REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 30, 1997	Final, 1994-1997

4. TITLE AND SUBTITLE

Basic Research on Three-Dimensional (3D) Electromagnetic (EM) Methods for Imaging the Flow of Organic Fluids in the Subsurface

5. FUNDING NUMBERS

Grant Number
F49620-95-1-0004

6. AUTHOR(S)

Ben K. Sternberg
Steven L. Dvorak

AFOSR-TR-97

0198

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Arizona
Laboratory for Advanced Subsurface Imaging (LASI)
Tucson, AZ 85721-0012

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFOSR/PKA - USAF
110 Duncan Avenue, Room B115
Bolling AFB DC 20332-8050

NA

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release;
distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Currently there are no systems available which allow for economical and accurate subsurface imaging of remediation sites. In some cases, high-frequency ground penetrating radar (GPR) has been shown to be capable of accurately mapping the movement of contaminant plumes. Unfortunately, high-frequency GPR has much too limited a depth of penetration in many soils to be useful for a large number of environmental problems. Lower frequencies are needed in order to obtain a reasonable depth of penetration. During this project, we conducted research on a prototype 8 MHz null-field electromagnetic (EM) system to address this need. This project has been very successful in showing a promising new direction for high-resolution subsurface imaging. Our tests with a prototype Electromagnetic Sensitive Null Array Probe (EM-SNAP) showed that we were able to obtain very sensitive measurements over subsurface dielectric targets. Although more basic research must be done, this approach holds great promise for imaging the flow of organic fluids in the subsurface.

14. SUBJECT TERMS

Electromagnetics, geophysics, environmental, contamination

15. NUMBER OF PAGES
130

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

Basic Research on Three-Dimensional (3D) Electromagnetic (EM) Methods for Imaging the Flow of Organic Fluids in the Subsurface

Final Report
Air Force Office of Scientific Research
Grant No. F49620-95-1-0004

Principal Investigators:

Dr. Ben K. Sternberg
Department of Mining and Geological Engineering

Dr. Steven L. Dvorak
Department of Electrical and Computer Engineering

April 30, 1997

Laboratory for Advanced Subsurface Imaging
University of Arizona
Tucson, AZ 85721

DISCLAIMER
This work was sponsored by the Air Force Office of Scientific Research, USAF, under grant number F49620-95-1-0004. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

ABSTRACT

Currently there are no systems available which allow for economical and accurate subsurface imaging of remediation sites. In some cases, high-frequency ground penetrating radar (GPR) has been shown to be capable of accurately mapping the movement of contaminant plumes. Unfortunately, high-frequency GPR has much too limited a depth of penetration in many soils to be useful for a large number of environmental problems. Lower frequencies are needed in order to obtain a reasonable depth of penetration. During this project, we conducted research on a prototype 8 MHz null-field electromagnetic (EM) system to address this need.

This project has been very successful in showing a promising new direction for high-resolution subsurface imaging. Our tests with a prototype Electromagnetic Sensitive Null Array Probe (EM-SNAP) showed that we were able to obtain very sensitive measurements over subsurface dielectric targets. Although more basic research must be done, this approach holds great promise for imaging the flow of organic fluids in the subsurface.

Contents

1 EXECUTIVE SUMMARY

Currently there are no systems available which allow for economical and accurate subsurface imaging of remediation sites. In some cases, high-frequency ground penetrating radar (GPR) has been shown to be capable of accurately mapping the movement of contaminant plumes. Unfortunately, high-frequency GPR has much too limited a depth of penetration in many soils to be useful for a large number of environmental problems. Lower frequencies are needed in order to obtain a reasonable depth of penetration. Electromagnetic induction instruments have sufficient depth of penetration for environmental problems. These systems are effective for determining depths to layers (for example, depth to water table or depth to bedrock) and for detection of some buried targets. However, commercially-available EM induction instruments do not currently operate at high-enough frequencies to detect subsurface organic contaminant plumes.

Much of the recent work in EM induction and GPR has been concentrated on sophisticated methods to recognize subsurface targets (including contaminant plumes) in the data, including electromagnetic modeling (both numerical and analytical), pattern-recognition techniques (including neural networks), and identification of resonances. Unfortunately, these methods have met with limited success. This is not so much due to the failure of these sophisticated algorithms, but much more to the lack of truly-diagnostic information in the original data.

We therefore have been led to an alternate approach, i.e., focus on the front-end data acquisition in order to greatly improve the back-end data interpretation. In principle, electromagnetic (EM) methods are imbued with unlimited resolving power. If we could obtain data with unlimited precision at all frequencies, at all points on the surface of the earth, we should be able to detect and accurately map the extent of organic contaminant plumes in the subsurface environment. Unfortunately, the response from an organic contaminant plume may be minute and may occur over a limited frequency range and only within a small area. Our research, therefore, has involved the beginning steps toward development of a data-acquisition approach which enables the detection of minute EM responses similar to those associated with contaminant plumes. Our novel prototype system employs a frequency in the transition region where both conduction and displacement current effects are important. The frequency is low enough to provide adequate penetration depth, and it is high enough to allow for the mapping of dielectric features, such as those that characterize organic contaminant plumes. In addition, our prototype system employs new null-field techniques that cancel the strong primary field (which includes the homogeneous earth background response), thereby providing the opportunity to accurately measure the weak secondary field associated with the subsurface target. Our prototype system will be referred to as an Electromagnetic Sensitive Null-Array Probe (EM-SNAP).

1.1 Summary of Significant Results

1. Although our previous work has shown the power of ellipticity measurements for mapping extended targets, such as layered-earth structures, we found in this project that ellipticity measurements have severe limitations for imaging relatively small, three-dimensional targets.

2. In order to obtain greater resolution for three-dimensional targets, during this research project, we investigated various methods for canceling the direct plus background magnetic fields, i.e., null-field measurements.

(a) We investigated canceling the primary field using three different geometric-null configurations. Although in theory this should be an effective method, we found that at high frequencies there are some severe limitations.

(b) We also investigated electrical nulling. We found this method to be reliable, well-suited for computer automation, and relatively efficient.

(c) We took advantage of the strengths associated with both electrical and geometrical nulling by combining the two techniques in a hybrid electrical/geometrical nulling system.

3. We validated EM modeling codes for calculating fields over buried three-dimensional targets.

4. We analyzed the null-coupling problem via the EM modeling codes.

5. We set up a series of controlled experiments at our test facility using fixed transmitter-receiver arrays and moving targets in buried PVC pipes. We found this to be a very effective technique for the initial proof-of-principle tests on the nulled-field system.

6. We have made a number of advances on the coil antennas used for these measurements, including:

   (a) Much research was devoted to finding effective techniques for electric field shielding of the loop antennas. After achieving limited success with standard Faraday shielding techniques, further research led us to the highly-effective doubly-loaded loop antenna.

   (b) We designed coil antennas which have greatly improved sensitivity over a wide band using coil-connecting turns in the coil, instead of our previous coil-cutting turns.

   (c) We designed low-VSWR matching circuits for these antennas.

   (d) We developed a tuned receiver loop which provides substantial noise rejection outside of the desired passband.

7. In order to accomplish the electrical nulling, we developed a computer-controlled amplitude and phase-adjustment circuit for injecting the cancelation signal.

8. The first electrical nulling system exhibited substantial cross-talk between the in-phase and quadrature components in the electrical cancelation circuitry. We significantly reduced the cross-talk problem by going to a modular design which employs voltage-controlled amplifiers.

9. In order to verify safety and FCC compliance for these measurements, we conducted EM radiation tests.

10. We performed a thorough literature search in this field, including patent literature.

11. We developed the capabilities to make electrical property measurements in order to provide the background information needed to interpret data using the system.

12. We found that it is necessary to electrically isolate the operator from the system. We achieved this goal by developing a Pentium-notebook-computer-based automatic-control interface (programmed in LabVIEW) that is optically isolated from the electrical nulling circuitry.

13. We successfully used this prototype system to measure the EM responses due to subsurface metal targets. These initial tests over metallic targets were of great help in debugging the prototype system.

14. In order to provide an analog for an organic contaminant plume, we used the prototype system to measure the EM response over a dielectric target. We again found a diagnostic response which demonstrated the potential of this approach for mapping the flow of organic fluids in the subsurface.

1.2 Personnel Supported

- Faculty:
  - Dr. Ben K. Sternberg
  - Dr. Steven L. Dvorak
- Graduate Students:
  - Pat Debroux, Ph.D. Student
  - Pixuan "Joe" Zhou , Ph.D. Student
  - Gary Lewis, M.S. Student
  - Charles Abernethy, Ph.D. Student
  - John Glaser, Ph.D. Student
  - Catherine York, M.S. Student
  - Emily Sullivan, M.S. Student
  - Jeffrey Seligman, Ph.D. Student
  - Charles James, M.S. Student
  - Charles Thompson, M.S. Student
  - Pat Cicero, M.S. Student
- Undergraduate Students:
  - Jeff McDermott
  - Timothy Randolph
  - Sanjev Pandey
  - Greg Gartland
  - Ray Grymko
  - Brandon Sexton

- Senior Electronics Technicians:
  - Robert Esterline
  - Richard Zito
  - James Laird
- Consultant:
  - Terry Leach

2 OBJECTIVES AND RELEVANCE TO THE AIR FORCE MISSION

The overall objective of this research is a reduction in the amount of time and money required for subsurface remediation projects. Currently there are no systems available which allow for economical and accurate subsurface imaging of remediation sites. Therefore, any new methods which help solve this problem will greatly benefit the Air Force as well as other Defense Agencies and Contractors. Since the application of EM methods to subsurface imaging is still in its infancy, there are a number of issues that still need to be addressed. The basic research which was conducted in this grant addresses a number of these issues. The knowledge that was gained from this research brings us closer to a solution for this very difficult problem.

Much theoretical research must still be conducted in order to fully understand this complex problem. The transition region between the conduction current regime and the displacement current regime is a difficult area for EM theory. At present, reliable EM results for resistive targets are not available for this frequency region. This frequency range, however, is potentially the most useful for organic contaminant plume mapping. Basic research in these areas has tremendous potential for adding knowledge and capability to the environmental restoration and cleanup program.

3 MOTIVATION

3.1 Conventional Ground Penetrating Radar Approaches

Where high-frequency ground penetrating radar (GPR) has sufficient depth of penetration, it can be an extremely powerful tool for subsurface mapping of metallic and dielectric targets. Figure 1 shows GPR records over a hydrocarbon contaminant plume [1] at the site of a former gasoline station. Anomalous GPR responses were detected well above the expected depth of mobile LNAPL. The shallow depths of the anomalous responses indicate that a residual or possibly vapor contaminant found in the non-saturated zone produces the characteristic GPR anomaly detected during this survey. Additional examples of GPR records taken over controlled LNAPL spills are given in [2, 3]. These tests showed conclusively that GPR could accurately detect subsurface contaminant plumes. Additional processing can also be applied to the GPR data in order to recover dielectric constant information which may distinguish contaminants from other background variations [4].

Unfortunately, in more lossy soils (which occur frequently at environmental restoration sites), the use of GPR is not nearly so straightforward. Field experience has shown that GPR is often

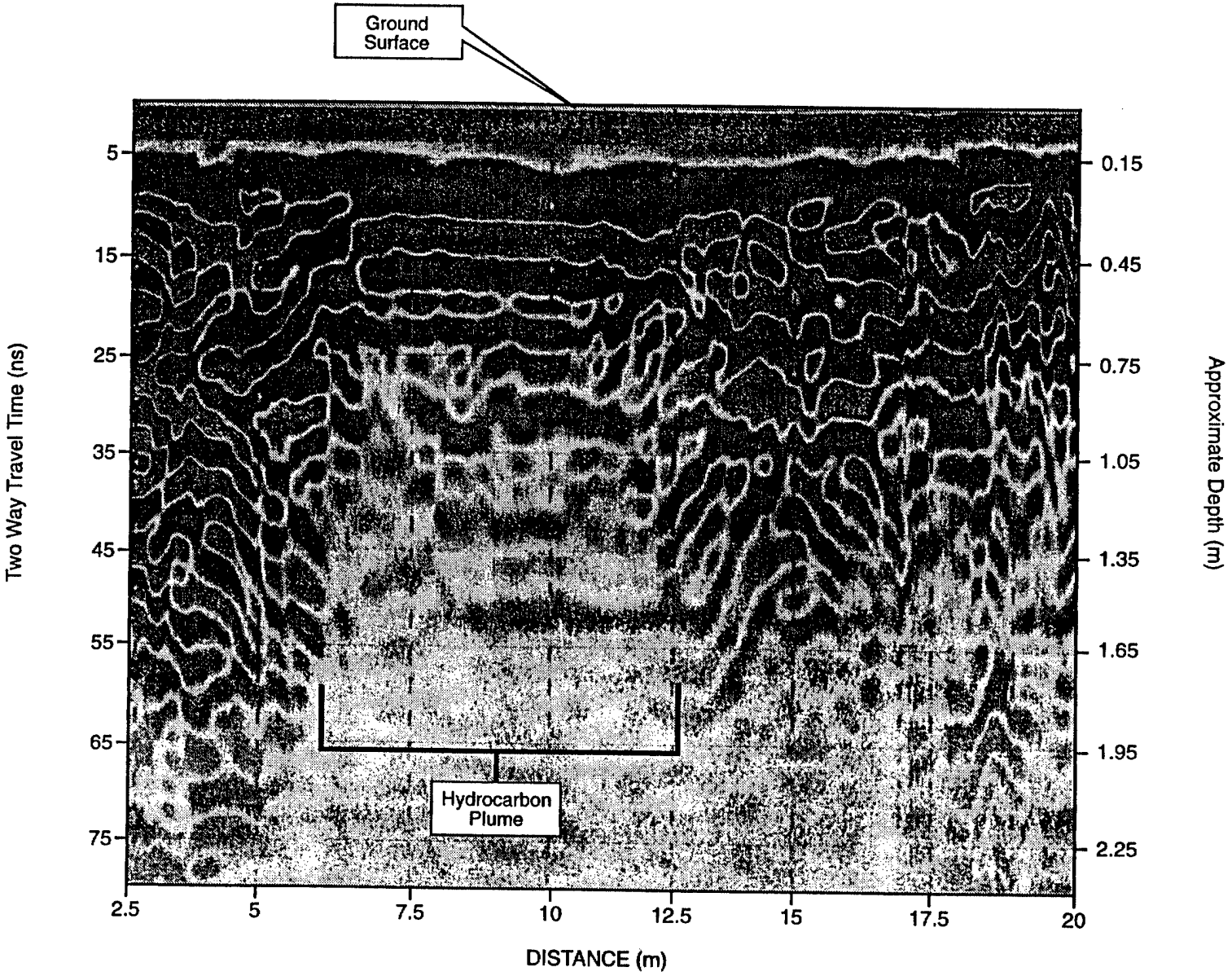


Figure 1. Typical GPR response over LNAPL plume (from Maxwell and Schmok, 1995).

severely limited. In some areas we find depths of penetration for GPR of many meters; in a few cases, as many as tens of meters. Short distances away, however, penetration may decrease to less than 1 m. In many areas of the United States, the depth of penetration is consistently less than 1 m.

3.1.1 Soil Electrical-Property Measurements

In order to better understand this phenomenon, the Laboratory for Advanced Subsurface Imaging (LASI) has developed the capability to measure the electrical properties of soils. The measured data in Figure 2 shows the reason for the variability in the GPR penetration depths. The curves labeled "Brookhaven" in Figure 2 show the attenuation for a low-loss soil. This soil is representative of the attenuation in a clean sand or other high-resistivity, clay-free soil. The curves labeled "Avra" are for a relatively high-loss soil. This example is representative of soil containing a moderate amount of clay, such as montmorillonite clay, or other relatively low-resistivity soils. In low-loss soil, attenuation at the usual radar frequencies of 100s of MHz is only a few dB/m. This allows penetration into the earth of many meters. In higher-loss situations, the attenuation in the frequency range of 100s of MHz is approximately 50 dB/m. Such high-loss situations are a common occurrence based on our own experience as well as the experience of others. This has severely limited the application of GPR.

With attenuations of 50 dB/m and larger, the potential gains from high-power transmitters or signal processing are extremely limited. In order to have sufficient depth of penetration in lossy soils, frequencies of the order of MHz to tens of MHz must be used rather than hundreds of MHz which is used in most conventional GPR systems. Lower-frequency (10 - 20 MHz) GPR antennas have been used with conventional GPR systems, but the data obtained with these antennas have much too low a resolution and accuracy to be effective for mapping subsurface organic contaminant plumes. At these lower frequencies, the EM energy does not propagate as a simple wave, but instead is a highly dispersed wave involving both conduction-current and displacement-current effects. Standard wave-propagation interpretation techniques (e.g., SAR or migration, wave-propagation based modeling, etc.) do not apply in this situation. Since little attention has been focused on electromagnetic imaging systems which employ frequencies in the MHz to tens of MHz range, much basic research is needed to better understand the phenomenology associated with this problem.

3.2 Electromagnetic Induction

A number of commercial electromagnetic instruments are currently used for shallow subsurface electromagnetic induction surveys. These systems are effective, for example, for determining depth to water table or depth to bedrock. Commercially-available EM-induction systems, however, do not operate at high enough frequencies to detect subsurface organic contaminant plumes. This has led us in the LASI Laboratory to pursue the development of higher-frequency electromagnetic induction systems as described in the next section.

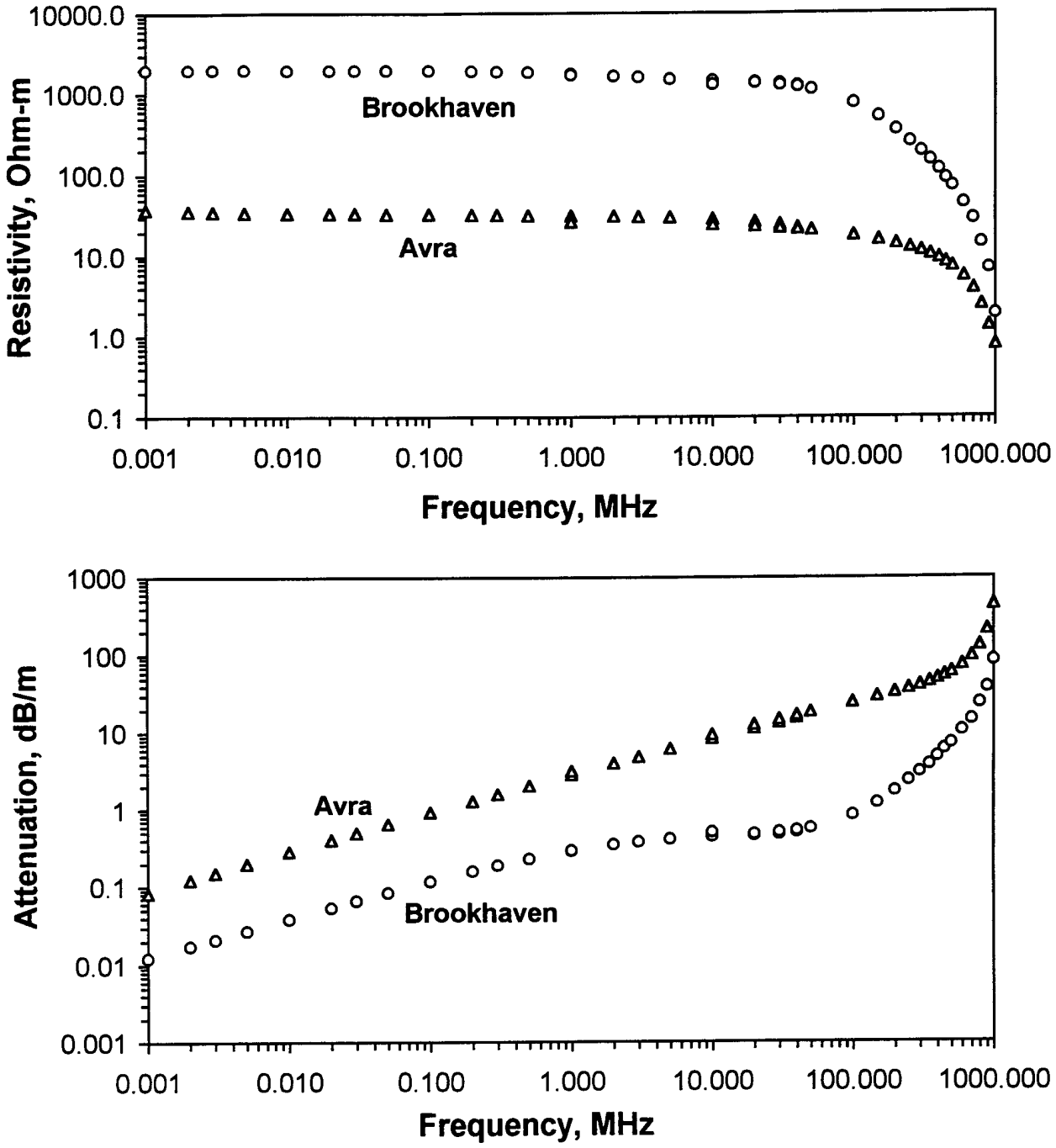


Figure 2. Comparison of high-attenuation and low-attenuation soils. Soil sample 5-2.0 from the University of Arizona Avra Valley Geophysical Test Site is representative of moderately high-attenuation soils. Soil sample I,2-4 from Brookhaven National Lab, New York, is representative of low-attenuation soils. The soil-moisture wetness for both samples was about 10%. Conventional ground-penetrating radar (GPR) at frequencies of several hundred MHz may have a depth of penetration of ten meters or more in low-attenuation soils, but only one meter in high-attenuation soils. Lower frequencies and alternate measuring schemes are needed in high-attenuation soils.

4 REVIEW OF PROJECT DIRECTIONS AND ACCOMPLISHMENTS

4.1 EM Research at the Laboratory for Advanced Subsurface Imaging (LASI)

We have developed a series of high-resolution, high-frequency EM ellipticity systems at LASI. Support for this development has included: Electric Power Research Institute [5], The Copper Research Center [6, 7], U.S. Geological Survey [8], U.S. Bureau of Mines [9], Dept. of Energy [10], U.S. Army [11], and University of Arizona. One of the recent developments in this series is a 30 kHz to 30 MHz EM ellipticity system, which was developed primarily with Department of Energy (DOE) funding [10]. Figure 3 shows a photo of the high-frequency ellipticity system. The Principle Investigator for the data-acquisition system development was Dr. Ben Sternberg.

The LASI high-frequency ellipticity EM system is the only operational geophysical system which has successfully recorded high-accuracy EM data in the transition from the conduction-current region to the displacement-current region. One of the principle breakthroughs in this system involves the elimination of electric-field interference on the magnetic-field measurements. This capability to accurately record EM fields in the 1 to 30 MHz range has allowed us to interpret for the first time the dielectric constant at depths that are of interest in environmental studies in lossy soils. Mapping the dielectric constant is crucial since the presence of many environmental targets (like organic contaminant plumes) in the soil is most likely to alter the dielectric constant rather than the conductivity (or resistivity).

4.2 Ellipticity Measurements

All of the previous LASI systems were based on ellipticity measurements. The ellipticity measurement is discussed by Hoversten [12] and Ward et al. [13]. A plot of the 3D polarization ellipse is shown in Figure 4. In a 2D problem that is uniform in the y-direction, a sinusoidal source produces a total magnetic field at the receiver that traces out an ellipse in the $x - z$ plane as a function of time. The ellipticity is defined as the ratio of the major to minor axes of this ellipse, and can be written in terms of the relative magnitude and phase of the H_x and H_z fields,

$$e = \left| \frac{H_z \cos\alpha - H_x \sin\alpha}{H_z \sin\alpha + H_x \cos\alpha} \right|, \tag{1}$$

where

$$\tan(2\alpha) = \frac{2\left|\frac{H_z}{H_x}\right| \cos(\phi_z - \phi_x)}{1 - \left|\frac{H_z}{H_x}\right|^2} \tag{2}$$

and ϕ_x and ϕ_z are the phase angles for the components of the total field.

The standard procedure for calculation of ellipticity uses just H_x and H_z. However, the LASI ellipticity systems measure all three components and then a mathematical rotation algorithm is used to obtain the ellipticity. This assures quick and accurate measurements for the ellipticity since the 3D coil can be placed in any orientation, thus avoiding the manual nulling procedures that must be applied when 1D or 2D coils are used.

There are a number of characteristics that make the LASI frequency-domain, ellipticity systems extremely useful:

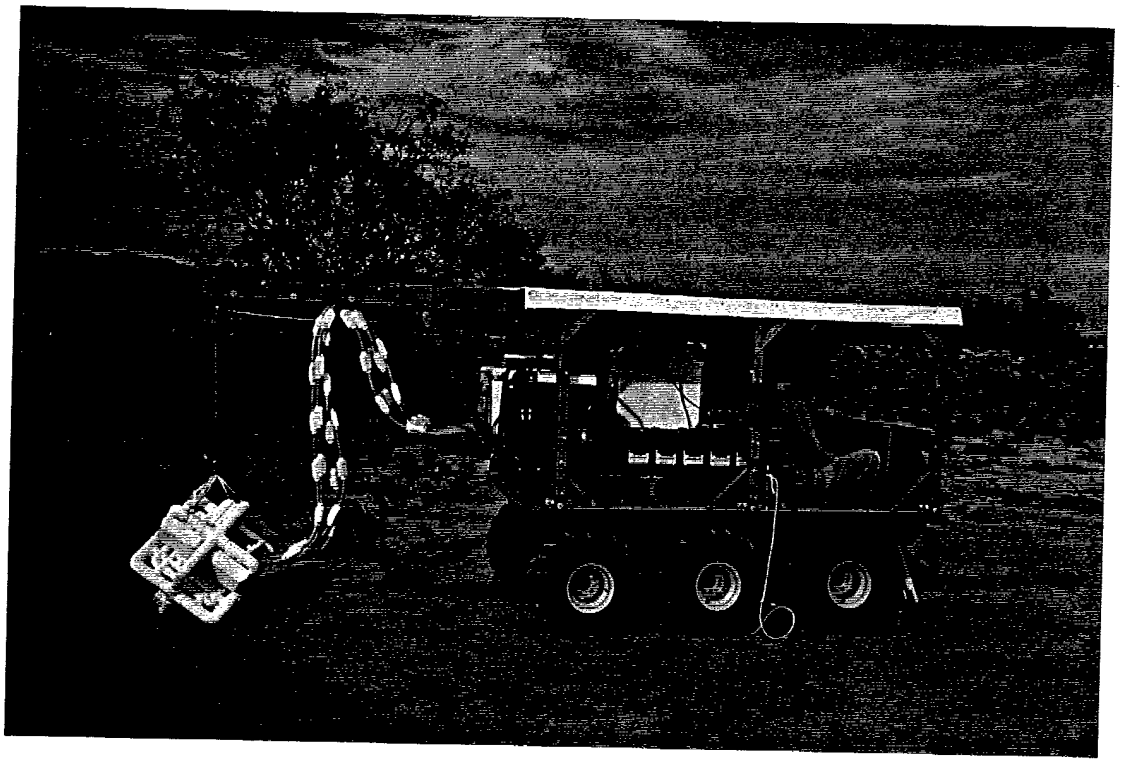


Figure 3: This picture shows the LASI high-frequency (30 kHz to 30 MHz) ellipticity system.

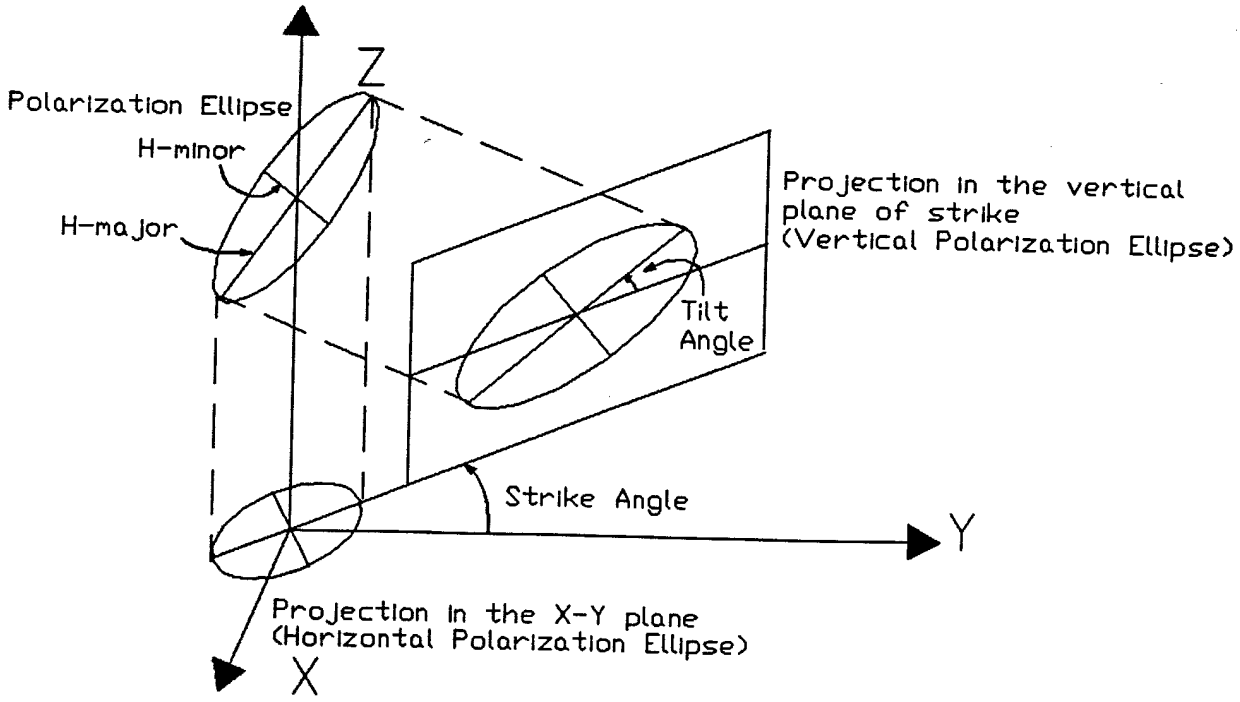


Figure 4: Illustration of the polarization ellipse, the tilt angle, the strike angle, the horizontal polarization ellipse, and the vertical polarization ellipse.

- High tolerance to industrial noise
- High sensitivity to layered-earth structure
- Low sensitivity to geometric errors
- Convenient measurement
- High accuracy

Hoversten, in a comparison of time- and frequency-domain EM sounding techniques, showed that for a layered earth, the frequency-domain ellipticity measurement is superior to any other frequency-domain or time-domain measurement [12]. He showed that "the ellipticity measurement provides smaller parameter standard errors than the time-domain data". "In addition, the model parameters arrived at through the least-squares inverse are much less correlated with each other when ellipticity is used."

Because of the previous successes associated with the application of ellipticity to subsurface characterization, we had proposed to extend the capabilities of the ellipticity systems by measuring additional parameters, e.g., strike angle, tilt angle, and ellipticities for different transmitter polarizations. Initial tests, however, demonstrated that ellipticity measurements are ideally suited for measuring broad subsurface structures, such as layering. Ellipticity systems, however, have limited capabilities for imaging relatively small 3D subsurface targets that are weak scatterers. Since the ellipticity depends on both the primary and secondary fields, ellipticity measurements are most useful for mapping anomalies which produce relatively large secondary fields. Basically, the ellipticity provides an average measurement for the electrical properties associated with the region between the transmitter and receiver. For example, ellipticity has been shown to be a particularly sensitive parameter for mapping depths to buried layers in the subsurface [12]. It has also been shown to be useful in the detection of long, buried, conductive pipes [14]. Note that the large, induced conduction currents within the pipe result in a relatively strong secondary field for this case.

Unfortunately, smaller three-dimensional targets, buried at any appreciable depth, produce very weak secondary fields at the earth's surface. In order to detect the weak secondary signal, one must separate it out from the overwhelming primary field. Null-field receivers are best suited for this purpose.

4.3 Previous Efforts at High-Resolution EM Imaging

As discussed earlier, high-frequency GPR has been shown to be capable of accurately mapping subsurface contaminant plumes. Unfortunately, high-frequency GPR has much too limited a depth of penetration in many soils to be useful for a large number of environmental problems. Lower frequencies are needed to obtain a reasonable depth of penetration. The powerful processing techniques which can be applied to GPR (e.g., SAR or migration, wave-propagation based modeling, etc.) do not apply at these lower frequencies. The goal is therefore to provide the imaging capabilities of GPR in all soil types.

Electromagnetic-induction instruments do have sufficient depth of penetration for environmental problems. However, commercially-available EM-induction instruments do not operate at high enough frequencies to detect organic contaminants.

Much of the recent work in GPR has been concentrated on sophisticated methods to recognize targets (like subsurface contaminant plumes) in the data, including electromagnetic modeling (both numerical and analytical), pattern-recognition techniques (including neural networks), and identification of resonances. Unfortunately, these methods have met with limited success. This is not so much due to the failure of these sophisticated algorithms, but more to the lack of truly-diagnostic information in the original data.

An example may help to illustrate these points. Idaho National Engineering Laboratory (INEL) is a location where high-attenuation soils occur. Several investigators have argued that with the right data processing, usable GPR data could be obtained at the INEL site. Investigators tried sophisticated signal-processing techniques, target-identification algorithms, and various instruments. None of the approaches provided usable data beyond the order of tens of cm depth. Nevertheless, the view persisted that there must be a "silver-bullet" processing algorithm that would extract the needed data.

In September of 1994, Dr. Sternberg was involved with some of these tests at the INEL Cold Test Pit (CTP) using a state-of-the-art GSSI System-10 GPR and a 500 MHz center-frequency antenna. A large steep-sided ditch had been excavated at the site and we drove a re-bar rod into the side of the ditch at a depth of 20 cm. We then ran the GPR along the surface near the edge of the trench and recorded an excellent response from the re-bar rod. The response disappeared when the rod was pulled out confirming what was target response versus background response. We then drove the same re-bar into the side of the ditch at a depth of 40 cm. No response was observed on the plotted records. We then examined the raw incoming data when the antenna was directly over the re-bar and we pulled the re-bar out of the soil. Not one digitized bit toggled when the rod was pulled out! We repeated this test in several locations with identical results. This simple, but diagnostic test, dramatically brought home to those in attendance that all the signal processing that could ever be applied will not bring images out of data in these circumstances.

4.4 Null-Field Receivers

Null-field receivers are designed to cancel out the primary field, thereby enabling the accurate measurement of the much smaller secondary field. Here we define the primary field as being the response in the absence of the target, i.e., it includes the direct field plus signals resulting from the currents induced in the background. There are basically two different techniques used for nulling — geometrical and electrical nulling. These two methods were investigated in great detail during the first year of the grant. After discussing how nulling techniques can be used to detect subsurface targets, we outline our findings on both the geometrical and electrical-nulling techniques. We then discuss how electrical- and geometrical-nulling techniques were combined in the new hybrid electrical/geometrical-nulling system.

A randomly oriented receiver coil will sense both the primary and secondary magnetic fields. As previously discussed, the large primary field must be cancelled before the secondary field, which contains the information about the target, can be accurately measured. One possible method for detecting the subsurface target involves first nulling out the entire received field at a fixed location. Then the transmitter and receiver are moved in unison and a new field reading is recorded. The received signal will provide a clear indication of any changes in the background. If the electrical properties change relatively slowly over the distance of the motion, then there will only be a small signal received. However, when the transmitter-receiver array moves over a target, a larger signal

may be registered. In addition, the target of interest may possess distinctive spatial characteristics in comparison with the response due to background variations.

