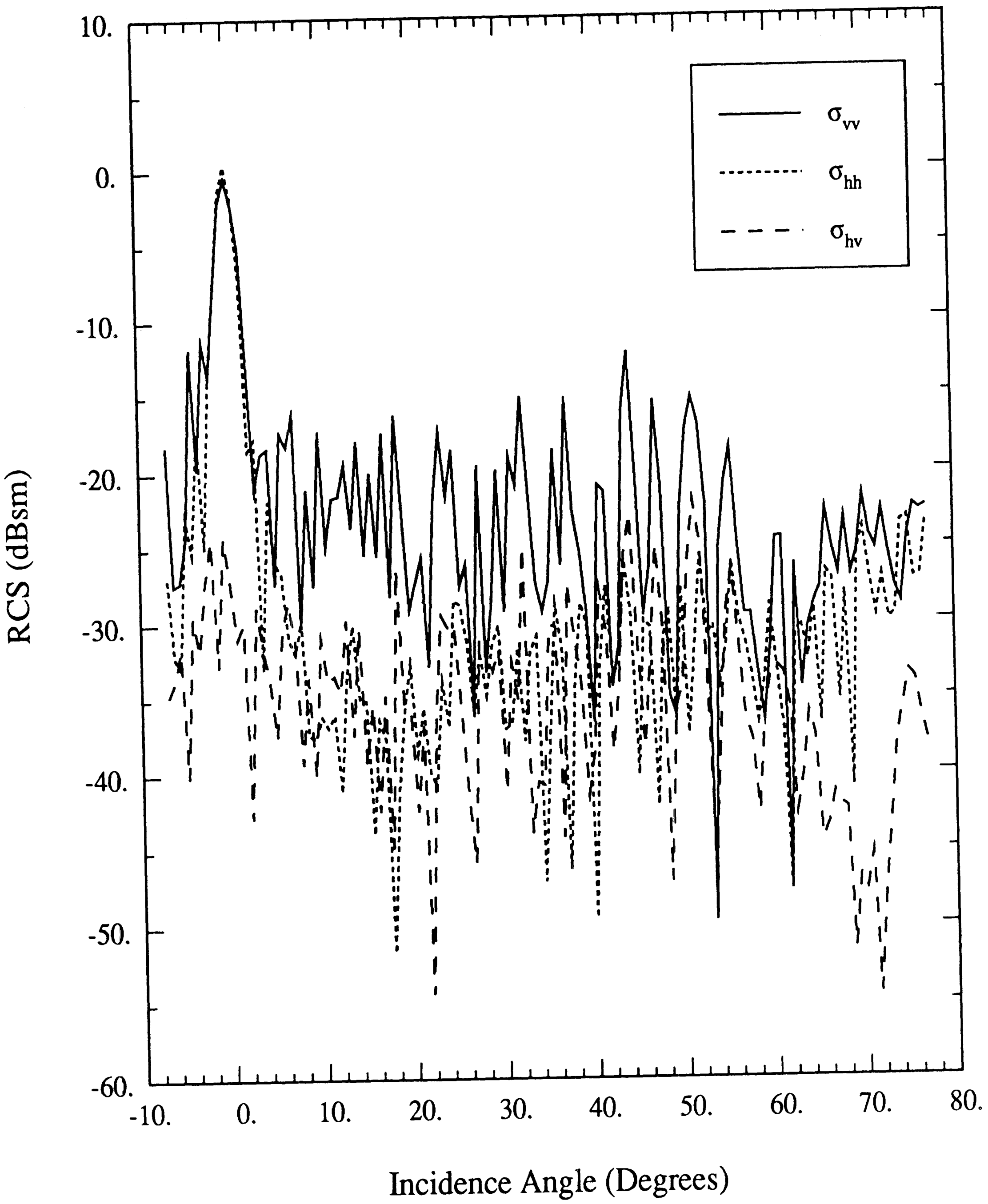


Figure 40: The near field radar backscatter cross section of 120-cm-long cable #4 at 9.5 GHz versus incidence angle.

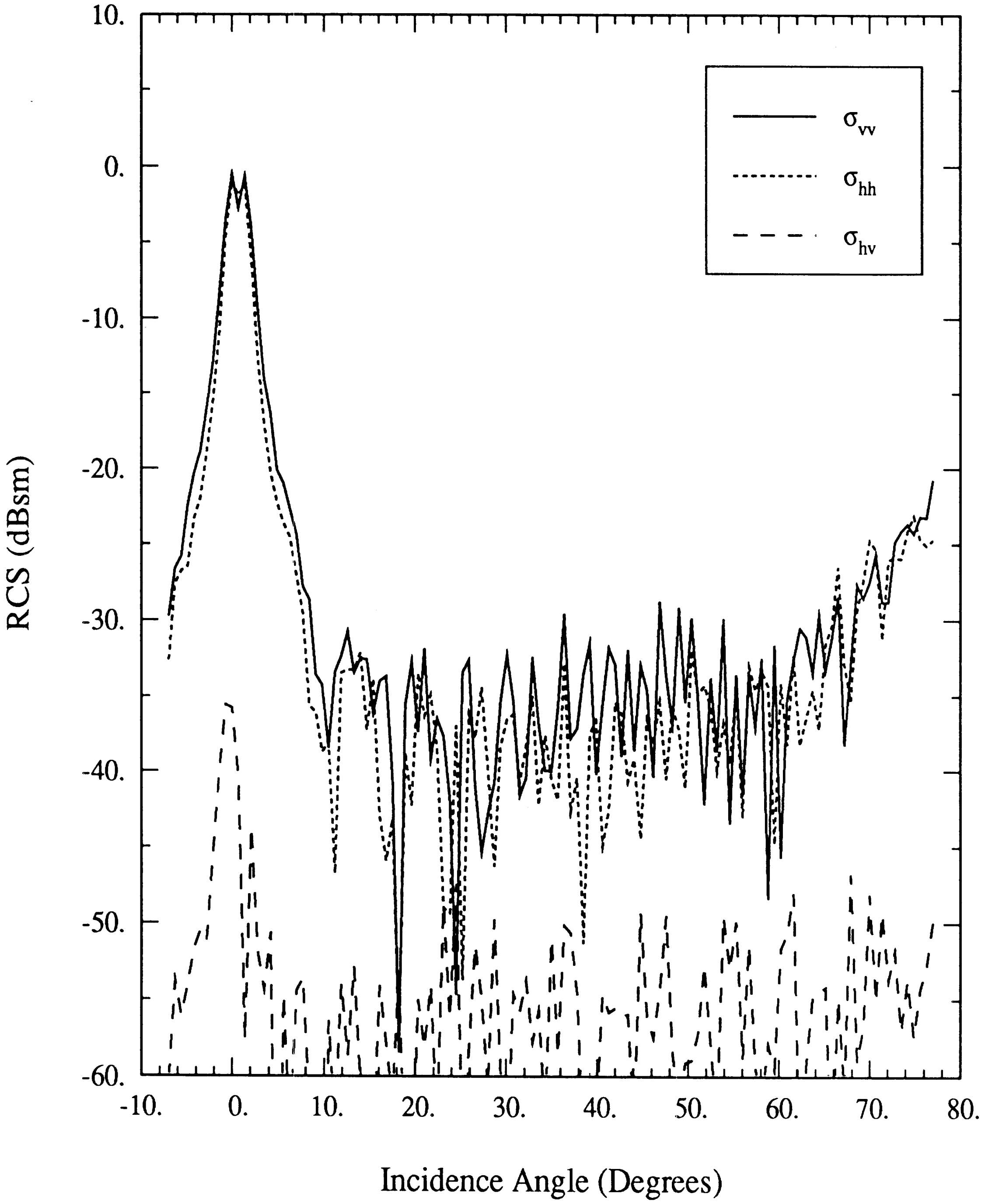


Figure 41: The near field radar backscatter cross section of 120-cm-long smooth cylinder of diameter 3.49 cm at 9.5 GHz versus incidence angle.

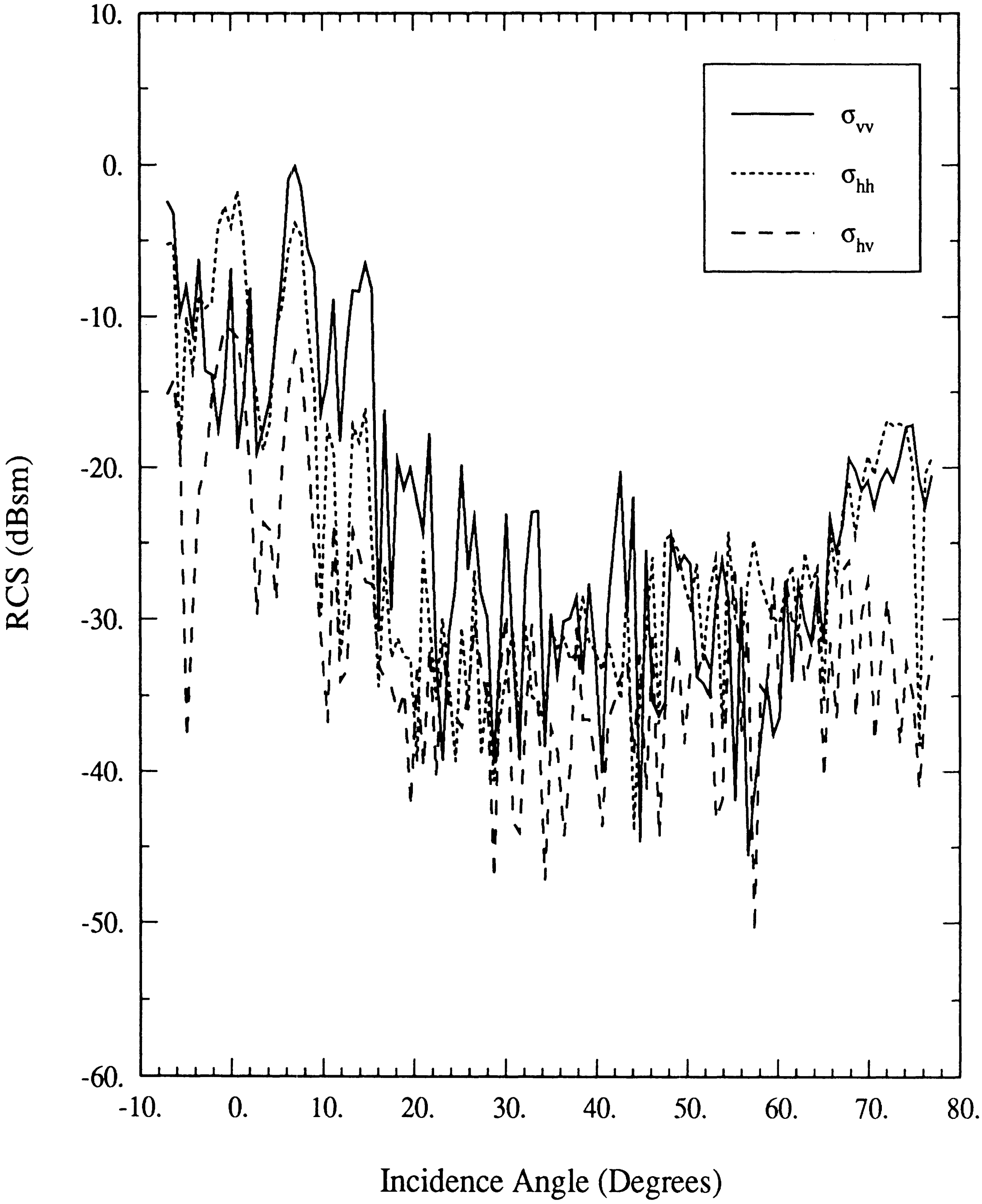


Figure 42: The near field radar backscatter cross section of 120-cm-long prototype cable at 9.5 GHz versus incidence angle.

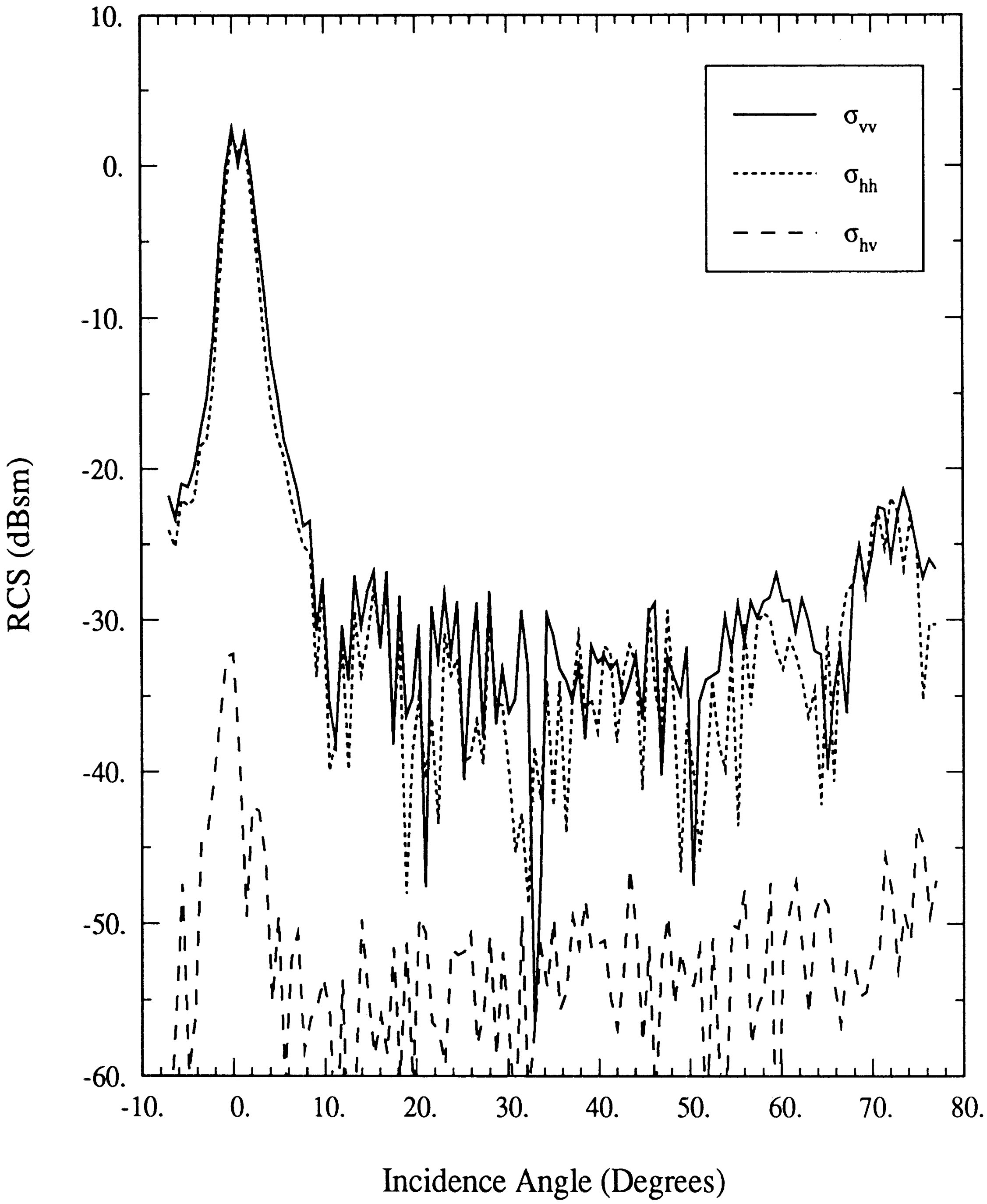


Figure 43: The near field radar backscatter cross section of 120-cm-long smooth cylinder of diameter 7.62 cm at 9.5 GHz versus incidence angle.

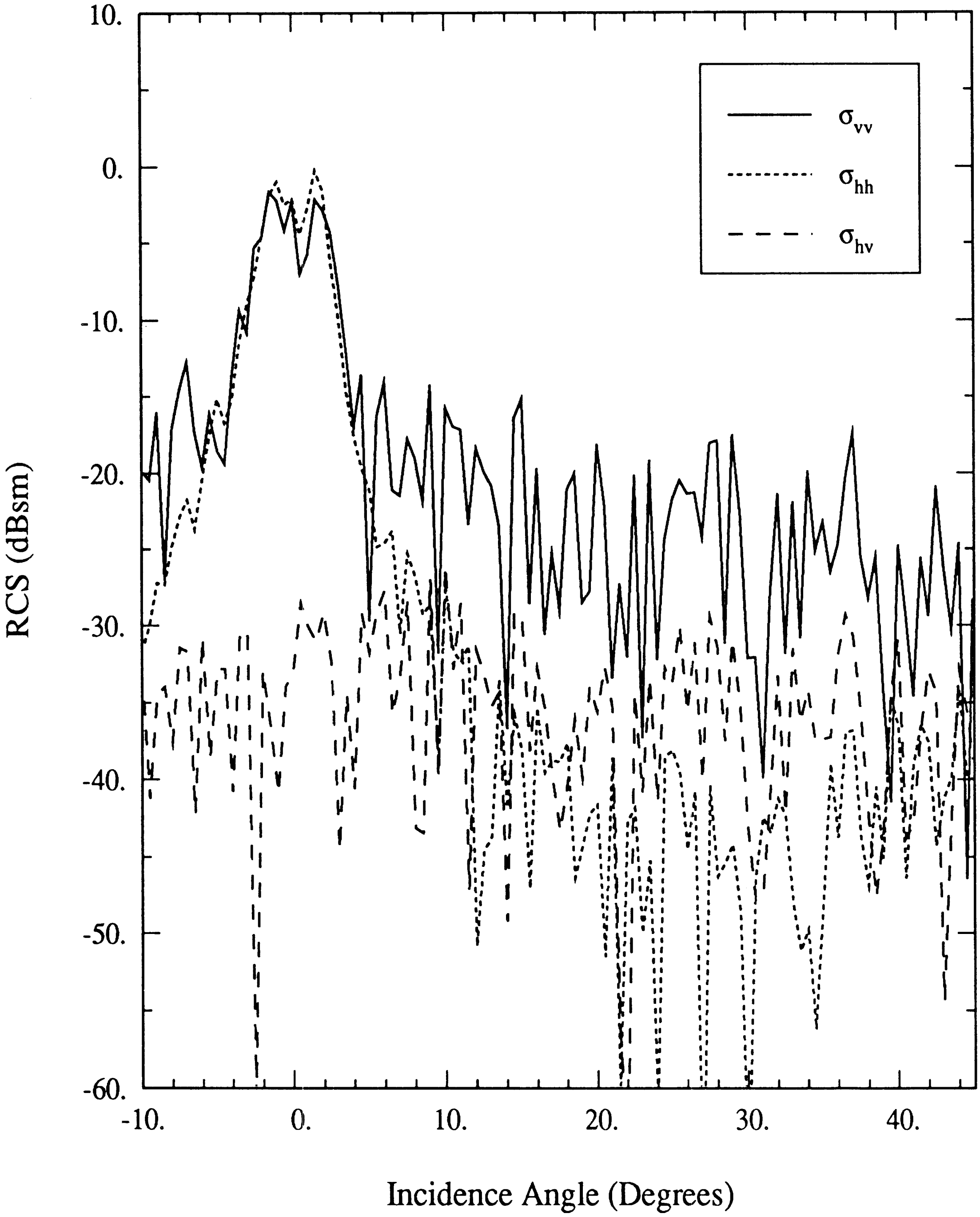


Figure 44: The near field radar backscatter cross section of 180-cm-long cable #4 at 9.5 GHz versus incidence angle.

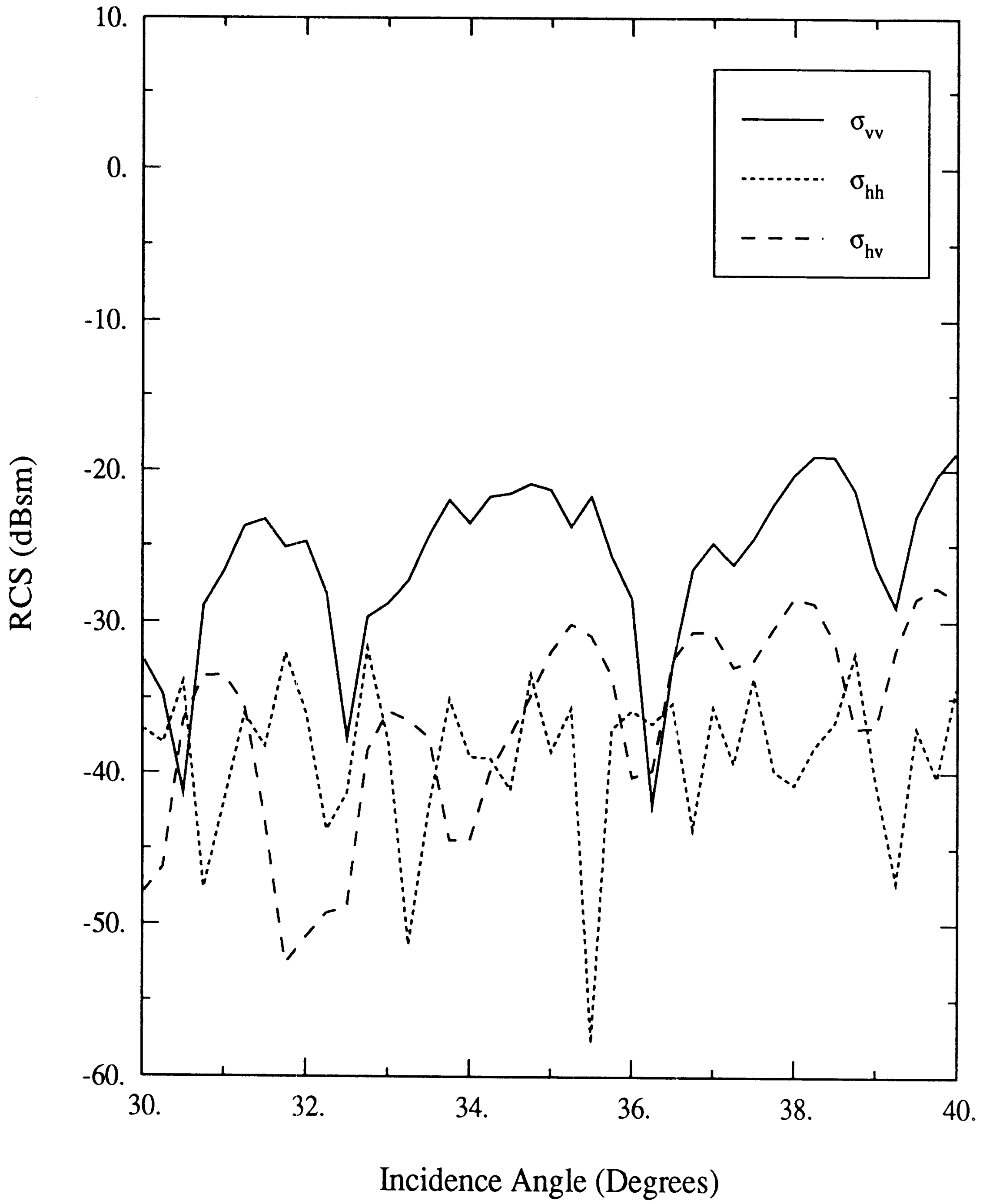


Figure 45: The near field radar backscatter cross section of 90-cm-long cable #3 at 9.5 GHz versus incidence angle.

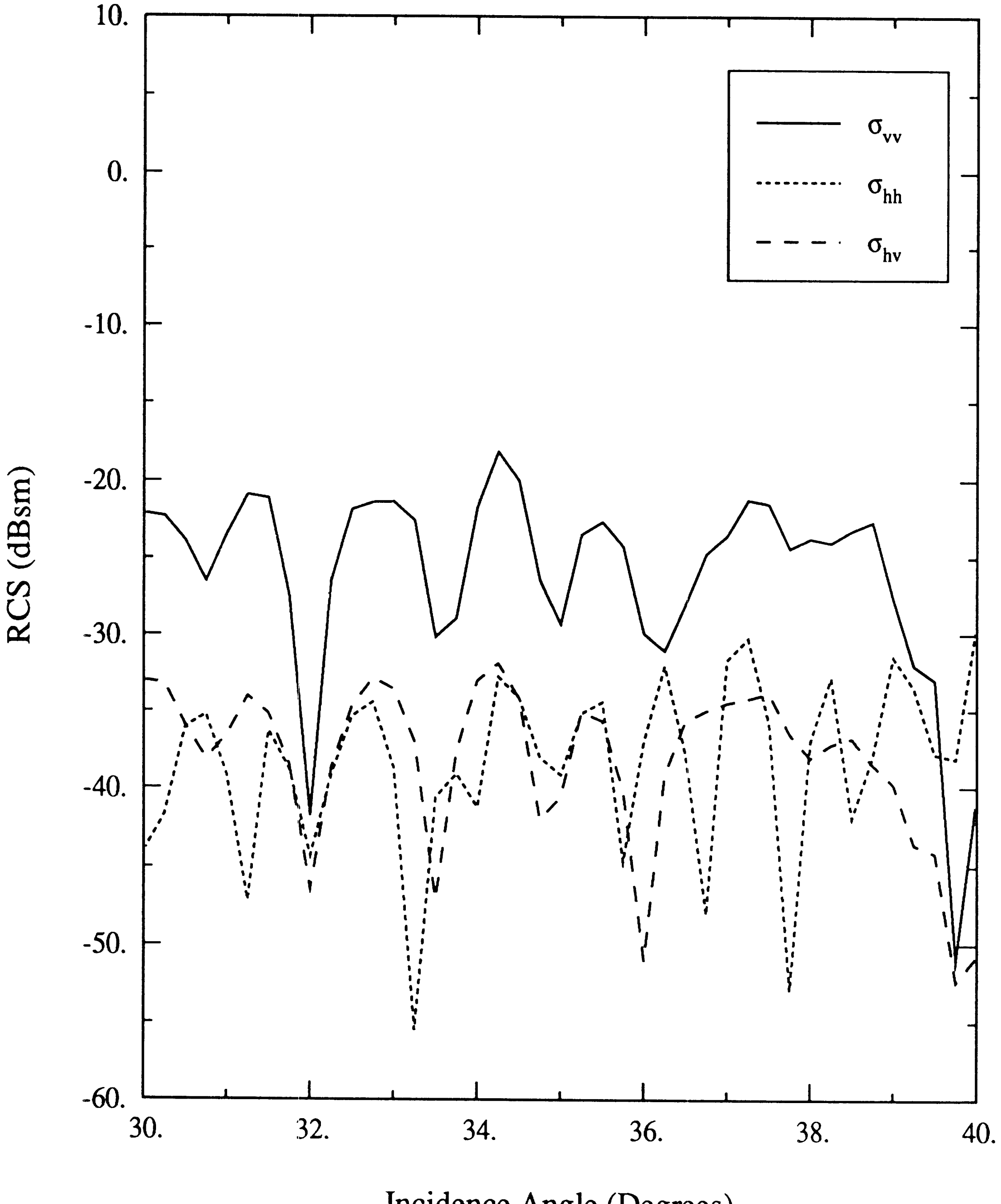


Figure 46: The near field radar backscatter cross section of 120-cm-long cable #4 at 9.5 GHz versus incidence angle.

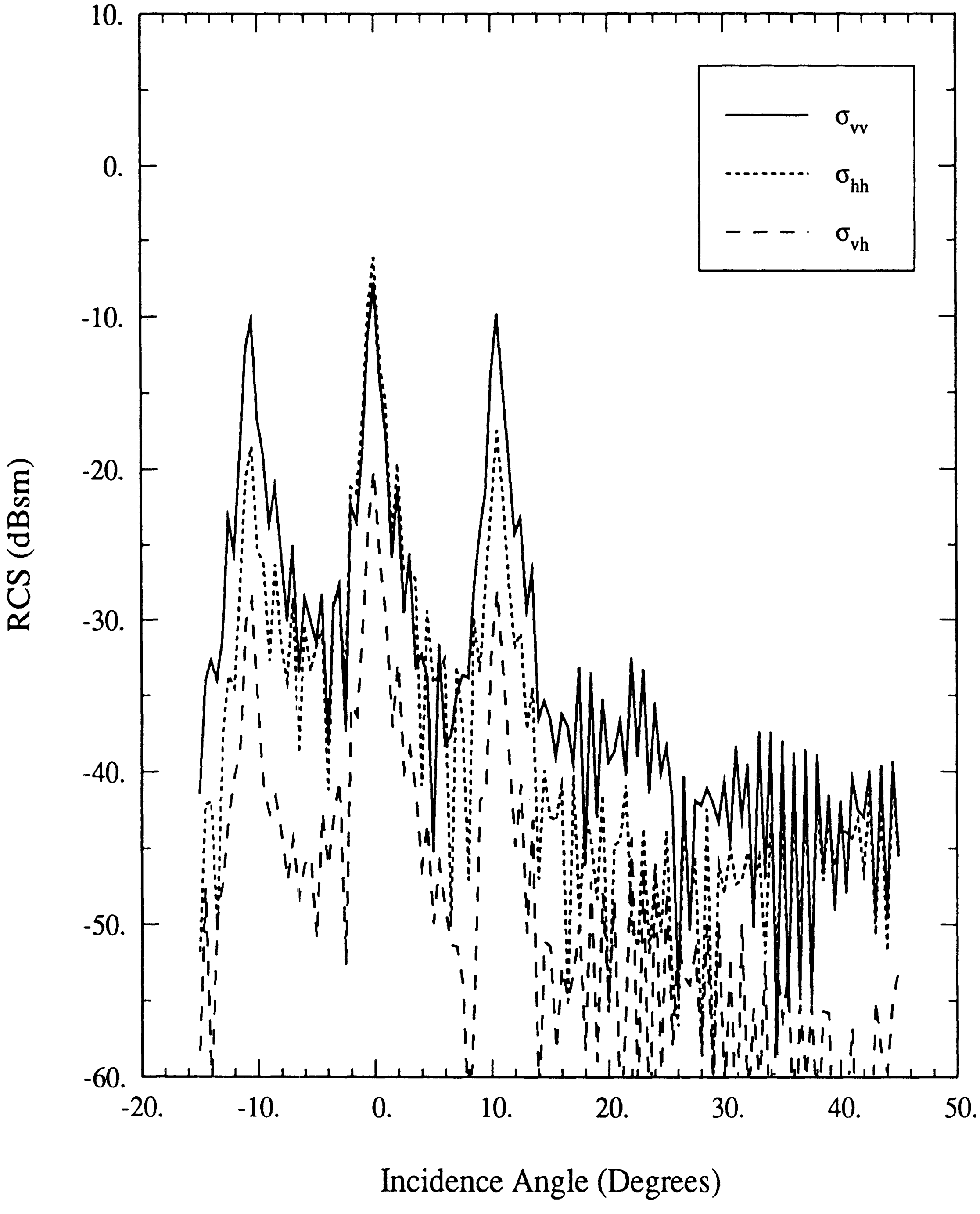


Figure 47: The radar backscatter cross section of 30-cm-long cable #1 at 34.5 GHz versus incidence angle.

