AD-A201 672

(2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

COMPUTER ALGORITHMS FOR MEASUREMENT
CONTROL AND SIGNAL PROCESSING
OF TRANSIENT SCATTERING SIGNATURES

by

Soonpuen Sompaee

September 1988

Thesis Advisor: Michael A. Morgan

Approved for public release; distribution is unlimited

88 12 27 107

Unclassified

security classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified	1b Restrictive Markings
2a Security Classification Authority	3 Distribution Availability of Report Approved for public release; distribution is unlimited.
2b Declassification Downgrading Schedule	
4 Performing Organization Report Number(s)	5 Monitoring Organization Report Number(s)
6a Name of Performing Organization Naval Postgraduate School — 6b Office Symbol *(if applicable)* 32	7a Name of Monitoring Organization Naval Postgraduate School
6c Address *(city, state, and ZIP code)* Monterey, CA 93943-5000	7b Address *(city, state, and ZIP code)* Monterey, CA 93943-5000
8a Name of Funding Sponsoring Organization — 8b Office Symbol *(if applicable)*	9 Procurement Instrument Identification Number
8c Address *(city, state, and ZIP code)*	10 Source of Funding Numbers Program Element No / Project No / Task No / Work Unit Accession No

11 Title *(include security classification)*
COMPUTER ALGORITHMS FOR MEASUREMENT
CONTROL AND SIGNAL PROCESSING
OF TRANSIENT SCATTERING SIGNATURES

12 Personal Author(s) Soonpuen Somapee

13a Type of Report	13b Time Covered	14 Date of Report *(year, month, day)*	15 Page Count
Master's Thesis	From To	September 1988	91

16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms *(continue on reverse if necessary and identify by block number)*
Field	Group	Subgroup	Radar Cross Section, Transient, Scattering, Signature.

19 Abstract *(continue on reverse if necessary and identify by block number)*

This thesis describes the development of computer algorithms for experimental measurement control and subsequent signal processing of transient signatures to synthesize scattering impulse responses of scale model targets. The theories behind transient scattering are considered in order to construct the algorithms. Up-to-date hardware and software technology are selectively implemented to optimize the resultant signal to noise ratio. The detailed explanation of the hardware and software operation is provided. A noise model for the system is also discussed. Measurement and programming validations are described, where comparisons are made with numerical computations for transient scattering by selected canonical targets. Results are documented for futher research in the transient scattering problem.

20 Distribution Availability of Abstract	21 Abstract Security Classification
☒ unclassified unlimited ☐ same as report ☐ DTIC users	Unclassified

22a Name of Responsible Individual	22b Telephone *(include Area code)*	22c Office Symbol
Michael A. Morgan	(408) 646-2677	62Mw

DD FORM 1473,84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete — security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

Computer Algorithms for Measurement
Control and Signal Processing
of Transient Scattering Signatures

by

Soonpuen Somapee
Lieutenant, Royal Thai Navy
B.S.E.E., Royal Thai Naval Academy, 1983

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1988

Author: ____________________
Soonpuen Somapee

Approved by: ____________________
Michael A. Morgan, Thesis Advisor

____
Richard W. Adler, Second Reader

____
John P. Powers, Chairman,
Department of Electrical and Computer Engineering

____
Gordon E. Schacher,
Dean of Science and Engineering

ABSTRACT

This thesis describes the development of computer algorithms for experimental measurement control and subsequent signal processing of transient signatures to synthesize scattering impulse responses of scale model targets. The theories behind transient scattering are considered in order to construct the algorithms. Up-to-date hardware and software technology are selectively implemented to optimize the resultant signal-to-noise ratio. The detailed explanation of the hardware and software operation is provided. A noise model for the system is also discussed. Measurement and programming validations are described, where comparisons are made with numerical computations for transient scattering by selected canonical targets. Results are documented for futher research in the transient scattering problem.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I wish to thank Professor Michael A. Morgan for his thoughtful and patient guidance throughout the course of this research.

This thesis is dedicated to my child and my wife [redacted] who for over two and a half years has kept the home fires burning while I have pursued my education. I could not have done it without her.

I. INTRODUCTION

A. BACKGROUND

In circuit theory, it is well known that the characteristics of a linear time-invariant system can be obtained by exciting the circuit with an impulse function. The theory of transient scattering measurements is analogous to this method, and it also provides a unique parametric description of the scatterer via the complex pole locations in the Laplace domain transfer function. There are two approaches to obtaining the impulse response. One utilizes frequency domain methods and the other employs time domain techniques. The former has to sweep frequencies over the entire band but the latter uses direct pulse excitation. As a result, the latter can save time and expense. Therefore, direct transient scattering measurements are a viable alternative to the more conventional CW method for obtaining impulse responses of targets.

In 1965 Kennaugh and Moffatt [Ref. 1] investigated the properties of the impulse response waveforms backscattered from metallic targets. The approximations for transient scattering and impulse responses they made formed the basis for "ramp-response" imaging.

In 1967 DeLorenzo [Ref. 2] presented a system that could be used to obtain an approximate measurement of the impulse response of scattering bodies. His major development was an indoor laboratory scattering range which facilitated the time domain analysis. He showed the agreement between his calculations and the measurement of the impulse response of a rectangular plate.

Also in 1967, Murchison and Falk presented a hybrid system for measurement of impulse responses [Ref. 3]. This measurement system was separated into two frequency ranges. For frequencies below 4 Ghz, they used an impulse generator while for the frequencies above 4 Ghz, they used a swept frequency CW source.

In 1971 Baum [Ref. 4] demonstrated theoretically that characteristic resonances can be obtained for scattering objects. His technique was called the Singularity Expansion Method (SEM). It was postulated that the resonances could be obtained from the transient scattering measurement for a complex object.

In 1973 Tesche [Ref. 5] applied the SEM to to numerically evaluate complex resonances of simple metallic scattering objects.

In 197ᶠ Blaricum and Mittra [Ref. 6] proposed a technique for extracting the poles and residues of a system directly from its transient response. Their technique provided a number of numerical advantages.

In 1980 the original Transient Electromagnetic Scattering Laboratory (TESL) became operational at the Naval Postgraduate School (NPS). Hammond [Ref. 7] developed this laboratory for inverse scattering research based upon synthesized ramp responses. This laboratory utilized an outside ground-plane range. This range was limited to symmetric scattering objects which could be "mirror imaged".

In 1983 the free-field TESL began operation at NPS [Ref. 8]. This laboratory is located in Spanagel 535. The TESL facilitated research efforts in high quality transient scattering measurements [Ref. 9] used for radar target identification and characterization using complex pole natural resonances [Ref. 10].

Prior to the current undertaking, McDaniel [Ref. 11] showed good agreement between the computation and the measurement of the impulse responses for a sphere and a thin wire. These measurements and signal processing were performed using the Tektronix hardware set up, as originally employed in the 1980 TESL. In 1987 a revolutionary new Hewlett Packard digital processing oscilloscope became available. In addition, computational hardware had evolved dramatically since the early 1980's. This thesis describes the implementation of this new technology to enhance the performance of transient scattering and signal processing at the NPS TESL.

B. OVERVIEW OF THIS EFFORT

The main objective of this thesis was to incorporate new measurement and computational hardware and to design and implement new software to automate high-quality transient scattering measurements. In this thesis, much work has been done in the development of the transient scattering work station which has the ability to measure and synthesize the scattering impulse response of scale model targets. The major hardware includes a microcomputer, a digital programable oscilloscope (DPO) and a shielded anechoic chamber. Automatic transient scattering measurements have been implemented by utilizing an IEEE-bus controller.

With the new HP54120T DPO, the impulse responses of scale models are synthesized at a high signal-to-noise ratio (SNR). The acquisition and signal processing algorithms run much faster on the IBM PC-AT vis-a-vis the Tektronix interpreted BASIC computer previously used. The programs now use QUICK BASIC and FORTRAN compilers. Validations of software were made by way of comparing experimental measurements with computations for simple canonical targets. Various scale models were measured and documented in detail. Furthermore, a noise calculation was provided for further research.

This thesis is broken into six chapters. Chapter II describes system facilities and hardware. This includes laboratory description, anechoic chamber, HP54120T digital processing oscilloscope and the GURU II hardware interface.

Chapter III presents the theory of transient scattering measurements. This is composed of the free field range system representation, mathematical model and transient response solution.

Chapter IV deals with acquisition algorithms and signal processing. This chapter covers operational description, GURU II software interface and the deconvolution algorithm. In addition, some selected outputs are illustrated.

Chapter V emphasizes the experimental results. Two types of targets are presented. One is a canonical target and the other is a scale model target. The noise estimation of the system is also discussed in this chapter.

Chapter VI summarizes all work and provides some future considerations.

II. SYSTEM FACILITIES AND HARDWARE

A. LABORATORY DESCRIPTION

Figure 1 on page 5 shows the general layout of the scattering range. It is composed of

- Digital Programmable Oscilloscope (DPO)
- Step pulse generator
- Low loss cables
- Broad band solid state power amplifier
- Broad band transmitting and receiving antennas
- Anechoic chamber
- Personal computer (PC).

Note that the DPO includes the sampling and pulse generator unit. The DPO is used as both the master pulse source and as the receiver. The antenna connection cables were obtained from the DPO accessory kit for use in low loss network measurements. They are flexible for hard-to-access areas. The characteristics are shown in Table 21 and Table 22 on page 78 in Appendix C.

Transmitting and receiving antennas are of the double-ridged horn type. Their frequency range is 1-12.4 GHz, with less than 1dB of ripple in the boresight gain over this multi-octive band. The Avantek solid state power amplifier has an advertised 3dB bandwidth of 2 to 6 GHz but, by our observations, has a usable bandwidth of 1 to 7 GHz. A 15V supply is needed for the amplifier.

The anechoic chamber serves two purposes. First, it is used to isolate the target signal from the electromagnetic noise and interference outside of the chamber. Secondly, the absorbing material on the walls, floor and ceiling of the chamber are designed to provide minimal reflections, thus approximating unbounded space. The computer, which serves both acquisition and signal processing rolls, is an IBM-AT with 640 K RAM, a 20 megabyte hard disk, and a GPIB-PC2A interface card. The GPIB-PC2A was added to the PC in order to have the ability to talk or listen to the HP54120T DPO. The PC was set to run at 9 MHz. It has two floppy disk drives, and two output ports. One is a parallel port for the printer and the other is a serial port for the plotter. The color graphic board is NEC GB1. The display is NEC EGA color monitor.