During the later tests of the prototype nulling system, the transmitter and receiver coils were mounted on a platform, and the platform was rolled smoothly on fiberglass tracks. However, for our initial tests, we held the transmitter and receiver coils fixed and moved the target. This was accomplished by pulling a conductive target through a buried PVC pipe (see subsection 7 for more details).

It should be noted that time-domain sounding only cancels the direct signal, i.e., the received signal contains information about both the target and the background medium. We have found that there are also other advantages associated with frequency-domain null measurements, e.g., narrow band filtering techniques can be employed and one does not have to worry about the dispersion which is inherent to time-domain measurements.

4.4.1 Geometrical Nulling

The concept of geometrical nulling is very simple. At a point in space, the magnetic field vector, associated with a monochromatic source, can be thought of as tracing out an elliptical pattern as time progresses. Since the magnetic field vector always lies within a plane, an ideal coil antenna, which is placed in the plane of the ellipse (i.e., coil moment perpendicular to the plane of the ellipse), will pick up no magnetic field. A coil antenna, oriented as such, is null-coupled to the magnetic field. Note that the receiver must be shielded against unwanted electric field effects.

In order to test the geometrical-nulling technique, during the first year of the project we constructed a coil which we attached to a gimbal mount and controlled using micrometers. The transmitter/receiver assembly was then placed over the buried polyvinyl chloride (PVC) pipe. After geometrically nulling the field, a small conductive target was pulled through the buried PVC pipe, and the received field was monitored. We found that we were able to detect targets which produced secondary fields on the order of one one-millionth the magnitude of the primary field. It would be very difficult to detect this small of a field without first removing the primary field.

Geometrical nulling achieved the desired goal of removing the primary field, but there are a number of disadvantages associated with geometrical nulling:

- Each receiver requires a gimbal mount. This makes it difficult to utilize an array of receivers.
- After the receiver is nulled, the transmitter/receiver orientations must be held fixed while the assembly is moved. This is difficult with the geometrical-nulling apparatus since the receiver must be free to move on the gimbal mount.
- The mechanical positioning would make the optimization time consuming.
- The nulls are so sharp that it is easy to miss the null. We manually adjusted for the nulls, but a very sophisticated computer algorithm would be required to automatically find the optimal null.

We found that these limitations were especially severe at high frequencies.

4.4.2 Electrical Nulling

Because of the problems associated with the geometrical-nulling system, we also developed and tested an electrical-nulling system. The basic idea behind the electrical-nulling system is to inject an electrical signal into the receiver coil which cancels the primary field. In order to partially reduce the primary field, we mounted the receiver coil at an angle of 54.7 degrees relative to the transmitter coil. This position would correspond to a geometrical null if the transmitter/receiver assembly was located in free space. The remaining primary field is cancelled by the electrical-nulling system. This system consists of a sensing coil, a 90 degree phase shift network, and a control algorithm which automatically adjusts the magnitudes of the in-phase and quadrature components of the cancelation signal. By adjusting the magnitudes of the in-phase and quadrature components, it is possible to adjust both the magnitude and phase of the cancelation signal. This signal is then added to the received data signal, thereby allowing for the cancelation of the received signal, i.e., electrical nulling.

The electrical-nulling system was found to have a number of advantages over the previously discussed geometrical-nulling system. Some of the most important advantages are listed below:

- Since we directly monitor the signal, electrical nulling also allows for cancelation of some of the noise which is inherent to the signal.
- The transmitter and receiver can be rigidly mounted so that their relative orientations do not change during the survey.
- The hardware associated with the electrical-nulling system (i.e., voltage-controlled amplifiers, power splitters, and combiners) are much more compact than the equipment required for the geometrical-nulling system, i.e., gimbal mounts and micrometers. Thus, the electrical-nulling system is better suited for the array applications which we believe are necessary for subsurface imaging of contaminant plumes.
- The nulls associated with the electrical-nulling system are much easier to locate than those associated with the geometrical-nulling system.
- The electrical-nulling system is more robust and we have experienced fewer difficulties with this system than with the geometrical-nulling system.
- Electrical-nulling can be easily carried out automatically via a computer.

4.4.3 Hybrid Electrical/Geometrical Nulling

In the prototype EM-SNAP system we developed, we used a hybrid electrical/geometrical nulling technique. In the previously discussed electrical-nulling system, the receiver coil was mounted at an angle of 54.7 degrees relative to the transmitter coil. While this position corresponds to a geometrical null in free space, substantial signal levels are still obtained when the transmitter/receiver assembly is placed on the surface of the earth. In the EM-SNAP prototype, we placed the receiver coil orthogonal to the transmitter coil. This orthogonal "true-null" arrangement provided significantly lower signal levels than the 54.7 degree arrangement. Field tests indicated that the orthogonal "true-null" arrangement provided typically 40 dB of signal nulling

without highly-accurate mechanical adjustment. The improved electrical-nulling circuitry, which is discussed later, provided an additional 60 to 80 dB of signal nulling capability.

Based upon our preliminary studies, the hybrid electrical/geometrical-nulling system is far superior to either of the previous systems. The hybrid system has proven to be effective for removing the large, unwanted primary signal. Removal of the primary signal is mandatory if one hopes to detect the small secondary fields produced by 3D subsurface targets.

4.5 Comparison of Nulling with Other Methods

A time-domain GPR system records in the absence of the direct signal (except for the ringing of the transmitted waveform). Time-domain instruments, however, are still dominated by response from the background. A null system is needed to effectively isolate a target's response.

Although electrical nulling is frequently used in low-frequency metal detectors, for example, we had to develop new electrical-nulling techniques for high frequencies. There are a number of hurdles that must still be addressed in order to develop highly-sensitive, null-field arrays at high-frequencies, e.g., electromagnetic interference, shielding, temperature instabilities, mechanical instabilities, and noise.

Nulling has also been used in dielectric mine detectors. The high-frequency, prototype EM-SNAP system we developed is very different from anything in the mine detection literature.

4.6 Why We Chose to Make Measurements in the Frequency Domain

For many years geophysicists have debated the relative merits of time domain versus frequency domain. In the low-frequency range (0.1 Hz - 10 Hz) that is typically used in mineral exploration and similar applications, there are potential advantages to both time-domain and frequency-domain measurements. However, in the high-frequency range, we feel that measurements must be made in the frequency domain. In the range from Megahertz to tens of Megahertz, the dominant noise is spectral spikes due to radio stations. In a time-domain system, one must either use a high-power transmitter or integrate for very long times. In a frequency-domain system, we can narrow-band filter and eliminate this noise without large transmitter power or unreasonably long integration times. The FCC imposes severe constraints on the amount of power and the signal received from a transmitting source. Specifically, one limit that is often used is 15 μV/m signal strength at a distance of 300 meters. By using a frequency-domain approach, with narrow-band filtering, we were able to obtain highly-accurate data with reasonable power levels. For some applications, where larger depth of investigation is needed, it will be necessary to use relatively high power, and this would be required regardless whether one were using a time-domain or frequency-domain approach. In the frequency domain, it is possible to use what are called ISM frequencies, which refers to industrial, scientific and medical equipment. One can use very large power levels at these specified frequencies. The ISM frequencies in the range that would be of interest in mapping the extent of subsurface contaminant plumes include 6.78 MHz, 13.56 MHz, 27.12 MHz, and 40.68 MHz.

In summary, frequency-domain methods are required, we feel, in the high-frequency range. Furthermore, the high-frequency range is necessary in order to map the changes in dielectric constant associated with subsurface organic contaminant plumes.

4.7 Electric Field Interference

A number of our tests have clearly shown the importance of interference from electric fields. Even when the coil is in a perfect geometrical null it will pick up substantial signal. This is because the coil is an appreciable fraction of a wavelength (greater than 0.01 λ) at 8 MHz. When the coil can no longer be assumed to be electrically small, the coil will act like an electric field dipole. Standard Faraday shielding techniques do not eliminate this electric dipole response. However, we have found that a doubly-loaded loop antenna can significantly reduce the electric field pickup [16, 17]. Electrical nulling can cancel the remaining electric field pickup while geometric nulling can not.

4.8 Cross Talk

It is possible to adjust both the magnitude and phase of the cancelation signal by adjusting the magnitudes of the in-phase and quadrature components. Our original electrical-nulling circuitry included mixers mounted on a printed circuit board. Our tests indicated that there was significant levels of cross-talk between the in-phase and quadrature channels. This cross-talk made it difficult to consistently obtain the null. In order to circumvent this problem, we replaced the printed circuit board mounted mixers by voltage controlled amplifiers (VCAs) which are individually mounted in shielded boxes. In fact, we went to a complete modular system wherein all of the subsystems were mounted in shielded modules. The modularization dramatically reduced the amount of cross-talk between the in-phase and quadrature components, thereby resulting in an improved system that is capable of achieving deep, repeatable nulls. The modules also provided a much more flexible system for experimentation.

4.9 Optical Isolation

Our field tests also demonstrated that an electrical connection between the operator and the electrical-nulling circuitry can lead to instabilities when a deep null has been achieved. We found that the user, who acts as an antenna, could influence the level of the null by touching various pieces of equipment.

In the original nulling system, we used variable resistance pots to control the amplitude of the DC signals that were fed into one port of the mixers. Mixing the adjustable DC signals with the in-phase and quadrature components of the sampled signal allowed for the adjustment of the magnitude and phase of the cancelation signal. Unfortunately, tests demonstrated that touching the pots led to instabilities in the null.

In order to address this problem, we set up a remote Pentium computer (connected via a fiber optic link) to control the whole system. The Pentium computer, which is dedicated to this project, was purchased on University of Arizona funds. In order to allow for variable DC signals, we replaced the variable resistance pots by an HP programmable DC power supply. The HP programmable DC power supply provided the stability needed to obtain deep stable nulls. The user interface was built around LabVIEW. Through LabVIEW, the user can control the variable DC power supply and the HP network analyzer. This allowed the user to vary the magnitude and phase of the cancelation signal in order to manually search for the null. We found that the optical isolation resulted in a much stabler system. The added stability made it much easier to search and find the nulls.

5 OVERVIEW OF THE DEVELOPED INSTRUMENTATION

As previously discussed, small three-dimensional targets, buried at any appreciable depth, produce very weak secondary fields at the earth's surface. In order to detect the weak secondary signal, one must separate it from the overwhelming primary field. Null-field receivers are best suited for this purpose. Null-field receivers are designed to cancel out the primary field, thereby enabling the accurate measurement of the much smaller secondary field. One possible method for detecting the subsurface target involves first nulling out the entire received field at a fixed location. Then the transmitter and receiver are moved in unison and a new field measurement is recorded, i.e., a differential field measurement. The received signal will provide a clear indication of any changes in the subsurface. If the electrical properties change relatively slowly over the distance of the motion, then there will only be a small signal received. However, when the transmitter-receiver array moves over a target, a larger signal may be registered. Furthermore, the target of interest may possess distinctive spatial characteristics in comparison with the response due to background variations. As shown in Figure 5, the transmitter and receiver coils are mounted on a platform, and the platform rolls smoothly on fiberglass tracks. However, for our initial tests, we held the transmitter and receiver coils fixed and moved the target. This was accomplished by pulling dielectric targets through a buried PVC pipe. The buried-pipe test facility is discussed in more detail in subsection 7.2.

The prototype EM-SNAP system employs a network analyzer to collect the data. A picture of the equipment rack, which holds the network analyzer, a DC power supply, and an oscilloscope, is shown in Figure 6. A block diagram for the prototype EM-SNAP system is shown in Figure 7. The basic idea behind the electrical-nulling system is to add an adjustable electric signal with the measured data signal in order to cancel the primary field. The electrical-nulling system consists of a 90 degree phase shift network and a pair of voltage controlled amplifiers which are connected to a programmable DC voltage supply and is remotely controlled via National Instrument's LabVIEW software (Figure 7). By adjusting the magnitudes of the in-phase and quadrature components, it is possible to adjust both the magnitude and phase of a reference signal. This signal is added to the received signal, thereby allowing for cancelation of the received signal. We designed, constructed, and tested both geometrical and electrical nulling systems and found that each method had inherent strengths and weaknesses. We ultimately took advantage of the strengths associated with both electrical and geometrical nulling by combining the two techniques in the hybrid electrical/geometrical nulling system called EM-SNAP (Figures 5 and 7).

Some of the key elements in the EM-SNAP block diagram (Figure 7) are discussed below. A network analyzer is used as both the source and receiver in the system. The output of the network analyzer is first amplified, and it is then used to drive a tuned transmitting coil. Part of the amplified signal is also split off to be used as the reference signal in the electrical nulling circuitry. The transmitter and receiver coils are rigidly mounted on a movable platform (Figure 5). The signal picked up by the geometrically-nulled receiver coil is amplified, filtered, and then combined with the adjustable electrical-nulling signal. The combined signal, i.e., the combination of the received and cancelation signals, is then measured by the network analyzer. The network analyzer is ideal for measuring the combined signal since it provides a large dynamic range together with narrow band filtering (10 Hz) and signal averaging. Our field tests demonstrated that an electrical connection between the operator and the electrical nulling circuitry led to instabilities when a deep null was achieved. In order to address this problem, we set up a remote Pentium notebook

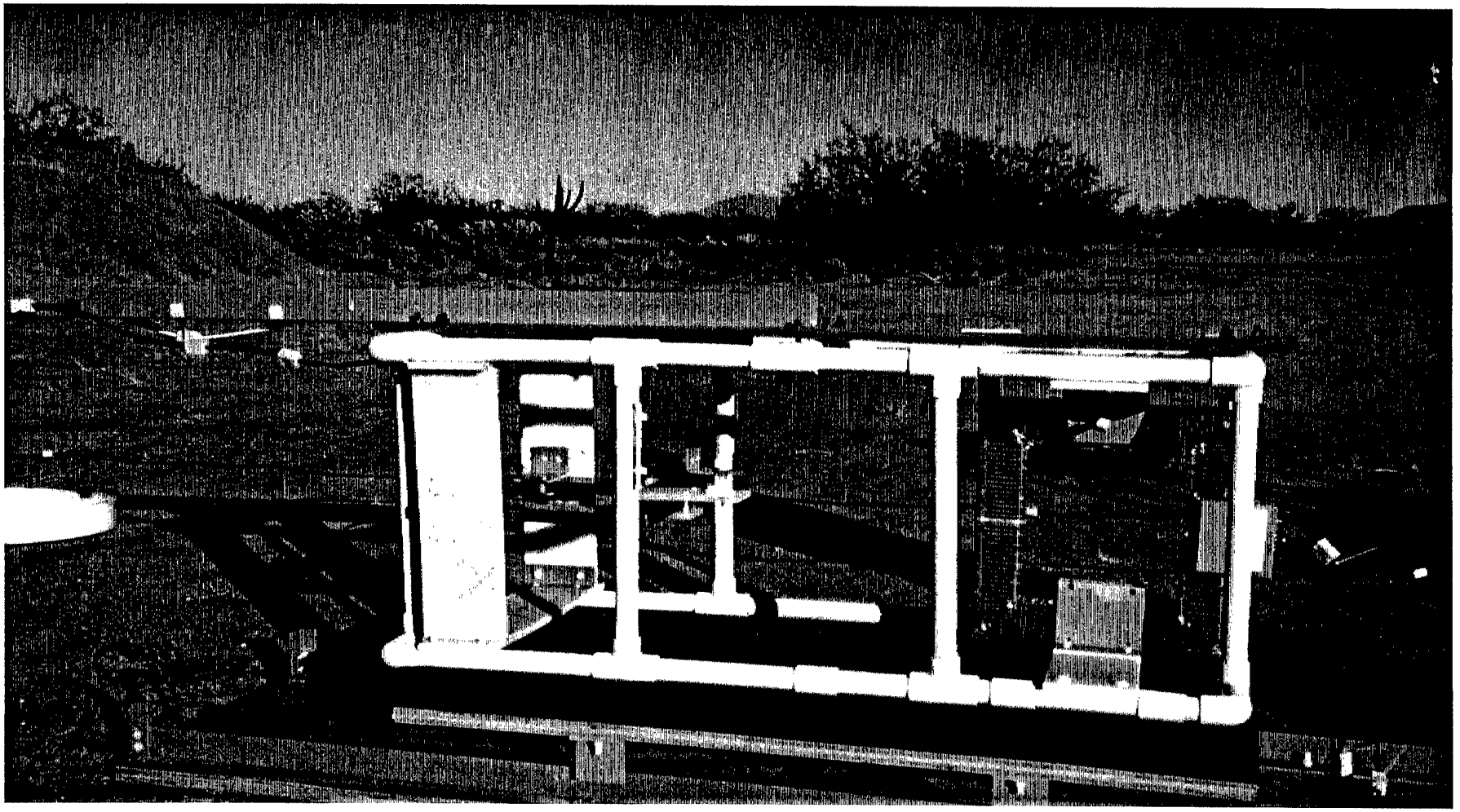


Figure 5. This picture shows the EM-SNAP platform that moves smoothly on a track. The moveable platform supports the rigidly mounted transmitter (right) and receiver (left) coil antennas.

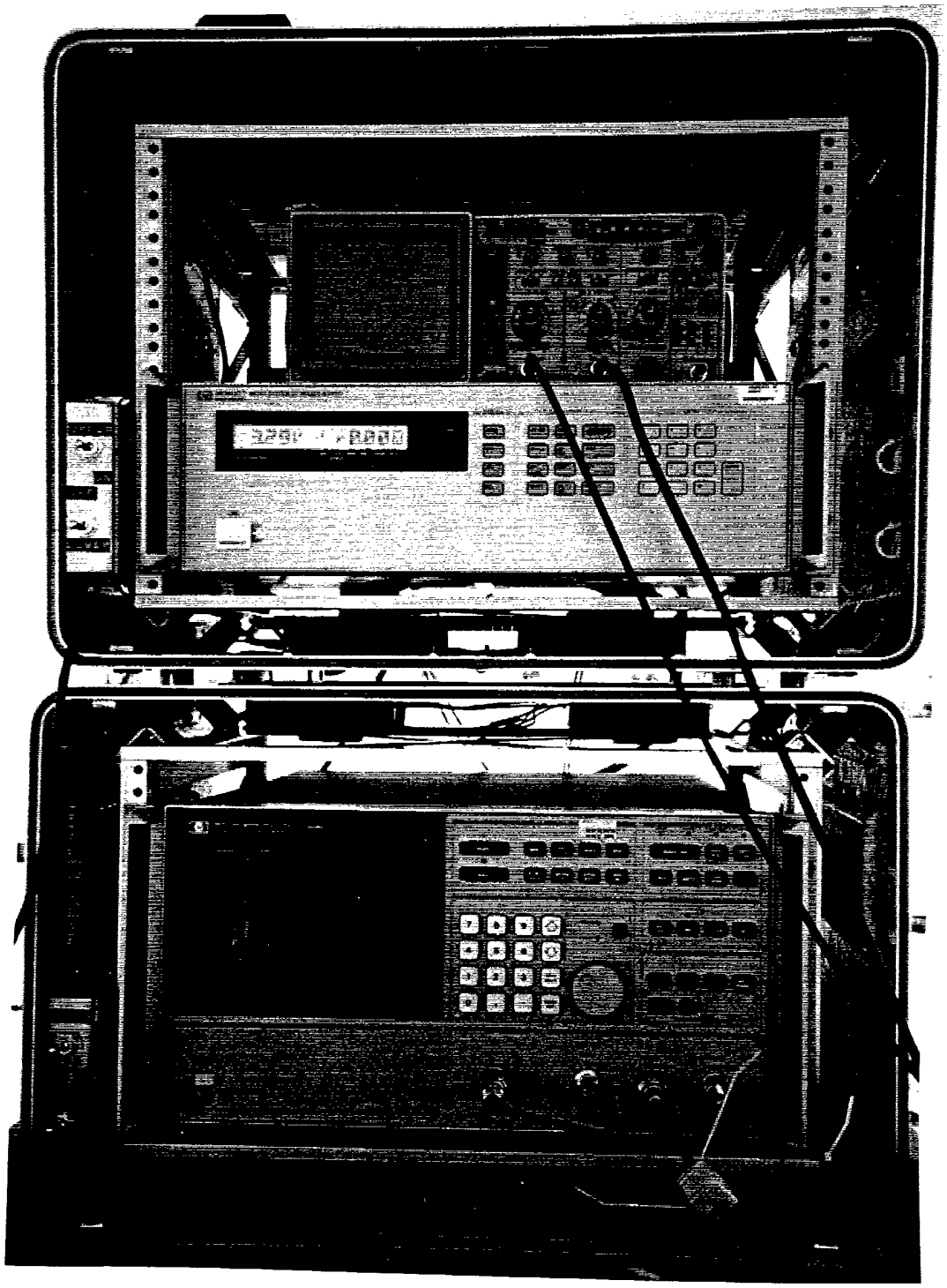


Figure 6: This picture shows the EM-SNAP equipment rack.

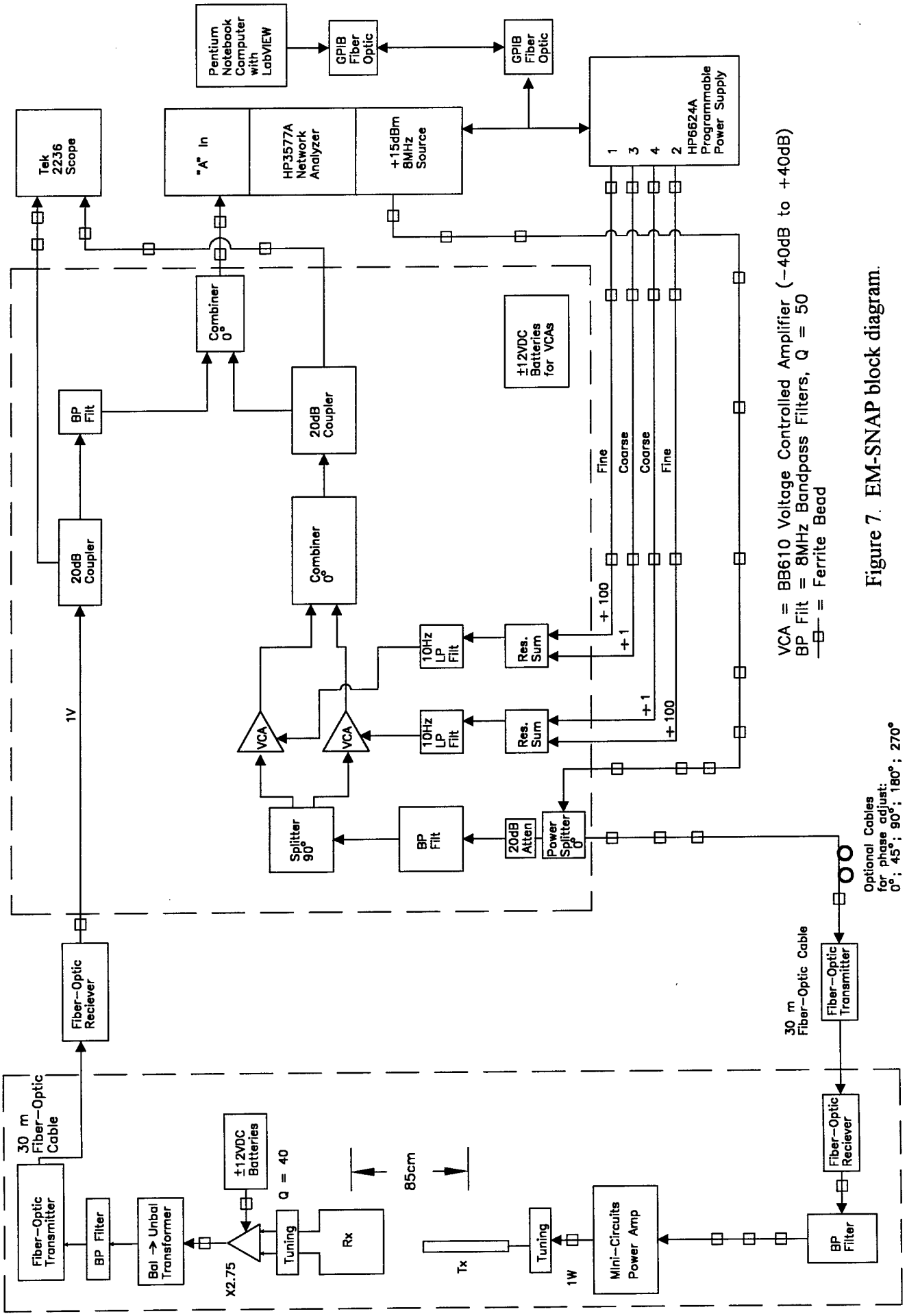


Figure 7. EM-SNAP block diagram.

computer (connected via a fiber optic link) to control the whole system. The user interface is built around LabVIEW.

6 AUTOMATIC COMPUTER-CONTROLLED NULLING SOFTWARE

It can take many minutes to find a deep null when using the manual nulling system with the LabVIEW interface. Thus a practical system requires automatic computer-controlled nulling capabilities. Instead of relying on a relatively slow iterative optimization algorithm, we modeled the behavior of the VCAs and the electrical-nulling circuitry via analytical equations that were obtained empirically. These equations were used to efficiently predict the values of the DC voltages that must be applied to the VCAs in order to obtain a null, thus allowing the computer to very rapidly iterate to the desired null. In addition to nulling, the computer algorithm allows the output of the detector to be sampled on a regular or one-shot basis. Sampled data are displayed in real-time to the user (see Figure 8) and can be saved to disk in user supplied or default file names. Sampled data are in a simple tab-delimited text format suitable for direct import into EXCEL or MathCAD.

The received signal is "nulled" via a cancel signal whose real and imaginary components can be amplified or attenuated via four control voltages (coarse and fine for each component). The received signal and current cancel signal are recombined to produce a detector signal.

The user interface provides for display of the detector signal in several real-time graphical forms in addition to floating-point representation of current real/imaginary components and current decibels of nulling. The user interface also provides for automatic or manual control of cancel-signal amplification voltages as well as display of these voltages.

The system hardware allows generation of a cancel signal only in a limited region of complex signal space. It is possible for the received signal to occupy a region where it cannot be completely or even partially canceled. Thus the software also checks to make sure that the received signal can be canceled, reporting necessary magnitude and phase adjustments which must be made to cancel a received signal. The necessary phase adjustments are currently handled by adding additional electrical length in the circuit. This is denoted by "Optional Cables for phase adjust" in Figure 7.

Basic use of the program is quite simple and consists of program start-up, application of the coarse null DOS program, followed by any number of applications of the fine null DOS program, and finally by data acquisition and storage of the detected signal.

Although nulling systems are not uncommon, nulling inter-related in-phase and quadrature components to very small levels is a challenge. Small non-linearities and cross-talk which is present in a real system further complicate the nulling process.

A user's manual, which documents the software developed by Terry Leach, is included in Appendix A. Additional programmer's notes are included in Appendix B.

7 TEST FACILITIES

7.1 Description of the Avra Valley Test Site

A vital resource for this project was our Avra Valley Geophysical Test Site [15]. A diagram of the site is shown on Figure 9. The test site is divided into 36 cells, each of which is designed

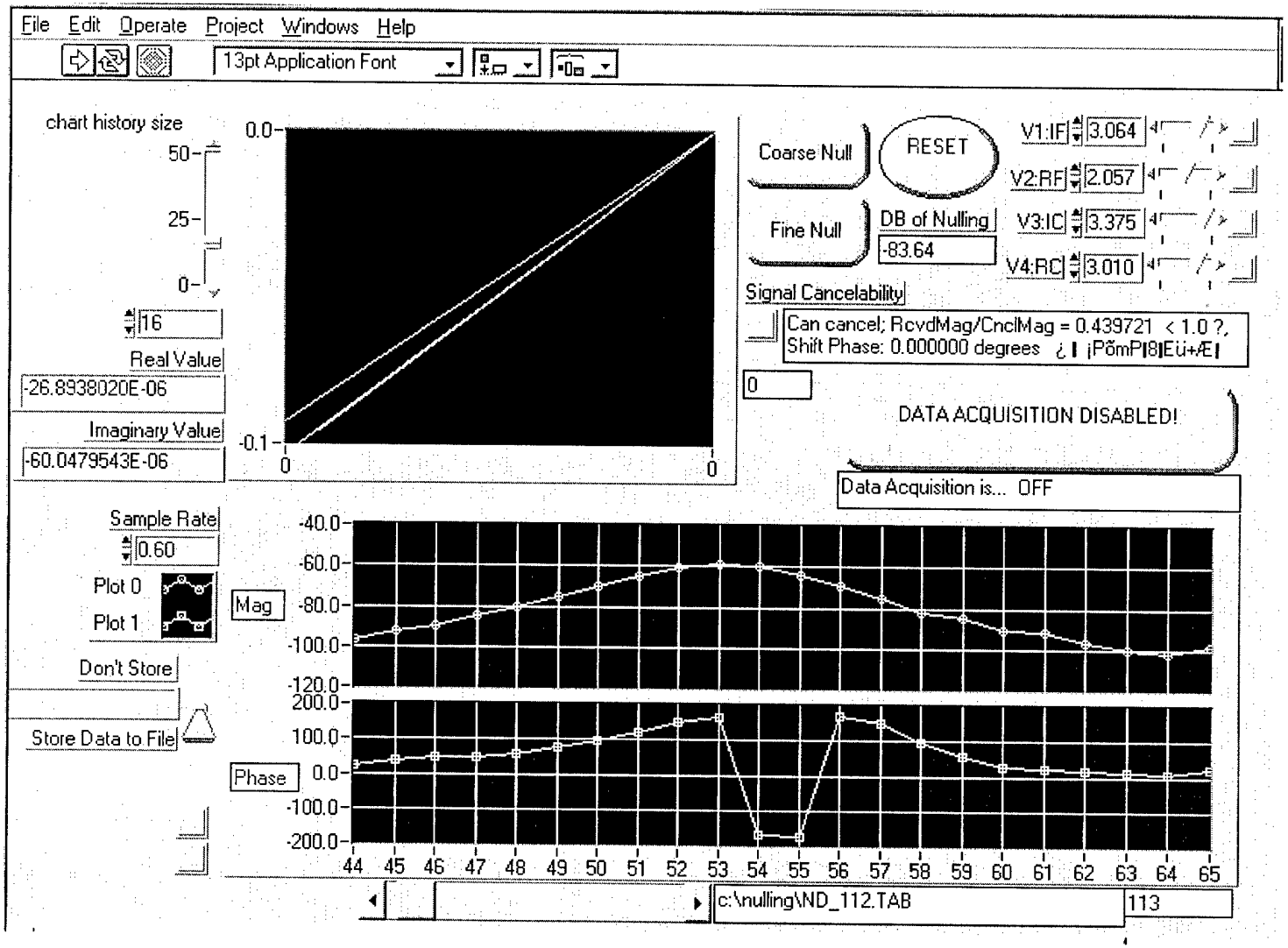


Figure 8. Screen capture from the Pentium notebook computer running the LabVIEW user interface.

to contain a separate experiment. Approximately half of the cells are currently developed for experiments. We have numerous targets buried in these cells.

7.2 Buried-Pipe Test Facility

In order to test the system, at our Avra Valley Test Site, we buried an 18 m long, 46 cm diameter PVC pipe at a depth of 0.6 m to the top of the pipe (Figure 10). The PVC pipe was filled with water and allowed us to move 3D targets such as small sheets or pipes through the PVC pipe while keeping the transmitter and receiver coils in a fixed position. The PVC pipe has screws through the walls of the pipe which are spaced every 25 cm around and 25 cm along the pipe to allow galvanic currents to flow between the target in the water and the surrounding soil. This test facility allowed us to determine the differential target response without the clutter associated with an inhomogeneous background. In order to validate the test facility, before burying the PVC pipe, we first buried a 5.55 m by 0.3 m aluminum sheet 1 m deep in the trench. For these tests, we measured the ellipticity of the magnetic field, which is defined as the ratio of the minor to major axes of the ellipse that is traced as a function of time by the magnetic field vector. We also measured the ellipticity across the sheet placed within the water-filled, buried PVC pipe. The responses for the sheet in the soil and in the PVC pipe are plotted in Figure 11. The close agreement between both sets of measured data indicates that the response due to a target in the PVC pipe accurately predicts the response for the same target buried in the soil. The response shown in Figure 11 is dominated by the metal sheet response. To verify this, we conducted an ellipticity survey over the buried, water-filled PVC pipe in the absence of the metal sheet and found a greatly reduced response.

We also buried four smaller PVC pipes horizontally at various depths at the Avra Valley test site, i.e., at 0.1 m, 0.4 m, 0.7 m, and 1.3 m. In order to allow for the flow of conduction currents through the PVC pipes, we again drilled a large number of holes in the PVC pipes and inserted metal screws. Then we filled the pipes with water so as to not interrupt the process of current channeling which occurs for conductive objects buried in conductive soil. We also constructed a similar test site behind the Department of Mining and Geological Engineering building on the University of Arizona campus. These additional pipes allowed us to test the response of EM-SNAP for smaller targets at various controlled depths.

In the prototype EM-SNAP system, the transmitter and receiver coils were mounted on a platform which rolls smoothly on tracks (Figure 5). However, for our initial tests, we held the transmitter and receiver coils fixed and moved the targets. This was accomplished by pulling conductive and dielectric targets through one of the buried PVC pipes. In essence, these tests isolated the target response by removing the background response. These tests were necessary to determine the "ideal" performance of the EM-SNAP system, i.e., the response for a target in a perfect horizontally-layered earth.

8 EXPERIMENTAL TESTS

8.1 Tests with Conductive Targets

For our initial tests, we employed metal pipes and sheets of various dimensions as our targets. They were attached to a rope which allowed us to pull them through one of the buried PVC pipes.