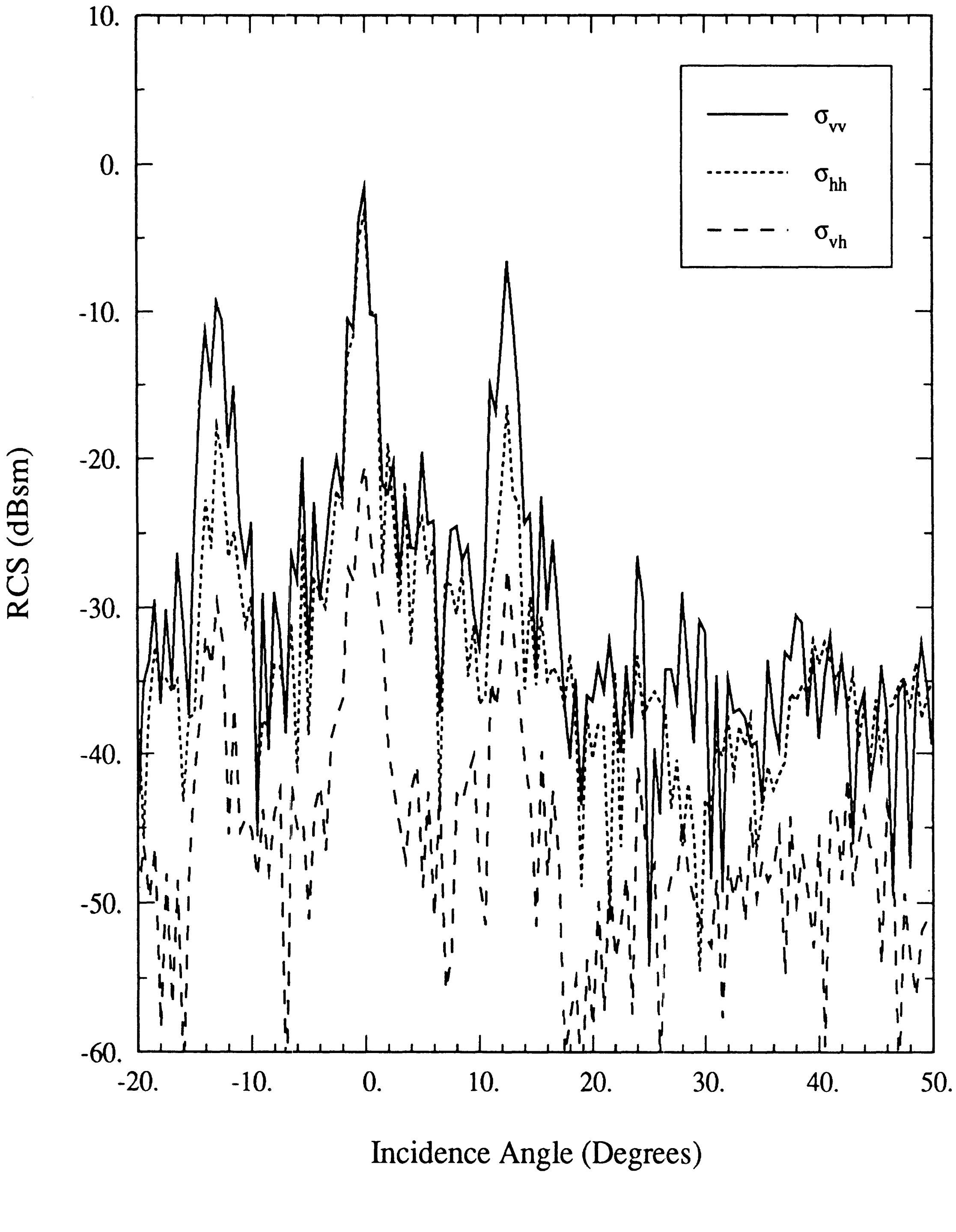


Figure 48: The radar backscatter cross section of 30-cm-long cable #2 at 34.5 GHz versus incidence angle.

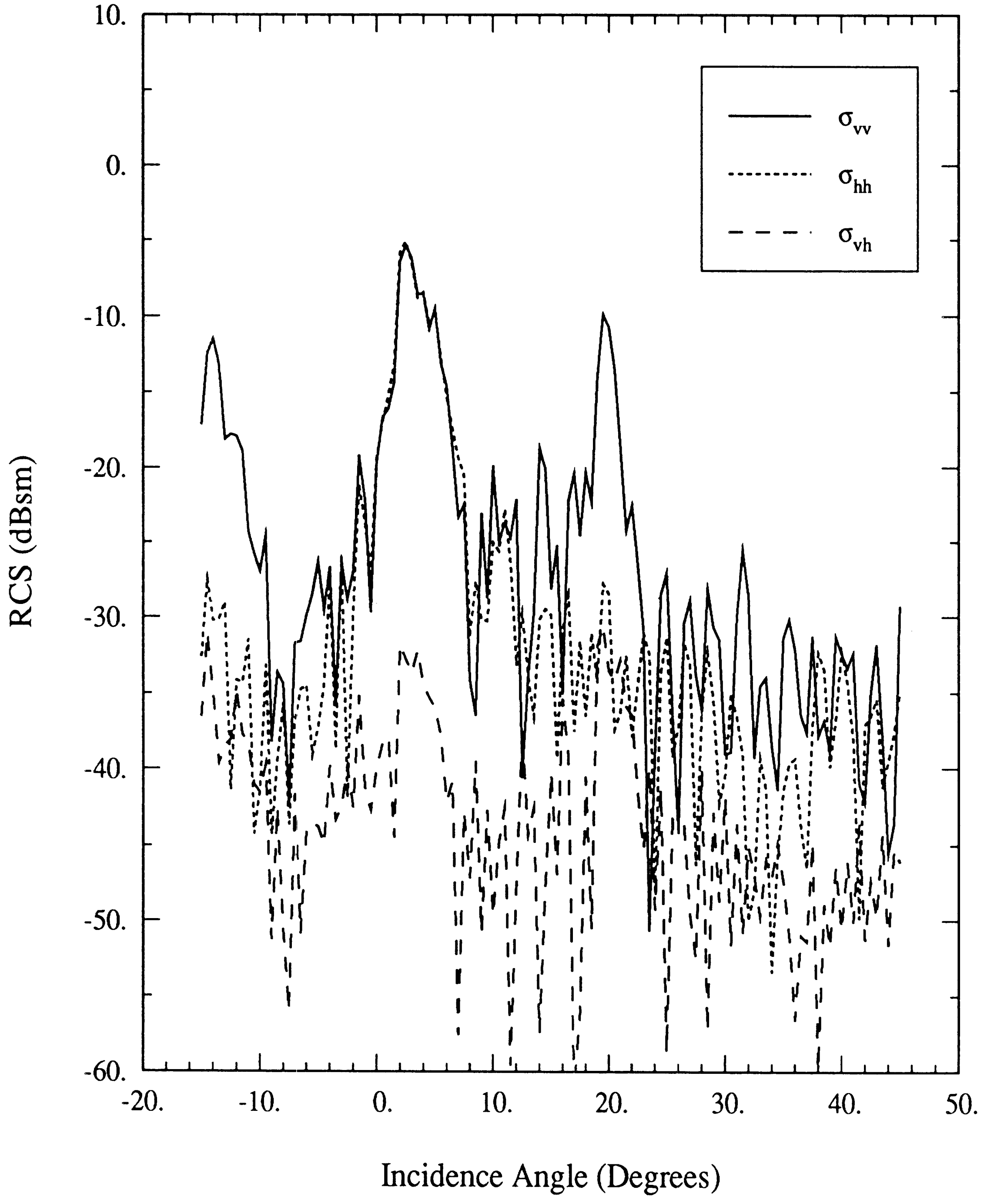


Figure 49: The radar backscatter cross section of 30-cm-long cable #3 at 34.5 GHz versus incidence angle.

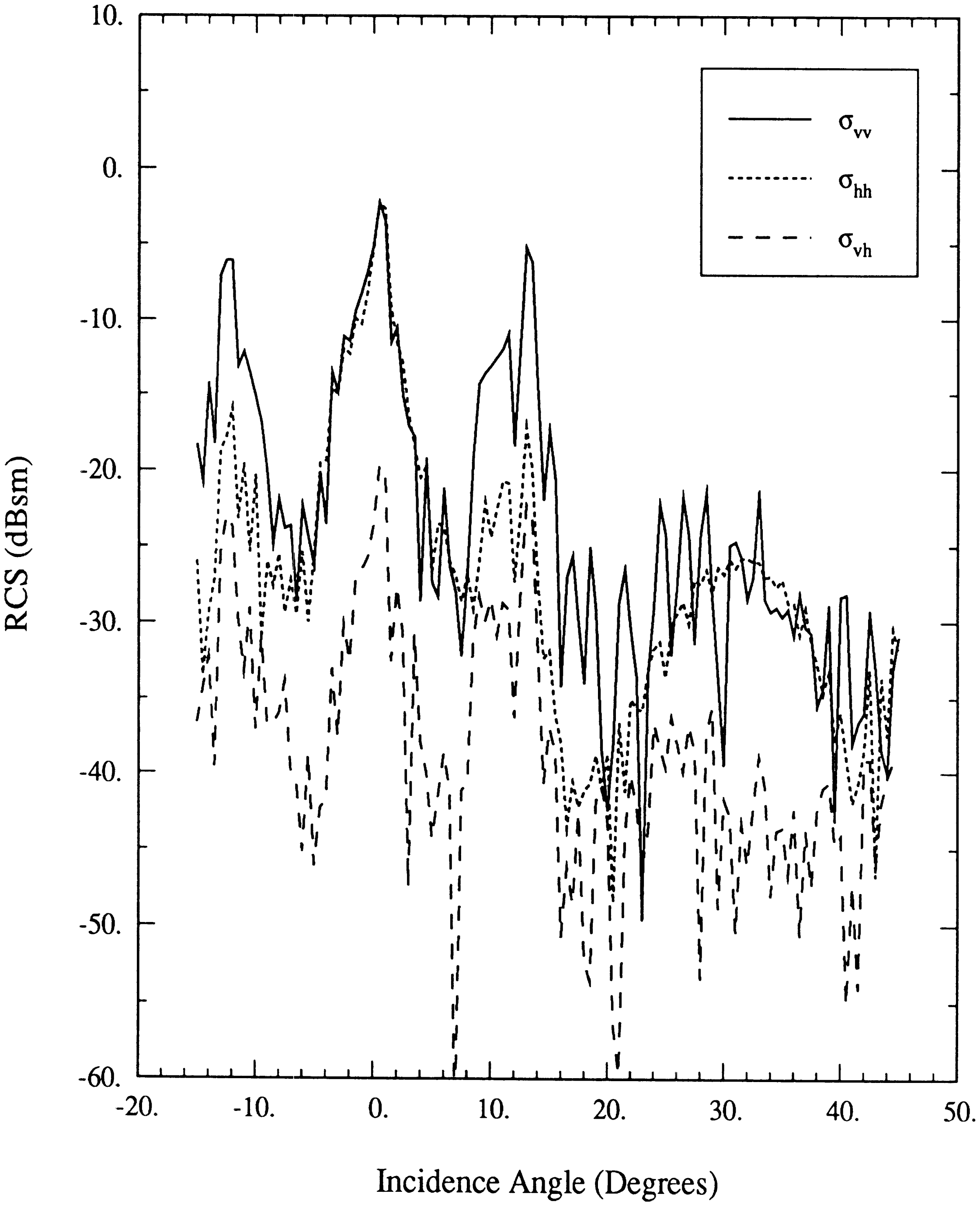


Figure 50: The radar backscatter cross section of 30-cm-long cable #4 at 34.5 GHz versus incidence angle.

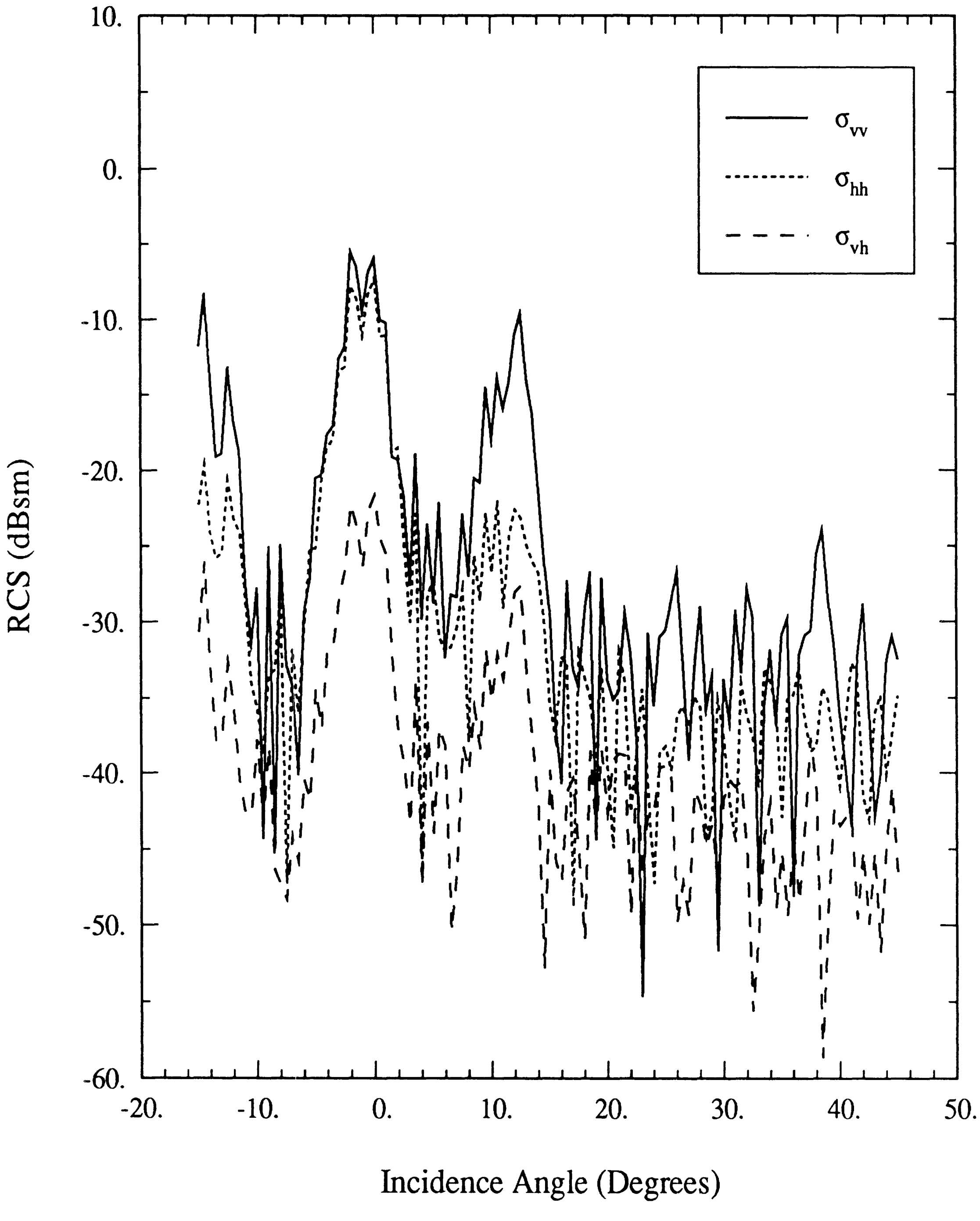


Figure 51: The radar backscatter cross section of curved 30-cm-long cable #2 at 34.5 GHz versus incidence angle (radius of curvature ≈10 m).

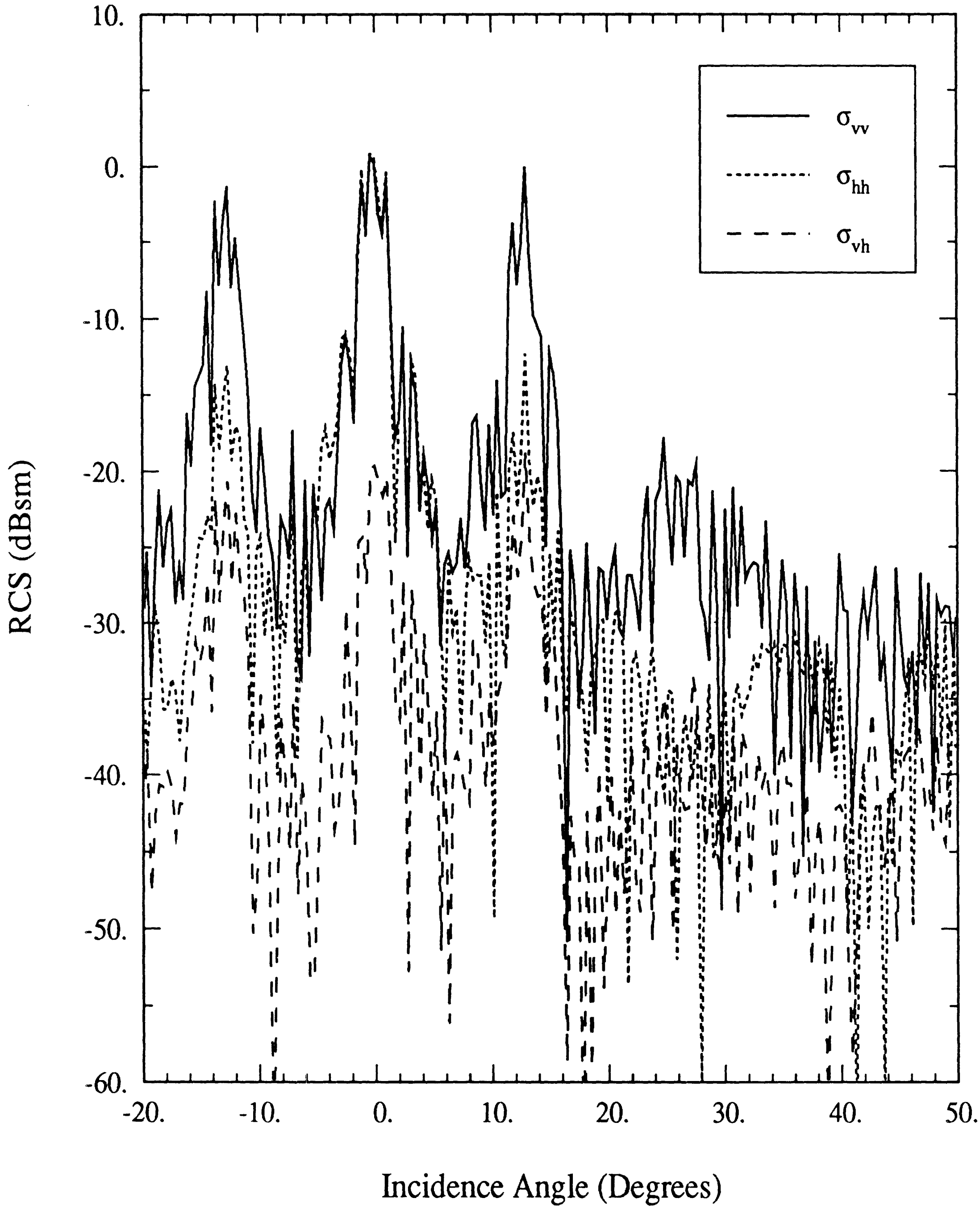


Figure 52: The near field radar backscatter cross section of 120-cm-long cable #4 at 34.5 GHz versus incidence angle.

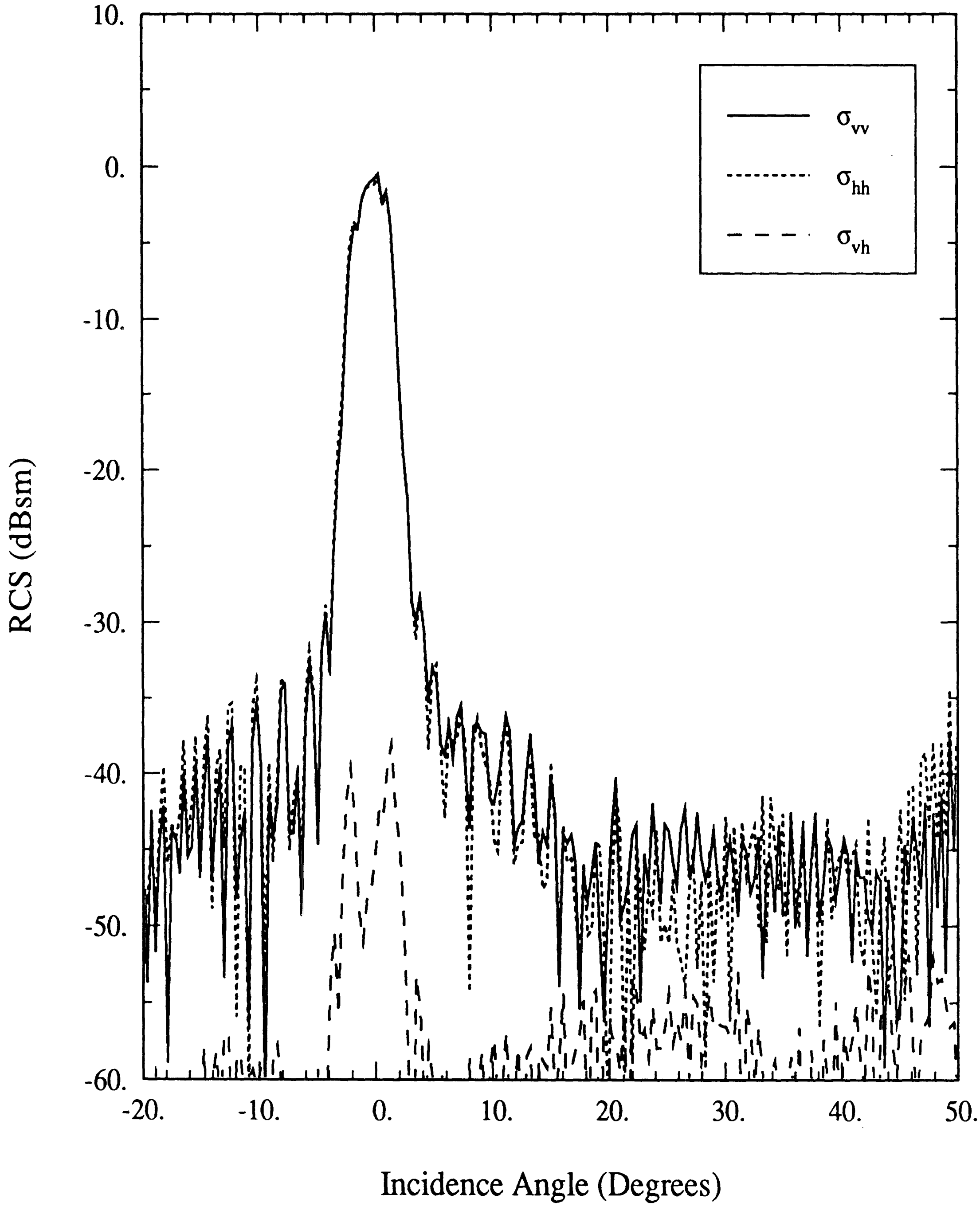


Figure 53: The near field radar backscatter cross section of 120-cm-long smooth cylinder of diameter 3.49 cm at 34.5 GHz versus incidence angle.

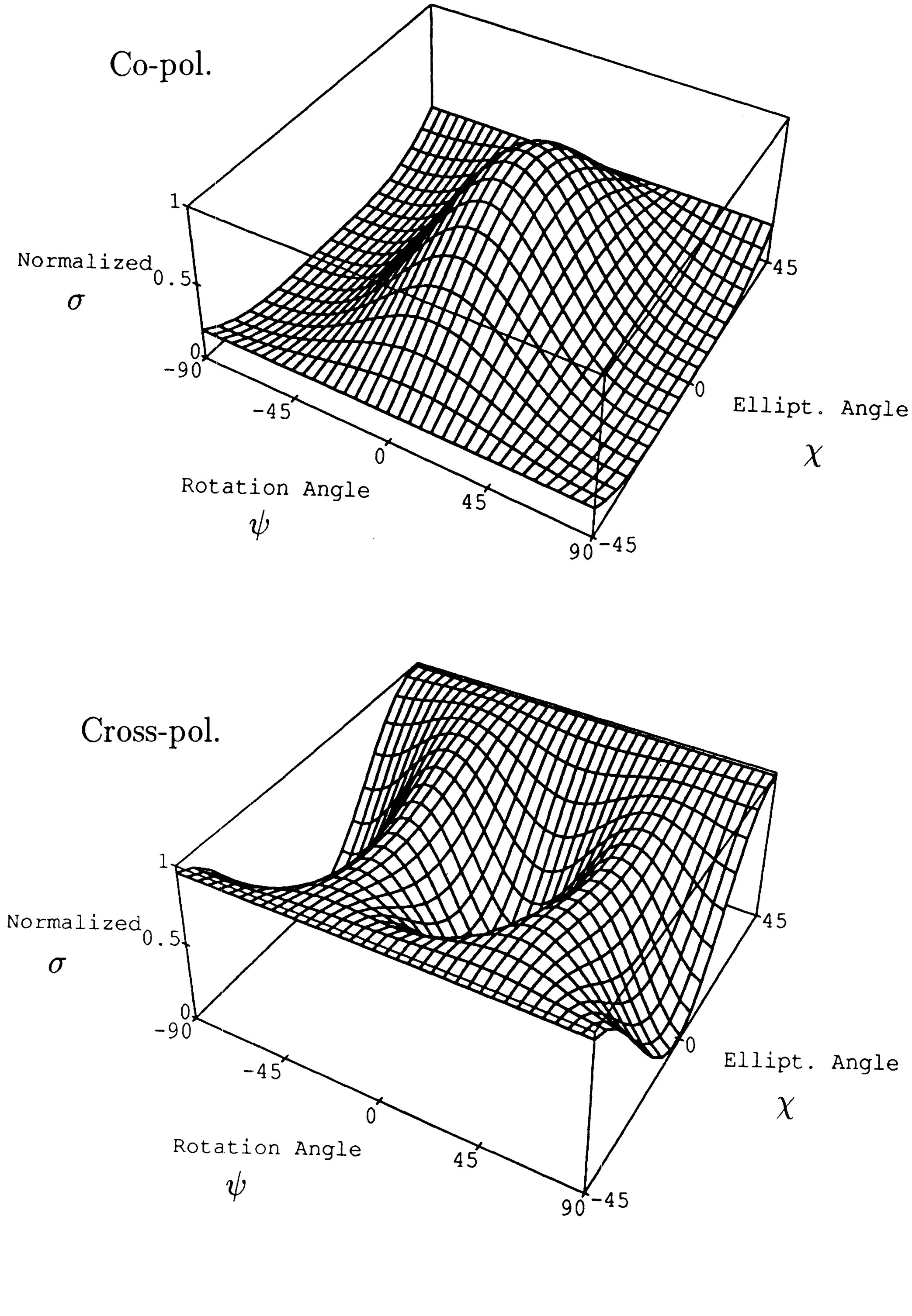


Figure 54: The co- and cross-polarized signatures of cable #4 (120 cm long) at 4.75 GHz and 19° incidence angle.

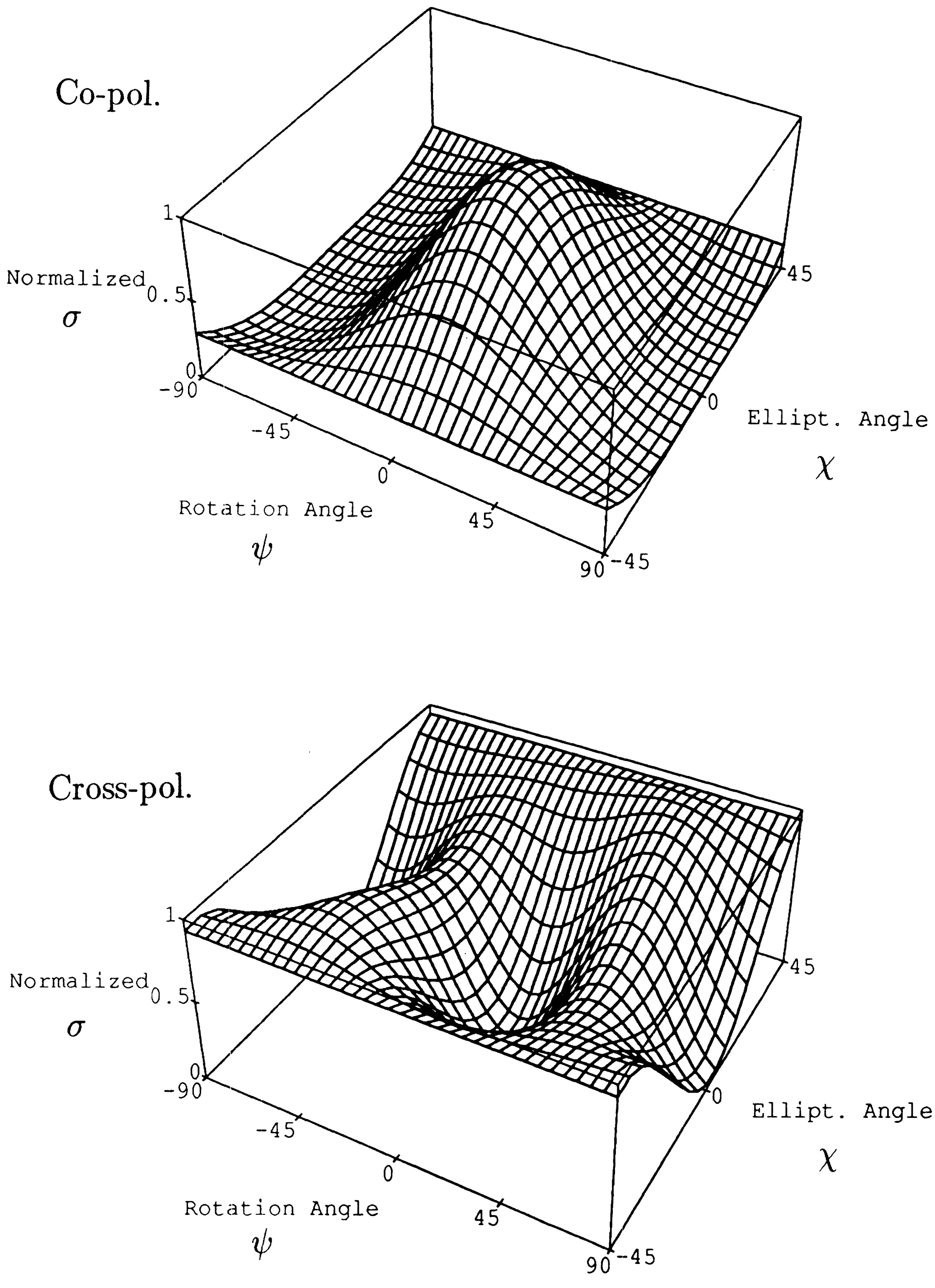
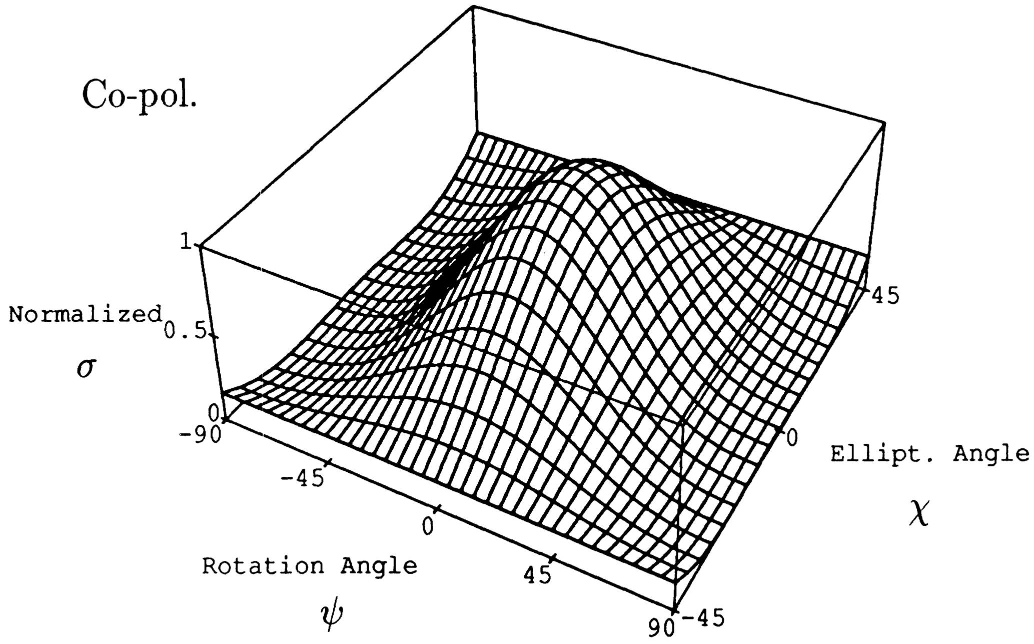


Figure 55: The co- and cross-polarized signatures of cable #4 (120 cm long) at 9.5 GHz and 43.4° incidence angle.