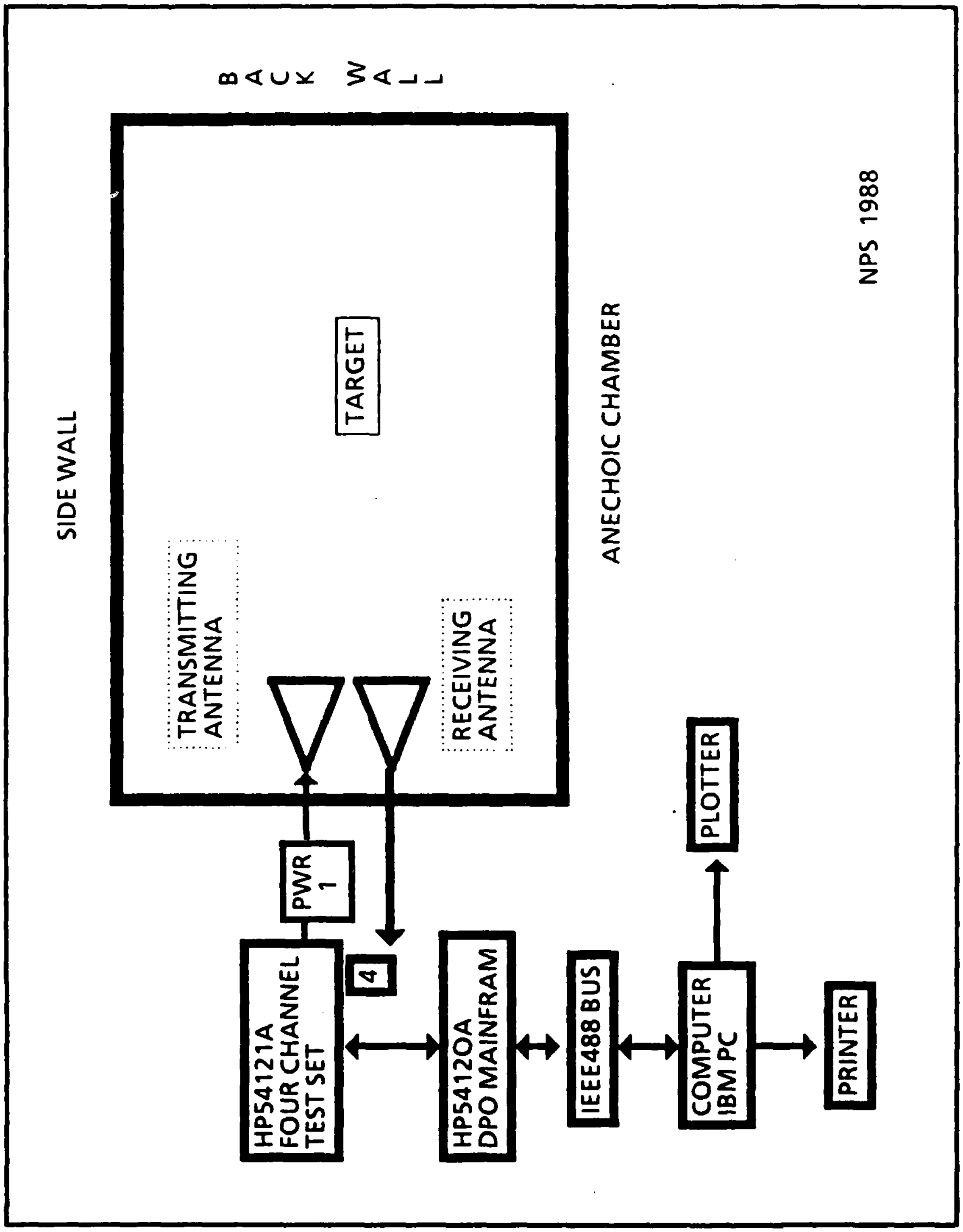


Figure 1. General layout of the scattering range

B. ANECHOIC CHAMBER

The anechoic chamber is located in room 535 of Spanagel Hall. This chamber incorporates a metallically shielded anechoic enclosure to minimize outside EM interference. Chamber dimensions are 20 ft. long with a 10 ft. width and 10 ft. height. Because targets are suspended within the chamber, there is no requirement that they have mirror symmetry, as is the case for ground plane based ranges. In the measurements that were taken for this thesis, the target is mounted on a thin styrofoam column at a distance of 8.5 feet from the transmitting and receiving antennas. Both receiving and transmitting antennas are placed close together to closely approximate monostatic measurements. The chamber interior is covered by a special carbon impregnated foam absorbing material. The material is cut into longitudinal wedges along the side-walls, floor and ceiling to act as conduits of EM energy towards the back wall. The back wall is covered by 18 inch long pyramids to act as slow absorbing transitions for incident fields. Further information about this free-field TESL can be found from Refs. 9, 11 and 12

C. HP54120T DIGITAL PROCESSING OSCILLOSCOPE

The Hewlett Packard 54120T is a fully programmable 20 GHz, four channel, digitizing oscilloscope with a nine inch color display. It is capable of automated measurements, digital storage, and TDR measurements. This DPO has only been in production since mid-1987 and represents the state-of-the-art in sampling technology. Previous to this thesis effort, a 6 year old 12 GHz DPO from Tektronix was used to perform transient scattering measurements in the TESL.

The HP-DPO is comprised of two major components. These are the HP54120A Digitizing Oscilloscope Mainframe and the HP 54121A Four Channel Test Set. It also contains extensive self-tests to ensure proper functioning. Both HP 54120A and HP 54121A were calibrated together as a system called the HP 54120T. The DPO can be remotely programmed by an HP-IB (IEEE-488) controller interface. The HP-IB complies with the IEEE 488.2 standard. The essential characteristics of the HP 54120T are summarized in Table 1 and Table 2 on page 7.

It should be noted that the HP 54121A 4 Channel Test Set is sensitive to electrostatic discharge (ESD). Precautions should be taken to keep ESD from damaging the equipment. Details can be found in Ref. 13.

Waveforms that are stored in the DPO memory can be sent to either an HP Thinkjet printer or an HP Plotter, both of which are accessed via the HP-IB bus. There are two ways to operate the instrument. One is manual operation via the front panel and the

Table 1. HP54120T CHANNEL CHARACTERISTICS

Bandwidth	20/12.4 GHz
Number of channels	4
Full scale	8 division
Maximum scale	80 mV/division
Minimum scale	1 mV/division
Programable dc offset	± 500 mV
Input maximum	± 2V dc + ac peak
Norminal impedance	50 ohm
Connectors	3.5 mm
Percent reflection	less than 5% for 30 ps risetime

Table 2. HP54120T TDR AND TIMEBASE CHARACTERISTICS

Risetime	less than 45 psec at 12.4 GHz bandwidth in average mode
Timebase full-scale	10 division
Minimum scale factor	10 psec/div
Maximum scale factor	1sec/div
Time interval accuracy	less than 10 psec ± 0.1 % of reading

other is to operate by sending HP-IB bus commands via the computer. A major effort in this thesis was developing programs to provide easy operator control and signal acquisition of the DPO using the computer. These algorithms are explained in detail in chapter IV.

Ensemble waveform averaging has been implemented in acquiring scattering signatures. The averaging reduces noise by $1/\sqrt{N}$, where N is the number of averages. The minimum discernible signal level in the digital processing oscilloscope system is approximately $35\mu V$. Typical noise amplitudes for the DPO sampling unit are shown in Table 3 on page 8.

Data display resolution is 1024 points horizontally × 256 points vertically (for display times of at least 200 ps/div). The display can be selected in persistence or average

Table 3. TYPICAL NOISE

20 GHz bandwidth, Avg = 1	Noise 1.2 mV (RMS)
20 GHz bandwidth, Avg = 256	Noise $80\mu V$ (RMS)
12.4 GHz bandwidth, Avg = 1	Noise $500\mu V$ (RMS)
12.4 GHz bandwidth, Avg = 256	Noise $35\mu V$ (RMS)
12.4 GHz bandwidth, persistence	Noise $400\mu V$ (RMS)

mode and the number of averages can be specified as a power of 2, up to 2048. The data output transfer rate and the data record lengths are shown in the following table.

Table 4. DATA OUTPUT TRANSFER RATE AND DATA RECORD LENGTHS

10 psec/div ≤ time/div < 20 psec/div	100, or 400 points/record
20 psec/div ≤ time/div < 50 psec/div	100, 400 or 800 points/record
50 psec/div ≤ time/div < 200 psec/div	100, 500 or 1000 points/record
200 psec/div ≤ time/div < 1 sec/div	128, 256, 500, 512 or 1024 points/record
Data output transfer rate	115 kbytes/s

For the time domain reflectometer (TDR) system, the risetime is an adjustable value based upon the timebase setting. The minimum is 10 ps or .08× time/div, whichever is greater. The maximum is 5 × time/div. Usually channel 1 is used for reflection measurements, however, in this thesis, it was used as the source generator.

The digitizer uses a 12-bit successive approximation A/D converter. The full-scale range of the A/D is 640 mV. The least significant bit (LSB) of the A/D converter equals 250 μV. This gives one part in 2560, or more than 11 bits of resolution. Averaging can stretch out the resolution to 32 μV. This improves the resolution to around 14 bits.

The signal is sampled and digitized at a rate determined by the trigger rate, repetition rate, time base range, display mode, and number of channels turned on. Further information concerning the characteristics of the HP 54120T can be found from Ref. 14.

D. GURU II HARDWARE

The Textronix supplied General-purpose interface bus User's Resource Utility (GURU II) is a combination hardware and software package that allows the use of any of the IBM personal computers (PC), and some PC compatibles, to control GPIB programable instruments. In this section, the focus is on the hardware. The software is explored in the next chapter.

The hardware consists of the PC2A GPIB interface card that plugs into the PC main board, and the GPIB cable. The GPIB PC2A interface card is an 8-bit half-size board which performs as a talker, listener and controller. It also can be implemented with the full range of talker, listener, serial poll, service request, and remote programming functions for as many as 14 devices on the bus. It can perform as a complete controller. Using the PC2A, data transfer between the system memory and the GPIB are performed at rates of more than 300 kilobytes per second. The PC2A is compatible with all revision levels of the IEEE-488 standard including the HP-IB.

The GPIB-PC2A interface card consists of these major sections:

- address decoding
- buffering and data routing
- interrupt arbitration
- direct memory access (DMA) arbitation.
- configuration switches and jumpers
- GPIB-adapter-TLC (talker/listener/controller)
- time of day clock with battery back-up.

The address decoding monitors the address lines to recognize when the GPIB-PC I/O address is present on the computer I/O channel and enables read and write access to the GPIB adapter register. The buffering and data routing handles data transfer between the IBM PC-AT I/O channel and the GPIB adapter through a bidirectional internal data bus.

Interrupt arbitation recognizes when interrupts have been enabled or disabled and passes or inhibits them accordingly. DMA arbitation recognizes when DMA operations are enabled or disabled, and when the last transfer has taken place. It also routes the DMA request and acknowledges signals to the selected DMA channel. The DMA is used to transfer data directly between a peripheral and memory without CPU.

The IEEE-bus controller forms an interface between the computer and the IEEE-488 bus. Conceptually, the IEEE-488 bus behaves in a way very similar to the system bus. Hewlett-Packard originally devised it [Ref. 15] and intended it to link together programmable instruments in a laboratory or industrial environment. By controlling test equipment and measuring devices from the IEEE-488 bus, implementing an automatic testing station is possible. A system under test is connected to the test equipment and measuring devices. The computer configures all equipment via the IEEE-488 bus and then reads the test results from the same bus. The GPIB has the following limitations:

- Half of the devices attached to the GPIB must have power on.
- The total length of the bus cable must not exceed 20 Meters.
- No more than 15 devices may be attached to one bus.

More detailed information can be found in Ref. 16.

III. THEORY OF TRANSIENT SCATTERING MEASUREMENTS

A. FREE-FIELD RANGE SYSTEM REPRESENTATION

The physical model of the system free field scattering range was first presented by Morgan in Ref. 12. The system block diagram is shown in Figure 2 on page 12. Important components and parameters of the system are summarized in the frequency domain representation in the following table.

