027587-1-T

DESIGN AND IMPLEMENTATION OF A C-BAND SINGLE ANTENNA POLARIMETRIC ACTIVE RADAR CALIBRATOR

James J. Ahne
Kamal Sarabandi
Fawwaz T. Ulaby

Radiation Laboratory
University of Michigan
Ann Arbor, Michigan

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

LIST OF APPENDICES

Appendix

ABSTRACT

This report serves as a documentation of the design parameters and performance characteristics of a C-band single antenna polarimetric active radar calibrator (SAPARC) developed for JPL and NASA at the University of Michigan's Radiation Laboratory. The device is one of four which are currently being constructed for future JPL/NASA Synthetic Aperture Radar (SAR) missions. The report includes details of the SAPARC's RF and digital / analog electronics design, as well as test results from a number of anechoic chamber measurements. Application notes and suggestions are also included throughout.

CHAPTER I

INTRODUCTION

Active and passive radar calibrators are often used in conjunction with airborne and space borne polarimetric imaging SAR platforms. When strategically placed, these devices serve as ground-based calibration targets with specified radar cross sections (RCS). Trihedrals / corner reflectors are by far the most common type of calibration device used; however, their physical size and weight make them undesirable for field deployment. The drawbacks associated with trihedrals are two-fold. First, an actual deployment of the device can be physically awkward and inconvenient. Trihedrals can be as large as 12 ft by 12 ft by 12 ft, and they can weigh up to 300 pounds. In addition to their cumbersome size and weight, trihedrals tend to act like large rain and snow collectors, thus complicating the chances of performing an accurate calibration.

The second drawback is a bit more subtle, but just as significant. The accuracy of an external calibration of a radar system directly relies on the knowledge of the scattering matrix of the calibration target. Although it is possible to estimate the elements of the scattering matrix of a calibration target analytically, manufacturing tolerances may leave a fair amount of uncertainty in the estimated values. Therefore, it is necessary to measure the calibration targets against a precise calibration target, such as a metallic sphere. This reveals the second drawback of passive calibrators with large physical dimensions, namely that the far field condition and uniform illumination criteria are difficult to meet in the laboratory. Hence, it becomes difficult to accurately define the performance characteristics of passive calibrators of this size and type.

Polarimetric active radar calibrators (PARCs), on the other hand, tend to be much smaller and easier to handle than their passive counterparts. A PARC also yields better calibration measurements since its SAR image can be translated over a dark background, thus providing a higher signal to background ratio. As a result of these advantages, PARCs are rapidly becoming the calibration device of choice for future space borne missions.

As a final point, PARCs traditionally are designed with two antennas which can cause severe degradation in their performance, as will be explained later. Here a new design for the C-band PARC is used which requires a single antenna.

The purpose of this report is to outline of the theory, design, and implementation of the C-band single antenna PARCs developed for NASA and JPL at the University of Michigan. The content of this project reflects the modifications and improvements made to previous PARC and SAPARC units (specifically, an L-band SAPARC prototype built by Sarabandi and Oh for the University of Michigan's Radiation Laboratory [1]). Currently, the C-Band SAPARCs are tentatively planned for field deployment in October 1993, where they will be used as calibration devices for NASA's SIR-C (Shuttle Imaging Radar -C) mission.

CHAPTER II

PARC THEORY / DESIGN CRITERIA

In its simplest form, a PARC consists of a receive antenna, an amplifier, and a transmit antenna (see Figure 2.1). With this configuration, the PARC merely acts like a repeater, whereby an incoming radar signal is received, amplified, and re-transmitted back to the SAR platform. Variations on this simple design do, however, lead to a variety of merits.

Figure 2.2 depicts the modifications which are employed in this project's SAPARC units. The most notable difference is the addition of a delay line along with an orthogonal mode transducer (OMT) / single antenna implementation. The device now serves as a specialized type of repeater, where the signal is captured with respect to one polarization and re-transmitted via its opposite polarization. The pre-amplifier and power amplifier ensure the proper amplification of the signal, while the delay line electrically delays the signal for reasons which will be given later. As a final note, the switches provide the attenuation needed for applicability to SIR-C as well a JPL AIRSAR missions.

2.1 PARC Radar Cross Section

The fundamental equation defining the radar cross section (RCS) of a PARC is given by

$$\sigma = G_{Loop} \frac{G_T G_R \lambda^2}{4\pi} \qquad [2]$$

where G_T and G_R are the transmit and receive antenna gains, and G_{Loop} is the net loop gain associated with the gains and losses from the system's amplifiers, switches, and delay line. Generally speaking, a larger RCS is more desirable. Hence, the driving

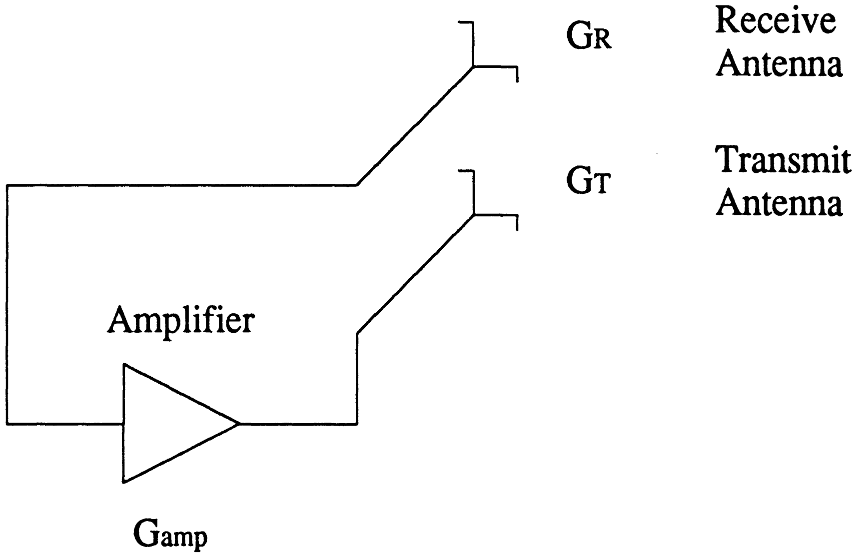


Figure 2.1: Basic PARC Configuration

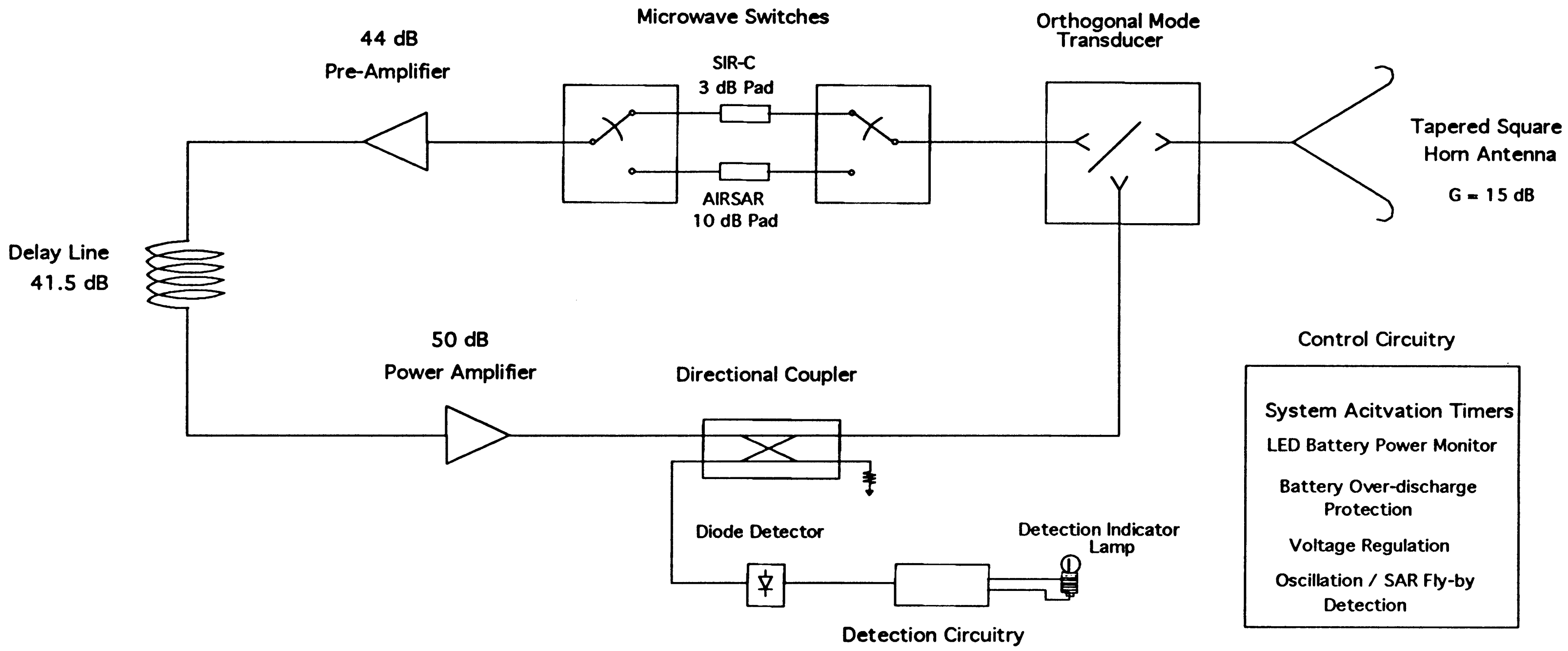


Figure 2.2: C-Band SAPARC Block Diagram

impetus behind most PARC designs is the maximization of G_{Loop}, G_T, and G_R. These parameters, in turn, are limited by beamwidth requirements, transducer isolation performance, and physical size and weight considerations. The following sections address each of these parameters in more detail.

2.2 Antenna Design

In the design of early PARC systems, two antennas, one for transmit and one for receive, were employed to achieve the necessary isolation between the receiver and transmitter modes of the PARC. The transmit and receive antennas were placed in close proximity to one another to meet the compactness requirement of the PARC design. However, since the antennas are in the near field of each other, the RCS pattern of the PARC becomes asymmetric and causes ripples in the phase and amplitude responses which tend to mar the PARC's performance [1] (see Figure 2.3). In order to counter these setbacks and yet to meet the compactness requirement, a single antenna PARC was considered. In this design, the PARC employs a dual polarized horn antenna with a very good polarization isolation and low return loss for both polarization channels. Wide bandwidth and beamwidth with high cross polarization isolation can be achieved through the implementation of an OMT (Orthogonal Mode Transducer) in conjunction with a piecewise tapered square horn.

The geometry of a piecewise tapered horn is shown in Figures 2.4a, 2.4b, and 2.4c. The waveguide discontinuity at a flared intersection excites higher order waveguide modes which are proportional to the flare angle. Since the waveguide is square, the higher order modes can couple energy into the orthogonal channel (TE_{10} to TE_{01}, for example). It was noticed that when the flare angle is less than 5°, the energy transfer from between the orthogonal channels is minimized. However, in order to get the desired aperture over a reasonable length, the square horn can be flared (with angles less than 5°) at many points along its length, thereby simulating an exponential taper. Note that the length of each section should be longer than the wavelength.

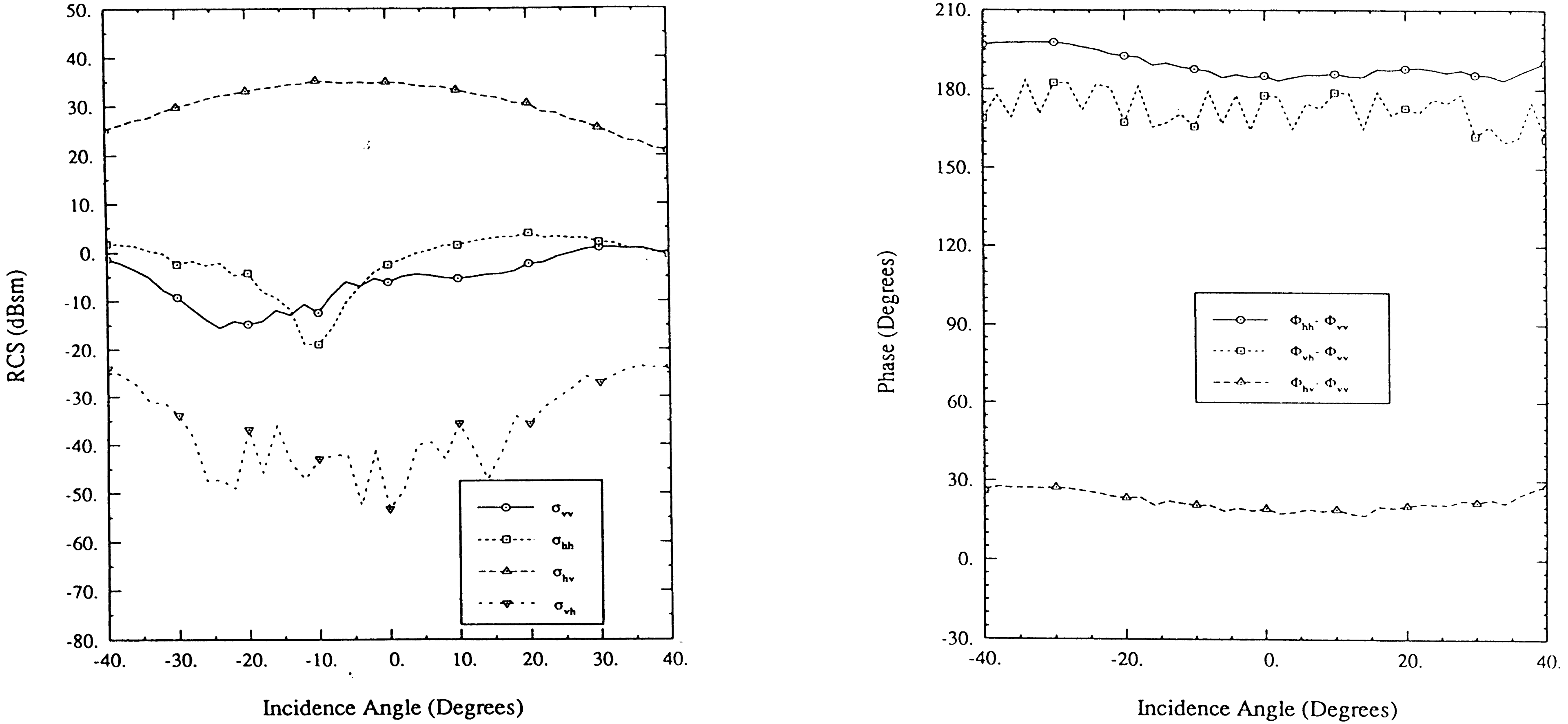
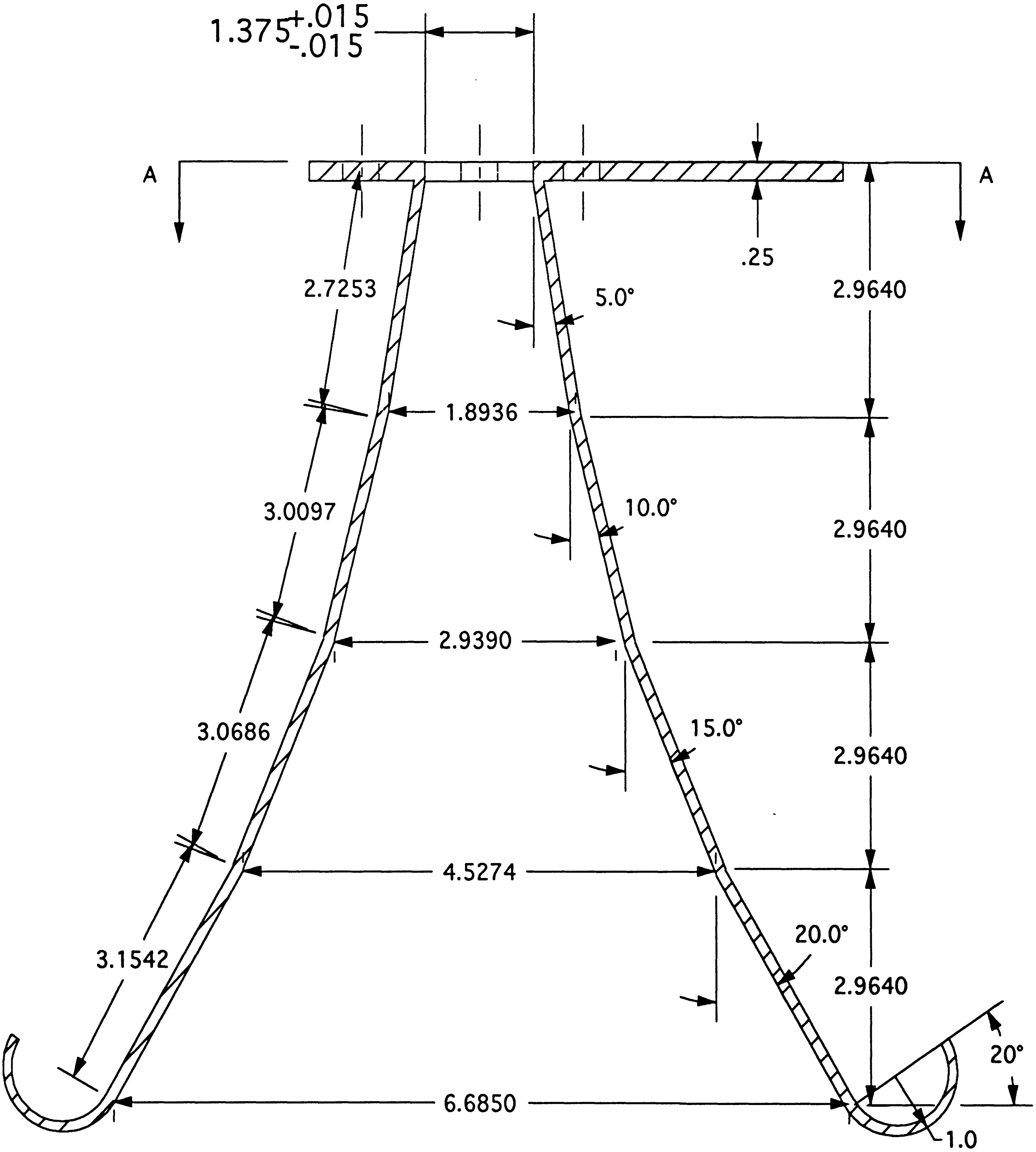
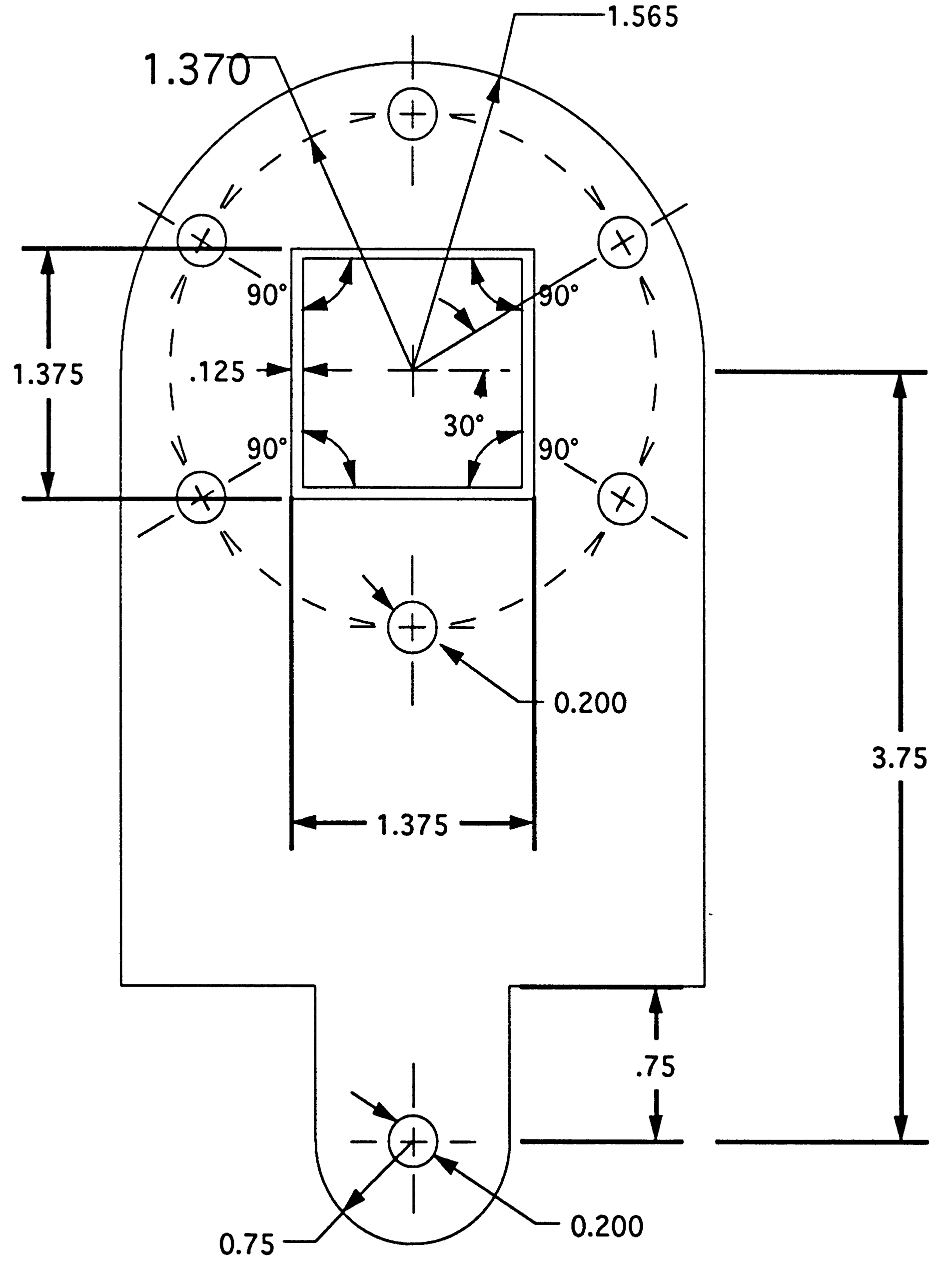


Figure 2.3: Magnitude and Phase Patterns for a Two Antenna PARC System



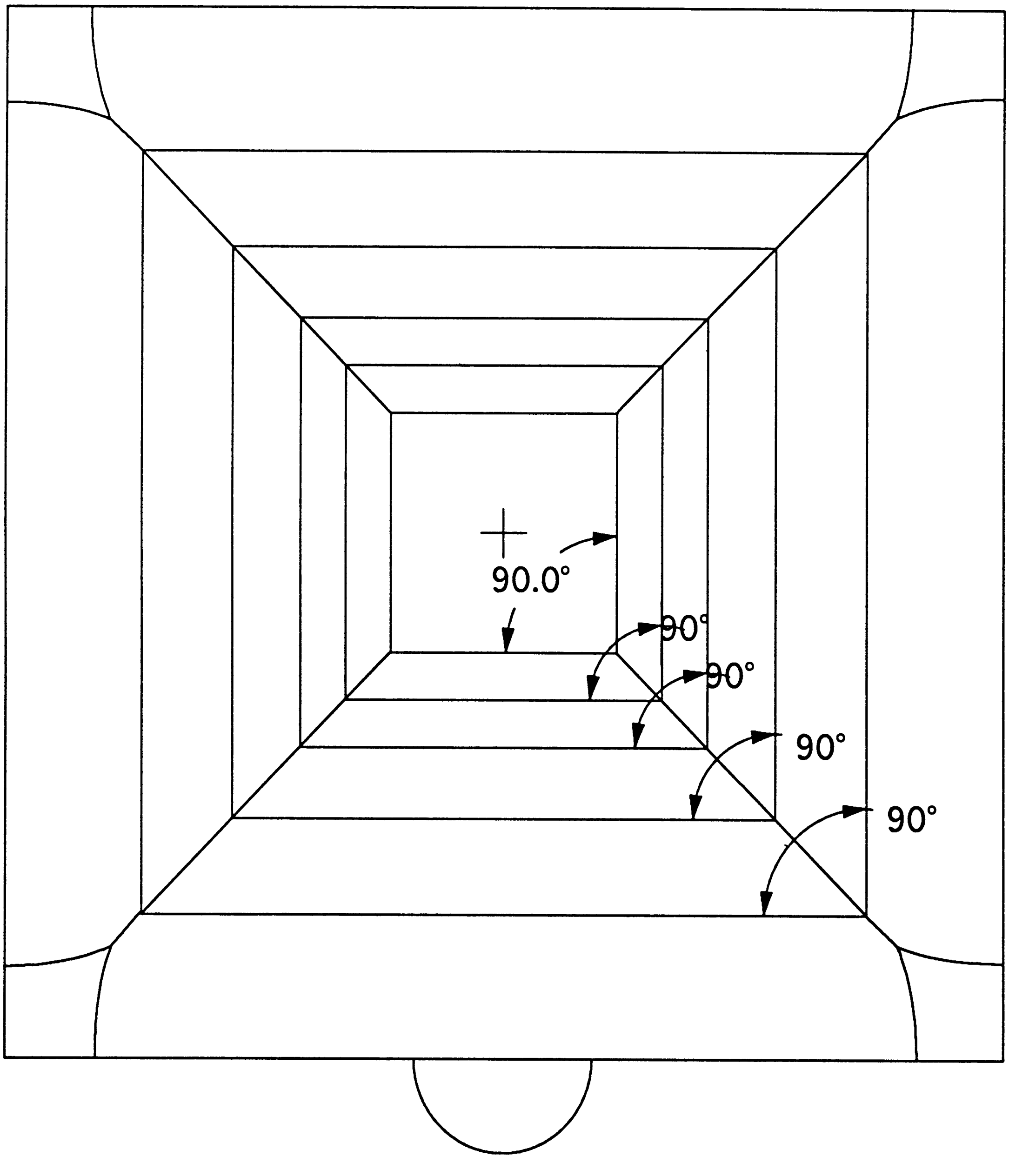
- All tolerances +-0.03 inches (unless noted otherwise)
- All angle tolerances +-1.0 degrees
- All dimensions are in inches and all values pertain to inner dimensions as shown

Figure 2.4a: Side View of the C-Band Tapered Square Horn Antenna



- All dimensions are in inches
- All tolerances +-0.03 inches
- All angle tloerances +-0.5 degrees

Figure 2.4b: Adapter Flange for the C-Band Tapered Square Horn Antenna



- All angle dimensions +-0.5 degrees
- All pertinent dimensions given for side view

Figure 2.4c: Frontal View of the C-Band Tapered Square Horn Antenna

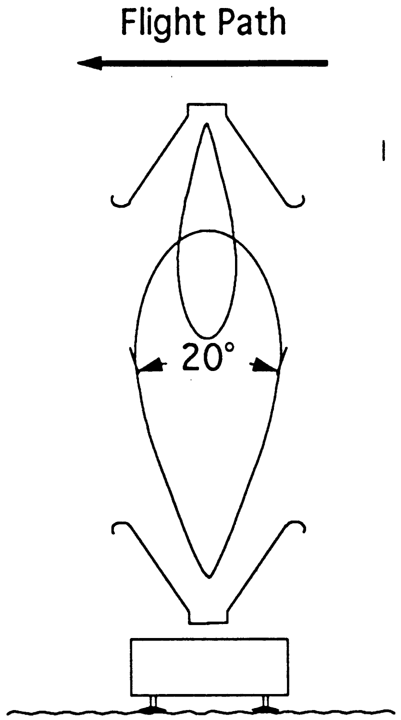
2.2.1 C-Band Antenna Specifications

The goal of the SAPARC's antenna design is to reduce the RF mismatch and cross-talk (i.e. cross polarization generation) while at the same time providing adequate gain, beamwidth, and bandwidth. For the C-band SAPARC, the primary concern of the design was the trade-off between the reciprocal parameters of antenna gain and beamwidth. Physical size and weight were not major factors due to the relatively small wavelength of the C-band system.

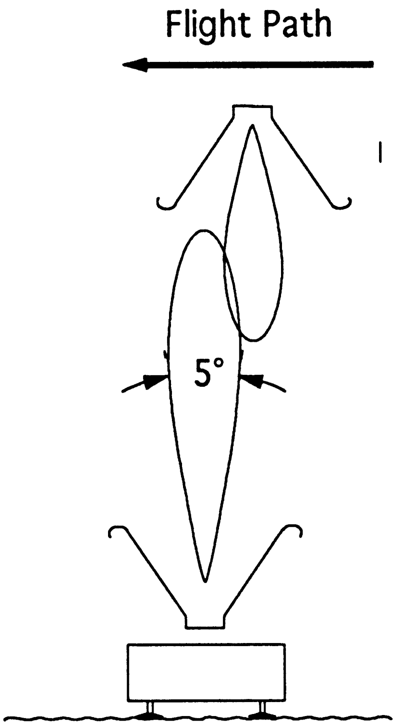
From a practical point of view, the scattering matrix of the SAPARC must be rather insensitive to orientation angles, i.e. a SAPARC should be immune to possible pointing errors). Thus, one of the design goals is to achieve a two-way antenna beamwidth of around 20°. Note that the relatively large beamwidth ensures a successful calibration even if the SAPARC is not directly within the line of sight of the SAR platform. Figure 2.5 demonstrates pictorially the importance of having a wide antenna beamwidth.

A secondary goal was to reduce the sidelobes radiating from the aperture, thereby minimizing the effect of multipath reflections to and from the SAPARC's ground-based position. Multipath contributions yield inaccurate RCS responses since unwanted electromagnetic energy is effectively being collected by the SAPARC antenna system (see Figure 2.6). The nominal RCS response, however, is measured within an anechoic chamber where multipath contributions are negligible. Hence, measurements taken within anechoic chamber and field environments may differ considerably. Using an antenna with small sidelobes is advantageous in that multipath contributions will be reduced; thus, measurements taken during actual field deployment conditions will more closely resemble measurements taken within the chamber environment.

One of the project's early prototypes incorporated the use of a corrugated horn with a square aperture. Note that corrugated horns generally offer improved performance since they reduce the sidelobes in the antenna pattern. Unfortunately, this prototype yielded a high degree of co-polarized mismatch and extremely poor cross polarization isolation. The concept of employing a dielectric lense was also tried; however, the costs of

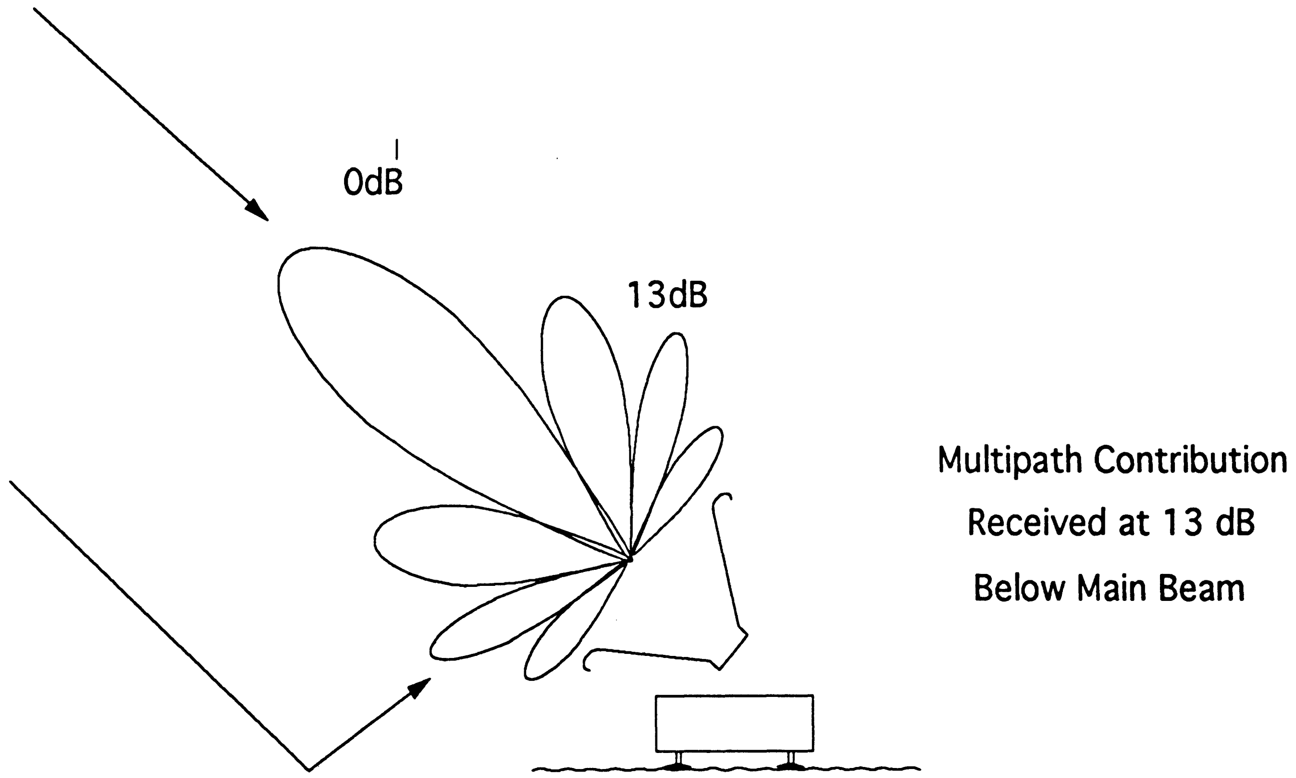


Adequate SAPARC antenna beamwidth

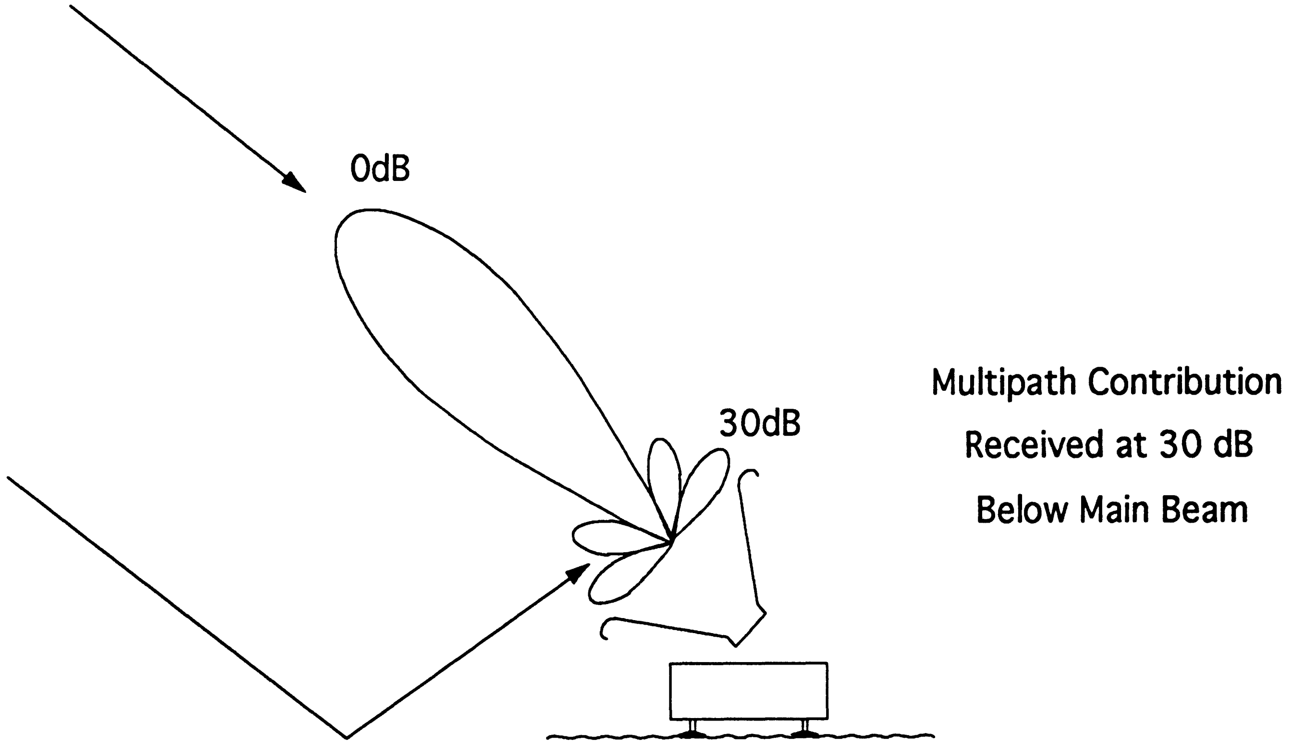


Inadequate SAPARC antenna beamwidth

Figure 2.5: Depiction of Adequate and Inadequate Beamwidths for the Tapered Square Horn



Large Sidelobes



Negligible Sidelobes

Figure 2.6: Multipath Contribution Scenarios for Horn Antennas With and Without Significant Sidelobes

constructing adequate lenses or custom made corrugated horns became much too prohibitive. Therefore, it was decided that the most economically feasible design would forego the multipath considerations.

Due to the high costs involved, first a prototype horn (made from copper plated printed circuit board material) was constructed and found to yield a one way 3 dB antenna pattern beamwidth of 26°. The physical dimensions of the prototype horn was chosen to be 28 cm in length with a square aperture of 17.6 cm by 17.6 cm (3.11λ by 3.11λ). Based on these promising results, a final design was implemented using four equi-length sections flared in 5° steps. As shown in Figure 2.4, the overall length of the horn is 30.11 cm with an aperture of 16.98 cm by 16.98 cm (3λ by 3λ). This final design also incorporated rounded aperture edges which, in theory, reduce the diffraction effects inherent with the abrupt edges of a typical horn [3]. For the center frequency of 5.3 GHz, the radius of curvature for the rounded edge was chosen to be 2.83 cm, which corresponds to a $\lambda / 2$ radius of curvature.

As a final point, the C-band SAPARC's design employs an OMT purchased from Atlantic Microwave (model # OM1370). This device provides cross polarization isolation of better than 50 dB with a VSWR smaller than 1.5 over the frequency range of 5.2 - 5.9 GHz. A VSWR of 1.07 to 1.08 is the typical value for this OMT. See Appendix A for the detailed OMT test specifications. Figure 2.7 depicts the completed horn and OMT combination.

2.2.2 C-Band Antenna Performance

The results from the final design varied from fair to excellent. The largest disappointment came in the form of the narrow 3 dB beamwidths demonstrated in Figures 3.24 and 3.26 (0° and 45° Orientations). As shown in these pattern measurements, the antenna provides two way beamwidths of 15° for both orientations. Although these beamwidths are relatively narrow, the overall capabilities of the SAPARC will not be degraded.

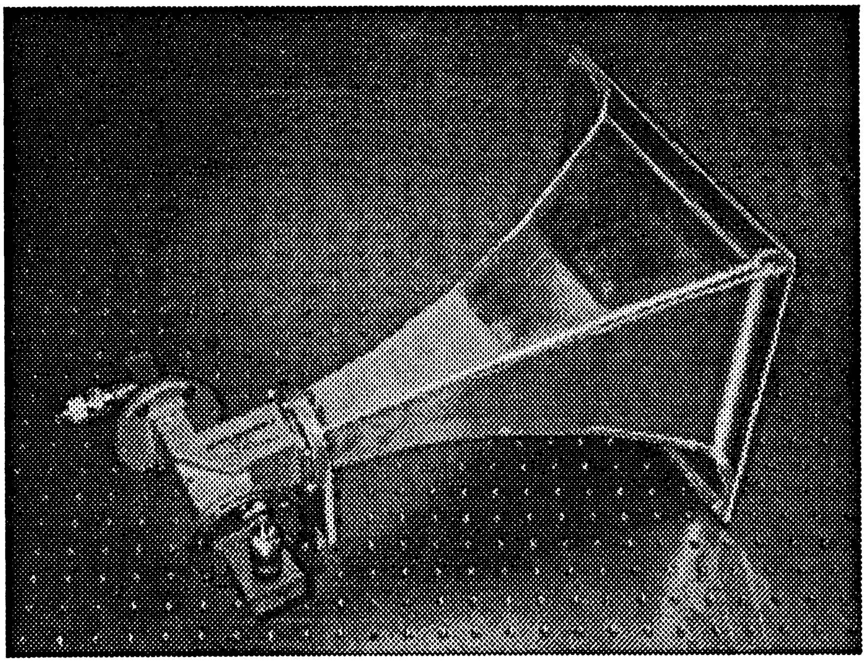


Figure 2.7: C-Band SAPARC Antenna System

The horn design does, however, provide exceptional cross polarization isolation between the receive and transmit ports. Appendix D outlines the steps taken in measuring the horn's cross polarization performance. These measurements were conducted within the UM Radiation Laboratory's anechoic chamber, as shown in Figure D.1. Figures D.2 a - h reflect the time and frequency domain responses taken with a Hewlett Packard 8510 network analyzer. Note that the frequency domain plots incorporated time gating and background subtraction over a frequency range of 5.0 - 5.6 GHz.