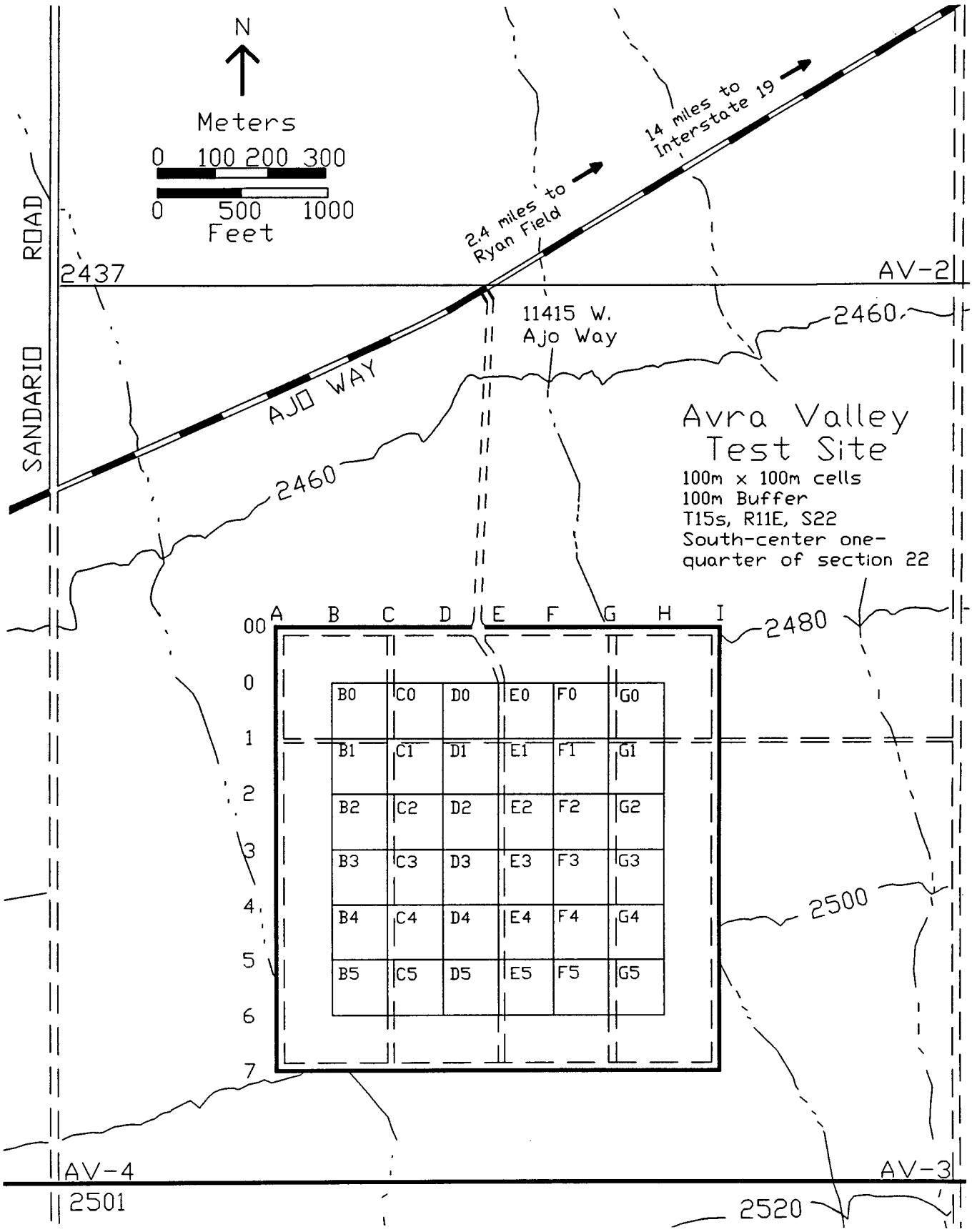


Figure 9: Avra Valley Geophysical Test Site.

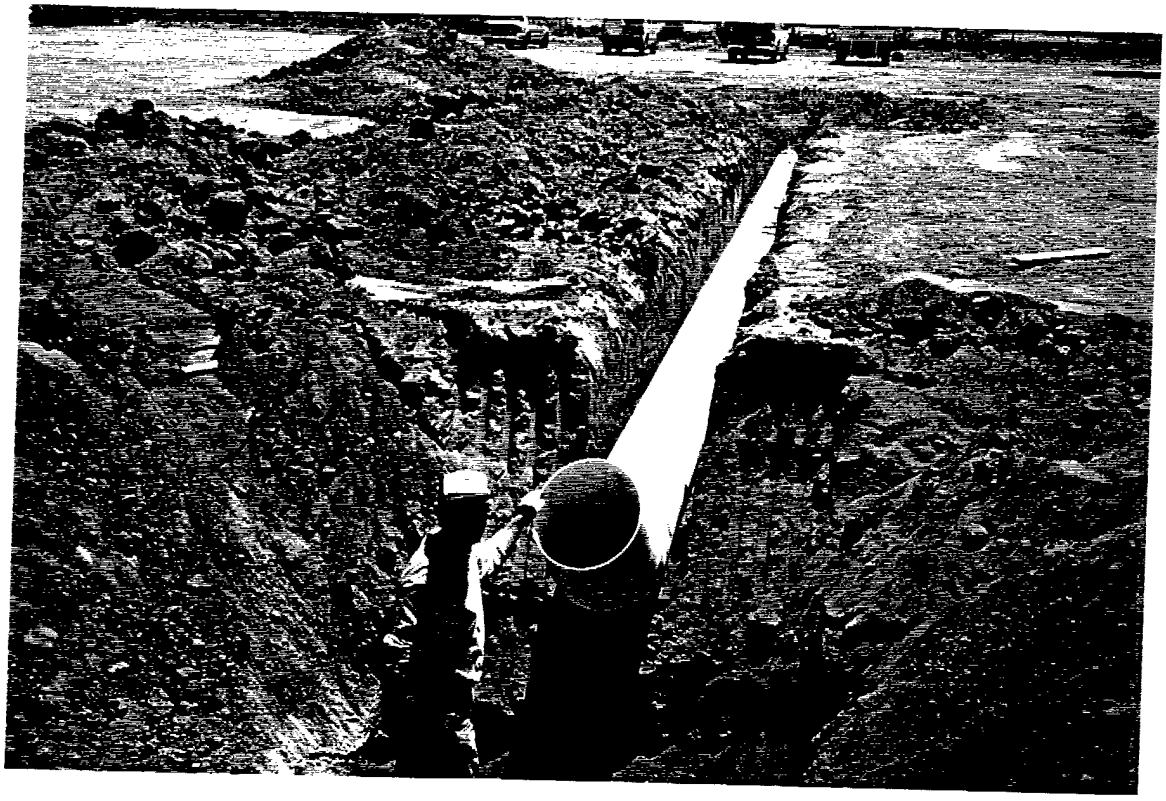


Figure 10: A picture showing the construction of the buried-pipe test facility.

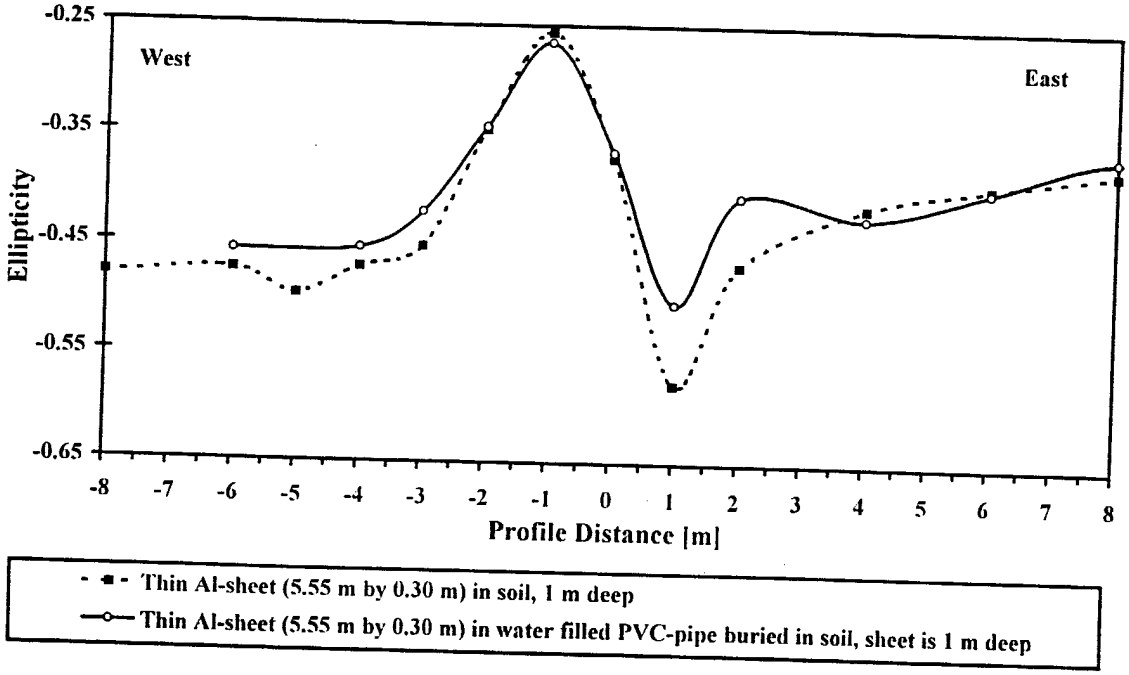


Figure 11: Ellipticity data for the metal sheet buried in earth versus inside the PVC pipe.

The transmitter/receiver assembly was first placed on the surface of the earth over one of the buried PVC pipes. After nulling the field (both geometrically and electrically), a small conductive target was pulled through the buried PVC pipe, and the received field was monitored. We found that we were able to detect targets which produced secondary fields on the order of one one-millionth the magnitude of the primary field during these initial tests.

An example of a typical conductive target response is shown in Figure 12. Note that the target provided a 35 dB magnitude and 190° phase signature. For comparison, the depth to this target is too large to allow detection by currently available high-frequency GPRs in these lossy soils. Note that the signal level does not return to the -84 dB background level at station 21. This is due to a 5 dB drift associated with the electronics in the prototype EM-SNAP system.

8.2 Tests with Dielectric Targets

Our objective is to map organic fluids, which are dielectric targets. For our initial experiments, we used a simple dielectric target, i.e., a 92.0 cm long by 25.0 cm diameter PVC pipe filled with sand. The target was attached to a rope which allowed us to pull it through the buried PVC pipe test facility. The depth to the center of this target was 0.94 m. The transmitter/receiver assembly was first placed on the surface of the earth over the test facility. After nulling the field, the dielectric target was pulled through the PVC pipe, and the received field was recorded. The magnitudes and phases of the received signals associated with three passes of the dielectric target through the PVC pipe are shown in Figure 13. This Figure shows that the dielectric target exhibits a 15 dB magnitude and 140° phase response relative to the nulled background response. It also demonstrates the repeatability of the measurements and that a larger response (i.e., 20 dB) can be obtained with a deeper null. As expected, the dielectric target exhibits a smaller response than the previously discussed conductive target.

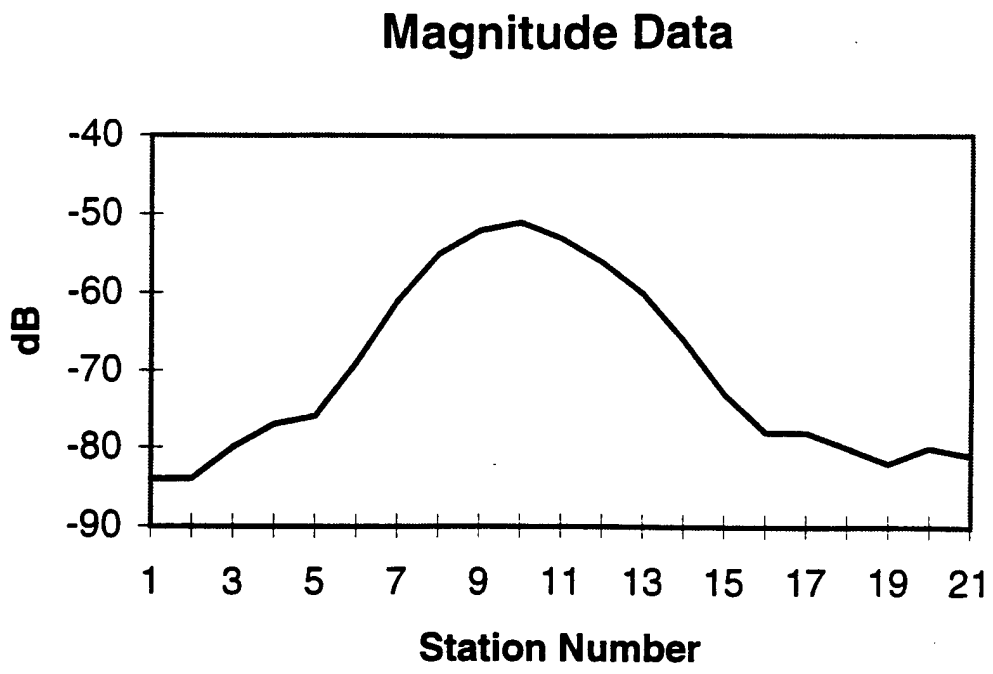
Note that two of the curves in Figure 13 clearly exhibit a 5 dB drift, which was similar to the conductive target case. The third set of data shows an even larger drift (12 dB). This is due to the additional nulling in this case, i.e., -93 dB instead of -87 dB. The instrumentation drift is one of the factors which currently limits the capabilities of the prototype EM-SNAP system. In subsection 10.2.2, we discuss a way to overcome this problem.

The data in Figure 13 clearly demonstrates the possibility of detecting dielectric targets in the absence of background clutter. EM-SNAP will rely on differential measurements, which can be thought of as recording the derivative at a number of different points on one of the curves in Figure 13. Unfortunately, the data will be contaminated by any significant changes in the composition of the background. Thus, new methods must be developed to distinguish the background variations from the distinctive target signature.

8.3 Implementation Issues

The previously-discussed preliminary tests, which were carried out in our buried-pipe test facility, clearly demonstrated the tremendous sensitivity associated with null-field receivers. In the absence of background variations, deeply-buried subsurface targets are clearly resolvable using the prototype EM-SNAP system. The EM-SNAP technology holds great promise for the very difficult problem of mapping subsurface organic contaminant plumes.

In order to test the application of this technology to realistic stationary targets, differential field data were collected by moving the EM-SNAP transmitter/receiver array instead of moving



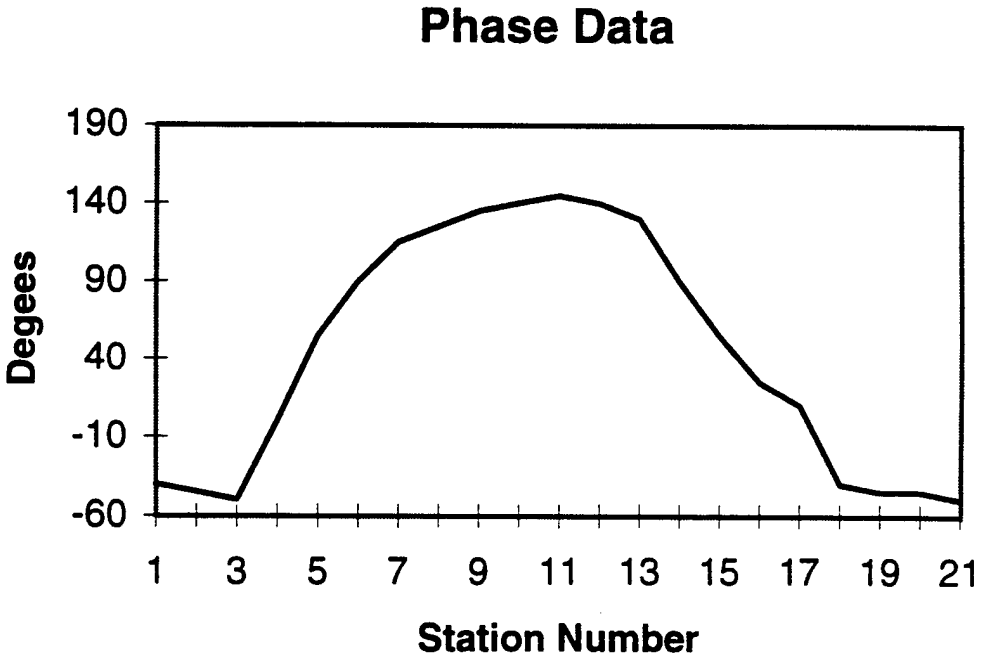


Figure 12. EM response over a conductive target located at station number 10 and 1 m depth. The magnitude and phase data were collected with the prototype EM-SNAP system while pulling a conductive target through the buried-pipe test facility. The conductive target consisted of a 2 m by 0.3 m aluminum sheet. The depth to this target is too large to allow detection by currently available high-frequency GPRs in these lossy soils.

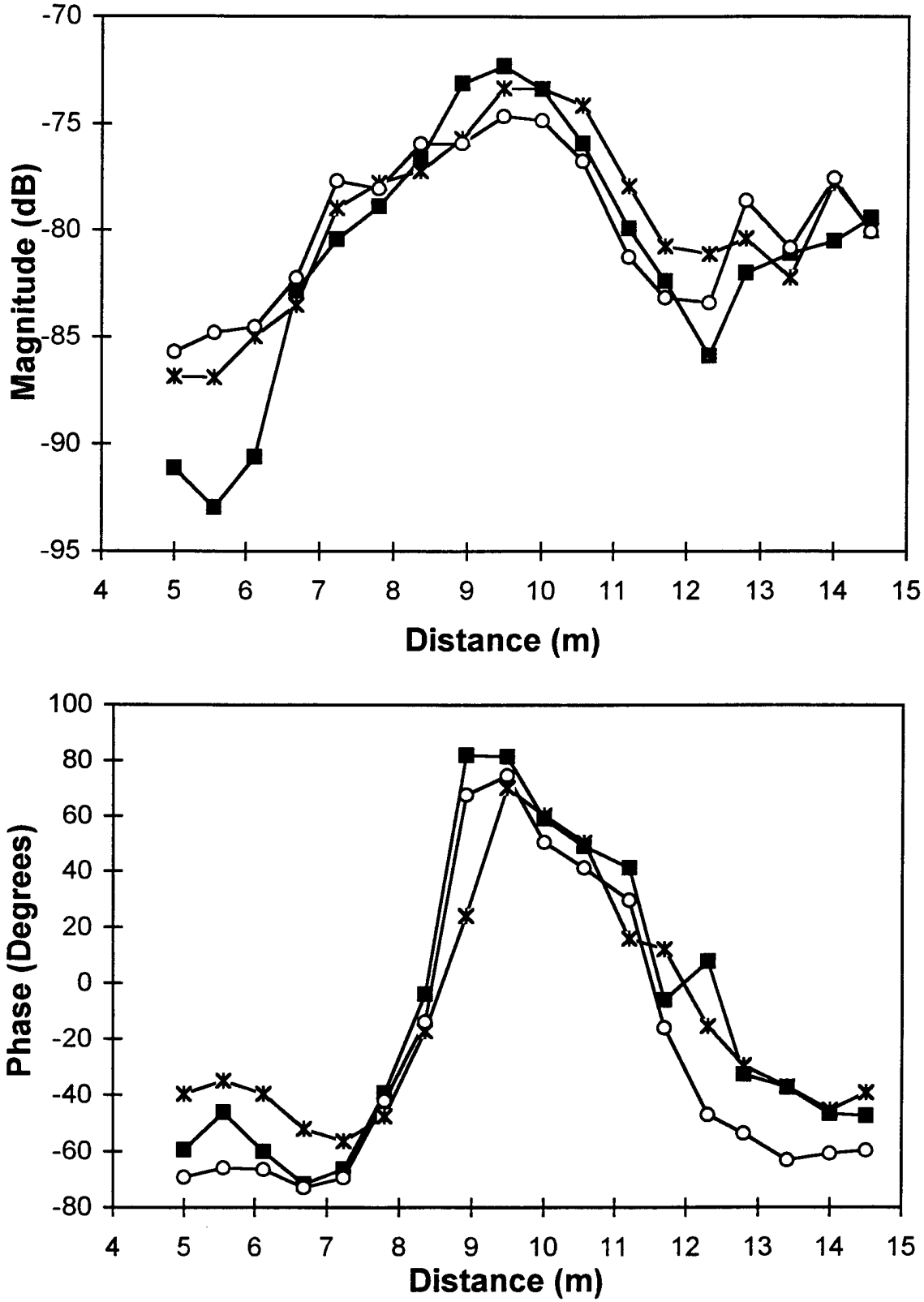


Figure 13: EM response over a small dielectric target located at a 1 m depth. The magnitude and phase data were collected with the prototype EM-SNAP system while pulling a dielectric target through the buried pipe test facility. The dielectric target consisted of a 92.0 cm long by 25.0 cm diameter PVC pipe filled with sand. The depth to this target is too large to allow detection by currently available high-frequency GPRs in these lossy soils.

the subsurface targets. The idea was to null the received magnetic field at one transmitter/receiver location, and then move the transmitter/receiver array a small horizontal distance in order to record the differential magnetic field. This process would then be repeated until the entire survey area was covered. Our tests were carried out over a relatively-flat ground, however, we believe that the technique can be adapted to an undulating ground by using relatively-small spatial increments, i.e., maintain a local tangent to the ground.

Our initial tests carried out over stationary subsurface targets were disappointing. During these tests, we measured the differential magnetic field responses for: 1) a subsurface target buried in the ground, and 2) for the ground in the absence of a target, i.e., the background response. Note that our buried-pipe test facility is ideally suited for such tests since the target can be easily removed for the background test. Since our previous tests indicated a large differential response when we moved the target (see Figure 12), we expected to see a large difference between the differential responses associated with the target and the background. Unfortunately, the two responses were very similiar, so it seemed as though the target response was overwhelmed by the background response.

Since the movement of the cart was a new variable, which had not previously been tested, this test did not separate out what portion of the differential response was due to the background versus the amount due to factors associated with the movement of the cart. Therefore, we devised a new experiment to provide additional insight into this issue. In order to isolate the effects due to the movement of the cart, we carried out further tests over a well-defined background. We decided that a perfectly-conducting earth would be the best environment for such tests since the differential magnetic field response should be zero in this case. We used an 8m $\times$ 15m ground plane, which was constructed by riveting thirty-five 1.2m $\times$ 3.0m aluminum sheets together, for these tests. Tests carried out using the LASI high-frequency ellipticity system demonstrated that the ground plane provided a very good approximation for a perfectly-conducting earth. Tests conducted with the prototype EM-SNAP system over the ground plane yielded large differential responses that were very similar to those obtained over the finite-conductivity earth. Since negligible differential responses are expected for measurements made over the ground plane, these tests proved that hardware problems were the cause of the unexpectedly large differential field measurements recorded over both the finite- and perfectly-conducting earths.

Further laboratory tests were employed to isolate the hardware problems. We found that even slight bending of the optical fibers caused changes in the output voltages for the amplitude-modulated (AM) signals. Even though the absolute voltage changes were small, they led to very large differential signals since these changes were large relative to the signal levels obtained via nulling. The AM optical-fiber links were employed to optically isolate the cart and reduce electric field pickup. The AM optical-fiber system was all that was available for the tests but a frequency-modulated optical-fiber link could greatly reduce this problem for future tests. A calibration technique, such as the one discussed in subsection 10.2.2, could also be used to remove variations associated with the optical-fiber links as well as variations due to other electronic components in the system.

We also believe that the unwanted electric field picked up by the receiver coil may have contributed to the problems we experienced. We have found that there can be substantial variations in the electrical field strength at closely-spaced points. Therefore, it is crucial to shield the receiver coil from these electric field effects. The prototype EM-SNAP system employed a standard Faraday shield on the receiver coil. Further laboratory tests demonstrated that doubly-loaded loop antennas

[16, 17] provide much better shielding from unwanted electric field effects.

In conclusion, the prototype EM-SNAP system has hardware limitations which make it impossible to currently test the application of the null-field receiver when spatial movement is employed to measure differential changes in the magnetic field. However, based on our studies associated with moving targets, it is clear that differential measurements, obtained with our null-field system, offer great potential for the kind of high-sensitivity measurements needed to map subsurface organic contaminant plumes. Furthermore, we believe that further basic research on the EM-SNAP concept will provide the in-depth understanding necessary to overcome the hardware limitations in the current prototype EM-SNAP system.

9 NUMERICAL EM MODELING

We employed electromagnetic modeling algorithms to help guide the EM-SNAP design process. Our initial studies were carried out using algorithms which are designed to handle conductive targets buried in a layered half space. We chose to use LLNL NEC4.1D [22, 23] and the University of California at Berkeley (UCB) Sheet algorithm [24, 25] for this purpose. Later, we used the Temporal Scattering and Response (TSAR) program [26] to model more complicated conductive targets and some dielectric targets.

Since NEC and TSAR were originally developed for higher-frequency applications, we had to verify the accuracy of these algorithms before they could be applied to our relatively-low-frequency geophysical application. NEC was first compared with a quasi-static sphere model to better understand the accuracies obtainable for solid conductive targets modeled by wire segments via NEC. We found very good agreement between the results produced by the two algorithms in all cases we tested. Thus, NEC proved to be capable of modeling coil antennas and spherical targets in free space even for the close electrical spacings required in geophysical applications.

Since most geophysical applications involve transmitters, receivers, and targets which are located in the vicinity of an earth/air interface, we also had to test out the accuracy of the half-space calculations produced by NEC. We employed the UCB Sheet algorithm for this purpose. We found that NEC had difficulties producing the correct results in some of the cases we tested. After a discussion with Dr. Burke, we believe that the difficulties we experienced with NEC are due to the fact that the Sommerfeld integral interpolation table employed by NEC was not optimized for the relatively-low-frequency geophysical applications that we are interested in. Since there is not a fundamental low-frequency limitation in the integral-equation formulation employed by NEC, we believe that it should be possible to modify (optimize) NEC for lower-frequency geophysical applications. However, because of the difficulties experienced with the current version of NEC, we relied on the Sheet algorithm and TSAR for many of our numerical studies. TSAR was validated via comparisons with other published results.

Once the numerical EM modeling algorithms were thoroughly tested, we then employed the algorithms in the initial phase of the design for the prototype EM-SNAP system. NEC was used to obtain a better understanding of the advantages of various geometrical-nulling configurations. Details on the EM modeling efforts, which were conducted by Pat Debroux, are given in Appendix C.

10 FUTURE RESEARCH

10.1 A Proposed Approach: Focus on the Front-End Data Acquisition in Order to Greatly Improve the Back-End Data Interpretation and Target Identification

In principle, EM methods are "imbued with unlimited resolving power" [18]. If we could obtain data with unlimited precision, at all frequencies, and at all points on the surface of the earth, we should be able to accurately map subsurface organic contaminant plumes. Unfortunately, the response from an environmental target may be minute and may occur only over a limited frequency range and only within a small area.

The challenge, therefore, is to develop a data-acquisition approach which will allow us to detect the minute differences in EM response due to an organic contaminant plume. Furthermore, the system must be broadband and be capable of economically acquiring dense data sets. If all these characteristics can be achieved, the system would provide a fundamentally-new capability to obtain highly-diagnostic responses from environmental targets. With a much more diagnostic response obtained with these novel data-acquisition procedures, one can then apply existing imaging algorithms, as well as new techniques developed specifically for such a system.

We have focused our basic research on the development of null-field receivers, which we have shown have the potential to extract minute secondary field (target) responses from the much larger primary field data.

10.2 Simultaneous Calibration

The theoretical justification for trying to improve the measurement accuracies in electromagnetic soundings is well established. For example, Slichter, in 1933, showed that if one were able to measure electromagnetic fields with perfect precision, at all points on the surface of the earth, and at all frequencies, then a unique solution exists for the variation of conductivity with depth [19]. Fullagar investigated horizontal-loop frequency soundings and demonstrated that these methods "are in principle imbued with unlimited resolving power" [18]. Unfortunately, only a small amount of error in the measured electromagnetic fields can lead to a large amount of error in the interpreted subsurface resistivity structure. The equations relating the surface EM fields to the resistivity distribution are highly nonlinear. How much error can be tolerated in the observed fields without leading to large uncertainties in the interpreted resistivity structure depends upon the particular earth model. The key point here is that the basic theoretical framework for this work is well established. What needs to be done is: using a very-high-accuracy EM probing system, we need to collect continuous profiling data, and then determine what subsurface information can be obtained from this data set. Unfortunately, the component drift and "noise" limited the accuracy of the data that can be obtained using our prototype EM-SNAP system. We feel that a simultaneous calibration is required to collect the very-high-accuracy EM data that are necessary for this inverse problem.

10.2.1 Calibration Employed in LASI Ellipticity Systems

The three requirements for very high-accuracy ellipticity measurements are: (1) high relative gain and phase accuracy of the H_x and H_z channels, (2) accurately known angles between the coils,

and (3) minimum cross talk between the H_x and H_z channels. The first requirement is satisfied using one of the two calibration techniques discussed below.

The frequency response of all the analog components in the receiver (the preamplifier, the filters, even the receiving coil itself) changes with time, temperature, and shock as it is transported in the field. Therefore, no calibration, no matter how accurate, can be accurately applied to a measurement made at a substantially different time. It is necessary to calibrate the overall frequency response at the same time as data are being collected. Note that time-domain systems are not immune to drift in the equipment response. Even if a wideband system is used which has very little analog filtering, the receiver coil and amplifiers still have a frequency-dependent response which may vary with time.

A very-high-accuracy, relative gain and phase calibration is accomplished using either the AFCAL (Adjacent Frequency Calibration) or STACAL (Swept Tracking Automatic Calibration) method. The AFCAL and STACAL methods are improvements over the previously developed HASCAL (High-Accuracy Simultaneous Calibration) method developed by Sternberg and Nopper [20, 21].

In the AFCAL calibration method, if a frequency of f_0 Hz is being transmitted, then frequencies of $f_0 + \Delta f_c$ Hz and $f_0 - \Delta f_c$ Hz will also be transmitted from a calibration coil near the receiver coil. The calibration frequencies are transmitted at the same time as the data frequency. A linear interpolation may be used to find the system response (both magnitude and phase) at f_0 from the closely adjacent calibration frequencies. The exact value of Δf_c is approximately one-thousandth of f_0 in order to obtain system accuracies of 0.001%. It is also possible to transmit more than two closely spaced calibration frequencies so that a more sophisticated interpolation than simply linear can be employed.

In the STACAL method, both calibration and data signals are transmitted at discrete frequencies of f_1, f_2, f_3, f_4, etc. The process begins by first transmitting a calibration signal at f_1 for a certain period of time. Then a data signal is transmitted at frequency f_1 and a calibration signal at f_2. These two signals are transmitted during the same time period. Next a calibration signal is transmitted on f_1, data on f_2, and calibration on f_3. In other words, the calibration signal is swept and tracks the data signal. The calibration is automatically performed by a computer. Although the calibration of f_1 does not occur at exactly the same time as data measurement of frequency f_1, the time delay between calibration and data is small. Furthermore, since the calibration is performed immediately before and immediately after the data recording at f_1, the two calibrations can be averaged to substantially eliminate any small drift that might have occurred over this short time. It is important to note that a data frequency is being recorded at the same time as the previous frequency and the next frequency are being calibrated so there is no lost time due to the calibration process. If conventional calibrations were used before and after the data measurement, prior to moving to the next data frequency, the recording time would be tripled.

10.2.2 Proposed Incorporation of AFCAL Calibration into EM-SNAP

As previously discussed, AFCAL calibration has been used very effectively in our high-frequency ellipticity system. After nulling as much as 120 dB, our prototype EM-SNAP system is susceptible to significant drift and errors in calibration. Since AFCAL, is very effective in removing these drifts and uncertainties in the equipment response, we plan to use this technique in future EM-SNAP prototypes. Using this approach, we expect to produce much-higher-accuracy measurements than

have ever been recorded in the past at these frequencies. This is a particularly challenging exercise at the high frequencies that are used in this project. High frequencies are needed, i.e., between 1 MHz and tens of MHz, so that we can record both the conductivity and dielectric constant information which is important for mapping the flow of organic fluids in the subsurface.

Based on what we have learned during this grant, we have submitted proposals for the continuation of basic research on the EM-SNAP concept. The AFCAL calibration concept is included in these proposals.

11 CONCLUSIONS

This project has been very successful in showing a promising new direction for high-resolution subsurface imaging. Our tests with a preliminary prototype Electromagnetic Sensitive Null Array Probe (EM-SNAP) showed that we were able to obtain very sensitive measurements over subsurface dielectric targets. Although much more basic research must be done, this approach holds great promise for imaging the flow of organic fluids in the subsurface. Based on the novel methods initially developed in this project, we have proposed further basic research on this technology including high-accuracy calibration, wideband measurements, arrays of sensors, greatly-improved nulling methods, and optimization of polarizations, array sizes, and frequencies for specific targets of interest.

12 ACKNOWLEDGMENTS

We would like to recognize the assistance of two AFOSR Project Managers during this grant: Colonel Martin Lewis and Captain Mike Chipley. In particular, we would like to thank them both for organizing the meetings which introduced us to the Air Force environmental problems and the key players we will need to work with as we apply this technology in the future.

Many students and staff at the University of Arizona contributed to this development. They are listed in the Executive Summary. We would particularly like to thank Terry Leach for programming the automatic nulling, Pat Debroux for the numerical calculations, and Tsylya Levitskaya for the electrical property measurements.

Charles Abernethy played a crucial role in the early electronics development for this project. Charles died February 14, 1996. His passing was a significant loss to our project. He is greatly missed as are his extraordinary knowledge, his unfailing good humor, and his dedication to making this project successful.

13 REFERENCES

[1] M. Maxwell and J. Schmok, "Detection and Mapping of an LNAPL Plume using GPR: A Case Study," *Proc. of the Symposium on the Application of Geophysics to Engineering and Environmental Problems*, Orlando, Florida, pp. 15–23,1995.

[2] S. M. DeRyck, J. D. Redman, and A. P. Annan, "Geophysical Monitoring of a Controlled Kerosene Spill, " *Proc. of the Symposium on the Application of Geophysics to Engineering and Environmental Problems*, San Diego, California, pp. 5-19, 1993.

[3] D. L. Campbell, J. E. Lucius, K. J. Ellefsen, and M. Deszcz-Pan, "Monitoring of a Controlled LNAPL Spill using Ground-Penetrating Radar," *Proc. of the Symposium on the Application of Geophysics to Engineering and Environmental Problems*, Keystone, Colorado, pp. 511- 517, 1996.

[4] A. L. Endres and J. D. Redman, "Modelling the Electrical Properties of Porous Rocks and Soils Containing Immiscible Contaminants," *Proc. of the Symposium on the Application of Geophysics to Engineering and Environmental Problems*, San Diego, California, pp. 21-38, 1993.

[5] B. K. Sternberg, S. J. Thomas, N. H. Bak, and M. M. Poulton, "High-Resolution Electromagnetic Imaging of Subsurface Contaminant Plumes," EPRI Report, EN-7519, Project 2485-11, Electric Power Research Institute, Palo Alto, CA, 1991.

[6] C. E. Glass and B. K. Sternberg, "Leachate Monitoring using EM Visi on Strategies," Final Report, Copper Research Center, Grant from Defense National Stockpile Center (GSA), number DN-004, 1991.

[7] N. H. Bak, B. K. Sternberg, S. L. Dvorak, and S. J. Thomas, "Rapid, High-Accuracy Electromagnetic Soundings using a Novel Four-Axis Coil to Measure Magnetic Field Ellipticity," *J. of Applied Geophysics*, Vol. 30, pp. 235–245, 1993.

[8] B. K. Sternberg and S. J. Thomas, "Applications of High-Resolution Subsurface Imaging Techniques to Water Resource Investigations, " Final Report on USGS Grant No. 14-08-0001-G1726.

[9] M. M. Poulton and B. K. Sternberg, "Identifying Subsidence Hazards using a Unique High-Resolution EM System and Neural Network Interpretation," US Bureau of Mines, Final Report on Contract #1432-J0220004, 1995.

[10] B. K. Sternberg and M. M. Poulton, "High-Resolution Subsurface Imaging and Neural Network Recognition: Non-Intrusive Buried Substance Location," Final Report, DOE Contract, DE-AC21-92MC29101 A001, 172p., 1997.

[11] B. K. Sternberg and M. M. Poulton, "Subsurface Void Detection using a Unique High-Resolution EM Imaging System and Neural Network Interpretation," US Army Corps of Engineers.

[12] G. M. Hoversten. *Comparison of Time and Frequency Domain E.M. Sounding Techniques.* Ph.D. thesis, University of California, Berkeley, 1981.

[13] S. H. Ward, J. Ryu, W. E. Glenn, G. W. Hohmann, A. Dey, and B. D. Smith. "Electromagnetic Methods in Conductive Terranes." *Geoexploration*, Vol. 12, 121-183, 1974.