Table 5. FREQUENCY DOMAIN FUNCTIONS

Pulse generator, driving the transmitting antenna	$X(f)$
Transmitting antenna response	$H_T(f)$
Receiving antenna response	$H_R(f)$
The multiple scattering between target and chamber wall	$H_S(f) \leftarrow \rightarrow H_C$
Direct couple of receiving and transmitting antenna	$H_A(f)$
The multiple scattering between calibration sphere and absorber	$H^C_{SC}(f)$
The multiple scattering between target and absorber	$H^T_{SC}(f)$
The calibration sphere transfer function	$H^C_S(f)$
The target transfer function	$H^T_S(f)$
Thermal noise	$N_n(f)$

The transient pulse source is the step generator in the TDR system. This generator not only supplies the low-level transmitted pulse, before amplification, but also acts to coherently trigger the sampling circuits of the DPO receiver. It thus is akin to the "master oscillator" in a coherent pulsed CW radar system. The risetime of the step generator is approximately 25 psec. This step pulse is first amplified and filtered by a broad-band GaAs FET amplifier.

The amplified pulse, which no longer resembles a step waveform, is then fed through the transmitting antenna which radiates into the anechoic chamber. The radiated field emanating from the antenna is approximately an angular weighted spherical wave. At the target position the incident field is a quasi-plane wave. Because of the close proximity of the transmitting and receiving antennas, some strong coupling of energy is induced in the receiving antenna.

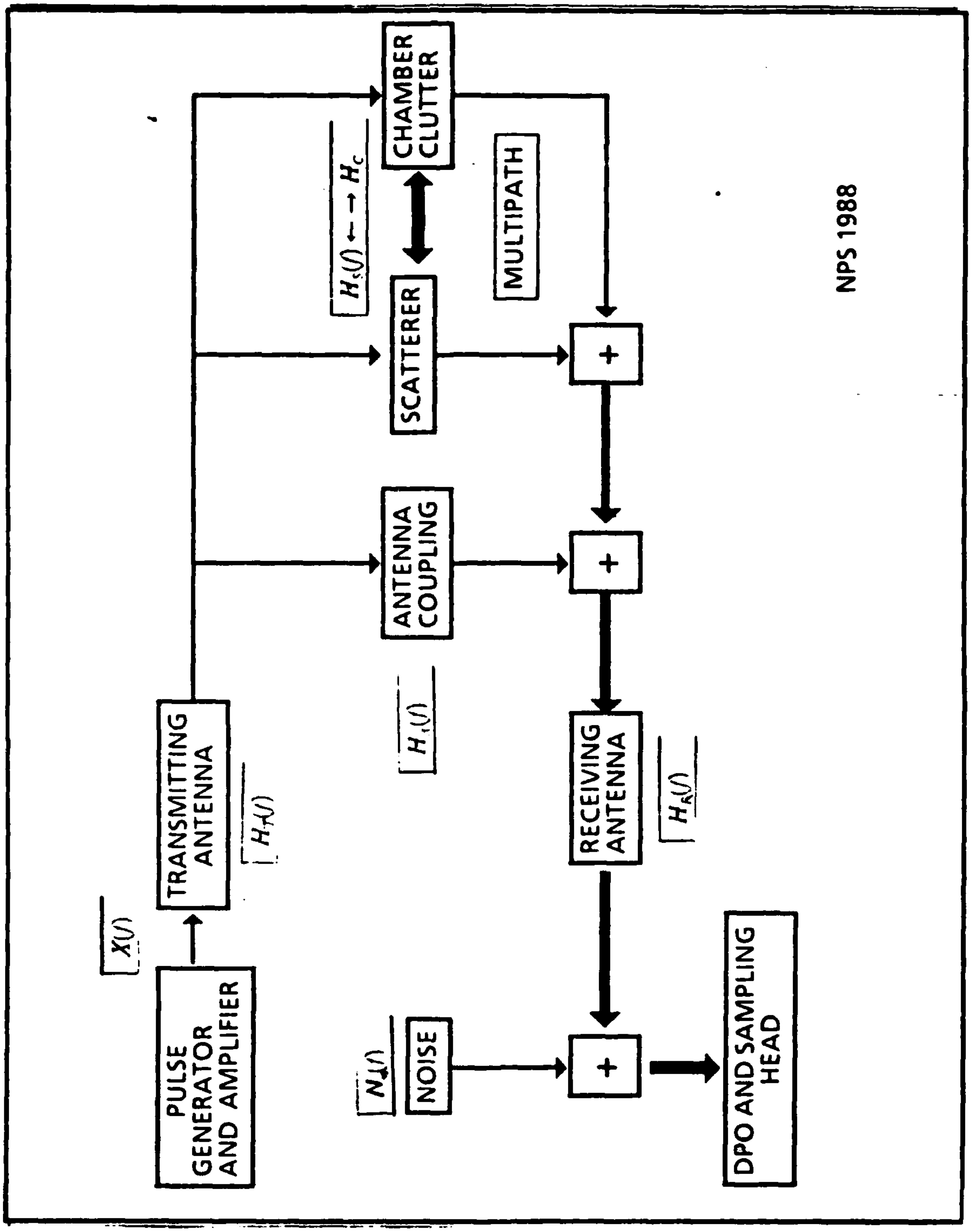


Figure 2. System representation of the TESL

This undesirable signal can be seen clearly if the time delay setting on the DPO is adjusted to begin many nanoseconds before the target response time-window. Therefore, the time setting delay must be checked when the target is present.

The initial calibration of the beginning of the target time-window was performed by using a rectangular copper plate, 1 ft wide and 1.5 ft high, as the test scatterer. The back scattering was observed with the copper plate turned broadside to the centroid of the horn positions and placed where the target would be mounted. The copper plate scattering signature was so large that it could be clearly discerned. The results of this measurement indicated a DPO time delay setting of 38 nsec was needed.

Multiple scattering between the target and the chamber absorber is represented by the two way interaction arrow connecting $H_C(f)$ to $H_S(f)$. The total noise of the system $N(f)$, is due to both thermal emission from the chamber and receiver noise in the sampling front end of the DPO. These sources are combined together and denoted as $N_n(f)$ for the n-th measurement that is made. To understand the theory of transient scattering measurements adequately, the mathematical model will be presented in the next section.

B. MATHEMATICAL MODEL

The five key parameters in Table 6 on page 14 play a major role in both the physical meaning and mathematical sense of the acquisition and signal processing algorithms. These five parameters can be related to the system representation, described in the previous chapter, by the following three equations:

$$Y_1(f) = H_T(f)\, H_R(f)\, X(f) \left[H_A(f) + H_C(f) + H_S^C(f) + H_{SC}^C(f)\right] + N_1(f) \tag{3.1}$$

$$Y_2(f) = H_T(f)\, H_R(f)\, X(f) \left[H_A(f) + H_C(f)\right] + N_2(f) \tag{3.2}$$

$$Y_3(f) = H_T(f)\, H_R(f)\, X(f) \left[H_A(f) + H_C(f) + H_S^T(f) + H_{SC}^T(f)\right] + N_3(f). \tag{3.3}$$

These three equations represent the three respective measurements that are necessary for processing the scattering signature of a given target. As shown below, these measurements are for a calibration sphere, whose measured scattering can be compared to an accurate calculation; the "background" which denotes no target present, and the actual target scattering measurement.

Equation 3.1 states that the response to the amplified pulse generator, X(f), due to the calibration sphere and the chamber is equal to the product of the pulse excitation, transmitting antenna transfer function $H_T(f)$, receiving antenna transfer function $H_R(f)$

Table 6. MATHEMATICAL NOTATION

The calibration measurement waveform	$Y_1(f)$
The background measurement waveform	$Y_2(f)$
The target measurement waveform	$Y_3(f)$
The subtracted calibration waveform	$Y_4(f)$
The subtracted target waveform	$Y_5(f)$

times the sum of the antenna coupling $H_A(f)$ plus the calibration sphere $H_S^c(f)$ and the multipath multiple scattering $H_{SC}^c(f)$. Added to this product is the thermal noise $N_n(f)$.

Equation 3.2 represents the background measurement, with no target present. The system response is similar to that previously considered, but with the absence of the transfer function for the calibration sphere. Equation 3.3 expresses the target measurement in the frequency domain. The intepretation follows the same pattern as equation 3.1 except the target transfer function, $H_S^T(f)$, appears instead of that for the calibration sphere. The multipath multiple scattering in equation 3.3 also changes and is denoted by $H_{SC}^T(f)$.

The three equations as stated above are three crucial measurements in the transient scattering problem. The backscatter measurement with the canonical (calibration) object $Y_1(f)$, measurements without target (background) $Y_2(f)$ and measurements with the desired target $Y_3(f)$ are obtained directly using the time domain acquisition program in Appendix B. This program was written in BASIC as part of this thesis effort. To speed up the execution of the program by a factor of about fifty times vis-a-vis direct interpretation, it has been compiled using the Microsoft Quick Basic Compiler.

After acquiring accurate data, the next step is to process the data and extract the impulse response. This is the topic of the next section.

C. TRANSIENT RESPONSE SOLUTION

The three measurements that were just considered form the basis for estimating the "smoothed" impulse response of the target. The band-limited nature of the measurements does not allow synthesis of the true infinite bandwidth impulse response which is due to an incident delta function plane wave. Instead, the "smoothed" term means that we can estimate the transient target response due to some predefined incident signal (a double-Gaussian shaped waveform is used here) whose significant spectral content does not exceed the bandwidth of the measurements. Aside from the currently used solid-state power amplifier, the bandpass range of the DPO-antenna system is about 1 GHz to 12 GHz. The spectral bottleneck is due to the GaAS FET amplifier, whose passband is about 1 GHz to 7 GHz.

In order to synthesize the smoothed transfer function of the target $H_S^T(f)$, the first step is to subtract the "clutter" portion of the responses in equations 3.1 and 3.3 which is embodied in equation 3.2. This subtraction results in equations 3.4 and 3.5 below:

$$Y_4(f) = Y_1(f) - Y_2(f) \tag{3.4a}$$

$$Y_4(f) = H_T(f)\, H_R(f)\, X(f) \left[H_S^C(f) + H_{SC}^C(f)\right] + N_4(f) \tag{3.4b}$$

$$N_4(f) = N_1(f) - N_2(f) \tag{3.4c}$$

$$Y_5(f) = Y_3(f) - Y_2(f) \tag{3.5a}$$

$$Y_5(f) = H_T(f)\, H_R(f)\, X(f) \left[H_S^T(f) + H_{SC}^T(f)\right] + N_5(f) \tag{3.5b}$$

$$N_5(f) = N_3(f) - N_2(f). \tag{3.5c}$$

Equations 3.4 and 3.5 are the very critical steps in obtaining the smoothed transient response of the target. The critical nature of the subtractions has to do with coherency in time of the subtracted clutter. The calibration sphere measurement $Y_1(f)$ had the background measurement $Y_2(f)$ subtracted in order to eliminate the clutter. The clutter signal is that which is present aside from the target and is composed of two main constituents. One is the direct coupling of the receiving and transmitting antennas, $H_A(f)$, and the other is direct scattering (as opposed to multipath) from the absorber in the chamber, $H_C(f)$. Since the measurements are made directly in the time domain, the subtractions are done in time domain. In a similar fashion, equation 3.5 can be obtained.