From the plots given in Figure D.2, we can deduce that the antenna system (horn and OMT combination) yields a cross polarization isolation exceeding 42 dB. This quality of the horn is instrumental in providing the SAPARC's 38 dB of overall cross polarization isolation (Figure 3.24) and relatively high G_{Loop} gain. Note the 4 dB discrepancy between the system's RCS cross polarization isolation and the isolation resulting from the antenna system alone (measured independently). One possible explanation is that the system's RCS cross polarization isolation was measured in conjunction with another horn and OMT combination (i.e. the radar's antenna system) which similarly possesses a finite isolation capability. The antenna system isolation measurements, on the other hand, used a metal sphere which theoretically acts like a perfectly pure reflective polarizer, i.e. the sphere cannot de-polarize incident electromagnetic waves (as can an imperfect horn antenna). Therefore, it is reasonable to expect that the 4 dB discrepancy is due, at least in part, to the imperfections of the radar's antenna system. Other sources of calibration error may contribute to the discrepancy as well.

Finally, it should be noted that the maximization of G_{Loop} is dependent upon the level of isolation between the receive and transmit ports on the antenna. Refer to Figure 2.1. From this simple diagram, one can see how a feedback scenario results whenever a small fraction of energy is coupled from the transmit antenna to the receive antenna. The coupled electromagnetic wave is then repeatedly amplified as the energy continues along the feedback loop. Eventually, the coupled energy will increase to a magnitude which saturates the amplifiers. For obvious reasons, this situation cannot be tolerated for a PARC design. Therefore, the antenna system's cross polarization isolation must be large enough to prevent the occurrence of a feedback loop.

2.2.3 Summary of Horn Performance

Physical Characteristics		Electrical Characteristics	
Aperture Size:	16.98cm X 16.98cm	Gain:	15.93 dB
Length:	30.11 cm	2-Way 3 dB Beamwidth:	15°
Weight:	3.5 lbs.	Cross Polarization Isolation:	42 dB
Material:	Aluminum		
Manufacturer:	Midwest Enterprises		

Table 2.1: C-Band Antenna System Characteristics

2.3 G_{Loop} Design

2.3.1 Delay Line

At the heart of any PARC system is the G_{Loop} component of the RCS. As mentioned above, a PARC can enhance a calibration measurement by translating its SAR response over a dark background (i.e. a background with a specular surface, such as an airport runway or a large body of water -- See Figure 2.8). This technique is easily implemented by adding a low loss delay line between the receiver and transmitter, as shown in Figure 2.2.

When calculating the length of the delay line, a number of system parameters had to be incorporated in order to insure an adequate SAR delay. The slant range resolution, r_y, is given as 6.67 m for JPL's AIRSAR. As shown in Figure 2.8, the SAPARC should "appear" as if it is situated directly over a body of water. The quantity Δp corresponds to the distance between the physical location of the PARC and its desired SAR image position. An acceptable Δp is approximately 10 pixels (i.e. 10 range bins); therefore,

$$\Delta p = 10 r_y = 66.7 m$$

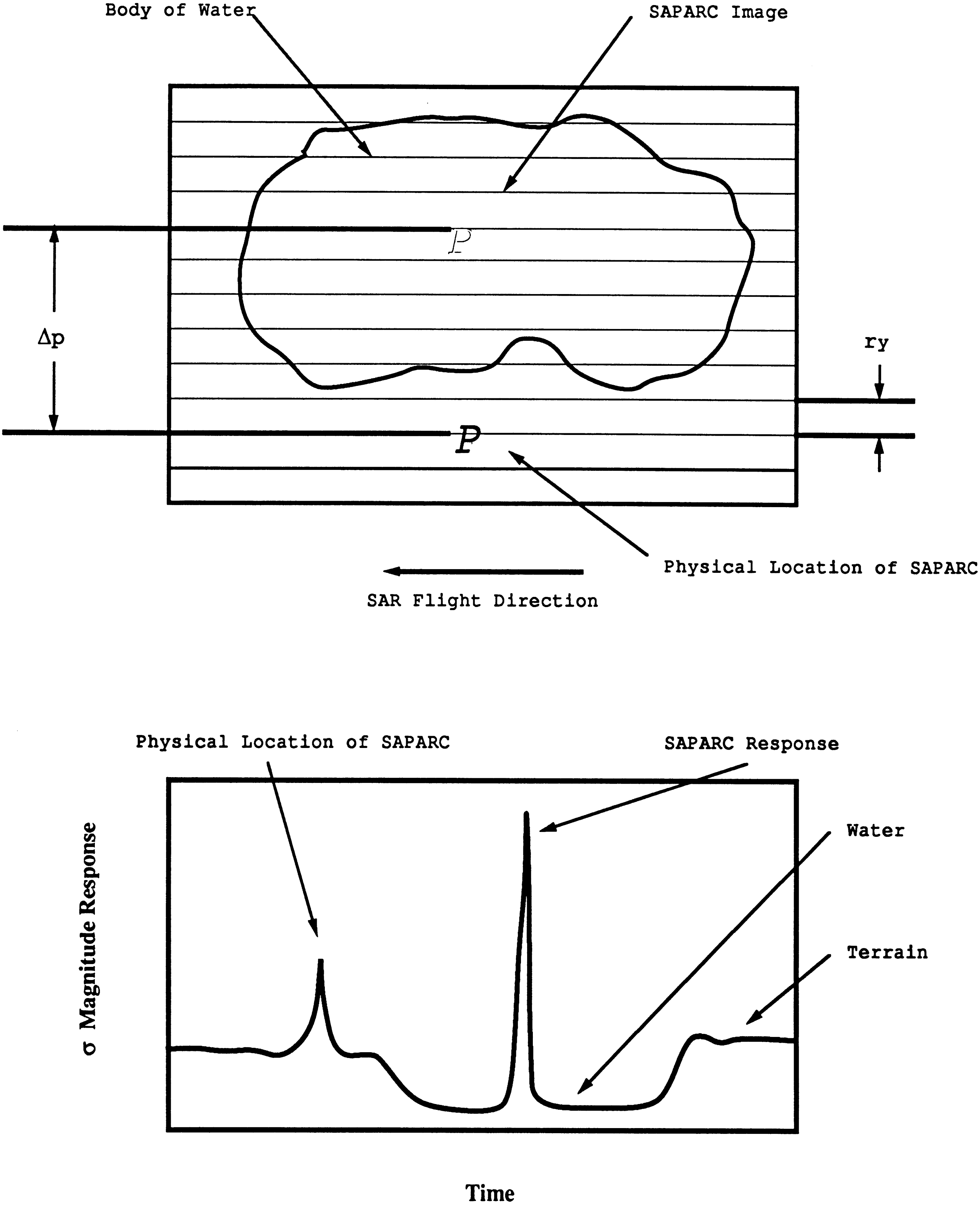


Figure 2.8: The Effects of Placing a Delay Line Within the PARC System

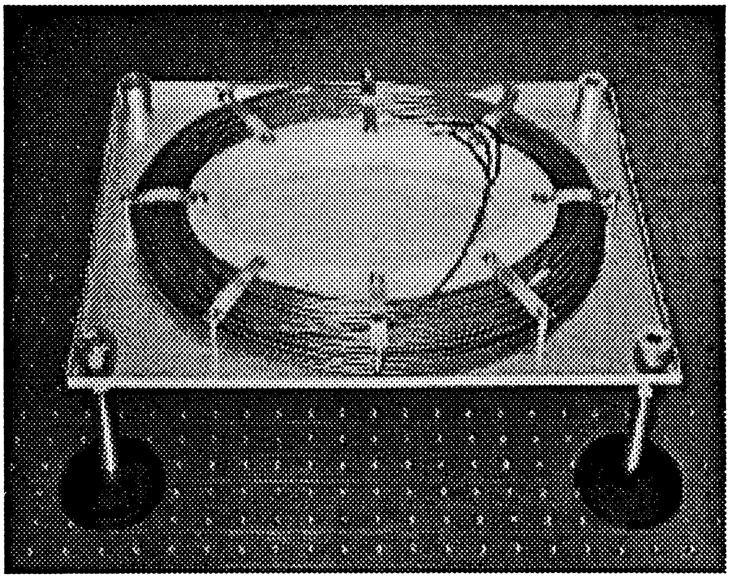


Figure 2.9: C-Band 0.141 Inch Semi-rigid Microporous Coaxial Delay Line

Figure 2.10: S_{11} Time Domain Response of the C-Band Delay Line

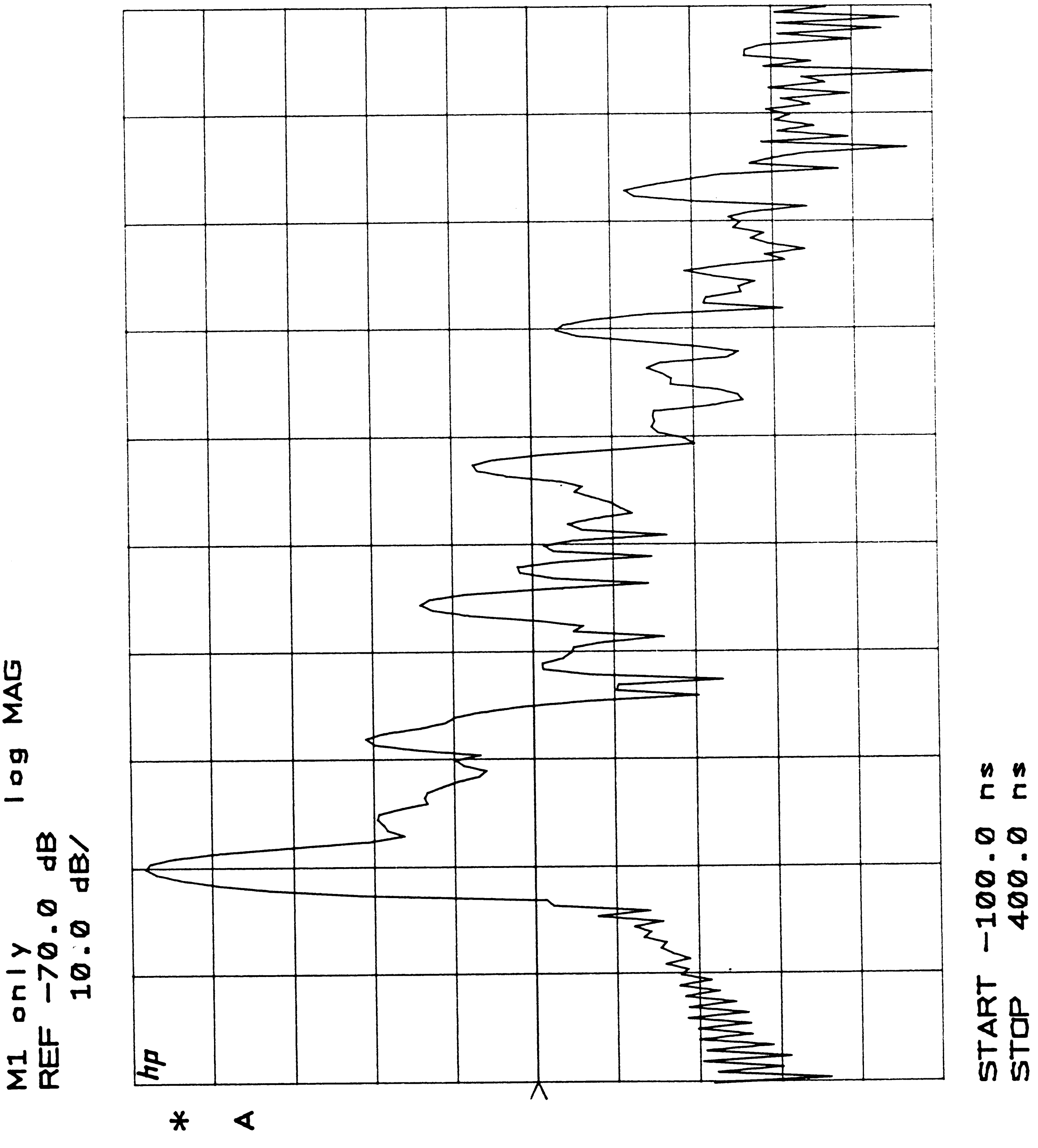
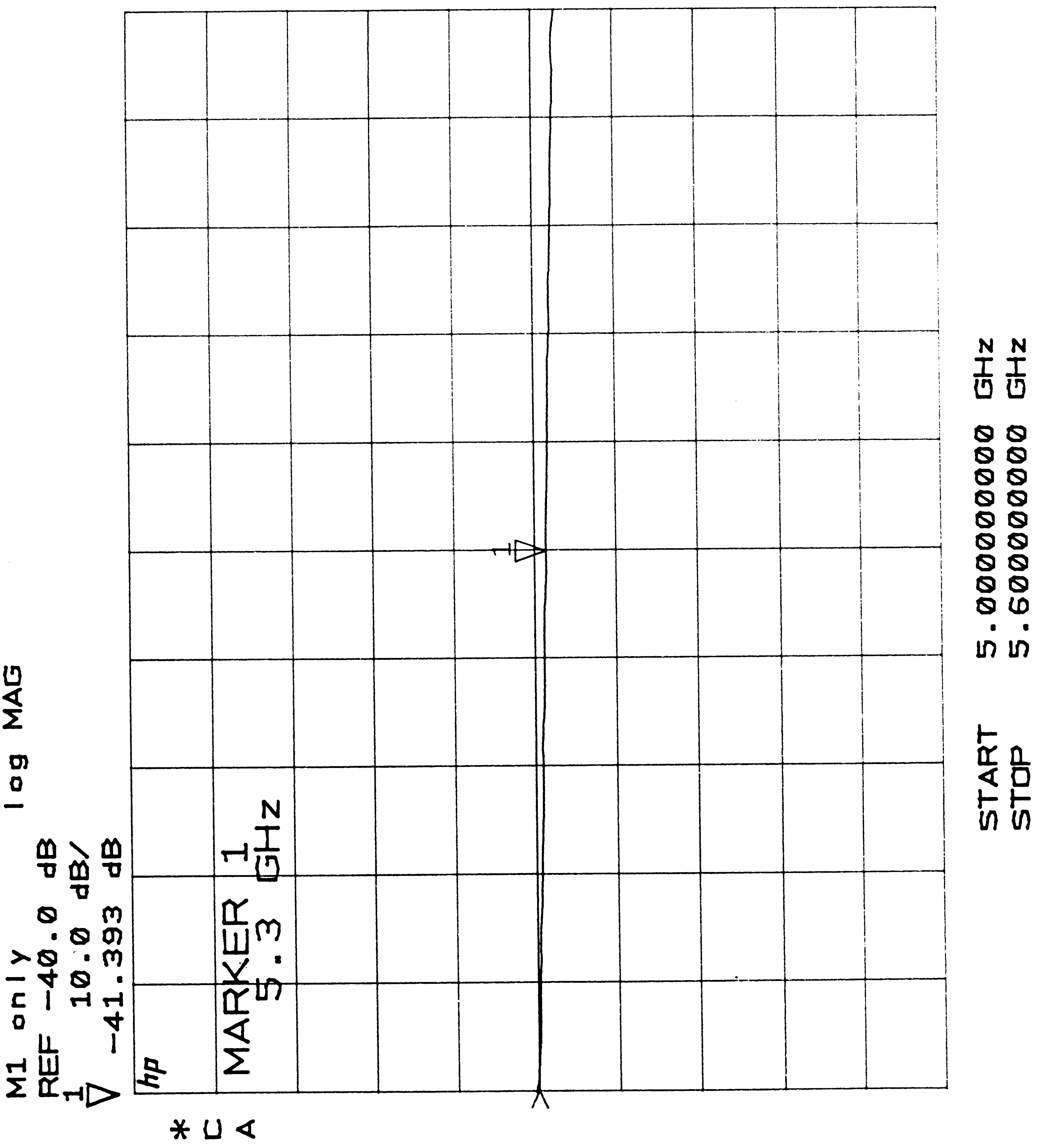


Figure 2.11: S_{21} Frequency Domain Response of the C-Band Delay Line



With Δp now known, the delay D can be found through the simple relationship

$$D = \frac{\Delta p}{c}$$

where c is the speed of light in free space. L_{min}, the minimum length of line needed, is

$$L_{\min} = Dv_{coax} = \frac{\Delta p(0.69c)}{c} = 46.02m = 150.99ft$$

Note that v_{coax} is the velocity of the wave within a coaxial medium. The minimum length of line required is approximately 151 feet, yet in actuality, all of the L- and C-band SAPARCs developed for this project use line lengths of 200 feet (therefore guaranteeing a sufficient delay). Figures 2.9 and 2.10 show the coil of delay line and its corresponding S_{11} time domain response for the C-band SAPARC.

At C-band frequencies, the losses associated with any delay line tend to be quite severe. These losses were minimized by using 0.141 inch semirigid microporous coaxial cable (manufactured in 25 ft-long pieces by Precision Tube, Inc.). The total attenuation resulting from the eight 25 ft-long sections was measured with a Hewlett Packard 8510 Network Analyzer, and was found to be approximately 41.4 dB (see Figure 2.11). Note that the measured loss and the losses quoted by the manufacturer, namely 22 dB, differed by almost 20 dB. It is assumed that this difference results from the losses associated with the 16 connectors.

2.3.2 Amplifiers

The role of amplification in a PARC is to increase the RCS of the antenna system and to compensate for the losses associated with the PARC's delay line and other passive components. The amplifier gain of a SAPARC system must be chosen such that the amplifier operates in the linear region. Amplifier saturation may occur for two reasons: 1. saturation due to high levels of input power received from the SAR platform, and 2. saturation due to feedback oscillations. The latter of the two results from a finite receive and transmit channel isolation (determined by the performance of the OMT and horn antenna).

2.3.2.1 Input Power Calculations

In order to insure that the amplifiers would not be saturated by the received RF, a number of preliminary calculations were made using the Friis transmission formula and known system parameters for JPL's AIRSAR and NASA's Shuttle Imaging Radar (SIR-C). Tables 2.2 and 2.3, respectively, summarize the JPL AIRSAR and NASA SIR-C parameters.

Peak Power	P_t = 1 kW (60 dBm)
Average Power	P_t = 19 W (42.79 dBm)
Wavelength	λ = 0.0566 m
Antenna Gain	G = 23.3 dB
Altitude	15,000 - 40,000 ft (4,572 - 12,192 m)
Incidence Angles	20° - 70°
Pixel Resolution	3.03 m or 12.01 m (1 or 4 Look Azimuth) 6.67 m (Slant Range)

Table 2.2: JPL AIRSAR Parameters

Peak Power	P_t = 2.2 - 2.25 kW (63.42 - 63.52 dBm)
Wavelength	λ = 0.0566 m
Antenna Gain	Unknown (Assume G $\approx$ 20 dB)
Altitude	200 - 225 km
Incidence Angles	15° - 55°
Pixel Resolution	10 - 60 m Range Resolution

Table 2.3: SIR-C Parameters

For JPL's AIRSAR system, the following Friis transmission calculations are applicable. Figure 2.12 depicts the geometry of a typical fly-by, where h is the height of the platform and R is the corresponding range (i.e. distance between the SAR and the calibration unit). The values used in this calculation are for the "worst case" scenario with respect to possible amplifier saturation. Therefore, the dimensions correspond to the case where the maximum amount of power will be received by the SAPARC.

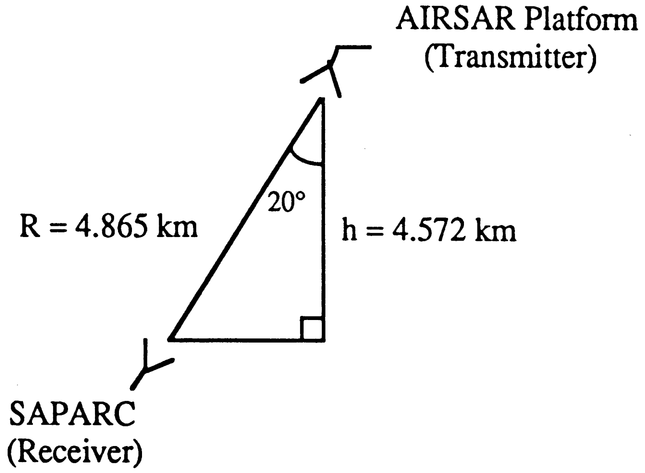


Figure 2.12: JPL AIRSAR Fly-by Geometry

The general form of the Friis transmission formula is

$$P_R = P_T\left(\frac{\lambda}{4\pi R}\right)^2 G_T G_R \qquad [4]$$

whereas

$$R = \frac{h}{\cos(\theta)} = 4.8654km$$

$$P_T = 1KW = 60dBm$$

$$\left(\frac{\lambda}{4\pi R}\right)^2 = \left(\frac{0.0566m}{4\pi(4.8654km)}\right)^2 = -120.67dB$$

$$G_T = 23.3dB$$

$$G_R \approx 15dB$$

Therefore, the maximum input power received by the first stage amplifier will be

$$P_R = 60 + 23.3 + 15 - 120.67 = -22.37dBm$$

Similarly, NASA's SIR-C system, shown in Figure 2.13, will yield the following results.

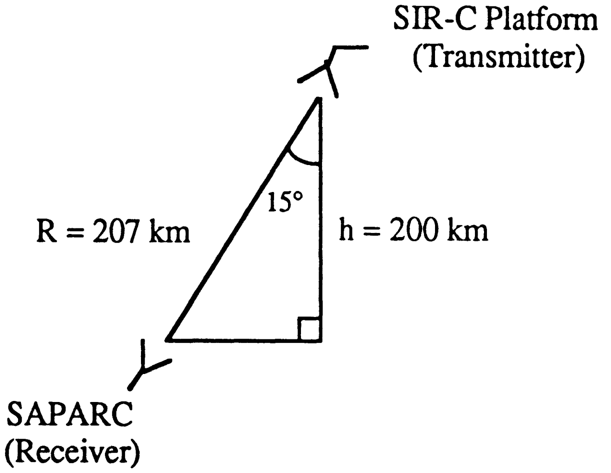


Figure 2.13: NASA SIR-C Fly-by Geometry

The Friis transmission formula gives

$$P_R = P_T\left(\frac{\lambda}{4\pi R}\right)^2 G_T G_R \qquad [4]$$

whereas

$$R = \frac{h}{\cos(\theta)} = 207.05km$$
$$P_T = 2.25KW = 63.52dBm$$
$$\left(\frac{\lambda}{4\pi R}\right)^2 = \left(\frac{0.0566m}{4\pi(207.05km)}\right)^2 = -153.25dB$$
$$G_T \approx 20dB$$
$$G_R \approx 15dB$$

Therefore, the maximum input power received by the first stage amplifier will be

$$P_R = 63.52 - 153.25 + 20 + 15 = -54.73 dBm$$

The gain of the first stage amplifier (i.e. the pre-amplifier) is at most 44 dB (see Appendix B). Hence, the first stage amplifier must be capable of producing the following output power levels in order to insure operation within the linear range of the amplifier.

$$AIRSAR\ P_{out} = P_R + G_{preamp} = -22.37\ dBm + 44\ dB = 21.63\ dBm$$

$$SIR\text{-}C\ P_{out} = P_R + G_{preamp} = -54.73\ dBm + 44\ dB = -10.73\ dBm$$

As shown in Figure 2.2, an additional attenuator of 10 dB was added to the front end of the pre-amplifier as an extra precaution to deter possible saturation during AIRSAR calibrations. Note that the 1 dB compression point for this amplifier is approximately 15 dBm.

Adding an attenuator "in front" of the amplifier degrades the signal to noise ratio; however, in this case, the signal level is much higher than the thermal noise, thus the effect of the additional attenuator is negligible. As will be pointed out in the next section, the noise inherent within the loop can lead to internal oscillations. Therefore, minimizing the noise will theoretically lead to a maximization of G_{Loop}. Yet, in light of the seriousness of amplifier saturation, it was agreed that the benefits resulting from this potentially lower noise performance could not outweigh the assurance that the pre-amplifier is operating within its proper linear range.

2.3.2.2 Feedback Oscillations

An equally serious problem can arise when the system is driven into a state of feedback oscillation. Section 2.3.2.1 alluded to the fact that noise inherent within the system can be amplified just as easily as any incoming RF signal. Oscillations result whenever the amplified noise exceeds the isolation of the antenna system. For the C-band SAPARC design, the net G_{Loop} gain must not exceed 42 dB (the antenna system's cross polarization isolation).

Since G_{Loop} must be less than 42 dB, it follows that

$$G_{Loop} = G_{Amp} + L_{Line} < 42\ dB$$

Rearranging this equation gives

$$G_{Amp} < G_{Loop} + L_{Line} = 42\ dB + 42\ dB$$

Therefore

$$G_{Amp} < 84\ dB$$

Due to the relatively high loss of the delay line, a second amplifier is needed to help boost the signal before it is transmitted back to the SAR platform. As was done with the preamplifier, care must be taken to insure that the second stage amplifier is not driven into saturation. For this reason, the 1dB compression point of the power amplifier was chosen to be 29 dBm.

2.3.3 Attenuation Switches

The principle goal of the attenuation switch (see Figure 2.2 and Appendix C, pg. 6) is to reduce the loop gain thereby allowing the SAPARC to be used for both JPL AIRSAR and NASA SIR-C missions with the maximum allowable RCS. As pointed out in Section 2.3.2.1, JPL's AIRSAR, which flies at significantly lower altitudes than SIR-C, has a correspondingly higher risk for saturating the SAPARC's amplifiers. Conversely, an excessively large G_{Loop} can lead to the saturation of the SAR platform's own receiver.

The original goal was to insert (via the microwave switches) a 10 dB pad within the G_{Loop} path for AIRSAR calibrations. This pad could then be "switched out" for SIR-C applications; however, it was found that the system would occasionally slip into a feedback oscillation mode whenever the 10 dB pad was out of the G_{Loop} line. These oscillations occurred most prevalently whenever the amplifiers were not warmed up, thus implying that the loop gain was right on the fringe of its maximum limit (the amplifiers can exhibit a 1-2 dB drift in gain between initial turn-on and steady state / room temperature operation). Reducing the loop gain by 3 dB eliminated this problem altogether for operating temperatures of 20° C and greater. (See section 3.3 for cold weather operation). Hence, for SIR-C usage, the Mini-Microcircuits switches insert a 3 dB attenuator into the loop, whereas a 10 dB attenuator is inserted for the JPL AIRSAR calibrations.

The Miteq pre-amplifier and power amplifier tandem provide approximately 96 dB of gain, which is well above the limit shown above in Section 2.3.2.2 (see Appendix B). Hence, this explains why an attenuator must be added in front of the first stage amplifier since doing so makes G_{Loop} less than the cross polarization isolation of the antenna system. It would appear as if a 13 dB (i.e. 96dB - 13 dB = 83 dB < G_{amp} max = 84 dB) attenuator is required; however, a trial and error approach showed that a 3 dB attenuator is sufficient to prevent feedback oscillations.

Note that the results shown throughout this report reflect the SAPARC's operation within the SIR-C mode. Similar results can easily be found for the AIRSAR case by simply subtracting 7 dB from the overall SIR-C RCS measurements.

2.4 Control and RF Detection Circuitry

The control and RF detection circuitry serves a two-fold purpose. First, it provides the necessary switching and timing functions for the various power loads; secondly, the circuits display the operating status of the entire system, thereby alerting the user of changes in battery capacity and calibration readiness. The system is comprised of three major components: the Control Printed Wiring Assembly (PWA), the Detection PWA, and the Control Panel. A more detailed description of each of these subsystems is given in the following sections.

2.4.1 Control Circuitry / Control Panel

The single antenna PARCs developed through this project feature custom made control and detection circuits. The features of the control circuitry are as follows:

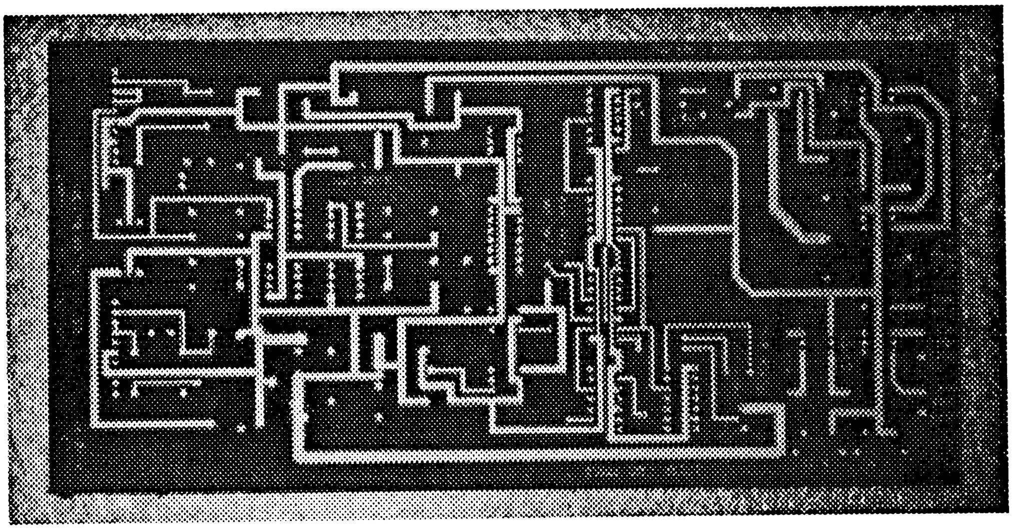


Figure 2.14: SAPARC Control Printed Wiring Assembly (Unstuffed)

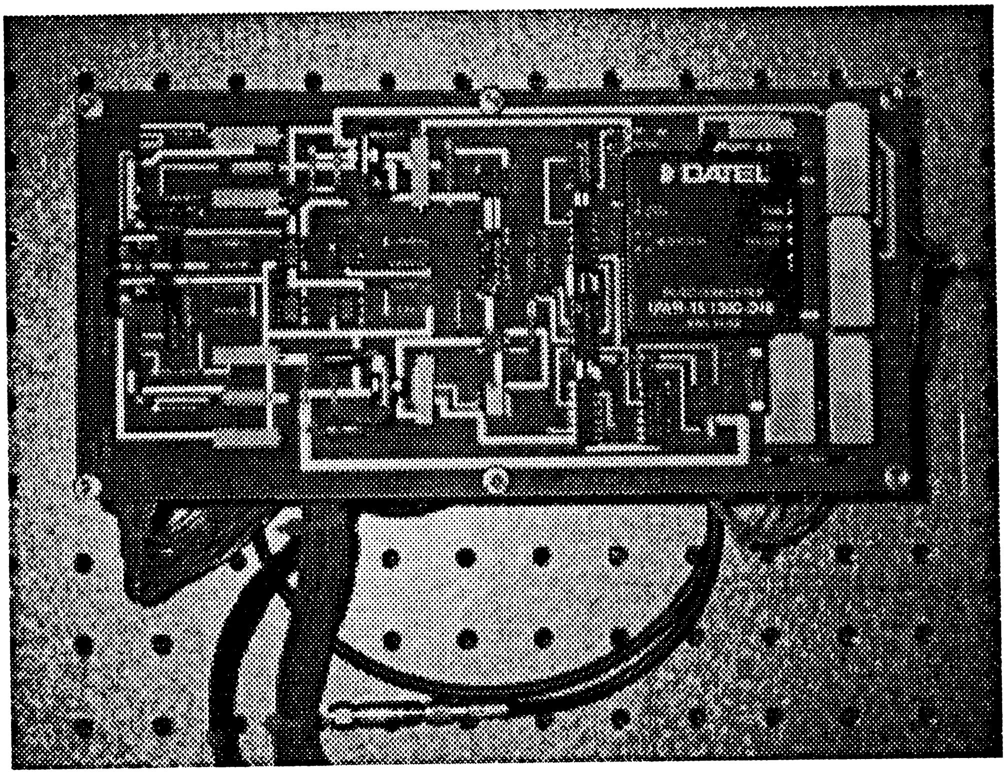


Figure 2.15: SAPARC Control Printed Wiring Assembly (Stuffed)

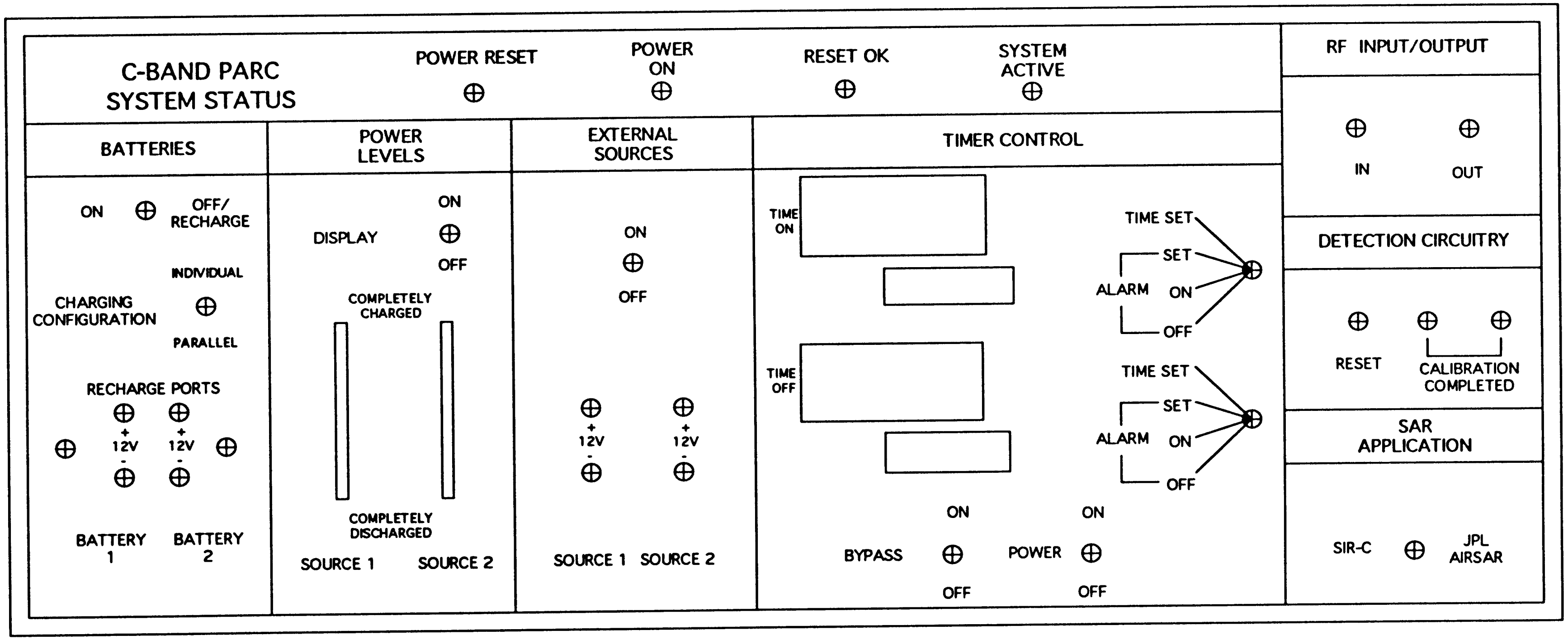


Figure 2.16: Blueprint of the SAPARC Control Panel

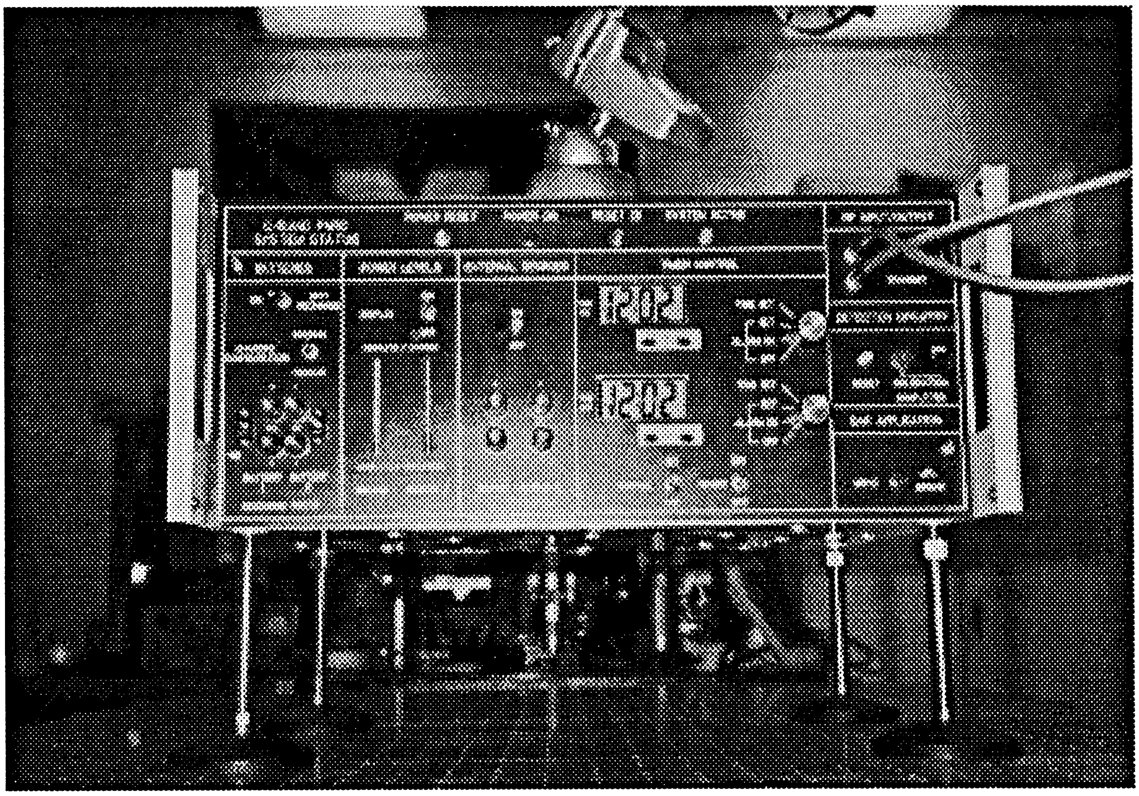


Figure 2.17: SAPARC Front Control Panel

- LED Battery Power Monitor
- Battery Over-discharge Protection
- Automatic System Activation Timers
- Voltage Regulation
- Auxiliary Switching Capability
- External Source Hook-up
- Recharge Ports for Internal Sources
- Easy Detachability for Maintenance

Most of the circuitry (for the functions listed above) is mounted on the Control PWA. This board was designed with EE Designer, a PWA layout software package which can be run on most IBM pc's. The PWA was then manufactured by L. Ross industries in Ann Arbor. (See Figures 2.14 and 2.15).

As a final note, the entire Control PWA / Detection PWA combination can be removed from the system chassis by disconnecting the 50 - pin connector. Before doing so, however, it is advised that the user first disconnect the internal supplies by removing the 7A fuses; the "BATTERIES" switch must then be turned on, and the "POWER RESET" button depressed (for 10 seconds) so that all residual charge held by the internal capacitance of the system can be safely discharged. (See Figure 2.16 and 2.17, both of which depict the front control panel).

2.4.1.1 LED Battery Power Monitor

The LED display mounted on the control panel is driven by a differential amplifier circuit which monitors the gradual drop in voltage of each separate lead acid battery (see Appendix C, pg. 3). Preliminary tests showed that this drop is a linear function of time, whereby the safe operating range exists between $10V \leq V \leq 12.5V$ (refer to the battery operating curves given in Figures 3.12, 3.13, and 3.14). The entire system becomes fully loaded whenever the LED display is activated, thus yielding a more accurate measurement of the battery's remaining capacity. Note that under room temperature conditions, the Yuasa 7 A·hr 12V batteries can operate for up to 5.5 hours (under a full load of 1.2A) before the lower operating voltage threshold is reached. Colder temperatures will significantly limit this capacity; at -10°C, the system can only operate for 4.5 hours before the same lower threshold causes the system to shut down.