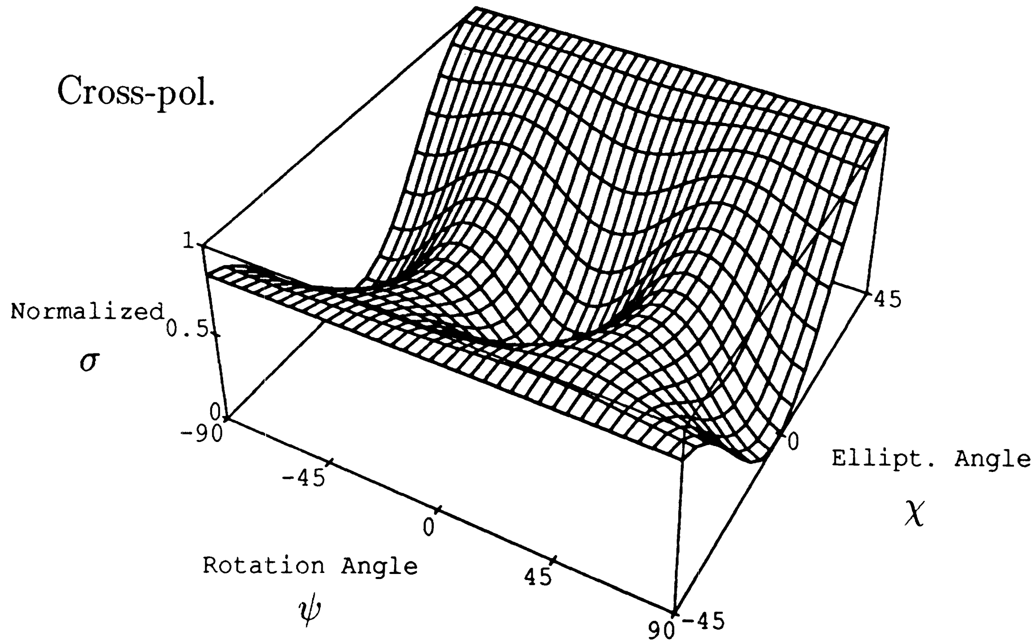


Figure 56: The co- and cross-polarized signatures of cable #4 (120 cm long) at 34.5 GHz and 24.4° incidence angle.

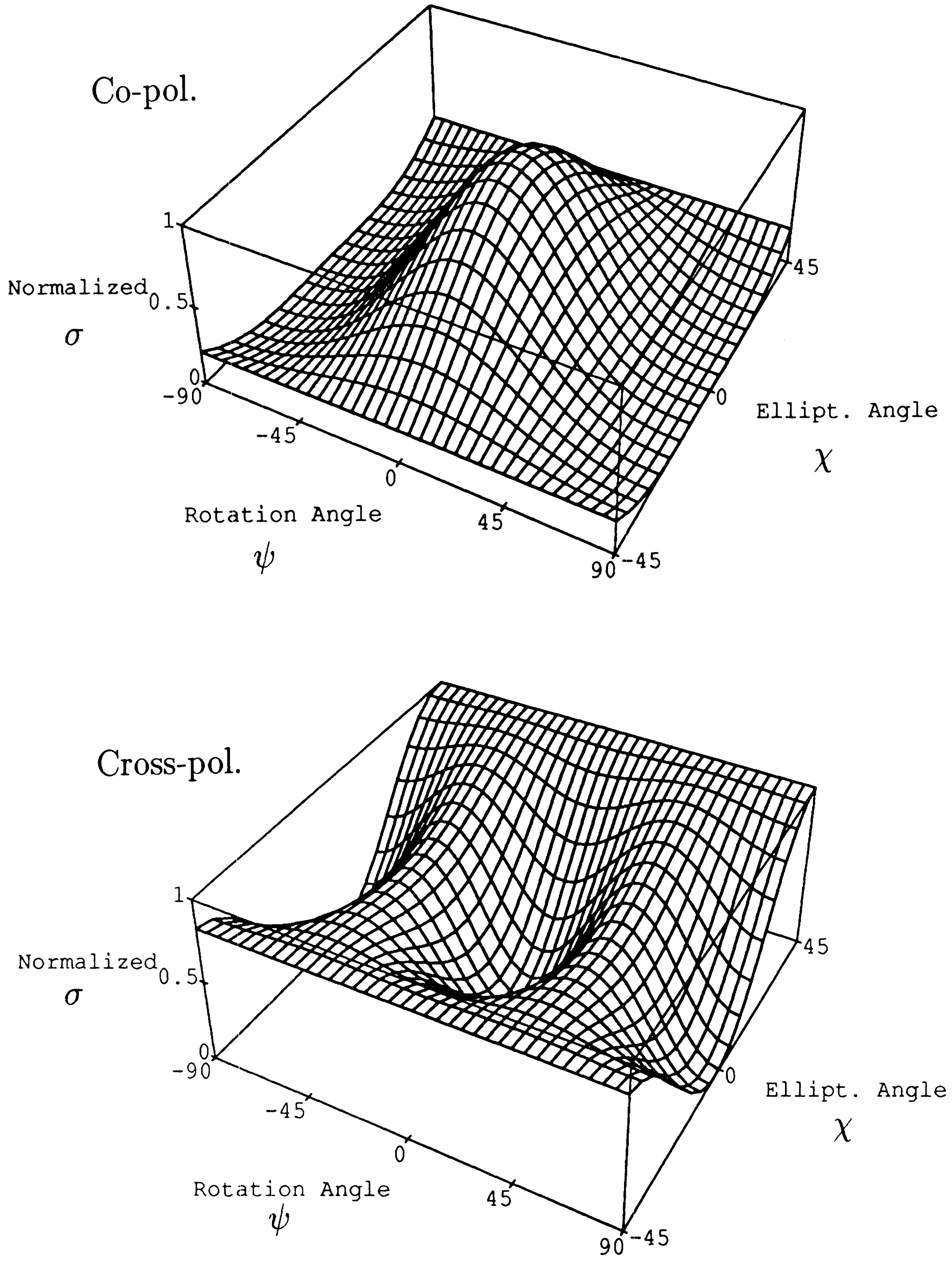


Figure 57: The co- and cross-polarized signatures of cable #4 (120 cm long) at 34.5 GHz and 13° incidence angle (Bragg angle).

APPENDIX A

POLARIMETRIC MEASUREMENT PROGRAMS

Data for polarimetric measurements of powerline cables are acquired automatically using two programs, CABLE_LCX for 4.75 GHz and 9.5 GHz measurements using LCX Polarimetric Scatterometers and CABLE_MMW for 34.5 GHz measurement using Millimeter-Wave Polarimetic Scatterometer.

These programs are written in HP BASIC to control the network analyzer based scatterometers as shown in the flowcharts of Fig. A-1 and A-2 and program lists.

To generate each RCS (radar cross section) pattern of an individual powerline cable or cylinder, four data sets were collected; 1) a target (powerline cable) data, 2) background data for the target, 3) a sphere data (reference target for calibration), and 4) background data for the sphere. The background data were subtracted from target data to increase SNR (signal to noise ratio). The data were averaged to reduce the effect of thermal noise by averaging factor of four for powerline measurements and ten for sphere (reference target) measurements. Time gating technique was also used in these programs to avoid the contribution of leakage signal return to the frequency domain data.

At the beginning of the program, parameters are selected manually, i.e., frequencies, number of traces (angles), number of averages, gate center, gate span, start angle for target rotation, increment angle, stop angle, rotation speed, etc. After parameter selection, the programs control the scatterometer systems to acquire polarimetric data automatically; i.e., rotating target, selecting frequencies and polarizations (VV, HH, HV, VH), taking data and storing the data set in disk drive, etc. The flow charts of data acquisition sequences are shown in Fig. A-1 and A-2.