The results, $Y_4(t)$ and $Y_5(t)$, are then transformed into frequency domain by discrete fast Fourier transform (FFT). This leads to the next step, which is to form an optimal deconvolution estimator for target response given by

$$X_O(f)\, H_S^T(f) = \frac{Y_5(f)\, Y_4^*(f)\, Y_{SC}(f)}{|Y_4(f)|^2 + C} \tag{3.6}$$

where $Y_4^*(f)$ is the complex conjugate of $Y_4(f)$, $Y_{SC}(f)$ is the computed calibration sphere and C is the smoothing parameter.

$$Y_{SC}(f) = X_O(f)\, H_S^C(f) \tag{3.7}$$

$$x_O(t) \leftarrow FFT \rightarrow X_O(f) \tag{3.8}$$

$$x_O(t) = C_1\, e^{-\alpha_1 (t-t_o)^2} + C_2\, e^{-\alpha_2 (t-t_o)^2}, \tag{3.9}$$

where, for a specified α_1 and α_2 in this "Double-Gaussian" (DG) pulse, the C_1 and C_2 are chosen such that

$$x_o(t_o) = 1 \tag{3.10}$$

and

$$\int_{-\infty}^{+\infty} x_o(t)dt = 0. \tag{3.11}$$

Each of the two Gaussian waveform in equation 3.9 will be specified here by a "10% pulse width", defined by Δt_k such that

$$e^{-\alpha_k \frac{(\Delta t_k)^2}{4}} = 0.1, \tag{3.12a}$$

which gives

$$\alpha_k = \frac{4}{(\Delta t_k)^2} \ln(10) = \frac{9.21}{(\Delta t_k)^2}. \tag{3.12b}$$

The estimator in equation 3.6 is due to Riad [Ref. 17] and results from an optimal least-square fidelity criterion for deconvolved target response. The result of equation 3.6, $X_O(f)\, H_S^T$, represents the frequency domain scattering response of the measured target to an incident plane wave whose Fourier transform is $X_O(f)$. This incident waveform is specified by numerically computing the scattering response of the calibration sphere to it. This numerically generated response of the calibration sphere is embodied in $Y_{SC}(f)$, as shown in equation 3.7. It is obtained from the SPRSCT program which uses the magnitude and phase data from the MIE series program. Both of these programs were written by Morgan.

Both subtraction and optimal deconvolution algorithms are implemented in the Deconvolution Program in Appendix A. The procedure to operate this program will be explained in chapter IV.

IV. ACQUISITION AND SIGNAL PROCESSING ALGORITHMS

A. GURU II INTERFACE

1. GPIB Principal

This section emphasizes the GURU II software interface. In order to understand the Acquisition program operation, the GPIB principle has to be understood. The General Purpose Interface Bus uses a twenty-four conductor cable. Sixteen of these are used to send the signal. Data is sent back and fourth between the listener and the talker in a bi-directional fashion. This means that both transmitted and received data travel on the same line. Eight lines are used for 8 bits of parallel data transfer. A byte is sent by serial mode but each bit is sent in parallel mode. Three lines act as the coodinator between the talker and the listener. These lines are assigned their duty as follows:

- no data available line
- data not ready line
- data not accepted line.

The talker on the bus controls the data available on the line and waits for the listener's response on the "not ready for data" line or "data not accepted" line. The remaining five lines are the attention line, the interface clear line, the remote enable line, the service request line and the end or identify line. When the attention line is active, the talker and the listener will be specified. The controller has full access to five of these lines and has the capability to directly address other devices to command them to talk or listen. The interface clear line is used to set the turn-on condition. When the remote enable line is asserted, all devices on the bus are under computer control and nothing external can interfere. The service request line is used by a device to indicate a requirement for attention and to request an interruption of the current sequence of events. There are two methods for the controller to determine the device and the message. One is the serial poll while the other is the parallel poll. The polls obtain numerical codes which show a specific event. The serial poll is a mechanism that the controller uses to obtain status from individual devices. As each device is polled, it returns a status byte which controller decodes to determine the device requesting service. The parallel poll is a mechanism for accepting and decoding the general status of as many as eight devices simultaneously. The end or identify line is used by the talker to indicate the end of the message.

2. Basic Software Requirements

There are 3 files which the GURU requires in order to run the application program. These files are

- GPIB.COM
- IBCONF.EXE
- CBIB.OBJ.

The primary GPIB device drivers are read in via the CONFIG.SYS file at boot-up time. This file is in the root directory and must include the following DOS statement:

- DEVICE=GPIB.COM.

This causes the GPIB driver in GPIB.COM to be loaded when the system is booted.

The IBCONF.EXE is the executable program to configure the GPIB.COM file containing the device drivers for each instrument. HPDEV is the name of the device driver for all Hewlet-Packard instruments, including the HP 54120T, whose decimal primary address is 17. The IBCONF.EXE (Interface Bus Configure) program allows a match between the softwere and the hardware. The program prompts with menus of details about the hardware parameters associated with the interface card. This device for Hewlett-Packard is already configured in the system. When the IBCONF.EXE is run, the device drivers are mapped for each instrument in the display. By selecting the HPDEV, and editing it, the characteristics of the HPDEV show on the screen. This allows the operator to change the characteristics to be compatible with the hardware. After reconfiguration of the GPIB.COM file, the operator must return to the operating system and then re-boot the PC for the new parameters to be in effect.

CBIB.OBJ is the object file which is linked with any compiled BASIC program which needs to interface with the GPIB.COM device drivers. In our case, the Acquisition Program, written in GW-BASIC, is first compiled using the Microsoft QUICK BASIC compiler. This process creates an object (.OBJ) file. The CBIB.OBJ object file is then linked with the Acquisition object file. The final result is an executable (.EXE) file.

B. ACQUISITION ALGORITHM

The Acquisition program was written in Microsoft GW-BASIC. The hardware is interfaced with this program by way of the CBIB object file. This file comes with the

GURU II software. After compiling the source code, the object file of the Acquisition has to be linked to the CBIB object file in order to complete the creation of the Acquisition executable file. To run the program, the Brun3087 execution file is needed in the same DOS Directory. This file is supplied with Microsoft QUICK BASIC. The Acquisition program uses a friendly and interactive menu format. The program begins with a displayed welcome message, including the time and date. This screen also provides a brief procedure on how to measure the transient scattering response. The screen gives operator the choice of using the system default or the last response. The system default was designed for first time operators. The last response was designed for experienced operators. A main menu is displayed on the screen after the operator decides whether to use the system default or the last response.

The main menu contains many importance parameters. A first item is the number of data points in the acquired time series. This number has to be a power of 2, but not more than 1024. This limitation could be modified in the program by defining a larger array size up to 2048 points. The second item is the number of sub-averages. This parameter determines the number of values to average for each time point when the HP54120T is in the average mode. This parameter has to be an integer from 1 to 2048, also in a power of 2. The third item is the number of data blocks. This determines the number of sets of data in one record. The fourth and the fifth items are a user supplier identification for the target and the date of measurement. The sixth item tells the operator about the typical sampling time interval. The seventh parameter is the type of the waveform being acquired on the measurement being made. As discussed in the previous chapter, there are three scattering waveforms which need to be acquired for each target being considered: calibration (sphere), background and actual target. The eighth item tells the operator the time window in nanoseconds. The ninth, tenth and eleventh items are the maximum vertical setting, the data file name output and the auto time setting. At the bottom of the menu, the operator is asked if he wants to change the items.

The second menu appears on the screen when the operator decides to change some item on the main menu. This "overlay" menu contains all changeable parameters and allows a change to these parameters interactively. When the changes are completed, the operator can go back to the main menu by typing the letter "R". This brings the operator to the main menu and displays the new item on the menu. When everything is ready, the operator selects "no change" to begin to automate the DPO. The program starts by initializing the DPO and setting all specified parameters as directed by the main menu. The program stops when the DPO has finished the setting operation. The program then asks for the bandwidth of the measurement. There are two choices: 20 Ghz or 12.4 Ghz. To protect from errors in the initial setting, the program stops for the final check. As soon as the operator initiates the acqisition cycle, the beginning time shows on the screen. For a typical acquisition composed of 2048 averages and 1 data block, it takes about 7 minutes to finish the operation. In the meantime, the computer waits for the DPO to provide the raw data. When the DPO finishes this operation, the ending time is displayed and the PC opens a temporary hard-disk file for storing the raw data. Before the operation is ended, the header which is to be placed on the data file is displayed on the screen. The operator can then observe any error in the data set, should there be any.

The header is transferred as a separate file. The header provides important characteristics of the data, such as format type, number of points, number of averages, x increment, x origin, x reference, y increment, y origin, y reference and range setting. All of these parameters allow the program to recover meaningful data from the raw data. The remainder of the program constructs the standard data file structure and the hard copy output.

At the end of the program, there is a pause with a query regarding the operator's intention to continue with an additional measurement. To complete the acquisition cycle for one target, the three measurements previously discussed have to be made in order. These three files will then be supplied to the deconvolution program in the next processing step. For acquiring more than three measurements, the background must be measured for each target so it can be subtracted from the respective target measurement. The background measurement occurs either immediately before or after the target measurement. This reduces errors induced by long term nonstationary drifts in the system. The need for such temporal coherence of the target-background measurements has been verified experimentally during the course of this work. Typical outputs are as shown in Figure 3 on page 21.

C. DECONVOLUTION ALGORITHM

The deconvolution program was written in Ryan-McFarland (RM) FORTRAN in order to improve the run time. The source code is standard ANSI 77 FORTRAN so it may be recompiled under virtually any PC FORTRAN compiler such as Microsoft or Lahey. This program was originally written in Tektronix BASIC by Morgan [Ref 11] and was restricted to use on a Tektronix 4052 microcomputer.