[14] N. H. Bak, *Development of an Advanced Electromagnetic Subsurface Imaging System and Interpretation of Data*, MS Thesis, University of Arizona, 1991.

[15] B. K. Sternberg, M. F. Miletto, D. J. LaBrecque, S. J. Thomas, and M. M. Poulton, "The Avra Valley (Ajo Road) Geophysical Test Site: Geophysical Surveys, Geological Data, and Initial Development of the Test Site," LASI Report, LASI-91-2, 1991.

[16] H. Whiteside and R. W. P. King, "The Loop Antenna as a Probe," *IEEE Trans. Antennas Propagat.,* pp. 291-297, 1964.

[17] R. W. P. King and C. W. Harrison, *Antennas and Waves: A Modern Approach,* MIT Press, p. 612, 1969.

[18] P. Fullagar, "A Uniqueness Theorem for Horizontal Loop Electromagnetic Frequency Soundings," *Geophys. J. R. Astr. Soc.,* pp. 559-566, 1984.

[19] L. B. Slichter. "An Inverse Boundary Value Problem in Electrodynamics," *Physics,* Vol. 4, 411-418, 1933.

[20] B. K. Sternberg and R. W. Nopper. "High-accuracy, Simultaneous Calibration of Controlled-source EM Field Systems." 56th Annual International SEG Meeting, Houston, Texas, November 2-6, 1986.

[21] B. K. Sternberg and R. W. Nopper. "Method and Apparatus for Obtaining High Accuracy Simultaneous Calibration of Signal Measuring Systems," U.S. Patent #4,613,821, September 23, 1986.

[22] G. J. Burke and A. J. Poggio, "Numerical Electromagnetics Code (NEC)–Method of Moments," *Lawrence Livermore National Laboratories Document,*, UCID-18834, 1981.

[23] G.J. Burke, *Numerical Electromagnetics Code - NEC-4, Method of Moments, Part I: User's Manual,* Lawrence Livermore National Laboratory, UCRL-109338, 1992.

[24] P. Weidelt, "Dipole Induction on a Thin Plate with a Host Medium Overburden," Res. Project NTS 83, No. 89727, *Federal Inst. Earth Sci. and Raw Materials,* Hannover, FRG, 1981.

[25] Q. Zhou, *Audio-Frequency Electromagnetic Tomography for Reservoir Evaluation,* Ph.D. Dissertation, University of California, Berkeley, 1989.

[26] R. R. McLeod, "Temporal Scattering and Response Software User's Manual version 2.3," *Lawrence Livermore National Laboratory,* UCRL-MA 104861 V 2.3, 1992.

[27] Grant and West, *Interpretation Theory in Applied Geophysics,* McGraw-Hill, p515, 1965.

[28] G. S. Lohda and G. F. West, Practical Airborne EM (AEM) Interpretation using a Sphere Model, *Geophysics,* Vol. 41, No. 6A, pp. 1157–1169, 1976.

[29] Mur, G., Absorbing Boundary Conditions for Finite-Difference Approximation of the Time-Domain Electromagnetic-Field Equations, *IEEE Trans. Electromagnetic Compatibility,* Vol. 23, pp. 1073-1077, 1981.

[30] Stewart D., Anderson W. L., Grover, T. P. , and Labson, V. F., "New Instruments and Inversion Program for Near-Surface Mapping: High-Frequency EM Sounding and Profiling in the Frequency Range 300 KHz to 30 MHz," SEG Annual Meeting Expanded Abstract 4/23-27, San Francisco, 1990.

[31] Ryu J., Morrison F., and Ward S. H., "Electromagnetic Fields About a Loop Source of Current," *Geophysics,* Vol. 35, No. 5, pp. 862-897, 1970.

[32] W. L. Anderson, "Approximate Inversion of High-Frequency Electromagnetic Sounding using Complex Image Theory," *Geophysics*, Vol. 56, No.7, pp. 1087-1 092, 1991.

[33] A. Roy, "On the Effect of Overburden on EM Anomalies-A Review," *Geophysics,* Vol. 35, No. 4, p. 646, 1970.

[34] V. K. Gaur, O. P. Verma, and C. P. Gupta, "Enhancement of Electromagnetic Anomalies by a Conductive Overburden," *Geophysical Prospecting,* Vol. 20, No. 3, p. 580, 1972.

[35] B. B. Bhattacharya, D. S. Mukherjee, and D. Chatterjee, "Enhancement of electromagnetic anomalies-A circuit theory approach," *Geophysics,* Vol. 48, No. 9, p. 1248, 1983.

[36] W. A. San Filipo, P. A. Eaton, and G. W. Hohmann, "The Effect of a Conductive Half-Space on the Transient Electromagnetic Response of a Three-Dimensional Body," *Geophysics,* Vol. 50, No. 7, p. 1144, 1985.

14 APPENDIX A: USER'S MANUAL FOR NULLING SOFTWARE

8MHz Null-Signal Control and Data-Acquisition Software

INTRODUCTION

This manual documents the use of the Nulling application software. This software is an integration of LabVIEW and a DOS program on the Windows 95 platform.

The main purpose of the software is to null a received signal on the system hardware thereby creating a sensitive detector. The output of the detector can be sampled on a regular or one-shot basis. Sampled data can be saved to disk in user supplied or default file names. Sampled data are in a simple tab-delimited text format suitable for direct import into EXCEL or MathCAD.

The received signal is "nulled" via a cancel signal whose real and imaginary components can be amplified or attenuated via four control voltages (coarse and fine for each component). The received signal and current cancel signal are recombined to produce a detector signal.

The user interface provides for display of the detector signal in several real-time graphical forms in addition to floating-point representation of current real/imaginary components and current decibels of nulling. The user interface also provides for automatic or manual control of cancel-signal amplification voltages as well as display of these voltages.

The system hardware allows generation of a cancel signal only in a limited region of complex signal space. It is possible for the received signal to occupy a region where it cannot be completely or even partially canceled. Thus the software also checks to make sure that the received signal can be canceled, reporting necessary magnitude and phase adjustments which must be made to cancel a received signal.

Basic use of the program is quite simple and consists of program start-up, application of the coarse null DOS program, followed by any number of applications of the fine null DOS program, and finally by data acquisition and storage of the detector signal.

INSTALLING THE PROGRAM

The nulling system software is a collection of files residing on a single floppy disk. It is intended to run on the Windows 95 platform under LabVIEW version 4.0.

HARDWARE REQUIREMENTS

GPIB interface card.
Sufficient computing resources to run Win95 and LabVIEW
1 MB additional disk space plus space for data collection
Monitor and video card capable of 800X600 screen resolution

SOFTWARE REQUIREMENTS

GPIB drivers for DOS *and* Win95
800X600 display driver active
LabVIEW 4.0 installed
Creation of a DOS Only icon on the Win95 desktop
(see the National Instruments GPIB install manuals)
Location of the supplied PIF files on the Win95 desktop

Place the installation disk in your floppy disk drive and type

INSTALL<CR>

The following files will be copied to your harddrive...

In C:\Nulling

GPIBTST.EXE	The GPIB interpretor program performing all DOS based commands.
*.GIB	Various command files for GPIBTST.EXE performing operations such as nulling, resetting, and received signal cancelability checks.
*.BAT	Batch files letting you run existing nulling system DOS functionality from a DOS session.
NULLING6.VI	The user interface to the nulling system. This file is used by the LabVIEW application.

In C:\WINDOWS\DESKTOP

*.PIF	Definitions of DOS sessions which will be activated from within the LabVIEW application to perform DOS based functions such as nulling, resetting, and cancelability checks.

NOTE: If you wish to install the software on a drive different from the C drive, you will have to change each of the *.PIF and *.BAT files manually to reflect the equivalent location to C:\Nulling.

STARTING THE PROGRAM

To start the program simply double-click the "Nulling" short cut icon on the Windows 95 desktop. Alternatively you may first open LabVIEW and then open C:\NULLING\NULLING6.VI.

Once opened, the program will make 1 automatic pass at which point it will restore your previous operating conditions and check received signal cancelability. This process takes approximately 12 seconds to complete. You will see the "Signal Cancelability" text box change from "UNDEFINED" to display results of the cancelability test.

At this point you have the option of running the program continuously or a single shot at a time. Both these options are accomplished from the LabVIEW tool bar, located just below the program's title bar.

General program operation is discussed in the following section.

OPERATIONAL OVERVIEW

Upon program start up, the application performs three main functions. First it loads 7 parameters which characterize the state of operation the last time the program was run. These parameters are...

The default file number for the next automatic data storage file.
The sample rate last in effect.
The Re/Im chart history size.
The Data Acquisition Button state.
The current Data Acquisition state.
The data storage control flag.
The name of the current data acquisition file.

Next the program starts a DOS session running GPIBINT.EXE to check received signal cancelability. The results of this test will enable or disable the two nulling buttons on the front panel. Finally a single data acquisition is performed to display the current Re/Im components and determine the current values of the four programmable voltages.

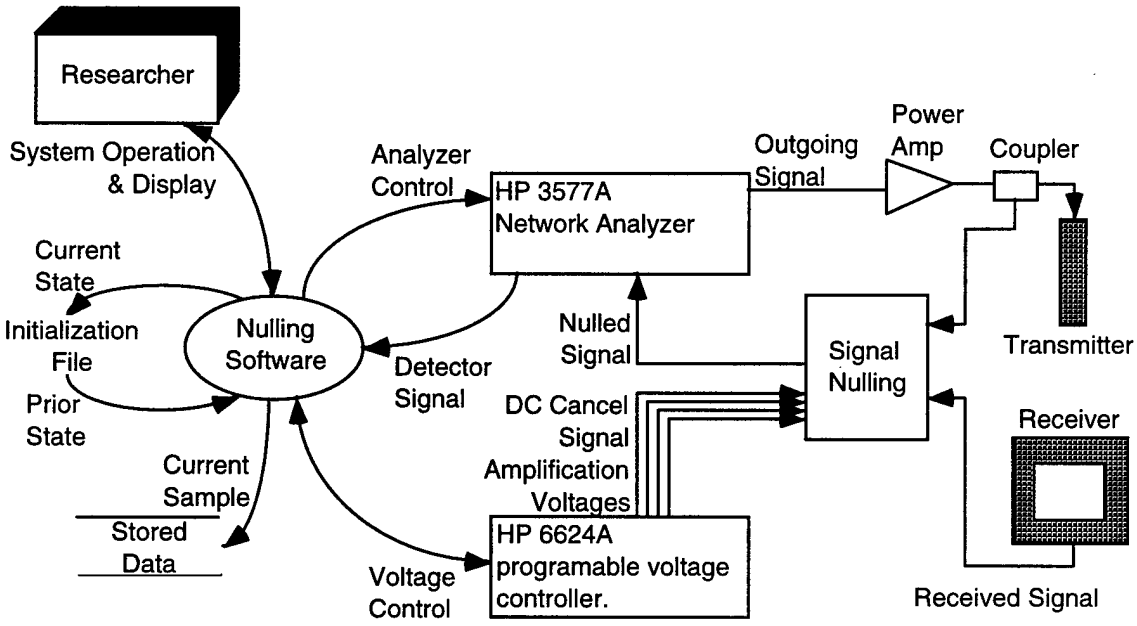
Upon completion of the first automatic pass, the program then becomes idle waiting for you to run it either continuously or on a one-shot basis. Both these operations are accomplished through the standard LabVIEW tool bar.

The next section discusses how data are processed by the software.

GENERAL FLOW OF DATA

The Nulling Software facilitates the use of the Transmitter/Receiver antenna pair as a detector. The outgoing signal from the network analyzer is amplified and sent to the TX antenna. The cancel signal is derived *directly* from the TX signal via a 30db coupler. The cancel signal is in turn split into real and imaginary components which can then be separately amplified or attenuated using the programmable voltage supply. The objective is to "null" the received signal using a cancel signal of the same amplitude but 180 degrees out of phase with the received signal. The resulting detector signal (often 70db to 80db below the raw received signal) becomes a very sensitive detector to any changes in the immediate physical environment.

The researcher controls the programmable voltage supply either manually or automatically to meet the objective of creating a "null". Real time charts of magnitude, phase, and current detector signal real/imaginary components are displayed to the Researcher. The researcher can control data acquisition on a one shot or sampled basis. The acquired data can be accumulated in a disk file.



On each pass of the LabVIEW nulling program, user controls are checked for changes. This consists of activating requested DOS programs or implementing requested manual voltage changes (part of System Operation & Display). DOS programs perform coarse and fine automatic nulling, reset and received signal cancelability checks. All these operations involve reading the Detector Signal, followed by computation and subsequent implementation of Voltage Control. In most cases the LabVIEW application is synchronized with the DOS programs via pure time delays of a separate duration for each DOS program.

After servicing user controls, the program acquires the current real and imaginary components of the Detector Signal. Once these are acquired, the program checks to see if it is likely that system voltages have

changed (A DOS program was run or the user made manual changes). If so, then the current Voltage Control values are also acquired.

Processing of acquired data begins with computation of Detector Signal magnitude (in units of nulling dB) and phase (degrees). The real/imaginary chart is updated followed by the magnitude and phase strip charts (if data acquisition is active).

Acquired magnitude and phase data are then stored to disk (Current Sample is appended to Stored Data) if appropriate. Following this current operating state is saved to Initialization File via Current State. This consists of seven parameters characterizing current program operation. These parameters are stored in tab-delimited ASCII form in the following order:

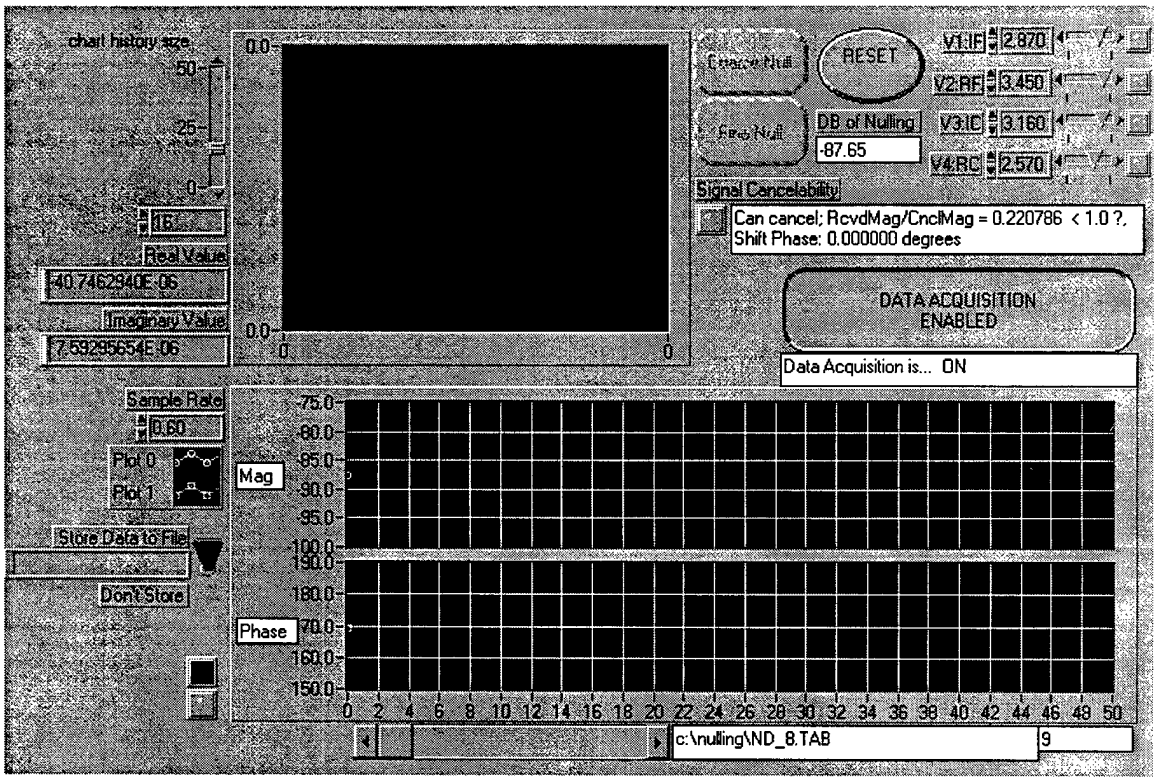
The default file number for the next automatic data storage file.
The sample rate last in effect.
The Re/Im chart history size.
The Data Acquisition Button state.
The current Data Acquisition state.
The data storage control flag.
The name of the current data acquisition file.

Finally the program waits the number of seconds indicated by the "Sample Rate" control before beginning the next pass.

LABVIEW NULLING FRONT PANEL

Shown below is the entire front panel display of the Nulling Software. Each of the controls and displays are discussed in the following document sections. Front Panel is a term employed in the LabVIEW environment to describe the user interface associated with a LabVIEW application.

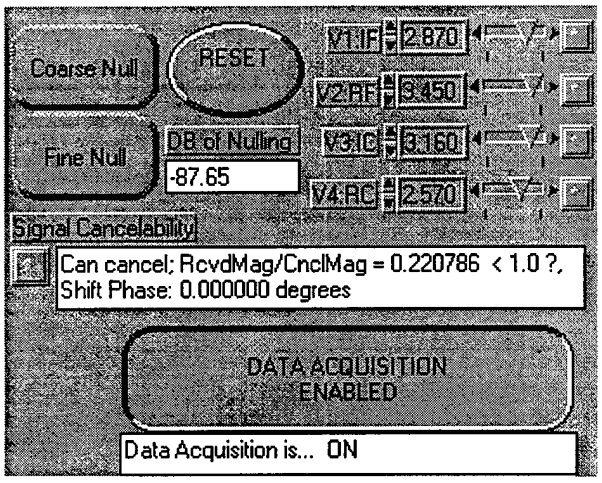
The Nulling Software is an integration of LabVIEW and a DOS program. Predefined sets of commands can be given to the DOS program for interpretation. The command sets are invoked by the Coarse Null, Fine Null and RESET buttons on the front panel.



Basically the front panel is designed to display the current system detector signal in real time. It also coordinates the use of DOS programs and allows for manual voltage control. Finally it allows for storage of acquired data.

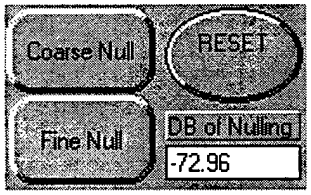
MAIN CONTROLS

There are four areas of functionality provided by the application main controls. The first is control of DOS programs. Secondly, you may control system voltages manually. Thirdly, you can override the signal cancelability test results. Finally, you can activate/deactivate data acquisition. Lets look in more detail at each of these functional groups.



Nulling Program Control

You control DOS programs via the Coarse Null, Fine Null, and RESET buttons. The Coarse Null button starts a DOS program which resets the programmable voltages to zero, and then performs three successive nulling iterations. During these iterations, it attempts to keep the Real:Fine and Imaginary:Fine voltages near 2.0V. This sets up the hardware for maximum controllability for successive use of the Fine Null button. It is suggested (but by no means required) that Coarse Null be the first button pressed when attempting to null the system. It is only necessary to press the Coarse Null button *once* for each new nulling attempt. Because the Coarse Null button resets the programmable voltage supply to zero prior to each Coarse Nulling sequence no improvement in nulling is obtainable by pressing the button multiple times. *Unless* the physical environment has changed significantly.

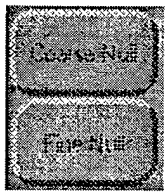


The Fine Null button can be used at any time, any number of times subject to controllability limits on the fine voltage adjustments. It is suggested that you use Coarse Null *once* before Fine Null but this is not a requirement. If you choose to simply start using Fine Null multiple times, you run the risk of running into a physical adjustment limit on the fine voltage adjustments as you reach a null.

Note that you are free to intermix manual voltage control with the Fine Null button in any order.

The RESET button resets the programmable voltages to 0.0V *and* performs a cancelability check. Note that this check *must* be positive for the Nulling buttons to be enabled. It is not possible to run one of the nulling programs until the cancelability check has been passed *or manually overridden*. See the section devoted to Cancelability Indicator & Automatic Control for the procedure on manual override of the cancelability check.

If the nulling buttons appear as shown below, the signal cancelability test has failed. You will need to correct any hardware or received signal problems and then press the RESET button.

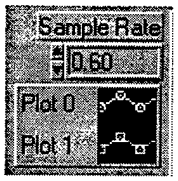


The DB of Nulling display shows you the current decibels of nulling *relative* to a 1V signal on the Network analyzer.

One final note on automatic/programmatic voltage control. It is possible to halt LabVIEW program operation using the stop button. In this state you could conceivably depress one of the nulling program buttons *and* make some manual voltage entries. When program operation was resumed there would be both an automatic voltage change required by the relevant nulling program and a manual voltage change entered by the user. You could also achieve this condition if you were very fast with the mouse. Should this condition arise *programmatic voltage changes take priority over manual voltage changes; the manual voltage changes are discarded.*

Controlling the Sample Rate or Chart Update frequency

Through the "Sample Rate" control you set the frequency of program operation. If the program happens to be in data acquisition mode, this also sets the sample rate for data acquisition. Note that this controls how fast you can set programable voltages or refresh the real/imaginary chart.

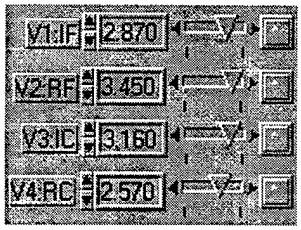


To change the program operation rate, click the numerical entry box and enter your desired operation rate in seconds. Values below 0.6 seconds are not allowed, and if entered will be set to 0.6 seconds. The reason for this limit is to allow settling time for the network analyzer which has been configured to read 8MHz signals on a 10Hz bandpass. Faster sample rates (you could edit this control by stopping the program and placing the .VI file in edit mode) will simply cause the data acquistion operation to fail and will report the most recently acquired valid values.

You can also use the up and down arrows to the left of the sample rate entry box to increment or decrement the sample rate in tics of .1 seconds.

Voltage display and optional manual control

The manual Voltage control section serves as both a display and control for system voltages. The program detects manual changes to the current voltages and sends the new voltages to the HP 6624A *only* when they have changed. Likewise voltages are only read when they are likely to have changed (program start up, DOS program invocation, manual entry). This means that *any drift* in the voltage supply will *not* be displayed on your front panel. The displayed voltages are not a real time display of current voltages. This was done to minimize unnecessary GPIB traffic.



To within the limits of the HP 6624A, these displays are an effective indication of system voltages. Lets look at the ways in which you can make manual voltage changes.

Direct Digital Entry

One way to change a given voltage is to directly enter the value in the box provided. Place the mouse cursor over the entry box and click once. Enter the new value and hit return or click anywhere else on the front panel. If the program is running your change will be detected and sent to the programmable voltage supply.

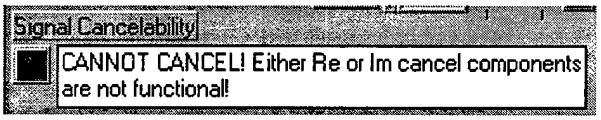
If the program is not currently running, your change will be sent the next pass of the program.

You can also change the voltage by "dialing" it up or down using the small arrows to the left of the entry box. Place the mouse cursor over the appropriate arrow and either click once or hold the left mouse button. A single click will cause the voltage to change +/- one division (currently 0.01V). Holding the button down will cause the voltage to scroll in the direction indicated.

Slide Control of Voltage

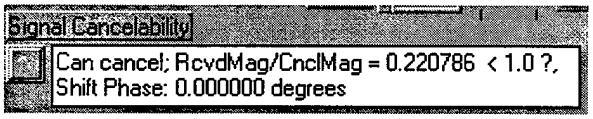
You can also use the slide voltage control as an alternative means to controlling system voltages. To use the slide control, simply place the mouse cursor over the inverted triangle and hold the left mouse button down. Move the slide to any new value desired (values are displayed in the entry box). The voltage thus selected will be sent to the programmable supply.

Cancelability Indicator & Automatic Control



The signal cancelability indicator is both an indicator and a control. Results of the current cancelability check are displayed in the text box to the right of the indicator. This text box will display one of three classes of messages.

The first class simply has the value "UNDEFINED" and indicates that the program has just started up and signal cancelability has not been checked.



The second class indicates signal cancelability. The signal is either cancelable ("Can cancel") or not ("CANNOT CANCEL!"). In each case the ratio of received signal magnitude to cancel signal magnitude is displayed along with any required phase shift to make the signal cancelable. Note that if the signal is cancelable, the required phase shift is simple set to zero.

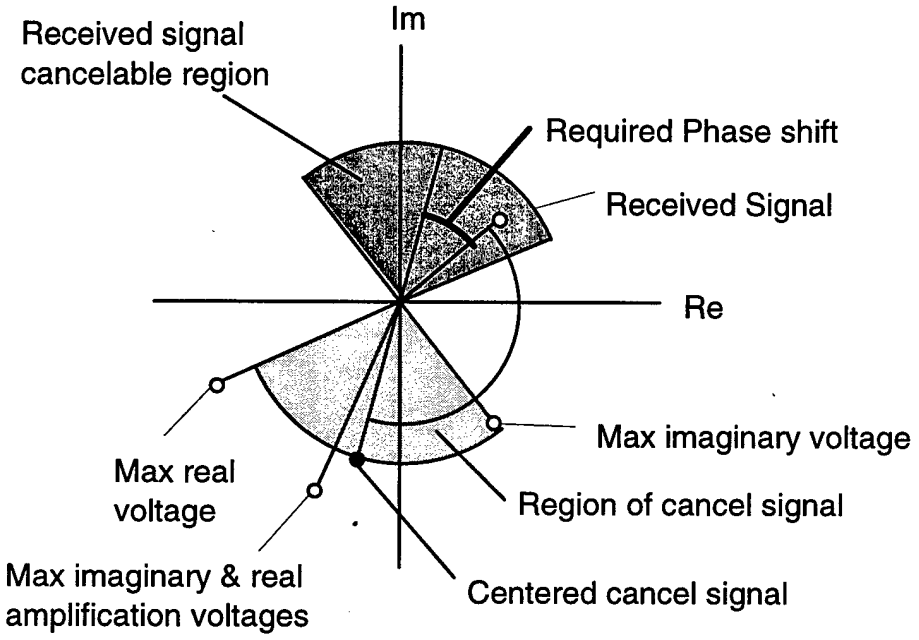
The third class of message indicates there may be something wrong with the system hardware. Either the real or imaginary amplification voltage was determined to have little or no effect on the cancel signal.

If you receive a blank message, it means the DOS cancelability check program is crashed or stalled. This can happen if you have not physically connected the computer to the hardware via the GPIB bus. In this case simply reconnect the hardware and press the RESET button.

Lets look in more detail at what happens during a signal cancelability check. The DOS program checking signal cancelability makes four separate acquisitions from the network analyzer with the following voltages.

Real Amp Voltage	Imag Amp Voltage	Description
0.0V	0.0V	Received Signal
4.0V	0.0V	Max real voltage
0.0V	4.0V	Max imaginary voltage
4.0V	4.0V	Max real/imag voltage

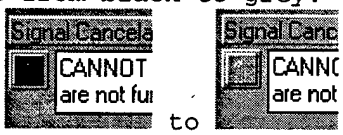
From the last three readings, the "Region of cancel signal" is computed. The magnitude for this region is taken as the minimum magnitude of the three readings. Once the "Region of cancel signal" has been established a "Centered cancel signal" is computed. The centered cancel signal and a phase width are used to characterize the "Region of cancel signal". Received signals which are 100% cancelable fall in the "Received signal cancelable region". This is characterized by a magnitude which is less than or equal to the *centered cancel signal* and a "Required Phase shift" which is less than one half the phase width of the "Region of cancel signal".



There are some cases in which you might wish to over-ride the results of the signal cancelability check. To do this you must...

1 Stop the program, using the LabVIEW stop button.

2 Click on the Cancelability control. (the indicator should change from black to grey.



3 Restart the program.

Making signals cancelable

What to do when signal cancelability comes back as "CANNOT CANCEL!". First determine if the message was generated because the received signal does not fall into a cancelable region or if there is a problem with the cancel signal hardware.

If there is a problem with cancel signal hardware, use the manual voltage controls in conjunction with the real/imaginary graph to determine if real, imaginary, or both cancel signal components are not functioning. Once determined, trace system circuitry to locate the problem.

If the received signal is not in a cancelable region, determine if the problem lies with the received signal magnitude or phase or potentially both.

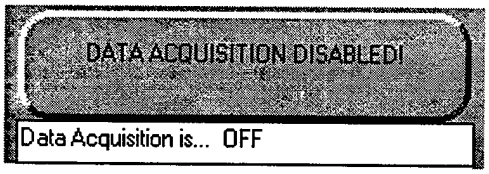
If the received signal magnitude is too large as indicated by reported ratio of received signal/cancel signal being greater than unity, insert one of the attenuators in the line connected to the transmitter fiber optic box. Alternatively you may change the geometric nulling by adjusting the TX/RX relative antenna positions.

If the received signal has the wrong phase, add one or more of the ready made cables to produce the required phase shift. You should remember that you have approximately 45 degrees of leeway associated with the required phase shift reported by the software. The reported phase shift is associated with the *centered cancel signal*. For example if the reported phase shift were 75 degrees, you could add 45 degrees of shift and obtain a cancelable signal.

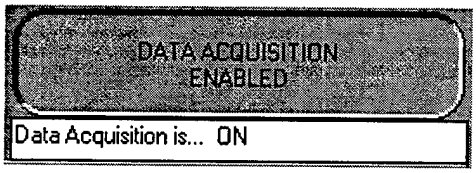
Once you have made these changes, use the RESET button to check signal cancelability once again.

Data Acquisition Control

Sampled data can be displayed in strip charts as well as stored to disk. Handling of sampled data is controlled by the DATA ACQUISITION button shown below.



The button appears as shown when data acquisition is disabled. Neither the magnitude and phase strip chart updates nor data storage will take place when data acquisition has been disabled.

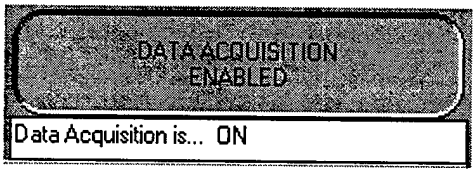


To enable data acquisition, place the mouse over the button and click the left mouse button. The button should appear as follows...

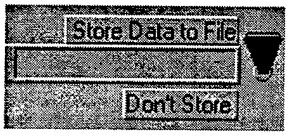
This will cause the magnitude and phase strip charts to clear and begin updating. If data storage has been allowed, acquired data will also be stored to disk.

Controlling Data Storage

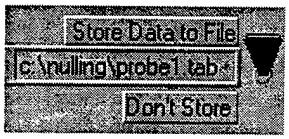
Data storage is controlled through five controls on the LabVIEW front panel interface. Data acquisition must be enabled to allow data to be stored to a file.



Once enabled, the file storage switch must be in the "Store Data to File" position. The "Don't Store" position allows you to separate valid experimental data from data collected during various set-up operations. The "Don't Store" position also lets you view the nulling process on the strip charts without saving data to a file.

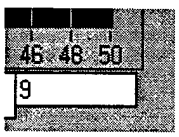


Note the user entry field to the left of the file storage switch. If this field is not blank, data will be stored to the path and file indicated.



For the example left, data will be stored in file probe1.tab located in the c:\nulling directory.

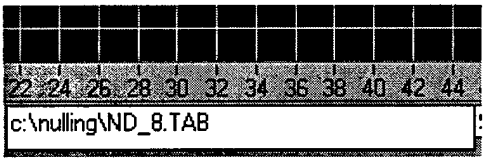
If the user specified file field is blank, then data will be stored to a default location and file name. Default files are always located in the c:\nulling directory. The default file name has the form "ND_#####.TAB", where ##### is a number generated by the program automatically. This number is tracked in the initialization file and is loaded at program start up.



The number to be used in the *next* default file name is displayed just below the lower right corner of the phase strip chart. A default file name is generated each time the data acquisition button *transitions from OFF to ON.*

You can change this number to any value (no more than 5 digits" by simply editing the box.

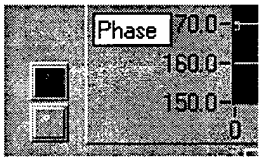
The current file being used for data storage is displayed just below the middle of the phase strip chart. This box contains either a default file name or the user specified file name if supplied.



You cannot change the display of the current data storage file *while the program is running*. While it is possible to change this value while the program is stopped, it is not recommended.

NOTE THAT IF YOU DO NOT SEE AN APPROPRIATE FILE NAME DISPLAYED IN THE ABOVE BOX OR THE MAG/PHASE STRIP CHARTS DO NOT CLEAR **DATA IS NOT BEING STORED!** Check this indicator regularly. If you do not see the display clear then cycle the data acquisition button until it does. This problem arises from a timing bug associated with servicing the data acquisition button.

Data Acquisition Indicators



Near the lower left corner of the phase strip chart are two un-marked indicator lamps. These simply exist to provide feedback associated with data acquisition state.

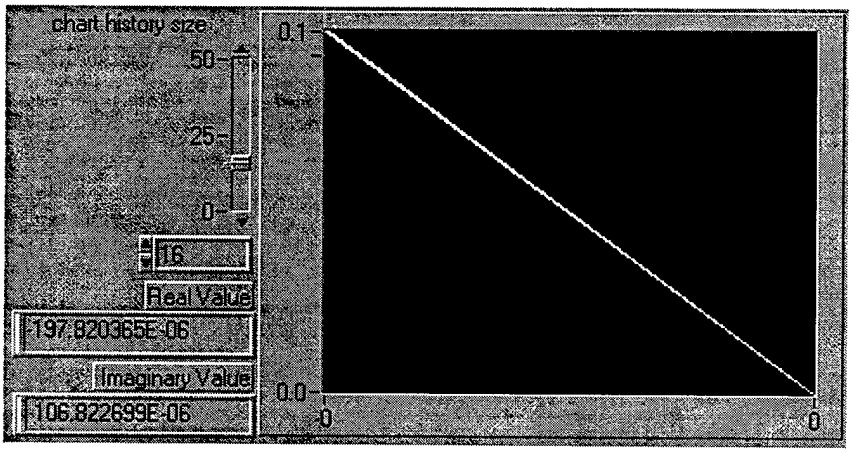
The top indicator shows that the data acquisition state is active *and has been active* for one or more iterations. The lower indicator shows that data acquisition has just *transitioned* from OFF to ON.

REAL TIME DATA DISPLAY

In this section we look at real time charting of the detector signal. The LabVIEW front panel contains three charts. The magnitude and phase strip charts, which are active only when data acquisition is on, and the real/imaginary component display chart which is active any time the program is running.

Real Time Real/Imaginary Component Display

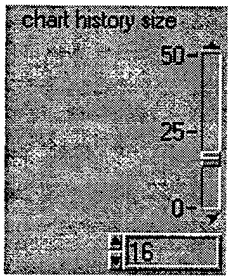
The real/imaginary display serves two main functions. It gives direct visual feed back of system operation. Second it can be used effectively to support both automatic and manual nulling operations.



The horizontal/vertical axes represent the most recent real/imaginary detector-signal *history* in units of Volts. It is important to remember that the chart is auto scaling (signals will span 6 to 7 orders of magnitude) and that each historical "point" is a line segment anchored at the origin.