The LED display is currently set to measure a voltage range of $10.5V \leq V \leq 12.5V$ for each supply. This range can be adjusted for supplies 1 and 2 by tweaking the potentiometers R8 and R17, respectively.

2.4.1.2 Battery Over-discharge Protection

In conjunction with the LED bar graph display, the discharge protection circuitry similarly monitors the supply voltages through the use of comparitors (see Appendix C, pg. 4). When the lower voltage threshold is reached (i.e. 10V), power to the entire system will be shut off, thus protecting the lead acid batteries from excessive discharging. (The battery's capacity is severely degraded whenever this lower voltage threshold is exceeded for extended periods of time.) This lower threshold can be easily adjusted by tweaking potentiometers R3 and R15 (for supplies 1 and 2, respectively).

2.4.1.3 Automatic System Activation Timers

The system's built-in activation timers can control the operation interval for the SAPARC. In almost all practical situations, the SAPARC needs to be on for a relatively short period of time which can be programmed using the activation timers, thus prolonging the SAPARC's use by conserving battery capacity (see Appendix C, pp. 5-6). Two separate clocks are used: one for activating the high load components, and one for deactivating the entire circuit. The wiring design consists of a number of buffers and opto-isolators which connect the output of the timers (i.e. piezo-electric connections) with the rest of the control circuitry. Note that these connections were made with shielded 20 gauge wire; the first prototype, which did not use shielded wire, experienced occasional transient responses resulting from the switching of the high load components (e.g. amplifiers, heater, etc.). Proper shielding and the use of opto-isolators eliminated this problem altogether.

The timer activation mode can be bypassed for manual operation as well. In the manual mode, the system loads are all activated for immediate and constant operation, thus making this mode ideal for testing purposes. When deploying the SAPARC within a field environment, one should use the bypass in order to insure that the system cannot be driven into a feedback state. Feedback oscillations will occur whenever an object is placed within the SAPARC's antenna beam pattern. Hence, a low-lying tree branch or other similar object may drive the system into an oscillation state. Using the bypass allows the user to "see" if any objects are within range of causing such problems. If the system can operate correctly in the bypass state, then the user will have confidence that the SAPARC will also work while in the automatic mode.

As a last note, the activation timers are independently powered by small cell 1.25V batteries. These cells can be easily replaced by removing the top cover on the SAPARC chassis. A small plastic cover on the clock units must also be removed in order to gain access to the battery compartments.

2.4.1.4 Voltage Regulation

The various subsystems within the SAPARC require supply voltages of ±15V, ±8V, and +5V. The +5V and ±8V regulation is performed by basic 7800 series regulators, whereas the ±15V modes are supplied from DC-DC converters (one on the Detection PWA, and the second on the Control PWA - see Appendix C, pp. 3,4, and 7). The +15V DC-DC converter possesses an efficiency of greater than 80%; hence, the converter outperforms conventional voltage regulation by a considerable margin (i.e. in terms of efficient power use). It should also be noted that conventional regulators cannot supply the relatively large amount of current which is required for the operation of the amplifiers and other possible auxiliary loads.

2.4.1.5 Auxiliary Switching Capability

As mentioned above, the Control PWA is configured so that additional loads can be added (and thus controlled) as the user sees fit (see Appendix C, pg. 2). The voltage output for these auxiliary ports includes ±15V and +24V. Possible loads include recording devices which can monitor the RF power levels received during SAR fly-bys, thereby providing a means for measuring the pattern of the SAR's illuminating footprint.

The Detection PWA does provide a correlation between detected RF and a specific DC output voltage. This capability may be utilized for use with recording devices. Heaters can also be connected to this circuit; however, testing has shown that their use is of little value for reliable temperature stabilization. (See section 2.5).

2.4.1.6 External Source Hook-up

The user can bypass the internal battery supplies by employing the use of the external hook-up jacks located on the front control panel (see Figures 2.16 and 2.17, as well as Appendix C, pg. 1). If external sources are to be used, simply flip the "EXTERNAL SOURCES" switch to the ON position. Hit the "POWER RESET" pushbutton and continue the system operation in the normal fashion. As a final note, **DO NOT CONNECT THE GROUNDS FROM THE EXTERNAL BATTERIES TOGETHER.**

2.4.1.7 Recharge Ports

The user can also recharge the internal batteries via the recharging ports located on the front control panel (see Figures 2.16 and 2.17, as well as Appendix C, pg. 1). Two different port types are provided for the support of varying recharging devices. When recharging, have the "BATTERIES" switch in the OFF / RECHARGE position. The "CHARGING CONFIGURATION" switch permits the user to charge the batteries individually or together in a parallel mode.

2.4.2 Detection Circuitry

The Detection PWA was acquired from an existing two-antenna PARC system developed by Applied Microwave (see Appendix C, pg. 7). This subsystem monitors the power levels which exist at the output of the power amplifier. The threshold for this detection has been set low enough ($P_{\text{min detection}}$ = -48.6 dBm) so that oscillations as well as SAR fly-bys can be recorded. In the original circuit design, a detection would illuminate a small red bulb; in addition to this, a .25A circuit breaker switch (which serves as a permanent recording device) has been added to signal the user that a successful calibration is complete. Note that the circuit breaker takes approximately 60 seconds to trip once a detection is made.

When deploying the SAPARC, the user must be certain that feedback oscillations will not occur during the calibration. Therefore, one must always monitor the detection lamp during final setup preparations. (Recall that the SAPARC system is extremely sensitive to adjacent objects which may reside within the antenna's beamwidth. These objects include nearby bushes, tree limbs, etc.). If a feedback scenario is present, simply press the RESET to clear the Detection PWA circuitry. Continue to re-position the SAPARC as needed so that no errant detections are made.

2.5 Temperature Stabilization

During the initial design phase, one of the primary goals was to develop a system which was insensitive to changes in the ambient temperature. It was assumed that the most sensitive devices would be those which are active, namely the preamp and power amp. To this end, a 24W hybrid heater had been placed on the amplifier combination. Unfortunately, the temperature stabilization tests showed that the most sensitive device was the passive delay line, and not the amplifiers as first suspected. The following test results demonstrate this fact:

Component	Ambient Temp (20° C) Measurement	Lower Extreme (-10° C) Measurement	Difference in Measurement
Power Amp Gain (with heater)	51.12 dB	50.46 dB	0.66 dB
Preamp Gain (with heater)	43.33 dB	42.67 dB	0.66 dB
Delay Line Loss	42.53 dB	39.39 dB	3.14 dB
Total Line Loss (without amps) (with 10dB pad)	53.94 dB	49.67 dB	4.27 dB

Table 2.4: Thermal Variations of Key SAPARC Components

As shown above, temperature stabilization would require either a number of high power heaters or a variable attenuator / gain feedback circuit. The former of these alternatives is somewhat impractical since it would require excessive amounts of battery power. Similarly, the latter option is too expensive for a practical implementation.

An acceptable solution requires a mapping of the G_{Loop} component of the RCS as a function of temperature. Such a mapping is shown in Section 3.3. The goal, then, is to accurately record the system's temperature during an actual field test. The temperature (recorded as a function of time) will then be compared to the G_{Loop} vs. Temperature chart from Section 3.3. Hence, an accurate description of the system's total RCS can be calculated for the exact fly-by time of the SAR platform.

An automatic measurement is obtained through the use of a Dickson 24-hour Temperature Recorder. This device is nestled within the delay line loop located at the base of the SAPARC chassis. For an actual field deployment scenario, the user must activate the temperature recorder while noting the exact time of initial operation. Once this is done, the user is free to leave the deployment area while the rest of the equipment remains in its automated mode.

2.6 Assembled Prototype

Figures 2.18 a-c show the SAPARC in its completed state. Note how the horn antenna is detachable for quick and easy transport of the device.

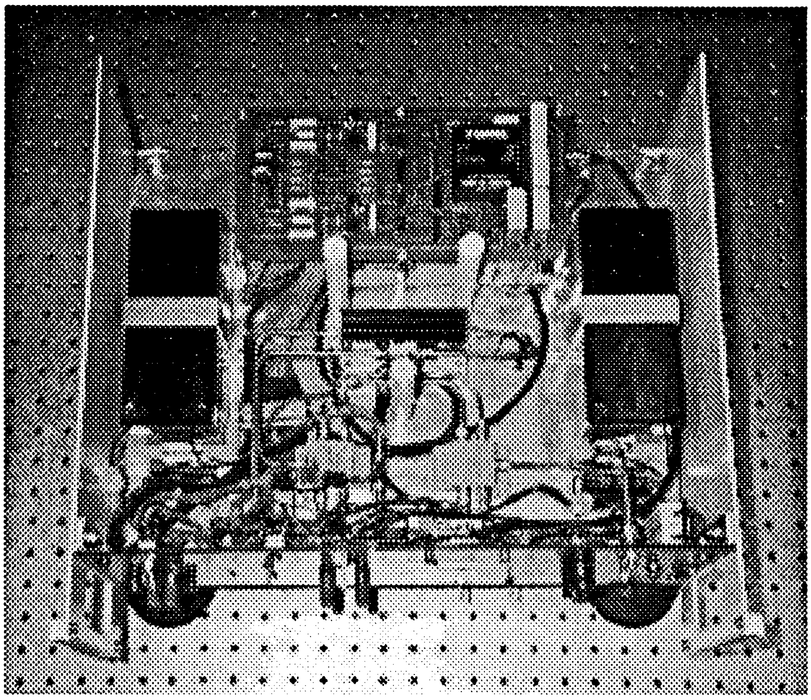


Figure 2.18a: Internal Components of the C-Band SAPARC

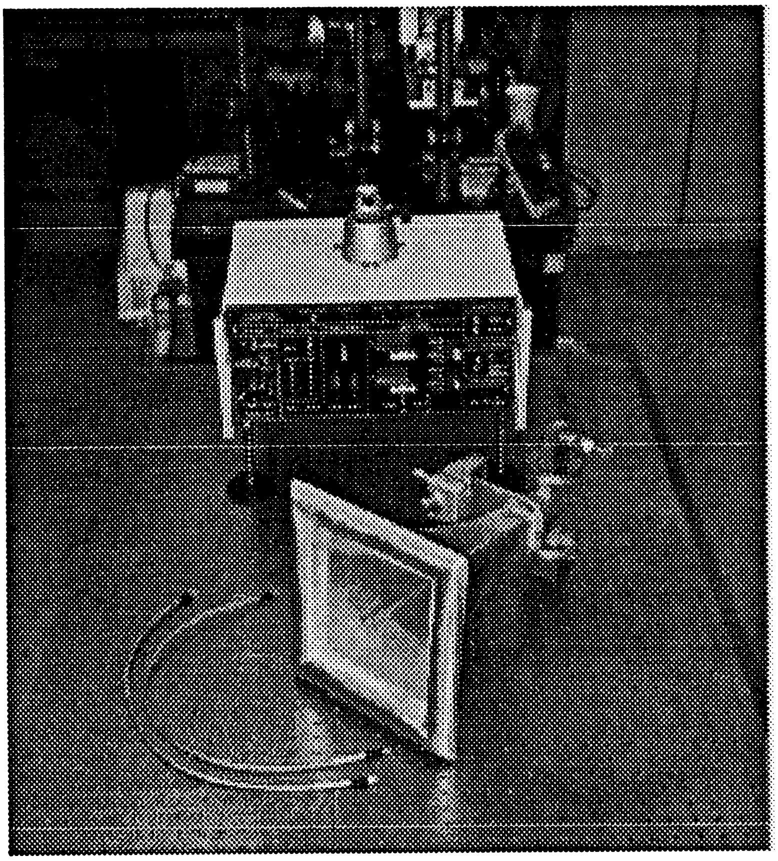


Figure 2.18b: C-Band SAPARC Disassembled For Transport

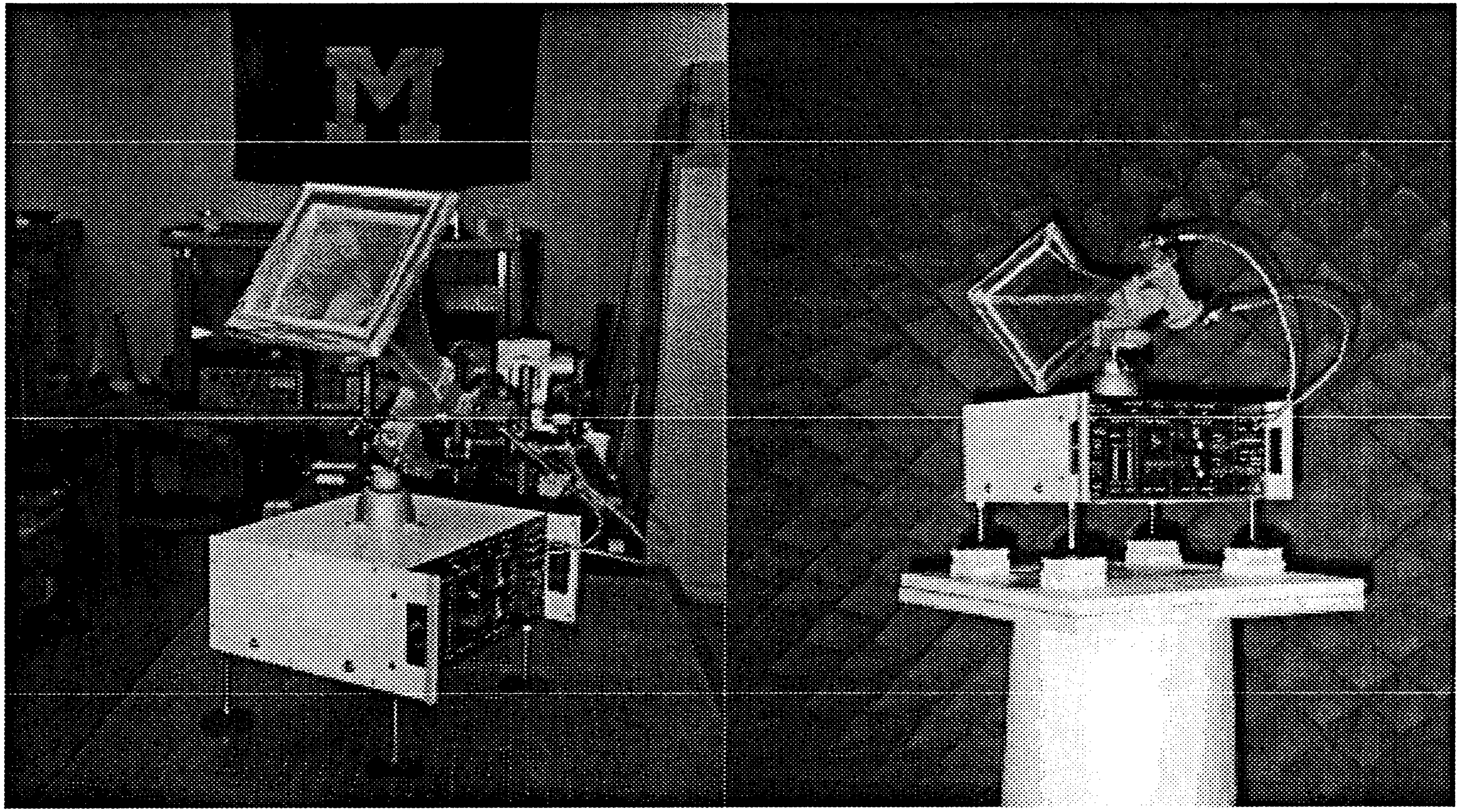


Figure 2.18c: Fully Assembled C-Band SAPARC

CHAPTER III

EXPERIMENTAL RESULTS

As mentioned in Chapter 1, the accuracy of a SAR calibration is highly dependent upon the measured performance of the calibration device. Hence, the measurements taken in accordance with this project must adhere to the following goals:

- Accurate measurement of the scattering matrix for the 0° and 45° antenna orientations.
- 0.2 dB accuracy in the mapping of the thermal gain variations.
- Overall characterization of SAPARC performance with respect to field deployment conditions, including extremes in temperature, all-weather performance, and battery capacity.

3.1 G_{Loop} Measurements

Figures 3.1 and 3.2 depict the S_{21} frequency responses of G_{Loop} for room temperature operation (22° C). For this measurement, a 50 dB attenuator was placed on the receive channel of the SAPARC in order to prevent amplifier saturation. The frequency response for this attenuator is similarly shown in Figure 3.3. From these measurements, G_{Loop} is found to be 44.08 dB. However, as section 3.3 will show, G_{Loop} is highly dependent upon the SAPARC's operating temperature. To find the correct value of G_{Loop} for each SAR calibration, one must refer to the thermal variation chart shown in Figure 3.14.

The SAPARC anechoic chamber tests were performed at room temperature (approximately 22° C). Figure 3.9 (SAPARC RCS for the 0° Orientation) shows that the maximum achievable value for the RCS is 40 dBsm. Using this data in conjunction with the theoretical equation given in section 2.1, the gain of the C-band antenna system is found to be 15.93 dB. The following calculations demonstrate this result.

$$\sigma = G_{Loop}\frac{G_T G_R \lambda^2}{4\pi}$$

where

$$\sigma \approx 40dBsm$$
$$G_{Loop} = 44.08dB$$
$$G_T = G_R = G_{Antenna}$$
$$\frac{\lambda^2}{4\pi} = \frac{(0.0566m)^2}{4\pi} = -35.94dB$$

Rearranging the equation gives

$$G_{Antenna} = \sqrt{\frac{\sigma}{G_{Loop}\frac{\lambda^2}{4\pi}}} = \frac{1}{2}(40.0dBsm - 44.08dB + 35.94dB)$$
$$G_{Antenna} = 15.93dB$$

The equations above demonstrate how the user can easily find the RCS of the SAPARC for any given operating temperature. In other words, when the operating temperature is known, the corresponding value of G_{Loop} will also be known, and hence so will the RCS of the SAPARC unit. The equations are similarly applicable to the 45° SAPARC orientation. For this case, simply subtract the 6 dB difference from the 0° orientation antenna results described above.

3.2 Anechoic Chamber Tests

The University of Michigan Radiation Laboratory maintains a fully equipped 60-footlong, tapered anechoic chamber which is used for conducting antenna pattern measurements and for measuring the scattering characteristics of man-made and natural targets. This chamber is ideal for making accurate measurements of the SAPARC's RCS within a relatively noise-free environment.

A major component of the Radiation Laboratory's polarimetric radar measurement facility is the LCX POLARSCAT system. The parameters of the C-band subsystem are as follows:

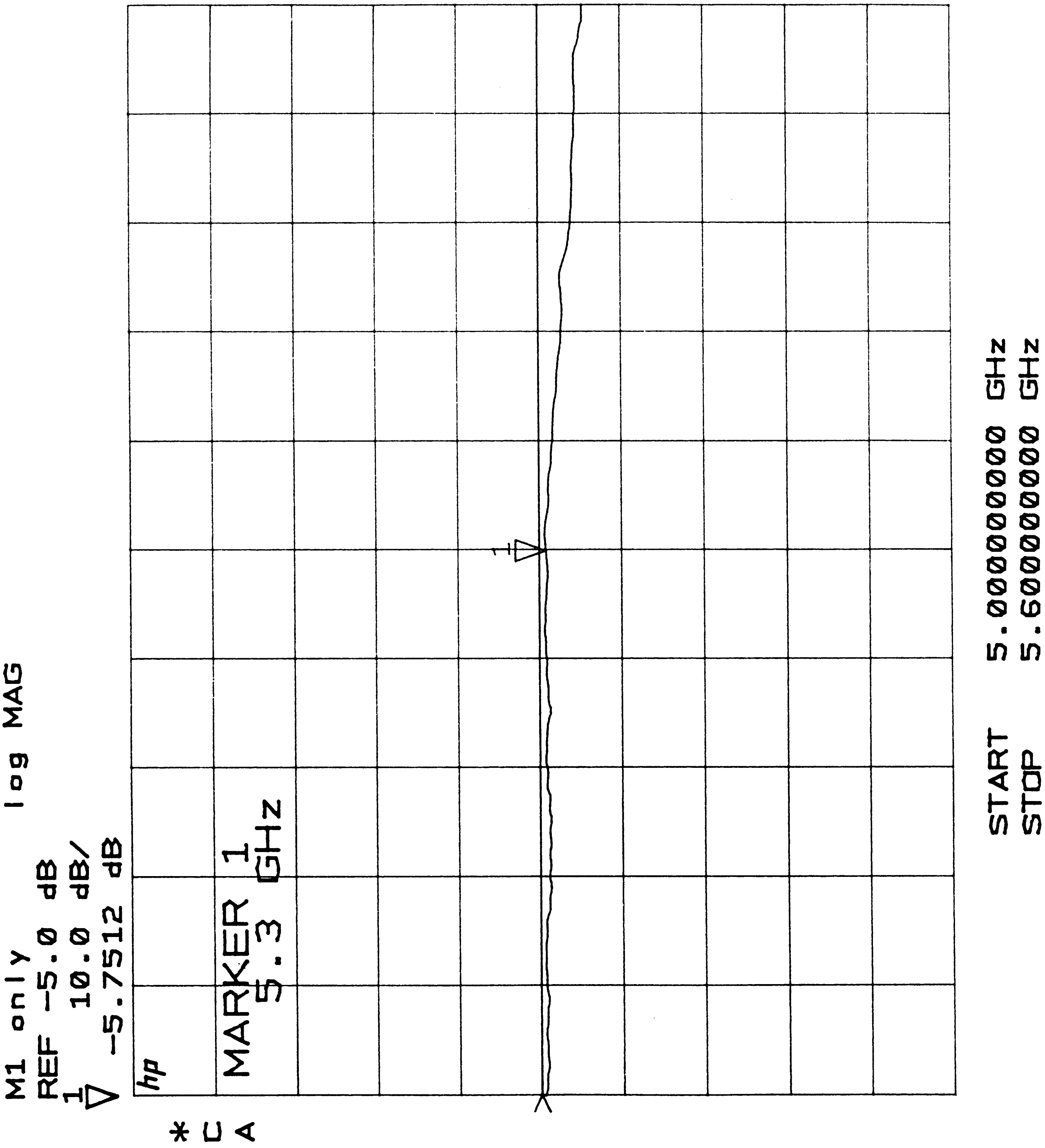
Center Frequency	5.3 GHz
Frequency Bandwidth	0.5 GHz
Antenna Type	Dual Polarized Pyramidal Horn
Antenna Gain	25.3 dB
Beamwidth	8.0°
Far Field ($2d^2 / \lambda$)	5.8 m
XPOL Isolation	45 dB
Calibration Accuracy	± 0.3 dB
Measurement Precision (N>100)	± 0.4 dB
Phase Accuracy	± 3°

Table 3.1: C-Band POLARSCAT Parameters

A large percentage of this system consists of Hewlett Packard components, including an HP 8753 Network Analyzer and HP 9000 Computer with an additional disc drive. Using computer control, polarimetric measurements of the phase and magnitude responses can be taken with respect to changes in target elevation and azimuth angles (Figure 3.4).

The chamber experiments required a center frequency of 5.3 GHz with a 600 MHz bandwidth. Calibrations were performed by using a 14" metallic sphere in accordance with a calibration technique developed by Sarabandi [5] (See Appendix E). Time gating was also employed, whereby a gate span of 10 ns (centered on the target's response) provides an automatic subtraction of background scatterers.

Figure 3.1: Frequency Domain Response of G_{Loop}



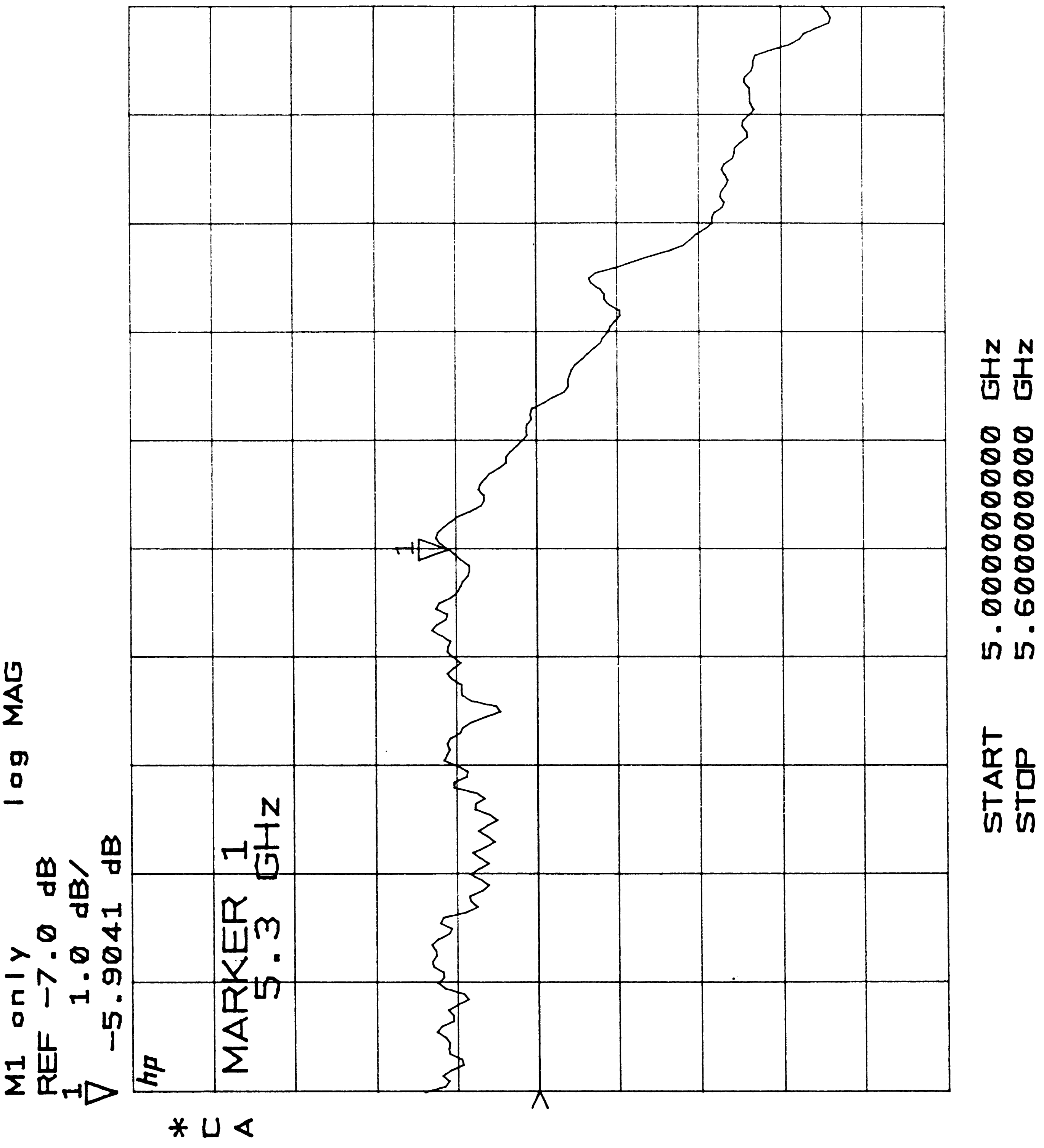
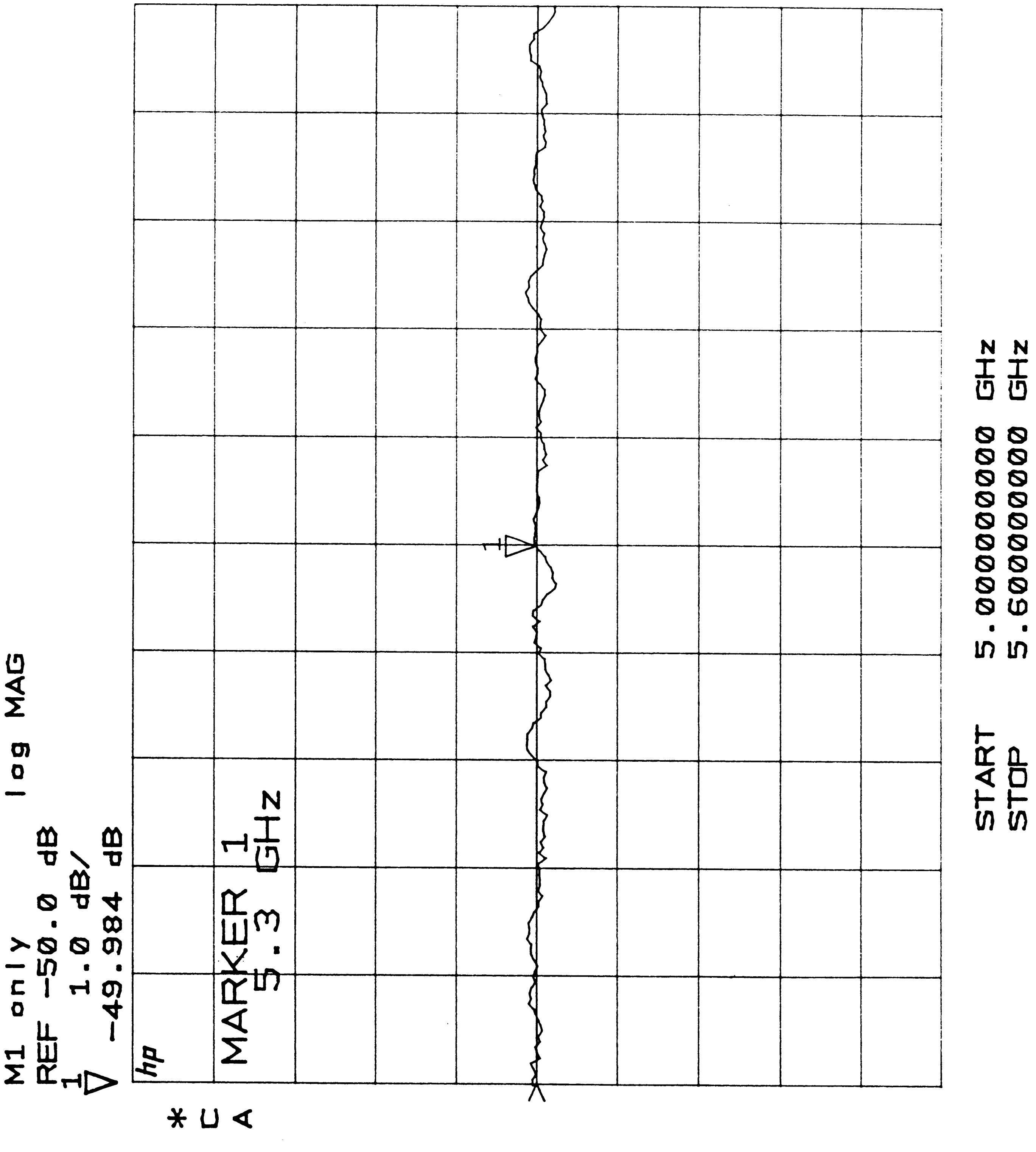


Figure 3.2: Detailed Frequency Domain Response of G_{Loop}

Figure 3.3: Frequency Domain Response of the 50 dB Attenuator



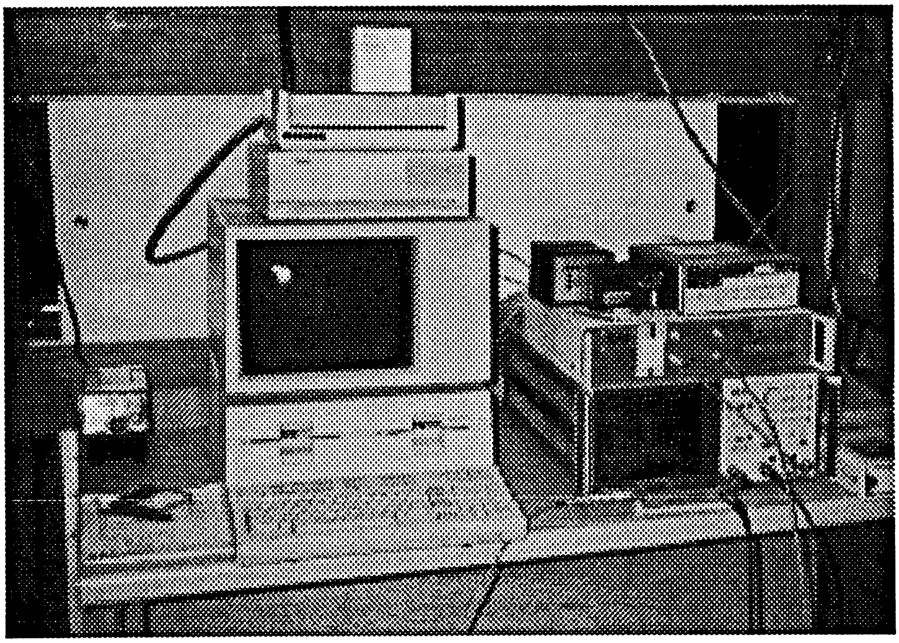


Figure 3.4: C-Band POLARSCAT Test Equipment

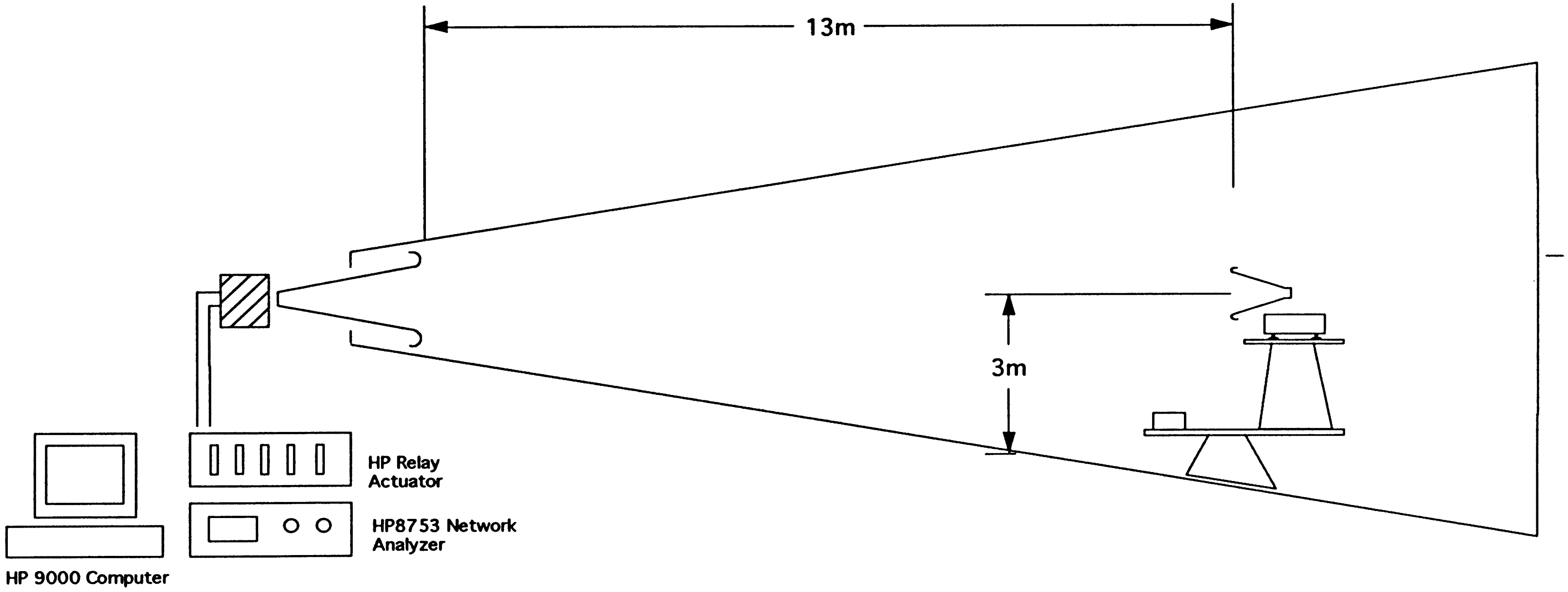
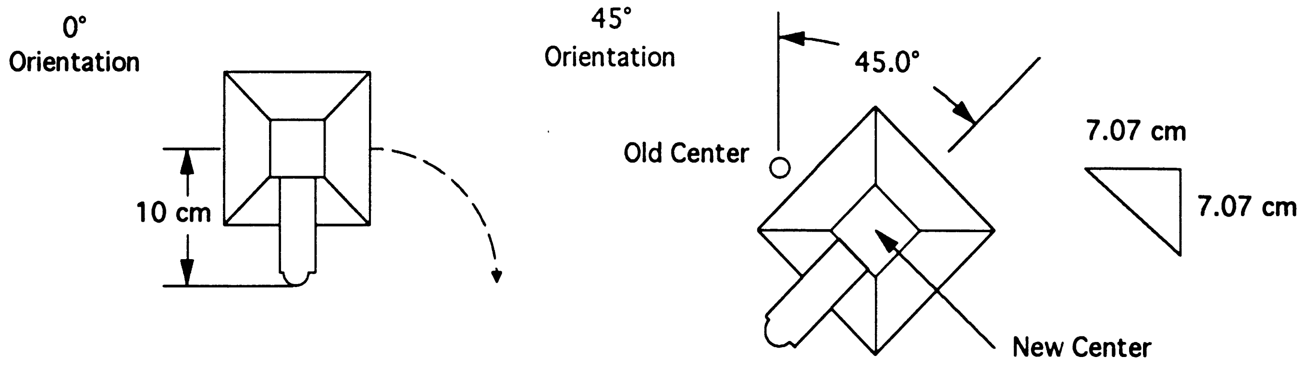


Figure 3.5: Anechoic Chamber Measurements at the University of Michigan's Radiation Laboratory



Note : SAPARC must be moved up by 7.07 cm and mounted to the left by 7.07 cm

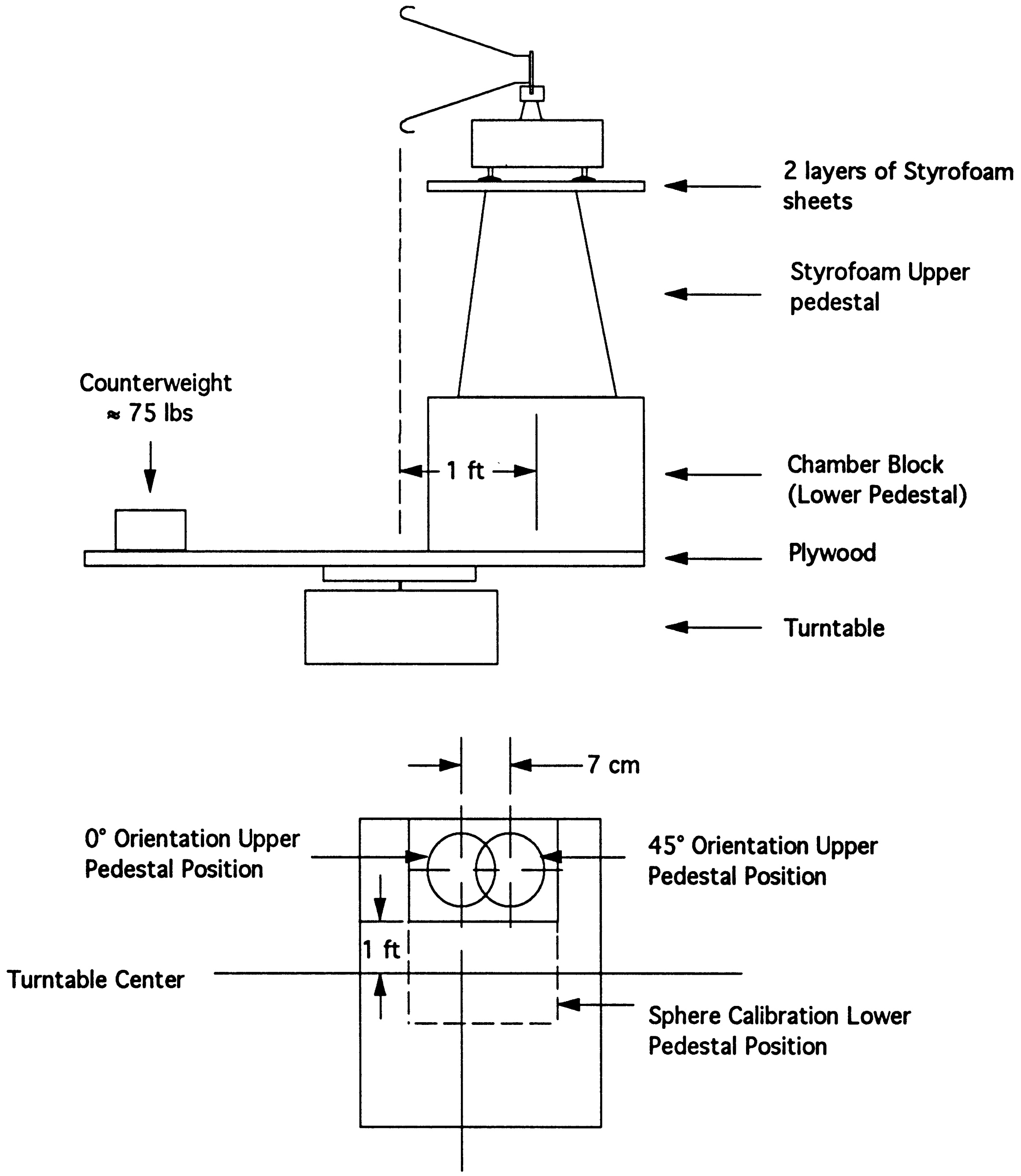


Figure 3.6: 0° and 45° Orientation Modifications

A block diagram of the measuring facility is given in Figure 3.5. Figure 3.6 shows the configurations used for the 0° and 45° Orientation tests.