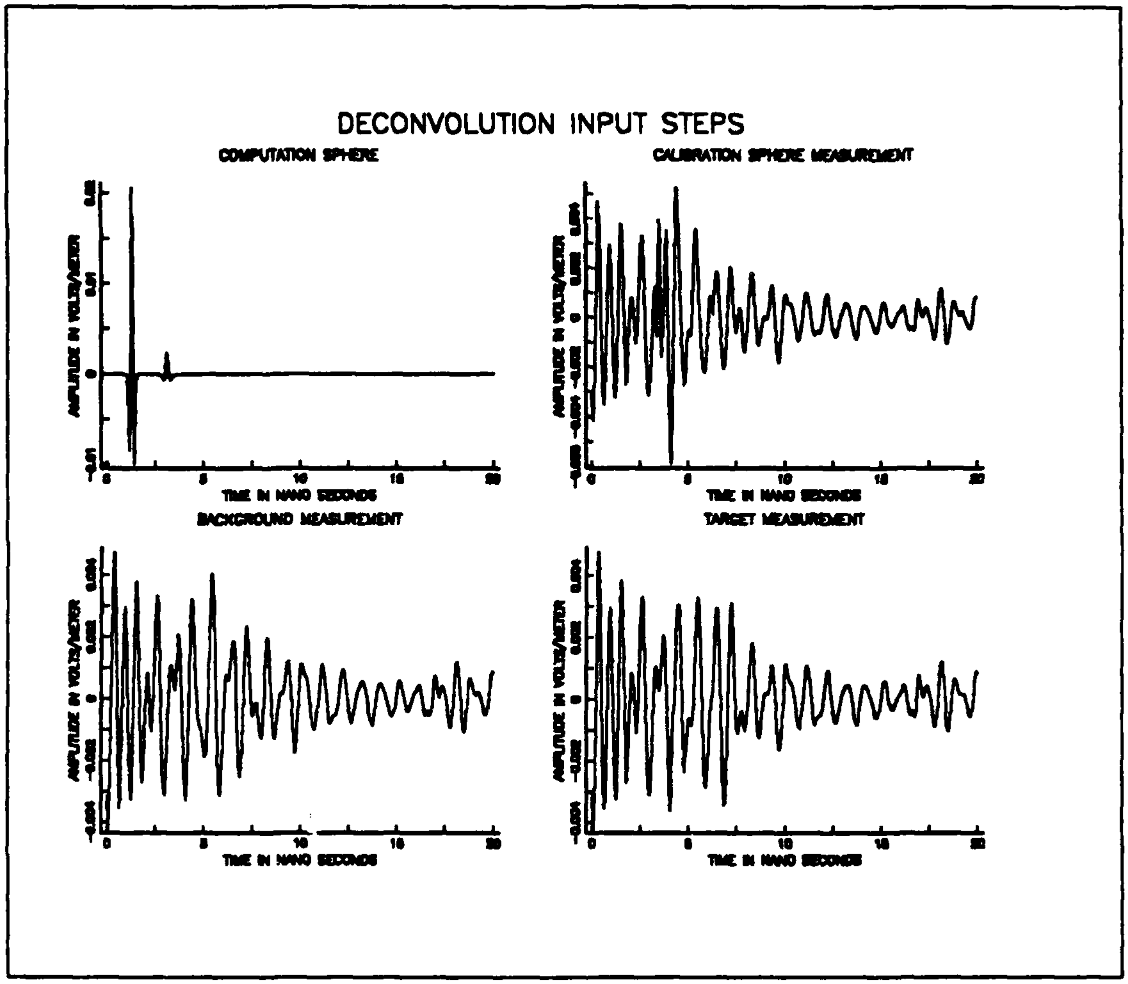


Figure 3. Deconvolution input waveform

The use of a standard FORTRAN-77 compiler generalizes the program so it can be run on any IBM PC compatible. This algorithm follows the mathematic deconvolution as described in chapter III. The program has graphic capabilities which are supplied by GRAFMATIC software. The procedure to compile the program is described in the RM FORTRAN manual. This program must be linked to the GRAFMATIC library in order to run properly.

The program begins by displaying the standard input data file. This data file format has been standardized for use with all of the software support systems. The header of each data file contains the important information such as identification, type of measurement, date of measurement, time of experiment, the maximum time scale, data block and number of points. Types of measurement are as explained in the previous section. The first data file needed for the deconvolution program is that for the computed scattering by the calibration sphere target. This computation data is obtained from the MIE series program and SPRCT program, written by Morgan, as described in the previous chapter. The input data required for these programs are summarized in Table 7 on page 23 .

The sphere scattering computation is transformed into the frequency domain and displayed on the screen in order to observe the bandwidth. The bandwidth of the computation has to be approximately the same as the estimated bandwidth of the measurement. If the bandwidth is not appropriate, the operator should to terminate the program via a Ctl-C and run the MIE series followed by SPRCT program to provide appropriate data.

In the next step, the program asks for the calibration and background file names. Both data are read in and plotted on the same axis in order to observe if time shifting has occurred, and if so, to be able to estimate how much. The operator is then allowed to shift one waveform either left or right and then re-observe the overlayed waveforms. Once a satisfactory alignment is observed, the calibration waveform is subtracted from the background waveform and the difference waveform is transformed into the frequency domain.

The program then stops and asks for the target file name. It then asks if the operator desires to use the same background or another, depending on which background was closer to the target completion acquiring time. In a similar fashion, the target data and the background data are plotted on the same axis. If there is an observed drift, the program allows the operator to shift the target waveform to the left or to the right until both target and background are lined up. Then the subtraction can be performed. The difference waveform is then transformed into the frequency domain. Typical scattering data being processed by deconvolution program is shown in Figure 3 on page 21. The overlay and shift is shown in Figure 4 on page 23 and the subtracted waveform is shown in Figure 5 on page 24. The target for this example was a 10 centimeter long "thick" copper wire, and the calibration target was a standard 0.1025 meter radius aluminum sphere. After the subtraction, the difference calibration waveform and the difference target waveform were tranformed into the frequency domain.

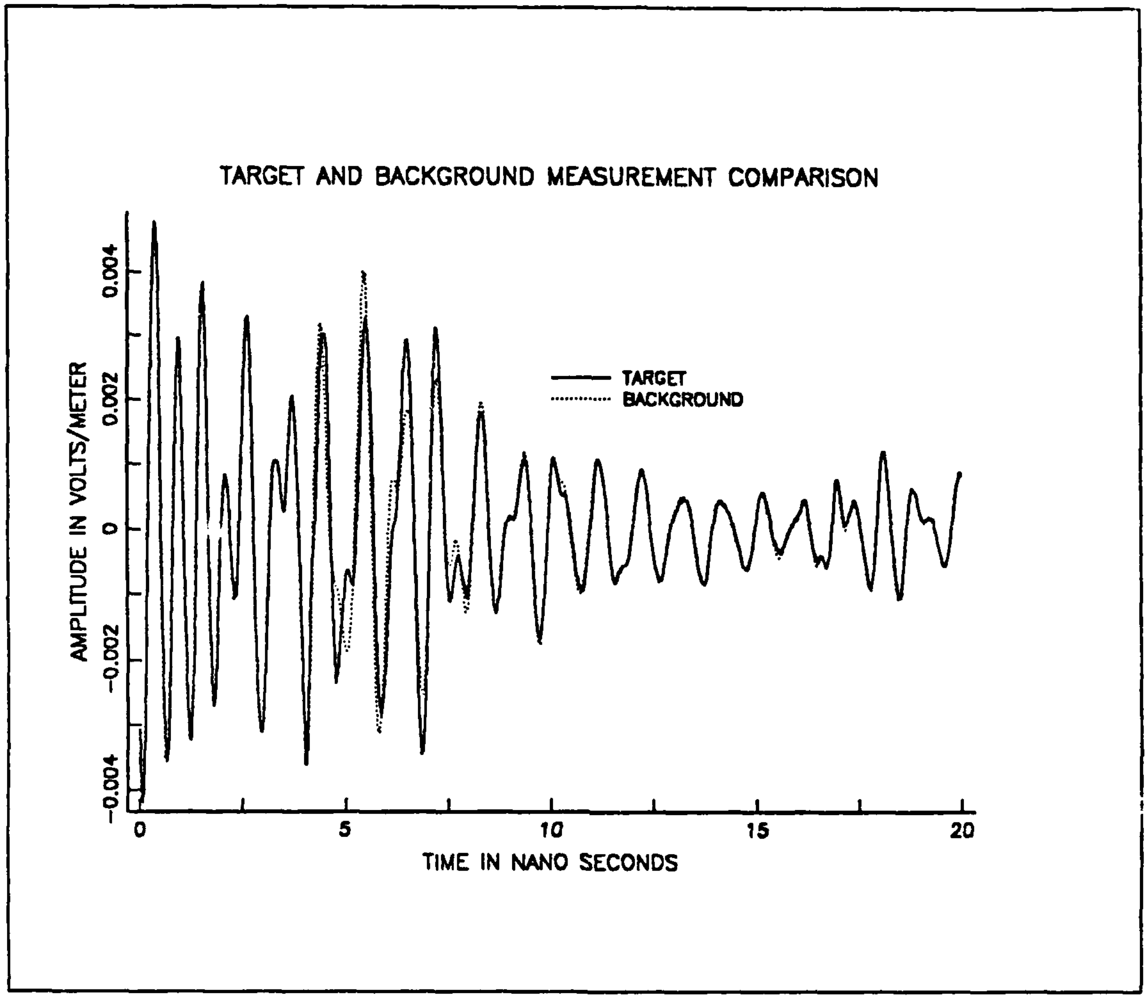


Figure 4. Overlay

Table 7. CALIBRATION SPHERE PARAMETERS

Sphere radius in meters	0.1025
Distance from sphere in meters	2.5
Bistatic angle in degrees	3.0
Scattering plane	E plane
Time window in nanoseconds	20.0
Nyquist frequency in Ghz	12.775
No. frequency points	257

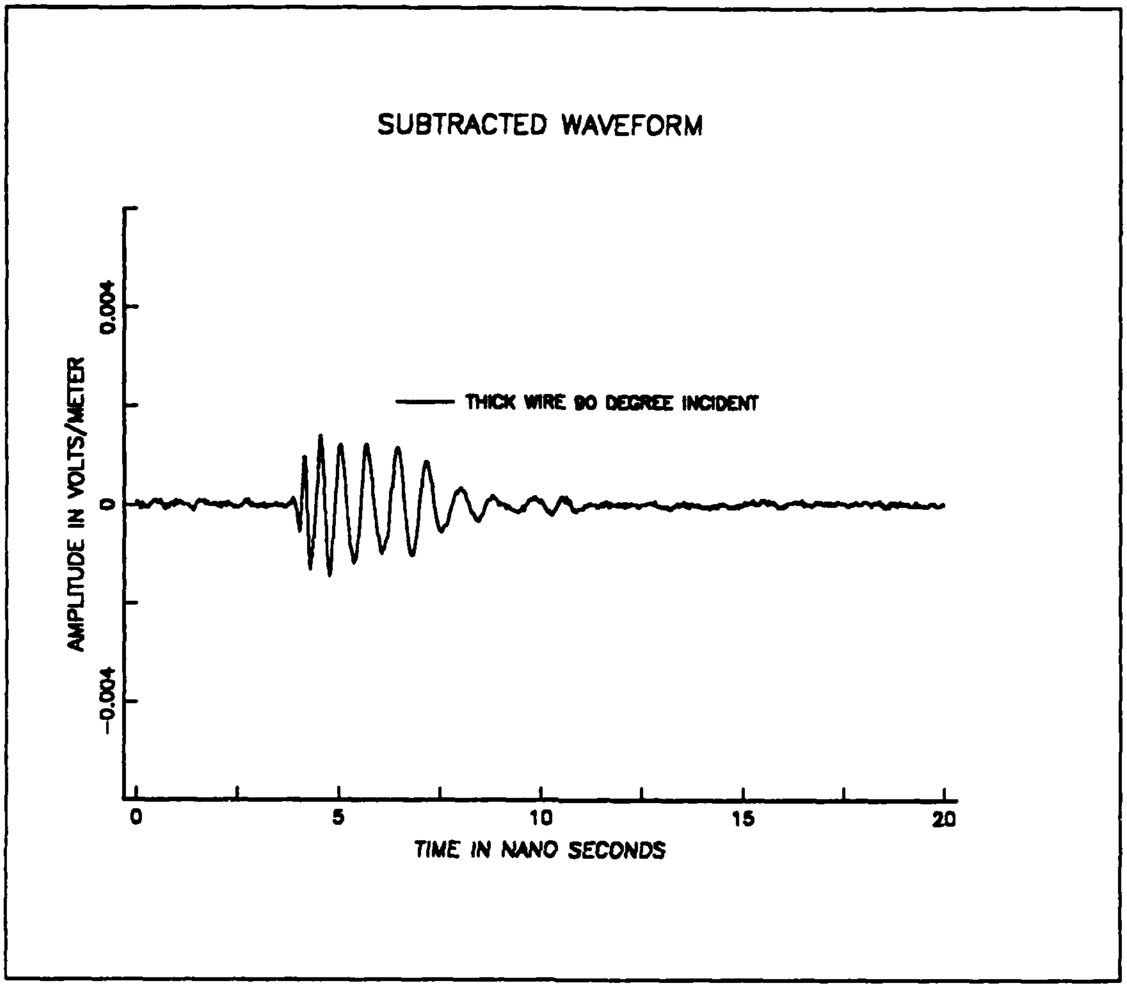


Figure 5. Subtraction waveform

The three frequency domain waveforms were formed into Riad's optimal deconvolution estimator given in equation 3.6. The smoothing constant "C" is selected by the operator. This constant is set to optimize the denominator threshold level in the frequency domain. If it is set too large, oversmoothing will result which reduces the spectral content in both the low and high frequency regimes. If the threshold is set to zero, a "naive" deconvolution results. This provides vastly amplified noise in the signal and typically looks like "garbage". A typical value for C is 0.1. Riad's method yields an estimator for the scattered waveform of the target due to an incident plane wave signal having signal shape equal to that which illuminates the computed calibation sphere using program SPRCT. This frequency domain estimator is transformed to the time domain and the result is displayed. At the end of the program, the operator may try another smoothing parameter. If the result satisfies the operator, it can be stored in the file whose file name was provided by the operator. The program also allows the processing of another target directly without rereading the computed data and the calibration data.