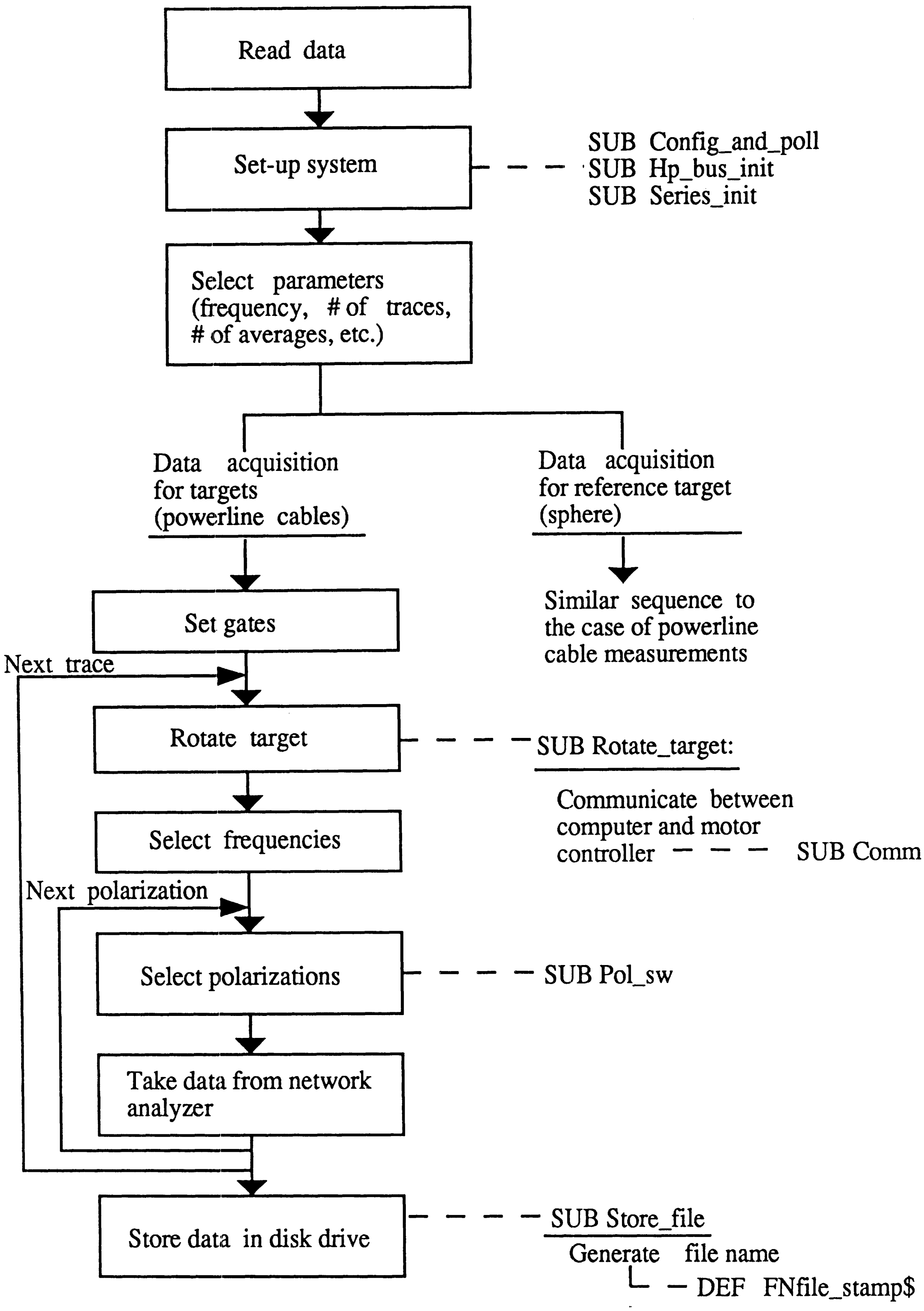


Figure A-1. Flow chart for CABLE_LCX

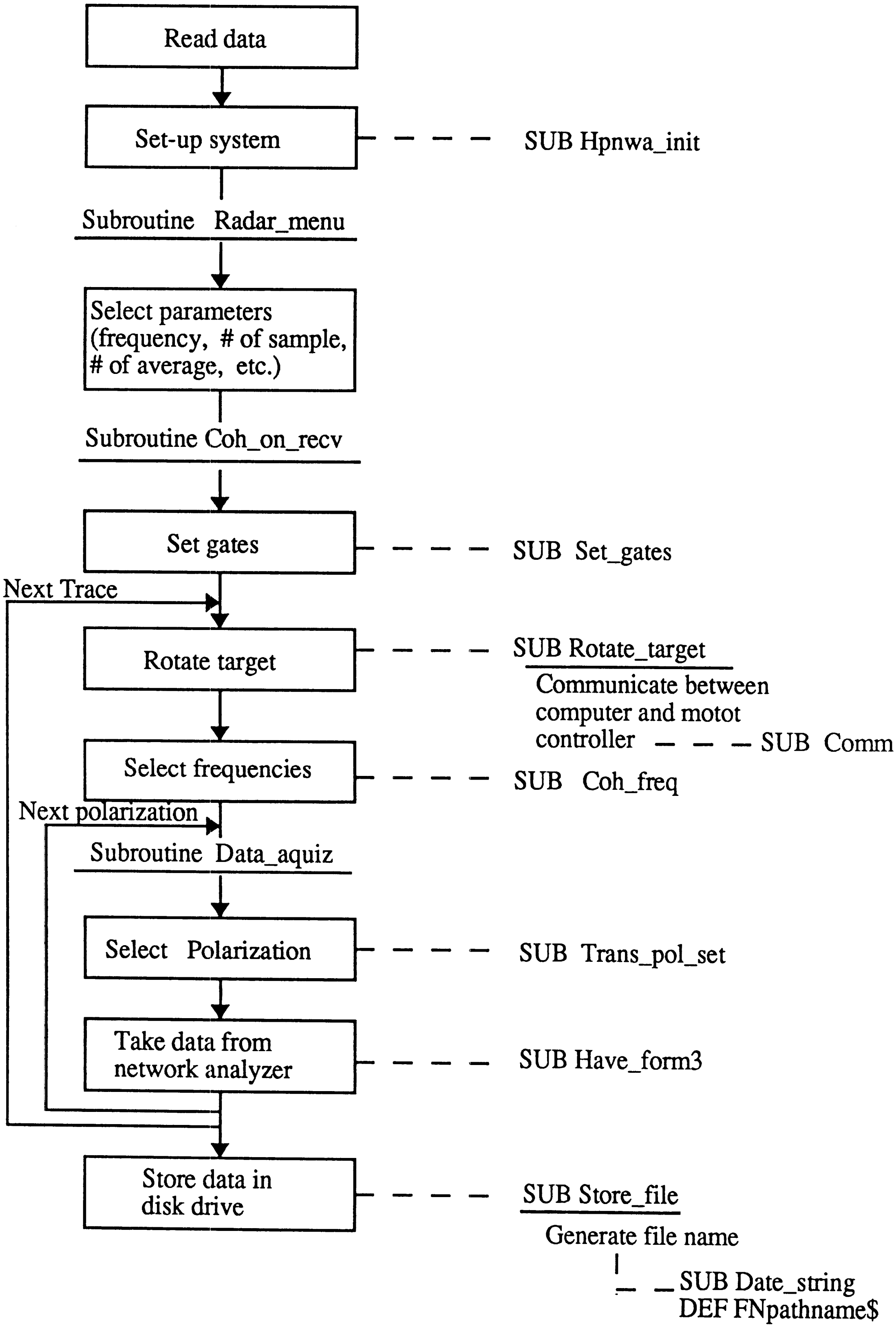


Figure A-2. Flow chart for CABLE_MMW

```
10   !**********************************************************************
20   !   L/C/X POLARIMETER MEASUREMENT PROGRAM
30   !       FILE: CABLE_LCX
40   !**********************************************************************
50   !   LAST EDIT: May 9, 1991      Changed for powerline cable 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$[10],Out_typ
e$[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$[10],Beam(3),INTEGER Npts,Ntrace,Average_factor
180  COM /Sys_7/ INTEGER Meas_flag(3)
190  COM /Com4/ INTEGER Rotation_state,REAL Inc_angle,Current_angle,Start_angle,Stop_an
gle,Old_home_angle,INTEGER Sets_per_pos
200  COM /Status/ INTEGER Sc,Connect_flg,E_flg,Debug_flg,Response$[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 1.25,1.75,1.5                                       ! FREQ_CENT
280  DATA .3,.5,.5                                            ! FREQ_SPAN
290  DATA 12.5,9.0,6.2                                        ! BEAMWIDTH
300  DATA "?*B3456","?*A34B56","?*A4B356","?*A3B456"          ! L
310  DATA "?*B3456","?*A56B34","?*A6B345","?*A5B346"          ! C
320  DATA "?*A34B56","?*B3456","?*A3B456","?*A4B356"          ! X
330  DATA ":,700,0",":,700,1",":MEMORY,0,7"                   ! DRIVE_A,B,C
340  DATA 100E-9,100E-9,100E-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  PRINT Meas_flag(*)
450  !
460  ! Set up error handling routine.
470  !
480  LOAD KEY "NOKEY:MEMORY,0,1"
490  MASS STORAGE IS "LCX:,700,0"
500  !
510  !
520  ! Initialize important parameters.
530  !
540  DEG
550  Rotation_state=-1
560  Current_angle=0.
570  MAT Meas_flag= (1)
580  Mode$="FAST ACQ"
590  F_disp=1
600  P_disp=1
610  Printer_flag=0
620  !
630  Hp_bus_init
640  IF Printer_flag=1 THEN Out_type$="PRINT/DISC"
650  !
660  Vel=2.99792458E+8
670  Ntrace=1
680  Npts=401
690  Nskip=40
700  Ndata=10
710  Average_factor=1
720  Angle$="0"
730  Angle=0
740  Ref_angle=0
750  Target$=""
760  Sound$="ON "
770  Debug_flag=0
780  Bell$=CHR$(7)
790  Exec_key$=CHR$(255)&CHR$(88)
800  Version$="Version 8.0 "
810  Exit_flag=0
820  Print_banner1
830  !
840  System_memory=VAL(SYSTEM$("AVAILABLE MEMORY"))
850  IF FNAsk("INITIALIZE RAM DISK?") THEN
860     INITIALIZE Drive_c$,0
870     INITIALIZE Drive_c$,INT((System_memory)/512)
880  ELSE
890     ASSIGN @Is_it_there TO Drive_c$;RETURN Outcome
900     IF Outcome=0 THEN
910        CAT Drive_c$;NO HEADER,COUNT Entries
920        IF Entries=0 THEN INITIALIZE Drive_c$,0
930     END IF
940     ASSIGN @Is_it_there TO *
950  END IF
960  !
970  Config_and_poll
980  OUTPUT @Nwa;"TIMDTRANON;LOGM;CONT;"
990  OUTPUT @Nwa;"POIN401;"
1000 Series_init
1010 !
1020 !
1030 Start_loop: !
1040           !
1050      Print_banner4
1060         ON KEY 0 LABEL "              ",FNTrap_level GOSUB Null
1070         ON KEY 1 LABEL " REFERENCE CAL",FNTrap_level GOSUB Ref_target
1080         ON KEY 2 LABEL "  TARGET RUN  ",FNTrap_level GOSUB Acq_target
1090         ON KEY 3 LABEL " SET FREQUENCY",FNTrap_level GOSUB Freq_set
1100         ON KEY 4 LABEL "    ANGLE     ",FNTrap_level GOSUB Set_angle
1110         ON KEY 5 LABEL " TARGET NAME  ",FNTrap_level GOSUB Set_target
1120         ON KEY 6 LABEL " # OF TRACES  ",FNTrap_level GOSUB Set_traces
1130         ON KEY 7 LABEL " # OF POINTS  ",FNTrap_level GOSUB Set_points
1140         ON KEY 8 LABEL "# OF AVERAGES ",FNTrap_level GOSUB Set_average
1150         ON KEY 9 LABEL "     QUIT     ",FNTrap_level GOTO Quit_fast_acq
1160      GOSUB Allocate_matrix
1170      LOOP
1180         EXIT IF Exit_flag=1
1190      END LOOP
1200      GOSUB Deallocate_mtrx
1210      Exit_flag=0
1220      GOTO Start_loop
1230      !
1240 Null: RETURN
1250 !
1260 !------------------------------------------------------------------
```

```
1270 !
1280 Ref_target: ! Acquire a reference target data set.
1290          !
1300      OFF KEY
1310      Clear_crt
1320      OUTPUT @Nwa;"TIMDTRANON; LOGM; GATEOFF;"
1330      OUTPUT @Nwa;"ELED 150NS;"
1340      OUTPUT @Nwa;"AUTO; STAR ONS; STOP 200NS;"
1350      PRINT TABXY(1,10);"Please point scatterometer assembly to reference  target."
1360      PRINT TABXY(1,12);"Press CONTINUE when ready..."
1370      PAUSE
1380      GOSUB Set_gates
1390      OUTPUT @Nwa;"TIMDTRANOFF; POLA; AVERFACT";VAL$(Average_factor);";"
1400      OUTPUT @Nwa;"AVEROON"
1410      INPUT "Enter the reference target angle: ",Ref_angle
1420      !
1430      ! Get the reference target response.
1440      !
1450      FOR T=1 TO Ntrace
1460      FOR F=1 TO 3
1470         IF Meas_flag(F) THEN
1480            Freq_set(F)
1490            Freq_sw(F)
1500                OUTPUT @Nwa;"GATEOFF;"
1510                OUTPUT @Nwa;"GATECENT";VAL$(Gate_cent(F));"S;"
1520                OUTPUT @Nwa;"GATESPAN";VAL$(Gate_span(F));"S;"
1530                OUTPUT @Nwa;"GATEON;"
1540                OUTPUT @Nwa;"TIMDTRANOFF;POLA;"
1550            FOR P=1 TO 4
1560                Pol_sw(F,P)
1570                OUTPUT @Nwa;"FORM3;NUMG";VAL$(Average_factor+1);";WAIT"
1580                OUTPUT @Nwa;"WAIT; OUTPFORM;"
1590                ENTER @Nwa_data1;Preamble,Bytes,Trace(*)
1600                MAT Target_response(1,P,*)= Trace
1610            NEXT P
1620                   Nskh=Nskip+1
1630            FOR P=1 TO 4
1640                   FOR Nt=Nskh TO Npts STEP Nskip
1650                       Nst=INT(Nt/Nskip)
1660                       Target_data(1,T,P,Nst)=Target_response(1,P,Nt)
1670                   NEXT Nt
1680            NEXT P
1690         END IF
1700      NEXT F
1710      NEXT T
1720      Store_file(Target_data(*),"REF",FNTime_stamp$,F)
1730      !
1740      ! Get the reference target mount response.
1750      !
1760      BEEP
1770      PRINT TABXY(1,10);"Please remove the reference target from its mount."
1780      PRINT TABXY(1,12);"Press CONTINUE when ready..."
1790      PAUSE
1800      Clear_crt
1810      PRINT TABXY(1,14);"Data for the mount is being collected .... "
1820      FOR T=1 TO Ntrace
1830      FOR F=1 TO 3
1840         IF Meas_flag(F) THEN
1850            Freq_set(F)
1860            Freq_sw(F)
1870                OUTPUT @Nwa;"GATEOFF;"
1880                OUTPUT @Nwa;"GATECENT";VAL$(Gate_cent(F));"S;"
1890                OUTPUT @Nwa;"GATESPAN";VAL$(Gate_span(F));"S;"
1900                OUTPUT @Nwa;"GATEON;"
1910            FOR P=1 TO 4
1920                Pol_sw(F,P)
1930                OUTPUT @Nwa;"FORM3;NUMG";VAL$(Average_factor+1);";WAIT"
1940                OUTPUT @Nwa;"WAIT; OUTPFORM;"
1950                ENTER @Nwa_data1;Preamble,Bytes,Trace(*)
1960                MAT Target_response(1,P,*)= Trace
1970            NEXT P
1980                   Nskh=Nskip+1
1990            FOR P=1 TO 4
2000                   FOR Nt=Nskh TO Npts STEP Nskip
2010                       Nst=INT(Nt/Nskip)
2020                       Target_data(1,T,P,Nst)=Target_response(1,P,Nt)
2030                   NEXT Nt
2040            NEXT P
2050         END IF
2060      NEXT F
2070      NEXT T
2080      Store_file(Target_data(*),"MNT",FNTime_stamp$,F)
2090      Pol_sw(F_disp,P_disp)
2100      DISP "Reference target mount response saved."
2110      Exit_flag=1
2120 RETURN
2130 !
2140 !
2150 !
2160 Acq_target: !
2170            !
2180      OFF KEY
2190      Clear_crt
2200      OUTPUT @Nwa;"TIMDTRANON; LOGM; GATEOFF;"
2210      OUTPUT @Nwa;"ELED 150NS;"
2220      OUTPUT @Nwa;"STAR ONS; STOP 200NS;"
2230      PRINT TABXY(1,10);"Please point scatterometer assembly at surface target."
2240      PRINT TABXY(1,12);"Press CONTINUE when ready..."
2250      PAUSE
2260      GOSUB Set_gates
2270      OUTPUT @Nwa;"TIMDTRANOFF; POLA; AVERFACT";VAL$(Average_factor);";"
2280      OUTPUT @Nwa;"GATEOFF;AVEROON;"
2290      !
2300      ! Get the target response.
2310      !
2320      FOR T=1 TO Ntrace
2330      !
2340      ! Get angles
2350      !
2360      IF T=1 THEN
2370         Rotation_state=-1
2380      ELSE
2390         Rotation_state=2
2400      END IF
2410      SELECT Rotation_state
2420      CASE =0
2430           Clear_crt
2440           PRINT TABXY(1,4);"When ready for measurement, press CONTINUE."
2450           BEEP
2460           PAUSE
2470           Clear_crt(3,16)
2480           PRINT TABXY(1,4);"Collecting data... "
2490      CASE ELSE
2500           PRINT TABXY(1,4);"Current angle is ";Current_angle;" degrees."
2510           Rotate_target    !
2520           WAIT 1
2530           Clear_crt(3,16)
2540           PRINT TABXY(1,4);"Collecting data ..."
```

```
2550      END SELECT
2560          FOR F=1 TO 3
2570           IF Meas_flag(F) THEN
2580             Freq_set(F)
2590             Freq_sw(F)
2600             OUTPUT @Nwa;"GATEOFF;"
2610             OUTPUT @Nwa;"GATECENT";VAL$(Gate_cent(F));"S;"
2620             OUTPUT @Nwa;"GATESPAN";VAL$(Gate_span(F));"S;"
2630             OUTPUT @Nwa;"GATEOON; WAIT;"
2640             PRINT Npts,Ntrace,Ndata
2650             PRINT "# OF TRACES LEFT=",Ntrace-T
2660             FOR P=1 TO 4
2670               Pol_sw(F,P)
2680               OUTPUT @Nwa;"NUMG";VAL$(Average_factor);";WAIT; FORM3; OUTPFORM;"
2690               ENTER @Nwa_data1;Preamble,Bytes,Trace(*)
2700               MAT Target_response(1,P,*)= Trace
2710             NEXT P
2720             Nskh=Nskip+1
2730             FOR P=1 TO 4
2740               FOR Nt=Nskh TO Npts STEP Nskip
2750                 Nst=INT(Nt/Nskip)
2760                 Target_data(1,T,P,Nst)=Target_response(1,P,Nt)
2770               NEXT Nt
2780             NEXT P
2790           END IF
2800         NEXT F
2810      NEXT T
2820      Store_file(Target_data(*),"GND",FNTime_stamp$,F)
2830      DISP "Surface target data saved."
2840      BEEP
2850        Rotation_state=4
2860        Rotate_target
2870      BEEP
2880      OUTPUT @Nwa;"CONT;"
2890      Exit_flag=1
2900 RETURN
2910 !
2920 !------------------------------------------------------------------------
2930 !
2940 Freq_set: GOSUB Deallocate_mtrx
2950           OFF KEY
2960          MAT Meas_flag_old= Meas_flag
2970          MAT Meas_flag= (0)
2980          Exit_flag=0
2990             ON KEY 0 LABEL "    L BAND    ",FNTrap_level GOSUB Set_l
3000             ON KEY 1 LABEL "    C BAND    ",FNTrap_level GOSUB Set_c
3010             ON KEY 2 LABEL "    X BAND    ",FNTrap_level GOSUB Set_x
3020             ON KEY 4 LABEL "    STORE     ",FNTrap_level GOTO Store_band
3030             ON KEY 5 LABEL "              ",FNTrap_level GOSUB Null
3040             ON KEY 6 LABEL "              ",FNTrap_level GOSUB Null
3050             ON KEY 7 LABEL "              ",FNTrap_level GOSUB Null
3060             ON KEY 8 LABEL "              ",FNTrap_level GOSUB Null
3070             ON KEY 9 LABEL "    CANCEL    ",FNTrap_level GOTO Cancel_band
3080          LOOP
3090             EXIT IF Exit_flag=1
3100          END LOOP
3110 RETURN
3120 Set_l:     OFF KEY 0
3130            Meas_flag(1)=1
3140            F_disp=1
3150 RETURN
3160 Set_c:     OFF KEY 1
3170            Meas_flag(2)=1
3180            F_disp=2
3190 RETURN
3200 Set_x:     OFF KEY 2
3210            Meas_flag(3)=1
3220            F_disp=3
3230 RETURN
3240 Store_band: Print_banner4
3250             Exit_flag=1
3260             GOSUB Allocate_matrix
3270 RETURN
3280 Cancel_band: !
3290              MAT Meas_flag= Meas_flag_old
3300              Exit_flag=1
3310              GOSUB Allocate_matrix
3320 RETURN
3330 !
3340 !------------------------------------------------------------------------
3350 !
3360 Set_angle:  !
3370             INPUT "Enter measurement angle: ",Angle
3380             Angle$=VAL$(Angle)&CHR$(179)&"   "    ! Degree sign.
3390             Print_banner4
3400 RETURN
3410 !
3420 !------------------------------------------------------------------------
3430 !
3440 Set_target: !
3450             LINPUT "Enter target type or name: ",Target$
3460             Target$=TRIM$(Target$)
3470             Target$=Target$&RPT$(" ",30-LEN(Target$))
3480             Print_banner4
3490 RETURN
3500 !
3510 !------------------------------------------------------------------------
3520 !
3530 Set_traces: !
3540             INPUT "Enter the number of traces( or angles) desired( >=3 ): ",Ntrace
3550             GOSUB Deallocate_mtrx
3560             GOSUB Allocate_matrix
3570             Print_banner4
3580 RETURN
3590 !
3600 !------------------------------------------------------------------------
3610 Set_points: !
3620             INPUT "Enter the number of sample points (Npts,201): ",Npts
3630             OUTPUT @Nwa;"POIN "&VAL$(Npts)&";"
3640             INPUT "Enter the data points to be stored (Ndata,10):",Ndata
3650             Nskip=INT(Npts/Ndata)
3660             Bytes=16*Ndata
3670             Print_banner4
3680             GOSUB Deallocate_mtrx
3690             GOSUB Allocate_matrix
3700 RETURN
3710 !
3720 !------------------------------------------------------------------------
3730 !
3740 Set_average: !
3750              INPUT "Enter averaging factor: ",Average_factor
3760              Print_banner4
3770 RETURN
3780 !
3790 !------------------------------------------------------------------------
3800 !
3810 Allocate_matrix: ! Allocate storage space for data.
3820                  !
```

```
3830      System_memory=VAL(SYSTEM$("AVAILABLE MEMORY"))
3840      Avail_traces=MIN(Ntrace,INT(System_memory-50000-3*4*16.*Npts)/(1*4*16.*Ndata))
3850      IF Avail_traces<Ntrace THEN
3860         BEEP
3870         PRINT TABXY(1,16);"Memory has capacity for only ";Avail_traces;" traces."
3880         PRINT "Press CONTINUE key to continue"
3890         PAUSE
3900         Ntrace=Avail_traces
3910      END IF
3920      ALLOCATE COMPLEX Trace(Npts),Target_response(1,4,Npts)
3930      ALLOCATE COMPLEX Target_data(1,Ntrace,4,Ndata)
3940   RETURN
3950 Deallocate_mtrx: ! Return to main program.
3960                 !
3970      DEALLOCATE Target_response(*),Trace(*)
3980      DEALLOCATE Target_data(*)
3990   RETURN
4000 !
4010 !-----------------------------------------------------------------------
4020 !
4030 Set_gates: ! Set gate centers and spans.
4040            !
4050    FOR F=1 TO 3
4060       IF Meas_flag(F) THEN
4070          Freq_set(F)
4080          Freq_sw(F)
4090          P=1
4100              Pol_sw(F,P)
4110              OUTPUT @Nwa;"TIMDTRANON; LOGM;"
4120              OUTPUT @Nwa;"ELED 150NS;"
4130              OUTPUT @Nwa;"STAR 0NS; STOP 200NS; WAIT;"
4140              OUTPUT @Nwa;"FORM3; OUTPACTI;"
4150              ENTER @Nwa;Gate_cent(F)
4160              OUTPUT @Nwa;"MARKOFF;"
4170              OUTPUT @Nwa;"CONT;"
4180              OUTPUT @Nwa;"GATESPAN";VAL$(Gate_span(F));"S;"
4190              OUTPUT @Nwa;"GATECENT";VAL$(Gate_cent(F));"S;"
4200              OUTPUT @Nwa;"KEY41; KEY59; KEY58; KEY59;"
4210              LOCAL @Nwa
4220              DISP "Adjust gate center to suit, and press CONTINUE."
4230              PAUSE
4240              OUTPUT @Nwa;"OUTPACTI;"
4250              ENTER @Nwa;Gate_cent(F)
4260              OUTPUT @Nwa;"GATESPAN";VAL$(Gate_span(F));";"
4270              OUTPUT @Nwa;"KEY41; KEY59; KEY58; KEY4;"
4280              LOCAL @Nwa
4290              DISP "Adjust gate span to suit, and press CONTINUE."
4300              PAUSE
4310              OUTPUT @Nwa;"OUTPACTI;"
4320              ENTER @Nwa;Gate_span(F)
4330       END IF
4340   NEXT F
4350   RETURN
4360   !
4370   !-----------------------------------------------------------------------
4380   !
4390 Quit_fast_acq: ! End of program
4400  DISP "PROGRAM EXIT"
4410  GOSUB Deallocate_mtrx
4420  LOAD KEY "EDITKEY:MEMORY,0,1"
4430  STOP
4440  END
4450  !
4460  !*********************************************************************
4470 !
4480 DEF FNAsk(Prompt$)
4490   OFF KEY
4500   DISP Prompt$;
4510   INPUT "",Yn$
4520   Yn$=UPC$(Yn$[1,1])
4530   SELECT Yn$
4540          CASE ="Y"
4550               RETURN 1
4560          CASE ="N",=""
4570               RETURN 0
4580          CASE ELSE
4590               RETURN 0
4600   END SELECT
4610 FNEND
4620 !
4630 !**********************************************************************
4640 !
4650 DEF FNFileloc$(File$,Dir$)
4660   INTEGER C  ! for the location of the ':' in Dir$ (minus 1)
4670   LET C=POS(Dir$,":")-1
4680   IF C<=0 THEN
4690     RETURN TRIM$(File$&Dir$)
4700   ELSE
4710     RETURN Dir$[1,C]&RPT$("/",Dir$[C,C]<>"/")&File$&Dir$[C+1,LEN(Dir$)]
4720   END IF
4730 FNEND ! Fileloc
4740 !
4750 !**********************************************************************
4760 !
4770  DEF FNTime_stamp$(OPTIONAL Time_format)
4780      !
4790      DIM Time_digits$[4],Year_digits$[6]
4800      DIM Machine_time$[8],Machine_date$[11]
4810      REAL Timedate_now
4820      !
4830      Timedate_now=TIMEDATE
4840      Machine_date$=DATE$(Timedate_now)
4850      Machine_time$=TIME$(Timedate_now)
4860      Time_digits$=Machine_time$[1,2]&Machine_time$[4,5]
4870      Year_digits$[1,2]=Machine_date$[10,11]
4880      IF Machine_date$[1,1]=" " THEN Machine_date$[1,1]="0"
4890      !
4900      SELECT Machine_date$[4,6]
4910      CASE ="Jan"
4920           Year_digits$[3,4]="01"
4930      CASE ="Feb"
4940           Year_digits$[3,4]="02"
4950      CASE ="Mar"
4960           Year_digits$[3,4]="03"
4970      CASE ="Apr"
4980           Year_digits$[3,4]="04"
4990      CASE ="May"
5000           Year_digits$[3,4]="05"
5010      CASE ="Jun"
5020           Year_digits$[3,4]="06"
5030      CASE ="Jul"
5040           Year_digits$[3,4]="07"
5050      CASE ="Aug"
5060           Year_digits$[3,4]="08"
5070      CASE ="Sep"
5080           Year_digits$[3,4]="09"
5090      CASE ="Oct"
5100           Year_digits$[3,4]="10"
```

```
5110       CASE ="Nov"
5120            Year_digits$[3,4]="11"
5130       CASE ="Dec"
5140            Year_digits$[3,4]="12"
5150       END SELECT
5160       !
5170       Year_digits$[5,6]=Machine_date$[1,2]
5180       SELECT NPAR
5190       CASE =0
5200            RETURN Year_digits$[5,6]&Time_digits$
5210       CASE =1
5220            IF Time_format=1 THEN
5230               RETURN Year_digits$&Time_digits$
5240            END IF
5250            IF Time_format=2 THEN
5260               RETURN Year_digits$[3,6]&Time_digits$
5270            END IF
5280       END SELECT
5290   FNEND
5300   !
5310   !**********************************************************************
5320   !
5330   DEF FNTrap_level
5340       RETURN VAL(SYSTEM$("SYSTEM PRIORITY"))+1
5350   FNEND
5360   !
5370   !**********************************************************************
5380   !
5390  SUB Config_and_poll
5400      COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
5410      COM /System/ System_memory
5420  !
5430  ! Find out what's out there.
5440  !
5450      ALLOCATE Device_list$(0:31)[20]
5460      ALPHA PEN 4
5470      KBD LINE PEN 3
5480      KEY LABELS PEN 5
5490      Clear_crt
5500      Netwrk_analyzer=0
5510      ALLOCATE Na_ident$[80]
5520      System_memory=VAL(SYSTEM$("AVAILABLE MEMORY")) ! How much memory for RAM-DISK
5530      PRINT "AVAILABLE MEMORY: ";System_memory;" BYTES"
5540      ON TIMEOUT 7,4 GOTO No_na       ! In case there is no network analyzer
5550 Is_na:  OUTPUT @Nwa;"FORM4; OUTPIDEN;"
5560         ENTER @Nwa_data2;Na_ident$
5570         IF POS(Na_ident$,"8510A") THEN Netwrk_analyzer=1
5580         IF POS(Na_ident$,"8510B") THEN Netwrk_analyzer=2
5590         IF POS(Na_ident$,"8720A") THEN Netwrk_analyzer=3
5600         IF POS(Na_ident$,"8720B") THEN Netwrk_analyzer=4
5610         IF POS(Na_ident$,"8753A") THEN Netwrk_analyzer=5
5620         IF POS(Na_ident$,"8753B") THEN Netwrk_analyzer=6
5630         LOCAL @Nwa
5640         PRINT
5650         PRINT Na_ident$
5660         PRINT Netwrk_analyzer
5670      PRINT
5680      PRINT
5690         IF Netwrk_analyzer=0 THEN
5700            !
5710            !
5720 No_na:     BEEP
5730            OFF CYCLE
5740            PRINT TABXY(1,5);"There is no active network analyzer on the HPIB bus."
5750            PRINT TABXY(1,6);"Please check connections, and press the RUN key."
5760            PRINT
5770            PRINT TABXY(1,7);"If you DO NOT want to use a network analyzer, press the
CONTINUE key."
5780            PAUSE
5790         END IF
5800         !
5810         !
5820 Check_hpib:  ! Check the rest of the bus
5830      ON TIMEOUT 7,.01 GOTO Nothing
5840      !
5850      FOR Device=700 TO 731
5860         DISP "Checking for device at address: ";Device
5870         Device_list$(Device-700)="NOTHING"
5880         ASSIGN @What_is_it TO Device
5890         Outcome=SPOLL(@What_is_it)
5900         Device_list$(Device-700)="SOMETHING"
5910         PRINT Device,"SOMETHING HERE","spoll: ";Outcome
5920         ASSIGN @What_is_it TO *
5930 Nothing: ! Skip to next device
5940      NEXT Device
5950      !
5960      OFF TIMEOUT 7
5970      ASSIGN @What_is_it TO *
5980      IF Device_list$(1)="SOMETHING" THEN
5990         DISP "Position the printer to Top-Of-Form and press CONTINUE..."
6000         PAUSE
6010         PRINTER IS PRT
6020         PRINT CHR$(27)&"&l1L"; ! Set Page Breaks
6030         Printer_flag=1
6040         PRINTER IS CRT
6050      END IF
6060      DEALLOCATE Na_ident$
6070      DEALLOCATE Device_list$(*)
6080      ABORT @Hpib
6090  SUBEXIT
6100  SUBEND
6110  !
6120  ! **********************************************************************
6130  !
6140  SUB Hp_bus_init
6150    COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
6160    COM /Sys_1/ Freq$(*),Freq_cent(*),Freq_span(*),Gate_cent(*),Gate_span(*)
6170    COM /Sys_2/ Pol$(*),Polsw$(*)
6180    COM /System_config/ INTEGER Printer_flag,Debug_flag,Version$,Mode$,Out_type$,Soun
d$,Bell$,Target$,Ref_target$
6190    !
6200    !  This subroutine configures the HP-IB bus and presets the HP8510.
6210    !
6220    ASSIGN @Hpib TO 7
6230    ASSIGN @Nwa TO 716
6240    ASSIGN @Nwa_data1 TO 716;FORMAT OFF
6250    ASSIGN @Nwa_data2 TO 716;FORMAT ON
6260    ASSIGN @Relay TO 710
6270    REMOTE @Hpib
6280    ABORT @Hpib
6290    CLEAR @Nwa
6300    IF Debug_flag=1 THEN OUTPUT @Nwa;"DEBUON;"
6310    IF Debug_flag=0 THEN
6320      OUTPUT @Nwa;"DEBUOFF;"
6330      OUTPUT @Nwa;"TITL """&Freq$(2)&" BAND   """
6340    END IF
6350  SUBEND
6360  !
```

```
6370  !*************************************************************************
6380  !
6390  SUB Series_init
6400  COM /System_config/ INTEGER Printer_flag,Debug_flag,Version$,Mode$,Out_type$,Sound
$,Bell$,Target$,Ref_target$
6410    DIM Input$[80]
6420    !
6430    !  This subroutine prints a header for the printout and sets the system
6440    !  date and time.
6450    !
6460    IF Printer_flag=1 THEN PRINTER IS PRT
6470    PRINT CHR$(12)
6480    Set_clock
6490    LINPUT "ENTER MEASUREMENT SERIES TITLE",Input$
6500    Preface$="*"&RPT$(" ",9)
6510    PRINT RPT$("*",70)
6520    PRINT Preface$&Input$
6530    LINPUT "ENTER OPERATOR NAME",Input$
6540    PRINT Preface$&Input$
6550    PRINTER IS CRT
6560    PRINT
6570    PRINT
6580    PRINT Preface$&"MEASUREMENT SERIES STARTED AT "&TIME$(TIMEDATE)
6590    PRINTER IS CRT
6600  SUBEND
6610  !
6620  !*************************************************************************
6630  !
6640  SUB Set_clock
6650      OPTION BASE 1
6660      INTEGER I
6670      DIM Chrono$[12],Month$(12)[3]
6680      Exec_key$=CHR$(255)&CHR$(88)
6690      READ Month$(*)
6700      DATA "JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP","OCT","NOV","DEC"
6710      OUTPUT KBD;"SCRATCH KEY "&Exec_key$;
6720      Clear_crt
6730      PRINT "                         Current system date: ";DATE$(TIMEDATE)
6740      PRINT "                         Current system time: ";TIME$(TIMEDATE)
6750      PRINT
6760 Ask: LINPUT "Enter date and time (YYMMDDHHMMss) :",Chrono$
6770      IF Chrono$="" AND DATE$(TIMEDATE)<>" 1 Mar 1900" THEN
6780         Clear_crt
6790         SUBEXIT
6800      END IF
6810      Year$=VAL$(1900+VAL(Chrono$[1,2]))
6820      IF (VAL(Chrono$[3,4])<=0 OR VAL(Chrono$[3,4])>12) THEN
6830         BEEP
6840         PRINT "Incorrect month value."
6850         GOTO Ask
6860      END IF
6870      Year$=Month$(VAL(Chrono$[3,4]))&" "&Year$
6880      Year$=Chrono$[5,6]&" "&Year$
6890      SET TIMEDATE (DATE(Year$))
6900      IF (VAL(Chrono$[7,8]))>23 THEN
6910         BEEP
6920         PRINT "Incorrect hour value."
6930         GOTO Ask
6940      END IF
6950      Day$=Chrono$[7,8]&":"
6960      IF VAL(Chrono$[9,10])>59 THEN
6970         BEEP
6980         PRINT "Incorrect minute value."
6990         GOTO Ask
7000      END IF
7010      Day$=Day$&Chrono$[9,10]&":"
7020      IF (LEN(Chrono$)>10 AND LEN(Chrono$)=12) THEN
7030         IF VAL(Chrono$[11,12])>59 THEN
7040            BEEP
7050            PRINT "Incorrect seconds value."
7060            GOTO Ask
7070         END IF
7080         Day$=Day$&Chrono$[11,12]
7090      ELSE
7100         Day$=Day$&"00"
7110      END IF
7120      SET TIME TIME(Day$)
7130      Clear_crt
7140  SUBEXIT
7150  SUBEND
7160  !
7170  !*************************************************************************
7180  !
7190  SUB Fix_error
7200  SELECT ERRN
7210  CASE ELSE
7220       PRINTER IS CRT
7230       PRINT "ERROR ";ERRN
7240       PRINT ERRM$
7250       PRINT " PROGRAM IS PAUSED.  FIX ERROR, IF POSSIBLE, AND CONTINUE."
7260       PAUSE
7270  END SELECT
7280  SUBEND
7290  !
7300  !*************************************************************************
7310  !
7320 SUB Clear_crt(OPTIONAL INTEGER Start_line,Num_of_lines)
7330      !
7340      INTEGER I
7350      DIM Clear_line$[80]
7360      Clear_line$="                                          "
7370      IF NPAR=0 THEN
7380         OUTPUT KBD;CHR$(255)&CHR$(75);
7390      ELSE
7400         PRINT TABXY(1,Start_line);"";RPT$(Clear_line$,Num_of_lines)
7410         PRINT TABXY(1,Start_line);"";
7420         SUBEXIT
7430      END IF
7440 SUBEND
7450 !
7460 !*************************************************************************
7470 !
7480 SUB Print_banner1
7490      Clear_crt
7500      PRINT
7510      PRINT
7520      PRINT TABXY(3,16);"**************************************************"
7530      PRINT TABXY(4,16);"*                                               *"
7540      PRINT TABXY(5,16);"*                      LCX                      *"
7550      PRINT TABXY(6,16);"*       UNIVERSITY OF MICHIGAN RADIATION LAB    *"
7560      PRINT TABXY(7,16);"*            L/C/X MEASUREMENT PROGRAM          *"
7570      PRINT TABXY(8,16);"*                  (VERSION 8.0)                *"
7580      PRINT TABXY(9,16);"*                                               *"
7590      PRINT TABXY(10,16);"*                 May 9, 1991                 *"
7600      PRINT TABXY(11,16);"*                                               *"
7610      PRINT TABXY(12,16);"**************************************************"
7620 SUBEXIT
7630 SUBEND
```

```
7640  !
7650  !************************************************************************
7660  !
7670  SUB Print_banner4
7680  COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
7690  COM /Constants/ Vel,Zero(*),Exec_key$
7700  COM /System_config/ INTEGER Printer_flag,Debug_flag,Version$,Mode$,Out_type$,Soun
d$,Bell$,Target$,Ref_target$
7710  COM /Sys_1/ Freq$(*),Freq_cent(*),Freq_span(*),Gate_cent(*),Gate_span(*)
7720  COM /Sys_2/ Pol$(*),Polsw$(*)
7730  COM /Sys_3/ INTEGER F_disp,P_disp
7740  COM /Sys_4/ Drive_a$,Drive_b$,Drive_c$,INTEGER Preamble,Bytes
7750  COM /Sys_5/ INTEGER Nskip,Ndata
7760  COM /Sys_6/ Ref_angle,Angle,Angle$,Beam(*),INTEGER Npts,Ntrace,Average_factor
7770  COM /Sys_7/ INTEGER Meas_flag(*)
7780      !
7790      !
7800   OFF KEY
7810   Clear_crt
7820   PRINT
7830   PRINT
7840   PRINT "          PARAMETER                    CURRENT VALUE"
7850   PRINT
7860   PRINT "                   FREQUENCY             ";
7870   FOR F=1 TO 3
7880     IF Meas_flag(F) THEN PRINT Freq$(F)&" ";
7890   NEXT F
7900   PRINT "       "
7910   PRINT "         "
7920   PRINT "                   ANTENNA ANGLE        "&Angle$
7930   PRINT "                   TARGET TYPE          "&Target$
7940   PRINT "                   MEASUREMENT MODE     "&Mode$&"       "
7950   PRINT "         "
7960   PRINT "                   # OF TRACES/SET      ";Ntrace
7970   PRINT "                   # OF SAMPLE POINTS   ";Npts
7980   PRINT "                   # OF DATA POINTS     ";Ndata
7990   PRINT "                    (to be stored)      "
8000   PRINT "                   # OF AVERAGES        ";Average_factor
8010  SUBEXIT
8020  SUBEND
8030  !
8040  !************************************************************************
8050  !
8060  SUB Store_file(COMPLEX Matrix(*),File_type$,Filename$,INTEGER F)
8070      !
8080      COM /Sys_1/ Freq$(*),Freq_cent(*),Freq_span(*),Gate_cent(*),Gate_span(*)
8090      COM /Sys_2/ Pol$(*),Polsw$(*)
8100      COM /Sys_5/ INTEGER Nskip,Ndata
8110      COM /Sys_6/ Ref_angle,Angle,Angle$,Beam(*),INTEGER Npts,Ntrace,Average_factor
8120      COM /Sys_7/ INTEGER Meas_flag(*)
8130      COM /System_config/ INTEGER Printer_flag,Debug_flag,Version$,Mode$,Out_type$,S
ound$,Bell$,Target$,Ref_target$
8140      !
8150      !
8160      INTEGER Records_per_set,T
8170      REAL Bytes_per_set
8180      DIM Suffix$[2]
8190      ALLOCATE COMPLEX Trace(Ndata)
8200      !
8210      !
8220      DISP "Saving file."
8230      SELECT File_type$
8240      CASE ="REF"
8250           Bytes_per_set=16*Ndata
8260           Records_per_set=4*SUM(Meas_flag)*Ntrace
8270            IF SUM(Meas_flag)=3 THEN
8280              Suffix$="RA"
8290            ELSE
8300             FOR F=1 TO 3
8310               IF Meas_flag(F)=1 THEN
8320                 Mf=F
8330               END IF
8340             NEXT F
8350             Suffix$="R"&Freq$(Mf)
8360            END IF
8370              GOSUB Save_upux
8380           !
8390           !
8400      CASE ="MNT"
8410           Bytes_per_set=16*Ndata
8420           Records_per_set=4*SUM(Meas_flag)*Ntrace
8430            IF SUM(Meas_flag)=3 THEN
8440              Suffix$="MA"
8450            ELSE
8460             FOR F=1 TO 3
8470               IF Meas_flag(F)=1 THEN
8480                 Mf=F
8490               END IF
8500             NEXT F
8510             Suffix$="M"&Freq$(Mf)
8520            END IF
8530              GOSUB Save_hpux
8540           !
8550           !
8560      CASE ="GND"
8570           Bytes_per_set=16*Ndata
8580           Records_per_set=Ntrace*4*SUM(Meas_flag)
8590            IF SUM(Meas_flag)=3 THEN
8600              Suffix$="GA"
8610            ELSE
8620             FOR F=1 TO 3
8630               IF Meas_flag(F)=1 THEN
8640                 Mf=F
8650               END IF
8660             NEXT F
8670             Suffix$="G"&Freq$(Mf)
8680            END IF
8690              GOSUB Save_hpux
8700      END SELECT
8710      DEALLOCATE Trace(*)
8720      SUBEXIT
8730           !
8740           !
8750 Save_hpux: !
8760                ! Save data file by HP_UX type.
8770                !
8780     IF NOT Debug_flag THEN
8790        CREATE Filename$&Suffix$&Drive_c$,200000
8800     END IF
8810     IF Debug_flag THEN
8820        ASSIGN @Disc TO PRT
8830        OUTPUT @Disc;"FILE: ",Filename$,Suffix$
8840        PRINT "Debug_flag= 1"
8850     ELSE
8860        ASSIGN @Disc TO Filename$&Suffix$&Drive_c$;FORMAT ON
8870        FOR T=1 TO Ntrace
8880          FOR F=1 TO 3
8890          IF Meas_flag(F)=1 THEN
```

```
8900           FOR P=1 TO 4