3.1.1 SAPARC Time Domain Response

Section 2.3 mentioned that a SAR calibration can be enhanced by time shifting the PARC's radar response so that it appears to originate over a dark background (refer to Figure 2.8). Recall that a 200 foot delay line is incorporated into the SAPARC design to accomplish such a feat. The effect of this delay is clearly shown in Figure 3.7, the time domain response of the C-band SAPARC system. As an addendum, Tables 3.2 and 3.3 provide an identification and quantification of the five markers given in Figure 3.7.

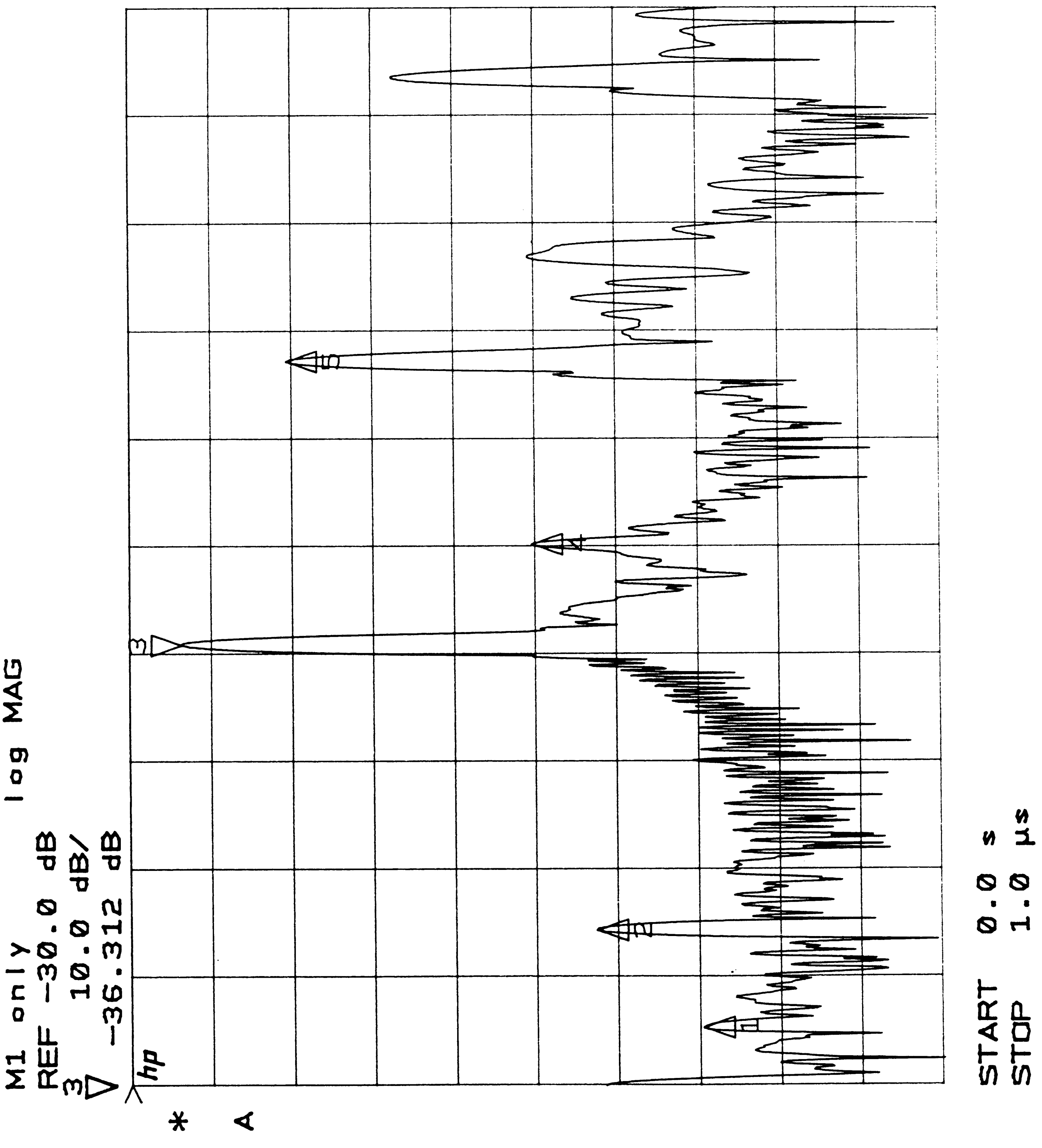
Marker	Identification
1	Leakage (i.e. cross talk) between the receive and transmit horns used in the measurement
2	Backscatter from the SAPARC's antenna and chassis (physical location of the SAPARC)
3	Primary time-delayed SAPARC response
4	First delayed multipath reflection
5	Response due to the *ringing* of the SAPARC unit

Table 3.2: Marker Identification for Figure 3.2

		Time Delay (ns)		Electrical Distance (m)	
Marker	Magnitude (dB)	Measured	Referenced to Marker 1	Measured	Referenced to Marker 1
1	-100.39	53.75	0.00	16.11	0.00
2	-87.40	143.75	90.00	43.10	26.99
3	-36.31	408.75	355.00	122.54	106.43
4	-79.59	502.50	448.75	150.65	134.54
5	-49.37	672.50	618.75	201.61	185.50

Table 3.3: Magnitude and Position Values for the Markers Shown in Figure 3.2

Figure 3.7: SAPARC Time Domain Response



The values given in Tables 3.2 and 3.3 lead to a number of important conclusions. First, the electrical length of the delay line is found to be the difference between markers 2 and 3, namely 265 ns. This, in turn, corresponds to an electrical length of 79.44 m; hence, the SAPARC's SAR response has effectively been translated by nearly 80 m (i.e. approximately 260 feet). Also note the ringing effect (marker 5) where a replica of the original SAR response is periodically repeated every 263.75 ns, or 79.1 m. The subsequent replicas are a product of the limited isolation of the OMT. During the transmission of the first SAR response, a small amount of leakage RF makes its way through the SAPARC loop where it is amplified, delayed, and re-transmitted as another SAR response. Figure 3.7 shows how each of the recurring responses will decay by approximately 13 dB; hence, this process continues until the net amount of leakage becomes negligible. When processing the imaging data, the ringing effect inherent with each SAPARC allows for easy identification and location of the calibration system, thus providing another advantage over passive calibration devices.

Marker 4 shows the delayed response of the first multipath reflection. The distances between markers 1 and 2 and markers 3 and 4 are virtually identical; therefore, it is believed that the response labeled by marker 4 corresponds to a component of the original signal which experienced multiple reflections from the SAPARC horn / chassis and the receive / transmit antennas on the radar platform. In other words, this signal originally reached the SAPARC unit where it was then reflected back towards the radar platform. Once reaching the platform, the signal was then reflected back again towards the SAPARC. Upon reaching the SAPARC for the second time, the signal was received, delayed, amplified, and re-transmitted back to the radar platform. Similar multipath signals are shown throughout Figure 3.7.

As a final point, the SAPARC provides an exceptional signal to noise ratio (SNR). The difference between the SAR response (labeled as marker 3) and the anechoic chamber's noise floor is over 70 dB. A 50 dB signal to clutter ratio (i.e. the difference between markers 2 and 3) is also shown to be quite extraordinary. These relatively large values of SNR will prove to be very beneficial for actual SAR calibrations.

3.1.2 0° Orientation Test

The 0° orientation of a SAPARC refers to the case when there is no polarization mismatch between the radar's antenna and the SAPARC antenna. In this mode, the SAPARC provides a calibration of σ_{hv}, where a received vertically polarized signal is amplified, delayed, and transmitted back to the radar with a horizontal polarization. The phasor polarizations are given in Figure 3.8.

Figure 3.9 demonstrates the measured azimuthal pattern response for this orientation. As shown, the SAPARC yields a maximum RCS response of 40 dBsm with a 15° half-power beamwidth (for the σ_{hv} case). (Note that the traditional convention of listing the target's polarimetric RCS as σ_{xy}, where x and y refer to the received and transmitted polarizations, respectively, is used.) A cross polarization isolation of 38 dB exists between σ_{hv} and σ_{vv} , σ_{hh} , thereby giving credence to the excellent cross polarization isolation performance of the horn / OMT design described in section 2.2.

As a final point, the σ_{vh} response reveals the "noise floor" inherent with this measurement. This RCS response is characterized by a 100% polarization mismatch for both the radar and SAPARC antennas, and hence the extremely low RCS response of -40 to -60 dBsm is expected.

3.1.3 45° Orientation Test

The 45° orientation is accomplished by rotating the SAPARC horn as demonstrated in Figure 3.10. Doing so allows a complete calibration of the Scattering Matrix since each transmit and receive combination, namely σ_{vv}, σ_{hh}, σ_{hv}, and σ_{vh}, yields the same RCS azimuthal pattern response with a half-power beamwidth of 15°. Figure 3.11 depicts the RCS azimuthal pattern response for the 45° orientation. Note how each trace is symmetric and virtually equal over a 40° beamwidth, as expected. Also note how the peak RCS of 34 dBsm is exactly 6 dB below the 0° orientation response of 40 dBsm. Again, this result is in excellent agreement with the theoretical expectations (the two 45° polarization mismatches, one for transmit and the second for receive, correspond to a total loss in power of $1/2 \bullet 1/2 = 1/4 = 6$ dB).

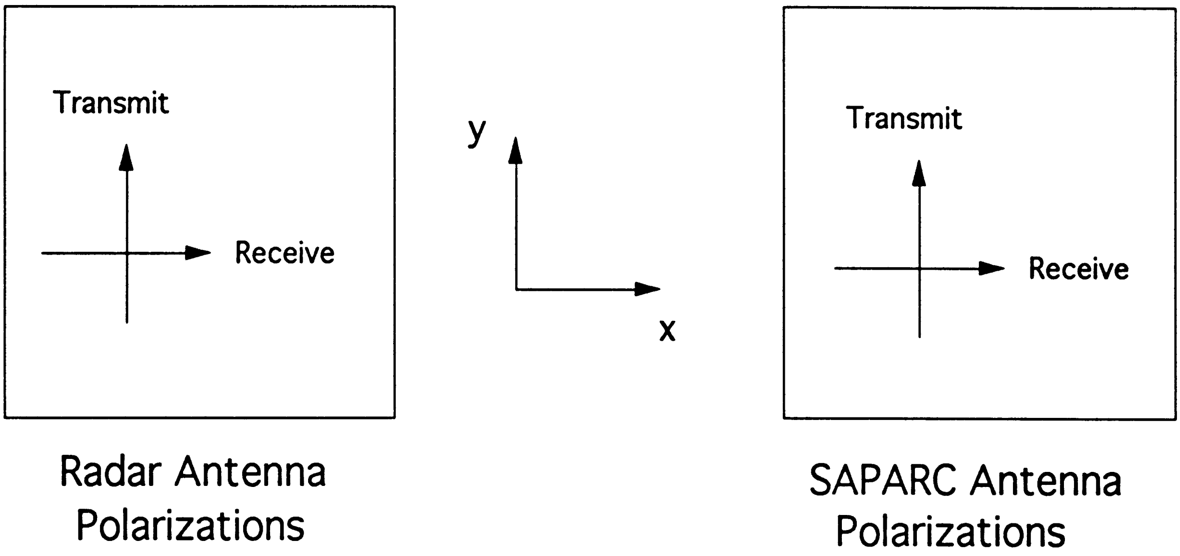
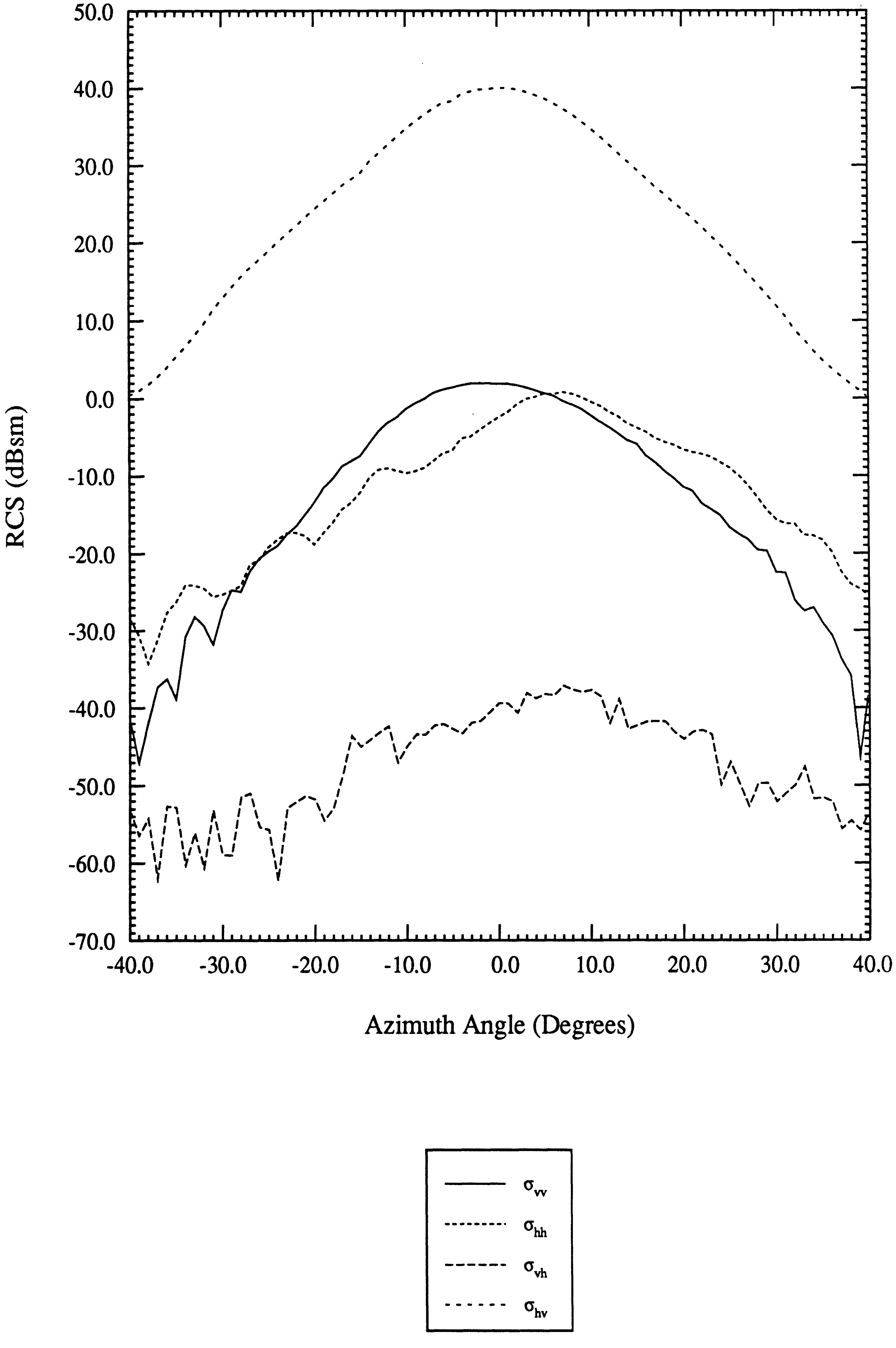
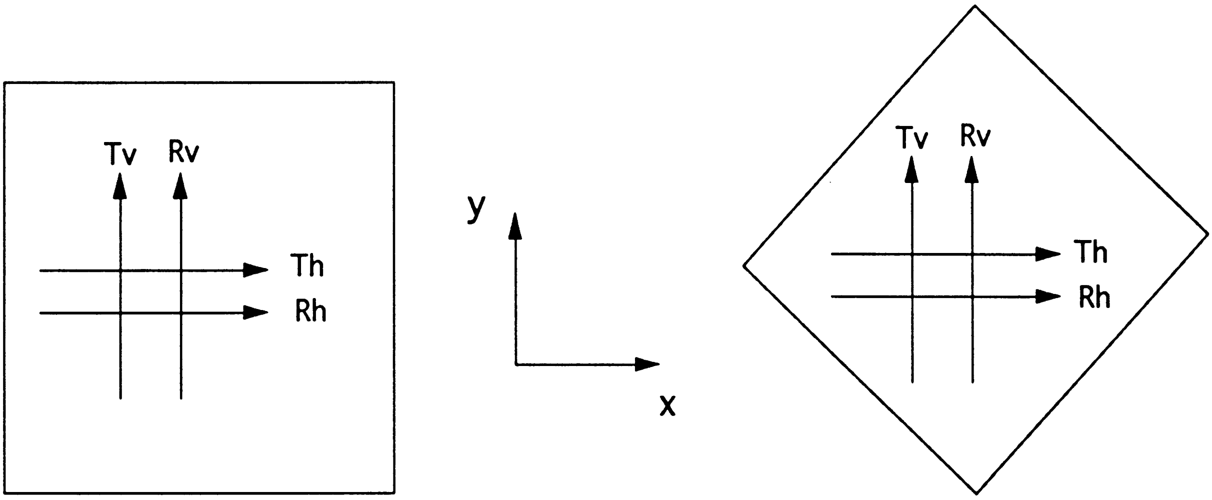


Figure 3.8: SAPARC 0° Orientation Phasor Polarizations

Figure 3.9: SAPARC 0° Orientation RCS Azimuthal Patterns (SIR-C Mode)





Radar Antenna Polarizations

SAPARC Antenna Polarizations

Tv: Transmit Vertically Polarized Wave Component
Rv: Receive Vertically Polarized Wave Component
Th: Transmit Horizontally Polarized Wave Component
Rh: Receive Horizontally Polarized Wave Component

Figure 3.10: SAPARC 45° Orientation Phasor Polarizations

Figure 3.11: SAPARC 45° Orientation RCS Azimuthal Patterns (SIR-C Mode

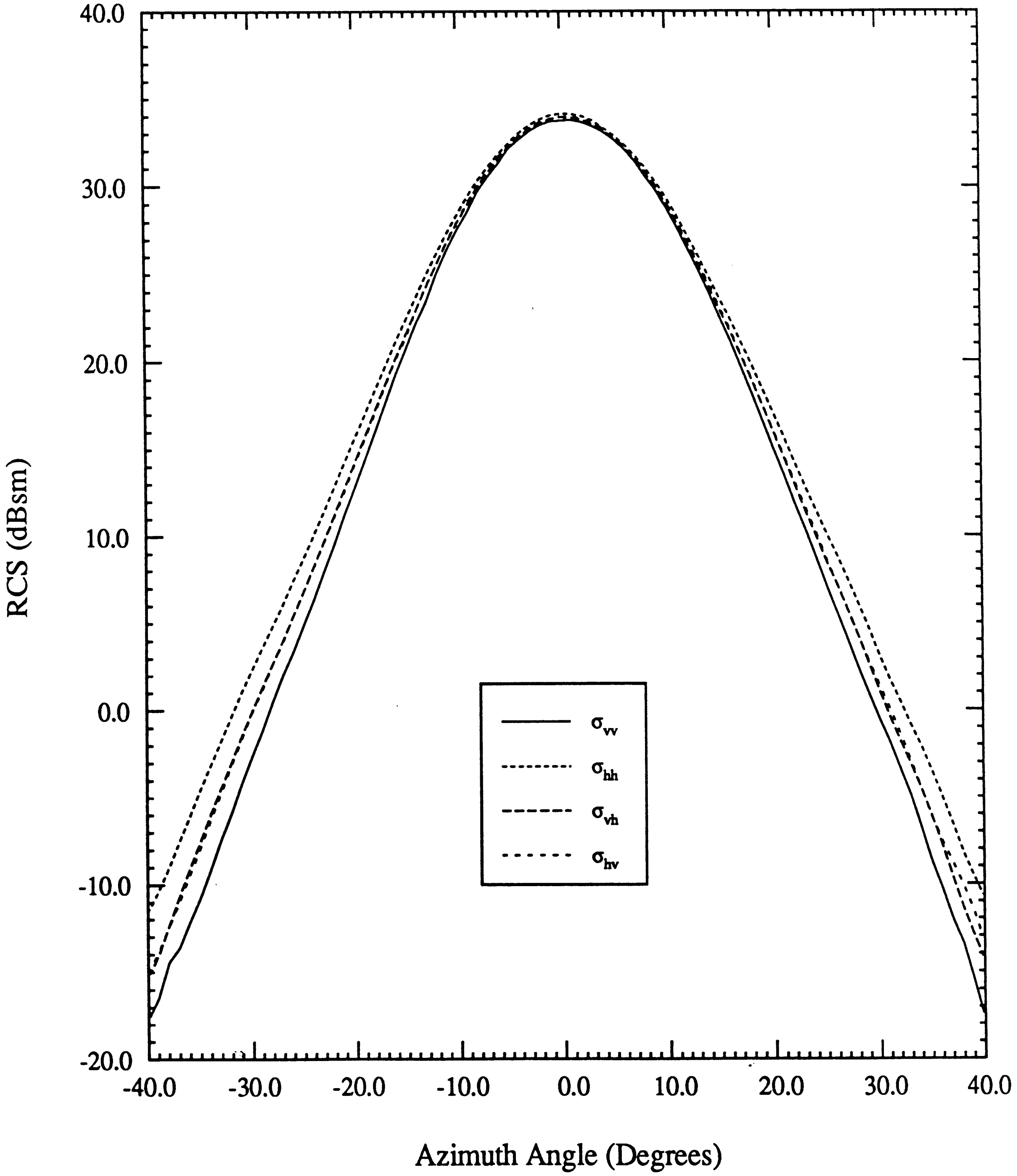
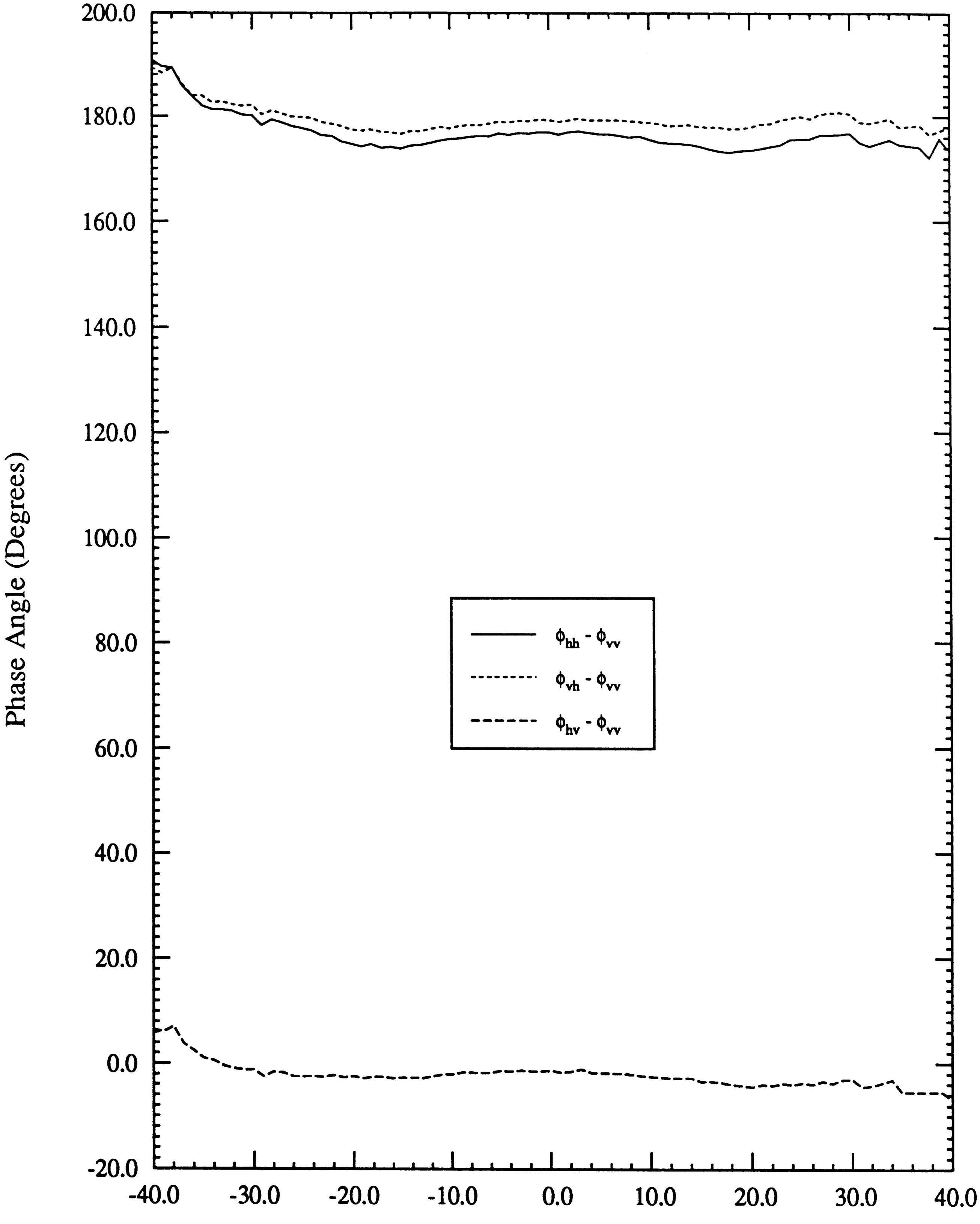


Figure 3.12: SAPARC 45° Orientation Phase Patterns (SIR-C Mode)



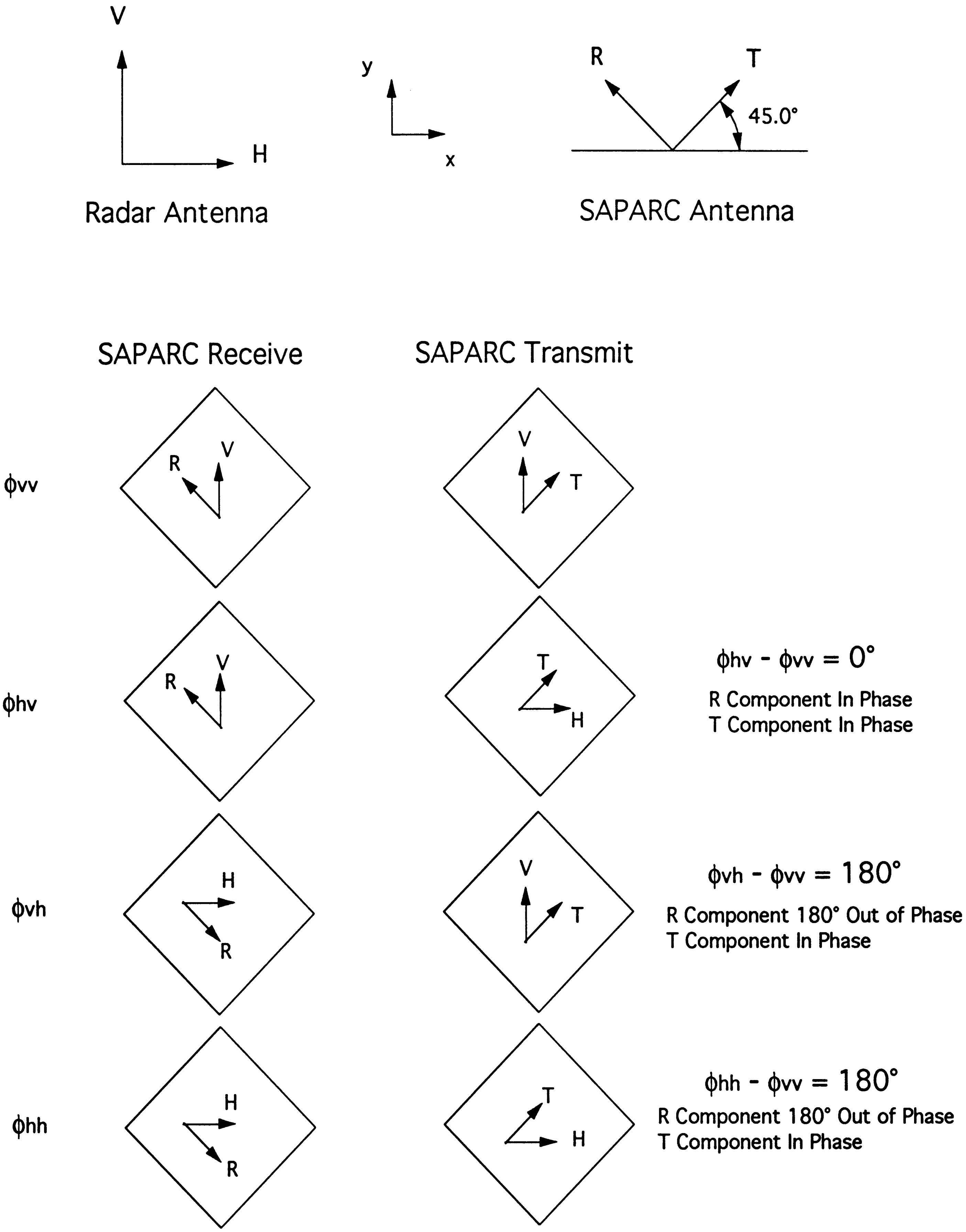


Figure 3.13: Phasor Diagrams for the 45° Orientation

The phase responses shown in Figure 3.12 are also noteworthy. Theoretically, we expect

$$\phi_{hv} - \phi_{vv} = 0°$$

$$\phi_{hh} - \phi_{vv} = \phi_{vh} - \phi_{vv} = 180°$$

over an 80° beamwidth. The phase diagrams in Figure 3.13 help to explain these results.

Section 2.2 referred to the drawbacks encountered when using a two-antenna PARC system; more specifically, these problems include pattern asymmetry and ripples in the phase and magnitude responses. Figure 2.3 is an example of one two-antenna system tested by Sarabandi and Oh [1]. By comparing Figure 2.3 with those in Figures 3.11 and 3.12, one can easily see the notable SAPARC improvements in magnitude and phase performance.

3.3 Thermal Gain Testing

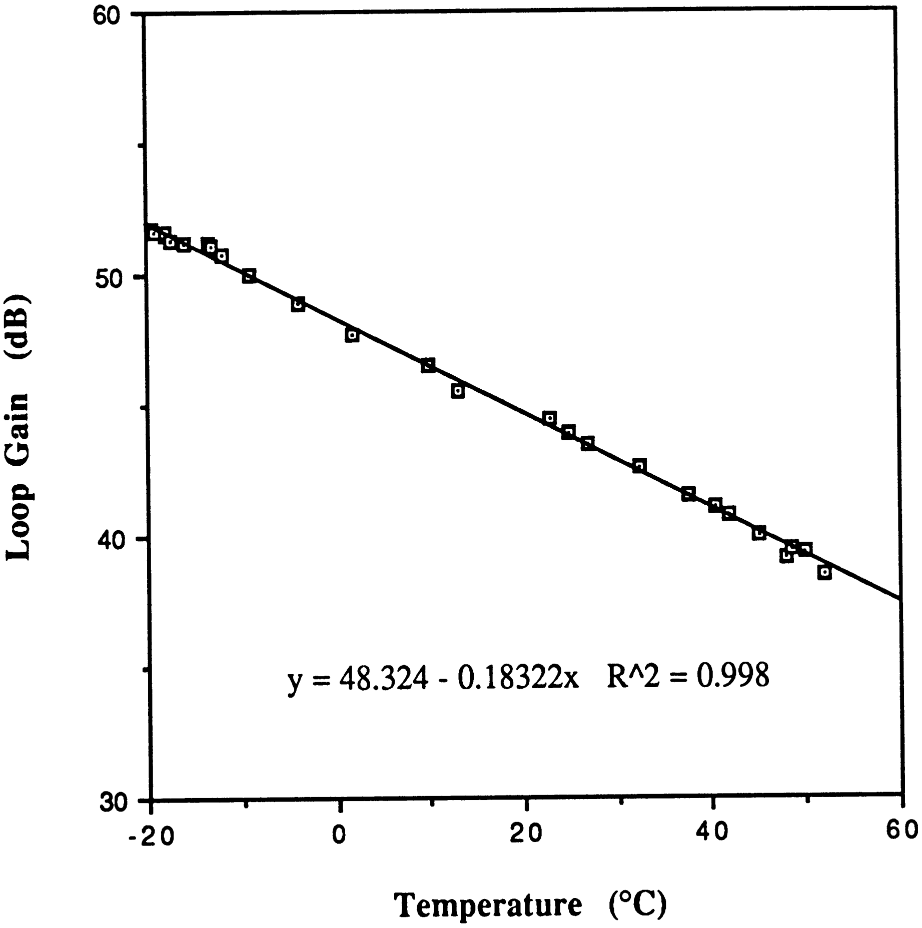
Section 2.5 alluded to the fact that PARC's are susceptible to gain variations due to changes in the ambient temperature. Countering these thermal gain variations is formidable task; therefore, it is much easier to compensate for the changes in the SAPARC's RCS by mapping the G_{Loop} dependency on ambient temperature.

The thermal gain tests were performed at ERIM (Environmental Research Institute of Michigan) where a temperature-controllable refrigerator could be obtained. Figure 3.14 reflects the results of the experiment whereby G_{Loop} is plotted over a temperature range of -20°C to 50°C.

The user of the device must remember, however, that component aging may alter the overall performance of G_{Loop}. Therefore, periodic calibrations of G_{Loop} vs. temperature is recommended.

As a final note, special precautions must be taken when operating the SAPARC in cold weather scenarios. Currently, the system is configured to provide the

Figure 3.14: C-Band SAPARC Thermal Gain Variations



maximum allowable RCS for temperatures of 20°C or greater (see section 2.3.3). However, Figure 3.14 clearly shows how G_{Loop} can increase by as much as 7 dB for temperatures below 20°C. Therefore, the user is encouraged to use the JPL AIRSAR mode of operation whenever cold operating temperatures are anticipated. Recall that the AIRSAR mode uses a microwave switch to insert 7 dB of attenuation into the loop, thereby providing the needed cold weather protection against feedback oscillations.

3.4 Field Deployment Conditions

3.4.1 Battery Capacity

The SAPARC units developed through this project require two 12V, 7 Amp•H lead acid batteries. The power demands on Supply 2 (see Appendix C, pg. 1) is given as follows:

SAPARC Operating Condition	Current Draw	Power Demand
Timing Circuitry On	0.37 A	4.44 W
System Active	1.09 A	13.08 W
System Active with LED Display On	1.20 A	14.40 W

Table 3.4: Power Demands on Supply 2

Figure 3.15 (1 A case) depicts the capacity performance of the Yuasa 12V battery used in the SAPARC design. Figure 3.16 shows the results for the full load case (i.e. a current draw of 1.2 A). Under full load conditions, the SAPARC can operate (at temperatures above 20°C) for up to 5.5 hours; longer operating times are achievable when using the Activation Timers.

Figure 3.17 shows the marked decrease in capacity during cold weather operation. During this test, the Yuasa 12V battery was subjected to a temperature of -10°C while providing a current of 1.2 amps. Under these conditions, the SAPARC's full load operating time is reduced by one hour; therefore, the user must take special precautions when planning to operate the SAPARC in cold climates.