D. EXAMPLE OUTPUT

The output of the deconvolution program is in both the time domain and frequency domains, but the following example shows only the time domain output. This output can be stored on a hard disk or a floppy disk.

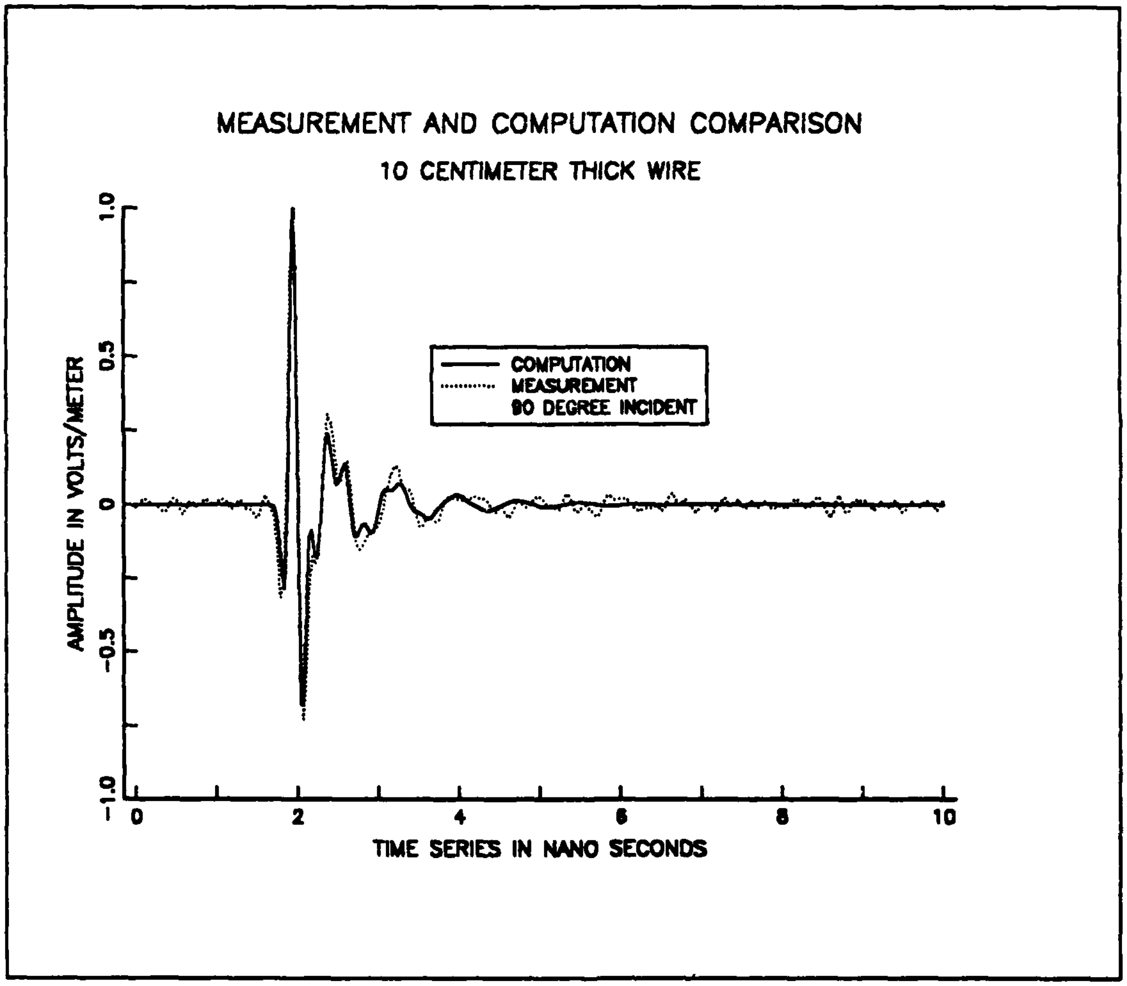


Figure 6. Thick wire measurement

The impulse response shown in Figure 6 belonged to the signature of the thick copper wire having a 90 degree incident wave and a 90 degree reflected wave. The number of averages used in the measurement was 2048. The smoothing was C=0.1. There was no time drift observed. One data block was used. Other parameters can be found in the following table.

Table 8. 90 DEGREE THICK WIRE PARAMETERS

Wire length in meters	0.1
Wire radius in meters	0.0026
Incident angle in degrees	90
DG pulse minimum 10% spread in nanoseconds	0.28
DG pulse maximum 10% spread in nanoseconds	0.30
Time window in nanoseconds	10.00
Impact delay in nanoseconds	2
Scattering angle in degrees	90

Note that the minimum and the maximum 10% spreads determine the respective α_1 and α_2 in the DG waveform of equation 3.9.

First, the frequency spectrum of the smoothed impulse response was calculated from the output and then transformed to the time domain by an FFT [Ref. 18]. The reason for calculating the frequency response first is that it is more convenient to work in the frequency domain when using Raid's method to extract the smoothed impulse response using equation 3.6

The dotted line in Figure 6 on page 25 is the actual measurement. Because the measurement was contaminated by noise, the result did not agree exactly. The continuous line represents the computed thick wire smoothed impulse response due to the same excitation. This computation was performed directly in the time domain using an E-field integral equation. Measurements for other types of structures will be illustrated in the next chapter.

V. EXPERIMENTAL RESULTS

A. CANONICAL MODEL SCATTERING

In addition to the 10 cm long thick wire considered in Chapter 4, four additional canonical targets were selected:

- 10 centimeter thin wire (broadside and canted incidence angles)
- 8.1 centimeter diameter sphere
- 12.2 centimeter diameter sphere
- 15.5 centimeter diameter sphere

The Mie series program, MIE, was used to compute the frequency domain scattering transfer function for the various metallic spheres. These transfer functions were then used as inputs to the SPRCT program to compute the transient scattering responses due to the specified Double-Gaussian (DG) pulse incident plane wave, as specified by equation 3.9. The thin wire computations were performed using the TDIG program. This program utilizes a numerically evolved time-domain electric field integral equation for the induced current distribution on the wire. The scattered field is then calculated by appropriate numerical space-time integrations using this induced current. This and the sphere scattering programs were written by Morgan. The measured and computed DG pulse scattering responses are overlayed in each plot to validate both the fidelity of the experimental procedure and the accuracy of the deconvolution algorithm. The characteristics of each canonical target scattering case are summarized in Tables 9-13 on pages 28-33. A noteworthy item is that there is a slight bistatic angle (3 degrees) between incident and scattered aspects. This has been incorporated into the computations. As can be readily observed, the agreement between measurements and computations in Figures 7-11 on pages 28-32 is quite good in all cases. The additive "rippling" of the experimental results is due in part to the residual noise and clutter contributions. This latter pollutant includes uncancelled multipath interactions of the target and canonical sphere scatterers due to imperfect absorption by the anechoic chamber surfaces. Additional differences in the comparisons are due to the approximations which are innate to the computations.

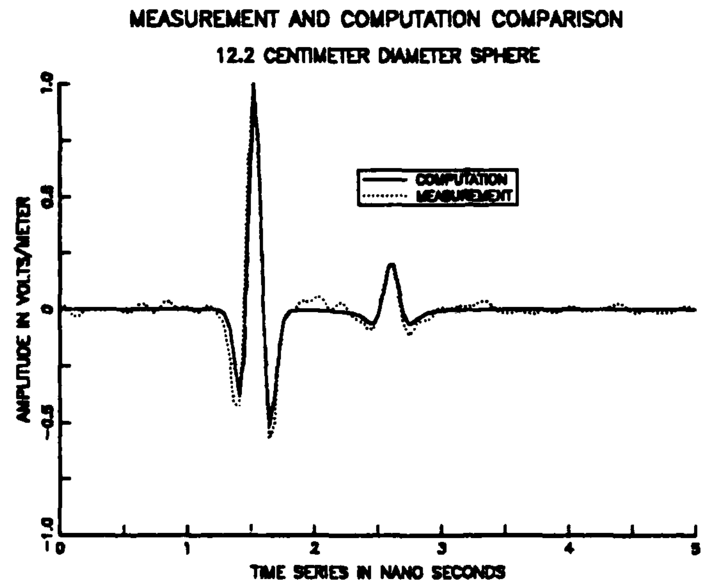


Figure 7. The 12.2 cm diameter sphere validation.