8910               MAT Trace= Matrix(1,T,P,*)
8920                OUTPUT @Disc;Trace(*)
8930            NEXT P
8940          END IF
8950         NEXT F
8960       NEXT T
8970     END IF
8980     ASSIGN @Disc TO *
8990 RETURN
9000  !
9010  !-----------------------------------------------------------------------
9020  !
9030 Save_traces: !
9040              ! Save the ground target data file.
9050              !
9060     IF NOT Debug_flag THEN
9070         CREATE BDAT Filename$&Suffix$&Drive_c$,Records_per_set,Bytes_per_set
9080         Base_record=0
9090     END IF
9100     IF Debug_flag THEN
9110        ASSIGN @Disc TO PRT
9120        OUTPUT @Disc;"FILE: ";Filename$;Suffix$
9130        OUTPUT @Disc USING Image_5;Ndata,Ntrace
9140        OUTPUT @Disc USING Image_3;Target$
9150        FOR T=1 TO Ntrace
9160          FOR F=1 TO 3
9170           IF Meas_flag(F)=1 THEN
9180            OUTPUT @Disc USING Image_1;Version$,Freq_cent(F),Freq_span(F)
9190            FOR P=1 TO 4
9200                OUTPUT @Disc USING Image_4;Pol$(P),Gate_cent(F),Gate_span(F),T
9210                MAT Trace= Matrix(1,T,P,*)
9220                OUTPUT @Disc;Trace(*)
9230            NEXT P
9240           END IF
9250          NEXT F
9260         NEXT T
9270     ELSE
9280        ASSIGN @Disc TO Filename$&Suffix$&Drive_c$;FORMAT OFF
9290        FOR T=1 TO Ntrace
9300          FOR F=1 TO 3
9310          IF Meas_flag(F)=1 THEN
9320            FOR P=1 TO 4
9330                MAT Trace= Matrix(1,T,P,*)
9340                 OUTPUT @Disc,Base_record+P;Trace(*)
9350             NEXT P
9360            Base_record=Base_record+4
9370           END IF
9380          NEXT F
9390        NEXT T
9400     END IF
9410     ASSIGN @Disc TO *
9420 RETURN
9430  !
9440  !-----------------------------------------------------------------------
9450  !
9460 Image_1: IMAGE (1X,12A,5X,"FREQ CENTER: ",2D.4D,5X,"FREQ SPAN: ",2D.4D)
9470 Image_2: IMAGE ("NUMBER OF POINTS: ",5D,5X,"NUMBER OF AVERAGES: ",5D)
9480 Image_3: IMAGE ("TARGET: ",30A,"GATING TARGET TYPE: ",2D)
9490 Image_4: IMAGE ("POLARIZATION: ",2A,5X,"GATE CENTER: ",SD.14DE,/,5X,"GATE SPAN: ",S
D.14DE,"TRACE: ",3D)
9500 Image_5: IMAGE ("NUMBER OF POINTS: ",5D,5X,"NUMBER OF TRACES: ",5D)
9510 Image_6: IMAGE (5X,SD.14DE,5X,SD.14DE)
9520 SUBEND
9530  !
9540  !*************************************************************************
9550  !
9560  SUB Freq_set(INTEGER Ifreq)
9570  COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
9580  COM /Sys_1/ Freq$(*),Freq_cent(*),Freq_span(*),Gate_cent(*),Gate_span(*)
9590    !
9600    !  This subroutine sets the transmit frequency for the HP8753.
9610    !
9620    IF Ifreq=1 THEN
9630        OUTPUT @Nwa;"POWE0"
9640    ELSE
9650        OUTPUT @Nwa;"POWE0"
9660    END IF
9670    SELECT Netwrk_analyzer
9680    CASE =3,=4,=5,=6
9690         OUTPUT @Nwa;"TIMDTRANOFF;"
9700    CASE =1,=2
9710         OUTPUT @Nwa;"FREQ;"
9720    END SELECT
9730    OUTPUT @Nwa;"CENT "&VAL$(Freq_cent(Ifreq))&" GHZ;"
9740    OUTPUT @Nwa;"SPAN "&VAL$(Freq_span(Ifreq))&" GHZ;"
9750  SUBEND
9760  !
9770  !*************************************************************************
9780  !
9790  SUB Freq_sw(INTEGER Ifreq)
9800  COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
9810    SELECT Ifreq
9820    CASE 1
9830      OUTPUT @Relay;"?*A2B1"
9840    CASE 2
9850      OUTPUT @Relay;"?*A1B2"
9860    CASE 3
9870      OUTPUT @Relay;"?*B12"
9880    END SELECT
9890    WAIT .1
9900  SUBEND
9910  !
9920  !************************************************************************
9930  !
9940  SUB Pol_sw(INTEGER Ifreq,Ipol)
9950      COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
9960      COM /Sys_1/ Freq$(*),Freq_cent(*),Freq_span(*),Gate_cent(*),Gate_span(*)
9970      COM /Sys_2/ Pol$(*),Polsw$(*)
9980    !
9990    !  This subroutine sets the transmit and receive polarization by
10000   !  sending the proper command over the HPIB to the polarization
10010   !  relays.
10020   !
10030   OUTPUT @Relay;Polsw$(Ifreq,Ipol)
10040   OUTPUT @Nwa;"TITL "" "&Freq$(Ifreq)&" BAND - "&Pol$(Ipol)&""""
10050   WAIT .1
10060 SUBEND
10070 !
10080 !************************************************************************
10090 !
10100 SUB Rotate_target
10110 OPTION BASE 1
10120 COM /Com4/ INTEGER Rotation_state,REAL Inc_angle,Current_angle,Start_angle,Stop_ang
le,Old_home_angle,INTEGER Sets_per_pos
10130 COM /Status/ INTEGER Sc,Connect_flg,E_flg,Debug_flg,Response$[80]
10140 INTEGER Fs_flag,Ss_flag,Speed,Imc_status,Confirm_answer
10150    !
```

```
10160     !
10170     Confirm_answer=1
10180     Imc_status=0
10190     Debug_flg=0
10200     Fs_flag=-1
10210     Ss_flag=-1
10220     Clear_crt(3,16)
10230     !
10240     !
10250     SELECT Rotation_state
10260     CASE =-1
10270          IF FNAsk("Do you wish to use the rotator?") THEN
10280             Connect_flg=0
10290             GOSUB Init_imc
10300             GOSUB Init_graph_pos
10310             GOSUB Manual_loop
10320             PRINT "Set Auto Mode Please....."
10330          ELSE
10340             Rotation_state=0
10350             GCLEAR
10360             GRAPHICS OFF
10370          END IF
10380     CASE =0
10390          SUBEXIT
10400     CASE =1
10410          GOSUB Check_position
10420          GOSUB Print_angles
10430          GOSUB Manual_loop
10440     CASE =2
10450          GOSUB Check_position
10460          GOSUB Auto
10470     CASE =3
10480          GOSUB Check_position
10490          GOSUB Manual_loop
10500          GOSUB Auto
10510     CASE =4
10520          GOSUB Check_position
10530          GOSUB Go_home
10540     CASE =5
10550          GOSUB Check_position
10560          Rotation_state=1  ! Switch to manual mode.
10570     END SELECT
10580     SUBEXIT
10590     !
10600     !
10610 Init_imc: ! Initialize the IMC unit.
10620     GOSUB Check_4_fault
10630     PRINT TABXY(1,3);"INITIALIZING IMC"
10640     Clear_crt(4,15)
10650     Comm("4WB")             ! Set warm boot (clear flags).
10660     PRINT TABXY(1,4);"WB"
10670     Comm("4EB")             ! Clear IMC buffer.
10680     PRINT TABXY(1,4);"EB"
10690     Encoder_ratio=4096  ! 32000
10700     Comm("4ER"&VAL$(Encoder_ratio))        ! Load encoder ratio.
10710     PRINT TABXY(1,4);"ER"&VAL$(Encoder_ratio)
10720     IF FNAsk("Do you wish to set home at the current position?") THEN
10730        Comm("4RS",Confirm_answer)
10740        ENTER Response$;Old_home_angle
10750        Old_home_angle=Old_home_angle/93.3
10760        Comm("4PIZ0")            ! Set IMC at 0.
10770        PRINT TABXY(1,4);"PIZ"&RPT$(" ",LEN(VAL$(Encoder_ratio)))
10780        Comm("4PIA0")            ! Set IMC at 0.
10790        PRINT TABXY(1,4);"PIA"
10800           Current_angle=0
10810        END IF
10820        Comm("4SP100")          ! Set speed to (50pps).
10830        PRINT TABXY(1,4);"SP "&RPT$(" ",LEN(VAL$(Encoder_ratio)))
10840        Comm("4AC500")          ! Set acceleration (500pps^2).
10850        PRINT TABXY(1,4);"AC "
10860        Comm("4DC500")          ! Set deceleration (500pps^2).
10870        PRINT TABXY(1,4);"DC "
10880        GOSUB Check_position
10890        Rotation_state=1
10900        Clear_crt
10910 !
10920 !
10930        PRINT TABXY(1,4);"DONE INITIALIZING IMC"
10940        PRINT TABXY(1,5);"Turntable currently in manual mode."
10950        PRINT TABXY(1,6)
10960 Print_angles:!
10970        PRINT TABXY(1,7);"Current  angle is: ";Current_angle;" degrees."
10980        PRINT TABXY(1,8);"Starting angle is: ";Start_angle;" degrees."
10990        PRINT TABXY(1,9);"Stopping angle is: ";Stop_angle;" degrees."
11000 RETURN
11010 !
11020 !
11030 Manual_loop:! Main activation loop.
11040        LOOP
11050          ON KEY 0 LABEL "FAST SLEW CW  ",FNTrap_level GOSUB Fs_cw
11060          ON KEY 1 LABEL "FAST SLEW CCW ",FNTrap_level GOSUB Fs_ccw
11070          ON KEY 5 LABEL "SLOW SLEW CW  ",FNTrap_level GOSUB Ss_cw
11080          ON KEY 6 LABEL "SLOW SLEW CCW ",FNTrap_level GOSUB Ss_ccw
11090          ON KEY 2 LABEL "MANUAL CONTROL",FNTrap_level GOSUB Manual
11100          ON KEY 3 LABEL "TARGET GO HOME",FNTrap_level GOSUB Go_home
11110          ON KEY 4 LABEL "STOP ROTATION ",FNTrap_level GOSUB Stop_turn
11120          ON KEY 7 LABEL "SET AUTO MODE ",FNTrap_level GOSUB Set_auto
11130          ON KEY 8 LABEL "SET TARGET HOME",FNTrap_level GOSUB Set_position
11140          ON KEY 9 LABEL "RETURN        ",FNTrap_level GOTO Quit
11150          GOSUB Check_position
11160        END LOOP
11170 !
11180 !---------------------------------------------------------------------
11190 !
11200 Fs_cw:! Fast slew clockwise.
11210          IF Fs_flag<0 THEN
11220             Comm("4SP500")
11230             Comm("4SFN")
11240             Fs_flag=-1*Fs_flag
11250             Clear_crt(3,15)
11260             PRINT TABXY(1,15);"ROTATING CW (FAST)"
11270          ELSE
11280             Comm("4ST")
11290             Fs_flag=-1*Fs_flag
11300             Clear_crt(3,15)
11310             PRINT TABXY(1,15);"ROTATION STOPPED"
11320             GOSUB Check_position
11330          END IF
11340 RETURN
11350 !
11360 !---------------------------------------------------------------------
11370 !
11380 Fs_ccw:! Fast slew counterclockwise.
11390          IF Fs_flag<0 THEN
11400             Comm("4ST")
11410             Comm("4SP500")
11420             Comm("4SRN")
11430             Fs_flag=-1*Fs_flag
```

```
11440         Clear_crt(3,10)
11450         PRINT TABXY(1,15);"ROTATING CCW (FAST)"
11460      ELSE
11470         Comm("4ST")
11480         Fs_flag=-1*Fs_flag
11490         Clear_crt(3,15)
11500         PRINT TABXY(1,15);"ROTATION STOPPED"
11510         GOSUB Check_position
11520      END IF
11530 RETURN
11540 !
11550 !----------------------------------------------------------------------
11560 !
11570 Ss_cw:! Slow slew clockwise.
11580      IF Ss_flag<0 THEN
11590         Comm("4ST")
11600         INPUT "Speed?",Sp
11610         Comm("4SP"&VAL$(INT(Sp)))
11620         Comm("4SFN")
11630         Ss_flag=-1*Ss_flag
11640         Clear_crt(3,15)
11650         PRINT TABXY(1,15);"ROTATING CW (SLOW)"
11660      ELSE
11670         Comm("4ST")
11680         Ss_flag=-1*Ss_flag
11690         Clear_crt(3,15)
11700         PRINT TABXY(1,15);"ROTATION STOPPED"
11710         GOSUB Check_position
11720      END IF
11730 RETURN
11740 !
11750 !----------------------------------------------------------------------
11760 !
11770 Ss_ccw:! Slow slew counterclockwise.
11780      IF Ss_flag<0 THEN
11790      INPUT "Speed?",Sp
11800         Comm("4SP"&VAL$(INT(Sp)))
11810         Comm("4SRN")
11820         Ss_flag=-1*Ss_flag
11830         Clear_crt(3,15)
11840         PRINT TABXY(1,15);"ROTATING CCW (SLOW)"
11850      ELSE
11860         Comm("4ST")
11870         Ss_flag=-1*Ss_flag
11880         Clear_crt(3,15)
11890         PRINT TABXY(1,15);"ROTATION STOPPED"
11900         GOSUB Check_position
11910      END IF
11920 RETURN
11930 !
11940 !----------------------------------------------------------------------
11950 !
11960 Manual: INPUT "ANGLE (IN DEGREES)=?",Inc_angle
11970         INPUT "SPEED? (~100-~500 RECOMMENDED)",Speed
11980         Comm("4SP"&VAL$(Speed))
11990 Auto:   SELECT Rotation_state
12000         CASE =4
12010              GOSUB Go_home
12020              Rotation_state=2
12030              GOTO Auto
12040         CASE ELSE
12050              Angl2=Inc_angle*93.3
12060              Angl1=INT(Angl2)
12070              IF Angl2-Angl1>=.5 THEN Angl1=Angl1+1
12080 !            Current_angle=Current_angle+Inc_angle
12090              Inc_angle$=VAL$(Angl1)
12100              Comm("4IM"&Inc_angle$)
12110              Comm("4RFI")
12120         END SELECT
12130         Imc_status=0
12140         Clear_crt(3,7)
12150         PRINT TABXY(1,14);"ROTATING TARGET, PLEASE WAIT."
12160         !
12170         !
12180         WHILE NOT BIT(Imc_status,0)           ! Wait for motor to stop.
12190               Comm("4RS",Confirm_answer)
12200               ENTER Response$;Imc_status
12210               PRINT TABXY(1,15);DVAL$(Imc_status,2)
12220               GOSUB Check_position
12230               WAIT 1
12240         END WHILE
12250         Imc_status=0
12260         !
12270         !
12280         Clear_crt(3,16)
12290         PRINT TABXY(1,16);"CURRENT TARGET POSITION IS ";Current_angle;" DEGREES.
              "
12300         WAIT 2  ! Wait for target settling.
12310 RETURN
12320 !
12330 !----------------------------------------------------------------------
12340 !
12350 Stop_turn:Comm("4ST")
12360           WHILE NOT BIT(Imc_status,0)           ! Wait for motor to stop.
12370                 Comm("4RS",Confirm_answer)
12380                 ENTER Response$;Imc_status
12390                 WAIT .1
12400           END WHILE
12410           Clear_crt(3,16)
12420           PRINT TABXY(1,15);"ROTATION STOPPED"
12430           GOSUB Check_position
12440           Imc_status=0
12450 RETURN
12460 !
12470 !----------------------------------------------------------------------
12480 !
12490 Set_auto: Comm("4SP500")
12500           GOSUB Check_position
12510           Clear_crt(3,16)
12520           PRINT TABXY(1,3);"Current starting  angle: ";Start_angle;" degrees"
12530           PRINT TABXY(1,4);"Current increment angle: ";Inc_angle;" degrees"
12540           PRINT TABXY(1,5);"Current stopping  angle: ";Stop_angle;" degrees"
12550           PRINT TABXY(1,6);"Current rotation  speed: ";Speed
12560           PRINT TABXY(1,7);RPT$(" ",80)
12570           PRINT TABXY(1,8);"Rotator  positioned  at: ";Current_angle;" degrees"
12580           INPUT "Enter starting angle value (degrees): ",Start_angle
12590           INPUT "Enter increment angle (degrees): ",Inc_angle
12600           INPUT "Enter stopping angle (degrees): ",Stop_angle
12610           INPUT "Enter rotation speed of target (~500 recommended): ",Speed
12620           Speed=INT(Speed)
12630           Comm("4SP"&VAL$(Speed))
12640           IF ABS(Start_angle-Current_angle)>.1 THEN
12650              PRINT TABXY(1,9);RPT$(" ",80)
12660              PRINT TABXY(1,10);"Rotating target to starting angle..."
12670              Temp_angle=Inc_angle
12680              Inc_angle=Start_angle-Current_angle
12690              GOSUB Auto
12700              Inc_angle=Temp_angle
```

```
12710          END IF
12720          Rotation_state=2
12730          Clear_crt
12740          PRINT TABXY(1,20);"Turntable is in automatic mode. (press the RETURN sof
tkey)"
12750 RETURN
12760 !
12770 !----------------------------------------------------------------------
12780 !
12790 Set_position:INPUT "LOCK IN CURRENT TARGET POSITION AS REFERENCE POSITION?",Yn$
12800           IF Yn$="Y" OR Yn$="y" THEN
12810              Comm("4RS",Confirm_answer)
12820              ENTER Response$;Old_home_angle
12830              Old_home_angle=Old_home_angle/93.3
12840              Comm("4PIA0") ! Set absolute position to zero.
12850              Comm("4PIZ0") ! Set incremental postition to zero.
12860              Current_angle=0
12870           ELSE
12880              PRINT "POSITION WAS NOT SET."
12890           END IF
12900 RETURN
12910 !
12920 !----------------------------------------------------------------------
12930 !
12940 Go_home: IF Speed<200 THEN Speed=200
12950          Comm("4SP"&VAL$(Speed))
12960          Comm("4AM0") ! Move to zero absolute position.
12970          Comm("4RAN") ! Initiate movement.
12980          Comm("4MW")  ! Make sure the move is completed.
12990          Imc_status=0
13000          Clear_crt(3,15)
13010          PRINT TABXY(1,14);"ROTATING TARGET TO HOME POSITION, PLEASE WAIT."
13020          WHILE NOT (BIT(Imc_status,0) AND BIT(Imc_status,5))
13030               GOSUB Check_status
13040               PRINT TABXY(1,15);"CURRENT STATUS: ";DVAL$(Imc_status,2)
13050               GOSUB Check_position
13060               WAIT .1
13070          END WHILE
13080          Clear_crt(3,16)
13090          PRINT TABXY(1,15);"TARGET AT HOME POSITION."
13100          GOSUB Check_position
13110          Imc_status=0
13120 RETURN
13130 !
13140 !----------------------------------------------------------------------
13150 !
13160 Check_status:! Keep an eye on the Whedco controller status.
13170          Comm("4RS",Confirm_answer)
13180          ENTER Response$;Imc_status
13190 RETURN
13200 !
13210 !----------------------------------------------------------------------
13220 !
13230 Check_position:! Get the current turnstile position in degrees.
13240          Comm("4RP",Confirm_answer)
13250          ENTER Response$;Motor_position
13260          Current_angle=Motor_position/93.3
13270    !     Current_angle=Current_angle+Inc_angle
13280          PRINT TABXY(1,16);"CURRENT TARGET POSITION IS ";Current_angle;" DEGREES.
"
13290          GOSUB Draw_positions
13300 RETURN
13310 !
13320 !----------------------------------------------------------------------
13330 !
13340 Check_4_fault: ! Check the IMC for a fault condition and correct or
13350                ! notify the user if necessary.
13360                !
13370      Comm("4FC",Confirm_answer)
13380      ENTER Response$;Fault$
13390      SELECT Fault$
13400      CASE ="Power failure" ! Loss of power
13410           RETURN
13420      CASE ="Force DAC" ! Force DAC command was given
13430           BEEP
13440           PRINT "Force DAC command was given..."
13450           DISP "Press CONTINUE to resume..."
13460           PAUSE
13470      RETURN
13480      CASE ="Over-current" ! Over-current condition exists.
13490           BEEP
13500           PRINT "An over-current condition has been detected on the IMC."
13510           PRINT
13520           PRINT "Cycle the power to the IMC until the OV-CUR LED goes out"
13530           DISP "Press CONTINUE to reinitialize the IMC"
13540           PAUSE
13550           GOSUB Init_imc
13560      RETURN
13570      END SELECT
13580      RETURN
13590 !
13600 !----------------------------------------------------------------------
13610 !
13620 Init_graph_pos: ! Creates a graphical depiction of where the target is.
13630                 !
13640     GINIT
13650     GCLEAR
13660     GRAPHICS ON
13670     SHOW 0,100,0,100
13680     PENUP
13690     MOVE 90,70
13700     PEN 1        ! Draw circle
13710     POLYGON 12,360,360
13720     PENUP
13730     MOVE 90,70  ! Draw old home orientation.
13740     PEN 2
13750     DRAW 90+11*COS(Old_home_angle),70-11*SIN(Old_home_angle)
13760     PENUP
13770     MOVE 90,70  ! Draw current home orientation.
13780     PEN 4
13790     DRAW 90,58
13800     PENUP
13810     MOVE 90,70  ! Draw current target orienation.
13820     PEN 3
13830     X_pos=90+11*COS(Current_angle)
13840     Y_pos=70-11*SIN(Current_angle)
13850     DRAW X_pos,Y_pos
13860 RETURN
13870 !
13880 !----------------------------------------------------------------------
13890 !
13900 Draw_positions:  ! Draws out the angular orienations.
13910     MOVE 90,70   ! Draw old home orientation.
13920     PEN 2
13930     DRAW 90-11*SIN(Old_home_angle),70-11*COS(Old_home_angle)
13940     PENUP
13950     MOVE 90,70  ! Draw current home orientation.
13960     PEN 4
```

```
13970     DRAW 90,58
13980     PENUP
13990     DISABLE
14000     MOVE 90,70  ! Draw current target orienation.
14010     PEN -3
14020     DRAW X_pos,Y_pos
14030     MOVE 90,70
14040     PEN 3
14050     X_pos=90-11*SIN(Current_angle)
14060     Y_pos=70-11*COS(Current_angle)
14070     DRAW X_pos,Y_pos
14080     PENUP
14090     ENABLE
14100 RETURN
14110 !
14120 !-----------------------------------------------------------------------
14130 !
14140 Quit: !
14150       SUBEXIT
14160 SUBEND
14170 !
14180 !***********************************************************************
14190 !
14200 SUB Comm(C$,OPTIONAL INTEGER Confirm_answer)
14210 !
14220 !  PROGRAM MODULE:  Comm
14230 !
14240 !  PURPOSE:         Modified version of the Comm module to be used
14250 !                   for direct two way communication with the WHEDCO
14260 !                   IMC stepping motor controller.
14270 !
14280 !  UPDATE:  3.0     Version 3.0 checks to see if the card being used
14290 !                   is the HP98628A (Datacomm) or the HP98626A (Serial).
14300 !                   Depending on which card is used, the appropriate
14310 !                   registers are selected.
14320 !
14330      OPTION BASE 1
14340      COM /Status/ INTEGER Sc,Connect_flg,E_flg,Debug_flg,Response$
14350      INTEGER Baud_rate,B,Num_chars,Response_flg,Index1
14360      DIM Input$[256],Term$[256],In$[256] BUFFER,From_232$[256]
14370      DIM Num_chars$[6],Num_ltrs$[6],Out$[256] BUFFER
14380      DIM White_print$[1],Crlf$[2]
14390      IF Debug_flg THEN PRINT TABXY(1,1);"ENTERING Comm "
14400      ON ERROR GOSUB Error
14410      !
14420      !
14430      !
14440      IF Connect_flg THEN After_init
14450      Sc=30
14460      ASSIGN @Find_it TO Sc;RETURN Outcome
14470      IF Outcome=0 THEN
14480         ASSIGN @Find_it TO *
14490         CONTROL Sc,0;1                   ! Reset RS-232 interface.
14500         CONTROL Sc,3;1                   ! Async link protocol.
14510         CONTROL Sc,0;1                   ! Set Async toggle.
14520         CONTROL Sc,8;1+2                 ! Set RTS and DTR lines.
14530         CONTROL Sc,16;0                  ! Disable connection timeout.
14540         CONTROL Sc,17;0                  ! Disable no activity timeout.
14550         CONTROL Sc,18;0                  ! Disable NO CARRIER timeout.
14560         CONTROL Sc,19;0                  ! Disable transmit timeout.
14570         CONTROL Sc,20;14                 ! TX baud speed = 9600
14580         CONTROL Sc,21;14                 ! RX baud speed = 9600
14590         CONTROL Sc,22;0                  ! No handshake with Whedco.
14600         CONTROL Sc,23;0                  ! No hardwired handshake.
14610         CONTROL Sc,34;2                  ! 7 bits/character.
14620         CONTROL Sc,35;0                  ! 1 stop bit.
14630         CONTROL Sc,36;1                  ! ODD parity.
14640         Connect_flg=1
14650      ELSE
14660         Sc=8
14670         ASSIGN @Find_it TO *
14680         ASSIGN @Find_it TO Sc;RETURN Outcome
14690         IF Outcome<>0 THEN
14700            PRINT "RS-232 card not installed. Please install and reboot."
14710            ASSIGN @Find_it TO *
14720            STOP
14730         END IF
14740         ASSIGN @Find_it TO *
14750         RESET Sc
14760         CONTROL Sc,0;1              ! Reset the RS-232 interface.
14770         CONTROL Sc,3;Baud_rate      ! Set the baud rate.
14780         CONTROL Sc,4;8+2            ! UART 8 bits/char.  ODD parity.
14790         CONTROL Sc,5;3              ! UART DTR line active.
14800         CONTROL Sc,12;128+32+16     ! Disable CD,DSR,CTS
14810         STATUS Sc,3;B               ! Confirm speed to user.
14820         Connect_flg=1
14830      END IF
14840 After_init:!
14850      White_print$=CHR$(136)
14860      Crlf$=CHR$(13)&CHR$(10)
14870      PRINT CHR$(128)&CHR$(136);     ! Set up the screen.
14880      ASSIGN @Screen TO CRT
14890      ASSIGN @Kbd TO KBD
14900      ASSIGN @Rx TO BUFFER In$
14910      ASSIGN @Tx TO BUFFER Out$
14920      ASSIGN @Uart_out TO Sc
14930      ASSIGN @Uart_in TO Sc
14940      Response_flg=0                 ! Reset command acknowledge flag.
14950      Response$=""                   ! Null out response string.
14960      !
14970      !
14980         ENABLE INTR Sc              ! Enable interrupt on card.
14990         TRANSFER @Tx TO @Uart_out;CONT ! Enable transfer buffers.
15000         TRANSFER @Uart_in TO @Rx
15010         ON INTR Sc,FNTrap_level GOSUB Read_loop ! Process card interrupts.
15020         IF C$<>"" THEN
15030            GOSUB Send_com           ! Send command out to controller.
15040         ELSE
15050            GOTO Quit                ! If null command, exit quick.
15060         END IF
15070         !
15080         !
15090         !
15100 Wait_for_it:WHILE NOT Response_flg     ! Waiting for acknowledgement.
15110                  GOSUB Read_loop
15120                  IF NPAR=2 THEN       ! We are waiting for data to be
15130                                       ! sent by the Whedco controller.
15140                     LOOP
15150                           GOSUB Read_loop
15160                           IF (POS(Response$,"*")) THEN
15170                              Response$=Response$[POS(Response$,"*"),LEN(Response$
))]
15180                              Response_flg=1
15190                           END IF
15200                     EXIT IF ((Response_flg=1) AND (POS(Response$,Crlf$)))
15210                     END LOOP
15220                  ELSE
15230                     WHILE NOT ((POS(Response$,"*")) OR (POS(Response$,"?")))
```

```
15240                    GOSUB Read_loop
15250                 END WHILE
15260                 Index1=POS(Response$,"*")
15270                 IF Index1=0 THEN ! Must be a "?" (Whedco command error).
15280                    ! Must be a "?" (Whedco command error).
15290                    E_flg=1       ! Notify via error flag.
15300                    Response_flg=1
15310                 ELSE
15320                    ! Normal command interpretation.
15330                    E_flg=0
15340                    Response_flg=1
15350                 END IF
15360              END IF
15370           END WHILE
15380       GOTO Quit
15390      !
15400      !
15410 Read_loop:! Read in serial data from Whedco.
15420            !
15430      STATUS @Rx,4;Num_chars               ! Number of characters to
15440      IF Num_chars=0 THEN RETURN           ! receive, if 0 try again.
15450      Num_chars$="#,"&VAL$(Num_chars)&"A"  ! Set up the IMAGE for ENTER.
15460      ENTER @Rx USING Num_chars$;From_232$       ! Transfer contents.
15470      Response$=Response$&From_232$              ! Build up dialogue.
15480      RETURN                                     ! Update pointers.
15490 !
15500 !
15510 Send_com:Term$=Crlf$[1,1]&C$&Crlf$
15520         Num_ltrs$="#,"&VAL$(LEN(Term$))&"A"
15530         OUTPUT @Tx USING Num_ltrs$;Term$
15540         Term$=""
15550         RETURN
15560 !
15570 !
15580 Quit:  OFF ERROR
15590        STATUS @Tx,10;Stat
15600        STATUS @Rx,4;Num_bytes
15610        ABORTIO @Uart_out
15620        ASSIGN @Tx TO *
15630        CONTROL @Rx,8;0
15640        STATUS @Rx,10;Stat
15650        STATUS @Rx,4;Num_bytes
15660        ABORTIO @Uart_in
15670        ASSIGN @Rx TO *
15680        DISABLE INTR Sc
15690        SUBEXIT
15700 !
15710 !
15720 Error:PRINT "HANDLING Comm ERROR"
15730        IF ERRN<>167 THEN Other_error
15740        IF Sc=8 THEN ! Process the simple card.
15750           STATUS Sc,10;Uart_error
15760           IF BIT(Uart_error,1) THEN Overrun
15770           IF BIT(Uart_error,2) THEN Parity
15780           IF BIT(Uart_error,4) THEN Break1
15790           IF BIT(Uart_error,3) THEN Framing
15800           E_flg=1
15810           PAUSE
15820           RETURN
15830         ELSE
15840           PRINT ERRM$
15850           E_flg=1
15860           PAUSE
15870           RETURN
15880         END IF
15890 !
15900 !
15910 Other:  PRINT "UART error status: ";Uart_error
15920         E_flg=1
15930         RETURN
15940 !
15950 !
15960 Overrun:PRINT "Overrun"
15970         E_flg=1
15980         RETURN
15990 !
16000 !
16010 Parity: PRINT "Parity"
16020         E_flg=1
16030         RETURN
16040 !
16050 !
16060 Break1: PRINT "Break"
16070         E_flg=1
16080         RETURN
16090 !
16100 !
16110 Framing:PRINT "Framing"
16120         E_flg=1
16130         RETURN
16140 !
16150 !
16160 Other_error:PRINT "Error message: ";ERRM$
16170             PAUSE
16180             E_flg=1
16190             SUBEXIT
16200 !
16210 !
16220 SUBEND
16230 !
16240 !**********************************************************************
```

```
10    !*********************************************************************
20    !   MILLIMETER-WAVE POLARIMETER MEASUREMENT PROGRAM
30    !       FILE: CABLE_MMW       VERSION 6.0
40    !*********************************************************************
50    !   LAST EDIT: 12 JUN 1991
60    !
70    !   NAME    DATE       VER.   CHANGE
80    !   RTA   08 FEB 1989  5.0    Update to take advantage of BASIC 5.11
90    !                             enhancements (MAT, COMPLEX).  Gate setting
100   !                             procedure improved.  Data files now in
110   !                             ASCII.  Electrical Delay always on.
120   !    MK   17 FEB 1989  5.1    Update to create RAM disc cache for holding
130   !   RTA                       various key definitions and MMWCAL
140   !                             files.
150   !                             Minor bug in file-disc error handling fixed.
160   !   RTA   05 JUN 1989  5.2    Extensive comments added.  Documentation
170   !                             developed.
180   !    AN    1 SEP 1989  5.3    Added the 215 radar system,the illumination in
190   !                             -tegrals routine was modified to handle the
200   !                             215 GHz radar.
210   !    MK   10 JAN 1990  5.4    Radiometer routine added.
220   !                             Better configuration poll routine.
230   !                             HFS capability added.
240   !    AN   15 FEB 1990  5.5    Radiometer routine was completed to control
250   !                             the lock_in amplifier.
260   !    AN   16 MAY 1990  5.6    Modified to work on Viper card
270   !    YO   12 JUN 1991  6.0    Added subroutines of Rotate_target, Comm,
280   !                             Set_gates, Store_file, etc. for powerline
290   !                             cable measurements.
300   !*********************************************************************
310   !                             VARIABLES USED                          *
320   !*********************************************************************
330   ! @Disc                Addr  Path used for disc I/O
340   ! @Nwa                 Addr  Path used for HP8510 I/O
350   ! @Nwa_data            Addr  Path used for HP8510 data transfer
360   ! Angle                Real  Angle of incidence in degrees
370   ! Angle$[10]           Char  Angle of incidence in degrees
380   ! Beam(1:4)            Real  Beamwidths of three radars
390   ! Debug$[3]            Char  ="ON" if HP8510 debug mode is enabled
400   ! End_flag             Intg  Set if QUIT was selected in Main Menu
410   ! Err_file$[20]        Char  Name of current file is stored here to be
420   !                            used in error messages in Fix_error routine
430   ! Fdef                 Intg  "Default" frequency (displayed at end of set)
440   ! Freq$(1:4)[3]        Char  Labels for the three frequencies ("35" etc.)
450   ! Freq_cent(1:4)       Real  Center IF frequency of three channels
460   ! Freq_span(1:4)       Real  Span of IF frequencies of three channels
470   ! HP8510               Intg  Set if 8510 is active network analyzer
480   ! Hp8753               Intg  Set if 8753 is active network analyzer
490   ! Kp(1:4,1:4,1:401)    Real  Cal constant for each freq/pol/trace point
500   ! Meas_flag(1:2,1:4)   Intg  Matrix denoting which channels will be used
510   !                            in measurement (format same as Cal_flag)
520   ! Msi$[80]             Char  Mass storage directory.
530   ! Npts                 Intg  Number of points in HP8510 trace (def. 401)