Controlling History Size

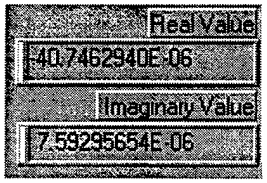
Chart history size is controlled with the elements shown left. The history size is applied to the real/imaginary graph and controls the number of data points which will be displayed on that graph. Each detector signal acquisition is represented as a line segment anchored at the origin. To each acquired real/imaginary point the value 0,0 is added by the program. Thus the number of detector signal history values are exactly half the indicated chart history size. The control has been designed so that only multiples of two can be entered.



Note that each time you adjust chart history size, the current history is flushed.

The control gives you the choice of direct entry or slide entry.

Display of the Current Real/Imaginary value in floating point



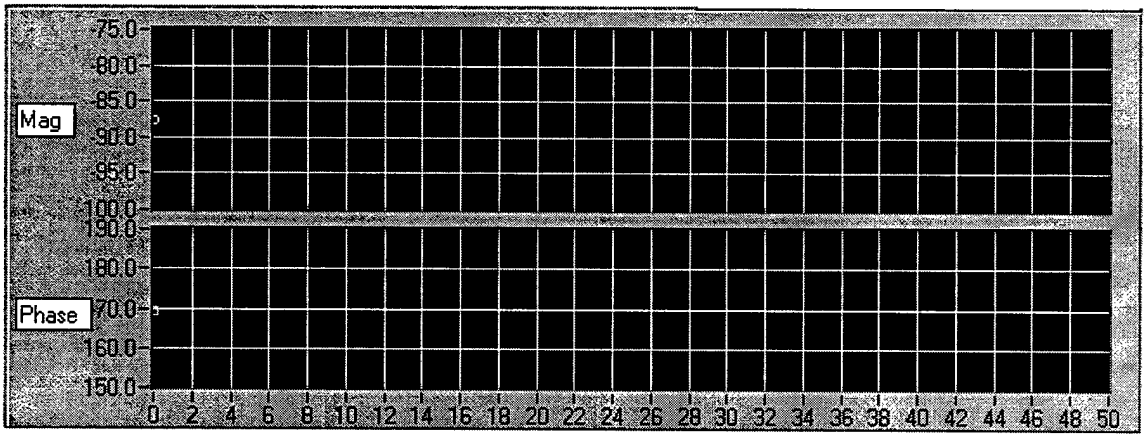
Associated with the real/imaginary chart is a floating point display of the most recent detector signal real and imaginary component values. These values are displayed in units of volts and can be found near the lower left corner of the real/imaginary chart.

This display becomes especially useful if you are attempting to manually null the received signal. You can also use this display to collect data for checking cancel signal equations, or general debugging purposes.

Magnitude & Phase strip charts

The magnitude and phase strip charts let you view the detector signal at a frequency set by the current sample rate value. Data acquisition must be ON for these charts to be updated.

The magnitude chart displays a history of detector signal magnitude in decibels of nulling (relative to a 1V magnitude). The phase chart displays a history of detector signal phase in degrees. 1024 points are stored in each history. The researcher may scroll back and forth through the histories or change many aspects of the charts through the normal LabVIEW controls. The horizontal axis for each chart is *sample number*.



Each data point in the history is displayed with a circle around it. Scrolling can be accomplished by using the scroll bar located below the lower left corner of the phase chart.

15 APPENDIX B: PROGRAMMER'S NOTES FOR NULLING SOFTWARE

8MHz Null-Signal Control and Data-Acquisition Software

DOCUMENT SCOPE

This document provides detailed information associated with nulling system software internals. The nulling system software is an integration of the Labview 4.0 W95 environment, a DOS interpretor program, and DOS batch files. This document will be focused on interpretor internals and data flow within the nulling system software. Discussion of the Labview programming environment is left to the Labview manuals and online help, both of which are excellent. Discussion of nulling system use can be found in the User's manual along with details concerning command file syntax and programming.

Please note that it is possible to program the nulling system at three levels. The first being the Labview environment, the second being the interpreted command files, and finally by modifying the C code of the DOS interpretor program. The nulling system has been designed to minimize C programming. It is possible to accomplish many proceedural tasks using the command files and/or the Labview environment. With the exception of the code found in nulling.c all C code supports a general GPIB interpretor program. Unless you are concerned with

We will also look at the math behind development of the analytical nulling equations, the kernal upon which the nulling system is based. A discussion of the tools used to develop and test these equations is also included.

A discussion of future development directions and a bug list complete the document.

DATA FLOW

In order to understand how the Labview environment, interpretor command files, and analytical nulling computations interact, we will look at the overall flow of system data in the context of system hardware and the reasearcher.

The nulling system software supports high level Researcher commands, controls system hardware, displays key operating information, and optionally logs this information to disk. Operational displays and data logging are handled entirely within the Labview environment. High level commands associated with data logging, and manual voltage control are also performed in the Labview environment. High level commands associated with analytical nulling, system resets, and received cancelability checks are performed by the DOS interpretor program via appropriate command files.

The basic data flow is discussed in the following sections.

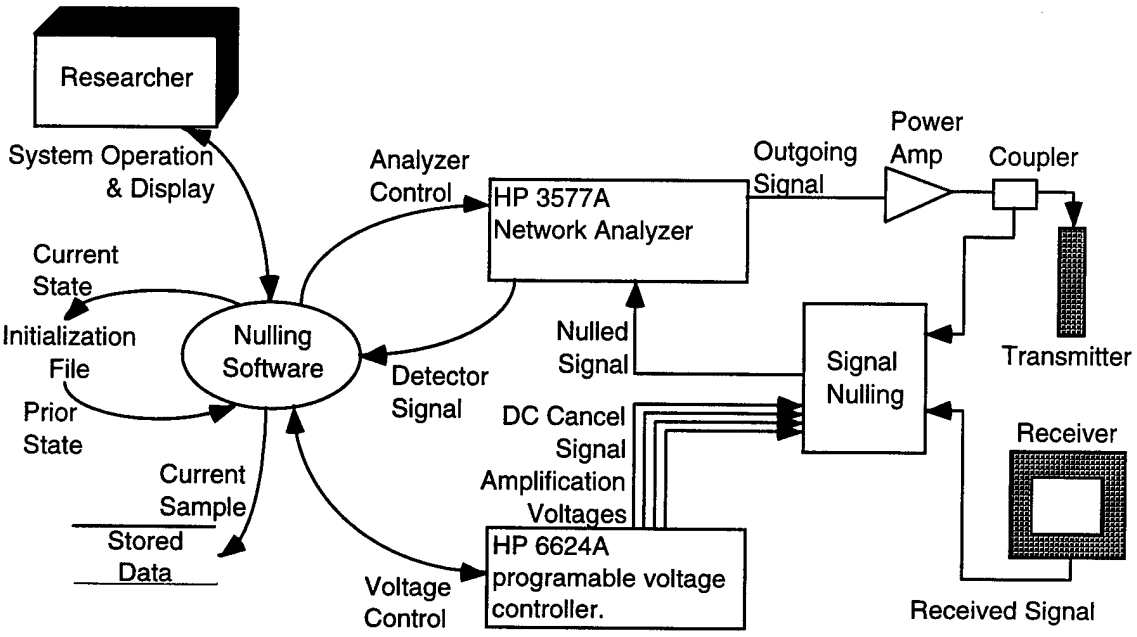
PROGRAM CONTEXT

The context diagram below shows the main components of the nulling system. The entire focus of the system is to support the Researcher in data collection and system control activities. System hardware consists of the HP 3577A network analyzer, the HP 6624A programmable voltage supply, a signal combiner box and a transmitter and receiver antenna. The HP devices can be controlled over the system's GPIB.

The hardware is designed to transmit an 8Mhz signal whose power is controllable via the network analyzer and also via manual use of several inline attenuators. This signal is then received and canceled producing a "null" at signal levels approximately 110db below native received signal levels. The physical configuration of the antennas is responsible for 40db of nulling. When operating in such a null state the system becomes a sensitive detector for any changes in the immediate physical environment. Operation, observation, and logging of the "Dectector Signal" shown in the context diagram is the purpose of the system software.

The "Researcher" drives the system through high level commands (part of the "System Operation & Display" dataflow). Commands which affect the state ("Current State") of the "Nulling Software" are recorded on disk ("Initialization File") and are used to initialize the program during start up ("Prior State"). The "Researcher" can create the "Detector Signal" via automatic programs or manual voltage adjustments. This signal is then displayed in real time at a specified sample rate. The sampled "Detector Signal" can optionally be logged to disk ("Current Sample" and "Stored Data").

Setup of the HP 3577A as well as command streams to sample the current detector signal are represented by the "Analyzer Control" data flow. The "Detector Signal" is created by manipulating phase and magnitude of the "Cancel Signal" via the HP 6624A programmable voltage supply. Command streams used to control the HP 6624A are represented by the "Voltage Control" data flow. These commands are sent when the "Researcher" makes either manual or automatic voltage adjustments.



Context Diagram

Now that we have looked at the software on an external level, lets examine the internals of the "Nulling Software" funtional group. As shown in the following diagram, the software is split between the Labview and DOS environments. The DOS environment performs automatic functions associated with "nulling" the "Received Signal" using the "Cancel Signal". The Labview environment contains the user interface.

Remember that the "Researcher"'s interface was represented by the "System Operation & Display" data flow. This has been expanded to include more details. "Researcher Commands" resulting in manual voltage adjustments, triggering of DOS based programs, or changes in data logging and display are provided by the "Researcher" through the Labview graphics control interface. "Graphical Displays" and "Current Operating Conditions" represent the feedback to the "Researcher" concerning system operation. Once per sample period, the Labview program saves its current operating state to disk via the "Current State" data flow. Upon program start up, it reads "Prior State" from disk, restoring previous program operating conditions.

Notice that the "Detector Signal" data flow is supplied to both the Labview Interface and the DOS environment (GPIB Interpretor). The Labview Interface makes use of this data flow in the real time displays while the GPIB Interpretor uses this data flow to compute automatic nulls, and assess signal cancelability.

Note that the "Voltage Control" data flow has also be expanded in this diagram. Should the Researcher make manual voltage adjustments, the new values are embeded in the HP 6624A command syntax and sent over the GPIB. The

"Manual Voltages" data flow represents these embeded commands. The Labview interface keeps track of manual voltage changes, and if made requests confirmation of the new voltages from the HP 6624A. This confirmation is reflected in the "Current Voltages" data flow.

The GPIB Interpretor portion of the software runs under DOS and is actually a single program which is involked with an appropriate command file associated with desired program operation. This command file is then interpreted. There is very little communication between the Labview and GPIB Interpretor, this being limited to "External Function Trigger" which simply starts the interpretor, and "Cancelability Check" which is a file containing the results of a signal cancelability check.

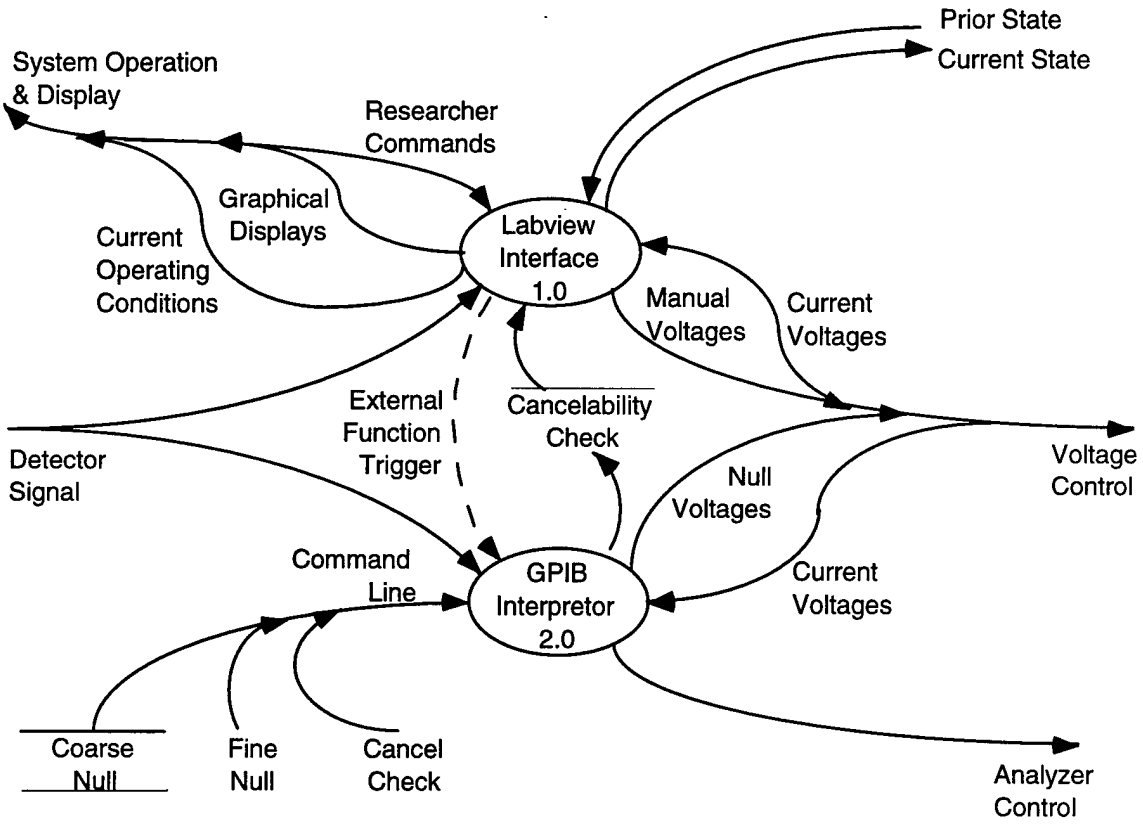


Figure 0

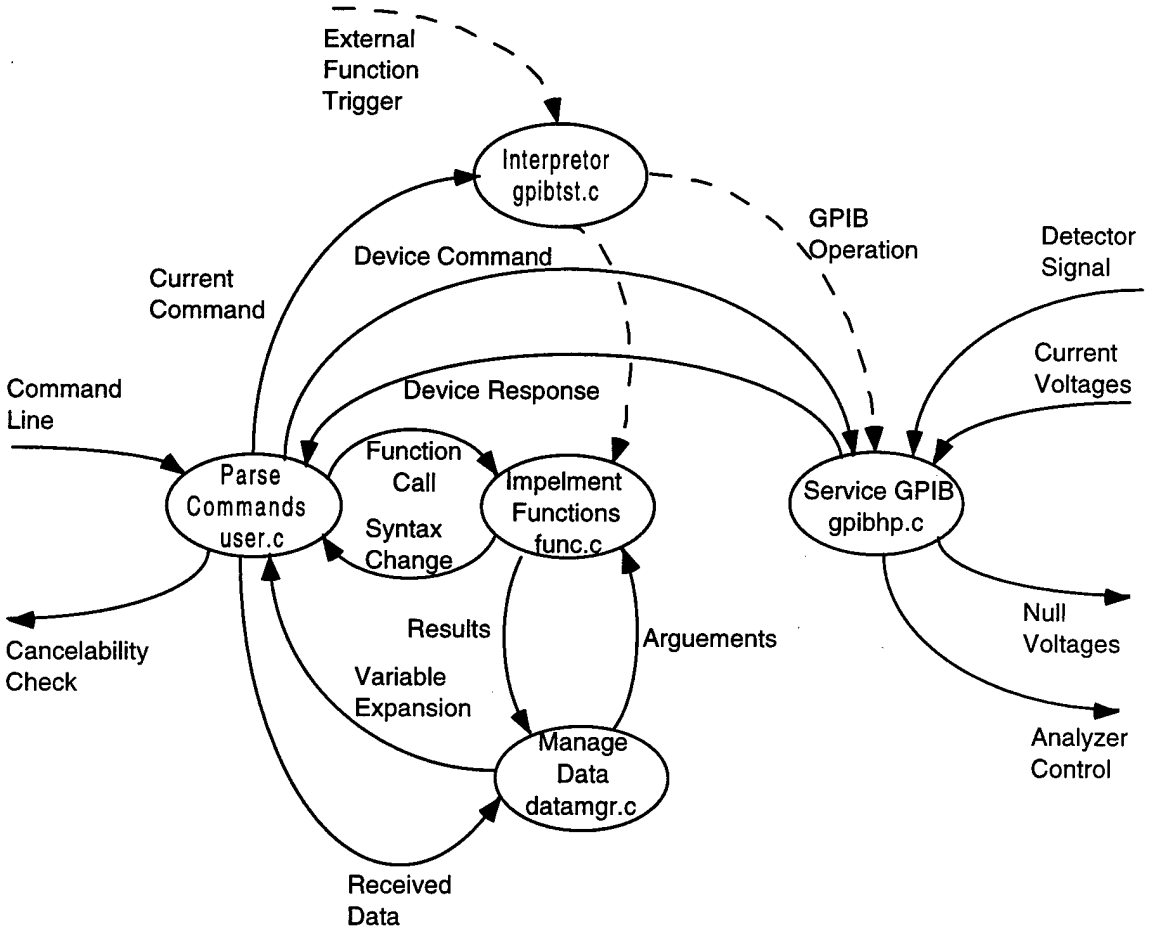
Having been involked with one of the command files ("Coarse Null" ,"Fine Null" ,"Cancel Check"), the GPIB Interpretor begins line by line processing represented by the "Command Line" data flow. In all cases the GPIB Interpretor reads the "Detector Signal" one or more times having set up the network analyzer and programmable voltage supply appropriately via the "Analyzer ontrol" and "Null Voltages" data flows. The end result of running the GPIB Interpretor is either implementation of the "Null Voltages" data flow or the output of the "Cancelability Check" file.

The details of the command files have been docuemented in the User's Manual, and will not be re-discussed here. However we will look inside the interpretor program itself to gain a better understanding of the functional groups inside the program. We will look no further into the Labview interface as the "source" code is itself diagramatic, so representing it in another diagramatic form would be redundant.

INSIDE THE INTERPRETOR PROGRAM

The data flow diagram below depicts the five main functional groups of the GPIB Interpretor software. This software has been designed to insulate the programmer from the details of the user interface and the GPIB. Unless a global improvement to the interpretor itself is required software changes to the nulling system should be *confined* to func.c (or nulling.c) *only*.

"Implement Functions" serves as a *transition* between the command file syntax and your C functions performing application specific calculations and logic tests. Note that all function arguements are taken from "Manage Data" and function results are stored via "Manage Data" Application specific functions *never* concern themselves with the user interface or direct GPIB I/O. They simply process their input arguements to produce their computed results.



Note that program operation is begun with the "External Function Trigger" which simply starts the program having routed an appropriate command file into standard input. The command file is processed a line at a time as represented

by the "Command Line" data flow. Each command line is begun with a single character command which is passed to the "Interpretor" allowing subsequent command processing to be routed to an appropriate handler. There are three handlers; "Implement Function", "Service GPIB", and "Parse Commands". Routing of command processing is all the "Interpretor" function does. "Service GPIB" sends device spefic commands and queries over the GPIB and services associated query responses. "Implement Function" provides an interface into application specific function calls (such as computing a null). Finally the "Parse Commands" function simply processes two classes of commands. The classes may be thought of as external and internal. External commands are simply passed on to be routed to another handler. Internal commands are concerned with memory, disk, and screen access.

In the following sections, we will look at each of the five main functional groups in more detail. The "Implement Functions" group will be covered in complete detail as this is the immediate area you should focus on to develop or expand the nulling application.

The User Interface (Parse Commands, user.c)

The user interface software group is built around a traditional command line interface. The user interface both parses and operates on command line contents. *Interpretation* of the command line is left to a higher level routine. Command line operations include results display, variable management, and special GPIB preparation. First, lets look at how command lines are parsed by the user interface.

In order to parse commands, there must be an agreed upon list of *delimiters* used to separate command elements from one another (if you are to have a free format command line). The user interface has a *default* list of delimiters....

\t , Tab, space, and comma are treated as *white space* by the user interface.

Note that this delimiter set is not static but can change dynamically. This is necessary to support various contexts associated with certain commands. For example, Implement Function will cause the user interface to add the following two characters '()' to the delimiter list. Likewise during processing of a GPIB write, the GPIB interface will suspend parsing of commands for any quoted data. There are two routines which facilitate changing the delimiter list.

USER_Parse

Gets you a pointer to the current delimiter list and

USER_AddParse

Adds one or more delimters to the current list. Finally

USER_SetParse

sets the delimiter list to the supplied list.

The user interface assumes the following structure of command lines...

[token1 [token2 ... [tokenN] ...][; [comments]]

where [] indicate optional data.

Note that there are really only two assumptions of structure. One is allowance for comment lines. The second is that tokens will be separated by one or more delimiters. The fact that the GPIB Interpretor software

and the DOS portion of the nulling system is built on single character commands is a function of the *higher level* interpretor software *not* the user interface.

To complete our discussion of command line parsing, you should note that higher levels of code built on the user interface *never* see comments. These are removed from the data stream before it is passed upward.

The first token on the command line is available by calling...

USER_Command

Subsequent tokens are available by calling...

USER_NextCmd

A new command line is loaded by calling...

USER_GetLine

As metioned earlier, the user interface can also operate on command line contents. This happens because a higher level routine calls the user interface *back* to ask for further processing. Currently the user interface preprocesses GPIB[1] output, displays or stores GPIB input, and handles disk I/O.

Lets look at GPIB output preprocessing. For some GPIB devices, there is an intersection between the Interperator's command syntax and that of the GPIB device. Also some devices support direct transfer of binary data to the device (configuration, example waveforms or traces etc). Finally it is often convenient to use computed variable values in a command to a GPIB device (sending null voltages for example). The

USER_GPIBCmd

routine was designed to support the above conditions. Lets look at the syntax intersection. In the nulling system, a device specific command is sent to the device as follows...

< "device_command"

The quotes surronding device_command are removed prior to sending device_command to the GPIB device. These quotes are a part of the

1 The user interface never knows that the data is associated with GPIB access. It simply treats the data as a stream of bytes. In the nulling system application however, this stream is always associated with GPIB input/output.

Interpretor syntax. Note that device_command can contain spaces *and other quotes.* In order to support embeded quotes, the ' character is used *on the command line* but is expanded to ". For example...

< "MESSAGE 1,4 'Hello Device'"

This might tell a GPIB device to place a message on its screen beginning at row 1 column 4. The message being "Hello Device". The actual data sent to the device would be...

MESSAGE 1,4 "Hello Device"

Other GPIB command pre-processing follows this same philosophy of substitution. To send binary data to a device it must be contained in a local variable. The user interface *expands* local variables in line prior to sending data to the device. For example to set a voltage in the nulling system...

< "VSET 1, <VRC>"

Where <VRC> indicates that the local variable VRC should be expanded in line. The data sent to the device would be...

VSET 1, 3.25

If VRC was a string variable containing the value 3.25. The user interface strips the expansion syntax, <>, from the outgoing data stream and replaces the variable name with its data. As you might expect the variable's data is obtained from the data manager. There are other expansion characters which control the format of the expanded data. Binary expansions, <>, are simply a direct substitution of the variable's contents in the outgoing data stream. ASCII expansions, {}, format the variable's data using a sprintf and a format specifier determined by the variable's data type. An IEEE binary expansion is available by using [] characters. This follows the same rules as a normal binary expansion, except a standard IEEE header is *prepended* to the expanded data.

The user interface is also called upon to display data on screen. Such display is triggered by a higher level routine. The routines...

USER_Receive

and

USER_DisplayVar

manage display of data on the user's screen. USER_Receive is responsible for handling input from GPIB devices. It either displays received data on the user's screen or stores it via the data manager. USER_DisplayVar displays the contents of data manager variables on the screen.

Standard Input/Output

The user interface is designed to read all *command* data from standard input. Note that the '*' command will cause standard input to be re-directed on-the-fly, however the rest of the user interface is unaware this has happened.

Reading data from standard input allows us to send data to the interpretor at the DOS level via the pipe command '|'.

Likewise the program writes all output to standard output unless otherwise directed by a specific command. This means that program results can be collected into a file.

Starting the program as below...

GPIBTST<CR>

will take command input from the keyboard and write results to the screen.

The following invokation...

TYPE MYCMNDS.TXT|GPIBTST<CR>

willl take command input from the text file mycmnds.txt and write results to the screen.

Finally...

TYPE MYCMNDS.TXT|GPIBTST>MYRESLT.OUT<CR>

will take comamnd input from the file mycmnds.txt and write results to the file myreslt.out.

Disk I/O

The user interface also manages disk I/O, which is perfomed as part of a more general assignment operation. In conjunction with the data

manager, the user interface can store or retreive binary images of variables only. It does currently support formatted record I/O.

The Data Manager (Manage Data, datamgr.c)

The data manager simply manages access to a block of memory used to store and retreive application data. At any given time, this block of memory is divided into used and unused space. The data manager makes sure that the block of memory is not fragmented by keeping the used and unused portions contiguous.

All data is simply stored as a set of bytes. Each set is associated with a name, data type, and length (in terms of data type). We can think of this association as a "variable". The data manager allows runtime creation and destruction of vairables. Operations, other than creation or destruction, on variables are not *specifically* supported by the data manager. Rather, its job is to return a *general* pointer to the variable which the caller can use to read and write values as desired.

The runtime access to variables is the reason for the data manager's existance. This runtime access is a core capability of an interpretor which the data manager was designed to support.

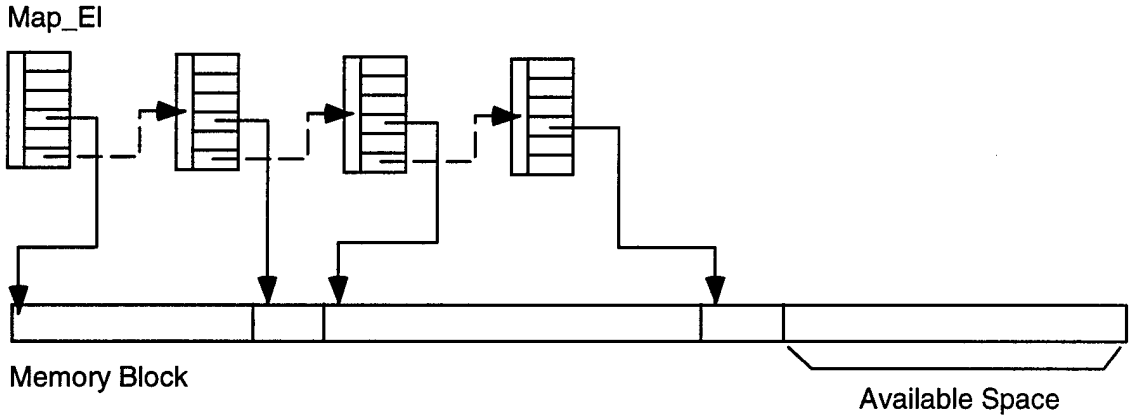
While structures can be stored/retreived by the data manager, it does not have the ability to *interpret* such structures at runtime. In otherwords at *runtime (ie accessing the data via the interpretor)*, you can not ask the data manager to store/retreive an element of a structure, but must instead access the structure as a whole. Within any C code you may write making use of data manager variables, you can of course access the strucutre through the normal structure syntax.

The key data structure of the data manager is given below. szName contains the name of the variable. nSize contains the *total* size in bytes occupied by the variable. nElSize contains the fundamental data type size. cType contains the single character mnemonic for the data type. nOffset is an integer byte offset into the memory block at which the variable begins. *pNext points to the next variable listed with the data manager.

```
struct _mapel {

        char            szName[ 32 ];
        int             nSize;
        int             nElSize;
        int             nOffSet;
        char            cType;
        struct _mapel   *pNext;
};
```

Graphically...



New variables are *always* added to the end of the Map_El list, and the first variable in Map_El list always begins at offset 0 in the memory block. Deleted variables are removed from the Map_El list in the normal manner of a foward linked list. The data associated with deleted variables is immediately lost as the data manager compresses the unused space.

Note that the data manager performs no initialization on new variables.

Data Types Supported

The data manager supports the common C data types. However it is not limited to these data types. You can define your own data types (by recompiling the data manager). The data types currently supported by the data manager are...

s	A string of an arbitrary byte length. Default element length = 1.
c	A single byte character
n	Machine storage for single precision integer (typically 2 bytes)
l	Machine storage to a double integer (typically 4 bytes)
f	Machine storage for a single precision floating point (typ. 4 bytes)
d	Machine storage for a double precision floating point (typ. 8 bytes)
p	User defined data type corresponding to a power spectrum

You can store an arbitrary length set of any data type (an array) subject to available memory.

Adding your own data type

Currently you must make a code change to the data manager in order to add your own data types. Basically this requires changing a few simple data structures in the data manager code. Your data type will require two peices of information. A single character used to represent your data type, and a base size (in bytes) for your data type.

Lets look at how support for the power density spectrum was added.

First "_CUR_DTYPES" was modified to include the 'p' data type which was the character decided on to represent the power spectrum data type.

```
#define _CUR_DTYPES      "vcsilfdp"
```

Next the DATA_Size function was modified to compute the size of the new data type. This involves adding a single line to the case statement such as...

```
case  'p':     return sizeof( _PSD );
```

Note that we have chosen to include the psd.h file in the data manager so that the specific definition of _PSD need not concern the data manager code. Alternatively we could have hard-coded the size of the new data type but this would most likely lead to inconsistencies down the line.

This is basically all there is to adding support for your data types to the data manager. There are however three subtleties which you should note. First a new data type of 'v' would confuse the data manager as this is the default data type (stands for 'void') used by the manager in the event an unknown type is passed. Second, your data types should have lower case/upper case directly related by bit 5 of the ASCII character representation of your data type. As long as you stick to lower case letters this is not a problem. However data types of '1' or '?' will lead to confusing use at the user interface level.

The last sublty is that you need not explicitly specify a size for your data type. A line such as...

```
case  'p':     return 1;
```

could be added to the DATA_Size routine. This basically means that the data type will default to an *element* size of 1 byte. It is then up to the calling C code to define the fundamental data type size at run-time via the DATA_StoreElSize routine. This approach is used in handling the string data type and allows for arrays of strings of any fixed arbitrary length.

Single Variables

The data manager views single variables simply as an array of one element. Thus nSize = nElSize in the Map_El structure discussed earlier.

Vectors of a given type

Vectors of a given data type are the only collection of data elements currently supported by the data manager. Eventually the data manager may be extended to support structures at run-time thus allowing collections of different data types.

At run-time (ie during interpretation of command files) vectors must be accessed, displayed, writen/read from files as a whole. The data manager makes no provision for accessing a sub-element of a vector. The data manager simply returns an address in the memory block corresponding to the begining of the variable's data. It knows the corresponding number of bytes from that address which contain *all* the variable's data.

Note that this is not a limitation within any C routines which you may write making use of data manager variable pointers. You can of course treat the address returned by data manager any way you like using cast and assignment syntax.

Memory Management

Currently the data manager simply allocates a fixed amount of space for holding *variable data*. It will support an arbitrary length list of variable names (subject to physical malloc limitations). This was done to guarantee that command files would always run consistently without having to worry about physical memory availability.

Ideally, the data manager should allocate memory as needed however this implies a whole raft of checks and house keeping. In addition it would not guarantee that adjacent variables are stored next to one another in physical memory, which property will allow the data manager to easily be extended to support structures on an element by element basis.

The amout of space initially allocated by the data manager is controlled by the _SND_REC_SIZE define in datamgr.c.

What to do if the Data Manager needs more variable space

If you need more space for your variables, simply modify the _SND_REC_SIZE value at the top of datamgr.c and re-compile.

The GPIB Interface (Service GPIB, gpibhp.c)

This block of routines has survived unchanged over three applications for a two year period. Its basic purpose is to allow high level access to devices on the GPIB. The core calls are...

GPIB_Init	Initialize the GPIB controller card in the PC.
GPIB_OpenDev	Make a GPIB device of a given name the focus of input/output operations.
GPIB_Send	Send a stream of bytes to the current device.
GPIB_Receive	Receive a stream of bytes from the current device.

The key drawback to the set of routines is that if you attempt a read on a device when no response is forth comming, the GPIB_Receive command will not time out. However in the normal course of Query/Response pairs this condition is never encountered.

GPIB errors are currently written to standard output, this could also be improved.

The Function Call Interface (Implement Functions, func.c)

The function call interface supported by the GPIB interpretor software is *managed* in func.c. The actual computations associated with your functions should be contained in a separate C source file. Application specific function prototypes should be included in func.c.

This document section will provide details on adding your own functions to func.c so that they will be available from the command line or command files processed by the GPIB interpretor program. Curently all application specific functions are contained in nulling.c for the nulling software.

Adding your own functions

The function call interface provides a *transisition* from the interpreted environment to the normal C environment. This involves the following steps. First the called function is tested for existance. Second the calling syntax of the function is checked in terms of number and possibly data type of arguements. Third the function arguements are collected from the data manager and preprocessed if necessary. Next the application specific function call is made. Finally results are post-processed and stored in the data manager.

You must implement the above steps in order to add support for a function call in the interpretor environment. Basically this is done by creating a trasition function in func.c as well as coding of your application specific function (contained in a separate file).

To add your own transition function in the interpretor environment create a prototype for the transition function in func.c. To add support for an application specific function MyFunc you might add ChkMyFunc to func.c as follows...

```
void  ChkMyFunc       ( int nArgs ,char **apszArguementList );
```

Note that all transition functions have the same arguement list.

Next add a name for the function to be used on the command line. Modify pszFuncList in FUNC_Call... for example...

```
char  *pszFuncList
//          .... ....1.... ....2.... ....3        9.... ....0.... ....1.... ....2
        = "WAVEFORM |PSD       STACK     ......   GETCANCEL MYFUNC              ";
```

Make sure to leave an extra 10 blank spaces following the 10 characters allocated for your function name. Also note that function names *always*

begin on a multiple of 11 offset into pszFuncList. The user will now be able to access your function by typing...

MYFUNC(arg1 arg2 ...)

on the command line.

Now add the name of the *transition* function call to be *mapped* to MYFUNC. This is done by adding the name you prototyped in the first step to the funcTable structure as shown below...

```
static void          (*funcTable[])() = {  FUNC_NotFound
o
o
                                          ,ChkMyFunc
                                        };
```

Next write the internals for the *transition* function ChkMyFunc. Lets look at how this was done for NULLSetParms (the function to parse and store current network analyzer operating parameters).

Remember that the Interpretor has passed us an arguement list in **szArgs. This arguement list is an array of parsed variable NAMES currently in use by the data manager.

```
void NULLSetParms( int nArgs ,char **szArgs )
{
   int       nIdx       ,nErr;
// ---------------------------
```

Optional support for the debugging flag used at compile time. Print the function call name and the passed arguement list.

```
   #ifdef _DEBUG
      printf( "SETPARMS( " );

      for( nIdx = 0;  nIdx < nArgs;  nIdx++ )  {

         printf(  "%s " ,szArgs[ nIdx ]  );
         if( nIdx < (nArgs-1) )  printf( "," );
         else                    printf( " )\n" );
      }
   #endif
```

Now check to see that the number of passed arguements is correct.

```
   if(   nArgs != 2  )  {

      printf( "Incorrect # arguements to SetParms\n" );
      printf( "... SetParms( ParmString )\n" );
      return;
   }
```

This error should never really happen, but keeps us from calling application specific function with bogus arguements. Note if you want to allow the user to supply constants to your function call (such as 5.0) the this is where you would process them.

```
   for( nErr = 0 ,nIdx = nArgs;  nIdx--;  nErr |= !szArgs[nIdx] );

   if(  nErr  )  {
      printf( "SetParms arguements not parsed correctly ?!\n" );
      return;
   }
```

Any specific pre-processing of arguements would go here... for example unit conversions, data type conversions etc.