Table 9. 12.2 CM SPHERE PARAMETERS

Sphere radius in meters	0.061
Distance from sphere in meters	2.5
Bistatic angle in degrees	3
Scattering plane	E plane
Time window in nanoseconds	20.0
Narrow 10% pulse width in nanoseconds	0.28
Wide 10% pulse width in nanoseconds	0.30

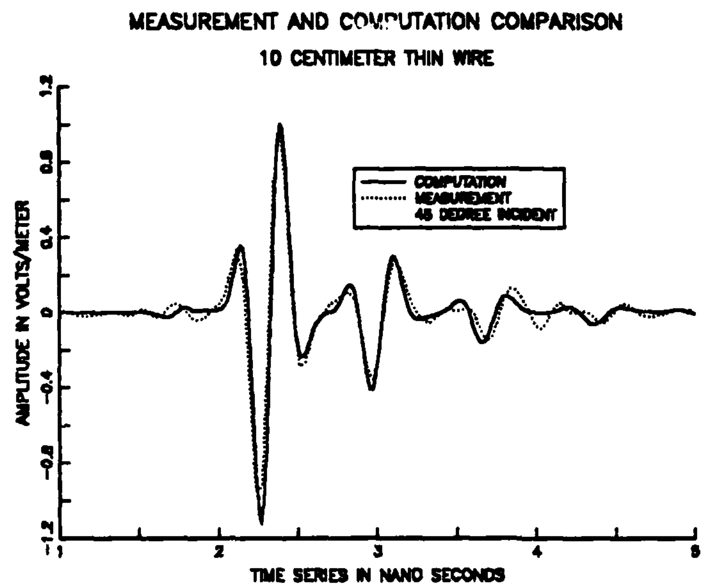


Figure 8. The 10 cm thin wire validation at 45 degrees

Table 10. 45 DEGREE THIN WIRE PARAMETERS

Wire length in meters	0.1
Wire radius in meters	0.00095
Incident angle in degrees	45
DG pulse minimum 10% spread in nanoseconds	0.28
DG pulse maximum 10% spread in nanoseconds	0.30
Scattering angle in degrees	135
No. wire segments	20

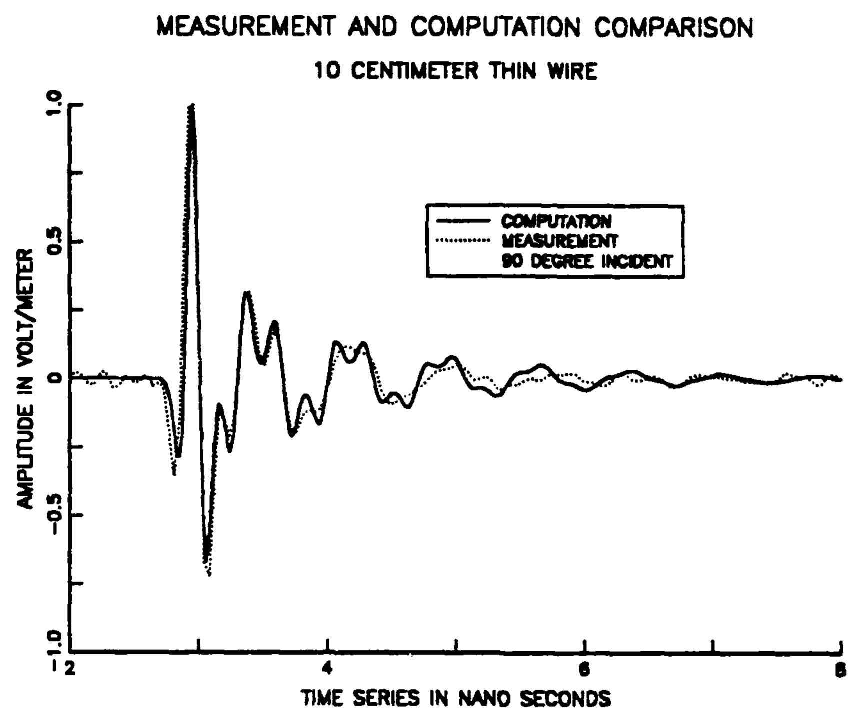


Figure 9. The 10 cm thin wire validation at 90 degrees

Table 11. 90 DEGREE THIN WIRE PARAMETERS

Wire length in meters	0.1
Wire radius in meters	0.00095
Incident angle in degrees	90
DG pulse minimum 10% spread in nanoseconds	0.28
DG pulse maximum 10% spread in nanoseconds	0.30
No. wire segments	20
Scattering angle in degrees	90

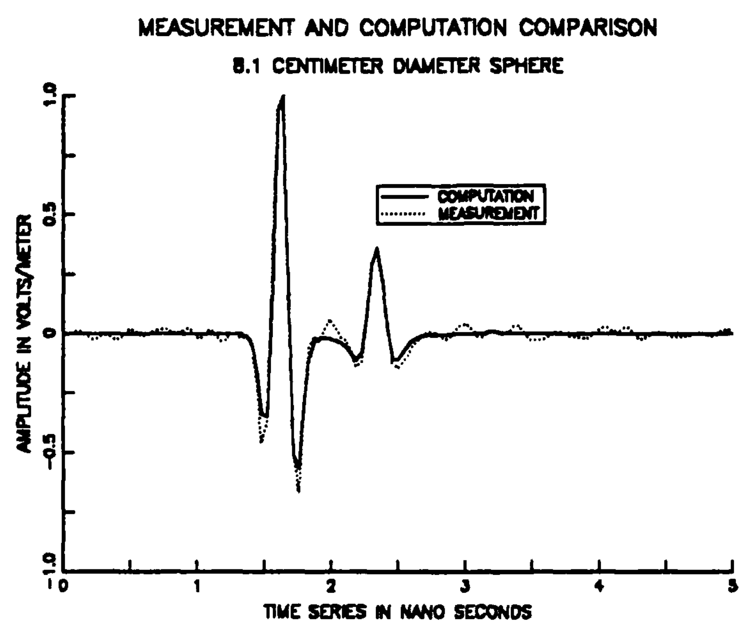


Figure 10. The 8.1 cm diameter sphere validation

Table 12. 8.1 CM SPHERE PARAMETERS

Sphere radius in meters	0.0405
Distance from sphere in meters	2.5
Bistatic angle in degrees	3.0
Time window in nanoseconds	20.0
Nyquist frequency in Ghz	12.775
Narrow 10% pulse width in nanoseconds	0.28
Wide 10% pulse width in nanoseconds	0.30

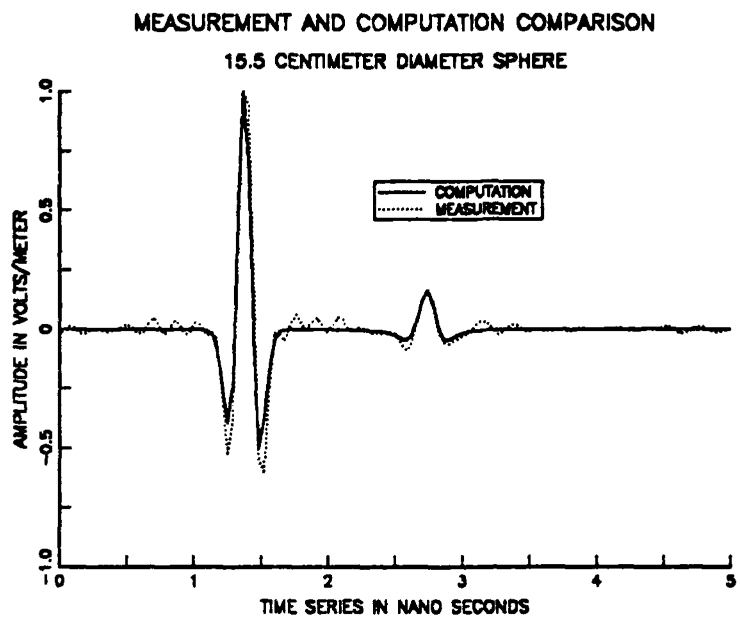


Figure 11. The 15.5 cm diameter sphere validation

B. SCALE MODEL SCATTERING

The scattering responses of the wire and sphere targets due to the incident DG incident pulse can be numerically evaluated using straightforward algorithms. This is not the case for more complex scattering objects, such as a highly realistic model of an aircraft. Primitive computational algorithms do currently exist for such complex structures but the poor accuracy of their results would be insufficient to properly validate the measurements being made here. On the other hand, the very good comparisons shown in the previous section for both low-damped (wires) and high-damped (spheres) strongly

Table 13. 15.5 CM SPHERE PARAMETERS.

Sphere radius in meters	0.0775
Distance from sphere in meters	2.5
Bistatic angle in degrees	3.0
Scattering plane	E plane
Time window in nanoseconds	20.0
Nyquist frequency in Ghz	12.775
No. frequency points	257

support the conjecture that high quality measurements can be obtained for all metallic targets having comparable physical dimensions.

After the validations were made using wires and spheres, extensive measurements were made of the transient backscattering from four different silver coated 1/72 scale plastic aircraft models. The full-size dimensions of each aircraft, whose identities will not be given here, are shown in Tables 14-17 on pages 34-37.

Measurements were made for various incident aspects in the wing-fuselage plane on each model, using horizontal incident polarization: 0 degrees (nose-on); 30 degrees; 90 degrees (broadside); and 180 degrees (tail-on). In addition, a broadside look-down aspect was considered, with backscattering measured for incident linear polarizations both parallel to ("vertical wing") and perpendicular to ("horizontal wing") the fuselage. Samples of these numerous measurements are displayed in Figures 12-15 on pages 34-37.

The complete library of waveforms is being used to support research in demonstrating the accuracy and aspect independence of natural resonance radar target identification.

Four scale model aircraft were selected. They are as follows:

- Scale model 1
- Scale model 2
- Scale model 3
- Scale model 4.

C. SYSTEM NOISE ESTIMATION

Noise in the data acquired from the acquisition program was investigated in this effort in order to develop a mechanism to estimate the signal-to-noise ratio (SNR). The