540   ! Ntrace               Intg  Number of traces per measurement (def. 30)
550   !                            ratio (usu. b1)
560   ! Out_type$[10]        Char  Type of data output (printer, disc, both)
570   ! Pdef                 Intg  "Default" polarization (displayed after set)
580   ! Pol$(1:4)[2]         Char  Labels for the 4 polarizations ("VV" etc.)
590   ! Target$[30]          Char  Target to be measured
600   ! Vel                  Real  Free-space velocity of light
610   ! Version$[12]         Char  Program version
620   ! Zero(1:4,1:4)        Real  Position of antenna faceplate on time-domain
630   !                            trace (varies due to differences in plumbing)
640   !*********************************************************************
650   !
660   ! Declare variables in common blocks
670   !
680   COM /Paths/ @Nwa,@Nwa_data,Hp8510,Hp8753
690   COM /Constants/ Vel,Zero(1:4,1:4),Far_field(1:4)
700   COM /Freq_data/ Freq$(1:4)[3],Freq_cent(1:4),Freq_span(1:4)
710   COM /Pol_data/ Pol$(1:4)[2],Numer$(1:4)[6]
720   COM /Trans_pol/ Tpol$(1:6)[4]
730   COM /Incoh_eled/ Polz(1:2,1:6)
740   COM /Radiometer1/ Rad_cal_stats(3,23),Rad_meas_stats(3,9),Calib_factors(3,2),Lockin
_stats(3,2)
750   COM /Radiometer2/ Cold_ref_temp,Hot_ref_temp,Pre_time_const(1:11),Post_time_const(1
:3)
760   COM /Labels/ Target$[30],Version$[12],Mode$[7],Out_type$[10],Debug$[3]
770   COM /Beam_data/ Angle,Angle$[10],Bin_rng(1:4),Beam(1:4)
780   COM /N_data/ INTEGER Npts,Ntrace
790   COM /Flags/ INTEGER Cal_flag(1:2,1:4),Meas_flag(1:2,1:4),Combinations,Fdef,Pdef
800   COM /Cal/ Sig_c(1:4,1:4),Pwr_cal(1:4,1:4),Delc(1:4),Rcal(1:4),Timc(1:4,1:4)
810   COM /Calmod/ Kp(1:4,1:4,1:401)
820   COM /Illum/ C(1:4,1:4,1:17),D(1:4,1:4,1:21),E(1:4,1:4,1:19)
830   COM /Errors/ Msi$[80],Err_file$[80]
840   COM /Gate_data/ Gate_cent,Gate_span,INTEGER Average_factor,Igate
850   COM /Com4/ INTEGER Rotation_state,REAL Inc_angle,Current_angle,Start_angle,Stop_ang
le,Old_home_angle,INTEGER Sets_per_pos
860   COM /Status/ INTEGER Sc,Connect_flg,E_flg,Debug_flg,Response$[80]
870   INTEGER End_flag
880   !
890   ! Load blank function keys (replacing the system typing-aid key defs)
900   !
910   LOAD KEY "NOKEY:MEMORY,0,1"
920   !
930   ! Data is read and assigned to arrays
940   !
950   DATA "035","094","140","215"                              ! Frequency
960   DATA "VV","HV","VH","HH"                                 ! Polarization
970   DATA "TV  ","T45 ","TH  ","TLHC","T135","TRHC"
980   DATA "NUMEB1","NUMEB1","NUMEB1","NUMEB1"                 ! Numerator
990   DATA 2.5,2.5,2.5,2.5                                     ! Freq_cent
1000  DATA 1.0,1.0,1.0,1.0                                     ! Freq_span
1010  DATA 4.2,1.8,5.0,2.37                                    ! Beamwidth*******
1020  DATA 10.,0.75,0.5,.25                                    ! Bin_rng********
1030  DATA 2.71,7.28,2.71,4.164                                ! Far_field*******
1040  READ Freq$(*)
1050  READ Pol$(*)
1060  READ Tpol$(*)
1070  READ Numer$(*)
1080  READ Freq_cent(*)
1090  READ Freq_span(*)
1100  READ Beam(*)
1110  READ Bin_rng(*)
1120  READ Far_field(*)
1130  !
1140  ! Enable error-trapping routine:
1150  ! When an error occurs, the program will jump to the Fix_error
1160  ! routine.  (Some errors can be fixed without stopping the program.)
1170  !
1180  ON ERROR CALL Fix_error
1190  !
1200  ! Read illumination integral coefficients from disc.
1210  ! Also define look-up table for FORM1 to FORM3 conversion.
1220  !
1230 ! IF FNViper THEN
1240 !  Msi$="\BLP\MMW:DOS,C"
1250 ! ELSE
```

```
1260    Msi$="MMW:CS80,700,0"
1270 ! END IF
1280  MASS STORAGE IS Msi$
1290  !
1300  ! Initialize important parameters
1310  !
1320  DEG                              ! Measure angles in degrees
1330  MAT Cal_flag= (0)
1340  MAT Meas_flag= (0)
1350  Mode$="SURFACE"                  ! Initial mode is SURFACE
1360  Out_type$="PRINT/DISC"           ! Initial output to printer & disc
1370  Debug$="OFF"                     ! HP8510 debug mode off
1380  Vel=2.99792458E+8
1390  Ntrace=30                        ! 30 traces/set
1400  Npts=401                         ! 401 points/trace
1410  Angle$=""
1420  Angle=0
1430  Fdef=1                           ! Default frequency = 35 GHz
1440  Pdef=1                           ! Default polarization = vv
1450  Target$=""
1460  Version$="Version 8.2 "
1470  Zero(1,1)=3.73E-7                ! Read electrical delays.  These
1480  Zero(1,2)=3.73E-7                ! corrections are necessary to make
1490  Zero(1,3)=3.73E-7                ! the antenna faceplate = 0.0 m
1500  Zero(1,4)=3.73E-7                ! in range.
1510  Polz(1,1)=3.73E-7
1520  Polz(1,2)=3.73E-7
1530  Polz(1,3)=3.73E-7
1540  Polz(1,4)=3.73E-7
1550  Polz(1,5)=3.73E-7
1560  Polz(1,6)=3.73E-7
1570  Zero(2,1)=3.7100E-7
1580  Zero(2,2)=3.7100E-7
1590  Zero(2,3)=3.7100E-7
1600  Zero(2,4)=3.7100E-7
1610  Polz(2,1)=3.71E-7
1620  Polz(2,2)=3.71E-7
1630  Polz(2,3)=3.71E-7
1640  Polz(2,4)=3.71E-7
1650  Polz(2,5)=3.71E-7
1660  Polz(2,6)=3.71E-7
1670  Zero(3,1)=9.1E-9
1680  Zero(3,2)=9.1E-9
1690  Zero(3,3)=9.1E-9
1700  Zero(3,4)=9.1E-9
1710  Zero(4,1)=3.8588E-7
1720  Zero(4,2)=3.8588E-7
1730  Zero(4,3)=3.8588E-7
1740  Zero(4,4)=3.8588E-7
1750  !
1760  ! "Zero(I,J)" is the two-way time interval from the t=0 of the
1770  ! HP8510 to the front of the antenna; i.e., if the front of the
1780  ! antenna appears at t=13.7 ns, then Zero=13.7E-9.  This delay is
1790  ! corrected using the ELECTRICAL DELAY function of the HP8510.
1800  !
1810  PRINTER IS CRT                   ! Direct printout to screen
1820  CLEAR SCREEN
1830  PRINT
1840  PRINT
1850  PRINT "              ************************************************"
1860  PRINT "              *                                              *"
1870  PRINT "              *                    MMWAVE                    *"
1880  PRINT "              *     University of Michigan Radiation Lab     *"
1890  PRINT "              *         Millimeter-Wave Polarimeter          *"
1900  PRINT "              *             Measurement Program              *"
1910  PRINT "              *                (Version 8.2)                 *"
1920  PRINT "              *                                              *"
1930  PRINT "              *                 JAN 28 1991                  *"
1940  PRINT "              *                                              *"
1950  PRINT "              ************************************************"
1960  !
1970  ! Subroutine configures the HP-IB bus and presets the HP8510.
1980  !
1990  PRINT " "
2000  PRINT " "
2010  PRINT " "
2020  PRINT "ENTER (Y) IF YOU WANT REMOTE POLARIZATION CONTROL"
2030  PRINT " "
2040  PRINT " "
2050  PRINT " "
2060  PRINT "NOTE: FOR COH-ON-RECV MEASUREMENTS YOU MUST BE IN REMOTE"
2070  PRINT "      WHILE FOR POWER MEASUREMENTS YOU MUST BE IN LOCAL"
2080  INPUT Pol_cont$
2090  CLEAR SCREEN
2100  !
2110  IF Pol_cont$="Y" OR Pol_cont$="y" THEN CALL Handshake
2120  !
2130  ! Establish connection with pol. control computer
2140  !
2150    PRINT "Be patient!  Need time to set-up..."
2160  CALL Hpnwa_init(@Nwa,@Nwa_data,@Hpib,Hp8510,Hp8753)
2170  !
2180  ! Print title, date, time, and comments to the printer.
2190  !
2200 Start_loop: !  Top of endless loop containing radar menu & meas. routine
2210  !
2220  ! Radar_menu is the main menu from which measurement parameters are
2230  ! chosen and calibration is done.  The subroutine is exited when
2240  ! beginning a measurement or ending the program.
2250  !
2260  CALL Radar_menu(End_flag)
2270  IF End_flag THEN GOTO Program_end
2280  !
2290  ! The following routines represent the type of measurement desired.
2300  ! Only the surface routine is updated for use on the HP8510.
2310  !
2320  !   IF Mode$="SURFACE" THEN CALL Surface
2330  !
2340  GOTO Start_loop                 ! After measurement, loop back to menu
2350 Program_end: !
2360  DISP "Program exit"
2370  LOAD KEY "SYSTKEY:MEMORY,0,1"   ! Re-load typing aid keys from memory
2380  END                             ! Program ends
2390 !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
2400  SUB Freq_set(INTEGER Ifreq,Timflag)
2410    !
2420    ! This subroutine prompts the user to change the radar frequency
2430    ! on the radar control panel.  It then sets the proper IF range
2440    ! on the HP8510.
2450    !
2460    ! This subroutine is called from the time or frequency domain.
2470    !
2480    !*******************************************************************
2490    !              VARIABLES USED  (* denotes subroutine argument)     *
2500    !*******************************************************************
2510    ! @Nwa                 Addr  Path used for HP8510 I/O
2520    ! Dummy                Real  Dummy variable; used for junk input
2530    ! Freq$(1:4)[3]        Char  Labels for the four frequencies ("35" etc.)
```

```
2540    ! Freq_cent(1:4)       Real  Center IF frequency of three channels
2550    ! Freq_span(1:4)       Real  Span of IF frequencies of three channels
2560    ! Hp8510               Intg  Set if 8510 is the active network analyzer
2570    ! Hp8753               Intg  Set if 8753 is the active network analyzer
2580    !*Ifreq                Intg  Frequency (1=35, 2=94, 3=140 ,4=215)
2590    !*Timflag              Intg  Set if subroutine called from time domain
2600    !*******************************************************************
2610    COM /Paths/ @Nwa,@Nwa_data,Hp8510,Hp8753
2620    COM /Freq_data/ Freq$(*),Freq_cent(*),Freq_span(*)
2630    DISP "Set to transmit "&Freq$(Ifreq)&" GHz  <CONTINUE>"
2640    PAUSE                       ! Wait for user to hit <CONTINUE>
2650    DISP "Working"              ! Let user know he has hit <CONTINUE>
2660    IF Timflag THEN             ! If in the time domain then...
2670         IF Hp8510 THEN
2680         OUTPUT @Nwa;"FREQ;"     ! ...switch to frequency domain.
2690         END IF
2700         IF Hp8753 THEN
2710         OUTPUT @Nwa;"TIMDTRANOFF;"
2720         END IF
2730    END IF
2740    IF Hp8510 THEN OUTPUT @Nwa;"FREQ;CENT "&VAL$(Freq_cent(Ifreq))&" GHZ;SPAN "&VAL$
(Freq_span(Ifreq))&" GHZ;ENTO;"          ! Set the IF center & span
2750    IF Hp8753 THEN OUTPUT @Nwa;"TIMDTRANOFF;CENT "&VAL$(Freq_cent(Ifreq))&" GHZ;SPAN
"&VAL$(Freq_span(Ifreq))&" GHZ;ENTO;"
2760    IF Timflag THEN            ! If subr. called from time domain then...
2770      !  ...switch back to the time domain, get the trace start time
2780      !  from the HP8510, store it in a dummy variable, and turn off the
2790      !  start time display on the HP8510 screen.  (Asking and waiting
2800      !  for the start time assures that the program will wait for the
2810      !  HP8510 to switch back to time domain before continuing.)
2820      IF Hp8510 THEN
2830      OUTPUT @Nwa;"TIMB;WAIT;STAR;OUTPACTI;"
2840      ENTER @Nwa;Dummy
2850      END IF
2860      IF Hp8753 THEN OUTPUT @Nwa;"TIMDTRANON;WAIT;"
2870      OUTPUT @Nwa;"ENTO"
2880    END IF
2890  SUBEND
2900  !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
2910  SUB Polariz_set(INTEGER Ipol,Ifreq)
2920    !
2930    !  This subroutine tells the user to set transmit polarization for
2940    !  the 35 GHz system (or transmit and receive polarization for the
2950    !  94 or 140 GHz systems) and gives the corresponding command
2960    !  to the HP8510.
2970    !
2980    !*******************************************************************
2990    !                VARIABLES USED  (* denotes subroutine argument)      *
3000    !*******************************************************************
3010    ! @Nwa                 Addr  Path used for HP8510 I/O
3020    !*Ifreq                Intg  Frequency (1=35, 2=94, 3=140, 4=215)
3030    !*Ipol                 Intg  Polarization (1=vv, 2=hv, 3=vh, 4=hh)
3040    ! Hp8510               Intg  Set if 8510 is the active network analyzer
3050    ! Numer$(1:4)[6]       Char  Channel to be used as numerator for HP8510
3060    !                            ratio (usu. b1)
3070    ! Pol$(1:4)[2]         Char  Labels for the 4 polarizations ("VV" etc.)
3080    !*******************************************************************
3090    COM /Paths/ @Nwa,@Nwa_data,Hp8510,Hp8753
3100    COM /Pol_data/ Pol$(*),Numer$(*)
3110  !!!!!!!!!!!!!!!
3120    SELECT Ifreq               ! Select from the following cases by frequency
3130     CASE 1,2       ! 35 OR 94 GHz
3140       DISP "Set to transmit "&Pol$(Ipol)[2,2]&" polarization  <CONTINUE>"
3150       PAUSE        ! Wait for user to press <CONTINUE>
3160     CASE 3,4       !  140 or 215 GHz
3170       DISP "Set the polarization config. to "&Pol$(Ipol)[1,2]&"   <CONTINUE>"
3180       PAUSE          ! Wait for user to press <CONTINUE>
3190     END SELECT
3200  !!!!!!!!!!!!!!
3210  SUBEND
3220  !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
3230  SUB Comments
3240    !
3250    ! This subroutine prints comments (text) on the printout.
3260    !
3270    !**********************************************************************
3280    !                VARIABLES USED  (* denotes subroutine argument)       *
3290    !**********************************************************************
3300    ! Input$[80]            Char  Stores user responses
3310    !**********************************************************************
3320    DIM Input$[80]
3330    CLEAR SCREEN
3340    PRINT "Enter comments.  When finished, begin the next line"
3350    PRINT "with a /, then <ENTER>."
3360    PRINT RPT$("*",70)                 ! Print row of * on screen
3370    PRINTER IS 26                      ! Direct printout to printer
3380    PRINT RPT$("*",70)                 ! Print row of * on printer
3390 Loop_start:  !
3400    LINPUT "",Input$                   ! Get a line of input from user
3410    IF Input$[1,1]="/" THEN Loop_end ! If line begins with "/", we're done
3420    PRINTER IS CRT                     ! Direct printout to screen
3430    PRINT Input$                       ! Print input line to screen
3440    PRINTER IS 26                      ! Direct printout to printer
3450    PRINT Input$                       ! Print input line to printer
3460    GOTO Loop_start                    ! Go back & get another line
3470 Loop_end:    !
3480    PRINT RPT$("*",70)                 ! Print row of * on printer
3490    PRINTER IS CRT                     ! Direct printout to screen
3500    CLEAR SCREEN
3510  SUBEND
3520  !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
3530  SUB Hpnwa_init(@Nwa,@Nwa_data,@Hpib,Hp8510,Hp8753)
3540    !
3550    ! This subroutine configures the HP-IB and sets several parameters.
3560    ! However, it does NOT set everything.  Instead, to save time at
3570    ! startup, most parameters and default states are recalled from
3580    ! instrument state #8.
3590    !
3600    !**********************************************************************
3610    !                VARIABLES USED  (* denotes subroutine argument)       *
3620    !**********************************************************************
3630    !*@Hpib                 Addr  Path used for HP-IB control
3640    !*@Nwa                  Addr  Path used for HP8510 I/O
3650    !*@Nwa_data             Addr  Path used for HP8510 data transfer
3660    ! Freq$(1:4)[3]         Char  Labels for the four frequencies ("35" etc.)
3670    !*Hp8510                Intg  Set if 8510 is active network analyzer
3680    !*Hp8753                Intg  Set if 8753 is active network analyzer
3690    ! Pol$(1:4)[2]          Char  Labels for the 4 polarizations ("VV" etc.)
3700    ! Zero(1:4,1:4)         Real  Position of antenna faceplate on
3710    !                             time-domain trace which varies due to
3720    !                             differences in plumbing
3730    ! Dev_id$[80]           Char  Device identification string used to
3740    !                             determine which network analyzer is
3750    !                             connected and if the lock-in amplifier is
3760    !                             there.
3770    !
3780    !**********************************************************************
3790    COM /Constants/ Vel,Zero(*),Far_field(*)
```

```
3800    COM /Freq_data/ Freq$(*),Freq_cent(*),Freq_span(*)
3810    COM /Pol_data/ Pol$(*),Numer$(*)
3820    ASSIGN @Hpib TO 7
3830    ALLOCATE Dev_id$[80]
3840   ASSIGN @Nwa TO 716                    ! Assign path for HP8510 commands
3850    ASSIGN @Nwa_data TO 716;FORMAT OFF   ! Assign path for data transfer
3860 Is_na: ! Check for existance and type of devices on the bus.
3870        Netwrk_analyzer=1  !!!!!!!!!!
3880        OUTPUT @Nwa;"FORM4; OUTPIDEN;"
3890        ENTER @Nwa;Dev_id$
3900        IF POS(Dev_id$,"8510") THEN
3910             Hp8510=1
3920        ELSE
3930             Hp8510=0
3940        END IF
3950        IF POS(Dev_id$,"8753") THEN
3960             Hp8753=1
3970        ELSE
3980             Hp8753=0
3990        END IF
4000        LOCAL @Nwa
4010       PRINT Dev_id$
4020 Setup:        OFF TIMEOUT  ! Continue with the setup procedures
4030               DEALLOCATE Dev_id$
4040    REMOTE @Hpib
4050    ABORT @Hpib
4060    CLEAR @Nwa
4070    OUTPUT @Nwa;"DEBUOFF;"             ! Disable debug mode
4080    IF Hp8510 THEN OUTPUT @Nwa;"RECA8;WAIT;ENTO;"
4090         ! Recall state 8
4100    IF Hp8753 THEN OUTPUT @Nwa;"RECA5;WAIT;ENTO;"
4110    OUTPUT @Nwa;"CORROFF;"             ! Make sure vector corr. is off
4120    IF Hp8510 THEN
4130    OUTPUT @Nwa;"DENOA1;LOCKA1;REDD;" ! Use a1 port
4140    END IF
4150    OUTPUT @Nwa;"REFV 0;REFP 10;"      ! Set ref. line position and value
4160 !  OUTPUT @Nwa;"ELED "&VAL$(Zero(1,1))&" S;ENTO;" ! Set electrical delay to value f
or 35 GHz, vv
4170    OUTPUT @Nwa;"TITL ""       FREQUENCY: "&Freq$(1)&" GHz, POLARIZATION: "&Pol$(1)&
"""" ! Set default screen title
4180  SUBEND
4190  !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
4200  SUB Radar_menu(INTEGER End_flag)
4210    !
4220    ! This is the main menu subroutine.  In this subroutine, measurement
4230    ! parameters are set, and calibration routines are called.  This
4240    ! subroutine is exited when beginning a measurement sequence or ending
4250    ! the program.
4260    !
4270    !**********************************************************************
4280    !                VARIABLES USED  (* denotes subroutine argument)       *
4290    !**********************************************************************
4300    ! Angle                     Real  Angle of incidence in degrees
4310    ! Angle$[10]                Char  Angle of incidence in degrees
4320    ! Angle_now                 Real  Angle corr. to resistance of Resist
4330    ! Debug$[3]                 Char  ="ON" if HP8510 debug mode is enabled
4340    !*End_flag                  Intg  Set if QUIT was selected in Main Menu
4350    ! Fdef                      Intg  "Default" freq. (displayed at end of set)
4360    ! Freq$(1:4)[3]             Char  Labels for the three freqs ("35" etc.)
4370    ! Hp8510                    Intg  Set when 8510 is the network analyzer
4380    ! Hp8753                    Intg  Set when 8753 is the network analyzer
4390    ! I                         Intg  Frequency loop index
4400    ! J                         Intg  Polarization loop index
4410    ! Kohm                      Real  Inclinometer resist. corr. to current ang
4420    ! Meas_flag(1:2,1:4)        Intg  Matrix denoting which channels will be
4430    !                                 used in measurement (fmt like Cal_flag)
4440    ! Meas_flag_old(1:2,1:4)  Intg  Previous values in Meas_flag matrix
4450    ! No_rec                    Real  Used in sphere response section
4460    ! Npts                      Intg  # of points in HP8510 trace (def. 401)
4470    ! Ntrace                    Intg  # of traces per measurement (def. 30)
4480    ! Out_type$[10]             Char  Type of output (printer, disc, both)
4490    ! Pdef                      Intg  "Default" pol (displayed after set)
4500    ! Pol$(1:4)[2]              Char  Labels for the 4 pols ("VV" etc.)
4510    ! Resist                    Intg  Inclinometer resistance
4520    ! Target$[80]               Char  Target to be measured
4530    ! Zero(1:4,1:4)             Real  Position of antenna faceplate on
4540    !                                 time-domain trace (varies due to
4550    !                                 differences in plumbing)
4560    !**********************************************************************
4570    COM /Paths/ @Nwa,@Nwa_data,Hp8510,Hp8753
4580    COM /Constants/ Vel,Zero(*),Far_field(*)
4590    COM /Freq_data/ Freq$(*),Freq_cent(*),Freq_span(*)
4600    COM /Pol_data/ Pol$(*),Numer$(*)
4610    COM /Labels/ Target$,Version$,Mode$,Out_type$,Debug$
4620    COM /Beam_data/ Angle,Angle$,Bin_rng(*),Beam(*)
4630    COM /N_data/ INTEGER Npts,Ntrace
4640    COM /Flags/ INTEGER Cal_flag(*),Meas_flag(*),Combinations,Fdef,Pdef
4650    COM /Gate_data/ Gate_cent,Gate_span,INTEGER Average_factor,Igate
4660    INTEGER I,J,T,Re,Im,Num,Flag,Iii
4670    ALLOCATE INTEGER Cal_flag_old(1:2,1:4)
4680    ALLOCATE INTEGER Meas_flag_old(1:2,1:4)
4690    End_flag=0
4700 Menu_clear:   ! Selections return to here when screen needs clearing
4710    OFF KEY                    ! Turn off soft keys
4720    CLEAR SCREEN
4730 Menu:  !         Selections return here when clearing is not wanted
4740    OUTPUT KBD;CHR$(255)&CHR$(84);  ! home display
4750    PRINT TABXY(1,1)                  ! Set print position to upper left corner
4760    PRINT                             ! The following lines print the main menu
4770    PRINT
4780    PRINT "                                MILLIMETER-WAVE"
4790    PRINT "                             Parameter Selection Menu"
4800    PRINT
4810    PRINT
4820    PRINT
4830    PRINT "                       PARAMETER              CURRENT VALUE"
4840    PRINT
4850    PRINT "                       Frequency              ";
4860    FOR I=1 TO 4
4870      ! List all frequencies currently selected
4880      IF Meas_flag(1,I) THEN PRINT Freq$(I)&" ";
4890    NEXT I
4900    PRINT "       "
4910    PRINT "                       Polarization           ";
4920    FOR J=1 TO 4
4930      ! List all polarizations currently selected
4940      IF Meas_flag(2,J) THEN PRINT Pol$(J)&" ";
4950    NEXT J
4960    PRINT "        "
4970    !
4980    ! Resistance value corresponding to antenna angle is calculated.
4990    !
5000    Kohm=(Angle-166.91813)/(-90.03136)
5010    PRINT "                       Antenna angle          "&Angle$;
5020    IF Angle$<>"" THEN    ! If angle has been set, print the corresponding
5030      PRINT "  R = "&VAL$(PROUND(Kohm,-4))&" K      " ! resistance,
5040    ELSE                  ! else print blank spaces.
5050      PRINT "   "
```

```
5060    END IF
5070    PRINT "                         Target type            "&Target$
5080    PRINT
5090    PRINT
5100    PRINT " ";
5110    PRINT " "
5120    PRINT "                         NOTE: Calibration & Power measurements for"
5130    PRINT "                               the 35 & 94 Radars are performed for only"
5140    PRINT "                               VV & VH  Polarizations ."
5150    !
5160    ! The following sets the function keys so that the operator can
5170    ! choose to set parameters, calibrate, or begin measurements.
5180    !
5190 Menu_1:  !  Main menu, page 1
5200    ! De-activate previous key definitions, if any.
5210    OFF KEY
5220               ! Define key 0 for selection of frequencies & polarizations
5230    ON KEY 0 LABEL " FREQ/POLARIZ " GOTO Freq_pol
5240               ! Define key 1 for selection of freq & pol to display immediately
5250    ON KEY 1 LABEL "DISPLAY    " GOTO Display
5260    ! Define key 2 to allow selection of an angle or finding an angle
5270    ON KEY 2 LABEL "ANGLE...." GOTO Angle
5280    ! Define key 3 for entering the target
5290    ON KEY 3 LABEL "TARGET...." GOTO Target
5300    ! Define key 4 to begin a measurement (and leave the menu subroutine)
5310    ON KEY 4 LABEL "BEGIN......" GOTO Begin
5320    ! Define key 5 to switch to the 2nd set of function keys
5330    ON KEY 5 LABEL "MORE....." GOTO Menu_2
5340    ! Define key 6 to perform a calibration
5350    ON KEY 6 LABEL "#OF AVERAGE" GOTO Set_average
5360    ! Define key 7 for selection of measurement mode
5370    ON KEY 7 LABEL "            " GOTO Null
5380    ! Define key 8 for printing of comments on printout
5390    ON KEY 8 LABEL "COMMENTS    " GOTO Comment_print
5400    ! Define key 9 to quit the program (and exit menu subroutine)
5410    ON KEY 9 LABEL "QUIT      " GOTO Quit
5420 Spin:GOTO Spin            ! Wait for user to press a softkey
5430 Menu_2:  !  Main menu, page 2
5440    ! De-activate previous key definitions (from page 1)
5450    OFF KEY
5460    ! Define key 1 for selection of various measurement parameters
5470    ON KEY 1 LABEL "PARAMETERS  " GOTO Parameters
5480    ON KEY 3 LABEL "SIX_POL.'S  " GOTO Menu_3
5490    ! Define key 5 to switch to 1st set of function keys
5500    ON KEY 5 LABEL "MORE       " GOTO Menu
5510    ON KEY 9 LABEL "FREQUENCY     " GOTO Manual_frq
5520    GOTO Spin                  ! Wait for user to press a softkey
5530  !-----------------------------------------------------------------
5540 Manual_frq:!
5550    OFF KEY
5560    CLEAR SCREEN
5570    PRINT "CHOOSE THE NUMBER CORRESPONDING FOR THE WANTED FREQUENCY"
5580    PRINT "(1) 35  GHz"
5590    PRINT "(2) 94  GHz"
5600    PRINT "(4) 215 GHz"
5610    INPUT Iii
5620    CALL Coh_freq(Iii)
5630    CLEAR SCREEN
5640    GOTO Menu_2
5650  !-----------------------------------------------------------------
5660 Menu_3: !
5670    OFF KEY
5680    ON KEY 1 LABEL "V_POL   " GOTO V_pol
5690    ON KEY 2 LABEL "45_POL  " GOTO T45_pol
5700    ON KEY 3 LABEL "H_POL   " GOTO H_pol
5710    ON KEY 4 LABEL "LHC_POL " GOTO Lhc_pol
5720    ON KEY 5 LABEL "135_POL " GOTO T135_pol
5730    ON KEY 6 LABEL "RHC_POL " GOTO Rhc_pol
5740    ON KEY 7 LABEL "RETURN  " GOTO Menu_2
5750    GOTO Spin
5760 V_pol:!
5770    CALL Trans_pol_set(1)
5780    GOTO Menu_3
5790 T45_pol:!
5800    CALL Trans_pol_set(2)
5810    GOTO Menu_3
5820 H_pol:!
5830    CALL Trans_pol_set(3)
5840    GOTO Menu_3
5850 T135_pol:!
5860    CALL Trans_pol_set(5)
5870    GOTO Menu_3
5880 Lhc_pol:!
5890    CALL Trans_pol_set(4)
5900    GOTO Menu_3
5910 Rhc_pol:!
5920    CALL Trans_pol_set(6)
5930    GOTO Menu_3
5940  !-----------------------------------------------------------------
5950 Null: RETURN
5960  !-----------------------------------------------------------------
5970 Set_average:    !
5980   INPUT "Enter averaging factor:",Average_factor
5990   GOTO Menu     ! Return to main menu
6000  !-----------------------------------------------------------------
6010 Freq_pol:  ! Choose frequencies and polarizations for measurements
6020    OFF KEY  ! De-activate softkeys
6030    MAT Meas_flag_old= Meas_flag     ! Save old set of selected freqs & pols
6040    MAT Meas_flag= (0)               ! Clear new set of freqs & pols
6050    !
6060    ! As each frequency and polarization is selected, the corresponding
6070    ! softkey is blanked, indicating that the key has been pressed.
6080    ! Pressing "STORE" will save the new selections, "CANCEL" will
6090    ! restore the old selections and return to the main menu.
6100    !
6110    ON KEY 0 LABEL "35 GHz...." GOTO Set_35
6120    ON KEY 1 LABEL "94 GHz...." GOTO Set_94
6130    ON KEY 2 LABEL "140 GHz..." GOTO Set_140
6140    ON KEY 3 LABEL "215 GHz..." GOTO Set_215
6150    ON KEY 4 LABEL "VV........" GOTO Vv_set
6160    ON KEY 5 LABEL "HV (rt)..." GOTO Hv_set
6170    ON KEY 6 LABEL "STORE...." GOTO Store
6180    ON KEY 7 LABEL "CANCEL...." GOTO Can
6190    ON KEY 8 LABEL "VH (rt)..." GOTO Vh_set
6200    ON KEY 9 LABEL "HH........" GOTO Hh_set
6210  !
6220  !
6230 GOTO Spin
6240 Set_35:   !
6250    OFF KEY 0              ! Turn off 35 GHz key
6260    Meas_flag(1,1)=1       ! Set 35 GHz flag
6270    GOTO Spin              ! Wait for user to press a softkey
6280 Set_94: !
6290    OFF KEY 1              ! Turn off 94 GHz key
6300    Meas_flag(1,2)=1       ! Set 94 GHz flag
6310    GOTO Spin              ! Wait for user to press a softkey
6320 Set_140: !
6330    OFF KEY 2              ! Turn off 140 GHz key
```

```
6340    Meas_flag(1,3)=1          ! Set 140 GHz flag
6350    GOTO Spin                 ! Wait for user to press a softkey
6360 Set_215:!
6370    OFF KEY 3                 ! TURN OFF 215 GHZ KEY
6380    Meas_flag(1,4)=1          ! SET 215 GHZ FLAG
6390    GOTO Spin                 !WAIT FOR USER TO PRESS A SOFTKEY
6400 Vv_set:  !
6410    OFF KEY 4                 ! Turn off vv key
6420    Meas_flag(2,1)=1          ! Set vv flag
6430    GOTO Spin                 ! Wait for user to press a softkey
6440 Hv_set:  !
6450    OFF KEY 5                 ! Turn off hv key
6460    Meas_flag(2,2)=1          ! Set hv flag
6470    GOTO Spin                 ! Wait for user to press a softkey
6480 Vh_set:  !
6490    OFF KEY 8                 ! Turn off vh key
6500    Meas_flag(2,3)=1          ! Set vh flag
6510    GOTO Spin                 ! Wait for user to press a softkey
6520 Hh_set:  !
6530    OFF KEY 9                 ! Turn off hh key
6540    Meas_flag(2,4)=1          ! Set hh flag
6550    GOTO Spin                 ! Wait for user to press a softkey
6560 Can:!  (cancels selections)
6570    OFF KEY                   ! Turn off softkeys
6580    MAT Meas_flag= Meas_flag_old    ! Restore old set of freqs & pols
6590    GOTO Menu                       ! Go back to main menu
6600 Store:! (saves new set of selections)
6610    OFF KEY                   ! Turn off softkeys
6620 !  Combinations=Meas_flag(1,1)+Meas_flag(1,2)+Meas_flag(1,3)+Meas_flag(1,4)
6630    Combinations=Meas_flag(1,3)+Meas_flag(1,4)
6640  !  Combinations=Combinations*(Meas_flag(2,1)+Meas_flag(2,2)+Meas_flag(2,3)+Meas_fl
ag(2,4))
6650     Combinations=Combinations*(Meas_flag(2,1)+Meas_flag(2,2)+Meas_flag(2,3)+Meas_fl
ag(2,4))+Meas_flag(1,1)+Meas_flag(1,2)
6660    !  Combinations = # freqs x # pols
6670    GOTO Menu                 ! Go back to main menu
6680  !---------------------------------------------------------------------------
6690 Display: !
6700    !
6710    ! The DISPLAY softkey is the preferred method of switching frequency
6720    ! and polarization when not running a measurement (if you just want to
6730    ! look at, but not measure, 94vh, for example).  Using the DISPLAY key,
6740    ! the computer will prompt you to switch the radar freq. & polarization
6750    ! and will send commands to the HP8510 to use the correct parameter
6760    ! and electrical delay.  The selected frequency & pol will become the
6770    ! default f & p, and the HP8510 will return to the default after making
6780    ! any measurements, if such measurements are ordered.  The initial
6790    ! defaults are 35 GHz, vv-pol.
6800    !
6810    OFF KEY    ! Deactivate previous softkey definitions
6820    DISP "Select frequency/polarization to view on HP8510"
6830    ON KEY 0 LABEL "35 GHz....." GOTO Show_35
6840    ON KEY 1 LABEL "94 GHz....." GOTO Show_94
6850    ON KEY 2 LABEL "140 GHz...." GOTO Show_140
6860    ON KEY 3 LABEL "215 GHz...." GOTO Show_215
6870    ON KEY 4 LABEL "VV........." GOTO Show_vv
6880    ON KEY 5 LABEL "HV (rt)...." GOTO Show_hv