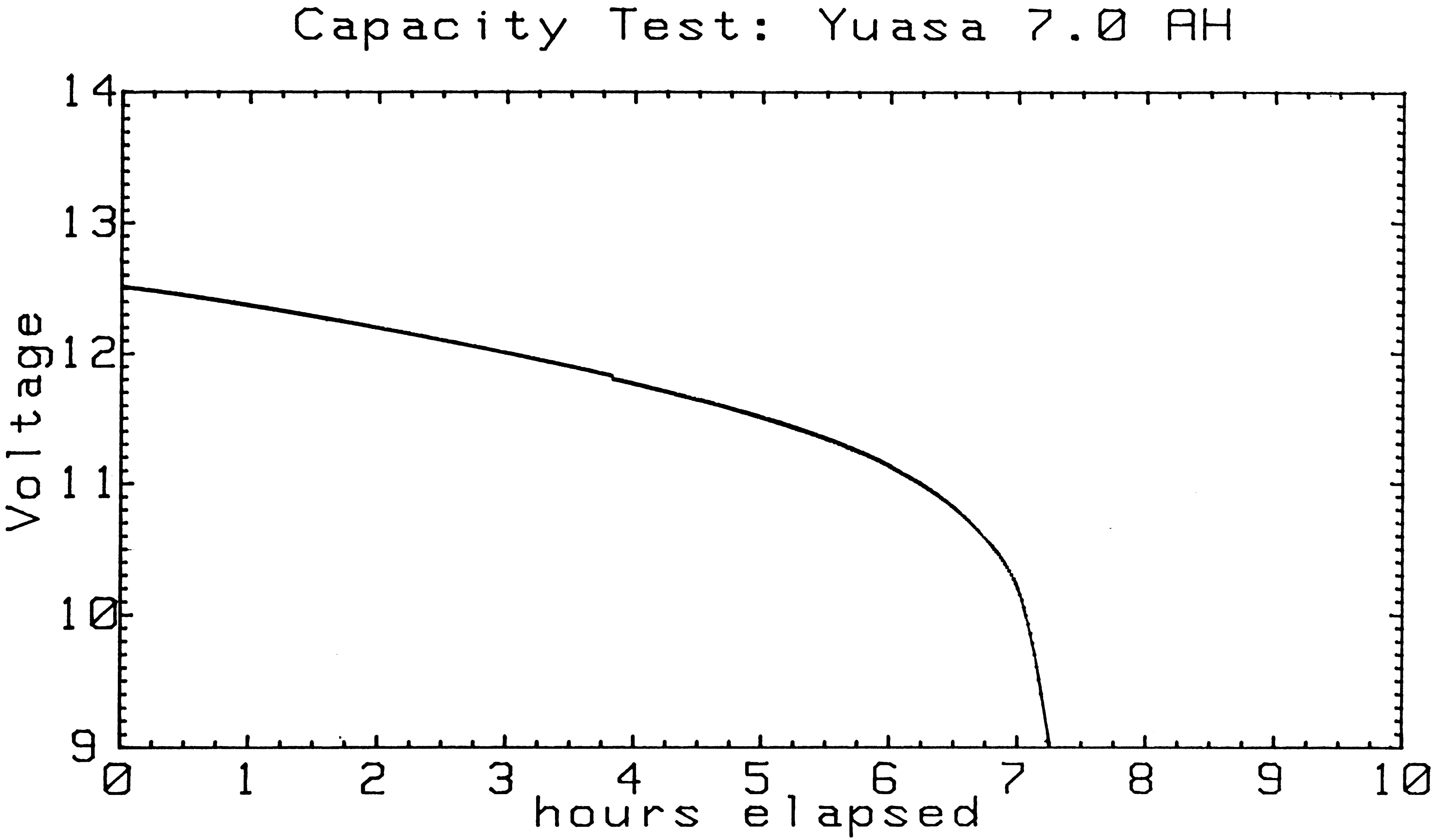


Figure 3.15: Battery Capacity Test (1.0A Load, 22°C)

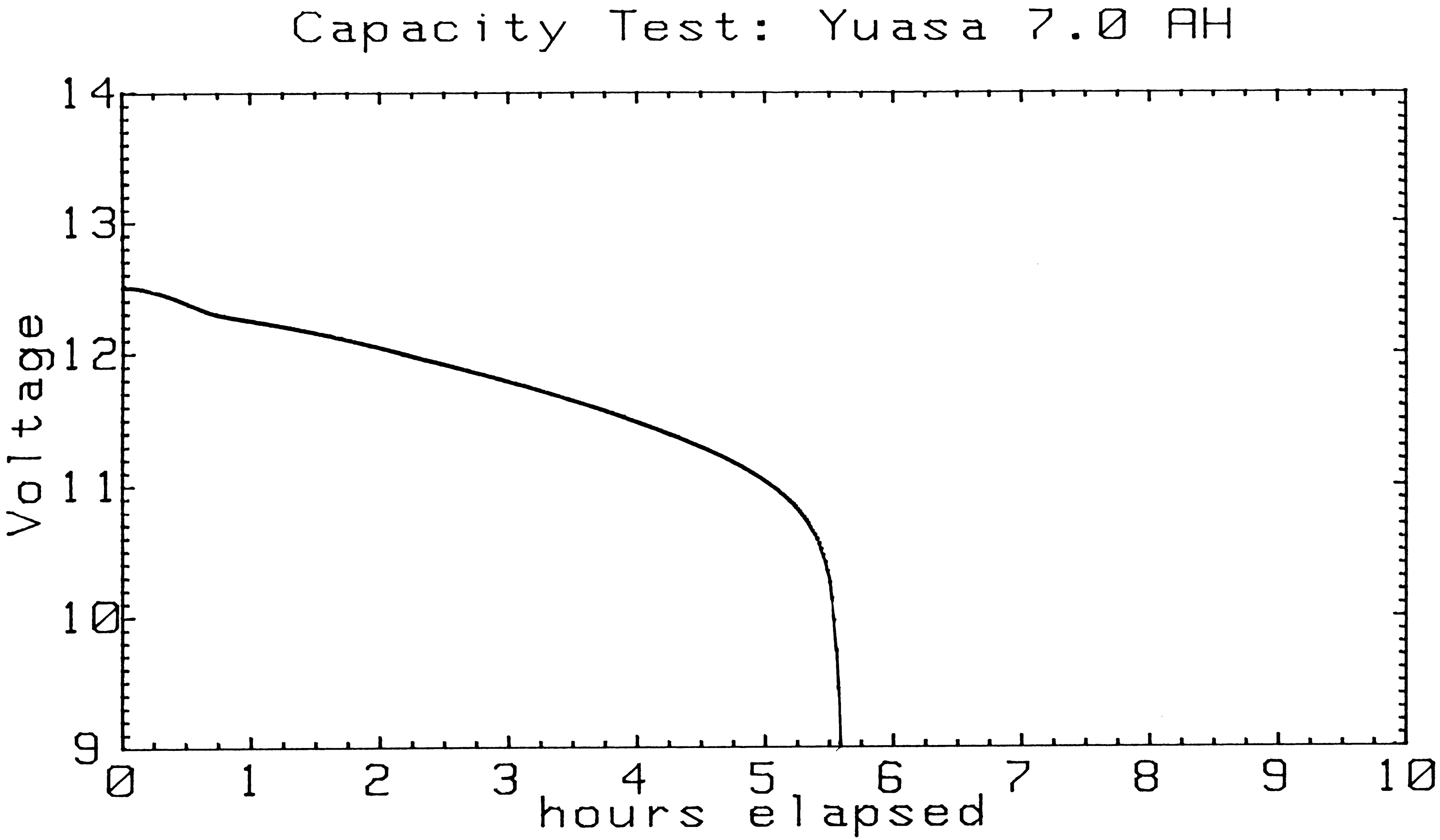


Figure 3.16: Battery Capacity Test (1.2A Load, 22°C)

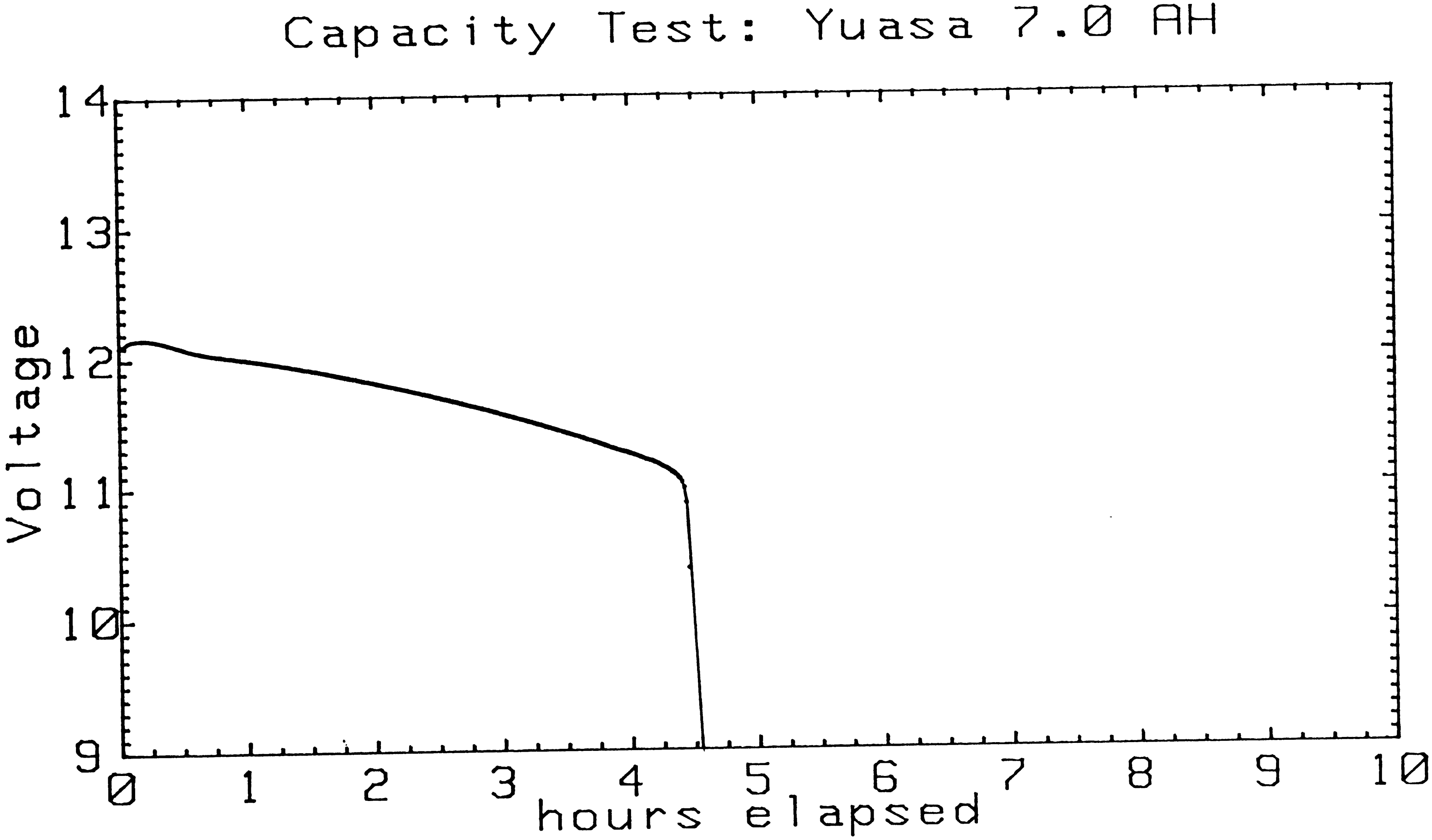


Figure 3.17: Cold Weather Battery Capacity Test (1.2A Load, -10°C)

3.4.2 All-Weather Performance

Ideally, the SAPARC can be used in all types of weather; of course, there are a number of practical limitations which concern the aperture on the horn antenna. Obviously, rain and snow can accumulate inside the horn / OMT, and thus the calibration would be ruined. In order to compensate for this occurrence, a polyethylene film (e.g. Saran Wrap) was placed over the aperture to serve as a radome. This film was very effective in keeping water out of the horn / OMT combination; however, an accumulation of water droplets on the radome eventually lead to a feedback oscillation scenario (see Sections 2.3.2.2 and 2.4.2). Therefore, applying a thin-filmed polyethylene radome is suggested for weatherproofing the horn, OMT, and waveguide adapters. Yet, do not expect the SAPARC to operate correctly in adverse weather conditions.

As a sidenote, if the rainfall ceases and the radome is allowed to dry, the SAPARC will "break-out" of its oscillation mode and return to its normal operating condition. The only notable change is that the detection circuitry will be triggered prematurely. Finally, the chassis of the SAPARC unit should be shrouded with a rain tarp to prevent excess exposure to the elements.

CHAPTER IV

CONCLUDING REMARKS

The report outlines the design and performance characteristics of the first single antenna polarimetric active radar calibrator (SAPARC) prototype developed for NASA's SIR-C mission at the University of Michigan's Radiation Laboratory. In addition to this specific unit, a second C-band and two L-band versions are currently being constructed as part of a continuation of this project.

This first C-band prototype possesses a nominal RCS of 34 dBsm with a 3 dB beamwidth of 15°. One of its best attributes, however, is the fact that it can outperform conventional PARCs though its implementation of a single dual-polarized antenna. More specifically, the pattern asymmetry and phase and magnitude ripples are eliminated through the use of this design.

In addition to these RF characteristics, the prototype is also noteworthy in that it provides a number of features which accommodate prolonged operation intervals and useful system status updates. In the future, subsequent modifications to this basic design will hopefully lead to more accurate and convenient calibrations of SAR platforms.

APPENDIX A

OMT SPECIFICATIONS

USTOMER Univ of Mich
USTOMER P.O. NO. N73622
ESTED BY SLH
Rwfy Chn

PART NO.
ATL. REF. NO. 7756
DATE 2/20/92

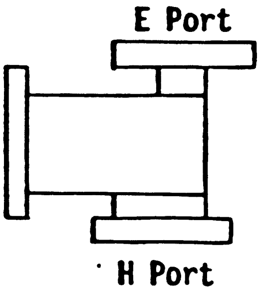
TEST RESULTS

ORTHOGONAL MODE JUNCTION

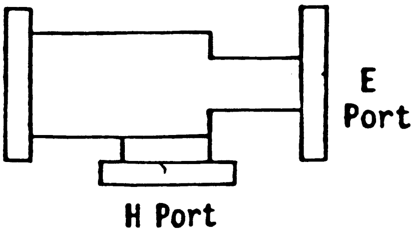
catalog number: [illegible]

erial umber	frequency in MHz		VSWR H port	VSWR E port	ISOLA-TION
42	5200	A	1.10	<1.05	50+
		B	1.11	1.06	|
		C	1.06	<1.05	|
		D	<1.05	1.06	|
	5900	E	1.06	<1.05	|
		A			
		B			
		C			
		D			
		E			
		A			
		B			
		C			
		D			
		E			
		A			
		B			
		C			
		D			
		E			
		A			
		B			
		C			
		D			
		E			
cification limits			1.15 maximum	1.15 maximum	50 dB minimum

serial number	frequency in MHz		VSWR H port	VSWR E port	ISOLA-TION
		A			
		B			
		C			
		D			
		E			
		A			
		B			
		C			
		D			
		E			
		A			
		B			
		C			
		D			
		E			
		A			
		B			
		C			
		D			
		E			
		A			
		B			
		C			
		D			
		E			



Style 1



Style 2

E-145

TLANTIC MICROWAVE
CORPORATION
OUTE 117 • BOLTON, MASS. 01740
EL (617) 779-5525 • TWX 710-392-1585

ATLANTIC

5-18-88 TR-107 REV. A

CUSTOMER COPY ☐
FILE COPY ☐

APPENDIX B

AMPLIFIER SPECIFICATIONS

MITEQ

100 Davids Drive, Hauppauge, N.Y. 11788-2034

TEL: (516) 436-7400
TELEX: 6718148
FAX: 516-436-7430

PROJECT No: P38858
MODEL No: AFS5-05000560-60-8P-5
SERIAL No: 249372
CUSTOMER: UNIVERSITY OF MICHIGAN
P.O. No: F33091

IMPORTANT - MUST USE HEAT SINK IF CASE TEMPERATURE EXCEEDS 70°C

SPECIFICATIONS AT +23° C:

FREQUENCY:	5.0 to 5.6 GHz	OUTPUT POWER @ 1dB GAIN COMPRESSION:	+8 dBm
MIN. GAIN:	40 dB	VOLTAGE:	+15 VOLTS
MAX.GAIN FLATNESS:	+/- 1 dB	MEASURED CURRENT:	150 mA
MAX. VSWR INPUT:	2 :1	MAX. NOISE FIGURE:	6.0 dB
MAX. VSWR OUTPUT:	2 :1	HOUSING No:	113110

NOTE: TEST DATA TAKEN WITH CASE TEMP. OF +23°C

FREQUENCY (GHz)	GAIN (dB)	VSWR		NOISE FIGURE (dB)	OUTPUT POWER (dBm) (@ 1dB GAIN COMPRESSION)
		IN	OUT		
5.0	44.0	1.28	1.71	1.30	+14
5.1	44.5	1.31	1.69	1.26	+14
5.2	44.6	1.32	1.65	1.39	+15
5.3	44.4	1.34	1.56	1.45	+15.5
5.4	44.2	1.35	1.50	1.61	+15
5.5	43.4	1.38	1.38	1.39	+16
5.6	42.6	1.36	1.29	1.46	+16

TESTED BY: Donald Maurice
(DONALD MAURICE)

DATE: 06/02/92

MITEQ

Project P38859

Internal Transfer

Customer P/N

Model AMF-5B-5056-29P

Serial No. 249167

SPECIFICATIONS

Frequency (GHz)	5.0-5.6 Ghz	Output Power (dBm) at 1 dB Compression	+29.0 dbm min
Gain (dB)	45.0 db min	Voltage	+15 VDC
Gain Flatness (dB)	±1.5 db max	IT Measured Current (mA)	826 mA
VSWR Input/Output	2.00:1 max		
Noise Figure (dB)	4.0 db max		
Noise Temp °K	—		

TEST DATA +25°C

Frequency (GHz)	Gain (dB)	VSWR In	VSWR Out	N. F. dB	Noise temp °K	Po at 1dB comp.	Po hard satu.
5.0	see	1.67:1	1.38:1	2.6	—	+30.7	—
5.3	attached	max	max	2.8	—	+30.7	—
5.6	Data	↓	↓	3.0	—	+30.7	—

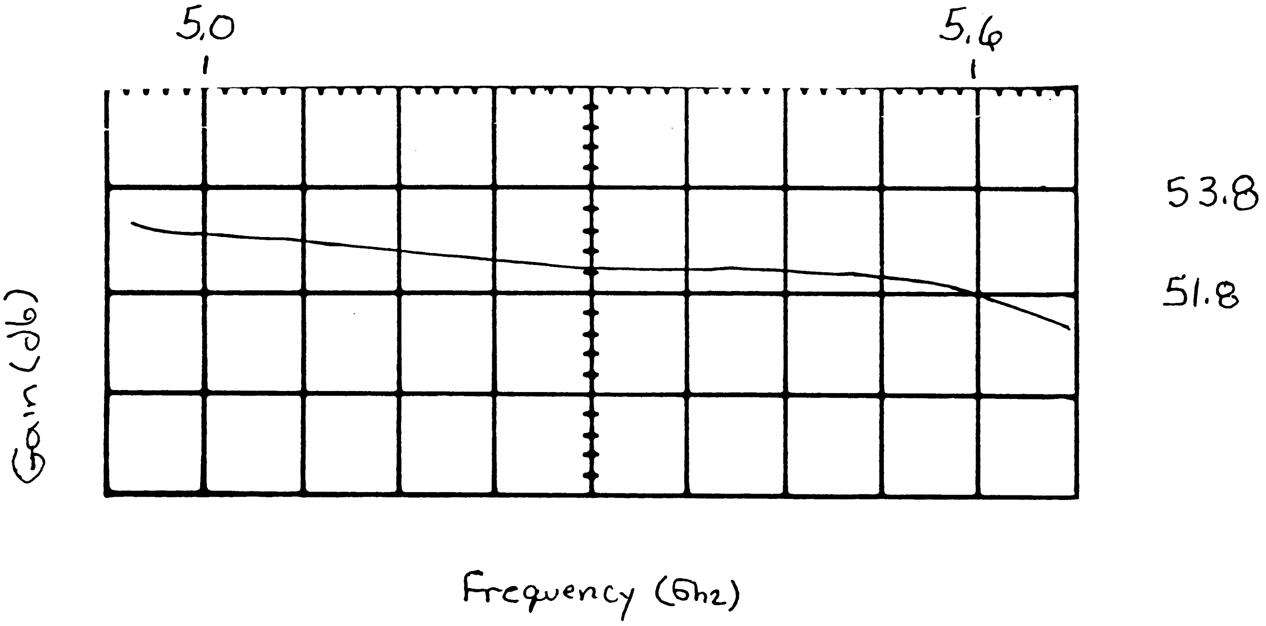
Tested By

Date 4/6/92

MITEQ

Project P38859
Internal Transfer
Customer P/N
Model AMF-5B-5056-29P
Serial No. 249167

GAIN BANDWIDTH



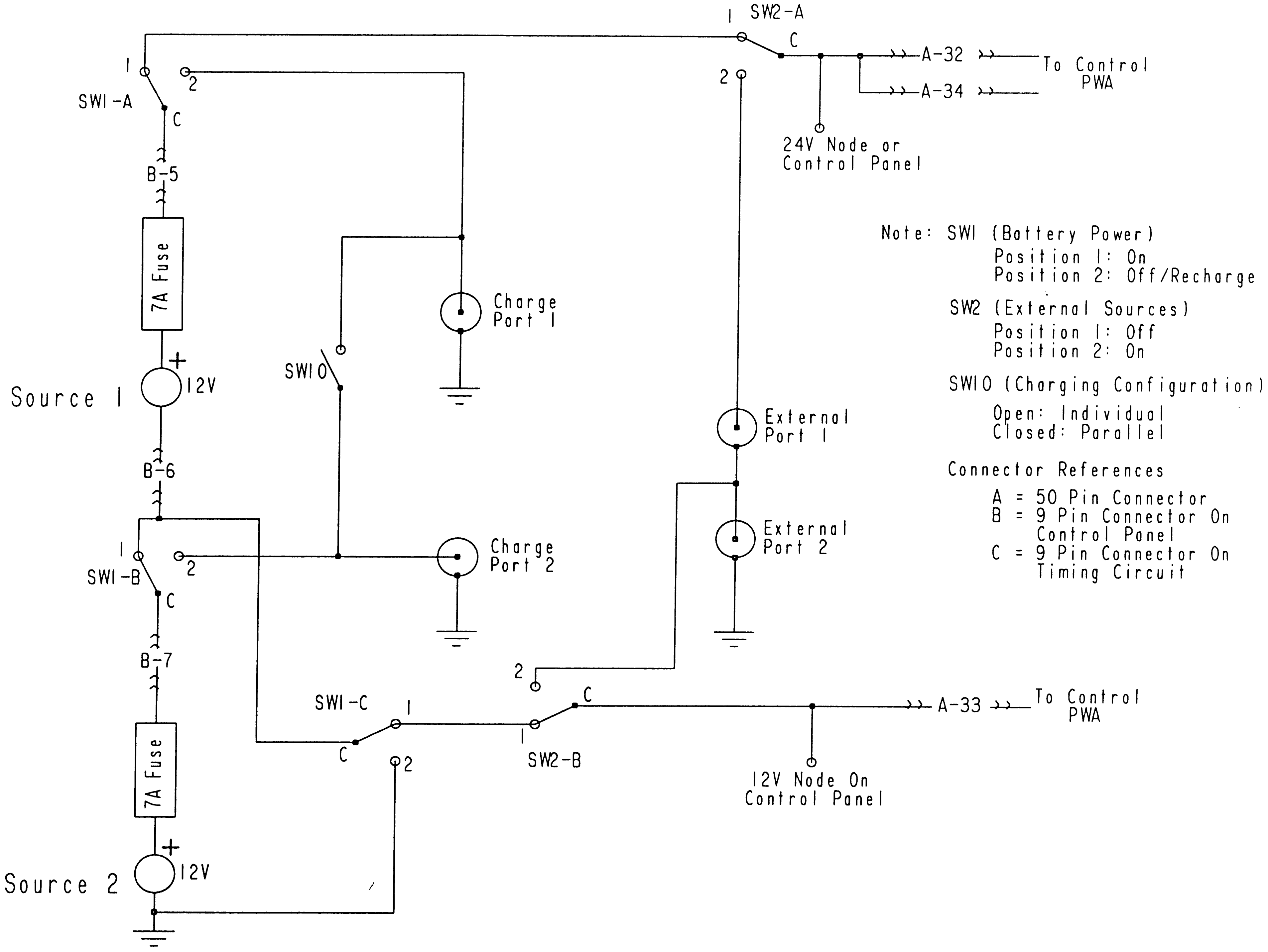
Tested By
Date 4/6/92

MITEQ INC. • 100 Davids Drive • Hauppauge, New York 11788-2086 • Tel. (516)436-7400

APPENDIX C

CONTROL CIRCUITRY SCHEMATICS

Control Panel Power Connections



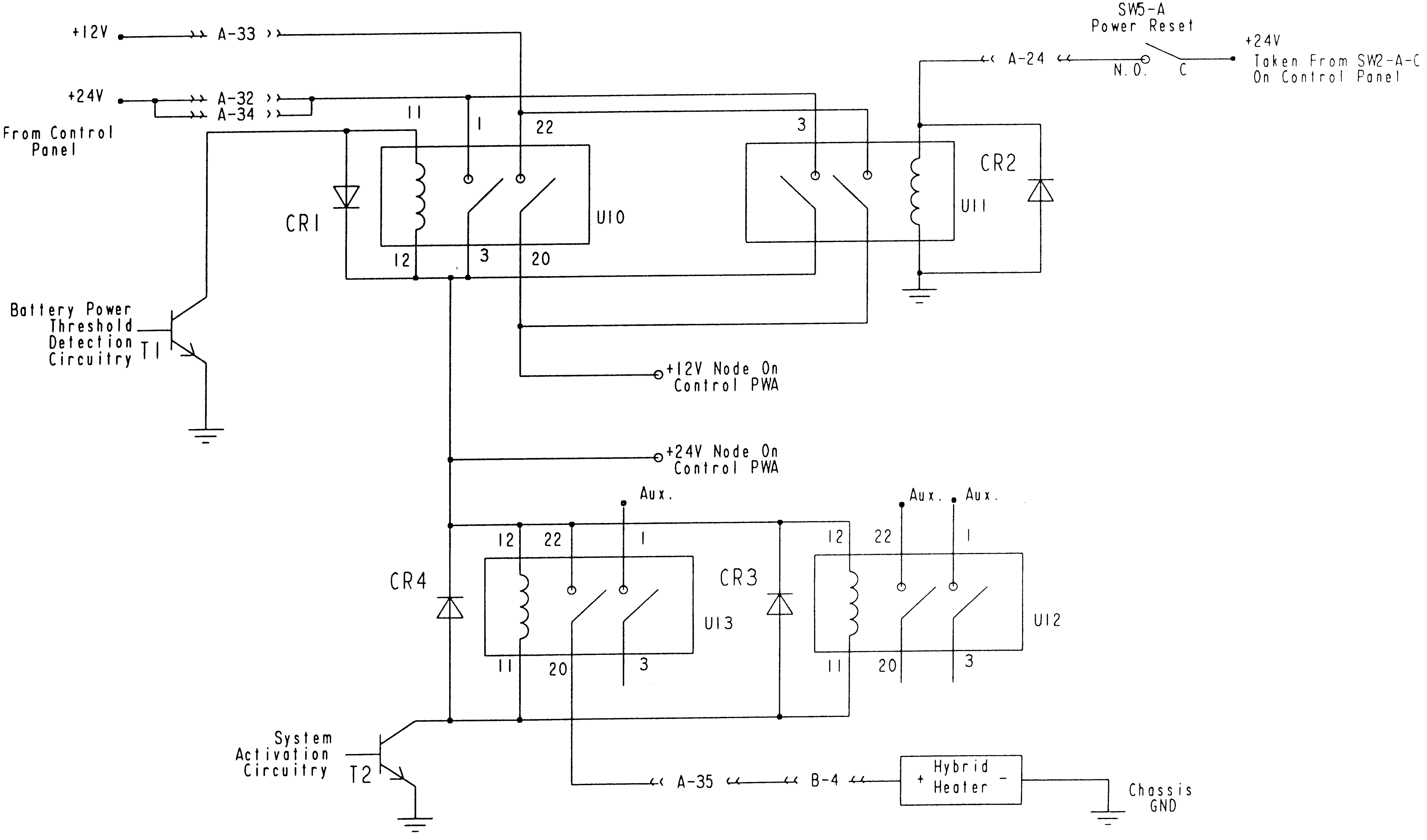
Note: SW1 (Battery Power)
Position 1: On
Position 2: Off/Recharge

SW2 (External Sources)
Position 1: Off
Position 2: On

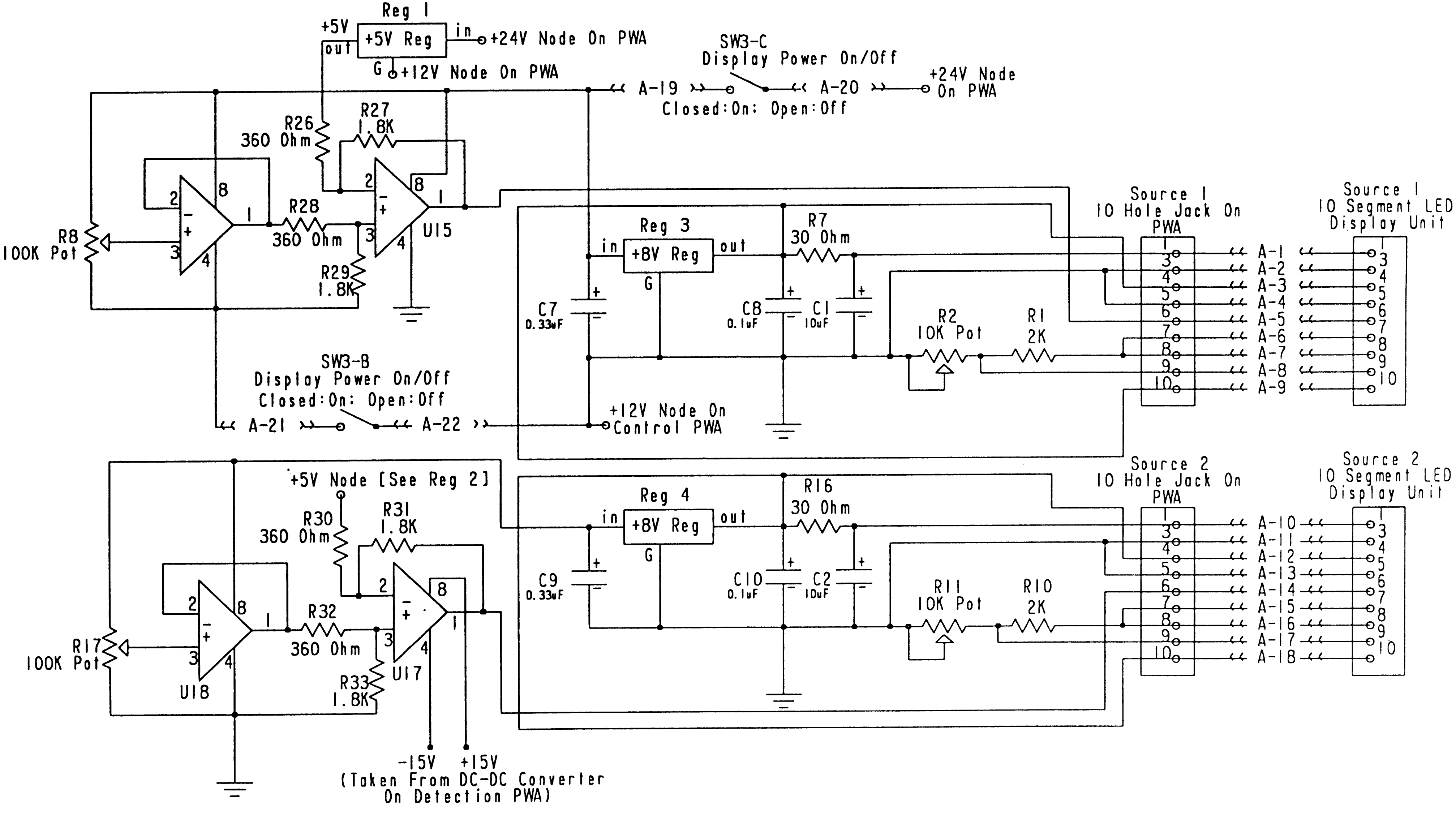
SW10 (Charging Configuration)
Open: Individual
Closed: Parallel

Connector References
A = 50 Pin Connector
B = 9 Pin Connector On Control Panel
C = 9 Pin Connector On Timing Circuit

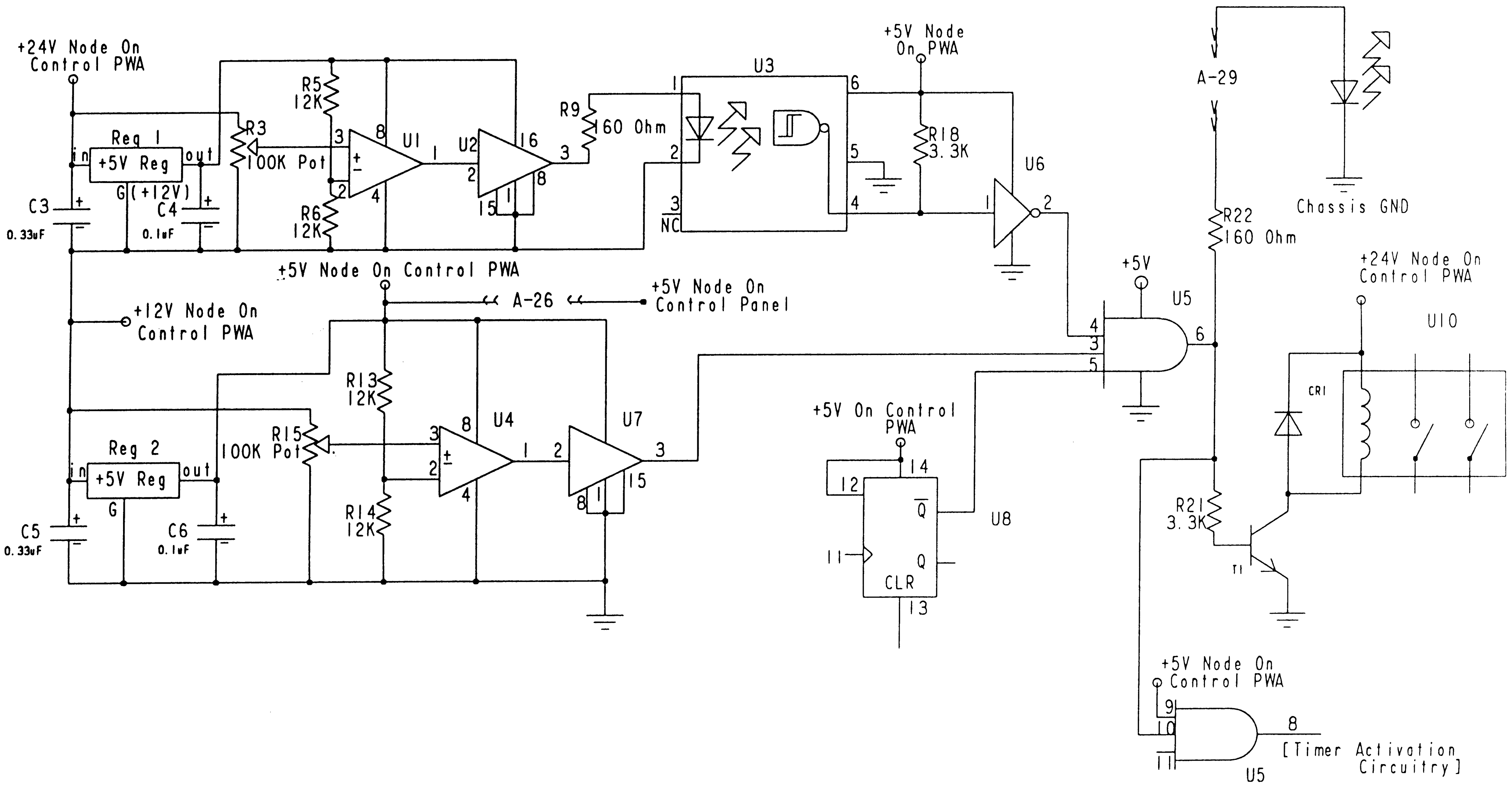
Relay Circuitry



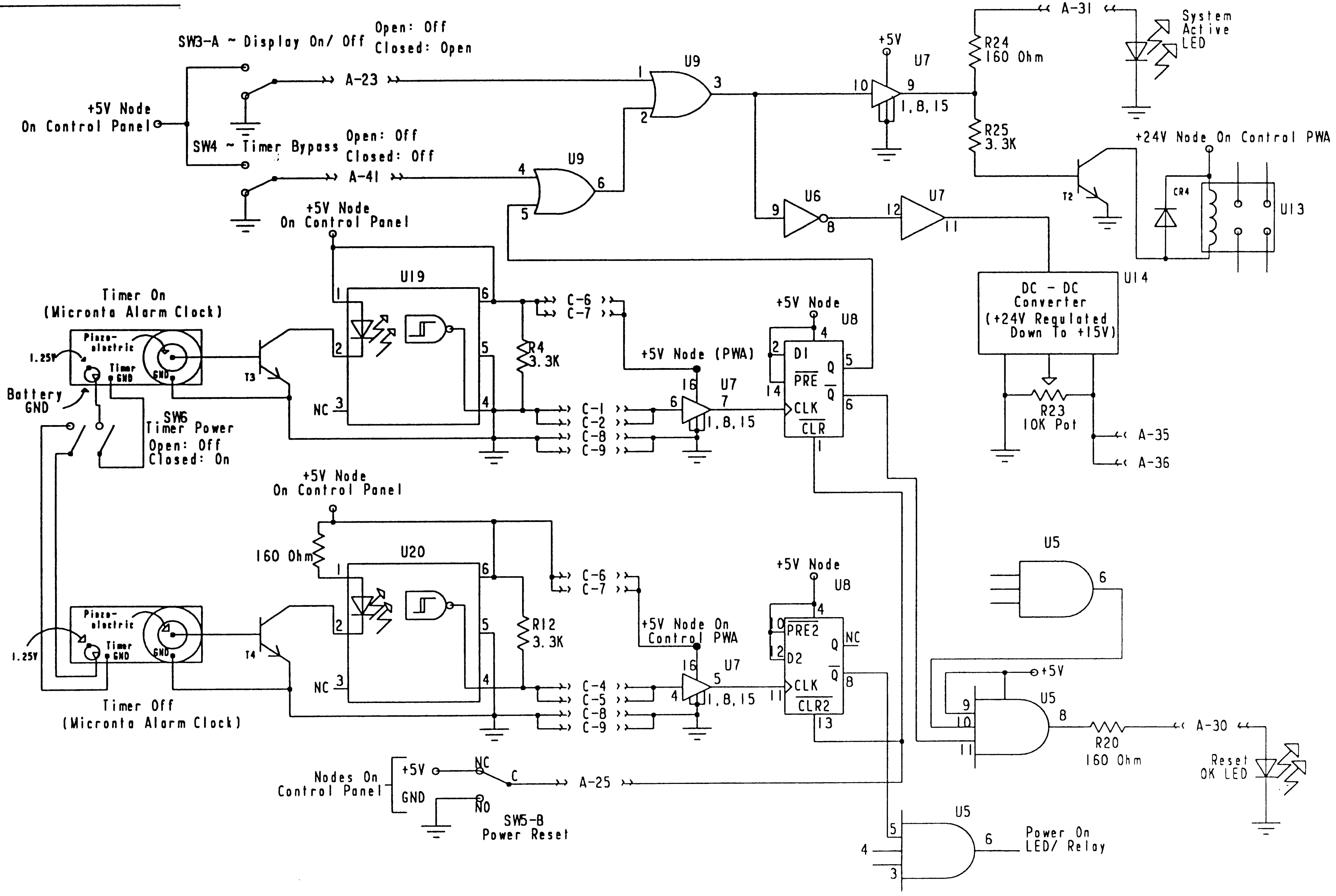
Battery Power Threshold Detection Circuitry



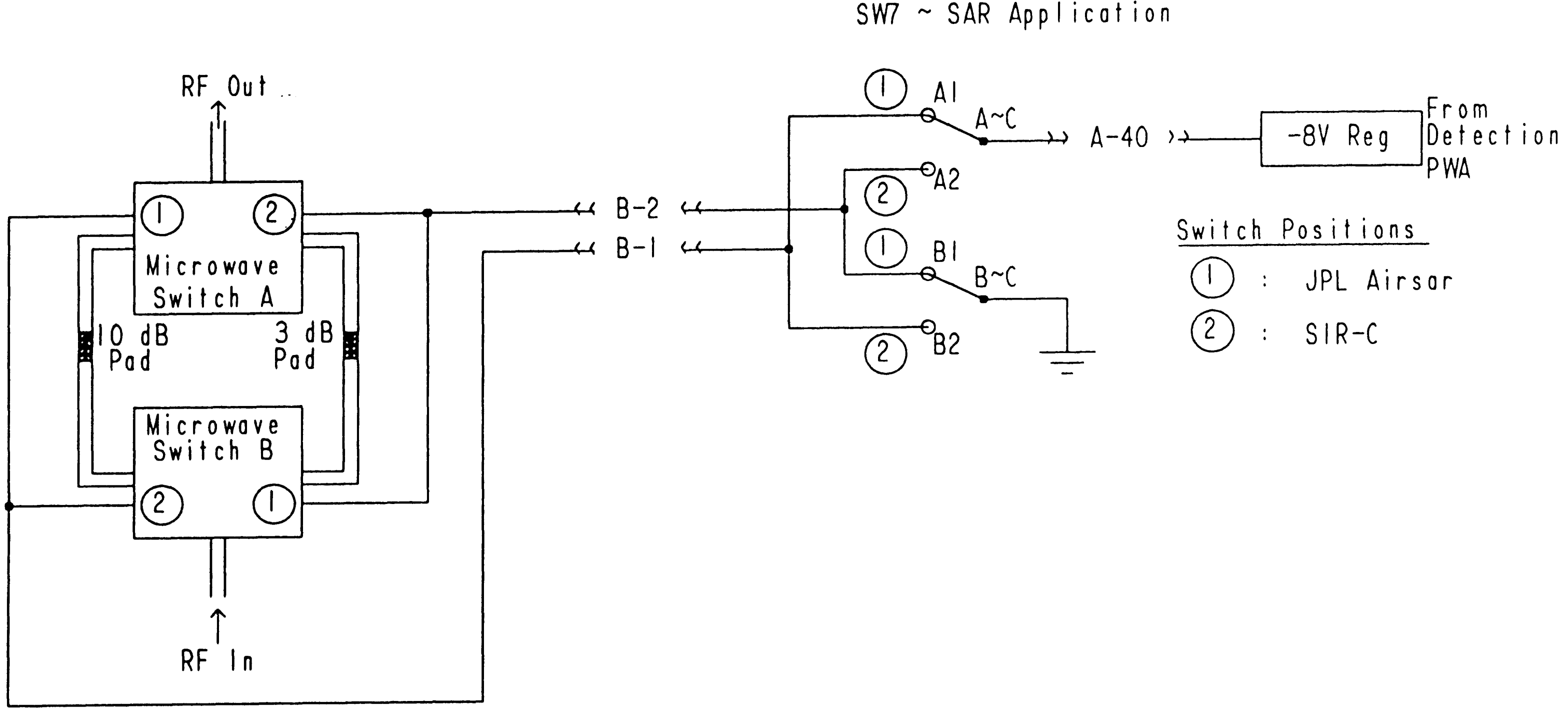
Battery Power Threshold Detection Circuitry - Part 2