Now pass pointers to two character strings which NULLParse has been programmed to receive. Check nulling.c for details.

```
NULLParse(  DATA_Retreive( szArgs[0] )
          ,DATA_Retreive( szArgs[1] )
         );
```

Post processing of results would go here... ie storage into data base variables if the type of the data base result variable differed from the application function results type. You would also need some post processing if your application specific function returned a value such as an int, which you then wanted to store in a data base results variable.

```
}
```

Finally write the computations associated with your application function. Note that your application function might already exist in some library or source file which you have previously developed.

Passing Arguements & Returning results

Passing of arguement data base variable names to your transition function is taken care of by the FUNC_Args call. Your transition function will receive an array of names ordered as follows...

MYFUNC(arg1 arg2 arg3)

gives...

arg1
arg2
arg3

as the list of names passed to your transition function. So arguements are listed from left to right.

You must handle arguement data type *transformations* in the transition function. For example, lets say that you have an existing function to take the square root of a number in a file SQRT.C which takes a single single precision floating point arguement...

float SQRT(float);

However you want the interpretor to support SQRT(string1) because the GPIB device you are communicating with happens to represent all

numbers as ASCII strings, and you decide that this is the most likely data form for function arguements. To keep the arguement processing to a minimum, you decide that SQRT(string1) will takes its input from and return its result to the same variable. Remebering that the expected type of the data base variable is string, we would write the code fragment below (after performing the template operations in the previous section)...

```
psz = DATA_Retreive( szArgs[0] );
sprintf(  psz ,"%f", SQRT( atof(psz) )  );
```

szArgs[0] contains the name of the arguement passed to us on the interpretor command line. We ask the data manager to get us the *address* of that arguement's variable which is of type *string* (hence the call to atof). After converting this arguement to a floating point value, we pass it by value to the SQRT function we are *transitioning* to. The result is placed back into the same arguement variable and reformatted to string via the sprintf call.

Lets say we didn't want to stomp on the value of the input arguement. The syntax we wish to support at the interpretor level is...

SQRT(arg1:string result1:string)

which we would involk as...

SQRT(inputVal resultVal)

where inputVal and resultVal are two local variables of type string. Our code would then look something like...

```
pszInput  = DATA_Retreive( szArgs[0] );
pszOutput = DATA_Retreive( szArgs[1] );
sprintf(  pszOutput ,"%f", SQRT( atof(pszInput) )  );
```

What if we wanted to support arguements of any data type at the interpretor level. Our calling syntax changes not the least bit but our code gets a little more complex and might look like...

```
pvIn  = DATA_Retreive( szArgs[0] );
pvOut = DATA_Retreive( szArgs[1] );

switch( DATA_Type(szArgs[0]) )  {

   case 's':  fIn = atof( (char *)pvIn );    break;
   case 'd':  fIn = *((double *) pvIn);      break;
   case 'f':  fIn = *((float *)  pvIn);      break;
   case 'n':  fIn = *((int *)    pvIn);      break;
   case 'l':  fIn = *((long *)   pvIn);      break;
   default:   fIn = 1.0;
}


fOut = SQRT( fIn );
```

```
switch( DATA_Type(szArgs[1] )  {

   case 's':  sprintf( (char *)pvOut ,"%f" ,fOut );     break;
   case 'd':  *((double *) pvOut) = fOut;               break;
   case 'f':  *((float *)  pvOut) = fOut;               break;
   case 'n':  *((int *)    pvOut) = fOut;               break;
   case 'l':  *((long *)   pvOut) = fOut;               break;
   default:   fOut = 1.0;
}
```

Note that "pv" indicates a C variable type of "pointer to void". Of course if you had more arguements you would have to make loops out of the two switch statements. Now we have supported SQRT at the interpretor level in almost an indestructable manner. This example has served to point out one drawback of the data manager. The data manager should HIDE data type from callers. If we were to extend the data manager, we might be able to write code such as...

```
DATA_GetFormatted( szArgs[0] ,"float" ,&fIn );

fOut = SQRT( fIn );

DATA_PutFormatted( &fOut ,"float" ,szArgs[1] );
```

Which is a whole lot simpler from the caller's point of view. However as long as we are concerned with standard data types, such an extension is slightly less efficient and doesn't buy us much. If however the data manager begins to support a large number of custom data types, such an extension would be very useful.

NULLING SYSTEM SOFTWARE

This document section discusses code which is specific to the nulling application. It contains a review of the functions contained in nulling.c which are used in support of the nulling system.

GENERAL OVERVIEW

C functions specific to the nulling application are used primarily to calculate nulling voltages. There are also functions to check received signal cancelability, and store network analyzer operating conditions[2]; these support the analytical nulling calculation.

Because the GPIB interpretor program does not currently support a syntax allowing looping, logic tests, or computation, system operations requiring these are coded into a C routine. The resulting routine is then integrated into the interpretor via the function call interface described earlier.

First, lets look at the basic equations used to compute nulling amplification voltages.

---

[2] The network analyzer returns a whole list of operating parameters as a giant string. These parameters are only available in this form from the network analyzer. The interpretor's command syntax is not yet sophiticated enough to support arbitrary parsing of input data streams. The quickest solution was to write a routine to do this. The ideal solution however is to extent the interpretor's command syntax.

ANALYTICAL NULLING EQUATIONS

The intent of the analytical nulling equations is to model the performance of the VCA circuits in the system hardware. There are two such circuits in the system. Given such a model, the object is to *predict* the *cancel signal* real and imaginary components as a function of the four system amplification voltages. The condition for a null is then...

$$\mathrm{Re}_m = -\mathrm{Re}_c(v1, v2, v3, v4)$$

$$\mathrm{Im}_m = -\mathrm{Im}_c(v1, v2, v3, v4)$$

Where the c subscript indicates cancel signal, and the m subscript indicates the current received signal. Re, and Im indicate the real and imaginary components of the respective signals. The four amplification voltages are indicated as v1, v2, v3, and v4.

FUNCTIONAL FORMS

The functional forms for each VCA circuit were based on Burr-Brown data for the VCA610, allowance for system cross-talk[3], and network analyzer output levels. These functional forms are given below...

$$\mathrm{Re}(n) = r_5 \times 10^{\left[r_4\left[v_4(n)+r_2 v_2(n)\right]\right]} + r_6 \times 10^{\left[r_3\left[v_3(n)+r_1 v_1(n)\right]\right]} + C + D(t)$$

$$\mathrm{Im}(n) = i_5 \times 10^{\left[i_4\left[v_4(n)+i_2 v_2(n)\right]\right]} + i_6 \times 10^{\left[i_3\left[v_3(n)+i_1 v_1(n)\right]\right]} + C + D(t)$$

where

$\mathrm{Re}(n)$	The real component of the cancellation signal in units of mV.
$\mathrm{Im}(n)$	The imaginary component of the cancellation signal in units of mV.
C	A potential constant offset voltage in mV.
$D(t)$	Drift in the signal. This is assumed to be slow enough that we can treat it as constant.
$r_1, r_2, \ldots$	Constants to be determined.
$i_1, i_2, \ldots$	Constants to be determined.
$v_1(n)$	The imaginary fine adjustment *amplification voltage* in Volts.
$v_2(n)$	The real fine adjustment.

3 Which as we will see actually modeling input cancel signal *orientation*.

$v_3(n)$	The imaginary coarse adjustment *amplification voltage.*
$v_4(n)$	The real coarse adjustment.

The values of the empirical coefficients are:

r_1	-0.01035
r_2	0.01
r_3	0.8105
r_4	0.9432
r_5	$0.03082 \times 10^{-\frac{A}{20}}$, A is the current output signal amplitude in dbm
r_6	$0.003293 \times 10^{-\frac{A}{20}}$

i_1	0.01
i_2	0.0661
i_3	0.9512
i_4	0.8796
i_5	$-0.001603 \times 10^{-\frac{A}{20}}$
i_6	$0.03051 \times 10^{-\frac{A}{20}}$

Coefficients associated with the functional forms were developed from empirical measurements (RE_IM2.XLS). The resulting equations were programmed into MathCAD (NULSOLVE.MCD), which was then used to predict the four voltages required to produce a "null". The resulting predictions were then implemented on the HP6624A. Three iterations of this procedure produced nulls of up to 100db.

DETERMINING REQUIRED COEFFICIENTS

Coefficients where determined by re-arranging the predictive equations so that empirical data would plot as constant. It would then be possible to obtain statistical information about each constant. Such statistical values could then be used to assess the validity of the proposed functional form. The rearranged equations are recorded here to allow for future testing of the current functional form validity.

$$r_4 = \frac{\log_{10}\Delta\mathrm{Re}(n_1) - \log_{10}\Delta\mathrm{Re}(n_2)}{v_4(n_1-1) - v_4(n_2-1)}$$

where: $\Delta\mathrm{Re}(n) = \mathrm{Re}(n) - \mathrm{Re}(n-1)$
note: v_1, v_2, v_3 set at 0.0V during data collection.

$$r_5 = \frac{\Delta\mathrm{Re}(n)}{10^{r_4 v_4(n)} - 10^{r_4 v_4(n-1)}}$$

This can be computed from same data set taken to determine r_4.

r_3 and r_6 can be arrived at in the same manner by collecting a new data set with v_1, v_2, v_4 to 0.0V.

We are left with the computation of r_1, r_2. Holding $v_1 = v_3 = 0V$ and $v_4 = const \neq 0$ and $v_2(n-1) = 0.0V$, we can perform a Taylor series expansion yielding

$$r_2 = \frac{\delta\Delta\mathrm{Re}(n)}{r_4 r_5 \ln(10) \times 10^{r_4 v_4} \times \delta v_2(n)} \approx \frac{\Delta\Delta\mathrm{Re}(n)}{r_4 r_5 \ln(10) \times 10^{r_4 v_4} \times \Delta v_2(n)}$$

likewise...

$$r_1 = r_1 = \frac{\Delta\Delta\mathrm{Re}(n)}{r_3 r_6 \ln(10) \times 10^{r_3 v_3} \times \Delta v_1(n)}$$

with $v_2 = v_4 = 0V$ and $v_3 = const \neq 0$ and $v_1(n-1) = 0.0V$.

Note that it is very important that $v_3 = const \neq 0$ as you must get a signal which is above the noise floor. Suggested values for *const* are 3.0 to 3.5V. You want to be above the noise floor but below any possibility of magnitude "roll off" caused by the VCA's.

Having determined the values for the above coefficients from judicious empirical measurements, we are ready to apply the equations to predict a new null given an initial value for the programmed voltages and the measured real and imaginary components of the combined signal (received + cancel).

$$\Delta \mathrm{Re}(n) = r_5 10^{r_4 v_r(n-1)} (10^{r_4 \Delta v_r(n)} - 1) + r_6 10^{r_3 v_i(n-1)} (10^{r_3 \Delta v_i(n)} - 1)$$

$$-\Delta \mathrm{Re}_{measured} = E1(10^{r_4 \Delta v_r(n)} - 1) + E2(10^{r_3 \Delta v_i(n)} - 1)$$

$$\Delta \mathrm{Im}(n) = i_5 10^{i_4 v_r(n-1)} (10^{i_4 \Delta v_r(n)} - 1) + i_6 10^{i_3 v_i(n-1)} (10^{i_3 \Delta v_i(n)} - 1)$$

$$-\Delta \mathrm{Im}_{measured} = E3(10^{i_4 \Delta v_r(n)} - 1) + E4(10^{i_3 \Delta v_i(n)} - 1)$$

where $\Delta v_r, \Delta v_i$ are the changes in the overall real and imaginary amplification voltages required to counter act the current measured value of the combined signal's real component.

Note that...

$$\Delta v_r = \Delta v_4 + r_2 \Delta v_2$$
$$\Delta v_i = \Delta v_3 + r_1 \Delta v_1$$

NOTE WELL: For arbitrary cancel signal orientations, v_1, v_2, v_3 etc can potentially change their chief direction of amplification. For some orientations, V4:V2 will provide most of their amplification *in the imaginary direction.*

Solving and substituting for Δv_i gives...

$$F(\Delta v_r) = \Delta \mathrm{Re}_{measured} + E1(10^{r_4 \Delta v_r} - 1) + E2\left(\left(\frac{\Delta \mathrm{Im}_{measured} - E3(10^{i_4 \Delta v_r} - 1)}{E4} + 1\right)^{\frac{r_3}{i_3}} - 1\right)$$

which is equal to zero at a solution value for Δv_r. This equation can be solved quite nicely via binary search. Once we have a value for Δv_r, we can directly solve for Δv_i as...

$$\Delta v_i = \frac{1}{i_3}\log_{10}\left(\frac{\Delta \mathrm{Im}_{measured} - E3(10^{i_4 \Delta v_r} - 1)}{E4} + 1\right)$$

Handy DOS support for data collection

There are several DOS batch files which can be used to collect data for the EXCEL spreadsheet designed to calcualte analytical nulling equation coefficients. These are...

n.bat
v.bat
volts.bat

The EXCEL spreadsheet for calculating coefficients

The current EXCEL spreadsheet for determining analytical nulling coefficients is RE_IM2.XLS. The appropriate cells are labeled with corresponding names used in #define statements in the NULLING.H file. Having determined a set of coefficients using the spreadsheet, copy the values from the spreadsheet to the NULLING.H file and recompile the program.

TESTING DEVELOPED EQUATIONS

The analytical nulling equations were first entered into a MathCADD worksheet which along with the supporting DOS macros allowed the equations to be tested "on-line". It is suggested that any new nulling equations be tested in a similar manner.

The MathCAD worksheet & Supporting DOS macros

Performing a "manual" nulling using MathCAD & DOS

IMPELEMENTING THE EQUATIONS

Command file function calls

BUILDING THE GPIB INTERPRETOR CODE

LIST OF FILES

The following files are needed to build the interpretor which supports the nulling system software.

gpibtst.c	Top level source code for the interpretor.
user.c	Code for a general command line user interface.
user.h	
datamgr.c	Code for a general data manager
datamgr.h	
func.c	Code to support function calls in the interpretor
func.h	
gpibhp.c	Source for accessing the GPIB at a high level
gpibhp.h	
gpibdefs.h	
ourmath.c	Some custom math routines
ourmath.h	
nulling.c	Functions specific to the nulling system.
nulling.h	

Note that the higer level *proceedures* of the nulling system are contained in command files processed by the interpretor not in the C source files listed above.

REQUIRED LIBRARIES

mcib.lib

USING THE GPIB INTERPRETER PROGRAM

The GPIB interpreter program is a DOS program which can be used to access and control devices on a GPIB bus. Program input/output can be redirected from the DOS command line facilitating batch operation.

A simple single command character syntax is interpreted by the program. This means you can focus on the data you need to send, receive, store, and display.

The program may be run interactively from the keyboard by typing:

GPIBTST<CR>

Alternatively, you may use the program in batch mode as follows:

TYPE MY_CMNDS.TXT|GPIBTST>MY_RESLT.OUT

Which interprets the commands in MY_CMNDS.TXT and places the results in MY_RESLT.OUT. Because of this capability, you can use GPIBTST.EXE in conjunction with DOS batch file processing to quickly give yourself application specific commands at the DOS prompt.

An overview of the command syntax is provided in the next section. This is followed by a detailed discussion of each command.

Command Syntax

The single character commands interpreted by GPIBTST are listed below along with any arguments. Optional arguments are enclosed in "[]", while choices are indicated as choice1|choice2|choice3 etc.

! [card_name]

Opens and initializes a GPIB controller card. Cards may be named such as GPIB0 etc. The set of names is controlled by the IBCONFIG program. This is typically the first command given. If no name is supplied the ! command defaults to GPIB0 which is the default name for the first GPIB controller card under the National Instruments configuration.

@ dev_name

Accesses a GPIB device for subsequent communication commands. Device names map to whatever was configured in the

IBCONFIG.EXE program. Typically default device names are DEV0, DEV11, etc.

< "aaaa[<var_name>|{var_name}|[var_name]]aaaa"

Send data to the currently active GPIB device. The literal string must be always be surrounded in double quotes.

For example a simple literal string would look like

< "FM1;DT1"

Tells the network analyzer to upload the current value on trace 1 in ASCII format.

Binary information may be transferred via the optional inline variable expansion.

For example

< "[scope_config]"

would perform a binary expansion of the local variable *scope_config* prior to sending the string to the currently active device.

or more concretely...

< "VSET 1, <VIF>"

Sets the channel 1 programmable voltage to the value contained in the local variable *VIF*.

MORE on inline variable expansion....

You can enclose the variable to be expanded in a variety of symbols which control the *format* of the expansion.

<> Means the variable is expanded directly from memory.
[] An IEEE header will be built for the variable which will then be expanded directly from memory.
{} Means the variable will be expanded as ASCII based on its type. Floating points will expand with a sprintf using %f format. Integers with a %d and strings with %s.

> [var_name [type]]

Receive data from the currently active GPIB device. NOTE that this command keeps trying until some data is received, there is no time out.

You can optionally define the type of the incoming data. NOTE this does not change the type of the variable which you are storing the data into.

If no variable is supplied data will simply be written to standard out in a combined hex and ASCII display.

+ var_name [type [length]]

Defines a local variable *var_name* and associates a data type and size for that variable. For example if you had a scope which stored wave forms as a series of 401 floating point numbers, you might make the following definition...

+ WAVE_FORM f 401

- var_name

Removes a local variable from memory, thus freeing the memory resource.

func_name(arg1 ... argn)

Makes a function call to *func_name* with the appropriate arguments *which MUST be local variables*, constants are not allowed.

? var_name

Displays the contents of *var_name* based on its data type.

\ fff.fff

Suspends command interpretation for fff.fff number of seconds. Use this to allow for device settling times, stacking, averaging etc.

file_name = var_name
var_name = file_name

Transfers data to/from a disk file.

Now that we have reviewed the single character commands supported by the GPIB interpreter, lets look at the individual commands in greater detail.

Initializing the GPIB card

The ! command serves the purpose of initializing the GPIB card in your computer. This command must be done once and only once prior to accessing any GPIB devices. While it is typically the first command in any command file, it is not required. It is possible to define local variables or transfer data in from disk files prior to use of this command. However use of this command *must* precede use of @, <, and > commands.

If the GPIB card is not present or there is no cabled connection from the card to a physical GPIB bus an error message will be generated.

Addressing a GPIB device

Prior to controlling a GPIB device on the bus, you must address the device which you plan on communicating with. This is done with the @ command. For example in the nulling application the HP3577A is device 11 on the GPIB bus. If we wanted to communicate with network analyzer (3577A), then we would first have to use the @ command as follows...

```
@ DEV11
```

The whole purpose of this command is to devote the use of the bus to communication with the device of interest and the controller card in your PC. Once a device has been addressed, we can send device specific commands and receive device specific results over the bus.

If you have more than one device on the GPIB, you simply use @ any time you wish to change devices. For example in the command sequence fragment below...

```
.
.
.

@ DEV5
< "VSET 1, 0.0"
;
@ DEV11
< "RC2;AV0"

.
.
.
```

We address the programmable voltage supply and set channel 1 voltage to 0.0 Volts. Next we address the network analyzer, recall the display and set up in storage register 2 and turn averaging off. The use of the < command is discussed in the following section.

Sending data to a GPIB device

The < command is used to send device specific commands to the device most recently addressed with the last @ command. The syntax for the command is...

< "device_commands"

There must be at least one space between the < command and any data that follows. You must always supply a non-empty string with the < command (it is assumed that it doesn't make any sense to send nothing to a device). Note that *device_commands* are specific to the GPIB device you are addressing. The syntax for *device_commands* will be defined in the remote access portion of that device's user manual. Note that the quotes are part of GPIBTST and must enclose the set of device commands. The quotes *are not* sent to the device.

In its simplest form the < command is used to send a single device specific command to a GPIB device. A simple example we have already seen is setting channel 1 voltage to 0.0V on the programmable voltage supply of the nulling system.

< "VSET 1, 0.0"

However, the < command supports a powerful *local variable* expansion syntax which allows you to transfer almost any type of information (including binary configuration, raw trace data, or IEEE formatted data) to GPIB devices. Lets look in more detail at this expansion syntax; for details concerning local variables, see the section devoted to this topic.

Local variables can be expanded in three ways by enclosing the variable name in an appropriate set of brackets:

<>	Memory dump expansion
{}	ASCII expansion
[]	IEEE format expansion

The type of expansion to use is dictated by the *format* of the device specific command which will be sent and the *data type* of the variable to be expanded.

A simple example of variable expansion which is frequently found in the nulling system command files is...

< "VSET 1, <VIF>"

In which the channel one voltage (imaginary fine amplification voltage) is set to the contents of the local variable *VIF*. Referring to the manual for the programmable voltage supply, we find that the VSET command expects its

voltage to be an ASCII string representation of a floating point value. Lets assume that *VIF* has a data type of "s" or string. Under these conditions the above command is valid. Note that...

```
< "VSET 1, VIF"
```

would NOT work and would generate a syntax error from the programmable voltage supply which expects an ASCII floating point value to follow the "," in the VSET command. Also note that if *VIF* had a data type of float, then the appropriate expansion would be...

```
< "VSET 1, {VIF}"
```

Receiving data from a GPIB device

The > command reads incoming data from a GPIB device. Typically you must precede this command by a device specific query prior to expecting any data from a device. For example if we wanted to know the voltage on channel one of the programmable supply...

```
< "VSET? 1"
>
 0020  0020  0030  002E  0030  0030  0030  000D   |   0.000
Bytes transferred: 8
```

would be the simplest sequence. The combined hex/ascii display is the default in the event that no variable name was supplied as a target for storage. Hex values are listed to the left of the '|' character and the equivalent ASCII translation is on the right. The complete syntax of the > command is...

```
> [var_name [incomming_data_type]]
```

Where *var_name* is any local variable name and *incomming_data_type* is an optional specification of how you want the incoming data treated. You should note that the > command *does not* require you to pre-define your variables prior to use as a target in the > command. The > command will automatically define *and size* the variable for you depending on the supplied optional data type and the amount of information received over the GPIB bus.

For example, if we wanted to save the channel 1 voltage in a local variable for latter use in the command file, one possible sequence would be...

```
< "VSET? 1"          ; Ask programmable supply for CH1 voltage
> CH1_VOLTS s        ; Save the answer in CH1_VOLTS as a string
```

```
? CH1_VOLTS              ; Now display the variable's value
  0.000
```

The > command can be used to collect strings, floating point arrays, IEEE data dumps, and binary device configurations in addition to single values. Here is an example of collecting a waveform trace from an HP 6000 series digital scope...

```
; ---------------------------------------------------------------------------------
; This file contains a sequence of commands to acquire a waveform
;  from the scope and put it in a disk file.
;
;  usage:  type getwave.gib|gpibtst[>wave.out]
;
;
!                                                    ; Open GPIB
communication
@  DEV1                                              ; Access the scope
< ":WAVEFORM:SOURCE CHANNEL1;POINTS 2000"       ; Define waveform
size
< ":WAVEFORM:FORMAT WORD;BYTEORDER MSBFIRST"
< ":WAVEFORM:PREAMBLE?"                            ; Query the waveform
preamble
> _WP s                                            ; Store it in a variable
< ":WAVEFORM:DATA?"                                ; Query the waveform data
> _WD                                              ; Store it in a variable
#  WAVEFORM( _WP _WD _WF )                         ; Waveform calculation
function
wave.bin = _WF                                     ; Save it to the PSD.Ref
file
? _WF                                              ; Send it to standard out
< "SYSTEM:LOCK OFF"                                ; Give scope back to user
```

This command file stores a 2000 point waveform trace in the disk file wave.bin. The waveform comes in as a sequence of integer values representing percent of scale. When combined with the pre-amble in the WAVEFORM function call the integer values are converted to floating point values. The 2000 point floating point array is then stored to disk as well as being echoed to standard out.

The next section dicusses the use of local variables in detail.

Local variables

It is possible to create local variables which exist for each continuous operation of the GPIBTST program. Local variables can be used to hold the contents of

disk files, store values or arrays acquired from a GPIB device, passed as arguments to function calls, be stored in disk files, or displayed at any time.

Local variables can be created automatically by the > command or by some function calls. Local variables can also be created explicitly using the + command. Local variables are never destroyed automatically but this can be done with the - command.

Definition

The syntax for defining local variables is...

```
+ var_name data_type size
```

Where *var_name* is the name you want to use in referencing the variable. *Data_type* is controls how much space the variable will occupy and how information will be treated when it is *read* from the variable. *Size* indicates the number of *elements* of *data_type* the variable refers to. For example if we wanted an array of 401 data points to hold one of the network analyzer traces we might define a variable of floating point values as follows...

```
+ TRACE_1 f 401
```

Which basically says create a local variable called *TRACE_1* which is an array of 401 floating point values. Allowed data types are as follows...

f	32 bit floating point values
d	64 bit floating point values
i	16 bit integer values
l	32 bit integer values
s	8 bit character values

Note that in some cases a local variable is defined *automatically* by some operations. However in all cases, the three elements (*var_name, data_type, and size*) must be supplied or *inferred*. For example in the following command fragment...

```
< "VSET? 1"
> CH1_VOLTS s
```

If the variable *CH1_VOLTS* does not already exist, the > command creates it as follows...

var_name = CH1_VOLTS
data_type = s
size = # of bytes received over the gpib bus

Note that if you do not supply a data type to the > command in the case of automatic variable creation a default of 'i' is used. Function calls may also create *output* variables automatically. Refer to the documented function calls in this manual for details.

Display

Local variables can be displayed with the ? command. The syntax for this command is...

? var_name

The output produced will depend on the variables data_type and size.

Transferring data to/from disk files via local variables

It is possible to load data from a disk file into a local variable. Some example uses of this capability are...

Loading a device configuration from disk
Loading a reference waveform trace from disk
Loading reference or comparison arrays from disk
Transferring data between successive command files

Data is transferred from disk to a local variable with the = command. The syntax for the = command is...

var_name = file_name

to read data from disk or...

file_name = var_name

to save data to disk. Note that all file names *must* contain an extension as this is how GPIBTST knows that you mean to use a file. For example...

```
VIF.DAT = VIF
VIC.DAT = VIC
VRF.DAT = VRF
VRC.DAT = VRC
```

Would save the current values of nulling system voltages (provided VIF, etc contain these values) to disk for latter use by a subsequent command file. Likewise in order to read these values back in at a latter time *in a possibly different command file*, the following command fragment could be used...

```
VIF_OLD = VIF.DAT
VIC_OLD = VIC.DAT
VRF_OLD = VRF.DAT
VRC_OLD = VRC.DAT
```

Making Function Calls

The # command allows you to call functions which have been compiled and linked into the interpreter. This section will document those functions added to GPIBTST to support the nulling system. See the programmer's manual for a description of how to add or modify functions supported by GPIBTST.

The syntax for making a function call is...

```
# func( arg1 arg2 ... argn )
```

When GPIBTST encounters the # command, it parses *func* from the command line and looks to see if the *name* of the function has been made part of GPIBTST's command syntax. If so, GPIBTST checks the arguments for validity, performs any required pre-processing, and then calls the actual function *code* performing the required function.

The # command is simple and lacks flexibility. Only local variables may be supplied as arguments.

The functions added to GPIBTST to support the nulling system are...

SETPARMS(NA_PARMS)

Gets a set of current operating parameters from the network analyzer. Parses and saves key values for latter use by other functions. NA_PARMS is a string variable.

Note that the variables can have any name but must *all* be *defined* prior to making this function call. This function does not automatically create any variables.

FINDNULL(REAL IMAG VRC VRF VIC VIF)

Computes a null based on the current reading from the network analyzer. REAL and IMAG are the current components of the detector signal. The V?? variables should contain the current system voltage values when the function call is made. When the function call returns, these variables will contain the voltages corresponding to the computed null of the system. All the variables are string variables.

Note that the variables can have any name but must *all* be *defined* prior to making this function call. This function does not automatically create any variables.

REALCHECK(R_R R_I C_R C_I)

This function computes and stores (for use by SIGCHECK) the maximum value of the "real" component cancellation signal. Note that the "real" component is produced by setting the real coarse voltage at maximum. The signal resulting from this may have real and imaginary components.

R_R and R_I are string variables containing the real and imaginary components of the received signal with no cancel signal. In other words V1 = V2 = V3 = V4 = 0.0V on the programmable voltage supply.

C_R and C_I are string variables containing the real and imaginary components of the received signal combined with the maximum value of the "real" component cancellation signal. In this case V1 = V2 = V3 = 0.0V and V4 = 4.0V.

Note that the variables can have any name but must *all* be *defined* prior to making this function call. This function does not automatically create any variables.

IMAGCHECK(R_R R_I C_R C_I)

Same as REAL check except C_R and C_I are collected with V1 = V2 = V4 = 0.0V and V3 = 4.0V.

SIGCHECK(R_R R_I C_R C_I RESULT)

This function uses information accumulated from REALCHECK and IMAGCHECK as well as information supplied from its input arguments to determine the cancelable region for the received signal (see the "Cancelability Indicator & Automatic Control" section of this document).

The input arguments are R_R, R_I, C_R, and C_I. The are all string variables. R_R and R_I contain the real and imaginary components of the received signal with no cancel signal. C_R and C_I contain the real and imaginary components of the received combined with a cancel signal produced by the following voltages... V1 = V2 = 0.0V and V3 = V4 = 4.0V.

The single output argument RESULTS is a string variable which holds the result of the cancelability test.

Comment lines

The comment command, ; tells GPIBTST to move on to the next command line as the current line is a comment line. This command *must* appear in column 1 in order to comment out an entire line.

This command can also be used on lines containing a valid command, providing you with line by line comment capabilities in your command files.

Building and running command files

Command files can be built with any text editor. *ALL* command files must end with a blank line. A blank line ends the GPIBTST interpreter program. Some text editors can be configured to strip blank lines from files as they are written to disk. If your editor has this capability and you want to use it to write GPIBTST command files, disable the strip blank lines capability.

Note that you can always use the NOTEPAD.EXE program supplied with Windows or W95 to write command files.

Rather than pursue a general discussion of command file structure, we will simply review the existing command files used by the nulling system.

Existing Command Files

This section contains a listing of the current command files in use by the nulling system. The file extension .GIB was used for all nulling system command files. This extension is not required but it is convenient when copying or listing files.

The Coarse Null command file

The command file which computes a coarse null is GETNULL.GIB. The file is listed below for hard copy reference. This is one of the most complicated files and is basically so because GPIBTST does not yet support the * command (which will let you call subfiles n number of times).

This file basically performs three successive nulling iterations duplicated three times in the file (GPIBTST currently has no iteration capabilities). A coarse null operation always begins from the same starting point by initializing all nulling system voltages to 0.0V.

In the following *indendented* **bold** comments are NOT contained in the file.

This section of the file is a comment header and provides general documentation concerning the file.

```
; ---------------------------------------------------------------------------
; This command file finds a null using the HP3577A and the HP6624A
;
;  You must have set up the two devices using the setup.gib file
;  prior to running this command file.
;
;  Use the command file from a DOS prompt as follows...
;
;    TYPE GETNULL.GIBIGPIBTST>GETNULL.OUT
;
```

```
;   or...
;
;   TYPE GETNULL.GIBIGPIBTST
;
;   to see output on your screen
;
```

The first active commands in the file are the ! command to initialize access to the GPIB card in the PC. This is followed by the @ command used to address the programmable voltage supply. Several commands are then sent to the programmable voltage supply initializing all voltages to zero.

```
!                           ; Initialize the GPIB card in PC
@ DEV5                      ; Access the programmable voltage supply
< "VSET 1, 0"               ; And initialize it
< "VSET 2, 0"
< "VSET 3, 0"
< "VSET 4, 0"
;
@ DEV11                     ; Access the network analyzer
```

Here we define the local variables which will be used by this command file. These can actually be defined anywhere in the file prior to use but are grouped together for aesthetic reasons.

```
+ REAL  s 20
+ IMAG  s 20
+ VRC   s 20
+ VRF   s 20
+ VIC   s 20
+ VIF   s 20
+ PARMS s 250               ; Current scope parameters
+ CANCEL s 50               ; Base cancel signal
```

Now we get the current operating parameters from the network analyzer. The analyzer provides all operating parameters in a single long ascii string. It does not allow queries for individual parameters. The SETPARMS function will save the operating parameters in global variables for use by other functions in the nulling system.

```
< "SV3"                     ; Save the user's current display
< "FM1;DCH"                 ; Only way to get certain parameters is to dump entire collection
> PARMS s                   ; Load the string variable
# SETPARMS( PARMS CANCEL )  ; Parse parms and save for latter function calls
```

Storage register 2 has been configured to contain the operating conditions for the network analyzer for use with the nulling system. It is not possible to completely configure the network analyzer via GPIB as there are insufficient handles. See the network analyzer user manual for a discussion of this. Because of this we have been forced to use the network analyzer storage register to establish nulling system operating conditions. If storage registers 1 or 2 are changed, then these command files will not work correctly.

Having recalled the nulling system operating conditions, we now collect the real and imaginary components of the received signal.

```
< "RC2;AV0"                 ; Recall the nulling trace displays
\ 1                         ; Let machine stabilize
< "FM1;DT1"                 ; Query network analyzer for real component
> REAL s                   ; Store the answer in local variable
< "FM1;DT2"                 ; Query for the imaginary component
> IMAG s                   ; Store in local variable
< "RC3;TR2;AV0"             ; Return device to user display
```

```
;
```

Next we get the values of the current voltages. I know we just set them to zero, however there can be several hundredths of a volt difference in what was asked to be set and what got set.

```
@ DEV5                          ; Access the programmable voltage supply
< "VSET? 1"                     ; Get the current settings of the voltages
> VIF s                        ; and store them in local variables
< "VSET? 2"
> VRF s
< "VSET? 3"
> VIC s
< "VSET? 4"
> VRC s
```

We now have the information needed to compute nulling voltages. The FINDNULL function does this for us. The new nulling voltages will be contained in VRC, VRF etc after this call. Once they are returned we will implement them.

```
;                               ; Make a first pass for nulling
# FINDNULL( REAL IMAG VRC VRF VIC VIF )     ; Compute the null in terms of the 4 voltages
? VRC                           ; REAL COARSE
? VRF                           ; Display nulling values (REAL FINE)
? VIC                           ; IMAGINARY COARSE
? VIF                           ; IMAGINARY FINE
;
< "VSET 1, <VIF>"                   ; And implement the changes
< "VSET 2, <VRF>"
< "VSET 3, <VIC>"
< "VSET 4, <VRC>"
```

We will now make a second pass at implementing a null.

```
;                               ; Make a second pass at nulling
@ DEV11                             ; Select the network analyzer
< "RC2;AV0"                         ; Recall the nulling trace displays
\ 1                             ; Wait 3 seconds
< "FM1;DT1"                         ; Query network analyzer for real component
> REAL s                       ; Store the answer in local variable
< "FM1;DT2"                         ; Query for the imaginary component
> IMAG s                       ; Store in local variable
< "RC3;TR2;AV0"                     ; Return device to user display
;
@ DEV5
# FINDNULL( REAL IMAG VRC VRF VIC VIF )     ; Compute the NULL
? VRC                           ; Display Results, REAL COARSE
? VRF                           ; REAL FINE
? VIC                           ; IMAGINARY COARSE
? VIF                           ; IMAGINARY FINE
;
< "VSET 1, <VIF>"                   ; And implement the changes
< "VSET 2, <VRF>"
< "VSET 3, <VIC>"
< "VSET 4, <VRC>"
```

Now the third pass.

```
;                               ; Make a third pass at nulling
@ DEV11                             ; Select the network analyzer
< "RC2;AV0"                         ; Recall the nulling trace displays
\ 3
< "FM1;DT1"                         ; Query network analyzer for real component
> REAL s                       ; Store the answer in local variable
< "FM1;DT2"                         ; Query for the imaginary component
> IMAG s                       ; Store in local variable
< "RC3;TR2"                         ; Return device to user display
;
@ DEV5
# FINDNULL( REAL IMAG VRC VRF VIC VIF )
? VRC                           ; REAL COARSE
? VRF                           ; Display nulling values (REAL FINE)
```

```
? VIC                          ; IMAGINARY COARSE
? VIF                          ; IMAGINARY FINE
;
< "VSET 1, <VIF>"                  ; And implement the changes
< "VSET 2, <VRF>"
< "VSET 3, <VIC>"
< "VSET 4, <VRC>"
;
@ DEV11                          ; Select the network analyzer
< "RC2;AV0"                      ; Recall the nulling trace displays
\ 5
< "FM1;DT1"                      ; Query network analyzer for real component
> REAL s                        ; Store the answer in local variable
< "FM1;DT2"                      ; Query for the imaginary component
> IMAG s                        ; Store in local variable
< "RC3;TR2;AV0"                      ; Return device to user display
;
@ DEV5
# FINDNULL( REAL IMAG VRC VRF VIC VIF )
? VRC                           ; REAL COARSE
? VRF                           ; Display nulling values (REAL FINE)
? VIC                          ; IMAGINARY COARSE
? VIF                          ; IMAGINARY FINE
;
< "VSET 1, <VIF>"                  ; And implement the changes
< "VSET 2, <VRF>"
< "VSET 3, <VIC>"
< "VSET 4, <VRC>"
```

The Fine Null command file

The command file used to compute a fine null is called 1PASSNUL.GIB and is listed below.