6890    ON KEY 8 LABEL "VH (rt)...." GOTO Show_vh
6900    ON KEY 9 LABEL "HH........." GOTO Show_hh
6910    GOTO Spin
6920 Show_35:  !
6930    Fdef=1                         ! Change default freq to 35 GHz
6940   CALL Freq_set(Fdef,1)         ! Prompt user to switch radar to 35 GHz
6950!   CALL Coh_freq(Fdef)
6960    CALL Polariz_set(Pdef,Fdef)   ! Prompt user to check pol. switches
6970    GOTO Change_cal               ! Skip to bottom of routine
6980 Show_94:  !
6990    Fdef=2                        ! Change default freq to 94 GHz
7000   CALL Freq_set(Fdef,1)         ! Prompt user to switch radar to 94 GHz
7010!   CALL Coh_freq(Fdef)
7020    CALL Polariz_set(Pdef,Fdef)   ! Prompt user to check pol. switches
7030    GOTO Change_cal               ! Skip to bottom of routine
7040 Show_140:  !
7050    Fdef=3                        ! Change default freq to 140 GHz
7060    CALL Freq_set(Fdef,1)         ! Prompt user to switch radar to 140 GHz
7070    CALL Polariz_set(Pdef,Fdef)   ! Prompt user to check pol. switches
7080    GOTO Change_cal               ! Skip to bottom of routine
7090 Show_215:!
7100    Fdef=4                        ! Change defualt freq to 215 GHz
7110    CALL Freq_set(Fdef,1)         ! Prompt user to switch radar to 215 GHz
7120 !  CALL Coh_freq(Fdef)
7130    CALL Polariz_set(Pdef,Fdef)   ! Prompt user to check pol. switches
7140    GOTO Change_cal               ! Skip to bottom of routine
7150 Show_vv:  !
7160    Pdef=1                        ! Change default pol. to vv
7170    CALL Polariz_set(Pdef,Fdef)   ! Prompt user to switch radar to vv
7180    GOTO Change_cal               ! Skip to bottom of routine
7190 Show_hv:  !
7200    Pdef=2                        ! Change default pol. to hv
7210    CALL Polariz_set(Pdef,Fdef)   ! Prompt user to switch radar to hv
7220    GOTO Change_cal               ! Skip to bottom of routine
7230 Show_vh:  !
7240    Pdef=3                        ! Change default pol. to vh
7250    CALL Polariz_set(Pdef,Fdef)   ! Prompt user to switch radar to vh
7260    GOTO Change_cal               ! Skip to bottom of routine
7270 Show_hh:  !
7280    Pdef=4                        ! Change default pol. to hh
7290    CALL Polariz_set(Pdef,Fdef)   ! Prompt user to switch radar to hh
7300 Change_cal:  !
7310    OUTPUT @Nwa;"TITL ""       FREQUENCY: "&Freq$(Fdef)&" GHz, POLARIZATION: "&Pol$(P
def)&""""    ! Update HP8510 display title
7320    OUTPUT @Nwa;"ELED "&VAL$(Zero(Fdef,Pdef))&" S;ENTO"  ! Set elect. delay
7330    DISP ""      ! Clear prompt line on computer screen
7340    GOTO Menu    ! Return to main menu
7350  !---------------------------------------------------------------------
7360 Angle:  !   Find radar angle or change it
7370    OFF KEY  ! Deactivate previous softkey definitions
7380    ! Press key 0 to set desired measurement angle
7390    ON KEY 0 LABEL "SET ANGLE  " GOTO Set_angle
7400    ! Press key 1 to determine current angle from inclinometer resistance
7410    ON KEY 1 LABEL "FIND ANGLE  " GOTO Find_angle
7420    ! Press key 9 to return to main menu
7430    ON KEY 9 LABEL "MAIN MENU  " GOTO Menu
7440    GOTO Spin    ! Wait for user to press softkey
7450 Set_angle:   !
7460    OFF KEY
7470    INPUT "Enter measurement angle",Angle
7480    Angle$=VAL$(Angle)&CHR$(179)&"   "   ! degree sign is appended to #
7490    GOTO Menu
7500 Find_angle:  !
7510    OFF KEY
7520    INPUT "Enter resistance value (in kilohms)",Resist
7530    Angle_now=166.91813-90.03136*Resist
7540    DISP "Resistance = ";VAL$(Resist);" K       Current angle = ";VAL$(PROUND(Angle_n
ow,-2));CHR$(179)         ! Display current angle & resistance
7550    GOTO Menu
7560  !---------------------------------------------------------------------
7570 Target:   !
```

```
7580     !
7590     ! The user is prompted to enter the target, which will be listed on
7600     ! the printout and in the disc file.  The target name cannot exceed
7610     ! 30 characters.
7620     !
7630     OFF KEY
7640     LINPUT "Enter target type",Target$
7650     Target$=TRIM$(Target$)     ! Trim leading & trailing blanks
7660     Target$=Target$&RPT$(" ",30-LEN(Target$))  ! Pad string with blanks
7670     GOTO Menu
7680  !----------------------------------------------------------------------
7690 Comment_print:   !
7700     !
7710     ! Pressing COMMENTS calls the Comments subroutine, which allows the
7720     ! user to print comments on the printout.
7730     !
7740     OFF KEY
7750     CALL Comments
7760     GOTO Menu
7770  !----------------------------------------------------------------------
7780 Quit:  ! Quit the program
7790     OFF KEY
7800     CLEAR SCREEN
7810     End_flag=1        ! Set flag which signals the end of the program
7820     GOTO Sub_end      ! Skip to the end of the subroutine
7830  !----------------------------------------------------------------------
7840 Parameters:!
7850     !
7860     ! Pressing PARAMETERS switches from the main menu to the parameters
7870     ! menu.  In this menu, the operator can select the numbers of traces
7880     ! per set, the number of points per trace, the output type, the debug
7890     ! mode, and the electrical delays.
7900     !
7910     OFF KEY      ! Deactivate previous softkey definitions
7920     CLEAR SCREEN
7930 No_clear: !     ! Return to this point to redraw menu without clear screen
7940     OUTPUT KBD;CHR$(255)&CHR$(84);  ! Home display
7950     PRINT TABXY(1,1)                 ! Begin printing at top left corner
7960     ! The following lines print the parameters menu
7970     PRINT
7980     PRINT "           PARAMETER                      CURRENT VALUE"
7990     PRINT
8000     PRINT "           # of traces/set                     ";Ntrace;"    "
8010     PRINT "           # of points                         ";Npts;"    "
8020     PRINT "           Volume bin size (meters)            ";Bin_rng(1);Bin_rng(2);Bin
_rng(3);"     "
8030     PRINT "           Output type                          ";Out_type$
8040     PRINT "           Debug mode                           ";Debug$;" "
8050     PRINT "           Electrical delay of 35 GHz: VV,HV   ";Zero(1,1);Zero(1,2)
8060     PRINT "           Electrical delay of 35 GHz: VH,HH   ";Zero(1,3);Zero(1,4)
8070     PRINT "           Electrical delay of 94 GHz: VV,HV   ";Zero(2,1);Zero(2,2)
8080     PRINT "           Electrical delay of 94 GHz: VH,HH   ";Zero(2,3);Zero(2,4)
8090     PRINT "           Electrical delay of 140 GHz: VV,HV  ";Zero(3,1);Zero(3,2)
8100     PRINT "           Electrical delay of 140 GHz: VH,HH  ";Zero(3,3);Zero(3,4)
8110     PRINT "           Electrical delay of 215 GHz: VV,HV  ";Zero(4,1);Zero(4,2)
8120     PRINT "           Electrical delay of 215 GHz: VH,HH  ";Zero(4,3);Zero(4,4)
8130     ! Press key 0 to select number of traces per set
8140     ON KEY 0 LABEL " # OF TRACES  " GOTO Trace
8150     ! Press key 1 to select number of points per trace
8160     ON KEY 1 LABEL " # OF POINTS  " GOTO Points
8170     ! Press key 2 to set bin sizes for volume measurements
8180     !ON KEY 2 LABEL "   BIN SIZE   " GOTO Bin_rng_set
8190     ! Press key 3 to select output type
8200     ON KEY 3 LABEL " OUTPUT TYPE  " GOTO Output_set
8210     ! Press key 4 to toggle debug mode
8220     ON KEY 4 LABEL "  DEBUG MODE  " GOTO Debug_set
8230     ! Press key 5 to set the electrical delays
8240     ON KEY 5 LABEL " ELEC. DELAY  " GOTO Delay
8250     ! Press key 9 to return to the main menu
8260     ON KEY 9 LABEL "  MAIN MENU   " GOTO Menu_clear
8270     GOTO Spin     ! Wait for user to press softkey
8280 Trace:            ! Select number of traces per set
8290     OFF KEY
8300     INPUT "Input the # of traces desired (integer)",Ntrace
8310     GOTO No_clear  ! Return to parameters menu
8320 Points:           ! Select number of points per trace (51,101,201,401)
8330     OFF KEY
8340     ON KEY 0 LABEL "51 POINTS   " GOTO Points_51
8350     ON KEY 1 LABEL "101 POINTS   " GOTO Points_101
8360     ON KEY 2 LABEL "201 POINTS   " GOTO Points_201
8370     ON KEY 3 LABEL "401 POINTS   " GOTO Points_401
8380     GOTO Spin
8390 Points_51:   !
8400     Npts=51
8410     OUTPUT @Nwa;"POIN51"     ! Set HP8510 to 51 points/trace
8420     GOTO No_clear  ! Return to parameters menu
8430 Points_101:  !
8440     Npts=101
8450     OUTPUT @Nwa;"POIN101"
8460     GOTO No_clear  ! Return to parameters menu
8470 Points_201:  !
8480     Npts=201
8490     OUTPUT @Nwa;"POIN201"
8500     GOTO No_clear  ! Return to parameters menu
8510 Points_401:  !
8520     Npts=401
8530     OUTPUT @Nwa;"POIN401"
8540     GOTO No_clear  ! Return to parameters menu
8550  !
8560  !
8570 Debug_set:     ! Toggle HP8510 debug mode
8580     OFF KEY
8590     IF Debug$[1,2]="ON" THEN
8600       Debug$="OFF"
8610       OUTPUT @Nwa;"DEBUOFF"
8620     ELSE
8630       Debug$="ON"
8640       OUTPUT @Nwa;"DEBUON"
8650     END IF
8660     GOTO No_clear  ! Return to parameters menu
8670 Output_set:   !      Output to printer, disc, or both.
8680     OFF KEY
8690     ON KEY 0 LABEL "PRINT...." GOTO Out_pr
8700     ON KEY 1 LABEL "DISC....." GOTO Out_d
8710     ON KEY 2 LABEL "PRINT/DISC  " GOTO Out_pr_d
8720     GOTO Spin
8730 Out_pr:   !
8740     OFF KEY
8750     Out_type$="PRINT     "
8760     GOTO No_clear  ! Return to parameters menu
8770 Out_d:    !
8780     OFF KEY
8790     Out_type$="DISC      "
8800     GOTO No_clear  ! Return to parameters menu
8810 Out_pr_d:   !
8820     OFF KEY
8830     Out_type$="PRINT/DISC"
8840     GOTO No_clear  ! Return to parameters menu
```

```
8850 Delay:      ! Set all 12 electrical delays
8860    INPUT "Electrical delay for 35 GHz, VV, in seconds (two-way)",Zero(1,1)
8870    INPUT "Electrical delay for 35 GHz, HV, in seconds (two-way)",Zero(1,2)
8880    INPUT "Electrical delay for 35 GHz, VH, in seconds (two-way)",Zero(1,3)
8890    INPUT "Electrical delay for 35 GHz, HH, in seconds (two-way)",Zero(1,4)
8900    INPUT "Electrical delay for 94 GHz, VV, in seconds (two-way)",Zero(2,1)
8910    INPUT "Electrical delay for 94 GHz, HV, in seconds (two-way)",Zero(2,2)
8920    INPUT "Electrical delay for 94 GHz, VH, in seconds (two-way)",Zero(2,3)
8930    INPUT "Electrical delay for 94 GHz, HH, in seconds (two-way)",Zero(2,4)
8940    INPUT "Electrical delay for 140 GHz, VV, in seconds (two-way)",Zero(3,1)
8950    INPUT "Electrical delay for 140 GHz, HV, in seconds (two-way)",Zero(3,2)
8960    INPUT "Electrical delay for 140 GHz, VH, in seconds (two-way)",Zero(3,3)
8970    INPUT "Electrical delay for 140 GHz, HH, in seconds (two-way)",Zero(3,4)
8980    INPUT "Electrical delay for 215 GHz, VV, in seconds (two-way)",Zero(4,1)
8990    INPUT "Electrical delay for 215 GHz, HV, in seconds (two-way)",Zero(4,2)
9000    INPUT "Electrical delay for 215 GHz, VH, in seconds (two-way)",Zero(4,3)
9010    INPUT "Electrical delay for 215 GHz, HH, in seconds (two-way)",Zero(4,4)
9020    PRINTER IS 26    ! Direct output to printer
9030    PRINT RPT$("*",70)  ! Print line of "*", then all elect. delays
9040    PRINT "Electrical delay of 35 GHz in seconds (two-way)";Zero(1,1);Zero(1,2);Zero
(1,3);Zero(1,4)
9050    PRINT "Electrical delay of 94 GHz in seconds (two-way)";Zero(2,1);Zero(2,2);Zero
(2,3);Zero(2,4)
9060    PRINT "Electrical delay of 140 GHz in seconds (two-way)";Zero(3,1);Zero(3,2);Zer
o(3,3);Zero(3,4)
9070    PRINT "Electrical delay of 215 GHz in seconds (two-way)";Zero(4,1);Zero(4,2);Zer
o(4,3);Zero(4,4)
9080    PRINT RPT$("*",70)  ! Print line of "*"
9090    PRINTER IS CRT      ! Direct output to screen
9100    OUTPUT @Nwa;"ELED "&VAL$(Zero(Fdef,Pdef))&" S;ENTO"  ! Set elect. delay of curre
nt f & p
9110    GOTO Parameters  ! Return to parameters menu
9120  !------------------------------------------------------------------------
9130  !
9140 Begin: !  Begin measurement sequence (exit menu subroutine)
9150    OFF KEY                         ! Deactivate softkeys
9160    CLEAR SCREEN
9170    PRINT "ENTER 'Y' IF YOU WANT COHERENT-ON-RECEIVE MEASUREMENTS"
9180    INPUT Coh$
9190    CLEAR SCREEN
9200    IF Coh$="Y" OR Coh$="y" THEN
9210      CALL Coh_on_recv
9220    END IF
9230  !-----------------------------------------------------------
9240  !
9250 Sub_end:  !
9260    DEALLOCATE Meas_flag_old(*)       ! Make memory used for these arrays
9270    DEALLOCATE Cal_flag_old(*)        ! available for general use.
9280  SUBEND                              ! Return to main program
9290  !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
9300  SUB File_srch(Filename$,Drive$,INTEGER Flag)
9310    !
9320    ! This subroutine gets a list of all files on a disc and
9330    ! checks to see if a given filename already exists.  If it
9340    ! does, then Flag is set.
9350    !
9360    !*********************************************************************
9370    !                 VARIABLES USED  (* denotes subroutine argument)      *
9380    !*********************************************************************
9390    !*Drive$                 Char  Specifies disc drive to search
9400    ! Dum$(20)[80]           Char  Dummy workspace for CAT
9410    !*Filename$              Char  Name of file to seek
9420    !*Flag                   Intg  Number of files found
9430    !*********************************************************************
9440    DIM Dum$(20)[80]
9450    CAT Drive$ TO Dum$(*);SELECT Filename$,NO HEADER,COUNT Flag
9460    IF Flag>1 THEN
9470      ! ...there are two files with name Filename$ on this disc, which
9480      ! should be impossible.
9490      PRINT "AAACK! THHHPT!  DUPLICATE FILES!"
9500      PAUSE  ! Wait for user to decide how to get out of this mess.
9510    END IF
9520  SUBEND
9530  !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
9540  DEF FNDb(Number)
9550    ! This functions converts a number to "decibels".
9560    Db=10*LGT(Number)
9570    RETURN Db
9580  FNEND
9590  !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
9600  SUB Fix_error
9610    !
9620    ! This is the error correction (hopefully) routine.  Certain errors,
9630    ! such as a disc full error, can be fixed without stopping the program.
9640    ! We have tried to anticipate such errors and include instructions
9650    ! to the user to make recovery possible.  Fatal errors are listed on
9660    ! the computer screen.
9670    !
9680    !*********************************************************************
9690    !                 VARIABLES USED  (* denotes subroutine argument)      *
9700    !*********************************************************************
9710    ! Err_file$[80]        Char  Name of current file is stored here to be
9720    !                            used in error messages in Fix_error routine
9730    !*********************************************************************
9740    !
9750    COM /Errors/ Msi$,Err_file$
9760    SELECT ERRN    ! Select from the following remedies, according to error #
9770    CASE 56              ! File not found error
9780      PRINTER IS CRT ! Direct output to screen
9790      PRINT "File "&Err_file$&" has not been found in the directory: ";Msi$
9800      PRINT "Please check the disc, then <CONTINUE>."
9810      PAUSE              ! Wait for user to press <CONTINUE>
9820    CASE 64              ! Disc full error
9830      PRINTER IS CRT ! Direct output to screen
9840      PRINT "No room for file "&Err_file$&" on disc."
9850      PRINT "Please insert another disc, then <CONTINUE>."
9860      PAUSE              ! Wait for user to press <CONTINUE>
9870    CASE 80              ! Disc changed or not in drive error
9880      PRINTER IS CRT ! Direct output to screen
9890      PRINT "File "&Err_file$&" has not been found."
9900      PRINT "Disc changed or not in drive."
9910      PRINT "Please insert the disc, then <CONTINUE>."
9920      PAUSE              ! Wait for user to press <CONTINUE>
9930    CASE 83              ! Write-protected disc error
9940      PRINTER IS CRT ! Direct output to screen
9950      PRINT "Cannot write file "&Err_file$&"."
9960      PRINT "Disc is write-protected."
9970      PRINT "Disable write-protection tab, then <CONTINUE>."
9980      PAUSE              ! Wait for user to press <CONTINUE>
9990    CASE 85              ! Disc not initialized error
10000     PRINTER IS CRT ! Direct output to screen
10010     PRINT "Cannot write file "&Err_file$&"."
10020     PRINT "Disc has not been initialized."
10030     PRINT "Initialize disc, then <CONTINUE>."
10040     PAUSE              ! Wait for user to press <CONTINUE>
10050   CASE ELSE            ! All other errors
10060     PRINTER IS CRT ! Direct output to screen
10070     ! Print error code on screen
```

```
10080     PRINT "ERROR ";ERRN
10090     ! Print error description on screen
10100     PRINT ERRM$
10110     ! The program will try to continue if the user presses <CONTINUE>.
10120     ! If the error has not been fixed, the error messge will recur.
10130     PAUSE
10140   END SELECT
10150   CLEAR ERROR
10160   CLEAR SCREEN
10170 SUBEND
10180 !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
10190 DEF FNPathname$(Filein$,Dirin$)
10200   OPTION BASE 1
10210   INTEGER Char_loc
10220   DIM Here$[80],Vol$[20],File$[14],Dir$[80],Heredir$[80],Herevol$[80]
10230   LET Dir$=TRIM$(Dirin$)
10240   LET File$=TRIM$(Filein$)
10250   LOOP                                !
10260     LET Char_loc=POS(Dir$,"\")        ! Fix stupid DOS directory divider
10270   EXIT IF NOT Char_loc                !
10280     LET Dir$[Char_loc,Char_loc]="/"   !
10290   END LOOP                            !
10300   !
10310   ! set volume, and raw dir$
10320   LET Char_loc=POS(Dir$,":")
10330   IF Char_loc THEN
10340     LET Vol$=Dir$[Char_loc]
10350     LET Dir$=Dir$[1,Char_loc-1]
10360   ELSE
10370     LET Here$=SYSTEM$("MSI")
10380     LOOP                                 !
10390       LET Char_loc=POS(Here$,"\")        ! Fix DOS directory divider
10400     EXIT IF NOT Char_loc                 !
10410       LET Here$[Char_loc,Char_loc]="/"   !
10420     END LOOP                             !
10430     LET Herevol$=Here$[POS(Here$,":")]
10440     LET Heredir$=Here$[1,POS(Here$,":")-1]
10450     LET Vol$=Herevol$
10460   END IF
10470   !
10480   ! filter Dir$ to standard form
10490   IF LEN(Dir$) THEN
10500     IF Dir$[LEN(Dir$)]<>"/" THEN LET Dir$=Dir$&"/" ! Append divider
10510     IF Dir$[1,2]=".." THEN                          !
10520       ALLOCATE Left$[LEN(Heredir$)]                 ! for ../somedir
10530       LET Left$=REV$(Heredir$)                      ! find full
10540       LET Dir$=REV$(Left$[POS(Left$,"/")])&"/"&Dir$ ! pathname
10550       DEALLOCATE Left$                              !
10560     END IF                                          !
10570     IF Dir$[1,1]="." THEN LET Dir$=Heredir$&"/"&Dir$ ! for ./somedir ...
10580     IF Dir$[1,1]<>"/" THEN                          !
10590       LET Dir$="/"&Dir$                             ! start at root,
10600       IF Herevol$=Vol$ THEN LET Dir$=Heredir$&Dir$ ! whatever root beedit 10250
10610     END IF                                          !
10620     LOOP                                    !
10630       LET Char_loc=POS(Dir$,"/../")         ! Filter out internal
10640     EXIT IF NOT Char_loc                    ! references to
10650       ALLOCATE Left$[Char_loc]              ! parent directory
10660       LET Left$=REV$(Dir$[1,Char_loc-1])    !
10670       LET Dir$=REV$(Left$[POS(Left$,"/")])&Dir$[Char_loc+3]
10680       DEALLOCATE Left$                      !
10690     END LOOP                                !
10700     LOOP                                     !
10710       LET Char_loc=POS(Dir$,"/./")           ! Filter out internal
10720     EXIT IF NOT Char_loc                     ! references to
10730       LET Dir$[Char_loc]=Dir$[Char_loc+2]    ! current directory
10740     END LOOP                                 !
10750     LOOP                                     !
10760       LET Char_loc=POS(Dir$,"//")            ! Filter out double
10770     EXIT IF NOT Char_loc                     ! directory dividers
10780       LET Dir$[Char_loc]=Dir$[Char_loc+1]    !
10790     END LOOP                                 !
10800   END IF
10810   !
10820   ! Filter Filename
10830   IF POS(Vol$,":DOS") THEN
10840     LET Dir$=UPC$(Dir$)           !
10850     LET File$=UPC$(File$)         ! DOS recognizes only upper case
10860     IF LEN(File$)>8 AND NOT POS(File$,".") THEN ! if filename is too
10870       LET File$=File$[1,8]&"."&File$[9]         ! long and no extension,
10880     END IF                                      ! insert a period
10890   ELSE ! if not a DOS directory...
10900     LET Char_loc=POS(File$,".") ! no periods allowed in filename
10910     IF Char_loc THEN LET File$=File$[1,Char_loc-1]&File$[Char_loc+1]
10920   END IF
10930   RETURN Dir$&File$&Vol$
10940 FNEND ! Pathname$
10950 !
10960 !----------------------------------------------------------------------
10970 !
10980 DEF FNViper
10990   ASSIGN @Dos TO 19;RETURN Dosgone
11000   IF NOT Dosgone THEN ASSIGN @Dos TO *
11010   RETURN NOT Dosgone
11020 FNEND
11030 !
11040 !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
11050 !
11060  SUB Handshake
11070  !
11080  !++++++++++++++++++++++++++++++++++++++++++++++++++++++++
11090    !This is the routine which establishes communication
11100    !with the polarization computer. It also initializes the
11110    !serial port.
11120    LET Ser_port=9
11130    CONTROL Ser_port,0;1              !Resets the RS-232-C interface
11140    CONTROL Ser_port,1;0              !Disables RS232 interrupt capability
11150    CONTROL Ser_port,3;9600
11160    CONTROL Ser_port,4;3              !Parity off,1 stop bit,8 bits data
11170    CONTROL Ser_port,5;1              !Set DTR line active
11180    CONTROL Ser_port,6;46             !Place handshake value in REG.
11190                                   !i.e. REG. is the serial port in
11200    STATUS Ser_port,6;Check           !Look at character in REG.
11210    DISP Check,"Waiting for confirmation from polarization control..."
11220    IF Check<>46 THEN 11200      !Wait until handshake completed
11230    CONTROL Ser_port,6;46
11240    CLEAR SCREEN                 !Delete the DISP prompt
11250    !Serial communication is now established
11260  SUBEND
11270  !
11280  !++++++++++++++++++++++++++++++++++++++++++++++++++++++++
11290  !
11300      SUB Coh_on_recv
11310  !
11320  !
11330  COM /Freq_data/ Freq$(*),Freq_cent(*),Freq_span(*)
11340  COM /Paths/ @Nwa,@Nwa_data,Hp8510,Hp8753
11350  COM /Pol_data/ Pol$(*),Numer$(*)
```

```
11360  COM /Trans_pol/ Tpol$(*)
11370  COM /Incoh_eled/ Polz(*)
11380  COM /Constants/ Vel,Zero(*),Far_field(*)
11390  COM /Labels/ Target$,Version$,Mode$,Out_type$,Debug$
11400  COM /Beam_data/ Angle,Angle$,Bin_rng(*),Beam(*)
11410  COM /Flags/ INTEGER Cal_flag(*),Meas_flag(*),Combinations,Fdef,Pdef
11420  COM /Errors/ Msi$,Err_file$
11430  COM /N_data/ INTEGER Npts,Ntrace
11440 COM /Gate_data/ Gate_cent,Gate_span,INTEGER Average_factor,Igate
11450  COM /Com4/ INTEGER Rotation_state,REAL Inc_angle,Current_angle,Start_angle,Stop_a
ngle,Old_home_angle,INTEGER Sets_per_pos
11460  COM /Status/ INTEGER Sc,Connect_flg,E_flg,Debug_flg,Response$[80]
11470  !
11480  COM /Store_path/ @Coh1,@Coh2,@Coh4
11490  COM /Trace_no/ INTEGER In_samples,In_trace
11500  INTEGER I,Irotate
11510  !
11520  PRINT "ENTER THE NUMBER OF SAMPLES TO BE TAKEN"
11530  INPUT In_samples
11540  CLEAR SCREEN
11550  !
11560  ALLOCATE COMPLEX Pol_store(1:4,1:11),Target_data(In_samples,4,11)
11570 !   PRINT "ENTER THE NUMBER OF 'TRACES' /SAMPLE/POLARIZATION TO BE TAKEN"
11580 !  PUT In_trace
11590  In_trace=1
11600  !
11610  IF Hp8510 THEN OUTPUT @Nwa;"FREQ;"
11620  IF Hp8753 THEN OUTPUT @Nwa;"TIMDTRANOFF;"
11630  !
11640  OUTPUT @Nwa;"COUCON;CHOPAB;DUACON;"
11650  !
11660  CLEAR SCREEN
11670  !
11680  OUTPUT @Nwa;"CHAN1; TIMDTRANON; LOGM;"
11690  OUTPUT @Nwa;"CHAN2; TIMDTRANON; LOGM;"
11700  BEEP
11710  PRINT TABXY(1,10);"Press CONTINUE when ready ..... "
11720  PAUSE
11730  IF FNAsk("Set gates?") THEN
11740     Igate=1
11750     Set_gates
11760  ELSE
11770     Igate=0
11780  END IF
11790  OUTPUT @Nwa;"CHAN1;TIMDTRANOFF;POLA;GATEOFF;" !
11800  OUTPUT @Nwa;"CHAN2;TIMDTRANOFF;POLA;GATEOFF;" !
11810  IF FNAsk("Rotate target?") THEN
11820    Irotate=1
11830  ELSE
11840    Irotate=0
11850  END IF
11860 !
11870  FOR Ii=1 TO In_samples
11880    PRINT "THE REMAINING SAMPLE NUMBER IS   ",In_samples-Ii
11890 !
11900  IF Irotate=1 THEN
11910    IF Ii=1 THEN
11920      Rotation_state=-1
11930    ELSE
11940      Rotation_state=2
11950    END IF
11960    PRINT TABXY(1,4);"Current angle is";Current_angle;"  degrees."
11970    Rotate_target  !
11980  ELSE
11990  END IF
12000    Clear_crt(3,16)
12010    PRINT TABXY(1,1);"Collecting data ...."
12020    FOR I=1 TO 4
12030       IF Meas_flag(1,I)=0 THEN Next_freq2
12040       IF I=1 OR I=2 OR I=4 THEN
12050         IF Ii=1 OR F_counter>1 THEN
12060           CALL Coh_freq(I)
12070         END IF
12080         IF Igate=1 THEN
12090           OUTPUT @Nwa;"CHAN1;GATECENT";VAL$(Gate_cent);"S;"
12100           OUTPUT @Nwa;"CHAN2;GATECENT";VAL$(Gate_cent);"S;"
12110           OUTPUT @Nwa;"CHAN1;GATESPAN";VAL$(Gate_span);"S;"
12120           OUTPUT @Nwa;"CHAN2;GATESPAN";VAL$(Gate_span);"S;"
12130           OUTPUT @Nwa;"CHAN1;GATEON; CHAN2;GATEON;"
12140         ELSE
12150         END IF
12160         CALL Data_aquiz(I,Pol_store(*))
12170       END IF
12180 Next_freq2:    !
12190    NEXT I
12200       FOR Np=1 TO 4
12210         FOR Nd=1 TO 11
12220           Target_data(Ii,Np,Nd)=Pol_store(Np,Nd)
12230         NEXT Nd
12240       NEXT Np
12250  NEXT Ii
12260  !
12270  Store_file(Target_data(*))
12280  !
12290  DEALLOCATE Target_data(*),Pol_store(*)
12300  OUTPUT @Nwa;"COUCOFF;ALTAB;DUACOFF;"
12310  OUTPUT @Nwa;"OPC?;CHAN1"
12320  ENTER @Nwa;Dumm_no
12330  !
12340! PRINT "END OF SAMPLES"
12350  !
12360  IF Meas_flag(1,1)=1 THEN ASSIGN @Coh1 TO *
12370  IF Meas_flag(1,2)=1 THEN ASSIGN @Coh2 TO *
12380  IF Meas_flag(1,4)=1 THEN ASSIGN @Coh4 TO *
12390  PRINT "END OF MEASUREMENTS"
12400  !
12410  OUTPUT @Nwa;"TIMDTRANON;LOGM;CONT;GATEOFF;"
12420  !
12430  !
12440  SUBEND
12450  !
12460  ! ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
12470  !
12480 SUB Date_string(Brief_date$,Realtime)
12490   !
12500   !  This subroutine converts the timedate as produced by the computer
12510   !  to a 10-digit number YYMMDDhhmm used for various purposes.  Realtime
12520   !  is the time at which a measurement was started, expressed in HP
12530   !  format, which is seconds since midnight, Mar. 1, 1900, or something
12540   !  like that.
12550   !
12560   !*******************************************************************
12570   !                   VARIABLES USED  (* denotes subroutine argument)      *
12580   !*******************************************************************
12590   !*Brief_date$            Char  10-character string of form YYMMDDhhmm
12600   ! Day$                   Char  Day in character format ("01" ... "31")
12610   ! Month                  Intg  Month (1=January, etc.)
12620   ! Month$                 Char  Month in character format ("01"=Jan, etc.)
```

```
12630   ! Realdate$             Char  Date in format "DD MMM YYYY"
12640   !*Realtime              Real  Time/date in HP format
12650   ! Realtime$             Char  Time in format "15:37"
12660   !*****************************************************************
12670   INTEGER Month
12680   Realdate$=DATE$(Realtime)      ! Realdate$ looks like "25 Dec 1989"
12690   Brief_date$=Realdate$[10,11]   ! Peel of the YY from the last two digits of the
year
12700   ! Select the proper case according to the month portion of Realdate$
12710   SELECT Realdate$[4,6]
12720   CASE "Jan"
12730     Month=1
12740   CASE "Feb"
12750     Month=2
12760   CASE "Mar"
12770     Month=3
12780   CASE "Apr"
12790     Month=4
12800   CASE "May"
12810     Month=5
12820   CASE "Jun"
12830     Month=6
12840   CASE "Jul"
12850     Month=7
12860   CASE "Aug"
12870     Month=8
12880   CASE "Sep"
12890     Month=9
12900   CASE "Oct"
12910     Month=10
12920   CASE "Nov"
12930     Month=11
12940   CASE "Dec"
12950     Month=12
12960   END SELECT
12970   ! Convert Month (integer) to character string Month$
12980   Month$=VAL$(Month)
12990   ! Pad Month$ with a leading zero if necessary to make length=2 --> MM
13000   IF LEN(Month$)=1 THEN Month$="0"&Month$
13010   ! Peel off DD portion of Realdate$ and make into minimum length string
13020   Day$=VAL$(VAL(Realdate$[1,2]))
13030   ! Pad Day$ with a leading zero if necessary to make length=2 --> DD
13040   IF LEN(Day$)=1 THEN Day$="0"&Day$
13050   ! Now Brief_date$ has format YYMMDD
13060   Brief_date$=Brief_date$&Month$&Day$
13070   ! Realtime$ looks like "15:37"
13080   Realtime$=TIME$(Realtime)
13090   ! Add time to Brief_date$ to yield YYMMDDhhmm
13100   Brief_date$=Brief_date$&Realtime$[1,2]&Realtime$[4,5]
13110 SUBEND
13120 !
13130 !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
13140 !
13150   SUB Coh_freq(INTEGER Ifreq)
13160   !**************************************************************
13170   !
13180   COM /Paths/ @Nwa,@Nwa_data,Hp8510,Hp8753
13190   COM /Freq_data/ Freq$(*),Freq_cent(*),Freq_span(*)
13200 !  DISP "Set to transmit "&Freq$(Ifreq)&" GHz  <CONTINUE>"
13210 !  PAUSE                        ! Wait for user to hit <CONTINUE>
13220 !  DISP "Working"               ! Let user know he has hit <CONTINUE>
13230 !
13240 ! This section is for auto control of frequency
13250  IF Ifreq=1 THEN B=NUM("K")
13260  IF Ifreq=2 THEN B=NUM("W")
13270  IF Ifreq=4 THEN B=NUM("O")
13280   CONTROL 9,3;9600
13290   STATUS 9,6;Prev                  ! Save current value of REG.
13300   CONTROL 9,6;B                    ! Puts desired command in REG.
13310   STATUS 9,6;Code                  ! Receive code from FREQ ctrl
13320   IF Code=Prev THEN 13310          ! Only accept a new command
13330   IF Code<>B THEN STOP
13340   Code$=CHR$(Code)                 ! Turn value into character
13350   Prev=Code                        ! Update storage of past code
13360   CONTROL 9,6;63                   ! Place request for confirm. in REG.
13370   STATUS 9,6;Check                 ! Reads REG. for confirm. char.
13380                                    !ASCII 46 is "." and 33 is "!"
13390   IF Check<>46 AND Check<>33 AND Check<>NUM("m") THEN 13370
13400   IF Check=33 THEN DISP "Error in communication"
13410   IF Check=NUM("m") THEN DISP "Manual Pol. control restored"
13420   IF Check=33 THEN STOP
13430   ! DAN.2
13440 SUBEND
13450 !
13460 !-------------------------------------------------------------------
13470 !
13480  SUB Set_gates
13490 COM /Paths/ @Nwa,@Nwa_data,Hp8510,Hp8753
13500 COM /Gate_data/ Gate_cent,Gate_span,INTEGER Average_factor,Igate
13510         OUTPUT @Nwa;"CHAN1;GATEOFF;TIMDTRANON;LOGM;"
13520         OUTPUT @Nwa;"CHAN2;GATEOFF;TIMDTRANON;LOGM;"
13530         OUTPUT @Nwa;"CHAN1;GATECENT 150 NS;WAIT;"
13540             OUTPUT @Nwa;"CHAN1;KEY41; KEY59; KEY58; KEY59;"
13550             LOCAL @Nwa
13560             DISP "Adjust gate center to suit, and press CONTINUE."
13570             PAUSE
13580             DISP ""
13590             REMOTE @Nwa
13600             OUTPUT @Nwa;"OUTPACTI;"
13610             ENTER @Nwa;Gate_cent
13620             OUTPUT @Nwa;"ENTO;"
13630             OUTPUT @Nwa;"GATESPAN 10 NS"