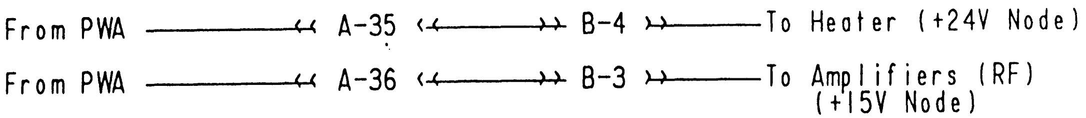
Timer Activation Circuitry



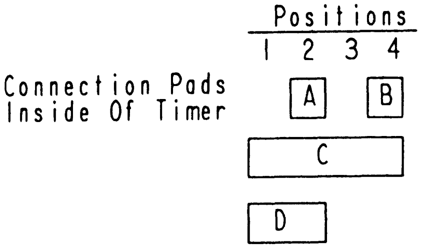
Microwave Switch Connections



RF Amps/ Heater Connections

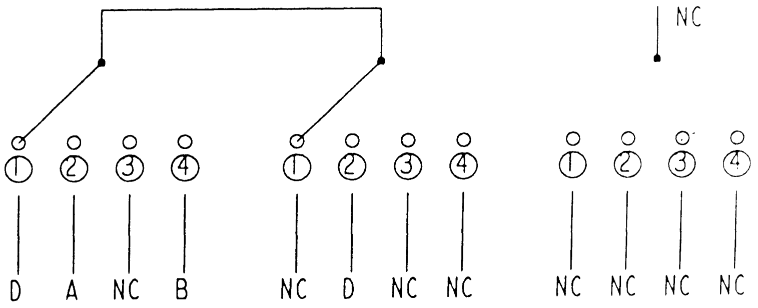


Timer Switch Replacement

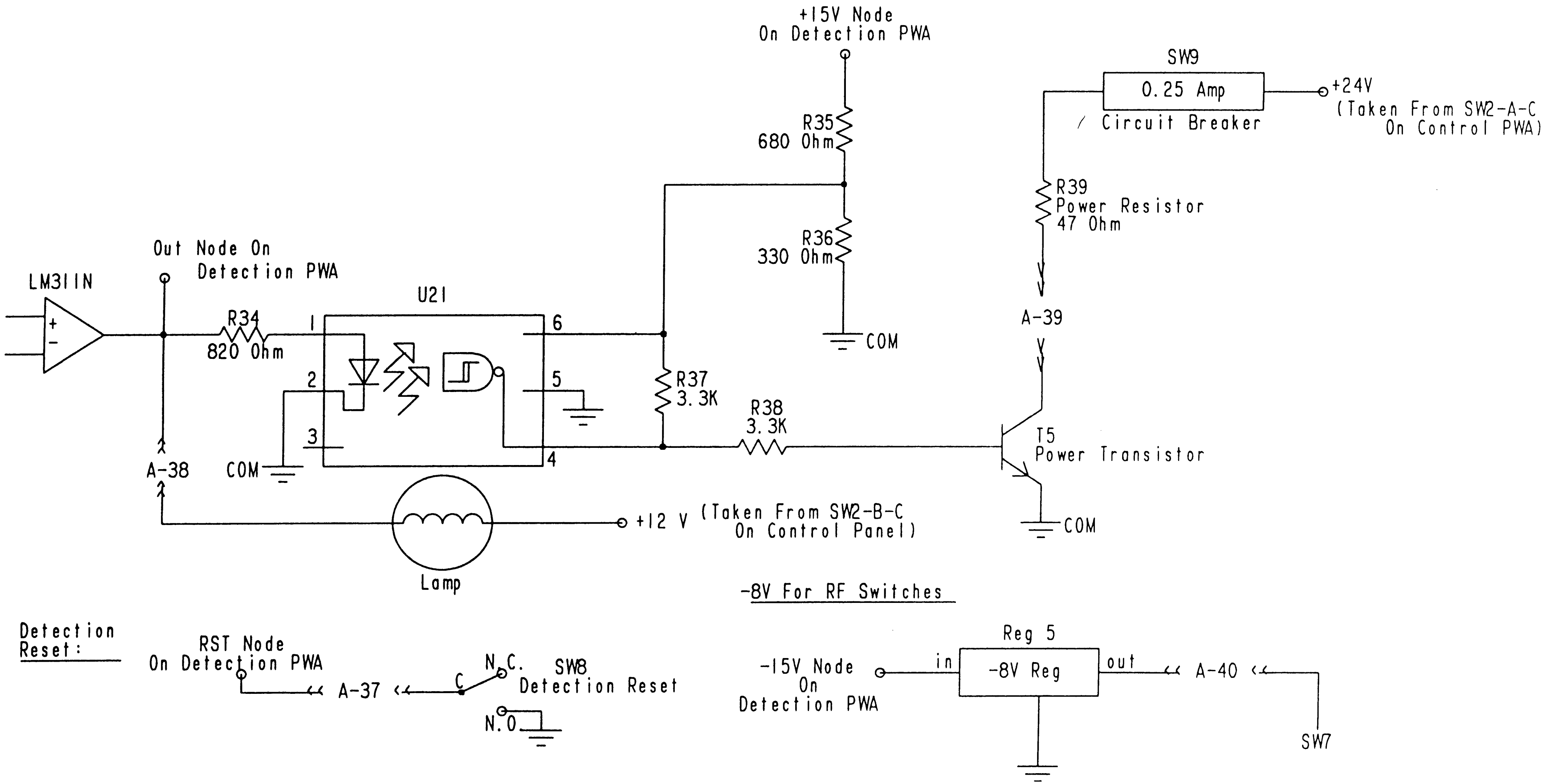


Function	Connected Pads	Position
Time Set	C, D	1
Alarm Set	A, C, D	2
Run - Alarm On	C	3
Run - Alarm Off	B, C	4

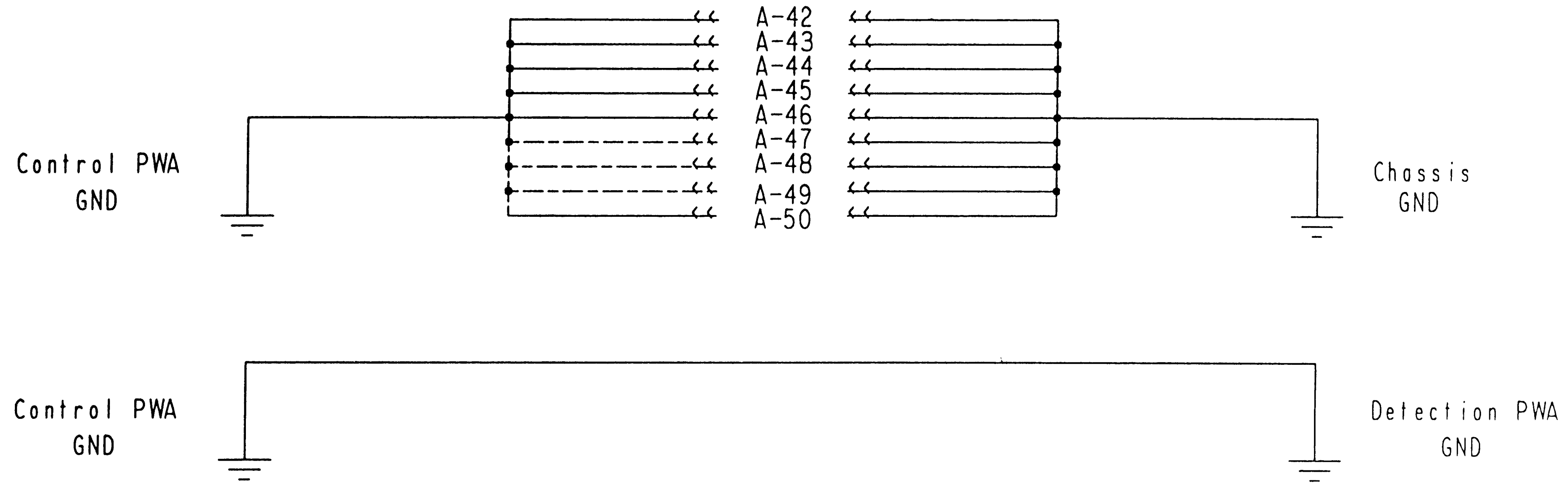
SW11 & SW12 (Timer Controls)
3 Pole - 4 Position Rotary Switch



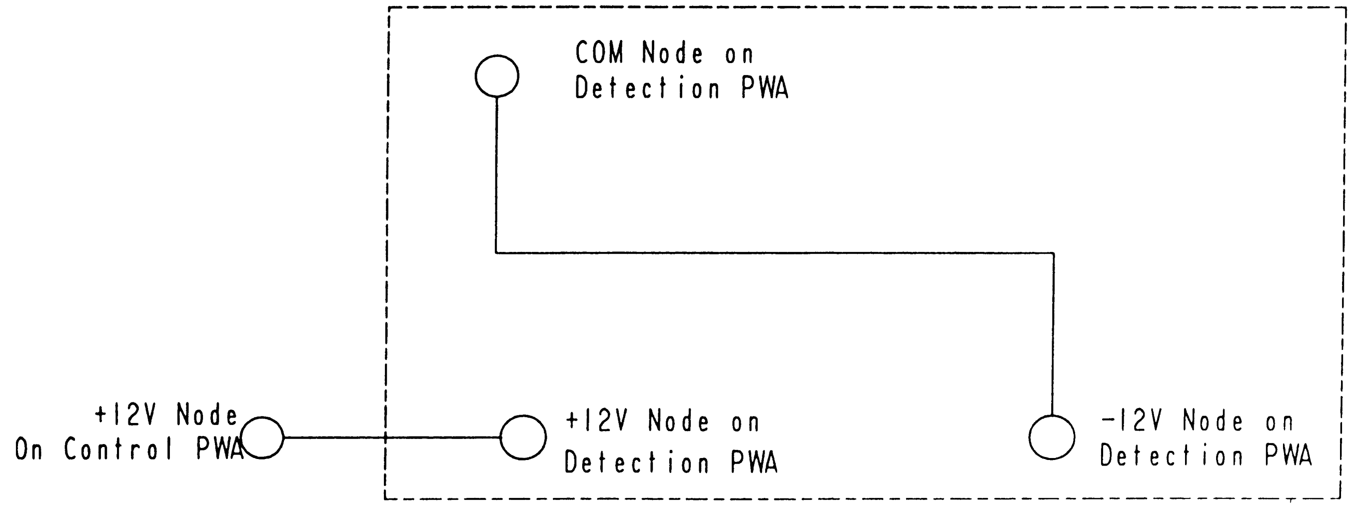
Modifications Made To Applied Microwave's Detection Circuitry



GROUND CONNECTIONS



POWER CONNECTIONS



APPENDIX D

CROSS POLARIZATION ISOLATION TESTS

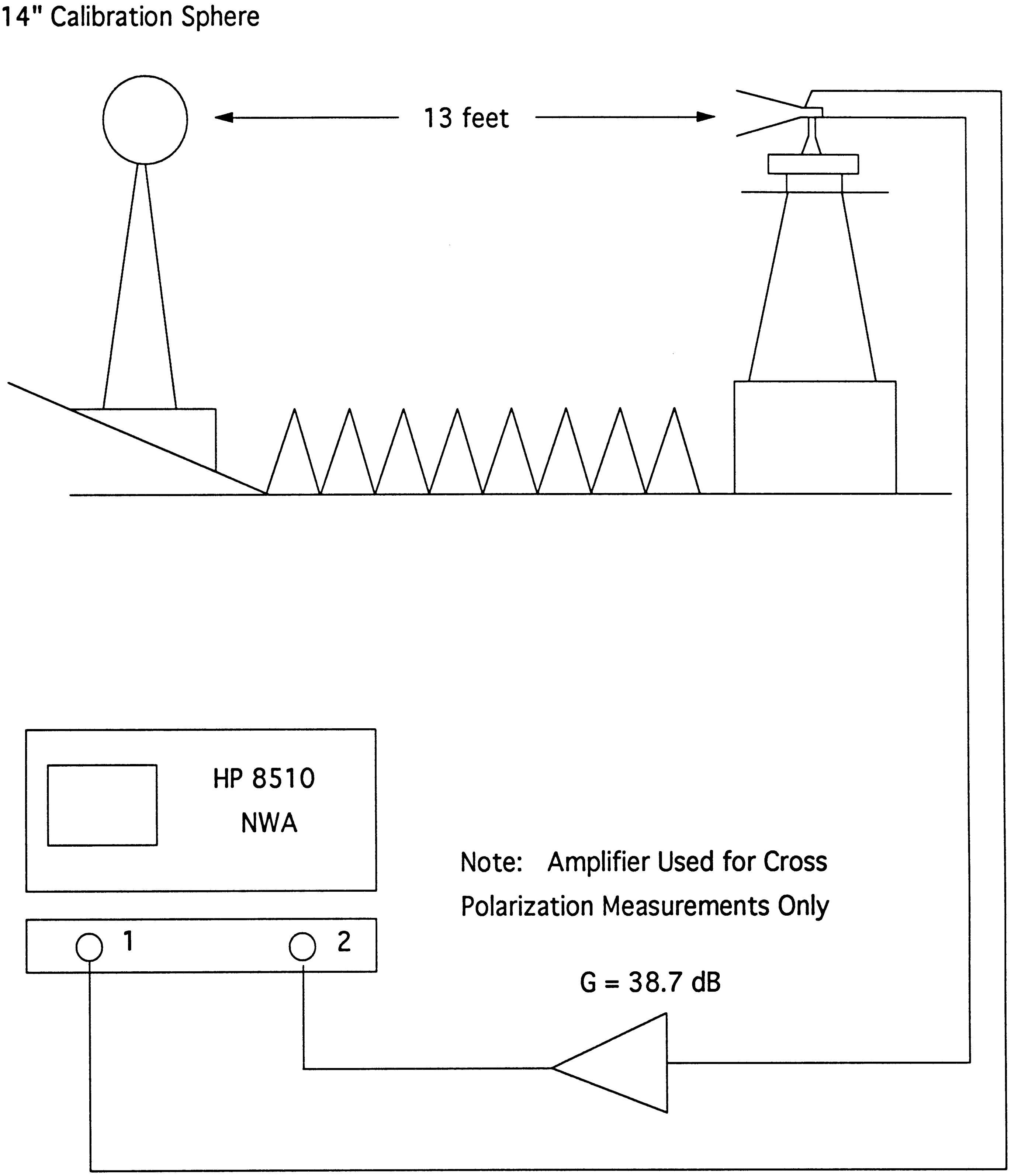


Figure D.1: Cross Polarization Isolation Test Measurements

Figure D.2a: S_{VV} Time Domain Plot of a 14" Calibration Sphere

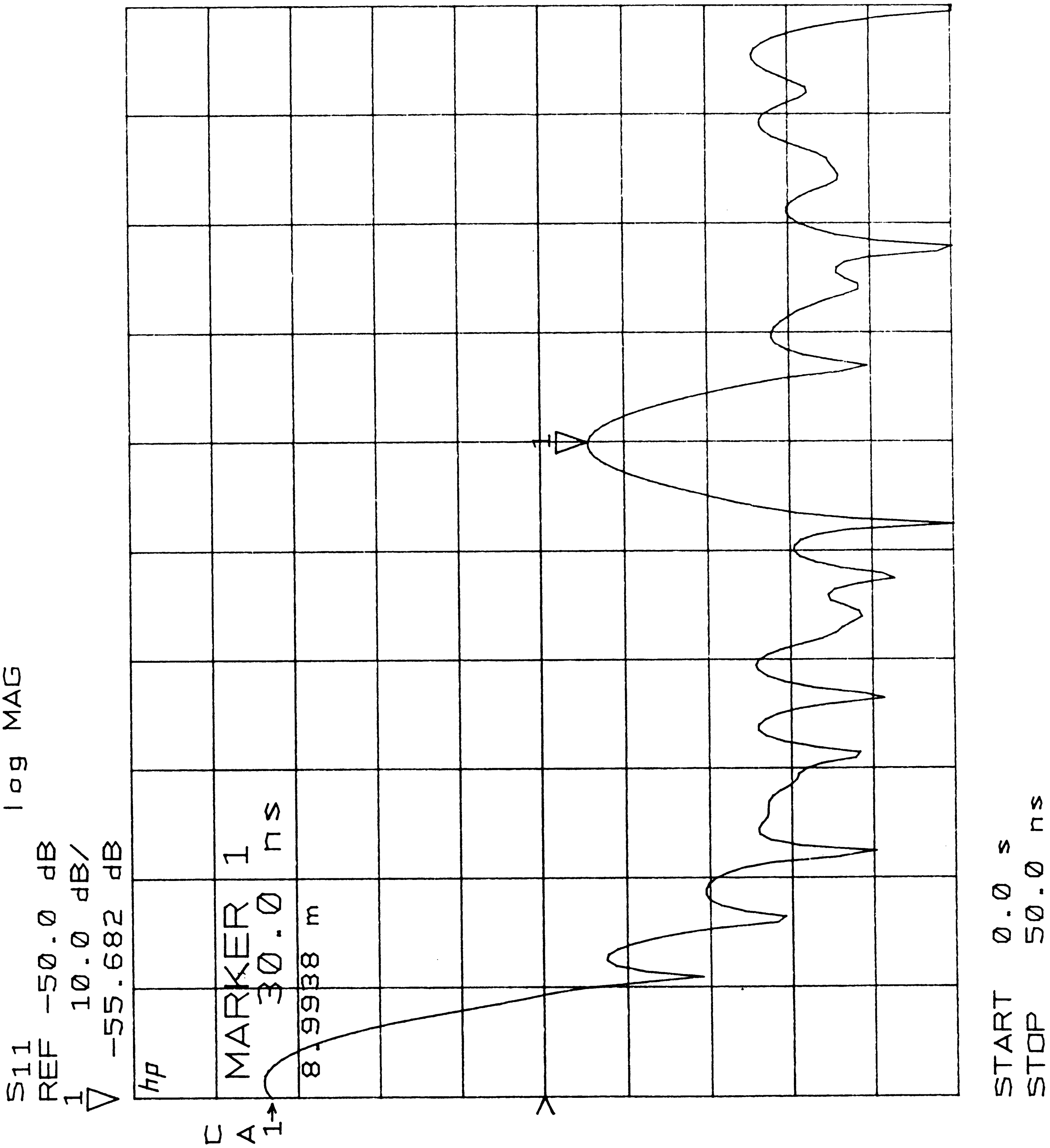


Figure D.2b: S_{vv} Frequency Domain Plot of a 14" Calibration Sphere (With Background Subtraction)

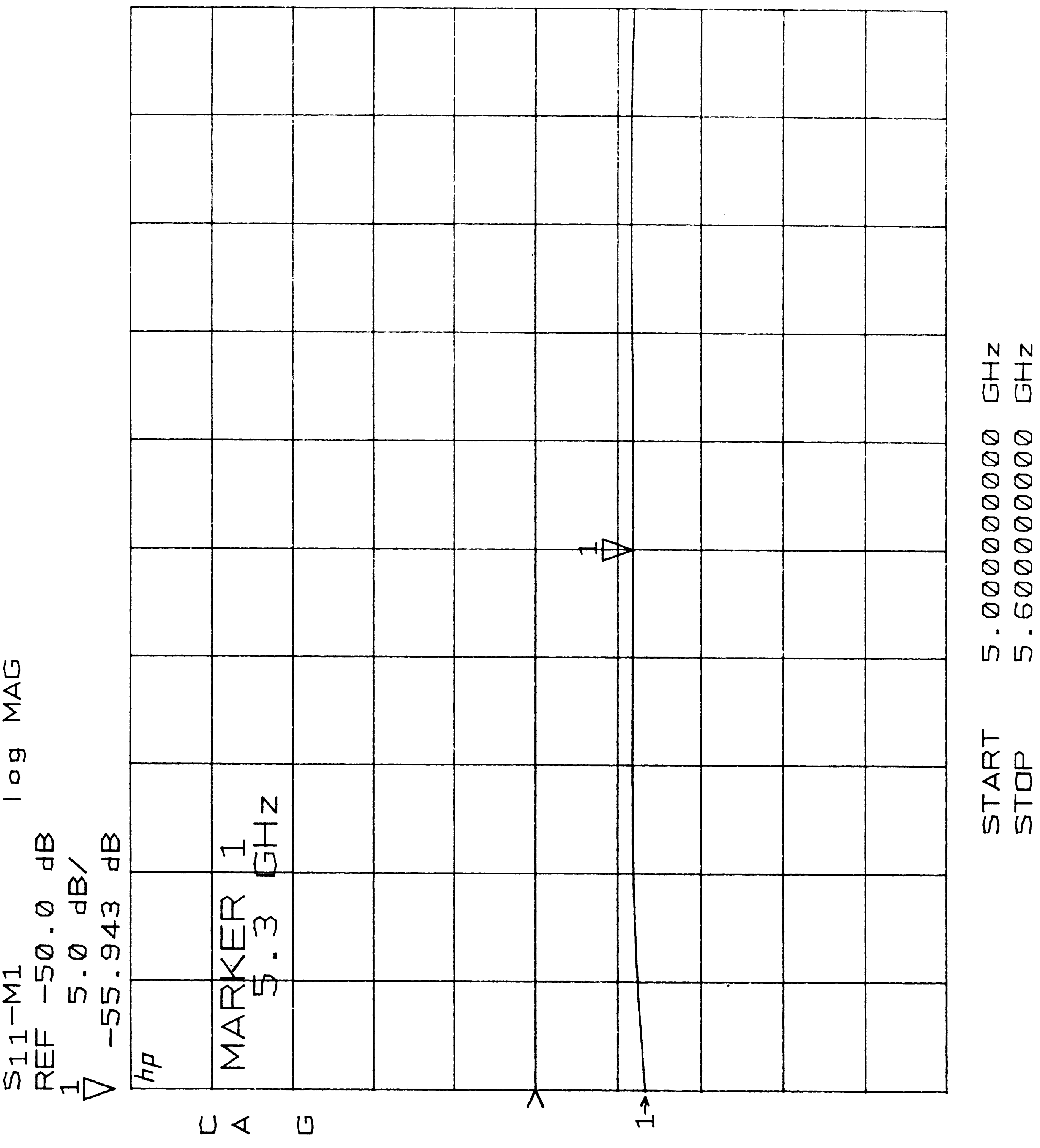


Figure D.2c: S_{hh} Time Domain Plot of a 14" Calibration Sphere

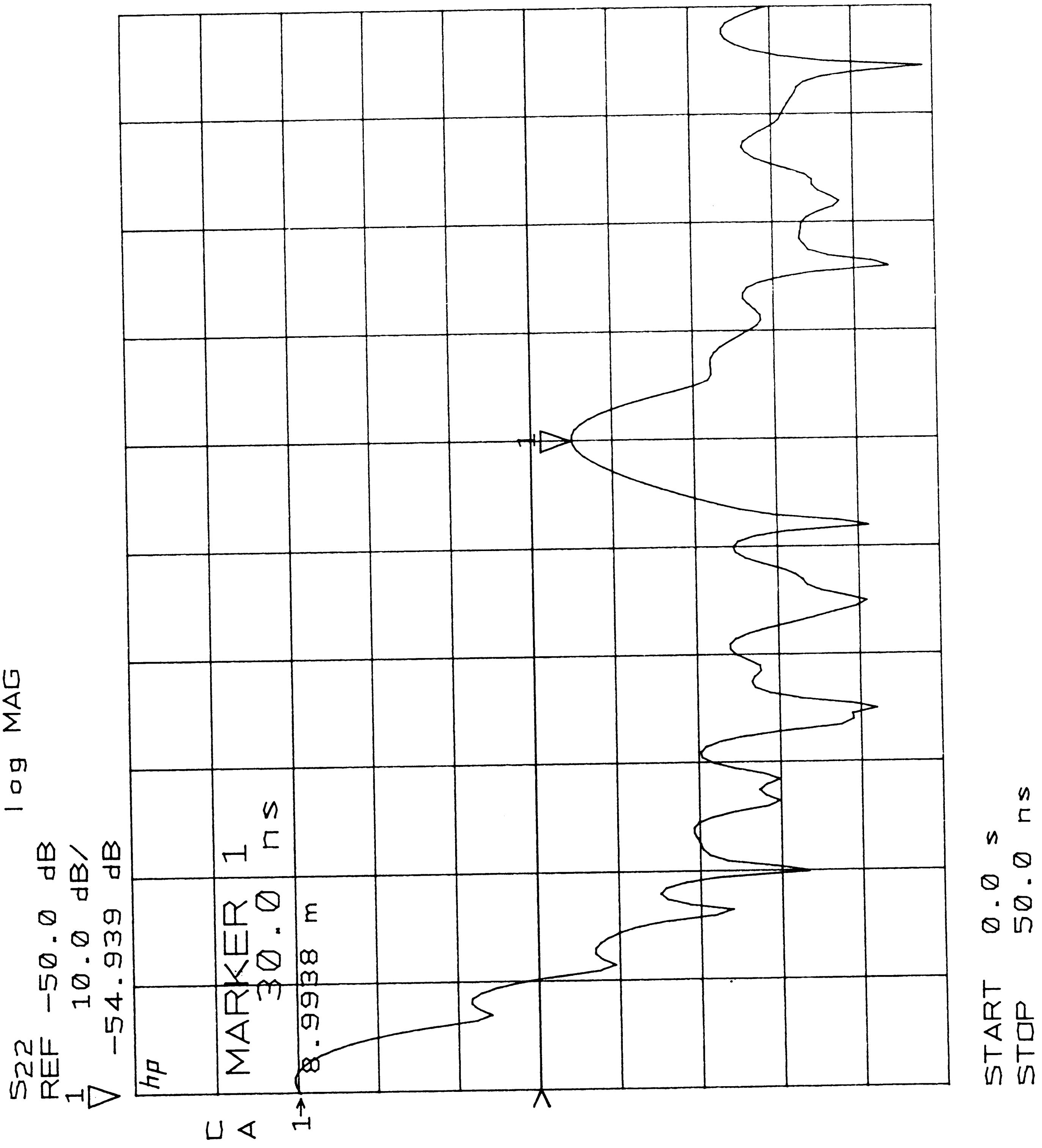


Figure D.2d: S_{hh} Frequency Domain Plot of a 14" Calibration Sphere (With Background Subtraction)

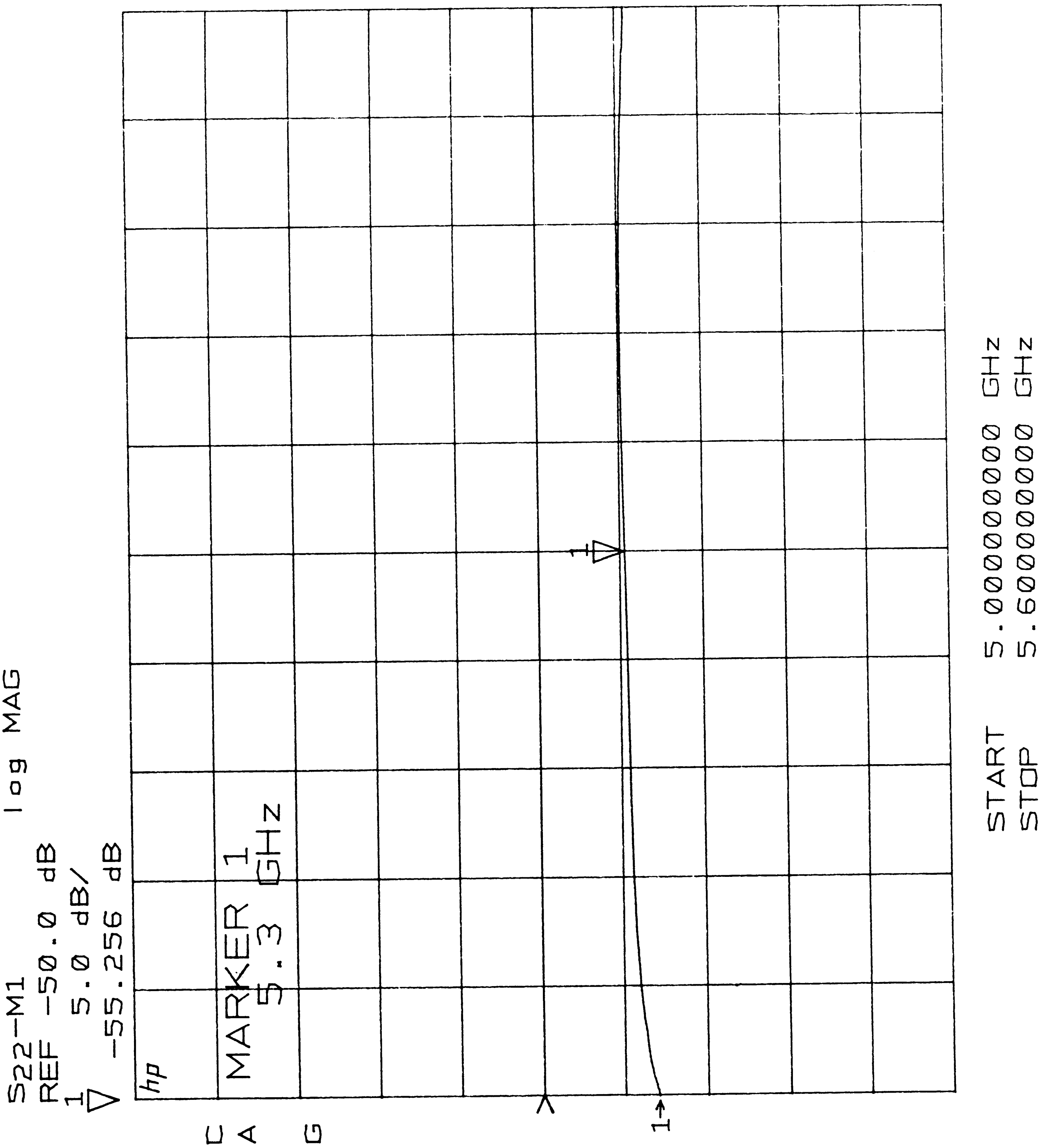


Figure D.2e: S_{hv} Time Domain Plot of a 14" Calibration Sphere

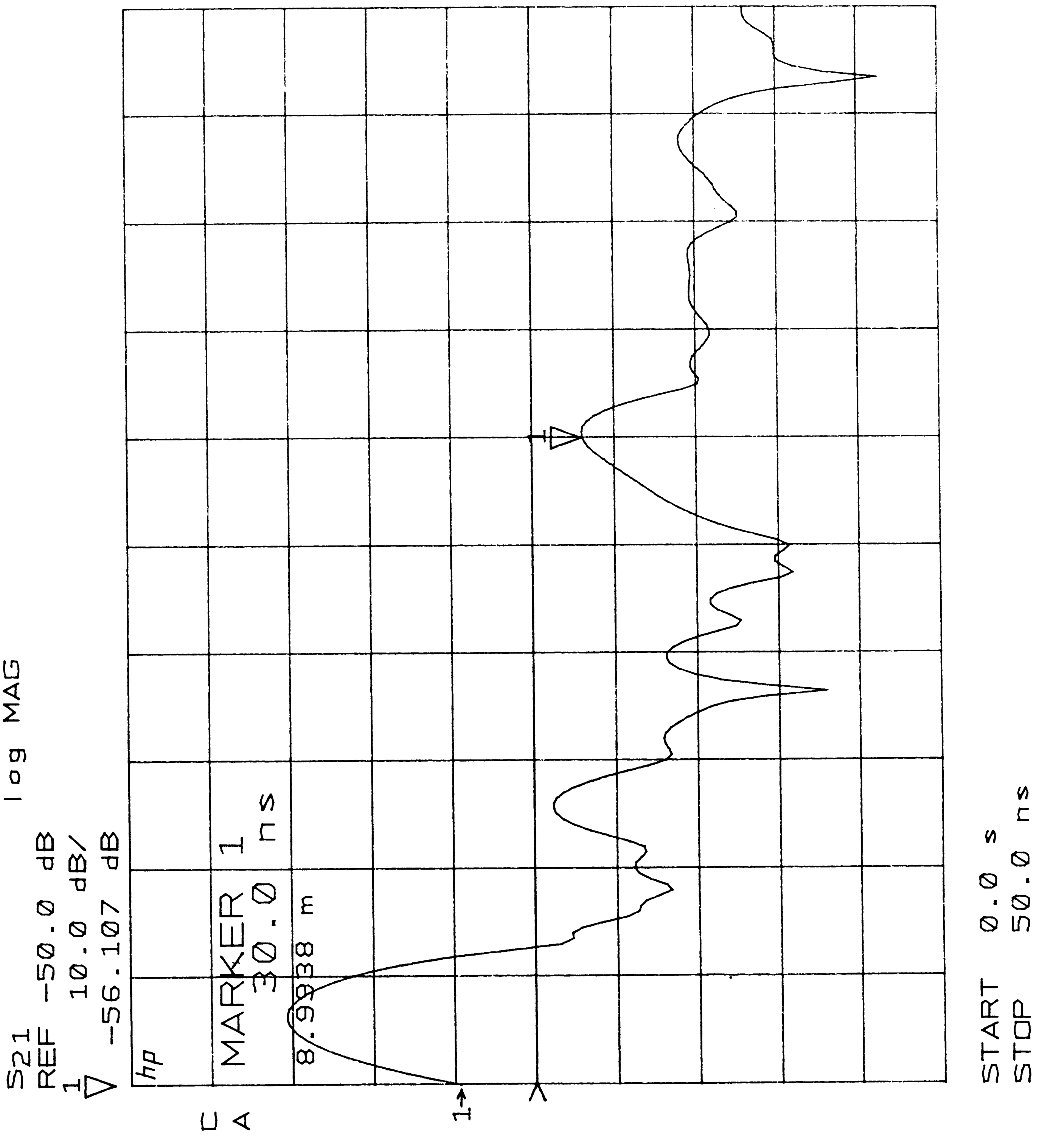


Figure D.2f: S_{hv} Frequency Domain Plot of a 14" Calibration Sphere (With Background Subtraction)

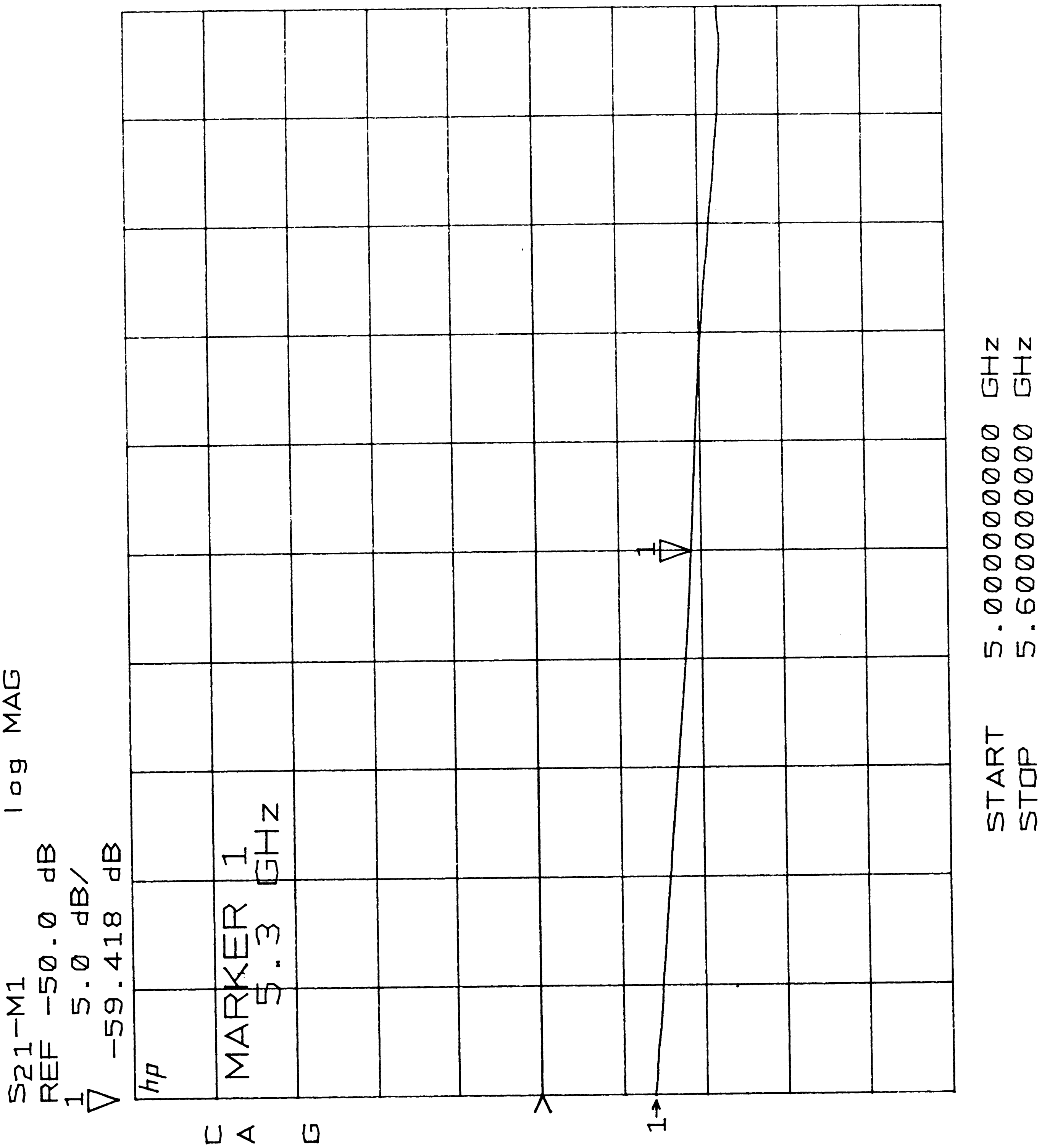


Figure D.2g: S_{vh} Time Domain Plot of a 14" Calibration Sphere

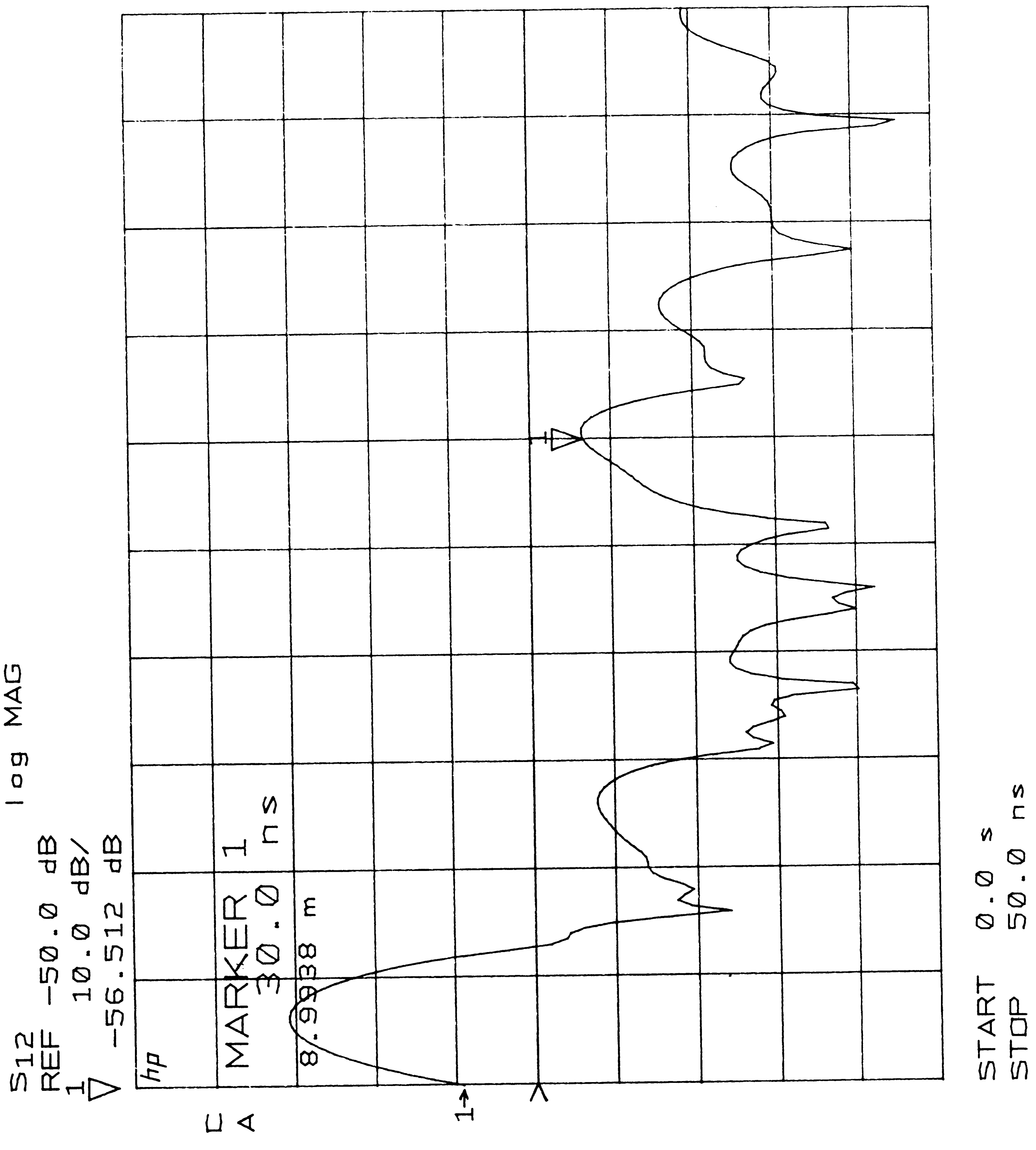
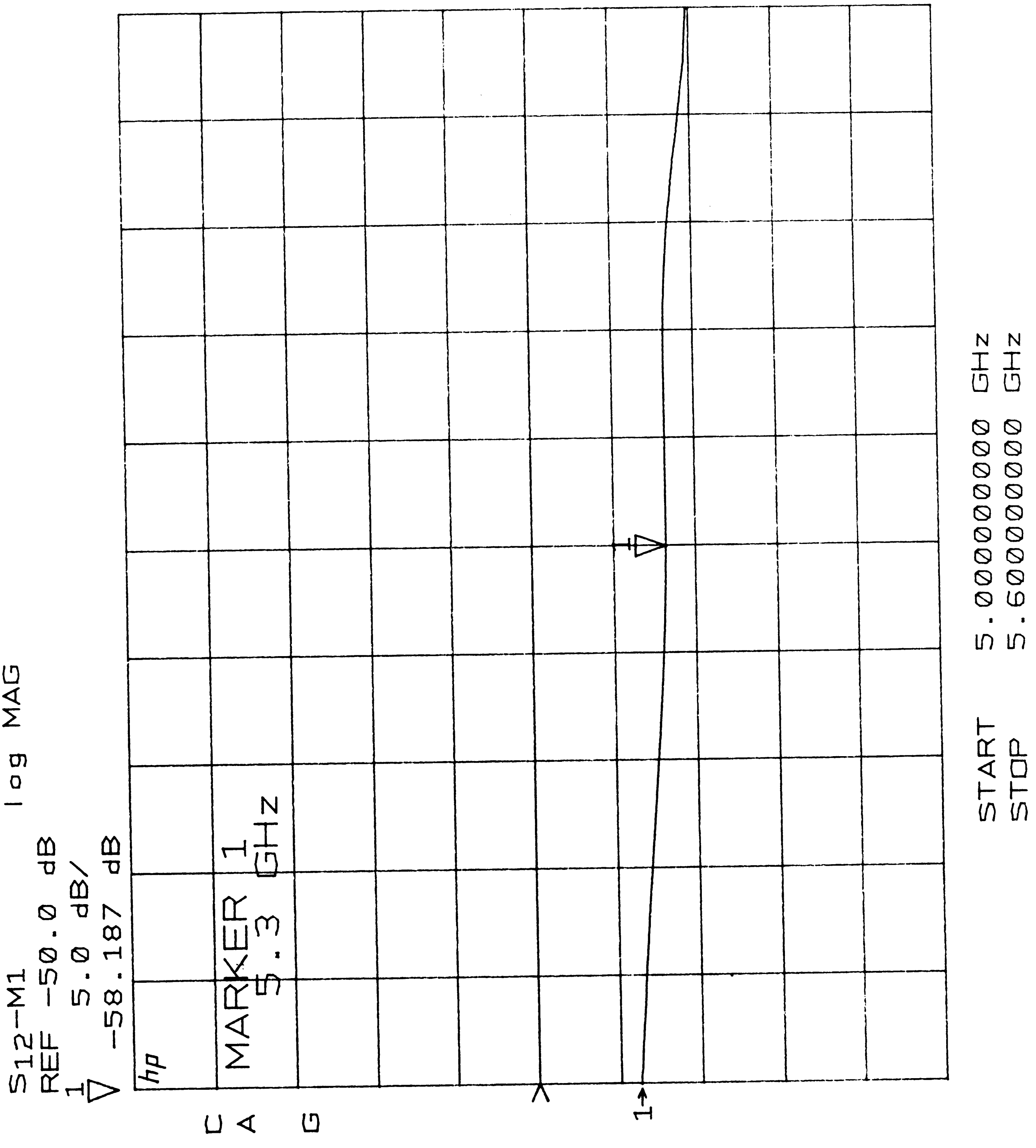