In the following... *indendented* **bold** comments are NOT contained in the file.

```
; ---------------------------------------------------------------------------
; This command file (1PASSNUL.GIB) performs a one pass computation
;
```

Initialize the GPIB card in the PC and access the network analyzer. Then define local variables to be used in this file.

Note that this file is meant to be used on an incremental basis at any given starting condition. Therefore we DO NOT initialize the voltages to zero.

```
!                           ; Open the GPIB card
@ DEV11                         ; Access the network analyzer
;                           ; Define the variables we will be using
+ REAL  s 20                    ; Real component of signal to cancel
+ IMAG  s 20                    ; Imaginary component of signal to cancel
+ VRC   s 20                    ; Cancel signal real component coarse gain
+ VRF   s 20                    ; Cancel signal real component fine gain
+ VIC   s 20                    ; Cancel signal imaginary coarse gain
+ VIF   s 20                    ; Cancel signal imaginary fine gain
+ PARMS s 250
+ CANCEL s 50                       ; Base cancel signal
;
CANCEL = cancel.out                 ; Read base cancel signal calc'd by SIGCHECK.GIB
```

Preserve the current user's environment in storage register 3. The get and save the current network analyzer operating conditions... remember these could have been changed in between successive invocations of this command file.

```
< "SV3"                         ; Save the user's current display
< "FM1;DCH"                     ; Only way to get certain parameters is to dump entire collection
> PARMS s                       ; Load the string variable
# SETPARMS( PARMS CANCEL )                  ; Parse parms and save for latter function calls
```

Now we set up the network analyzer for nulling system operation and obtain the real and imaginary components of the current signal.

```
< "RC2;AV0"                     ; Recall the nulling trace displays
\ 0.5                       ; Let network analyzer stabilize
< "FM1;DT1"                     ; Query network analyzer for real component
> REAL s                       ; Store the answer in local variable
< "FM1;DT2"                     ; Query for the imaginary component
> IMAG s                       ; Store in local variable
< "RC3;TR2;AV0"                     ; Return device to user display
;
```

Next we get the current voltages (they could have been set manually or by the last iteration of one of the nulling programs).

```
\ 0.5
@ DEV5                          ; Access the programmable voltage supply
< "VSET? 1"                     ; Get the current settings of the voltages
> VIF s                        ; and store them in local variables
< "VSET? 2"
> VRF s
< "VSET? 3"
> VIC s
< "VSET? 4"
```

```
> VRC s
```

Now we are ready to compute, display, and implement the nulling voltages.

```
;                                   ; Make a first pass for nulling
# FINDNULL( REAL IMAG VRC VRF VIC VIF )     ; Compute the null in terms of the 4 voltages
? VRC                               ; REAL COARSE
? VRF                               ; Display nulling values (REAL FINE)
? VIC                               ; IMAGINARY COARSE
? VIF                               ; IMAGINARY FINE
;
< "VSET 1, <VIF>"                   ; And implement the changes
< "VSET 2, <VRF>"
< "VSET 3, <VIC>"
< "VSET 4, <VRC>"
```

The Reset command file

The reset command file is used to reset the system to standard conditions. This command makes use of the n.bat file. We will review this file because it is a good illustration on extending the DOS command line to support nulling system operations.

```
@ECHO OFF
REM Arguments are RealCoarse RealFine ImagCoarse ImagFine

 ECHO !>setvolts.gib
 ECHO @ DEV5>>setvolts.gib
 ECHO { "VSET 4, %1;VSET 2, %2;VSET 3, %3; VSET 1, %4">>setvolts.gib
 TYPE CR>>setvolts.gib

 TYPE setvolts.gib|gpibtst
 TYPE getreim.gib|gpibtst

 ECHO !
 ECHO ! New null settings Rc %1  Rf %2  Ic %3  If %4
 ECHO !
```

In the above DOS batch file, we build a GPIBTST command file on the fly and then invoke GPIBTST to process the resulting file. The reset command simple makes a call to N.BAT shown above with the following arguments

n 0 0 0 0

N.BAT takes these arguments and builds SETVOLTS.GIB which looks as follows once it is built...

```
!
@ DEV5
{ "VSET 4, 0;VSET 2, 0;VSET 3, 0;VSET 4, 0"
```

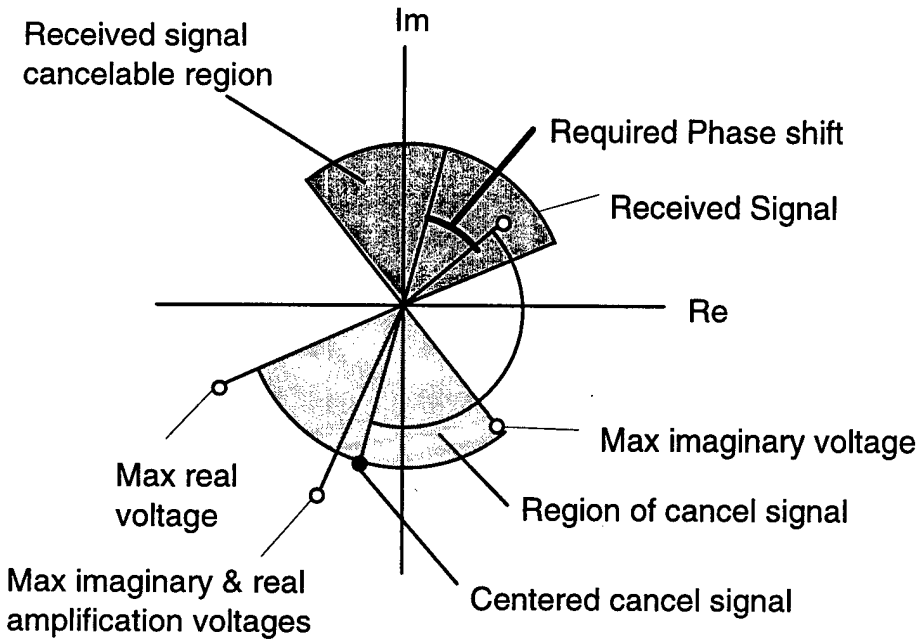
N.BAT then sends the newly built command file to GPIBTST for interpretation via the

TYPE setvolts.gig|gpibtst

command. As you have probably guessed by now n.bat can be used from the DOS command line to manually obtain a null.

The Signal Check command file

The signal check command file, named SIGCHECK.GIB, is used to check the received signal for cancelability. For convenient visualization purposes, the figure in the "Cancelability Indicator & Automatic Control" document section has been reproduced below.



Now lets look at SIGCHECK.GIB...

```
; ---------------------------------------------------------------------------
; This command file (SIGCHECK.GIB) checks the received signal for
;  "cancelability". A quadrant check and magnitude check are performed
;  by this file.
;
!                              ; Open the GPIB card
;
; Define variables used by this command file
;
```

Define the local variables to be used by this command file.

```
+ R_R s 20                     ; Real component of received signal
+ R_I s 20                     ; Imaginary component of received signal
+ C_R s 20                     ; Components of the combined signal.
+ C_I s 20                     ;
+ V1  s 20                     ; Original voltages
+ V2  s 20
+ V3  s 20
+ V4  s 20
+ RESULT s 100                 ; Result of cancelability check
+ CANCEL s 50                  ; Storage for base cancel signal
```

When performing a cancelability check, we must preserve the current operating environment of the equipment. This file can be

used with a partially nulled signal, so we do not want to pertebate those operating conditions.

```
;
; Save current voltages so we can leave it as
;  we found it.
;
@ DEV5                          ; Access the programmable voltage controller
< "VSET? 1"                     ; Save the current voltages
> V1 s
< "VSET? 2"
> V2 s
< "VSET? 3"
> V3 s
< "VSET? 4"
> V4 s
```

We must collect 4 real and imaginary component pairs as indicated in the diagram at the start of this section. First we must obtain a value for the raw received signal with no cancel signal. Since we can never eliminate the received signal it must be subtracted from 3 subsequent measurements to determine values for the cancel signal. The fact that the received signal may drift slightly during this procedure is of little concern as we will be taking measurements at maximum cancel signal values.

```
;
; ------------------------------------------------------------------------
; Measure the received signal
;
< "VSET 1, 0.0"                 ; Set the cancel signal to zero
< "VSET 2, 0.0"
< "VSET 3, 0.0"
< "VSET 4, 0.0"
;
@ DEV11                         ; Access the network analyzer
;
< "SV3"                         ; Save the user's current display
< "RC2;AV3"                     ; Recall the nulling trace displays
\ 2                             ; Let network analyzer stabilize
< "FM1;DT1"                     ; Query network analyzer for real component
> R_R s                         ; Store the answer in local variable
< "FM1;DT2"                     ; Query for the imaginary component
> R_I s                         ; Store in local variable
```

Having collected the raw received signal, we will now collect the maximum "imaginary" cancel signal component. This value is the "imaginary" cancel signal component in name only because V1 = V2 = V4 = 0.0V and V3 (imaginary coarse) = 4.0V. However it is not only possible but very likely that the resulting vector with have both real and imaginary components.

The IMAGCHECK function call will subtract the raw received signal from the values collected here and save the resulting maximum "imaginary" cancel signal component for use in the SIGCHECK function.

```
;
; ------------------------------------------------------------------------
; Measure several combinations of the cancel signal... we will have to subtract
;  out the received signal in each case
;
@ DEV5                          ; Access programmable voltage controller
```

```
\ 0.5
< "VSET 3, 4.0"                    ; Set maximum imaginary component
;
@ DEV11                            ; Access the network analyzer
\ 2
< "FM1;DT1"                        ; Query network analyzer for real component
> C_R s                           ; Store the answer in local variable
< "FM1;DT2"                        ; Query for the imaginary component
> C_I s                           ; Store in local variable
# IMAGCHECK( R_R R_I C_R C_I )     ; Compute and save just the imaginary portion
;
;
```

We will now do the same thing for the maximum "real" cancel signal component.

```
@ DEV5                             ; Access programmable voltage controller
\ 0.5
< "VSET 3, 0.0"                    ; Clear the imaginary component
< "VSET 4, 4.0"                    ; Set maximum real component
;
@ DEV11                            ; Back to network analyzer
\ 2
< "FM1;DT1"                        ; Query network analyzer for real component
> C_R s                           ; Store the answer in local variable
< "FM1;DT2"                        ; Query for the imaginary component
> C_I s                           ; Store in local variable
# REALCHECK( R_R R_I C_R C_I )     ; Compute and save just the real portion
;
;
```

At this point we have values for "Received Signal", "Max imaginary voltage", and "Max real voltage" in the diagram shown at the top of this section. In the following section we will obtain a value for "Max imaginary & real amplification voltages" and make a call to the SIGCHECK function. This function will compute the "Region of cancel signal" and the "Centered cancel signal". It will subsequently use these computations to see if "Received Signal" falls within "Received signal cancelable region" and will report the results of this check in the local variable RESULT.

```
@ DEV5                             ; Back to programmable voltage controller
\ 0.5
< "VSET 3, 4.0"                    ; Cancel signal is now at max
;
@ DEV11                            ; Back to network analyzer
\ 2
< "FM1;DT1"                        ; Query network analyzer for real component
> C_R s                           ; Store the answer in local variable
< "FM1;DT2"                        ; Query for the imaginary component
> C_I s                           ; Store in local variable
# SIGCHECK( R_R R_I C_R C_I RESULT )   ; Perform a complete cancelability check
```

Having check received signal cancelability, we restore the original operating conditions. Note we have not yet output the results of this check because *this is how we stay synchronized with the LabVIEW nulling application.*

```
;
; ------------------------------------------------------------------------
; Restore equipment to original condition
;
@ DEV5                             ; Restore original voltages
\ 0.5
< "VSET 1, <V1>"
< "VSET 2, <V2>"
< "VSET 3, <V3>"
< "VSET 4, <V4>"
```

```
;
@ DEV11                         ; Restore user's original display
< "RC3;TR2"                     ; Return device to user display
```

Now that we have completed all other tasks, the result of the cancelability check is output. This will let the LabVIEW application know that we have completed our task. RESULT will be written to standard output, which has been redirected to a disk file when this command file has been invoked from the LabVIEW nulling application. The LabVIEW nulling application will read the disk file, and display RESULT on the user's screen inside the LabVIEW nulling application.

```
;
;
# GETCANCEL( CANCEL )           ; Return the base cancel signal (calced by sigcheck )
cancel.out = CANCEL
sigcheck.out = RESULT
? RESULT
```

16 APPENDIX C: NUMERICAL EM MODELING

We employed electromagnetic modeling algorithms to help guide the design process. Our initial studies were carried out using algorithms which are designed to handle conductive targets buried in a layered half space. We chose to use LLNL NEC4.1D [22, 23] and the University of California at Berkeley (UCB) Sheet algorithm [24, 25] for this purpose. Later, we used the Temporal Scattering and Response (TSAR) program [26] to model more complex conductive targets and some dielectric targets.

16.1 Validation of EM Modeling Codes

Before employing the numerical modeling algorithms, we verified the accuracy of these algorithms. In order to validate the numerical algorithms, the numerical solutions were compared to analytic solutions of canonical problems, and to solutions of other, very different, numerical modeling algorithms, as well as to experimental data. The algorithms and our validation tests are discussed below.

16.1.1 Comparison of NEC with a Quasi-Static Sphere Model

The Numerical Electromagnetic Code (NEC) [22] was first developed in 1981. The algorithm development was funded by the US Navy for modeling ship-board HF antenna performance. It is a Method of Moment algorithm that models the electromagnetic scattering from structures (and antennas) built with small current segments. These current segments are joined together to build larger more complicated segments.

To model the effect of sea-water on the performance of Navy antennas, the presence of a lossy dielectric half-space was incorporated into the numerical algorithm. The effect of the half-space is taken into account by the implementation of the solution to the Sommerfeld integral, which calculates the electromagnetic energy reflected from, and transmitted into the half-space.

In later versions of the code (NEC3.0), electromagnetic models were allowed to have wire segments penetrating and completely below the half-space interface. This was added presumably to model the presence of antenna ground-screens. With this capability, it became possible to apply the code to geophysical modeling – that is, to use wire segments to model highly conductive structures buried within the half-space.

NEC can model the current generated on, and fields scattered from, wire segments excited by a variety of sources. NEC takes into account the possibility of an air-earth interface and can calculate the secondary field associated with subsurface conductive objects excited by a source placed above the interface. A large variety of targets can be modeled using wire segments, and the segments can be made resistive as long as skin-depth considerations are included in the model. The homogeneous half-space which models the earth can be assigned a complex conductivity, and the wire segments can be assigned series or parallel resistance, capacitance, and inductance, as well as bulk wire conductivity and permeability. No real frequency limitation exists for the integral-equation formulation that was employed in NEC, however, we found that the algorithm doesn't always produce the correct results at low frequencies for all types and orientations of targets. This is discussed in more detail later. One stated limitation of NEC is that conducting structures must be made up of at least 10 wire segments per effective wavelength. Most researchers have employed NEC for antenna performance and radar scattering applications. In fact, we have not found any

studies which demonstrate the accuracy of the results produced by NEC for the relatively low-frequency, induction problem that we are interested in modeling. Therefore, before using NEC to guide our design, we had to verify its accuracy.

NEC4.1D is the particular version of this method of moments code that we employed. In the models developed for NEC4.1D, the transmitter and receiver coils were constructed of wire segments. The transmitter coil was excited by a voltage gap source and the current generated on the receiver coil was used to predict the received total field.

In order to test the ability of NEC4.1D to accurately model the transmitter and receiver loops, we first compared with the results produced by an analytic (quasi-static) solution of a sphere in a free-space dipole field [27]. The solution of this canonical electromagnetic problem, simplified by the quasi-static approximation, was simple enough to be easily implemented numerically, and the geometry of the problem could also be modeled with NEC. The solution of the fields induced by a conducting and permeable sphere in the presence of a magnetic dipole source could thus be used to validate the free-space modeling capabilities provided by NEC. Further studies were required to test NEC's implementation of the half-space problem.

To model the conducting sphere, a wire cage formed by great circle wire elements and latitudinal segments separated by 18 degrees was employed. The model consisted of 10 great circles, and 9 latitudinal circles, for a total of 380 elements. To compare the NEC sphere model solution to those of Grant and West [27] as implemented by Lohda and West [28], the NEC sphere model, with perfectly conducting wire elements, was placed in free space. The transmitter and receiver antennas were modeled as horizontal square loops measuring 15 cm to a side. These loops were loaded with 50 ohm resistors to model signal amplifier and receiver input impedances. The currents excited on the transmitter loop, and generated on the receiver loop were internally calculated by NEC. These complex currents are proportional to the transmitted and received magnetic fields, and were used to calculate the normalized secondary field due to the presence of the conducting sphere.

Eight model geometries in all were considered: Six following the Lohda and West geometries as described in [28, Figures 4 and 5], and then two geometries comparing results of horizontal loop and null-configured loop surveys. In order to match the NEC model, the conductivity of the sphere in the analytical formulation was set very high. The operating frequency for the comparison was chosen to be 80 KHz.

Figures 1 and 2 show two curves that compare the NEC calculated solution to that of the analytic implementation. The transmitter and receiver loop antennas used for this geometry were coaxial. The transmitter and receiver antennas were separated by 1m, and the transmitter trailed the receiver. Figure 3 compares the analytic and NEC algorithm solutions for a 10 cm sphere when the transmitter and receiver coils are located 20 cm from the center of the sphere. This time the transmitter coils' axis is vertical, while the receiver coils' axis is horizontal and directed radially from the transmitter coil. Thus, the transmitter and receiver are arranged in a null-coupled configuration. All of the results show excellent agreement, thereby indicating that NEC can be used to accurately model coil antennas and spherical targets in free space.

4.1.2 Comparison of NEC with the UCB Sheet Algorithm

The University of California at Berkeley has recently made public a computer algorithm that calculates the electromagnetic fields of a magnetic dipole in the presence of a thin sheet of finite conductivity buried in a lossy medium. This program has its origin in a dissertation by Zhou

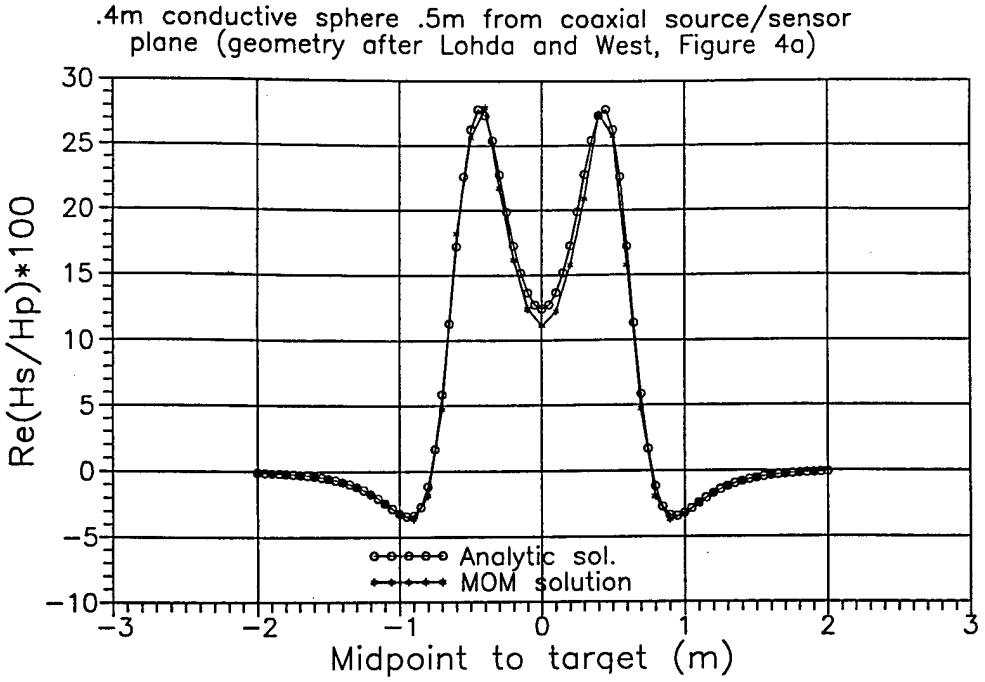


Figure 1: Normalized secondary fields of a conductive sphere with a 40 cm radius in free-space. The horizontal and coaxial transmitter/receiver pair is 50 cm from the center of the sphere.

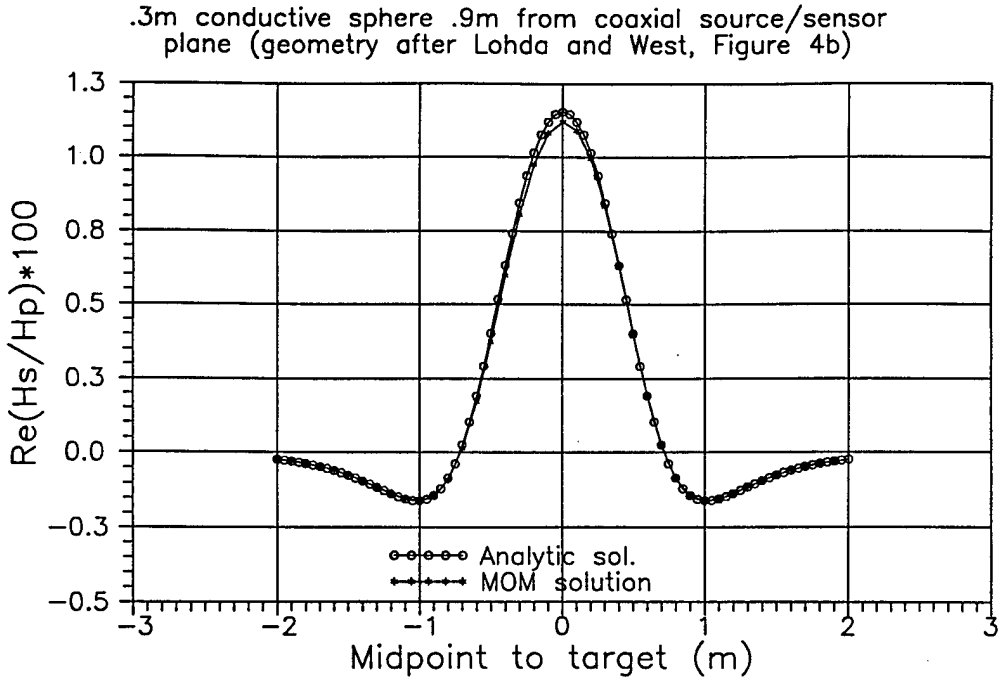


Figure 2: Normalized secondary fields of a conductive sphere with a 30 cm radius in free-space. The horizontal and coaxial transmitter/receiver pair is 90 cm from the center of the sphere.

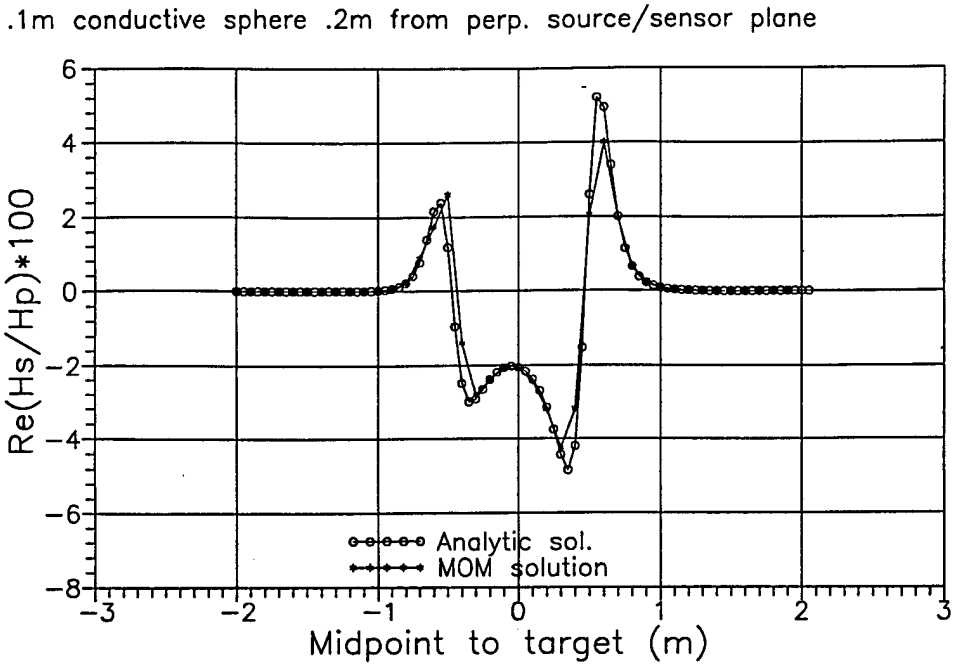


Figure 3: Normalized secondary fields of a conductive sphere with a 10 cm radius in free-space. The transmitter / receiver pair is null-coupled and 20 cm from the center of the sphere.

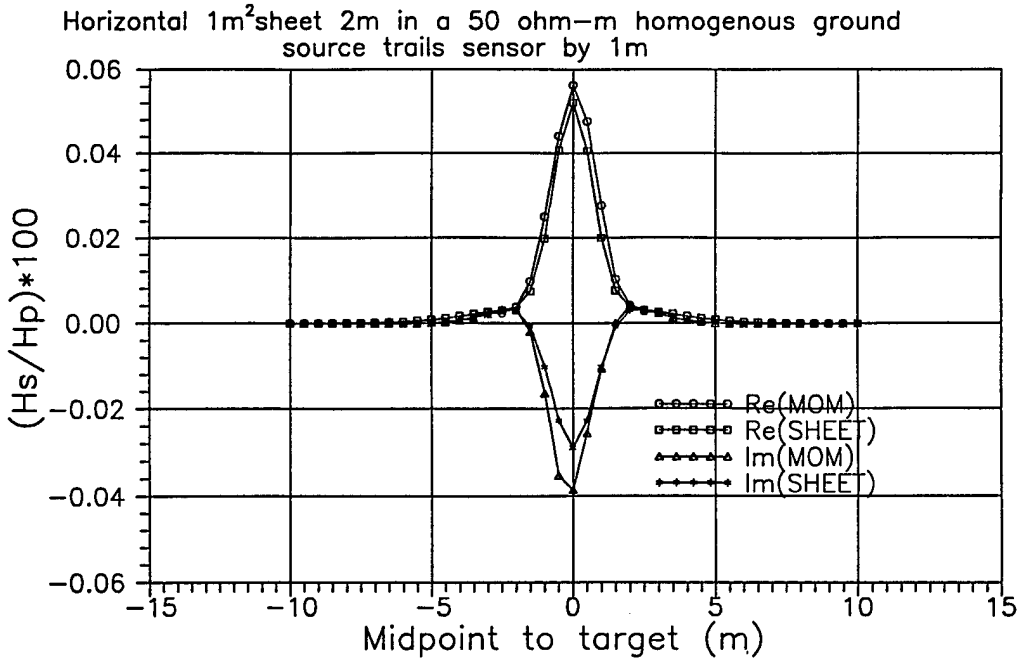


Figure 4: The secondary fields from a horizontal 1 m^2 conducting sheet buried 2 m in a σ=.02 S/m ground.

[25], and is a numerical implementation of the integral equation solution for the fields under the quasi-static approximation. The thin-sheet target of the UCB Sheet program can be assigned a thickness-conductivity product (conductance) value, as well as a conductivity for the earth in which the target is buried. The sheet target can be oriented in any direction, and buried at any depth within the layer.

The UCB Sheet algorithm has been widely tested and compared to other algorithms and published results in the geophysical context. It can model a highly-conductive target, while taking into account the effect of the half-space interface. Thus, even though its geometry is limited to thin sheet targets, it can be used to validate the NEC generated solutions for problems involving sources in the presence of a half space.

To this end, a 1 m square sheet, whose center was buried 2 m below the surface of a 50 ohm-m earth was modeled. The operating frequency, as input into the NEC and Sheet algorithms, was 1 MHz. The target in the UCB Sheet algorithm was assigned a conductance of 1 MSiemen, while the NEC target defaulted to a perfectly conducting wire model. As before, the transmitter and the receiver were separated by 1 m, with the transmitter trailing the receiver.

Three cases were modeled and compared: a horizontal sheet, a sheet inclined at a 45 degree angle to the survey line (dipping towards the receiver), and a vertical sheet. These comparisons are presented in Figures 4 through 11. Both the in-phase and quadrature components of the secondary fields are included in these figures, since the reflection coefficient from the air/earth interface is expected to be complex.

Figure 4 shows good agreement between the in-phase and quadrature components of the secondary fields as calculated by the two different algorithms. The NEC solution shows a slightly larger peak response than the UCB Sheet solution, especially for the quadrature component. The shapes of the normalized, secondary-field curves are very similar, however.

Figures 5 and 6 compare the in-phase and quadrature components of the two algorithms in the presence of the dipping sheet, respectively. Again NEC over-predicts the response of the sheet relative to the UCB Sheet algorithm, but all features, specifically the inflection to the right of the curve maxima, are similar for both algorithm solutions. Even in Figure 6 where the two solutions are found to deviate somewhat, the trends are similar.

Figures 7 and 8 compare the in-phase and quadrature components of the two algorithm solutions in the presence of a vertical sheet, respectively. The NEC solution, plotted in Figure 7, shows good agreement with the UCB result. As before, the magnitude of the secondary field maxima is slightly larger in the NEC solution. The shape of the curve, however, closely resembles that of the UCB solution, an important attribute for algorithms used to model the field response along a geophysical survey line. In Figure 8, the NEC solution deviates dramatically from the UCB Sheet solution. It is interesting to note that the maxima of this aberration occurs when the receiver antenna is directly over the vertical sheet target, a phenomenon that has been observed in other NEC model analyses. During a discussion with Dr. Burke (one of the authors of NEC), he mentioned that he wasn't surprised that NEC gave poor results for the case where the target was placed in a half space directly under the source. Instead of employing numerical integration to compute each one the the required Sommerfeld integrals, NEC precomputes a table of Sommerfeld integrals using numerical integration, and then uses the table to interpolate the value of the field at each observation point. He mentioned that the interpolation table for the Sommerfeld integrals has not been optimized for the parameters encountered in geophysical surveys. The normal frequency range of geophysical surveys, even for shallow target detection, is low due to the effects of frequency-dependent signal

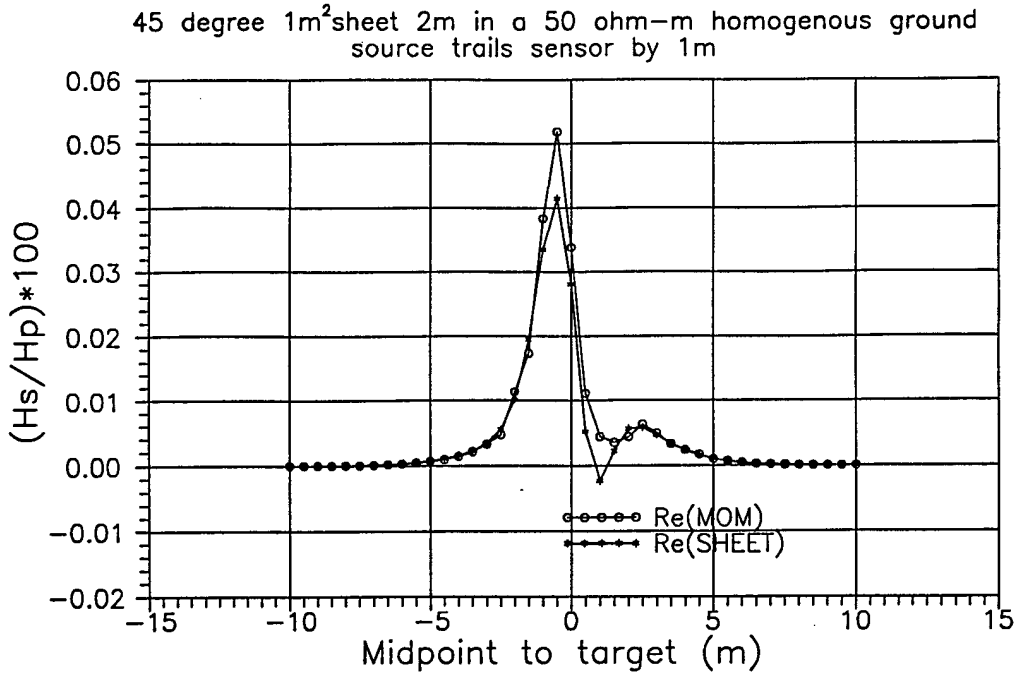


Figure 5: The in-phase secondary fields from a 45 degree dipping 1 m^2 conducting sheet whose center is buried 2 m in a σ=.02 S/m ground.