13640             OUTPUT @Nwa;"CHAN1;KEY41; KEY59; KEY58; KEY4;"
13650             LOCAL @Nwa
13660             DISP "Adjust gate span to suit, and press CONTINUE."
13670             PAUSE
13680             DISP ""
13690             REMOTE @Nwa
13700             Clear_crt
13710             OUTPUT @Nwa;"OUTPACTI;"
13720             ENTER @Nwa;Gate_span
13730             OUTPUT @Nwa;"ENTO;"
13740  SUBEND
13750 !
13760 !*********************************************************************
13770 !
13780     SUB Data_aquiz(INTEGER Ifreq,COMPLEX Pol_store(*))
13790  !++++++++++++++++++++++++++++++++++++++++++++++++++++++++
13800  COM /Paths/ @Nwa,@Nwa_data,Hp8510,Hp8753
13810  COM /Pol_data/ Pol$(*),Numer$(*)
13820  COM /Trans_pol/ Tpol$(*)
13830  COM /Incoh_eled/ Polz(*)
13840  COM /Labels/ Target$,Version$,Mode$,Out_type$,Debug$
13850  COM /Flags/ INTEGER Cal_flag(*),Meas_flag(*),Combinations,Fdef,Pdef
13860  COM /N_data/ INTEGER Npts,Ntrace
13870  !
13880  COM /Store_path/ @Coh1,@Coh2,@Coh4
13890  COM /Trace_no/ INTEGER In_samples,In_trace
```

```
13900  ALLOCATE COMPLEX Bin_arry(1:Npts)
13910  ALLOCATE COMPLEX Pol_data(1:4,1:Npts)
13920  INTEGER Jj,No_pol,Preamble,Bytes,Counter,Nskip,Nc,Np,Nstore
13930  !
13940  IF Ifreq=1 OR Ifreq=2 THEN No_pol=6
13950  IF Ifreq=4 THEN No_pol=4
13960  !
13970  ! FOR THE 215 GHz RADAR: Jj=1 == VV  ,  Jj=2 == HH  ,  Jj=3 == VH
13980  !                        Jj=4 == HV
13990  !
14000   Counter=0
14010  FOR Jj=1 TO 3 STEP 2
14020     CALL Trans_pol_set(Jj)
14030        IF Hp8753 THEN OUTPUT @Nwa;"SING;OUTPIDEN;"
14040        ENTER @Nwa;Dumm_no$
14050        IF (Ifreq=1) OR (Ifreq=2) THEN
14060          FOR Jjj=1 TO 2
14070            Counter=Counter+1
14080            IF Jjj=1 THEN OUTPUT @Nwa;"OPC?;CHAN1;"
14090            IF Jjj=2 THEN OUTPUT @Nwa;"OPC?;CHAN2;"
14100            ENTER @Nwa;Dumm_no
14110            CALL Have_form3(Bin_arry(*),Preamble,Bytes)
14120            MAT Pol_data(Counter,1:Npts)= Bin_arry
14130          NEXT Jjj
14140        END IF
14150     !
14160  NEXT Jj
14170     !
14180     Nskip=(Npts-1)/10
14190     FOR Nc=1 TO Counter
14200       FOR Np=1 TO Npts STEP Nskip
14210          Nstore=INT(Np/Nskip)+1
14220          Pol_store(Nc,Nstore)=Pol_data(Nc,Np)
14230       NEXT Np
14240     NEXT Nc
14250     !
14260  !
14270  DEALLOCATE Bin_arry(*),Pol_data(*)
14280  SUBEND
14290  !
14300  !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
14310  !
14320 SUB Trans_pol_set(INTEGER Ipol)
14330  !
14340  ! This subroutine prompts the user to set transmit polarization for the
14350  ! incoherent measurements for certain systems.
14360  !
14370  ! ********************************************************************
14380  !             Variables used(* denotes subroutine arguement)
14390  ! ********************************************************************
14400  !
14410  !  @Nwa            Addr     Path used for Hp8510 I/O
14420  !  *Ipol           Intg     Transmitted polarization (1=V, 2=H, 3=45L,
14430  !                                              4=135L , 5=LHC, 6=RHC )
14440  !  Tpol$(1:6)[4]  Char     Labels for the six transmitted polarizations.
14450  !
14460  ! ********************************************************************
14470  !
14480  COM /Paths/ @Nwa,@Nwa_data,Hp8510,Hp8753
14490  COM /Trans_pol/ Tpol$(*)
14500  !
14510  ! This section is for auto control of polarization
14520  IF Ipol=1 THEN B=NUM("Z")
14530  IF Ipol=2 THEN B=NUM("F")
14540  IF Ipol=3 THEN B=NUM("N")
14550  IF Ipol=4 THEN B=NUM("L")
14560  IF Ipol=5 THEN B=NUM("O")
14570  IF Ipol=6 THEN B=NUM("R")
14580   CONTROL 9,3;9600
14590   STATUS 9,6;Prev                    ! Save current value of REG.
14600   CONTROL 9,6;B                      ! Puts desired command in REG.
14610   STATUS 9,6;Code                    ! Receive code from pol ctrl
14620   IF Code=Prev THEN 14610            ! Only accept a new command
14630   IF Code<>B THEN STOP
14640   Code$=CHR$(Code)                   ! Turn value into character
14650   Prev=Code                          ! Update storage of past code
14660   CONTROL 9,6;63                     ! Place request for confirm. in REG.
14670   STATUS 9,6;Check                   ! Reads REG. for confirm. char.
14680                                      !ASCII 46 is "." and 33 is "!"
14690   IF Check<>46 AND Check<>33 AND Check<>NUM("m") THEN 14670
14700   IF Check=33 THEN DISP "Error in communication"
14710   IF Check=NUM("m") THEN DISP "Manual Pol. control restored"
14720   IF Check=33 THEN STOP
14730  SUBEND
14740 !!
14750 !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
14760 !
14770  SUB Have_form3(COMPLEX Bin_arry(*),INTEGER Preamble,Bytes)
14780  !
14790  !+++++++++++++++++++++++++++++++++++++++++++++++++++++++
14800  COM /Paths/ @Nwa,@Nwa_data,Hp8510,Hp8753
14810  !  INTEGER PREAMBLE,BYTES
14820  Preamble=9025
14830  OUTPUT @Nwa;"FORM3;OUTPFORM;"
14840  ENTER @Nwa_data;Preamble,Bytes,Bin_arry(*)
14850  SUBEND
14860 !
14870 !
14880 !+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
14890 !
14900 !
14910 SUB Rotate_target
14920 OPTION BASE 1
14930 COM /Com4/ INTEGER Rotation_state,REAL Inc_angle,Current_angle,Start_angle,Stop_ang
le,Old_home_angle,INTEGER Sets_per_pos
14940 COM /Status/ INTEGER Sc,Connect_flg,E_flg,Debug_flg,Response$[80]
14950 INTEGER Fs_flag,Ss_flag,Speed,Imc_status,Confirm_answer
14960    !
14970    !
14980    Confirm_answer=1
14990    Imc_status=0
15000    Debug_flg=0
15010    Fs_flag=-1
15020    Ss_flag=-1
15030    Clear_crt(3,10)
15040    !
15050    !
15060    SELECT Rotation_state
15070    CASE =-1
15080 !        IF FNAsk("Do you wish to use the rotator?") THEN
15090            Connect_flg=0
15100            GOSUB Init_imc
15110            GOSUB Init_graph_pos
15120            GOSUB Manual_loop
15130            PRINT "Set Auto Mode Please....."
15140 !        ELSE
15150 !           Rotation_state=0
15160 !           GCLEAR
```

```
15170 !         GRAPHICS OFF
15180 !       END IF
15190    CASE =0
15200         SUBEXIT
15210    CASE =1
15220         GOSUB Check_position
15230         GOSUB Print_angles
15240         GOSUB Manual_loop
15250    CASE =2
15260         GOSUB Check_position
15270         GOSUB Auto
15280    CASE =3
15290         GOSUB Check_position
15300         GOSUB Manual_loop
15310         GOSUB Auto
15320    CASE =4
15330         GOSUB Check_position
15340         GOSUB Go_home
15350    CASE =5
15360         GOSUB Check_position
15370         Rotation_state=1  ! Switch to manual mode.
15380    END SELECT
15390    SUBEXIT
15400    !
15410    !
15420 Init_imc: ! Initialize the IMC unit.
15430    GOSUB Check_4_fault
15440    PRINT TABXY(1,3);"INITIALIZING IMC"
15450    Clear_crt(4,15)
15460    Comm("4WB")            ! Set warm boot (clear flags).
15470    PRINT TABXY(1,4);"WB"
15480    Comm("4EB")            ! Clear IMC buffer.
15490    PRINT TABXY(1,4);"EB"
15500    Encoder_ratio=4096  ! 32000
15510    Comm("4ER"&VAL$(Encoder_ratio))        ! Load encoder ratio.
15520    PRINT TABXY(1,4);"ER"&VAL$(Encoder_ratio)
15530    IF FNAsk("Do you wish to set home at the current position?") THEN
15540       Comm("4RS",Confirm_answer)
15550       ENTER Response$;Old_home_angle
15560       Old_home_angle=Old_home_angle/93.3
15570       Comm("4PIZ0")          ! Set IMC at 0.
15580       PRINT TABXY(1,4);"PIZ"&RPT$(" ",LEN(VAL$(Encoder_ratio)))
15590       Comm("4PIA0")          ! Set IMC at 0.
15600       PRINT TABXY(1,4);"PIA"
15610       Current_angle=0
15620    END IF
15630    Comm("4SP100")         ! Set speed to (50pps).
15640    PRINT TABXY(1,4);"SP "&RPT$(" ",LEN(VAL$(Encoder_ratio)))
15650    Comm("4AC500")          ! Set acceleration (500pps^2).
15660    PRINT TABXY(1,4);"AC "
15670    Comm("4DC500")          ! Set deceleration (500pps^2).
15680    PRINT TABXY(1,4);"DC "
15690    GOSUB Check_position
15700    Rotation_state=1
15710    Clear_crt
15720 !
15730 !
15740    PRINT TABXY(1,4);"DONE INITIALIZING IMC"
15750    PRINT TABXY(1,5);"Turntable currently in manual mode."
15760    PRINT TABXY(1,6)
15770 Print_angles:!
15780    PRINT TABXY(1,7);"Current  angle is: ";Current_angle;" degrees."
15790    PRINT TABXY(1,8);"Starting angle is: ";Start_angle;" degrees."
15800    PRINT TABXY(1,9);"Stopping angle is: ";Stop_angle;" degrees."
15810 RETURN
15820 !
15830 !
15840 Manual_loop:! Main activation loop.
15850    LOOP
15860       ON KEY 0 LABEL "FAST SLEW CW  ",FNTrap_level GOSUB Fs_cw
15870       ON KEY 1 LABEL "FAST SLEW CCW ",FNTrap_level GOSUB Fs_ccw
15880       ON KEY 5 LABEL "SLOW SLEW CW  ",FNTrap_level GOSUB Ss_cw
15890       ON KEY 6 LABEL "SLOW SLEW CCW ",FNTrap_level GOSUB Ss_ccw
15900       ON KEY 2 LABEL "MANUAL CONTROL",FNTrap_level GOSUB Manual
15910       ON KEY 3 LABEL "TARGET GO HOME",FNTrap_level GOSUB Go_home
15920       ON KEY 4 LABEL "STOP ROTATION ",FNTrap_level GOSUB Stop_turn
15930       ON KEY 7 LABEL "SET AUTO MODE ",FNTrap_level GOSUB Set_auto
15940       ON KEY 8 LABEL "SET TARGET HOME",FNTrap_level GOSUB Set_position
15950       ON KEY 9 LABEL "RETURN        ",FNTrap_level GOTO Quit
15960       GOSUB Check_position
15970    END LOOP
15980 !
15990 !----------------------------------------------------------------------
16000 !
16010 Fs_cw:! Fast slew clockwise.
16020       IF Fs_flag<0 THEN
16030          Comm("4SP500")
16040          Comm("4SFN")
16050          Fs_flag=-1*Fs_flag
16060          Clear_crt(3,15)
16070          PRINT TABXY(1,15);"ROTATING CW (FAST)"
16080       ELSE
16090          Comm("4ST")
16100          Fs_flag=-1*Fs_flag
16110          Clear_crt(3,15)
16120          PRINT TABXY(1,15);"ROTATION STOPPED"
16130          GOSUB Check_position
16140       END IF
16150 RETURN
16160 !
16170 !----------------------------------------------------------------------
16180 !
16190 Fs_ccw:! Fast slew counterclockwise.
16200       IF Fs_flag<0 THEN
16210          Comm("4ST")
16220          Comm("4SP500")
16230          Comm("4SRN")
16240          Fs_flag=-1*Fs_flag
16250          Clear_crt(3,10)
16260          PRINT TABXY(1,15);"ROTATING CCW (FAST)"
16270       ELSE
16280          Comm("4ST")
16290          Fs_flag=-1*Fs_flag
16300          Clear_crt(3,15)
16310          PRINT TABXY(1,15);"ROTATION STOPPED"
16320          GOSUB Check_position
16330       END IF
16340 RETURN
16350 !
16360 !----------------------------------------------------------------------
16370 !
16380 Ss_cw:! Slow slew clockwise.
16390       IF Ss_flag<0 THEN
16400          Comm("4ST")
16410          INPUT "Speed?",Sp
16420          Comm("4SP"&VAL$(INT(Sp)))
16430          Comm("4SFN")
16440          Ss_flag=-1*Ss_flag
```

```
16450        Clear_crt(3,15)
16460        PRINT TABXY(1,15);"ROTATING CW (SLOW)"
16470     ELSE
16480        Comm("4ST")
16490        Ss_flag=-1*Ss_flag
16500        Clear_crt(3,15)
16510        PRINT TABXY(1,15);"ROTATION STOPPED"
16520        GOSUB Check_position
16530     END IF
16540 RETURN
16550 !
16560 !----------------------------------------------------------------------
16570 !
16580 Ss_ccw:! Slow slew counterclockwise.
16590     IF Ss_flag<0 THEN
16600     INPUT "Speed?",Sp
16610        Comm("4SP"&VAL$(INT(Sp)))
16620        Comm("4SRN")
16630        Ss_flag=-1*Ss_flag
16640        Clear_crt(3,15)
16650        PRINT TABXY(1,15);"ROTATING CCW (SLOW)"
16660     ELSE
16670        Comm("4ST")
16680        Ss_flag=-1*Ss_flag
16690        Clear_crt(3,15)
16700        PRINT TABXY(1,15);"ROTATION STOPPED"
16710        GOSUB Check_position
16720     END IF
16730 RETURN
16740 !
16750 !----------------------------------------------------------------------
16760 !
16770 Manual: INPUT "ANGLE (IN DEGREES)=?",Inc_angle
16780         INPUT "SPEED? (~100-~500 RECOMMENDED)",Speed
16790         Comm("4SP"&VAL$(Speed))
16800 Auto:   SELECT Rotation_state
16810         CASE =4
16820              GOSUB Go_home
16830              Rotation_state=2
16840              GOTO Auto
16850         CASE ELSE
16860              Angl2=Inc_angle*93.3
16870              Angl1=INT(Angl2)
16880              IF Angl2-Angl1>=.5 THEN Angl1=Angl1+1
16890         !    Current_angle=Current_angle+Inc_angle
16900              Inc_angle$=VAL$(Angl1)
16910              Comm("4IM"&Inc_angle$)
16920              Comm("4RFI")
16930         END SELECT
16940         Imc_status=0
16950         Clear_crt(3,7)
16960         PRINT TABXY(1,14);"ROTATING TARGET, PLEASE WAIT."
16970         !
16980         !
16990         WHILE NOT BIT(Imc_status,0)           ! Wait for motor to stop.
17000               Comm("4RS",Confirm_answer)
17010               ENTER Response$;Imc_status
17020               PRINT TABXY(1,15);DVAL$(Imc_status,2)
17030               GOSUB Check_position
17040               WAIT 1
17050         END WHILE
17060         Imc_status=0
17070         !
17080         !
17090         Clear_crt(3,16)
17100         PRINT TABXY(1,16);"CURRENT TARGET POSITION IS ";Current_angle;" DEGREES.
"
17110         WAIT 2  ! Wait for target settling.
17120 RETURN
17130 !
17140 !----------------------------------------------------------------------
17150 !
17160 Stop_turn:Comm("4ST")
17170           WHILE NOT BIT(Imc_status,0)           ! Wait for motor to stop.
17180                 Comm("4RS",Confirm_answer)
17190                 ENTER Response$;Imc_status
17200                 WAIT .1
17210           END WHILE
17220           Clear_crt(3,16)
17230           PRINT TABXY(1,15);"ROTATION STOPPED"
17240           GOSUB Check_position
17250           Imc_status=0
17260 RETURN
17270 !
17280 !----------------------------------------------------------------------
17290 !
17300 Set_auto: Comm("4SP500")
17310           GOSUB Check_position
17320           Clear_crt(3,16)
17330           PRINT TABXY(1,3);"Current starting  angle: ";Start_angle;" degrees"
17340           PRINT TABXY(1,4);"Current increment angle: ";Inc_angle;" degrees"
17350           PRINT TABXY(1,5);"Current stopping  angle: ";Stop_angle;" degrees"
17360           PRINT TABXY(1,6);"Current rotation  speed: ";Speed
17370           PRINT TABXY(1,7);RPT$(" ",80)
17380           PRINT TABXY(1,8);"Rotator  positioned  at: ";Current_angle;" degrees"
17390           INPUT "Enter starting angle value (degrees): ",Start_angle
17400           INPUT "Enter increment angle (degrees): ",Inc_angle
17410           INPUT "Enter stopping angle (degrees): ",Stop_angle
17420           INPUT "Enter rotation speed of target (~500 recommended): ",Speed
17430           Speed=INT(Speed)
17440           Comm("4SP"&VAL$(Speed))
17450           IF ABS(Start_angle-Current_angle)>.1 THEN
17460              PRINT TABXY(1,9);RPT$(" ",80)
17470              PRINT TABXY(1,10);"Rotating target to starting angle..."
17480              Temp_angle=Inc_angle
17490              Inc_angle=Start_angle-Current_angle
17500              GOSUB Auto
17510              Inc_angle=Temp_angle
17520           END IF
17530           Rotation_state=2
17540           Clear_crt
17550           PRINT TABXY(1,20);"Turntable is in automatic mode. (press the RETURN soft
key)"
17560 RETURN
17570 !
17580 !----------------------------------------------------------------------
17590 !
17600 Set_position:INPUT "LOCK IN CURRENT TARGET POSITION AS REFERENCE POSITION?",Yn$
17610              IF Yn$="Y" OR Yn$="y" THEN
17620                 Comm("4RS",Confirm_answer)
17630                 ENTER Response$;Old_home_angle
17640                 Old_home_angle=Old_home_angle/93.3
17650                 Comm("4PIA0") ! Set absolute position to zero.
17660                 Comm("4PIZ0") ! Set incremental postition to zero.
17670                 Current_angle=0
17680              ELSE
17690                 PRINT "POSITION WAS NOT SET."
17700              END IF
```

```
17710 RETURN
17720 !
17730 !----------------------------------------------------------------------
17740 !
17750 Go_home: IF Speed<200 THEN Speed=200
17760          Comm("4SP"&VAL$(Speed))
17770          Comm("4AM0") ! Move to zero absolute position.
17780          Comm("4RAN") ! Initiate movement.
17790          Comm("4MW")  ! Make sure the move is completed.
17800          Imc_status=0
17810          Clear_crt(3,15)
17820          PRINT TABXY(1,14);"ROTATING TARGET TO HOME POSITION, PLEASE WAIT."
17830          WHILE NOT (BIT(Imc_status,0) AND BIT(Imc_status,5))
17840               GOSUB Check_status
17850               PRINT TABXY(1,15);"CURRENT STATUS: ";DVAL$(Imc_status,2)
17860               GOSUB Check_position
17870               WAIT .1
17880          END WHILE
17890          Clear_crt(3,16)
17900          PRINT TABXY(1,15);"TARGET AT HOME POSITION."
17910          GOSUB Check_position
17920          Imc_status=0
17930 RETURN
17940 !
17950 !----------------------------------------------------------------------
17960 !
17970 Check_status:! Keep an eye on the Whedco controller status.
17980         Comm("4RS",Confirm_answer)
17990         ENTER Response$;Imc_status
18000 RETURN
18010 !
18020 !----------------------------------------------------------------------
18030 !
18040 Check_position:! Get the current turnstile position in degrees.
18050         Comm("4RP",Confirm_answer)
18060         ENTER Response$;Motor_position
18070         Current_angle=Motor_position/93.3
18080    !    Current_angle=Current_angle+Inc_angle
18090         PRINT TABXY(1,16);"CURRENT TARGET POSITION IS ";Current_angle;" DEGREES.
                 "
18100         GOSUB Draw_positions
18110 RETURN
18120 !
18130 !----------------------------------------------------------------------
18140 !
18150 Check_4_fault: ! Check the IMC for a fault condition and correct or
18160                ! notify the user if necessary.
18170                !
18180      Comm("4FC",Confirm_answer)
18190      ENTER Response$;Fault$
18200      SELECT Fault$
18210      CASE ="Power failure" ! Loss of power
18220           RETURN
18230      CASE ="Force DAC" ! Force DAC command was given
18240           BEEP
18250           PRINT "Force DAC command was given..."
18260           DISP "Press CONTINUE to resume..."
18270           PAUSE
18280      RETURN
18290      CASE ="Over-current" ! Over-current condition exists.
18300           BEEP
18310           PRINT "An over-current condition has been detected on the IMC."
18320           PRINT
18330           PRINT "Cycle the power to the IMC until the OV-CUR LED goes out"
18340           DISP "Press CONTINUE to reinitialize the IMC"
18350           PAUSE
18360           GOSUB Init_imc
18370      RETURN
18380      END SELECT
18390      RETURN
18400 !
18410 !----------------------------------------------------------------------
18420 !
18430 Init_graph_pos: ! Creates a graphical depiction of where the target is.
18440                 !
18450     GINIT
18460     GCLEAR
18470     GRAPHICS ON
18480     SHOW 0,100,0,100
18490     PENUP
18500     MOVE 90,70
18510     PEN 1        ! Draw circle
18520     POLYGON 12,360,360
18530     PENUP
18540     MOVE 90,70  ! Draw old home orientation.
18550     PEN 2
18560     DRAW 90+11*COS(Old_home_angle),70-11*SIN(Old_home_angle)
18570     PENUP
18580     MOVE 90,70  ! Draw current home orientation.
18590     PEN 4
18600     DRAW 90,58
18610     PENUP
18620     MOVE 90,70  ! Draw current target orienation.
18630     PEN 3
18640     X_pos=90+11*COS(Current_angle)
18650     Y_pos=70-11*SIN(Current_angle)
18660     DRAW X_pos,Y_pos
18670 RETURN
18680 !
18690 !----------------------------------------------------------------------
18700 !
18710 Draw_positions:  ! Draws out the angular orienations.
18720     MOVE 90,70    ! Draw old home orientation.
18730     PEN 2
18740     DRAW 90-11*SIN(Old_home_angle),70-11*COS(Old_home_angle)
18750     PENUP
18760     MOVE 90,70  ! Draw current home orientation.
18770     PEN 4
18780     DRAW 90,58
18790     PENUP
18800     DISABLE
18810     MOVE 90,70  ! Draw current target orienation.
18820     PEN -3
18830     DRAW X_pos,Y_pos
18840     MOVE 90,70
18850     PEN 3
18860     X_pos=90-11*SIN(Current_angle)
18870     Y_pos=70-11*COS(Current_angle)
18880     DRAW X_pos,Y_pos
18890     PENUP
18900     ENABLE
18910 RETURN
18920 !
18930 !----------------------------------------------------------------------
18940 !
18950 Quit: !
18960       SUBEXIT
18970 SUBEND
```

```
18980 !
18990 !**********************************************************************
19000 !
19010 SUB Comm(C$,OPTIONAL INTEGER Confirm_answer)
19020 !
19030 !  PROGRAM MODULE:  Comm
19040 !
19050 !  PURPOSE:         Modified version of the Comm module to be used
19060 !                   for direct two way communication with the WHEDCO
19070 !                   IMC stepping motor controller.
19080 !
19090 !  UPDATE:  3.0     Version 3.0 checks to see if the card being used
19100 !                   is the HP98628A (Datacomm) or the HP98626A (Serial).
19110 !                   Depending on which card is used, the appropriate
19120 !                   registers are selected.
19130 !
19140    OPTION BASE 1
19150    COM /Status/ INTEGER Sc,Connect_flg,E_flg,Debug_flg,Response$
19160    INTEGER Baud_rate,B,Num_chars,Response_flg,Index1
19170    DIM Input$[256],Term$[256],In$[256] BUFFER,From_232$[256]
19180    DIM Num_chars$[6],Num_ltrs$[6],Out$[256] BUFFER
19190    DIM White_print$[1],Crlf$[2]
19200    IF Debug_flg THEN PRINT TABXY(1,1);"ENTERING Comm "
19210    ON ERROR GOSUB Error
19220    !
19230    !
19240    !
19250    IF Connect_flg THEN After_init
19260    Sc=30
19270    ASSIGN @Find_it TO Sc;RETURN Outcome
19280    IF Outcome=0 THEN
19290       ASSIGN @Find_it TO *
19300       CONTROL Sc,0;1                  ! Reset RS-232 interface.
19310       CONTROL Sc,3;1                  ! Async link protocol.
19320       CONTROL Sc,0;1                  ! Set Async toggle.
19330       CONTROL Sc,8;1+2                ! Set RTS and DTR lines.
19340       CONTROL Sc,16;0                 ! Disable connection timeout.
19350       CONTROL Sc,17;0                 ! Disable no activity timeout.
19360       CONTROL Sc,18;0                 ! Disable NO CARRIER timeout.
19370       CONTROL Sc,19;0                 ! Disable transmit timeout.
19380       CONTROL Sc,20;14                ! TX baud speed = 9600
19390       CONTROL Sc,21;14                ! RX baud speed = 9600
19400       CONTROL Sc,22;0                 ! No handshake with Whedco.
19410       CONTROL Sc,23;0                 ! No hardwired handshake.
19420       CONTROL Sc,34;2                 ! 7 bits/character.
19430       CONTROL Sc,35;0                 ! 1 stop bit.
19440       CONTROL Sc,36;1                 ! ODD parity.
19450       Connect_flg=1
19460    ELSE
19470       Sc=8
19480       ASSIGN @Find_it TO *
19490       ASSIGN @Find_it TO Sc;RETURN Outcome
19500       IF Outcome<>0 THEN
19510          PRINT "RS-232 card not installed. Please install and reboot."
19520          ASSIGN @Find_it TO *
19530          STOP
19540       END IF
19550       ASSIGN @Find_it TO *
19560       RESET Sc
19570       CONTROL Sc,0;1                 ! Reset the RS-232 interface.
19580       CONTROL Sc,3;Baud_rate         ! Set the baud rate.
19590       CONTROL Sc,4;8+2               ! UART 8 bits/char.  ODD parity.
19600       CONTROL Sc,5;3                 ! UART DTR line active.
19610       CONTROL Sc,12;128+32+16        ! Disable CD,DSR,CTS
19620          STATUS Sc,3;B                  ! Confirm speed to user.
19630          Connect_flg=1
19640       END IF
19650 After_init:!
19660       White_print$=CHR$(136)
19670       Crlf$=CHR$(13)&CHR$(10)
19680       PRINT CHR$(128)&CHR$(136);      ! Set up the screen.
19690       ASSIGN @Screen TO CRT
19700       ASSIGN @Kbd TO KBD
19710       ASSIGN @Rx TO BUFFER In$
19720       ASSIGN @Tx TO BUFFER Out$
19730       ASSIGN @Uart_out TO Sc
19740       ASSIGN @Uart_in TO Sc
19750       Response_flg=0                  ! Reset command acknowledge flag.
19760       Response$=""                    ! Null out response string.
19770       !
19780       !
19790          ENABLE INTR Sc                 ! Enable interrupt on card.
19800          TRANSFER @Tx TO @Uart_out;CONT ! Enable transfer buffers.
19810          TRANSFER @Uart_in TO @Rx
19820          ON INTR Sc,FNTrap_level GOSUB Read_loop ! Process card interrupts.
19830          IF C$<>"" THEN
19840             GOSUB Send_com              ! Send command out to controller.
19850          ELSE
19860             GOTO Quit                   ! If null command, exit quick.
19870          END IF
19880          !
19890          !
19900          !
19910 Wait_for_it:WHILE NOT Response_flg     ! Waiting for acknowledgement.
19920                   GOSUB Read_loop
19930                   IF NPAR=2 THEN       ! We are waiting for data to be
19940                                        ! sent by the Whedco controller.
19950                      LOOP
19960                         GOSUB Read_loop
19970                         IF (POS(Response$,"*")) THEN
19980                            Response$=Response$[POS(Response$,"*"),LEN(Response$
))]
19990                            Response_flg=1
20000                         END IF
20010                      EXIT IF ((Response_flg=1) AND (POS(Response$,Crlf$)))
20020                      END LOOP
20030                   ELSE
20040                      WHILE NOT ((POS(Response$,"*")) OR (POS(Response$,"?")))
20050                         GOSUB Read_loop
20060                      END WHILE
20070                      Index1=POS(Response$,"*")
20080                      IF Index1=0 THEN ! Must be a "?" (Whedco command error).
20090                         ! Must be a "?" (Whedco command error).
20100                         E_flg=1       ! Notify via error flag.
20110                         Response_flg=1
20120                      ELSE
20130                         ! Normal command interpretation.
20140                         E_flg=0
20150                         Response_flg=1
20160                      END IF
20170                   END IF
20180               END WHILE
20190          GOTO Quit
20200       !
20210       !
20220 Read_loop:! Read in serial data from Whedco.
20230          !
20240       STATUS @Rx,4;Num_chars              ! Number of characters to
```

```
20250      IF Num_chars=0 THEN RETURN          ! receive, if 0 try again.
20260      Num_chars$="#,"&VAL$(Num_chars)&"A"  ! Set up the IMAGE for ENTER.
20270      ENTER @Rx USING Num_chars$;From_232$       ! Transfer contents.
20280      Response$=Response$&From_232$              ! Build up dialogue.
20290      RETURN                                     ! Update pointers.
20300 !
20310 !
20320 Send_com:Term$=Crlf$[1,1]&C$&Crlf$
20330         Num_ltrs$="#,"&VAL$(LEN(Term$))&"A"
20340         OUTPUT @Tx USING Num_ltrs$;Term$
20350         Term$=""
20360         RETURN
20370 !
20380 !
20390 Quit:  OFF ERROR
20400        STATUS @Tx,10;Stat
20410        STATUS @Rx,4;Num_bytes
20420        ABORTIO @Uart_out
20430        ASSIGN @Tx TO *
20440        CONTROL @Rx,8;0
20450        STATUS @Rx,10;Stat
20460        STATUS @Rx,4;Num_bytes
20470        ABORTIO @Uart_in
20480        ASSIGN @Rx TO *
20490        DISABLE INTR Sc
20500        SUBEXIT
20510 !
20520 !
20530 Error:PRINT "HANDLING Comm ERROR"
20540        IF ERRN<>167 THEN Other_error
20550        IF Sc=8 THEN ! Process the simple card.
20560          STATUS Sc,10;Uart_error
20570          IF BIT(Uart_error,1) THEN Overrun
20580          IF BIT(Uart_error,2) THEN Parity
20590          IF BIT(Uart_error,4) THEN Break1
20600          IF BIT(Uart_error,3) THEN Framing
20610          E_flg=1
20620          PAUSE
20630          RETURN
20640        ELSE
20650          PRINT ERRM$
20660          E_flg=1
20670          PAUSE
20680          RETURN
20690        END IF
20700 !
20710 !
20720 Other:  PRINT "UART error status: ";Uart_error
20730        E_flg=1
20740        RETURN
20750 !
20760 !
20770 Overrun:PRINT "Overrun"
20780        E_flg=1
20790        RETURN
20800 !
20810 !
20820 Parity: PRINT "Parity"
20830        E_flg=1
20840        RETURN
20850 !
20860 !
20870 Break1: PRINT "Break"
20880        E_flg=1
20890          RETURN
20900 !
20910 !
20920 Framing:PRINT "Framing"
20930          E_flg=1
20940          RETURN
20950 !
20960 !
20970 Other_error:PRINT "Error message: ";ERRM$
20980              PAUSE
20990              E_flg=1
21000              SUBEXIT
21010 !
21020 !
21030 SUBEND
21040 !
21050 !**********************************************************************
21060 !
21070 DEF FNAsk(Prompt$)
21080   OFF KEY
21090   DISP Prompt$;
21100   INPUT "",Yn$
21110   Yn$=UPC$(Yn$[1,1])
21120   SELECT Yn$
21130          CASE ="Y"
21140               RETURN 1
21150          CASE ="N",=""
21160               RETURN 0
21170          CASE ELSE
21180               RETURN 0
21190   END SELECT
21200 FNEND
21210 !
21220 !**********************************************************************
21230 !
21240 SUB Clear_crtt
21250 SUBEND
21260 !
21270 !**********************************************************************
21280 !
21290  DEF FNTrap_level
21300     RETURN VAL(SYSTEM$("SYSTEM PRIORITY"))+1
21310  FNEND
21320  !
21330  !**********************************************************************
21340  !
21350 SUB Clear_crt(OPTIONAL INTEGER Start_line,Num_of_lines)
21360     !
21370     INTEGER I
21380     DIM Clear_line$[80]
21390     Clear_line$="                                        "
21400     IF NPAR=0 THEN
21410        OUTPUT KBD;CHR$(255)&CHR$(75);
21420     ELSE
21430        PRINT TABXY(1,Start_line);"";RPT$(Clear_line$,Num_of_lines)
21440        PRINT TABXY(1,Start_line);"";
21450        SUBEXIT
21460     END IF
21470 SUBEND
21480 !
21490 !**********************************************************************
21500 !
21510 SUB Store_file(COMPLEX Matrix(*))
21520     !
```

```
21530 COM /Freq_data/ Freq$(*),Freq_cent(*),Freq_span(*)
21540 COM /Paths/ @Nwa,@Nwa_data,Hp8510,Hp8753
21550 COM /Labels/ Target$,Version$,Mode$,Out_type$,Debug$
21560 COM /Beam_data/ Angle,Angle$,Bin_rng(*),Beam(*)
21570 COM /Flags/ INTEGER Cal_flag(*),Meas_flag(*),Combinations,Fdef,Pdef
21580 COM /Errors/ Msi$,Err_file$
21590 !
21600 COM /Store_path/ @Coh1,@Coh2,@Coh4
21610 COM /Trace_no/ INTEGER In_samples,In_trace
21620 COM /N_data/ INTEGER Npts,Ntrace
21630     !
21640     !
21650     INTEGER Records_per_set,T
21660     REAL Bytes_per_set
21670     DIM Suffix$[2],Dir$[4],Vol$[11]
21680     ALLOCATE COMPLEX Trace(11)
21690     !
21700     !
21710        Time=TIMEDATE
21720        CALL Date_string(Meas_keep$,Time)
21730        Suffix$="MM"
21740        Meas_keep$=Meas_keep$[5,10]&Suffix$
21750        ALLOCATE Temp$[80]
21760        ALLOCATE Ssss$[3]
21770        Ssss$=VAL$(In_samples)
21780        Temp$=FNPathname$(Meas_keep$,Msi$)
21790        PRINT TABXY(10,10);"Filename is "&Meas_keep$
21800     !
21810     DISP "Saving file."
21820     GOSUB Save_hpux
21830     DEALLOCATE Trace(*)
21840     SUBEXIT
21850          !
21860          !
21870 Save_hpux:!
21880               ! Save data file by HP_UX type.
21890               !
21900       CREATE Temp$,200000
21910       ASSIGN @Coh1 TO Temp$;FORMAT ON
21920       FOR T=1 TO In_samples
21930           FOR P=1 TO 4
21940               MAT Trace= Matrix(T,P,*)
21950                OUTPUT @Coh1;Trace(*)
21960            NEXT P
21970       NEXT T
21980    ASSIGN @Coh1 TO *
21990 RETURN
22000 !
22010 SUBEND
22020 !-------------------------------------------------------------------------
```