Figure D.2h: S_{vh} Frequency Domain Plot of a 14" Calibration Sphere (With Background Subtraction)



APPENDIX E

MEASUREMENT AND CALIBRATION PROGRAMS

```
C **********************************************************************
C
C                              Appendix E
C                    HPUX Data File Conversion Program
C
C                 JPL AIRSAR / NASA SIR-C SAPARC Project
C
C                             James J. Ahne
C                         Radiation Laboratory
C       Department of Electrical Engineering and Computer Science
C            University of Michigan, Ann Arbor, MI  48109-2122
C
C *********************************************************************
C
C  Read raw data and write converted data for Calibration by a Sphere.
C
      COMPLEX TARGET(81,4,20)
      REAL RAW(42)
      INTEGER NPTS,NTRACE
      CHARACTER POL(4)*2
C
      DATA POL/'VV','HH','HV','VH'/
C
C Read raw real data and convert to complex data for calculation.
C
      OPEN(1,FILE= '152319gc')
      NP=4
      NPTS=20
      NTRACE=81
        DO 10 IT=1,NTRACE
          DO 10 IP=1,NP
             READ(1,*)  (RAW(K),K=1,NPTS*2)
             DO 14 J=1,NPTS
                J2=J*2-1
   14           TARGET(IT,IP,J)=CMPLX(RAW(J2),RAW(J2+1))
   10 CONTINUE
C
      CLOSE(1)
C
C  Open files for outputs.
C
      OPEN(10,FILE='cparc_sirc_45af')
C
C
      DO IT=1,NTRACE
         DO IP=1,NP
            WRITE(10,*)'NT= ',IT,'POL= ',POL(IP)
            DO I=1,NPTS
               WRITE(10,*)TARGET(IT,IP,I)
            ENDDO
         ENDDO
      ENDDO
C
C
      CLOSE(10)
C
      STOP
      END
C
C ********************************************************************
C
      FUNCTION PHASE(Z)
C
C *******************************************************************
C
      COMPLEX Z
      PI=4.*ATAN(1.)
      X=REAL(Z)
      Y=AIMAG(Z)
      PHASE=(180./PI)*ATAN2(Y,X)
      RETURN
      END
```

```
C ********************************************************************
C
C                            Appendix E
C                  Scatterometer Calibration Program
C
C               JPL AIRSAR / NASA SIR-C SAPARC Project
C
C                           James J. Ahne
C                        Radiation Laboratory
C        Department of Electrical Engineering and Computer Science
C             University of Michigan, Ann Arbor, MI  48109-2122
C
C ********************************************************************
C
C
C   (THIS PROGRAM CALIBRATES THE SCATTEROMETERS USING THE NEW METHOD)
C
        COMPLEX C,A,A11,A12,A21,A22
        COMPLEX EVV0(20),EHH0(20),EVH0(20),EHV0(20)
        COMPLEX EVVM(20),EHHM(20),EVHM(20),EHVM(20)
        COMPLEX EVVS(150,20),EHHS(150,20),EVHS(150,20),EHVS(150,20)
        COMPLEX EVVU(150,20),EHHU(150,20),EVHU(150,20),EHVU(150,20)
        COMPLEX S11(150,20),S22(150,20),S12(150,20),S21(150,20)
        complex SIG11,SIG22,SIG12,SIG21
        character*1,char
        COMPLEX S0(20)
        open(9,file='sph_14')
        open(10,file='sph_dat.22')
        open(11,file='mnt_dat.22')
C       open(12,file='sap_bg')    Backgound Data - Not Applicable For Parc
        open(13,file='cparc_sirc_45af')
        PI=4.*ATAN(1.)
        ndata=20
        ntrace=81
C        cir=(0.6945+1.333)/(1.994+2.424)
C    READ THEORETICAL VALUE OF SPHERE
        DO I=1,ndata
        READ(9,*)DUM,S0(I)
        ENDDO
C   SPHERE DATA
       READ(10,103)
         DO I=1,ndata
         READ(10,*)evv0(I)
         ENDDO
       READ(10,103)
        DO I=1,ndata
        READ(10,*)ehh0(I)
        ENDDO
       READ(10,103)
        DO I=1,ndata
        READ(10,*)ehv0(I)
        ENDDO
       READ(10,103)
         DO I=1,ndata
         READ(10,*)evh0(I)
         ENDDO
       READ(11,103)
         DO I=1,ndata
         READ(11,*)evvm(I)
         EVV0(I)=EVV0(I)-evvm(I)
         ENDDO
       READ(11,103)
        DO I=1,ndata
        READ(11,*)ehhm(I)
```

```
         Ehh0(I)=Ehh0(I)-ehhm(I)
         ENDDO
        READ(11,103)
         DO I=1,ndata
         READ(11,*)ehvm(I)
         Ehv0(I)=Ehv0(I)-ehvm(I)
         ENDDO
        READ(11,103)
          DO I=1,ndata
          READ(11,*)evhm(I)
          Evh0(I)=Evh0(I)-evhm(I)
          ENDDO
        do i=1,ndata
        print *,20*alog10(cabs(evv0(i))),20*alog10(cabs(ehv0(i))),
     &          20*alog10(cabs(evh0(i))),20*alog10(cabs(ehh0(i)))
        enddo
        pause
C BACKGROUND DATA
C      do itrace=1,ntrace
C        READ(12,103)
C          DO I=1,ndata
C          READ(12,*)X,Y
C          EVVS(ITRACE,I)=CMPLX(X,Y)
C          ENDDO
C        READ(12,103)
C         DO I=1,ndata
C         READ(12,*)X,Y
C         EHHS(ITRACE,I)=CMPLX(X,Y)
C         ENDDO
C        READ(12,103)
C         DO I=1,ndata
C         READ(12,*)X,Y
C         EHVS(ITRACE,I)=CMPLX(X,Y)
C         ENDDO
C        READ(12,103)
C         DO I=1,ndata
C          READ(12,*)X,Y
C          EVHS(ITRACE,I)=CMPLX(X,Y)
C          ENDDO
C      enddo
C UNKNOWN DATA
       DO ITRACE=1,ntrace
         READ(13,103)
103      FORMAT(1X)
          DO I=1,ndata
          READ(13,*)evvu(itrace,i)
C         EVVU(ITRACE,I)=CMPLX(X,Y)-EVVS(ITRACE,I)
          ENDDO
         READ(13,103)
         DO I=1,ndata
         READ(13,*)ehhu(itrace,i)
C        EHHU(ITRACE,I)=CMPLX(X,Y)-EHHS(ITRACE,I)
         ENDDO
         READ(13,103)
         DO I=1,ndata
         READ(13,*)ehvu(itrace,i)
C        EHVU(ITRACE,I)=CMPLX(X,Y)-EHVS(ITRACE,I)
         ENDDO
         READ(13,103)
          DO I=1,ndata
          READ(13,*)evhu(itrace,i)
C         EVHU(ITRACE,I)=CMPLX(X,Y)-EVHS(ITRACE,I)
          ENDDO
         ENDDO
```

```fortran
C        do itrace=1,ntrace
C        print *,20*alog10(cabs(evvu(ITRACE,4))),
C     &            20*alog10(cabs(ehvu(itrace,4))),
C     &            20*alog10(cabs(evhu(itrace,4))),
C     &            20*alog10(cabs(ehhu(itrace,4)))
C        enddo
C
C This is a calibration based on c1=c2
C
        DO ITRACE=1,NTRACE
        DO I=1,ndata
        A=EVHO(I)*EHVO(I)/(EVVO(I)*EHHO(I))
        C=(1-CSQRT(1-A))/CSQRT(A)
        A11=(EVVU(ITRACE,I)/EVVO(I))*(1+C**2)*SO(I)
        A22=(EHHU(ITRACE,I)/EHHO(I))*(1+C**2)*SO(I)
        A12=(EVHU(ITRACE,I)/EVHO(I))*(2*C)*SO(I)
        A21=(EHVU(ITRACE,I)/EHVO(I))*(2*C)*SO(I)
        S12(ITRACE,I)=(A12+C**2*A21-C*(A11+A22))/(1-C**2)**2
        S21(ITRACE,I)=(A21+C**2*A12-C*(A11+A22))/(1-C**2)**2
        S11(ITRACE,I)=(A11+C**2*A22-C*(A12+A21))/(1-C**2)**2
        S22(ITRACE,I)=(A22+C**2*A11-C*(A12+A21))/(1-C**2)**2
        ENDDO
        PRINT *,ITRACE
       ENDDO
        open(15,file='cparc_sirc_am_45af')
        open(16,file='cparc_sirc_ph_45af')
C
C      The variable I now sets the frequency (for 20 data points,
C          use I=10 -- chooses the center frequency)
        I=10
C
C
10      print*,'input psi'
        read*,psi
        psi=psi*3.1415/180.
        sig11=s11(41,10)*cos(psi)**2-sin(psi)*cos(psi)*(s12(41,10)
     &      +s21(41,10))  +s22(41,10)*sin(psi)**2
        sig22=s11(41,10)*sin(psi)**2+sin(psi)*cos(psi)*(s12(41,10)
     &       +s21(41,10))+s22(41,10)*cos(psi)**2
        sig12=sin(psi)*cos(psi)*(s11(41,10)-s22(41,10))-sin(psi)**2
     &       *s21(41,10)+cos(psi)**2*s12(41,10)
        sig21=sin(psi)*cos(psi)*(s11(41,10)-s22(41,10))-sin(psi)**2
     &       *s12(41,10)+cos(psi)**2*s21(41,10)
        print*,cabs(sig11),cabs(sig22),cabs(sig12),cabs(sig21)
        read*,jnk
        if(jnk .eq.0)goto 10

       angmax=40.0
       angmin=-40.0
       print*, ntrace
       dang=(angmax-angmin)/(ntrace-1)
       do itr=1,ntrace
          ang=angmin+(itr-1)*dang
C
        sig11=s11(itr,i)*cos(psi)**2-sin(psi)*cos(psi)*(s12(itr,i)
     &      +s21(itr,i))  +s22(itr,i)*sin(psi)**2
        sig22=s11(itr,i)*sin(psi)**2+sin(psi)*cos(psi)*(s12(itr,i)
     &       +s21(itr,i))+s22(itr,i)*cos(psi)**2
        sig12=sin(psi)*cos(psi)*(s11(itr,i)-s22(itr,i))-sin(psi)**2
     &       *s21(itr,i)+cos(psi)**2*s12(itr,i)
        sig21=sin(psi)*cos(psi)*(s11(itr,i)-s22(itr,i))-sin(psi)**2
     &       *s12(itr,i)+cos(psi)**2*s21(itr,i)
        s11(itr,i)=sig11
        s22(itr,i)=sig22
        s12(itr,i)=sig12
        s21(itr,i)=sig21
         WRITE(15,*)ANG,10.*ALOG10(4.*PI*cabs(S11(ITR,I))**2)
     &             ,10.*ALOG10(4.*PI*cabs(S22(ITR,I))**2)
     &             ,10.*ALOG10(4.*PI*cabs(S12(ITR,I))**2)
     &             ,10.*ALOG10(4.*PI*cabs(S21(ITR,I))**2)
        WRITE(16,*)ANG,PHASE(S22(ITR,I)/S11(ITR,I))
     &              ,PHASE(S12(ITR,I)/S11(ITR,I))
     &              ,PHASE(S21(ITR,I)/S11(ITR,I))
        ENDDO
        STOP
        END
C*****************************************************************
C
        FUNCTION PHASE(Z)
C
C****************************************************************
        COMPLEX Z
        PI=4.*ATAN(1.)
        X=REAL(Z)
        Y=AIMAG(Z)
        PHASE=(180./PI)*ATAN2(Y,X)
        if(phase.lt.-100.0) phase=phase+360
        RETURN
        END
```

```
!*****************************************************************************
!
!                          Appendix E
!            HP Basic Scatterometer Measurement Program
!
!              JPL AIRSAR / NASA SIR-C SAPARC Project
!
!                          James J. Ahne
!                       Radiation Laboratory
!       Department of Electrical Engineering and Computer Science
!            University of Michigan, Ann Arbor, MI  48109-2122
!
!*****************************************************************************
!
10     !*********************************************************************
20     !   L/C/X POLARIMETER MEASUREMENT PROGRAM
30     !      FILE: SAPARC_4
40     !*********************************************************************
50     !   LAST EDIT: Sep 22, 1992      Change for C-Band SAPARC 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 1.25,1.2,1.5                                     ! FREQ_CENT
280    DATA .3,.6,.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 200E-9,200E-9,200E-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 "JIM_CPARC:,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$="FAST ACQ"
590    F_disp=2
600    P_disp=3
610    Printer_flag=0
620    !
630    Hp_bus_init
640    IF Printer_flag=1 THEN Out_type$="PRINT/DISC"
650    !
660    Vel=2.99792458E+8
670    Ntrace=81
680    Npts=401
690    Nskip=20
700    Ndata=20
710    Average_factor=4
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 ! OUTPUT @Relay;"?*B1256A34"
1010   Series_init
1020   !
1030   !
1040 Start_loop: !
1050           !
1060      Print_banner4
1070         ON KEY 0 LABEL "                ",FNTrap_level GOSUB Null
1080         ON KEY 1 LABEL " REFERENCE CAL",FNTrap_level GOSUB Ref_target
1090         ON KEY 2 LABEL "  TARGET RUN   ",FNTrap_level GOSUB Acq_target
1100         ON KEY 3 LABEL " SET FREQUENCY",FNTrap_level GOSUB Freq_set
1110         ON KEY 4 LABEL "     ANGLE     ",FNTrap_level GOSUB Set_angle
1120         ON KEY 5 LABEL " TARGET NAME   ",FNTrap_level GOSUB Set_target
```

```
1130         ON KEY 6 LABEL " # OF TRACES  ",FNTrap_level GOSUB Set_traces
1140         ON KEY 7 LABEL " # OF POINTS  ",FNTrap_level GOSUB Set_points
1150         ON KEY 8 LABEL "# OF AVERAGES ",FNTrap_level GOSUB Set_average
1160         ON KEY 9 LABEL "     QUIT     ",FNTrap_level GOTO Quit_fast_acq
1170      GOSUB Allocate_matrix
1180      LOOP
1190         EXIT IF Exit_flag=1
1200      END LOOP
1210      GOSUB Deallocate_mtrx
1220      Exit_flag=0
1230      GOTO Start_loop
1240      !
1250 Null: RETURN
1260 !
1270 !-------------------------------------------------------------------------
1280 !
1290 Ref_target: ! Acquire a reference target data set.
1300             !
1310      OFF KEY
1320      Clear_crt
1330      OUTPUT @Nwa;"TIMDTRANON; LOGM; GATEOFF;"
1340      OUTPUT @Nwa;"AUTO; ELED 100NS; STAR ONS; STOP 300NS;"
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(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(T,P,Nst)=Target_response(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(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(T,P,Nst)=Target_response(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$,F)
2090      Pol_sw(F_disp,P_disp)
2100      DISP "Reference target mount response saved."
2110      Exit_flag=1
2120 RETURN
2130 !
2140 !
2150 !
2160 Acq_target: !
2170             !
2180      OFF KEY
2190      Clear_crt
2200      OUTPUT @Nwa;"TIMDTRANON; LOGM; GATEOFF;"
2210      OUTPUT @Nwa;"ELED 100NS; STAR ONS; STOP 300NS;"
2220      PRINT TABXY(1,10);"Please point scatterometer assembly at surface target."
2230      PRINT TABXY(1,12);"Press CONTINUE when ready..."
2240      PAUSE
2250      GOSUB Set_gates
2260      OUTPUT @Nwa;"TIMDTRANOFF; POLA; AVERFACT";VAL$(Average_factor);";"
2270      OUTPUT @Nwa;"GATEOFF;AVEROON;"
2280      !
2290      ! Get the target response.
2300      !
2310      FOR T=1 TO Ntrace
2320      !
2330      ! Get angles
2340      !
2350      IF T=1 THEN
2360         Rotation_state=-1
2370      ELSE
2380         Rotation_state=2
2390      END IF
2400      SELECT Rotation_state
```

```
2410      CASE =0
2420           Clear_crt
2430           PRINT TABXY(1,4);"When ready for measurement, press CONTINUE."
2440           BEEP
2450           PAUSE
2460           Clear_crt(3,16)
2470           PRINT TABXY(1,4);"Collecting data... "
2480      CASE ELSE
2490           PRINT TABXY(1,4);"Current angle is ";Current_angle;" degrees."
2500           Rotate_target    !
2510           WAIT 1
2520           Clear_crt(3,16)
2530           PRINT TABXY(1,4);"Collecting data ..."
2540      END SELECT
2550          FOR F=1 TO 3
2560           IF Meas_flag(F) THEN
2570             Freq_set(F)
2580             Freq_sw(F)
2590             OUTPUT @Nwa;"GATEOFF;"
2600             OUTPUT @Nwa;"GATECENT";VAL$(Gate_cent(F));"S;"
2610             OUTPUT @Nwa;"GATESPAN";VAL$(Gate_span(F));"S;"
2620             OUTPUT @Nwa;"GATEON; WAIT;"
2630       PRINT Npts,Ntrace,Ndata,Nskip
2640             FOR P=1 TO 4
2650               Pol_sw(F,P)
2660               OUTPUT @Nwa;"NUMG";VAL$(Average_factor);";WAIT; FORM3; OUTPFORM;"
2670               ENTER @Nwa_data1;Preamble,Bytes,Trace(*)
2680               MAT Target_response(P,*)= Trace
2690             NEXT P
2700             Nskh=Nskip+1
2710             FOR P=1 TO 4
2720               FOR Nt=Nskh TO Npts STEP Nskip
2730                 Nst=INT(Nt/Nskip)
2740                 Target_data(T,P,Nst)=Target_response(P,Nt)
2750               NEXT Nt
2760             NEXT P
2770           END IF
2780         NEXT F
2790         PRINT "# OF TRACES LEFT=",Ntrace-T
2800      NEXT T
2810      Store_file(Target_data(*),"GND",FNTime_stamp$,F)
2820      DISP "Surface target data saved."
2830      BEEP
2840        Rotation_state=4
2850        Rotate_target
2860      WAIT 5
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_1
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           GOSUB Set_c
3120           GOTO Store_band
3130 RETURN
3140 Set_1:     OFF KEY 0
3150            Meas_flag(1)=1
3160            F_disp=1
3170 RETURN
3180 Set_c:     OFF KEY 1
3190            Meas_flag(2)=1
3200            F_disp=2
3210 RETURN
3220 Set_x:     OFF KEY 2
3230            Meas_flag(3)=1
3240            F_disp=3
3250 RETURN
3260 Store_band: Print_banner4
3270             Exit_flag=1
3280             GOSUB Allocate_matrix
3290 RETURN
3300 Cancel_band: !
3310              MAT Meas_flag= Meas_flag_old
3320              Exit_flag=1
3330              GOSUB Allocate_matrix
3340 RETURN
3350 !
3360 !---------------------------------------------------------------------------
3370 !
3380 Set_angle:  !
3390             INPUT "Enter measurement angle: ",Angle
3400             Angle$=VAL$(Angle)&CHR$(179)&"    "    ! Degree sign.
3410             Print_banner4
3420 RETURN
3430 !
3440 !---------------------------------------------------------------------------
3450 !
3460 Set_target: !
3470             LINPUT "Enter target type or name: ",Target$
3480             Target$=TRIM$(Target$)
3490             Target$=Target$&RPT$(" ",30-LEN(Target$))
3500             Print_banner4
3510 RETURN
3520 !
3530 !---------------------------------------------------------------------------
3540 !
3550 Set_traces: !
3560             INPUT "Enter the number of traces( or angles) desired( >=3 ): ",Ntrace
3570             GOSUB Deallocate_mtrx
3580             GOSUB Allocate_matrix
3590             Print_banner4
3600 RETURN
3610 !
3620 !---------------------------------------------------------------------------
3630 Set_points: !
3640             INPUT "Enter the number of sample points (Npts,201): ",Npts
3650             OUTPUT @Nwa;"POIN "&VAL$(Npts)&";"
3660             INPUT "Enter the data points to be stored (Ndata,10):",Ndata
3670             Nskip=INT(Npts/Ndata)
3680             Bytes=16*Ndata
```

```
3690            Print_banner4
3700            GOSUB Deallocate_mtrx
3710            GOSUB Allocate_matrix
3720 RETURN
3730 !
3740 !------------------------------------------------------------------------
3750 !
3760 Set_average: !
3770             INPUT "Enter averaging factor: ",Average_factor
3780             Print_banner4
3790 RETURN
3800 !
3810 !------------------------------------------------------------------------
3820 !
3830 Allocate_matrix: ! Allocate storage space for data.
3840                 !
3850      System_memory=VAL(SYSTEM$("AVAILABLE MEMORY"))
3860      Avail_traces=MIN(Ntrace,INT(System_memory-50000-3*4*16.*Npts)/(3*4*16.*Ndata))
3870      IF Avail_traces<Ntrace THEN
3880         BEEP
3890         PRINT TABXY(1,16);"Memory has capacity for only ";Avail_traces;" traces."
3900         PRINT "Press CONTINUE key to continue"
3910         PAUSE
3920         Ntrace=Avail_traces
3930      END IF
3940      ALLOCATE COMPLEX Trace(Npts),Target_response(4,Npts)
3950      ALLOCATE COMPLEX Target_data(Ntrace,4,Ndata)
3960   RETURN
3970 Deallocate_mtrx: ! Return to main program.
3980                 !
3990      DEALLOCATE Target_response(*),Trace(*)
4000      DEALLOCATE Target_data(*)
4010   RETURN
4020 !
4030 !------------------------------------------------------------------------
4040 !
4050 Set_gates: ! Set gate centers and spans.
4060            !
4070   FOR F=1 TO 3
4080      IF Meas_flag(F) THEN
4090         Freq_set(F)
4100         Freq_sw(F)
4110         P=3
4120            Pol_sw(F,P)
4130            OUTPUT @Nwa;"TIMDTRANON; LOGM;"
4140            OUTPUT @Nwa;"ELED 100NS; STAR 0NS; STOP 300NS; WAIT;"
4150            OUTPUT @Nwa;"FORM3; OUTPACTI;"
4160            ENTER @Nwa;Gate_cent(F)
4170            OUTPUT @Nwa;"MARKOFF;"
4180            OUTPUT @Nwa;"CONT;"
4190            OUTPUT @Nwa;"GATESPAN";VAL$(Gate_span(F));"S;"
4200            OUTPUT @Nwa;"GATECENT";VAL$(Gate_cent(F));"S;"
4210            OUTPUT @Nwa;"KEY41; KEY59; KEY58; KEY59;"
4220            LOCAL @Nwa
4230            DISP "Adjust gate center to suit, and press CONTINUE."
4240            PAUSE
4250            OUTPUT @Nwa;"OUTPACTI;"
4260            ENTER @Nwa;Gate_cent(F)
4270            OUTPUT @Nwa;"GATESPAN";VAL$(Gate_span(F));";"
4280            OUTPUT @Nwa;"KEY41; KEY59; KEY58; KEY4;"
4290            LOCAL @Nwa
4300            DISP "Adjust gate span to suit, and press CONTINUE."
4310            PAUSE
4320            OUTPUT @Nwa;"OUTPACTI;"
4330                ENTER @Nwa;Gate_span(F)
4340      END IF
4350   NEXT F
4360   RETURN
4370   !
4380   !------------------------------------------------------------------------
4390   !
4400 Quit_fast_acq: ! End of program
4410  DISP "PROGRAM EXIT"
4420  GOSUB Deallocate_mtrx
4430  LOAD KEY "EDITKEY:MEMORY,0,1"
4440  STOP
4450  END
4460  !
4470  !*********************************************************************
4480  !
4490  DEF FNAsk(Prompt$)
4500    OFF KEY
4510    DISP Prompt$;
4520    INPUT "",Yn$
4530    Yn$=UPC$(Yn$[1,1])
4540    SELECT Yn$
4550           CASE ="Y"
4560                RETURN 1
4570           CASE ="N",=""
4580                RETURN 0
4590           CASE ELSE
4600                RETURN 0
4610    END SELECT
4620  FNEND
4630  !
4640  !*********************************************************************
4650  !
4660  DEF FNFileloc$(File$,Dir$)
4670    INTEGER C  ! for the location of the ':' in Dir$ (minus 1)
4680    LET C=POS(Dir$,":")-1
4690    IF C<=0 THEN
4700      RETURN TRIM$(File$&Dir$)
4710    ELSE
4720      RETURN Dir$[1,C]&RPT$("/",Dir$[C,C]<>"/")&File$&Dir$[C+1,LEN(Dir$)]
4730    END IF
4740  FNEND ! Fileloc
4750  !
4760  !********************************************************************
4770  !
4780   DEF FNTime_stamp$(OPTIONAL Time_format)
4790       !
4800       DIM Time_digits$[4],Year_digits$[6]
4810       DIM Machine_time$[8],Machine_date$[11]
4820       REAL Timedate_now
4830       !
4840       Timedate_now=TIMEDATE
4850       Machine_date$=DATE$(Timedate_now)
4860       Machine_time$=TIME$(Timedate_now)
4870       Time_digits$=Machine_time$[1,2]&Machine_time$[4,5]
4880       Year_digits$[1,2]=Machine_date$[10,11]
4890       IF Machine_date$[1,1]=" " THEN Machine_date$[1,1]="0"
4900       !
4910       SELECT Machine_date$[4,6]
4920       CASE ="Jan"
4930            Year_digits$[3,4]="01"
4940       CASE ="Feb"
4950            Year_digits$[3,4]="02"
4960       CASE ="Mar"
```

```
4970            Year_digits$[3,4]="03"
4980       CASE ="Apr"
4990            Year_digits$[3,4]="04"
5000       CASE ="May"
5010            Year_digits$[3,4]="05"
5020       CASE ="Jun"
5030            Year_digits$[3,4]="06"
5040       CASE ="Jul"
5050            Year_digits$[3,4]="07"
5060       CASE ="Aug"
5070            Year_digits$[3,4]="08"
5080       CASE ="Sep"
5090            Year_digits$[3,4]="09"
5100       CASE ="Oct"
5110            Year_digits$[3,4]="10"
5120       CASE ="Nov"
5130            Year_digits$[3,4]="11"
5140       CASE ="Dec"
5150            Year_digits$[3,4]="12"
5160       END SELECT
5170       !
5180       Year_digits$[5,6]=Machine_date$[1,2]
5190       SELECT NPAR
5200       CASE =0
5210            RETURN Year_digits$[5,6]&Time_digits$
5220       CASE =1
5230            IF Time_format=1 THEN
5240               RETURN Year_digits$&Time_digits$
5250            END IF
5260            IF Time_format=2 THEN
5270               RETURN Year_digits$[3,6]&Time_digits$
5280            END IF
5290       END SELECT
5300   FNEND
5310   !
5320   !************************************************************************
5330   !
5340   DEF FNTrap_level
5350       RETURN VAL(SYSTEM$("SYSTEM PRIORITY"))+1
5360   FNEND
5370   !
5380   !************************************************************************
5390   !
5400  SUB Config_and_poll
5410      COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
5420      COM /System/ System_memory
5430  !
5440  ! Find out what's out there.
5450  !
5460      ALLOCATE Device_list$(0:31)[20]
5470      ALPHA PEN 4
5480      KBD LINE PEN 3
5490      KEY LABELS PEN 5
5500      Clear_crt
5510      Netwrk_analyzer=0
5520      ALLOCATE Na_ident$[80]
5530      System_memory=VAL(SYSTEM$("AVAILABLE MEMORY")) ! How much memory for RAM-DISK
5540      PRINT "AVAILABLE MEMORY: ";System_memory;" BYTES"
5550      ON TIMEOUT 7,4 GOTO No_na      ! In case there is no network analyzer
5560 Is_na:  OUTPUT @Nwa;"FORM4; OUTPIDEN;"
5570         ENTER @Nwa_data2;Na_ident$
5580         IF POS(Na_ident$,"8510A") THEN Netwrk_analyzer=1
5590         IF POS(Na_ident$,"8510B") THEN Netwrk_analyzer=2
5600         IF POS(Na_ident$,"8720A") THEN Netwrk_analyzer=3
5610         IF POS(Na_ident$,"8720B") THEN Netwrk_analyzer=4
5620         IF POS(Na_ident$,"8753A") THEN Netwrk_analyzer=5
5630         IF POS(Na_ident$,"8753B") THEN Netwrk_analyzer=6
5640         LOCAL @Nwa
5650         PRINT
5660         PRINT Na_ident$
5670         PRINT Netwrk_analyzer
5680      !  Clear_crt
5690      PRINT
5700      PRINT
5710         IF Netwrk_analyzer=0 THEN
5720            !
5730            !
5740 No_na:     BEEP
5750            OFF CYCLE
5760            PRINT TABXY(1,5);"There is no active network analyzer on the HPIB bus."
5770            PRINT TABXY(1,6);"Please check connections, and press the RUN key."
5780            PRINT
5790            PRINT TABXY(1,7);"If you DO NOT want to use a network analyzer, press the
CONTINUE key."
5800            PAUSE
5810         END IF
5820         !
5830         !
5840 Check_hpib:  ! Check the rest of the bus
5850      ON TIMEOUT 7,.01 GOTO Nothing
5860      !
5870      FOR Device=700 TO 731
5880          DISP "Checking for device at address: ";Device
5890          Device_list$(Device-700)="NOTHING"
5900          ASSIGN @What_is_it TO Device
5910          Outcome=SPOLL(@What_is_it)
5920          Device_list$(Device-700)="SOMETHING"
5930          PRINT Device,"SOMETHING HERE","spoll: ";Outcome
5940          ASSIGN @What_is_it TO *
5950 Nothing: ! Skip to next device
5960      NEXT Device
5970      !
5980      OFF TIMEOUT 7
5990      ASSIGN @What_is_it TO *
6000      IF Device_list$(1)="SOMETHING" THEN
6010         DISP "Position the printer to Top-Of-Form and press CONTINUE..."
6020         PAUSE
6030         PRINTER IS PRT
6040         PRINT CHR$(27)&"&l1L"; ! Set Page Breaks
6050         Printer_flag=1
6060         PRINTER IS CRT
6070      END IF
6080      DEALLOCATE Na_ident$
6090      DEALLOCATE Device_list$(*)
6100      ABORT @Hpib
6110  SUBEXIT
6120  SUBEND
6130  !
6140  ! ***********************************************************************
6150  !
6160  SUB Hp_bus_init
6170    COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
6180    COM /Sys_1/ Freq$(*),Freq_cent(*),Freq_span(*),Gate_cent(*),Gate_span(*)
6190    COM /Sys_2/ Pol$(*),Polsw$(*)
6200    COM /System_config/ INTEGER Printer_flag,Debug_flag,Version$,Mode$,Out_type$,Soun
d$,Bell$,Target$,Ref_target$
6210    !
6220    !  This subroutine configures the HP-IB bus and presets the HP8510.
```

```
6230    !
6240    ASSIGN @Hpib TO 7
6250    ASSIGN @Nwa TO 716
6260    ASSIGN @Nwa_data1 TO 716;FORMAT OFF
6270    ASSIGN @Nwa_data2 TO 716;FORMAT ON
6280    ASSIGN @Relay TO 710
6290    REMOTE @Hpib
6300    ABORT @Hpib
6310    CLEAR @Nwa
6320    IF Debug_flag=1 THEN OUTPUT @Nwa;"DEBUON;"
6330    IF Debug_flag=0 THEN
6340      OUTPUT @Nwa;"DEBUOFF;"
6350      OUTPUT @Nwa;"TITL """&Freq$(2)&" BAND   """
6360    END IF
6370  SUBEND
6380  !
6390  !**********************************************************************
6400  !
6410  SUB Series_init
6420  COM /System_config/ INTEGER Printer_flag,Debug_flag,Version$,Mode$,Out_type$,Sound
$,Bell$,Target$,Ref_target$
6430    DIM Input$[80]
6440    !
6450    !  This subroutine prints a header for the printout and sets the system
6460    !  date and time.
6470    !
6480    IF Printer_flag=1 THEN PRINTER IS PRT
6490    PRINT CHR$(12)
6500    Set_clock
6510 !  LINPUT "ENTER MEASUREMENT SERIES TITLE",Input$
6520 !  Preface$="*"&RPT$(" ",9)
6530 !  PRINT RPT$("*",70)
6540 !  PRINT Preface$&Input$
6550 !  LINPUT "ENTER OPERATOR NAME",Input$
6560 !  PRINT Preface$&Input$
6570    PRINTER IS CRT
6580    PRINT
6590    PRINT
6600    PRINT Preface$&"MEASUREMENT SERIES STARTED AT "&TIME$(TIMEDATE)
6610    PRINTER IS CRT
6620  SUBEND
6630  !
6640  !**********************************************************************
6650  !
6660  SUB Set_clock
6670      OPTION BASE 1
6680      INTEGER I
6690      DIM Chrono$[12],Month$(12)[3]
6700      Exec_key$=CHR$(255)&CHR$(88)
6710      READ Month$(*)
6720      DATA "JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP","OCT","NOV","DEC"
6730      OUTPUT KBD;"SCRATCH KEY "&Exec_key$;
6740      Clear_crt
6750      PRINT "                              Current system date: ";DATE$(TIMEDATE)
6760      PRINT "                              Current system time: ";TIME$(TIMEDATE)
6770      PRINT
6780 Ask: LINPUT "Enter date and time (YYMMDDHHMMss) :",Chrono$
6790      IF Chrono$="" AND DATE$(TIMEDATE)<>" 1 Mar 1900" THEN
6800         Clear_crt
6810         SUBEXIT
6820      END IF
6830      Year$=VAL$(1900+VAL(Chrono$[1,2]))
6840      IF (VAL(Chrono$[3,4])<=0 OR VAL(Chrono$[3,4])>12) THEN
6850         BEEP
6860         PRINT "Incorrect month value."
6870         GOTO Ask
6880      END IF
6890      Year$=Month$(VAL(Chrono$[3,4]))&" "&Year$
6900      Year$=Chrono$[5,6]&" "&Year$
6910      SET TIMEDATE (DATE(Year$))
6920      IF (VAL(Chrono$[7,8]))>23 THEN
6930         BEEP
6940         PRINT "Incorrect hour value."
6950         GOTO Ask
6960      END IF
6970      Day$=Chrono$[7,8]&":"
6980      IF VAL(Chrono$[9,10])>59 THEN
6990         BEEP
7000         PRINT "Incorrect minute value."
7010         GOTO Ask
7020      END IF
7030      Day$=Day$&Chrono$[9,10]&":"
7040      IF (LEN(Chrono$)>10 AND LEN(Chrono$)=12) THEN
7050         IF VAL(Chrono$[11,12])>59 THEN
7060            BEEP
7070            PRINT "Incorrect seconds value."
7080            GOTO Ask
7090         END IF
7100         Day$=Day$&Chrono$[11,12]
7110      ELSE
7120         Day$=Day$&"00"
7130      END IF
7140      SET TIME TIME(Day$)
7150      Clear_crt
7160   SUBEXIT
7170   SUBEND
7180   !
7190   !**********************************************************************
7200   !
7210   SUB Fix_error
7220   SELECT ERRN
7230   CASE ELSE
7240         PRINTER IS CRT
7250         PRINT "ERROR ";ERRN
7260         PRINT ERRM$
7270         PRINT " PROGRAM IS PAUSED.  FIX ERROR, IF POSSIBLE, AND CONTINUE."
7280         PAUSE
7290   END SELECT
7300   SUBEND
7310   !
7320   !**********************************************************************
7330   !
7340  SUB Clear_crt(OPTIONAL INTEGER Start_line,Num_of_lines)
7350      !
7360      INTEGER I
7370      DIM Clear_line$[80]
7380      Clear_line$=""
7390      IF NPAR=0 THEN
7400         OUTPUT KBD;CHR$(255)&CHR$(75);
7410      ELSE
7420         PRINT TABXY(1,Start_line);"";RPT$(Clear_line$,Num_of_lines)
7430         PRINT TABXY(1,Start_line);"";
7440         SUBEXIT
7450      END IF
7460  SUBEND
7470  !
7480  !**********************************************************************
7490  !
```

```
7500  SUB Print_banner1
7510      Clear_crt
7520      PRINT
7530      PRINT
7540      PRINT TABXY(3,16);"**********************************************************"
7550      PRINT TABXY(4,16);"**                                                     **"
7560      PRINT TABXY(5,16);"**                     LCX                             **"
7570      PRINT TABXY(6,16);"**     UNIVERSITY OF MICHIGAN RADIATION LAB            **"
7580      PRINT TABXY(7,16);"**          L/C/X MEASUREMENT PROGRAM                  **"
7590      PRINT TABXY(8,16);"**                (VERSION 8.0)                        **"
7600      PRINT TABXY(9,16);"**                                                     **"
7610      PRINT TABXY(10,16);"**              May 21, 1990                         **"
7620      PRINT TABXY(11,16);"**                                                     **"
7630      PRINT TABXY(12,16);"**********************************************************"
7640  SUBEXIT
7650  SUBEND
7660  !
7670  !*************************************************************************
7680  !
7690   SUB Print_banner2
7700   PRINT "Don't use Print_banner2."
7710   SUBEND
7720   !
7730   !*************************************************************************
7740   !
7750   SUB Print_banner3
7760   PRINT "Don't use Print_banner3."
7770   SUBEND
7780  !
7790  !*************************************************************************
7800  !
7810   SUB Print_banner4
7820   COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
7830   COM /Constants/ Vel,Zero(*),Exec_key$
7840   COM /System_config/ INTEGER Printer_flag,Debug_flag,Version$,Mode$,Out_type$,Soun
d$,Bell$,Target$,Ref_target$
7850   COM /Sys_1/ Freq$(*),Freq_cent(*),Freq_span(*),Gate_cent(*),Gate_span(*)
7860   COM /Sys_2/ Pol$(*),Polsw$(*)
7870   COM /Sys_3/ INTEGER F_disp,P_disp
7880   COM /Sys_4/ Drive_a$,Drive_b$,Drive_c$,INTEGER Preamble,Bytes
7890   COM /Sys_5/ INTEGER Nskip,Ndata
7900   COM /Sys_6/ Ref_angle,Angle,Angle$,Beam(*),INTEGER Npts,Ntrace,Average_factor
7910   COM /Sys_7/ INTEGER Meas_flag(*)
7920       !
7930       !
7940    OFF KEY
7950    Clear_crt
7960    PRINT
7970    PRINT
7980    PRINT "           PARAMETER                          CURRENT VALUE"
7990    PRINT
8000    PRINT "                          FREQUENCY              ";
8010    FOR F=1 TO 3
8020      IF Meas_flag(F) THEN PRINT Freq$(F)&" ";
8030    NEXT F
8040    PRINT "      "
8050    PRINT "      "
8060    PRINT "                          ANTENNA ANGLE          "&Angle$
8070    PRINT "                          TARGET TYPE            "&Target$