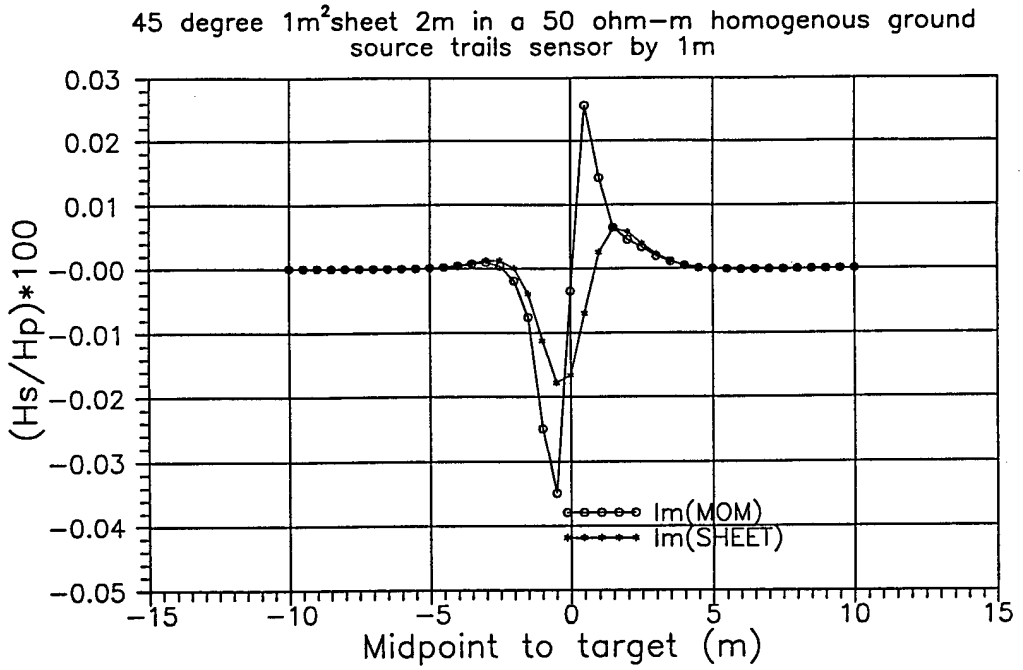


Figure 6: The quadrature secondary fields from a 45 degree dipping 1 m^2 conducting sheet whose center is buried 2 m in a σ=.02 S/m ground.

attenuation of the ground. Because of the low operating frequencies, the usual geophysical survey geometry places the source and sensor loops very close to the half-space interface. Furthermore, the targets are often located at shallow burial depths directly below the source and sensor loops. These spacings are very different from those usually employed in the higher-frequency EM modeling applications where NEC has been optimized and thoroughly tested.

After much study of the use of NEC to model the electromagnetic response of geophysical targets, two shortcomings of the code have been empirically identified. The first shortcoming is that NEC does not seem to correctly predict the received field over an imperfectly conducting half-space in the absence of a buried conducting target when the source antenna is located very close electrically to the half-space. This is thought to be due to NECs implementation of the Sommerfeld integral solution. This shortcoming is rendered negligible when the received fields due to a buried target are normalized by the free-space received fields or the fields received in the absence of the target (homogeneous half-space fields).

The second identified shortcoming is 'spiking' of the NEC calculated data when the source and receiver antennas are in certain orientations and distances from the target. Spiking is defined as a largely anomalous data point in a survey line taken over the target when the neighboring points make a smooth response curve over the survey.

Since NEC was designed for higher-frequency applications, it is not surprising that it occasionally yields poor results for the relatively low-frequency applications that we are interested in. Of course, NEC could be optimized for lower-frequency problems such as ours. However, given the current state of the program, we feel that the UCB Sheet algorithm is better suited for carrying out the relatively low-frequency simulations that are of interest in the geophysical community.

16.1.3 Validation of TSAR

NEC can sometimes be used to model conductive targets in a half space, but it cannot be applied to dielectric targets. Furthermore, the quasi-static formulation of Sheet cannot be applied to dielectric targets since displacement current effects are neglected in the earth. Therefore, we investigated the use of the Temporal Scattering and Response (TSAR) finite-difference time-domain (FDTD) algorithm to predict the fields associated with more general targets (e.g., dielectric and finite conductivity) and more complex and realistic models for the earth. TSAR was developed by Robert McLeod, Steve Pennock, and Scott Ray [26] at the Lawrence Livermore National Laboratory, in Livermore, California. This FDTD program incorporates the Mur [29] boundary conditions (among others) and uses the Leapfrog implementation to calculate the total fields at each successive cell in the model work-space.

Associated with TSAR are a series of pre-processing algorithms that allow for the automatic generation of the finite-difference mesh from a given solid geometry model. Visualization and editing programs that allow viewing and minor corrections to the generated mesh are also included in the FDTD package. TSAR allows the placement of magnetic and/or electric dipoles as radiators and receivers at arbitrary locations and directions within the model mesh and, as stated earlier, allows a variety of boundary conditions to be specified at the mesh boundaries. With this suite of programs, a very flexible, relatively easy to implement, and well maintained FDTD electromagnetic modeling package has become available.

Using TSAR, the time-domain response of a 3D vertical magnetic dipole over a flat earth was calculated at defined points in the model mesh. These time-history solutions were transformed

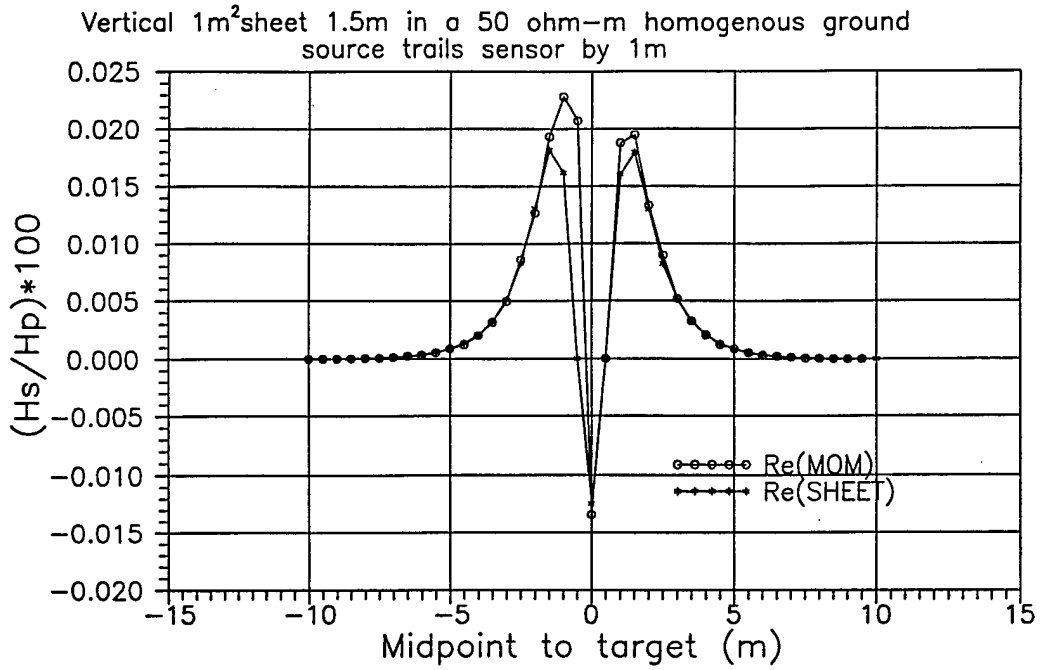


Figure 7: The in-phase secondary fields from a vertical 1 m^2 conducting sheet whose center is buried 2 m in a σ=.02 S/m ground.

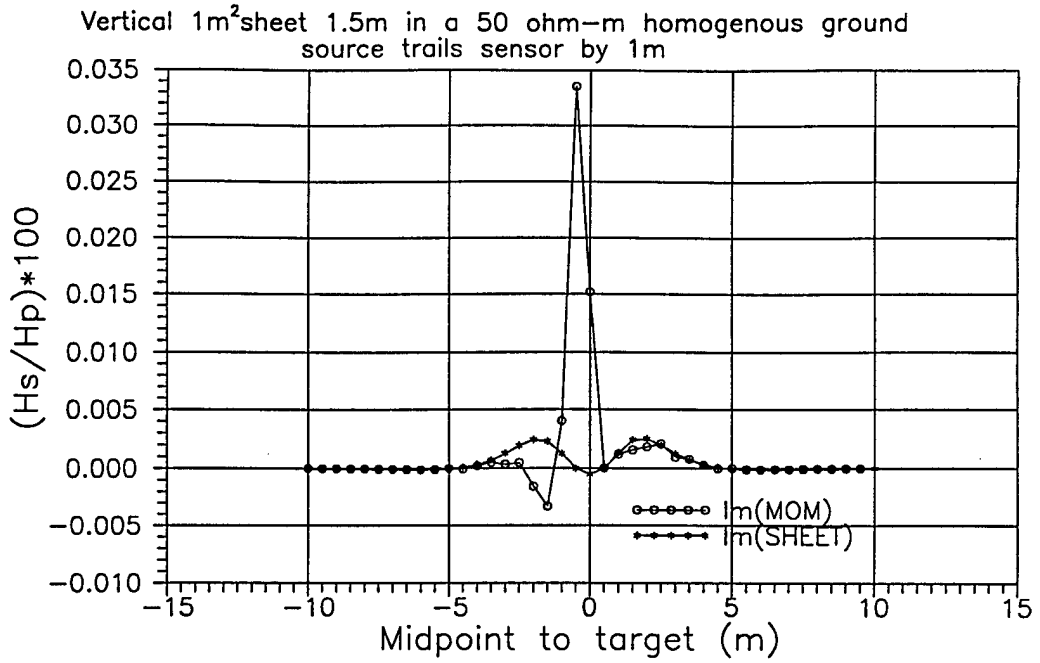


Figure 8: The quadrature secondary fields from a vertical 1 m^2 conducting sheet whose center is buried 2 m in a σ=.02 S/m ground.

into their frequency-domain counterparts using a Fast-Fourier Transform, and the ellipticity was calculated from the resulting total frequency-domain magnetic-field components.

Stewart et al. presented results of magnetic ellipticity calculations using a horizontal loop over a homogeneous earth having different electromagnetic parameters [30]. Their solutions were published for the 1 MHz to 100 MHz range, and were calculated by the numerical implementation of the general integral equation solution as reported by Ryu et al. [31]. Anderson, using the same implementation as Stewart et al's, obtained magnetic ellipticity responses of a 2 and 3 horizontally-layered earth [32]. The frequency range of Anderson's results includes the 1 MHz to 100 MHz range of interest here, and thus allowed for the validation of the FDTD calculations.

A mesh measuring 12 m by 16 m by 16 m, was constructed to validate the FDTD solution with Stewart et al's results. The cells in the workspace were cubic, measuring 0.10 m per side, and the half-space interface was modeled in the $x - y$ plane at $z = 10$ m. The source vertical magnetic dipole (VMD) was placed at $x = 6$ m, $y = 6$ m, and at $z = 10.26$ m, while the received magnetic field components were calculated at $x = 6$ m, $y = 10$ m, and $z = 10.26$ m (thus modeling the transmitter and receiver antennas that are 4 m apart, and 0.26 m above the interface). Parameters for the half-space model were taken from a case presented by Stewart et al. — a resistivity of 50 ohm-m and relative dielectric constant of 10 [30]. The FDTD program was allowed to run for 2048 time steps. The time-history file output from the FDTD program was transformed to the frequency-domain, and the ellipticity was calculated from the transformed fields. The FDTD results, which are superimposed on a digitized version of Stewart et al.'s curve in Figure 9, shows good agreement with his integral equation results.

Next, a model was constructed with dimensions 12 m by 16 m by 16 m in order to validate the FDTD results against Anderson's results [32]. The cubic cells in the work-space again measured 0.10 m per side. The half-space boundary was again modeled in the $x - y$ plane at $z = 10$ m. The interface between layer 1 and layer 2 was modeled in the $x - y$ plane at $z = 8$ m, and the boundary between layer 2 and layer 3 was located at $z = 5$ m. The source VMD was placed at $x = 6$ m, $y = 6$ m, and $z = 11$ m, and the magnetic field components were calculated at $x = 6$ m, $y = 10$ m and $z = 11$ m (modeling transmitter and receiver antennas that are 4 m apart, and 1 m above the interface). The top and bottom layers were modeled with a resistivity of 10 ohm-m, and with a relative dielectric constant of 10, while the middle layer was modeled with a resistivity of 1000 ohm-m and a relative dielectric constant of 4. The ellipticity curves calculated from the FDTD output are superimposed with Anderson's digitized curves in Figure 10. Once again, they show good agreement.

Figures 9 and 10 represent a sample of the FDTD validation performed with TSAR. All the responses of the geometries presented in the Stewart et al. [30] and Anderson [32] articles that were evaluated using the integral equation implementation were satisfactorily duplicated using TSAR. A notable exception to this is the results of a 2 layer model presented by Anderson in which he varies the transmitter / receiver spacing to produce a family of curves. The FDTD method does not reproduce the 1 m separation curve well because of the dipole sources used in the FDTD method versus the finite-sized transmit and receive loop antennas used by Anderson.

16.2 System Design Using EM Modeling Codes

In order to help better understand geometrical-nulling systems, we ran many numerical simulations in order to investigate the relative sensitivities of each null method, sensitivities of other

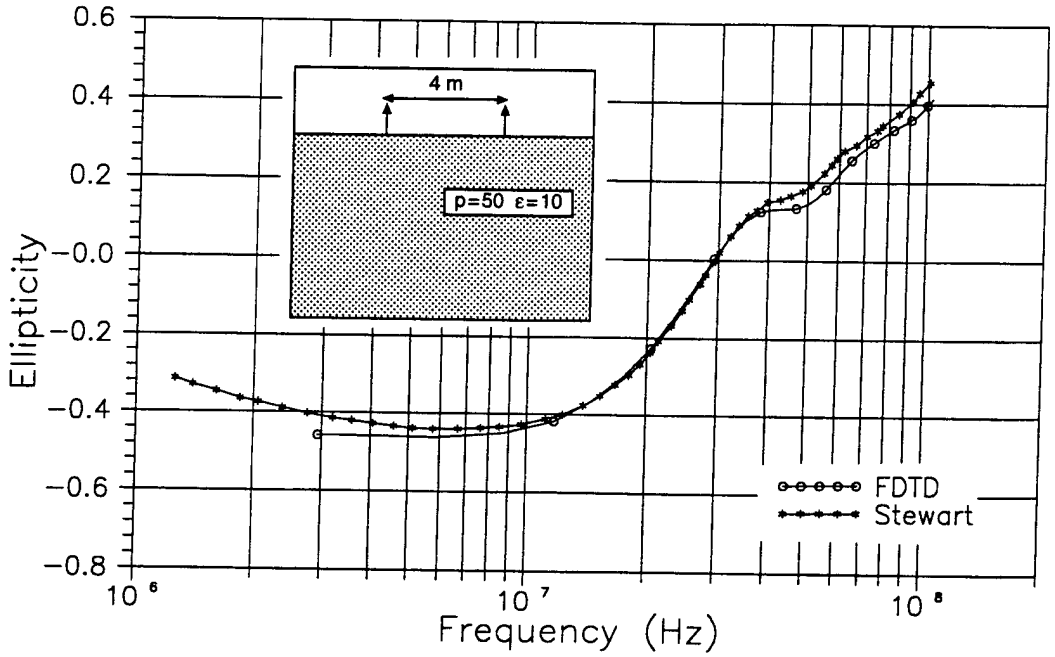


Figure 9: Ellipticity response of the FDTD calculations versus Stewart et al's solutions for loops placed over a homogeneous earth.

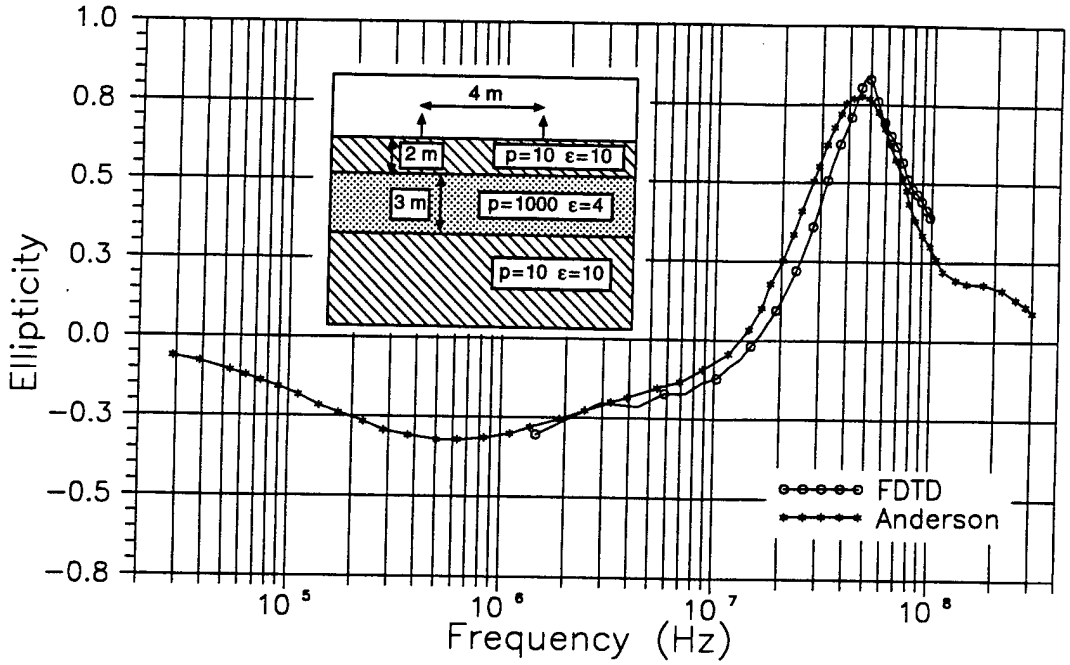


Figure 10: Ellipticity response of the FDTD calculations versus Anderson's solutions for loops placed over a 3 layer earth.

controlling parameters, and the expected responses of the null-coupled systems to shallow, perfectly-conducting targets. The three orthogonal null-antenna configurations shown in Figure 11 were analyzed using NEC4.1D. Orthogonal nulls are created when the receiver loop antenna is placed in a null of the transmitter loop antenna field pattern. The objective of this investigation was to find which nulling method is most exploitable, from a prototype design perspective, for geophysical target detection and characterization.

To test the sensitivity of the different null configurations, a target placed within the lossy half-space is needed. The first target we employed was a cylindrical, conductive pipe which was 40 cm long, 15 cm in diameter, and capped at both ends. To model this pipe using NEC, a cage, whose total surface area equaled the surface area of the cylinder, was constructed. The transmitter and receiver coils were modeled as 30 cm square loops. To mimic the effect of driving the transmitter antenna through a linear power amplifier having a 50 ohm output, the transmitter antenna model was loaded with a 50 ohm resistor. To mimic the impedance of the shunt capacitor tuned receiver antenna, the receiver antenna was loaded with a 1000 ohm resistor. To keep the modeled antennas as symmetric as possible (the symmetry of the antennas are important for the orthogonal null-configuration II, for example), the antennas were loaded opposite the feed and sensor points of the transmitter and receiver antennas, respectively. To illustrate, a schematic of the transmitter and receiver antenna models are shown in Figure 12.

With this NEC model, it was possible to calculate the coupling between the transmitter and receiver antennas when these antennas are positioned in different null configurations in free-space, above a lossy and dispersive homogeneous half-space, and when the half-space contains the cylindrical target. The antenna coupling was calculated as the ratio of the current magnitude induced on the receiver antenna, over the magnitude of the current injected into the transmitter antenna. This ratio was converted to a decibel scale and allowed the prediction of the anticipated antenna coupling for the different null configurations.

Table 1 presents the results of the NEC modeling for the antennas placed in the three orthogonal null-configurations. The first column of the table indicates the null configuration as defined in Figure 11. The second column is the free-space coupling of the antennas whose centers are separated by 80 cm. The third column reports the coupling levels when the antenna pair (the center of the loop antennas) is placed 35 cm over a lossy and dispersive half-space having a conductivity of 20 mS/m and a relative dielectric constant of 10. The fourth column indicates the change in coupling due to the presence of the half-space (that is, the difference between columns two and three). The fifth column gives the NEC results for the antenna coupling when the cylindrical target model is located at a depth of 50 cm below the interface. The axis of the target cylinder was chosen parallel to the line joining the antenna centers, and the target was located midway between the antenna centers. The sixth column in the table shows the difference in the antenna coupling caused by a parallel target versus a homogeneous half-space. The seventh column shows the results of the coupling calculations when the cylindrical target model was placed perpendicular to the line joining the antenna centers. Finally, the eighth column shows the difference caused by the perpendicular target relative to the homogeneous half-space.

From the fourth column of Table 1, it is seen that only configurations I and III are relatively insensitive to the presence of the half-space, while configuration II exhibits a large coupling due to the presence of the half-space. This effect can be explained using quasi-image theory.

Because of their lack of sensitivity to the half-space over which they are placed, configurations I and III demonstrate the best sensitivity towards the buried cylindrical target. Configurations

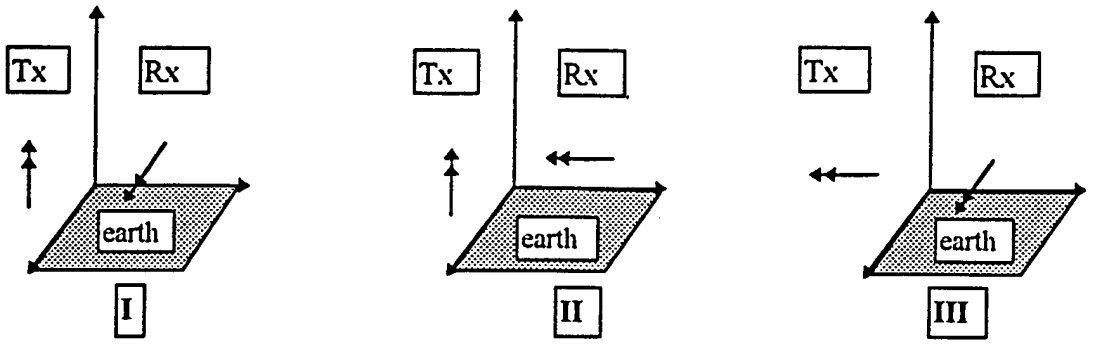


Figure 11: Magnetic dipole orthogonal null-configurations with numbering convention.

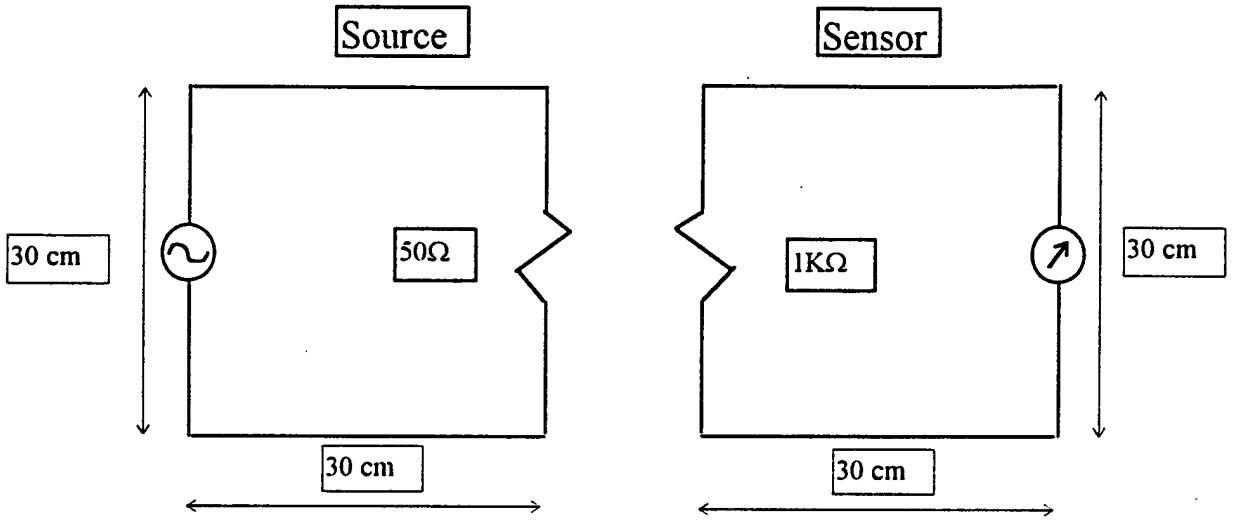


Figure 12: A schematic of the loaded antennas used in the sensitivity analysis model.

	Antenna coupling (dB)						
Config.	Free-space	Half-space	(Δ half -space)	Parallel Target	(Δ Para. target)	Perp. target	(Δ Perp. target)
I	-284.59	-285.56	-.97	-139.54	146.02	-153.83	131.73
II	-286.93	-81.70	205.24	-81.66	0.04	-81.54	.16
III	-315.71	-307.99	7.72	-152.25	155.74	-146.37	161.62

Table 1: A comparison of the depths of the orthogonal nulls, and their sensitivity to the presence of a half-space and of a buried target in parallel and perpendicular configuration.

I and III have about the same amount of sensitivity to the target. However, the magnetic fields generated with configurations I and III have different polarizations, and are expected to respond differently to targets which have different aspect ratios.

It is interesting to note that NEC calculated a peak coupling due to the perpendicular cylindrical target which was roughly 14 dB greater than that for the parallel cylindrical target for configuration I, while the parallel target gave a response about 6 dB greater than configuration III for the perpendicular target. This can be understood if Figure 11 is studied with the realization that the target provides the most coupling when its length is oriented in the direction of maximum electric-field generation (e.g. where more current can be induced) by the transmitter antenna. This implies that orientation information can be obtained from the response of targets with relatively large aspect ratios using null-coupled antenna systems of this kind.

The NEC algorithm also provided the ability to calculate the antenna coupling of the configurations as the antenna pair was moved over a survey line. Configuration I was chosen as the representative configuration for the survey analysis due to the results in Table 1. The antennas were modeled with the line segment joining their centers collinear with the survey line. The survey line data starts with the antenna mid-point (that is, the point between the centers of the transmitter and receiver antennas) -10 m away from the center of the target, with the horizontal transmitter antenna being furthest from the target. Data points were calculated every 0.5 m and the survey stopped when the antenna pair midpoint reached +10 m. The antenna coupling calculations for these surveys are presented in Figures 13 through 16. Figure 13 shows a family of curves for a survey calculated over parallel cylindrical targets buried at 0.25, 0.5, 0.75, and 1.0 m depths. Figure 14 shows a family of curves for surveys calculated with different antenna separation distances when

the parallel target was buried 0.5 m within the half-space. Related to Figures 13 and 14, Figures 15 and 16 show the same families of survey curves as Figures 13 and 14, but with the cylindrical target modeled perpendicular to the survey line.

The NEC model calculations show that the response of null configuration I when the target is perpendicular to the survey line is broader than when the target is parallel to the survey line. This implies that configuration I is more sensitive to a perpendicular target with a substantial aspect ratio than it is to the same target placed parallel to the survey line. This is in accordance with the conceptual view that the target will give the most response when aligned with its long axis perpendicular to the magnetic-field, thereby allowing more current to be induced, thereby yielding large secondary fields.

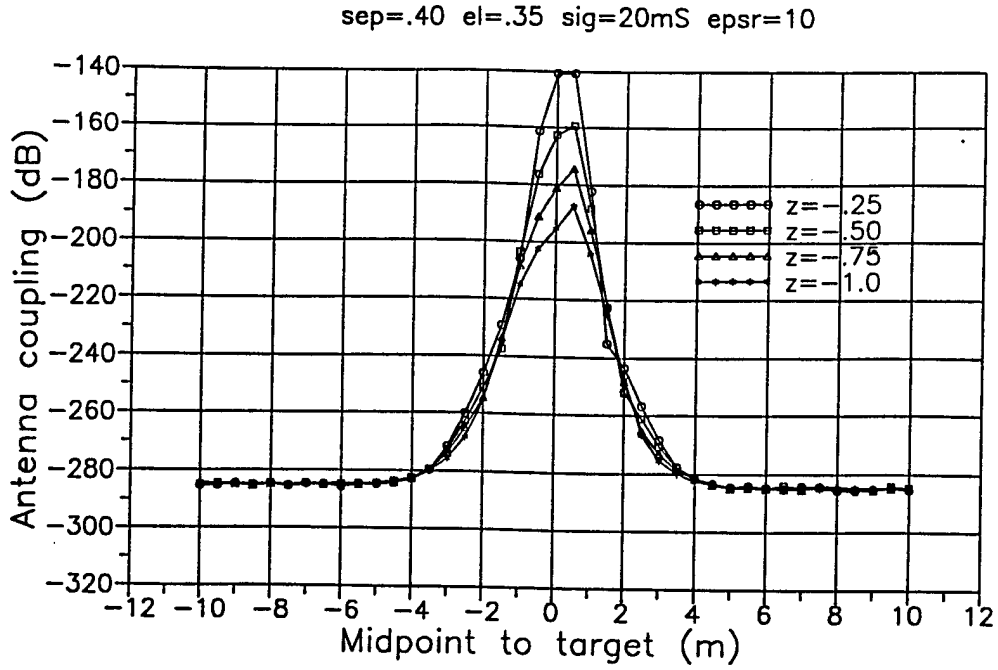


Figure 13: The calculated survey response of the configuration I null-coupled antenna system as it passes over a parallel cylindrical target buried to a 25, 50, 75, and 100 cm depth.

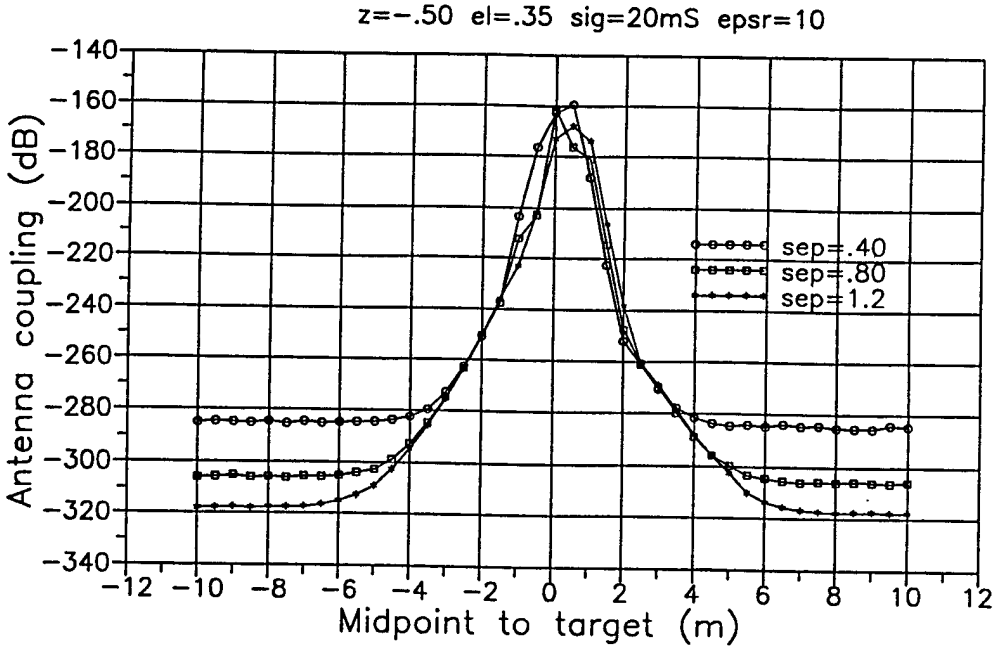


Figure 14: The calculated survey response of the configuration I null-coupled antenna system with antenna separation of 40, 80 , and 120 cm as it passes over a parallel cylindrical target buried to a 50 cm depth.

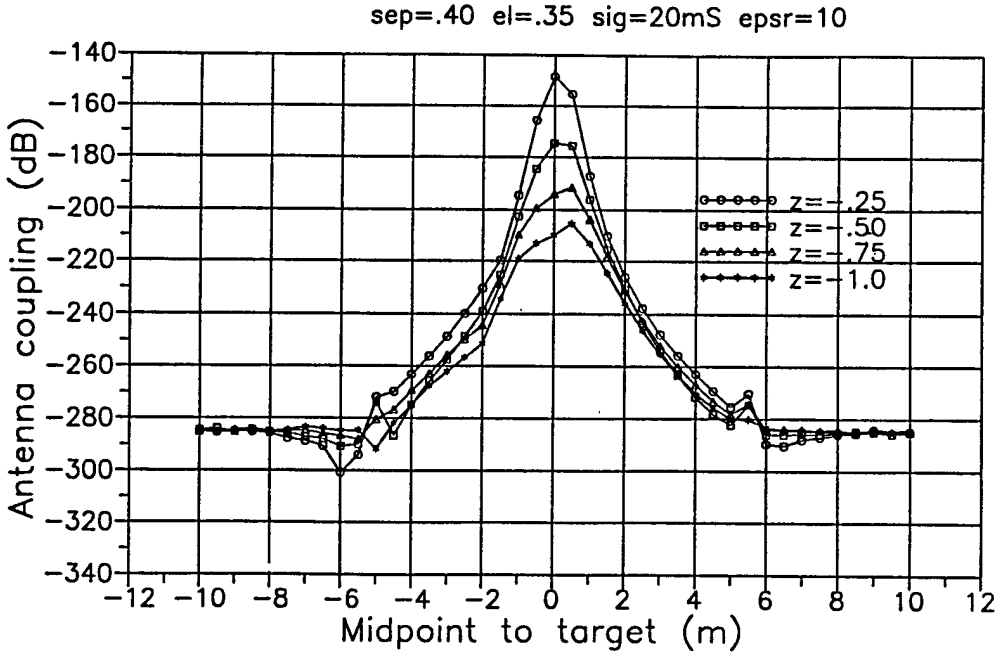


Figure 15: The calculated survey response of the configuration I null-coupled antenna system as it passes over a perpendicular cylindrical target buried to a 25, 50, 75, and 100 cm depth.

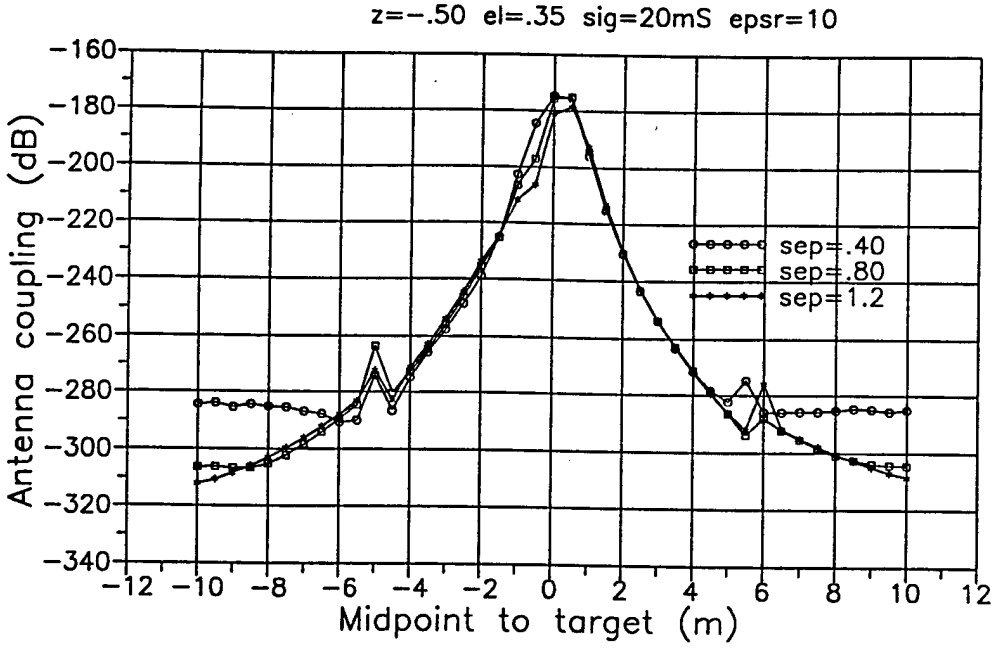


Figure 16: The calculated survey response of the configuration I null-coupled antenna system with antenna separation of 40, 80 , and 120 cm as it passes over a perpendicular cylindrical target buried to a 50 cm depth.