8080    PRINT "                          MEASUREMENT MODE       "&Mode$&"      "
8090  !  PRINT "                          CURRENT DISPLAY        "&Freq$(F_disp)&" "&Pol$(P_d
isp);
8100    PRINT "      "
8110    PRINT "                          # OF TRACES/SET        ";Ntrace
```

```
8120    PRINT "                          # OF SAMPLE POINTS     ";Npts
8130    PRINT "                          # OF DATA POINTS       ";Ndata
8140    PRINT "                           (to be stored)        "
8150    PRINT "                          # OF AVERAGES          ";Average_factor
8160  SUBEXIT
8170  SUBEND
8180  !
8190  !*************************************************************************
8200  !
8210  SUB Store_file(COMPLEX Matrix(*),File_type$,Filename$,INTEGER F)
8220      !
8230      COM /Sys_1/ Freq$(*),Freq_cent(*),Freq_span(*),Gate_cent(*),Gate_span(*)
8240      COM /Sys_2/ Pol$(*),Polsw$(*)
8250      COM /Sys_5/ INTEGER Nskip,Ndata
8260      COM /Sys_6/ Ref_angle,Angle,Angle$,Beam(*),INTEGER Npts,Ntrace,Average_factor
8270      COM /Sys_7/ INTEGER Meas_flag(*)
8280      COM /System_config/ INTEGER Printer_flag,Debug_flag,Version$,Mode$,Out_type$,So
und$,Bell$,Target$,Ref_target$
8290      !
8300      !
8310      INTEGER Records_per_set,T
8320      REAL Bytes_per_set
8330      DIM Suffix$[2]
8340      ALLOCATE COMPLEX Trace(Ndata)
8350      !
8360      !
8370      DISP "Saving file."
8380      SELECT File_type$
8390      CASE ="SKY"  ! Sky data.
8400              Bytes_per_set=16*Ndata
8410              Records_per_set=4*SUM(Meas_flag)*Ntrace
8420            IF SUM(Meas_flag)=3 THEN
8430              Suffix$="SA"
8440            ELSE
8450             FOR F=1 TO 3
8460               IF Meas_flag(F)=1 THEN
8470                 Mf=F
8480               END IF
8490             NEXT F
8500             Suffix$="S"&Freq$(Mf)
8510            END IF
8520              GOSUB Save_hpux
8530 !            GOSUB Save_traces
8540            !
8550            !
8560      CASE ="REF"
8570           Bytes_per_set=16*Ndata
8580           Records_per_set=4*SUM(Meas_flag)*Ntrace
8590            IF SUM(Meas_flag)=3 THEN
8600              Suffix$="RA"
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$="R"&Freq$(Mf)
8680            END IF
8690              GOSUB Save_hpux
8700 !            GOSUB Save_traces
8710            !
8720            !
8730      CASE ="MNT"
8740           Bytes_per_set=16*Ndata
```

```
8750          Records_per_set=4*SUM(Meas_flag)*Ntrace
8760           IF SUM(Meas_flag)=3 THEN
8770             Suffix$="MA"
8780           ELSE
8790            FOR F=1 TO 3
8800              IF Meas_flag(F)=1 THEN
8810                Mf=F
8820              END IF
8830            NEXT F
8840            Suffix$="M"&Freq$(Mf)
8850           END IF
8860             GOSUB Save_hpux
8870 !           GOSUB Save_traces
8880             !
8890             !
8900      CASE ="GND"
8910          Bytes_per_set=16*Ndata
8920          Records_per_set=Ntrace*4*SUM(Meas_flag)
8930           IF SUM(Meas_flag)=3 THEN
8940             Suffix$="GA"
8950           ELSE
8960            FOR F=1 TO 3
8970              IF Meas_flag(F)=1 THEN
8980                Mf=F
8990              END IF
9000            NEXT F
9010            Suffix$="G"&Freq$(Mf)
9020           END IF
9030             GOSUB Save_hpux
9040 !           GOSUB Save_traces
9050      END SELECT
9060      DEALLOCATE Trace(*)
9070      SUBEXIT
9080          !
9090          !
9100 Save_averaged: !
9110                ! Save the reference data file.
9120                !
9130     IF NOT Debug_flag THEN
9140        CREATE BDAT Filename$&Suffix$&Drive_c$,Records_per_set,Bytes_per_set
9150     END IF
9160     Base_record=0
9170     FOR F=1 TO 3
9180     IF Meas_flag(F)=1 THEN
9190     IF Debug_flag THEN
9200        ASSIGN @Disc TO PRT
9210        OUTPUT @Disc;"FILE: ",Filename$,Suffix$
9220        OUTPUT @Disc USING Image_1;Version$,Freq_cent(F),Freq_span(F)
9230        OUTPUT @Disc USING Image_2;Ndata,Average_factor
9240        OUTPUT @Disc USING Image_3;Ref_target$,T
9250        FOR P=1 TO 4
9260            OUTPUT @Disc USING Image_4;Pol$(P),Gate_cent(F),Gate_span(F)
9270            MAT Trace= Matrix(1,P,*)
9280            OUTPUT @Disc;Trace(*)
9290        NEXT P
9300     ELSE
9310        ASSIGN @Disc TO Filename$&Suffix$&Drive_c$;FORMAT OFF
9320        OUTPUT @Disc,Base_record+1;Version$,Freq_cent(F),Freq_span(F)
9330        OUTPUT @Disc,Base_record+1;Ndata,Average_factor
9340        OUTPUT @Disc,Base_record+1;Ref_target$,T
9350        FOR P=1 TO 4
9360            OUTPUT @Disc,Base_record+P;Pol$(P),Gate_cent(F),Gate_span(F)
9370            MAT Trace= Matrix(1,P,*)
9380            OUTPUT @Disc,Base_record+P;Trace(*)
9390        NEXT P
9400        Base_record=Base_record+4
9410     END IF
9420     END IF
9430     NEXT F
9440     ASSIGN @Disc TO *
9450 RETURN
9460  !
9470  !---------------------------------------------------------------------
9480  !
9490 Save_hpux: !
9500                ! Save data in HP-UX format.
9510                !
9520     IF NOT Debug_flag THEN
9530        CREATE Filename$&Suffix$&Drive_c$,240000
9540     END IF
9550     IF Debug_flag THEN
9560        ASSIGN @Disc TO PRT
9570        FOR T=1 TO Ntrace
9580          FOR F=1 TO 3
9590           IF Meas_flag(F)=1 THEN
9600            FOR P=1 TO 4
9610                MAT Trace= Matrix(T,P,*)
9620                OUTPUT @Disc;Trace(*)
9630            NEXT P
9640           END IF
9650          NEXT F
9660         NEXT T
9670     ELSE
9680        ASSIGN @Disc TO Filename$&Suffix$&Drive_c$;FORMAT ON
9690        FOR T=1 TO Ntrace
9700          FOR F=1 TO 3
9710          IF Meas_flag(F)=1 THEN
9720            FOR P=1 TO 4
9730                MAT Trace= Matrix(T,P,*)
9740                 OUTPUT @Disc;Trace(*)
9750             NEXT P
9760           END IF
9770          NEXT F
9780        NEXT T
9790     END IF
9800     ASSIGN @Disc TO *
9810  RETURN
9820  !
9830  !---------------------------------------------------------------------
9840  !
9850 Save_traces: !
9860                ! Save the ground target data file.
9870                !
9880     IF NOT Debug_flag THEN
9890         CREATE BDAT Filename$&Suffix$&Drive_c$,Records_per_set,Bytes_per_set
9900        Base_record=0
9910     END IF
9920     IF Debug_flag THEN
9930        ASSIGN @Disc TO PRT
9940        OUTPUT @Disc;"FILE: ";Filename$;Suffix$
9950        OUTPUT @Disc USING Image_5;Ndata,Ntrace
9960        OUTPUT @Disc USING Image_3;Target$
9970        FOR T=1 TO Ntrace
9980          FOR F=1 TO 3
9990           IF Meas_flag(F)=1 THEN
10000           OUTPUT @Disc USING Image_1;Version$,Freq_cent(F),Freq_span(F)
10010           FOR P=1 TO 4
10020               OUTPUT @Disc USING Image_4;Pol$(P),Gate_cent(F),Gate_span(F),T
```

```
10030              MAT Trace= Matrix(T,P,*)
10040              OUTPUT @Disc;Trace(*)
10050          NEXT P
10060         END IF
10070        NEXT F
10080       NEXT T
10090    ELSE
10100       ASSIGN @Disc TO Filename$&Suffix$&Drive_c$;FORMAT OFF
10110 !      OUTPUT @Disc,1;Ndata,Ntrace
10120 !      OUTPUT @Disc,1;Target$
10130       FOR T=1 TO Ntrace
10140         FOR F=1 TO 3
10150         IF Meas_flag(F)=1 THEN
10160 !         OUTPUT @Disc,Base_record+1;Version$,Freq_cent(F),Freq_span(F)
10170           FOR P=1 TO 4
10180 !              OUTPUT @Disc,Base_record+P;Pol$(P),Gate_cent(F),Gate_span(F),T
10190              MAT Trace= Matrix(T,P,*)
10200               OUTPUT @Disc,Base_record+P;Trace(*)
10210            NEXT P
10220           Base_record=Base_record+4
10230          END IF
10240         NEXT F
10250       NEXT T
10260    END IF
10270    ASSIGN @Disc TO *
10280 RETURN
10290 !
10300 !------------------------------------------------------------------------
10310 !
10320 Image_1:IMAGE (1X,12A,5X,"FREQ CENTER: ",2D.4D,5X,"FREQ SPAN: ",2D.4D)
10330 Image_2:IMAGE ("NUMBER OF POINTS: ",5D,5X,"NUMBER OF AVERAGES: ",5D)
10340 Image_3:IMAGE ("TARGET: ",30A,"GATING TARGET TYPE: ",2D)
10350 Image_4:IMAGE ("POLARIZATION: ",2A,5X,"GATE CENTER: ",SD.14DE,/,5X,"GATE SPAN: ",S
D.14DE,"TRACE: ",3D)
10360 Image_5:IMAGE ("NUMBER OF POINTS: ",5D,5X,"NUMBER OF TRACES: ",5D)
10370 Image_6:IMAGE (5X,SD.14DE,5X,SD.14DE)
10380 SUBEND
10390 !
10400 !**********************************************************************
10410 !
10420 SUB Freq_set(INTEGER Ifreq)
10430 COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
10440 COM /Sys_1/ Freq$(*),Freq_cent(*),Freq_span(*),Gate_cent(*),Gate_span(*)
10450   !
10460   !  This subroutine sets the transmit frequency for the HP8753.
10470   !
10480   IF Ifreq=1 THEN
10490       OUTPUT @Nwa;"POWE0"
10500   ELSE
10510       OUTPUT @Nwa;"POWE-10"
10520   END IF
10530   SELECT Netwrk_analyzer
10540   CASE =3,=4,=5,=6
10550        OUTPUT @Nwa;"TIMDTRANOFF;"
10560   CASE =1,=2
10570        OUTPUT @Nwa;"FREQ;"
10580   END SELECT
10590   OUTPUT @Nwa;"CENT "&VAL$(Freq_cent(Ifreq))&" GHZ;"
10600   OUTPUT @Nwa;"SPAN "&VAL$(Freq_span(Ifreq))&" GHZ;"
10610 SUBEND
10620 !
10630 !**********************************************************************
10640 !
10650 SUB Freq_sw(INTEGER Ifreq)
10660 COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
10670   SELECT Ifreq
10680   CASE 1
10690     OUTPUT @Relay;"?*A2B1"
10700   CASE 2
10710     OUTPUT @Relay;"?*A1B2"
10720   CASE 3
10730     OUTPUT @Relay;"?*B12"
10740   END SELECT
10750   WAIT .1
10760 SUBEND
10770 !
10780 !**********************************************************************
10790 !
10800 SUB Pol_sw(INTEGER Ifreq,Ipol)
10810     COM /Paths/ @Nwa,@Nwa_data1,@Nwa_data2,Netwrk_analyzer,@Hpib,@Relay
10820     COM /Sys_1/ Freq$(*),Freq_cent(*),Freq_span(*),Gate_cent(*),Gate_span(*)
10830     COM /Sys_2/ Pol$(*),Polsw$(*)
10840   !
10850   !  This subroutine sets the transmit and receive polarization by
10860   !  sending the proper command over the HPIB to the polarization
10870   !  relays.
10880   !
10890   OUTPUT @Relay;Polsw$(Ifreq,Ipol)
10900   OUTPUT @Nwa;"TITL "" "&Freq$(Ifreq)&" BAND - "&Pol$(Ipol)&""""
10910   WAIT .1
10920 SUBEND
10930 !
10940 !**********************************************************************
10950 !
10960 SUB Rotate_target
10970 OPTION BASE 1
10980 COM /Com4/ INTEGER Rotation_state,REAL Inc_angle,Current_angle,Start_angle,Stop_ang
le,Old_home_angle,INTEGER Sets_per_pos
10990 COM /Status/ INTEGER Sc,Connect_flg,E_flg,Debug_flg,Response$[80]
11000 INTEGER Fs_flag,Ss_flag,Speed,Imc_status,Confirm_answer
11010    !
11020    !
11030    Confirm_answer=1
11040    Imc_status=0
11050    Debug_flg=0
11060    Fs_flag=-1
11070    Ss_flag=-1
11080    Clear_crt(3,16)
11090    !
11100    !
11110    SELECT Rotation_state
11120    CASE =-1
11130         IF FNAsk("Do you wish to use the rotator?") THEN
11140            Connect_flg=0
11150            GOSUB Init_imc
11160            GOSUB Init_graph_pos
11170            GOSUB Manual_loop
11180            PRINT "Set Auto Mode Please....."
11190         ELSE
11200            Rotation_state=0
11210            GCLEAR
11220            GRAPHICS OFF
11230         END IF
11240    CASE =0
11250         SUBEXIT
11260    CASE =1
11270         GOSUB Check_position
11280         GOSUB Print_angles
```

```
11290         GOSUB Manual_loop
11300     CASE =2
11310         GOSUB Check_position
11320         GOSUB Auto
11330     CASE =3
11340         GOSUB Check_position
11350         GOSUB Manual_loop
11360         GOSUB Auto
11370     CASE =4
11380         GOSUB Check_position
11390         GOSUB Go_home
11400     CASE =5
11410         GOSUB Check_position
11420         Rotation_state=1  ! Switch to manual mode.
11430     END SELECT
11440     SUBEXIT
11450     !
11460     !
11470 Init_imc: ! Initialize the IMC unit.
11480     GOSUB Check_4_fault
11490     PRINT TABXY(1,3);"INITIALIZING IMC"
11500     Clear_crt(4,15)
11510     Comm("4WB")           ! Set warm boot (clear flags).
11520     PRINT TABXY(1,4);"WB"
11530     Comm("4EB")           ! Clear IMC buffer.
11540     PRINT TABXY(1,4);"EB"
11550     Encoder_ratio=4096  ! 32000
11560     Comm("4ER"&VAL$(Encoder_ratio))        ! Load encoder ratio.
11570     PRINT TABXY(1,4);"ER"&VAL$(Encoder_ratio)
11580     IF FNAsk("Do you wish to set home at the current position?") THEN
11590        Comm("4RS",Confirm_answer)
11600        ENTER Response$;Old_home_angle
11610        Old_home_angle=Old_home_angle/93.3
11620        Comm("4PIZ0")          ! Set IMC at 0.
11630        PRINT TABXY(1,4);"PIZ"&RPT$(" ",LEN(VAL$(Encoder_ratio)))
11640        Comm("4PIA0")          ! Set IMC at 0.
11650        PRINT TABXY(1,4);"PIA"
11660        Current_angle=0
11670     END IF
11680     Comm("4SP100")          ! Set speed to (50pps).
11690     PRINT TABXY(1,4);"SP "&RPT$(" ",LEN(VAL$(Encoder_ratio)))
11700     Comm("4AC500")          ! Set acceleration (500pps^2).
11710     PRINT TABXY(1,4);"AC "
11720     Comm("4DC500")          ! Set deceleration (500pps^2).
11730     PRINT TABXY(1,4);"DC "
11740     GOSUB Check_position
11750     Rotation_state=1
11760     Clear_crt
11770 !
11780 !
11790     PRINT TABXY(1,4);"DONE INITIALIZING IMC"
11800     PRINT TABXY(1,5);"Turntable currently in manual mode."
11810     PRINT TABXY(1,6)
11820 Print_angles:!
11830     PRINT TABXY(1,7);"Current  angle is: ";Current_angle;" degrees."
11840     PRINT TABXY(1,8);"Starting angle is: ";Start_angle;" degrees."
11850     PRINT TABXY(1,9);"Stopping angle is: ";Stop_angle;" degrees."
11860 RETURN
11870 !
11880 !
11890 Manual_loop:! Main activation loop.
11900     LOOP
11910       ON KEY 0 LABEL "FAST SLEW CW  ",FNTrap_level GOSUB Fs_cw
11920       ON KEY 1 LABEL "FAST SLEW CCW ",FNTrap_level GOSUB Fs_ccw
11930       ON KEY 5 LABEL "SLOW SLEW CW  ",FNTrap_level GOSUB Ss_cw
11940       ON KEY 6 LABEL "SLOW SLEW CCW ",FNTrap_level GOSUB Ss_ccw
11950       ON KEY 2 LABEL "MANUAL CONTROL",FNTrap_level GOSUB Manual
11960       ON KEY 3 LABEL "TARGET GO HOME",FNTrap_level GOSUB Go_home
11970       ON KEY 4 LABEL "STOP ROTATION ",FNTrap_level GOSUB Stop_turn
11980       ON KEY 7 LABEL "SET AUTO MODE ",FNTrap_level GOSUB Set_auto
11990       ON KEY 8 LABEL "SET TARGET HOME",FNTrap_level GOSUB Set_position
12000       ON KEY 9 LABEL "RETURN        ",FNTrap_level GOTO Quit
12010       GOSUB Check_position
12020     END LOOP
12030 !
12040 !------------------------------------------------------------------------
12050 !
12060 Fs_cw:! Fast slew clockwise.
12070       IF Fs_flag<0 THEN
12080          Comm("4SP500")
12090          Comm("4SFN")
12100          Fs_flag=-1*Fs_flag
12110          Clear_crt(3,15)
12120          PRINT TABXY(1,15);"ROTATING CW (FAST)"
12130       ELSE
12140          Comm("4ST")
12150          Fs_flag=-1*Fs_flag
12160          Clear_crt(3,15)
12170          PRINT TABXY(1,15);"ROTATION STOPPED"
12180          GOSUB Check_position
12190       END IF
12200 RETURN
12210 !
12220 !------------------------------------------------------------------------
12230 !
12240 Fs_ccw:! Fast slew counterclockwise.
12250       IF Fs_flag<0 THEN
12260          Comm("4ST")
12270          Comm("4SP500")
12280          Comm("4SRN")
12290          Fs_flag=-1*Fs_flag
12300          Clear_crt(3,10)
12310          PRINT TABXY(1,15);"ROTATING CCW (FAST)"
12320       ELSE
12330          Comm("4ST")
12340          Fs_flag=-1*Fs_flag
12350          Clear_crt(3,15)
12360          PRINT TABXY(1,15);"ROTATION STOPPED"
12370          GOSUB Check_position
12380       END IF
12390 RETURN
12400 !
12410 !------------------------------------------------------------------------
12420 !
12430 Ss_cw:! Slow slew clockwise.
12440       IF Ss_flag<0 THEN
12450          Comm("4ST")
12460          INPUT "Speed?",Sp
12470          Comm("4SP"&VAL$(INT(Sp)))
12480          Comm("4SFN")
12490          Ss_flag=-1*Ss_flag
12500          Clear_crt(3,15)
12510          PRINT TABXY(1,15);"ROTATING CW (SLOW)"
12520       ELSE
12530          Comm("4ST")
12540          Ss_flag=-1*Ss_flag
12550          Clear_crt(3,15)
12560          PRINT TABXY(1,15);"ROTATION STOPPED"
```

```
12570        GOSUB Check_position
12580     END IF
12590 RETURN
12600 !
12610 !-----------------------------------------------------------------------
12620 !
12630 Ss_ccw:! Slow slew counterclockwise.
12640     IF Ss_flag<0 THEN
12650     INPUT "Speed?",Sp
12660        Comm("4SP"&VAL$(INT(Sp)))
12670        Comm("4SRN")
12680        Ss_flag=-1*Ss_flag
12690        Clear_crt(3,15)
12700        PRINT TABXY(1,15);"ROTATING CCW (SLOW)"
12710     ELSE
12720        Comm("4ST")
12730        Ss_flag=-1*Ss_flag
12740        Clear_crt(3,15)
12750        PRINT TABXY(1,15);"ROTATION STOPPED"
12760        GOSUB Check_position
12770     END IF
12780 RETURN
12790 !
12800 !-----------------------------------------------------------------------
12810 !
12820 Manual: INPUT "ANGLE (IN DEGREES)=?",Inc_angle
12830        INPUT "SPEED? (~100-~500 RECOMMENDED)",Speed
12840        Comm("4SP"&VAL$(Speed))
12850 Auto:   SELECT Rotation_state
12860        CASE =4
12870             GOSUB Go_home
12880             Rotation_state=2
12890             GOTO Auto
12900        CASE ELSE
12910             Angl2=Inc_angle*93.3
12920             Angl1=INT(Angl2)
12930             IF Angl2-Angl1>=.5 THEN Angl1=Angl1+1
12940        !    Current_angle=Current_angle+Inc_angle
12950             Inc_angle$=VAL$(Angl1)
12960             Comm("4IM"&Inc_angle$)
12970             Comm("4RFI")
12980        END SELECT
12990        Imc_status=0
13000        Clear_crt(3,7)
13010        PRINT TABXY(1,14);"ROTATING TARGET, PLEASE WAIT."
13020        !
13030        !
13040        WHILE NOT BIT(Imc_status,0)            ! Wait for motor to stop.
13050              Comm("4RS",Confirm_answer)
13060              ENTER Response$;Imc_status
13070              PRINT TABXY(1,15);DVAL$(Imc_status,2)
13080              GOSUB Check_position
13090              WAIT 1
13100        END WHILE
13110        Imc_status=0
13120        !
13130        !
13140        Clear_crt(3,16)
13150        PRINT TABXY(1,16);"CURRENT TARGET POSITION IS ";Current_angle;" DEGREES.
             "
13160        WAIT 2  ! Wait for target settling.
13170 RETURN
13180 !
13190 !-----------------------------------------------------------------------
```

```
13200 !
13210 Stop_turn:Comm("4ST")
13220           WHILE NOT BIT(Imc_status,0)          ! Wait for motor to stop.
13230                 Comm("4RS",Confirm_answer)
13240                 ENTER Response$;Imc_status
13250                 WAIT .1
13260           END WHILE
13270           Clear_crt(3,16)
13280           PRINT TABXY(1,15);"ROTATION STOPPED"
13290           GOSUB Check_position
13300           Imc_status=0
13310 RETURN
13320 !
13330 !-----------------------------------------------------------------------
13340 !
13350 Set_auto: Comm("4SP500")
13360           GOSUB Check_position
13370           Clear_crt(3,16)
13380           PRINT TABXY(1,3);"Current starting  angle: ";Start_angle;" degrees"
13390           PRINT TABXY(1,4);"Current increment angle: ";Inc_angle;" degrees"
13400           PRINT TABXY(1,5);"Current stopping  angle: ";Stop_angle;" degrees"
13410           PRINT TABXY(1,6);"Current rotation  speed: ";Speed
13420           PRINT TABXY(1,7);RPT$(" ",80)
13430           PRINT TABXY(1,8);"Rotator  positioned  at: ";Current_angle;" degrees"
13440           INPUT "Enter starting angle value (degrees): ",Start_angle
13450           INPUT "Enter increment angle (degrees): ",Inc_angle
13460           INPUT "Enter stopping angle (degrees): ",Stop_angle
13470           INPUT "Enter rotation speed of target (~500 recommended): ",Speed
13480           Speed=INT(Speed)
13490           Comm("4SP"&VAL$(Speed))
13500           IF ABS(Start_angle-Current_angle)>.1 THEN
13510              PRINT TABXY(1,9);RPT$(" ",80)
13520              PRINT TABXY(1,10);"Rotating target to starting angle..."
13530              Temp_angle=Inc_angle
13540              Inc_angle=Start_angle-Current_angle
13550              GOSUB Auto
13560              Inc_angle=Temp_angle
13570           END IF
13580           Rotation_state=2
13590           Clear_crt
13600           PRINT TABXY(1,20);"Turntable is in automatic mode. (press the RETURN soft
key)"
13610 RETURN
13620 !
13630 !-----------------------------------------------------------------------
13640 !
13650 Set_position:INPUT "LOCK IN CURRENT TARGET POSITION AS REFERENCE POSITION?",Yn$
13660            IF Yn$="Y" OR Yn$="y" THEN
13670               Comm("4RS",Confirm_answer)
13680               ENTER Response$;Old_home_angle
13690               Old_home_angle=Old_home_angle/93.3
13700               Comm("4PIA0") ! Set absolute position to zero.
13710               Comm("4PIZ0") ! Set incremental postition to zero.
13720               Current_angle=0
13730            ELSE
13740               PRINT "POSITION WAS NOT SET."
13750            END IF
13760 RETURN
13770 !
13780 !-----------------------------------------------------------------------
13790 !
13800 Go_home: IF Speed<200 THEN Speed=200
13810         Comm("4SP"&VAL$(Speed))
13820         Comm("4AM0") ! Move to zero absolute position.
```

```
13830          Comm("4RAN")  ! Initiate movement.
13840          Comm("4MW")   ! Make sure the move is completed.
13850          Imc_status=0
13860          Clear_crt(3,15)
13870          PRINT TABXY(1,14);"ROTATING TARGET TO HOME POSITION, PLEASE WAIT."
13880          WHILE NOT (BIT(Imc_status,0) AND BIT(Imc_status,5))
13890               GOSUB Check_status
13900               PRINT TABXY(1,15);"CURRENT STATUS: ";DVAL$(Imc_status,2)
13910               GOSUB Check_position
13920               WAIT .1
13930          END WHILE
13940          Clear_crt(3,16)
13950          PRINT TABXY(1,15);"TARGET AT HOME POSITION."
13960          GOSUB Check_position
13970          Imc_status=0
13980 RETURN
13990 !
14000 !----------------------------------------------------------------------
14010 !
14020 Check_status:! Keep an eye on the Whedco controller status.
14030          Comm("4RS",Confirm_answer)
14040          ENTER Response$;Imc_status
14050 RETURN
14060 !
14070 !----------------------------------------------------------------------
14080 !
14090 Check_position:! Get the current turnstile position in degrees.
14100          Comm("4RP",Confirm_answer)
14110          ENTER Response$;Motor_position
14120          Current_angle=Motor_position/93.3
14130    !     Current_angle=Current_angle+Inc_angle
14140          PRINT TABXY(1,16);"CURRENT TARGET POSITION IS ";Current_angle;" DEGREES.
                "
14150          GOSUB Draw_positions
14160 RETURN
14170 !
14180 !----------------------------------------------------------------------
14190 !
14200 Check_4_fault: ! Check the IMC for a fault condition and correct or
14210                ! notify the user if necessary.
14220                !
14230       Comm("4FC",Confirm_answer)
14240       ENTER Response$;Fault$
14250       SELECT Fault$
14260       CASE ="Power failure" ! Loss of power
14270            RETURN
14280       CASE ="Force DAC" ! Force DAC command was given
14290            BEEP
14300            PRINT "Force DAC command was given..."
14310            DISP "Press CONTINUE to resume..."
14320            PAUSE
14330       RETURN
14340       CASE ="Over-current" ! Over-current condition exists.
14350            BEEP
14360            PRINT "An over-current condition has been detected on the IMC."
14370            PRINT
14380            PRINT "Cycle the power to the IMC until the OV-CUR LED goes out"
14390            DISP "Press CONTINUE to reinitialize the IMC"
14400            PAUSE
14410            GOSUB Init_imc
14420       RETURN
14430       END SELECT
14440       RETURN
14450 !
```

```
14460 !----------------------------------------------------------------------
14470 !
14480 Init_graph_pos: ! Creates a graphical depiction of where the target is.
14490                 !
14500     GINIT
14510     GCLEAR
14520     GRAPHICS ON
14530     SHOW 0,100,0,100
14540     PENUP
14550     MOVE 90,70
14560     PEN 1        ! Draw circle
14570     POLYGON 12,360,360
14580     PENUP
14590     MOVE 90,70  ! Draw old home orientation.
14600     PEN 2
14610     DRAW 90+11*COS(Old_home_angle),70-11*SIN(Old_home_angle)
14620     PENUP
14630     MOVE 90,70  ! Draw current home orientation.
14640     PEN 4
14650     DRAW 90,58
14660     PENUP
14670     MOVE 90,70  ! Draw current target orienation.
14680     PEN 3
14690     X_pos=90+11*COS(Current_angle)
14700     Y_pos=70-11*SIN(Current_angle)
14710     DRAW X_pos,Y_pos
14720 RETURN
14730 !
14740 !----------------------------------------------------------------------
14750 !
14760 Draw_positions:  ! Draws out the angular orienations.
14770     MOVE 90,70    ! Draw old home orientation.
14780     PEN 2
14790     DRAW 90-11*SIN(Old_home_angle),70-11*COS(Old_home_angle)
14800     PENUP
14810     MOVE 90,70  ! Draw current home orientation.
14820     PEN 4
14830     DRAW 90,58
14840     PENUP
14850     DISABLE
14860     MOVE 90,70  ! Draw current target orienation.
14870     PEN -3
14880     DRAW X_pos,Y_pos
14890     MOVE 90,70
14900     PEN 3
14910     X_pos=90-11*SIN(Current_angle)
14920     Y_pos=70-11*COS(Current_angle)
14930     DRAW X_pos,Y_pos
14940     PENUP
14950     ENABLE
14960 RETURN
14970 !
14980 !----------------------------------------------------------------------
14990 !
15000 Quit: !
15010        SUBEXIT
15020 SUBEND
15030 !
15040 !*********************************************************************
15050 !
15060 SUB Comm(C$,OPTIONAL INTEGER Confirm_answer)
15070 !
15080 !  PROGRAM MODULE:  Comm
15090 !
```

```
15100 !  PURPOSE:         Modified version of the Comm module to be used
15110 !                   for direct two way communication with the WHEDCO
15120 !                   IMC stepping motor controller.
15130 !
15140 !  UPDATE:  3.0     Version 3.0 checks to see if the card being used
15150 !                   is the HP98628A (Datacomm) or the HP98626A (Serial).
15160 !                   Depending on which card is used, the appropriate
15170 !                   registers are selected.
15180 !
15190      OPTION BASE 1
15200      COM /Status/ INTEGER Sc,Connect_flg,E_flg,Debug_flg,Response$
15210      INTEGER Baud_rate,B,Num_chars,Response_flg,Index1
15220      DIM Input$[256],Term$[256],In$[256] BUFFER,From_232$[256]
15230      DIM Num_chars$[6],Num_ltrs$[6],Out$[256] BUFFER
15240      DIM White_print$[1],Crlf$[2]
15250      IF Debug_flg THEN PRINT TABXY(1,1);"ENTERING Comm "
15260      ON ERROR GOSUB Error
15270      !
15280      !
15290      !
15300      IF Connect_flg THEN After_init
15310      Sc=30
15320      ASSIGN @Find_it TO Sc;RETURN Outcome
15330      IF Outcome=0 THEN
15340         ASSIGN @Find_it TO *
15350         CONTROL Sc,0;1                ! Reset RS-232 interface.
15360         CONTROL Sc,3;1                ! Async link protocol.
15370         CONTROL Sc,0;1                ! Set Async toggle.
15380         CONTROL Sc,8;1+2              ! Set RTS and DTR lines.
15390         CONTROL Sc,16;0               ! Disable connection timeout.
15400         CONTROL Sc,17;0               ! Disable no activity timeout.
15410         CONTROL Sc,18;0               ! Disable NO CARRIER timeout.
15420         CONTROL Sc,19;0               ! Disable transmit timeout.
15430         CONTROL Sc,20;14              ! TX baud speed = 9600
15440         CONTROL Sc,21;14              ! RX baud speed = 9600
15450         CONTROL Sc,22;0               ! No handshake with Whedco.
15460         CONTROL Sc,23;0               ! No hardwired handshake.
15470         CONTROL Sc,34;2               ! 7 bits/character.
15480         CONTROL Sc,35;0               ! 1 stop bit.
15490         CONTROL Sc,36;1               ! ODD parity.
15500         Connect_flg=1
15510      ELSE
15520         Sc=8
15530         ASSIGN @Find_it TO *
15540         ASSIGN @Find_it TO Sc;RETURN Outcome
15550         IF Outcome<>0 THEN
15560            PRINT "RS-232 card not installed. Please install and reboot."
15570            ASSIGN @Find_it TO *
15580            STOP
15590         END IF
15600         ASSIGN @Find_it TO *
15610         RESET Sc
15620         CONTROL Sc,0;1               ! Reset the RS-232 interface.
15630         CONTROL Sc,3;Baud_rate       ! Set the baud rate.
15640         CONTROL Sc,4;8+2             ! UART 8 bits/char.  ODD parity.
15650         CONTROL Sc,5;3               ! UART DTR line active.
15660         CONTROL Sc,12;128+32+16      ! Disable CD,DSR,CTS
15670         STATUS Sc,3;B                ! Confirm speed to user.
15680         Connect_flg=1
15690      END IF
15700 After_init:!
15710      White_print$=CHR$(136)
15720      Crlf$=CHR$(13)&CHR$(10)
15730      PRINT CHR$(128)&CHR$(136);      ! Set up the screen.
```

```
15740      ASSIGN @Screen TO CRT
15750      ASSIGN @Kbd TO KBD
15760      ASSIGN @Rx TO BUFFER In$
15770      ASSIGN @Tx TO BUFFER Out$
15780      ASSIGN @Uart_out TO Sc
15790      ASSIGN @Uart_in TO Sc
15800      Response_flg=0                ! Reset command acknowledge flag.
15810      Response$=""                  ! Null out response string.
15820      !
15830      !
15840         ENABLE INTR Sc                ! Enable interrupt on card.
15850         TRANSFER @Tx TO @Uart_out;CONT ! Enable transfer buffers.
15860         TRANSFER @Uart_in TO @Rx
15870         ON INTR Sc,FNTrap_level GOSUB Read_loop ! Process card interrupts.
15880         IF C$<>"" THEN
15890            GOSUB Send_com             ! Send command out to controller.
15900         ELSE
15910            GOTO Quit                  ! If null command, exit quick.
15920         END IF
15930         !
15940         !
15950         !
15960 Wait_for_it:WHILE NOT Response_flg     ! Waiting for acknowledgement.
15970                   GOSUB Read_loop
15980                   IF NPAR=2 THEN       ! We are waiting for data to be
15990                                        ! sent by the Whedco controller.
16000                      LOOP
16010                            GOSUB Read_loop
16020                            IF (POS(Response$,"*")) THEN
16030                                Response$=Response$[POS(Response$,"*"),LEN(Response$
)]
16040                                Response_flg=1
16050                            END IF
16060                      EXIT IF ((Response_flg=1) AND (POS(Response$,Crlf$)))
16070                      END LOOP
16080                   ELSE
16090                      WHILE NOT ((POS(Response$,"*")) OR (POS(Response$,"?")))
16100                            GOSUB Read_loop
16110                      END WHILE
16120                      Index1=POS(Response$,"*")
16130                      IF Index1=0 THEN ! Must be a "?" (Whedco command error).
16140                         ! Must be a "?" (Whedco command error).
16150                         E_flg=1        ! Notify via error flag.
16160                         Response_flg=1
16170                      ELSE
16180                         ! Normal command interpretation.
16190                         E_flg=0
16200                         Response_flg=1
16210                      END IF
16220                   END IF
16230             END WHILE
16240         GOTO Quit
16250      !
16260      !
16270 Read_loop:! Read in serial data from Whedco.
16280           !
16290      STATUS @Rx,4;Num_chars               ! Number of characters to
16300      IF Num_chars=0 THEN RETURN           ! receive, if 0 try again.
16310      Num_chars$="#,"&VAL$(Num_chars)&"A"  ! Set up the IMAGE for ENTER.
16320      ENTER @Rx USING Num_chars$;From_232$       ! Transfer contents.
16330      Response$=Response$&From_232$              ! Build up dialogue.
16340      RETURN                                     ! Update pointers.
16350 !
16360 !
```

REFERENCES

[1] Ulaby, Fawwaz T. Lecture Notes on "Remote Sensing II: Radar", pg. 58. February 13, 1992.

[2] Sarabandi, Kamal, et al. "Performance Characterization of Polarimetric Active Radar Calibrators and a New Single Antenna Design", *IEEE Transactions on Antennas and Propagation,* Vol. 40, No. 10, October 1992.

[3] Burnside, Walter D. "An Aperture-Matched Horn Design", *IEEE Transactions on Antennas and Propagation*, Vol. AP-30, No. 4, July 1982.

[4] Balanis, Constantine A. *Antenna Theory: Analysis and Design*, New York: John Wiley and Sons, 1982.

[5] Sarabandi, Kamal, et al. "A Convenient Technique for Polarimetric Calibration of Radar Systems," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 28, pp. 1022-1033, 1990.