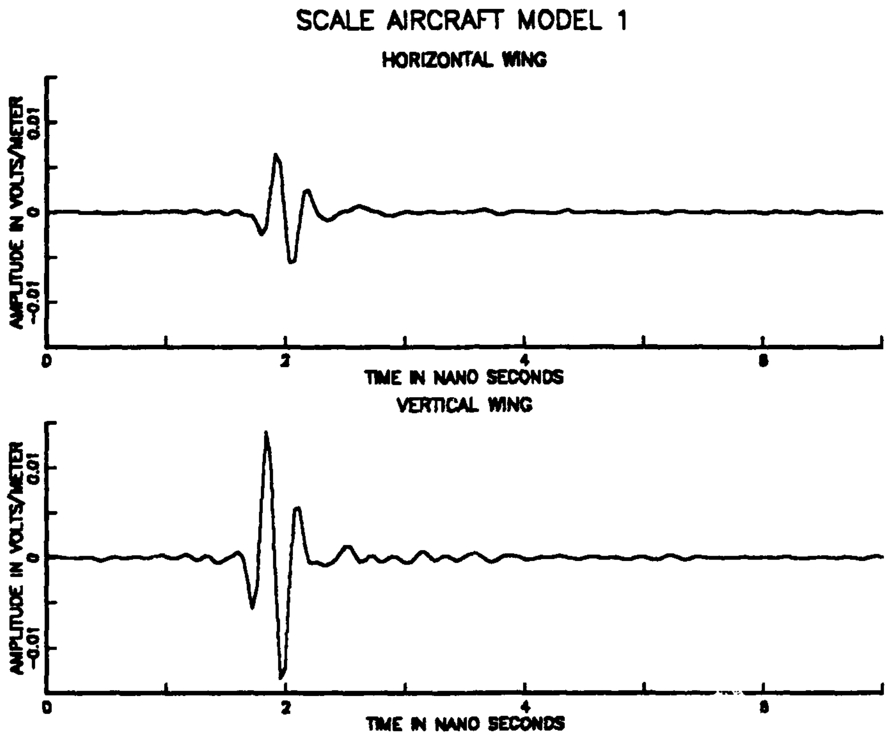


Figure 12. Vertical and horizontal wing scale aircraft model 1

Table 14. DIMENSIONS OF SCALE AIRCRAFT MODEL 1

Scale	1/72
Length	36 feet 4 inches
Wingspan	31 feet

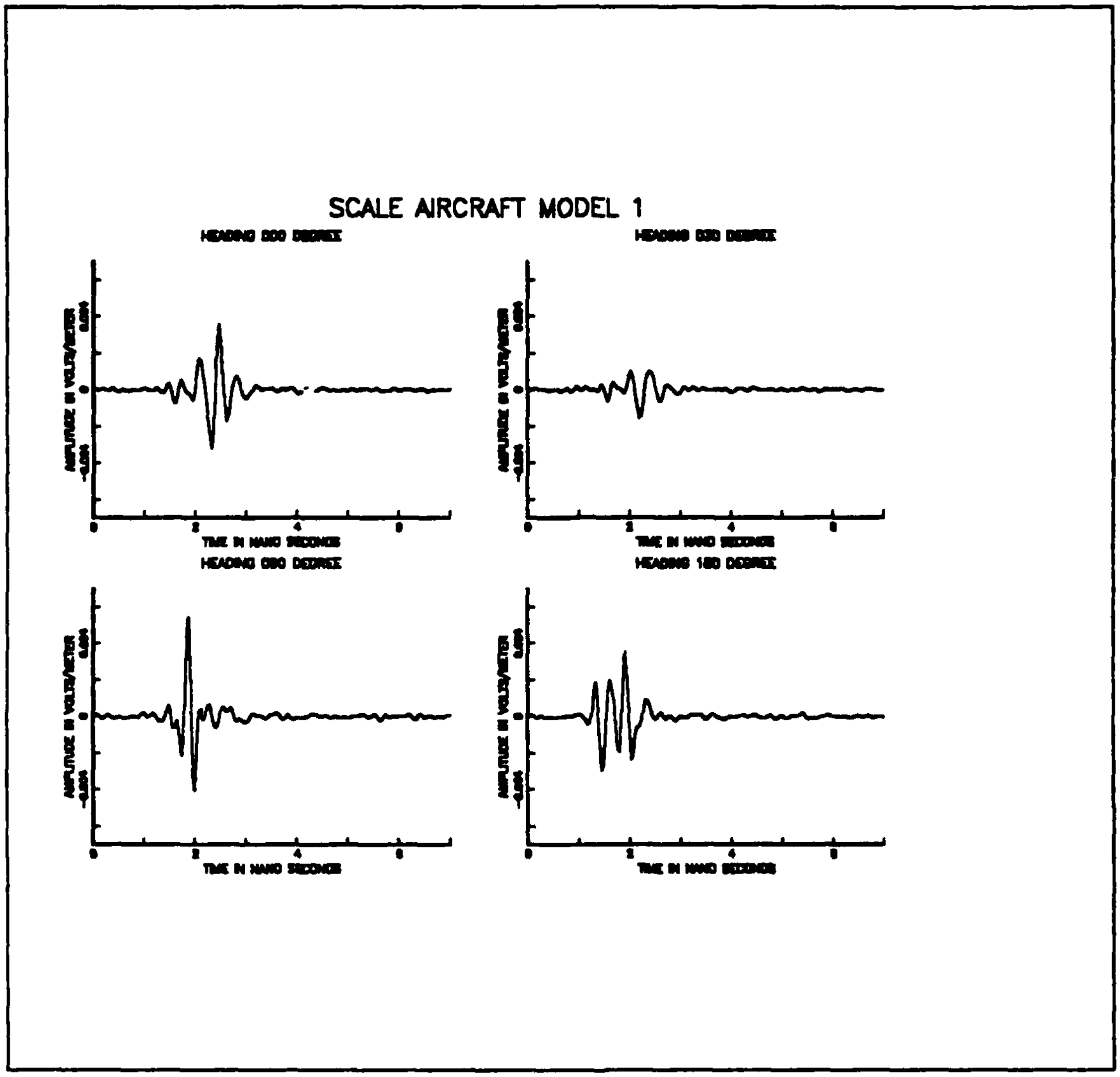


Figure 13. Scale aircraft model 1 at 000, 030, 090, 180 degree aspect angle

Table 15. DIMENSIONS OF SCALE AIRCRAFT MODEL 2

Scale	1/72
Overall length	53 feet
Extended wingspan	28 feet 7 inches

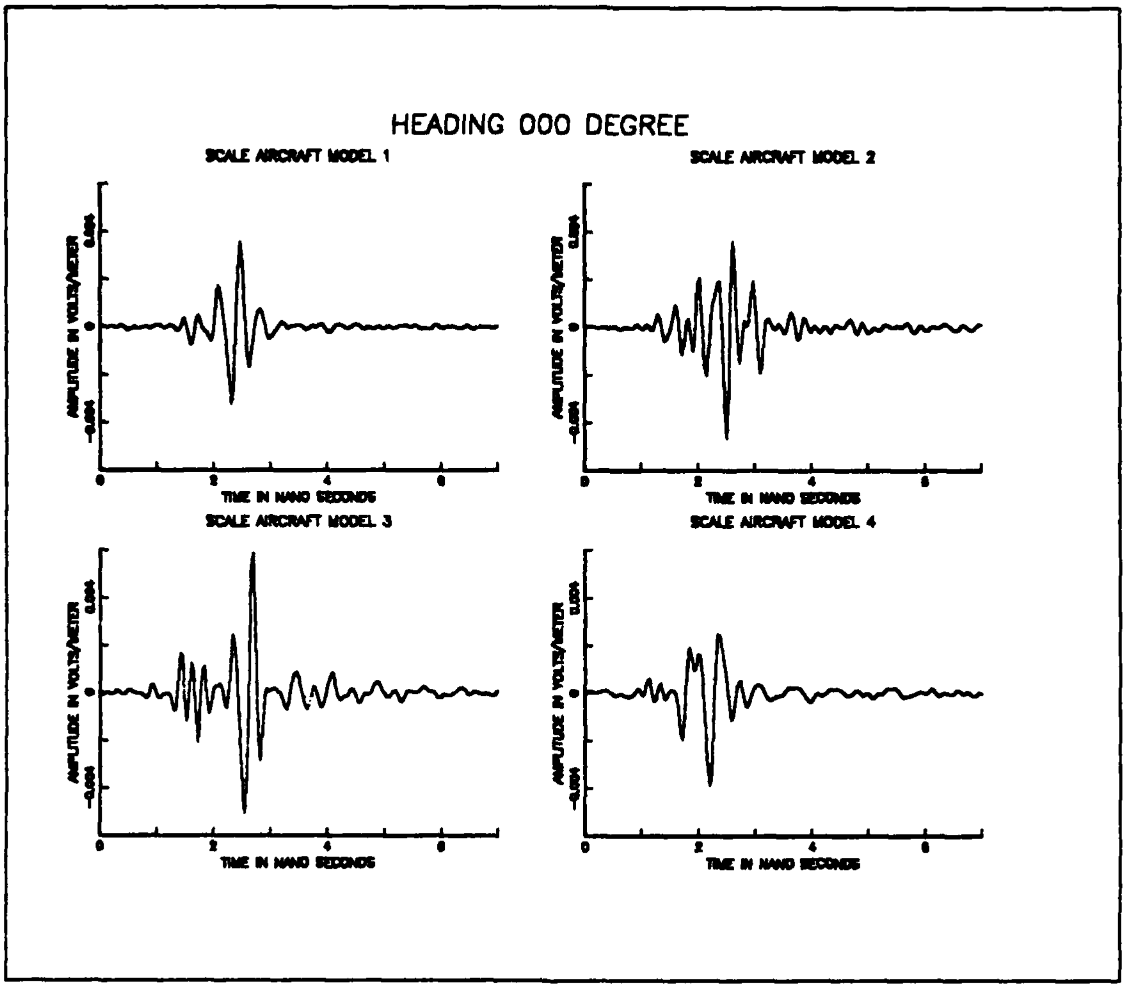


Figure 14. Scale aircraft model comparisons at 000 degree aspect angle

Table 16. DIMENSIONS OF SCALE AIRCRAFT MODEL 3

Scale	1/72
Length	56 feet
Wingspan	40 feet 8 inches
Height	15 feet 4 inches

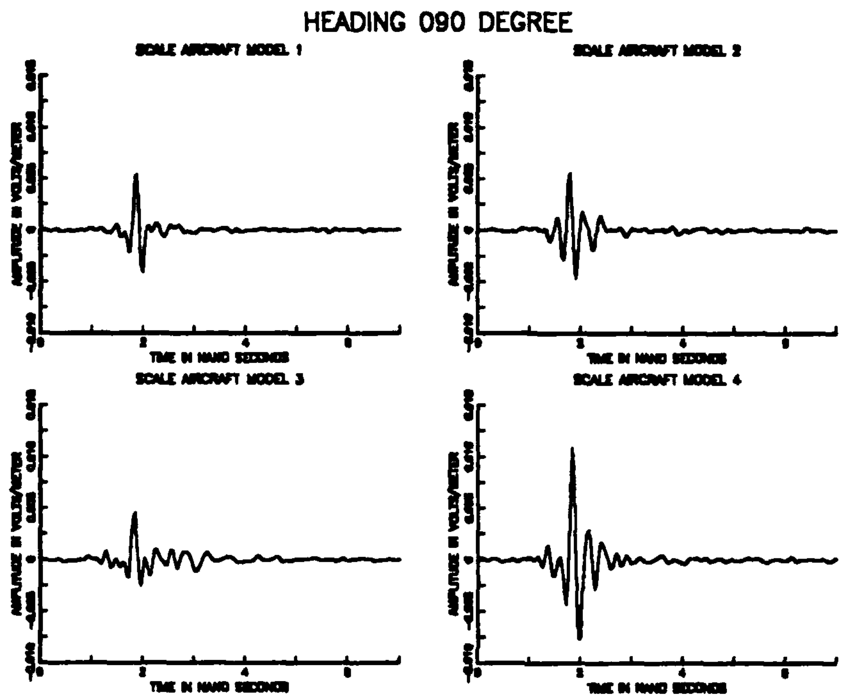


Figure 15. Scale aircraft model comparisons 090 degree aspect angle

Table 17. DIMENSIONS OF SCALE AIRCRAFT MODEL 4

Scale	1/72
Length	63.8 feet
Wingspan	42.8 feet
Height	18.6 feet

SNR is a major determining factor for the measurement fidelity. Much of this SNR assessment involved the computing some statistics of the waveform. All of these statistical calculations were performed using GRAFSTAT with APL.

In the acquisition step, the calibration, the background, the target and a second background were measured consecutively. The two backgrounds were measured for the purpose of noise estimation. In the following analysis, a 2048 ensemble-average mode was used to acquire the data. The acquisition of each waveform took about 9 minutes to complete.

In a separate investigation, the stationarity of the system noise was investigated by acquiring seven consecutive background measurements. These backgrounds were then subtracted from each other to observe the noise occuring during the consecutive measurement. For example, the second background and third background were subtracted, as is shown in Figure 16 on page 39 along with various statistical parameters in Table 18 on page 39 mentioned, the statistics of each subtracted waveform was calculated by the GRAFSTAT software packgage. The variance of the noise can be calculated from the standard deviation and expected power can be approximated by this variance when the noise has a small mean. This approximation was reasonable because the squares of the mean for the actual measurement data are very small, (see Figure 16 on page 39). The computed noise power is plotted versus time in Figure 17 on page 40. As can be seen, the measured noise power varied only slightly with time. A linear least-squares fit yields the equation

$$Y = -93.042 + 0.035727X \ , \tag{5.1}$$

where Y is the expected noise power and X is the time in minutes. This model was based on only seven background measurements with only 2048 averages per measurement. For practical purposes the system noise can be assumed to be fixed in time, depending only upon the number of ensemble averages used to complete the waveform measurement.

To calculate the SNR, the thick wire measurement was used. The signal plus noise waveform is displayed in Figure 18 on page 41 while the computed statistical parameters are given in Table 20 on page 41.

This waveform was time windowed to eliminate the noise outside of the window before the statistical parameters were calculated. The signal plus noise power was calculated by way of the variance using the following equation.

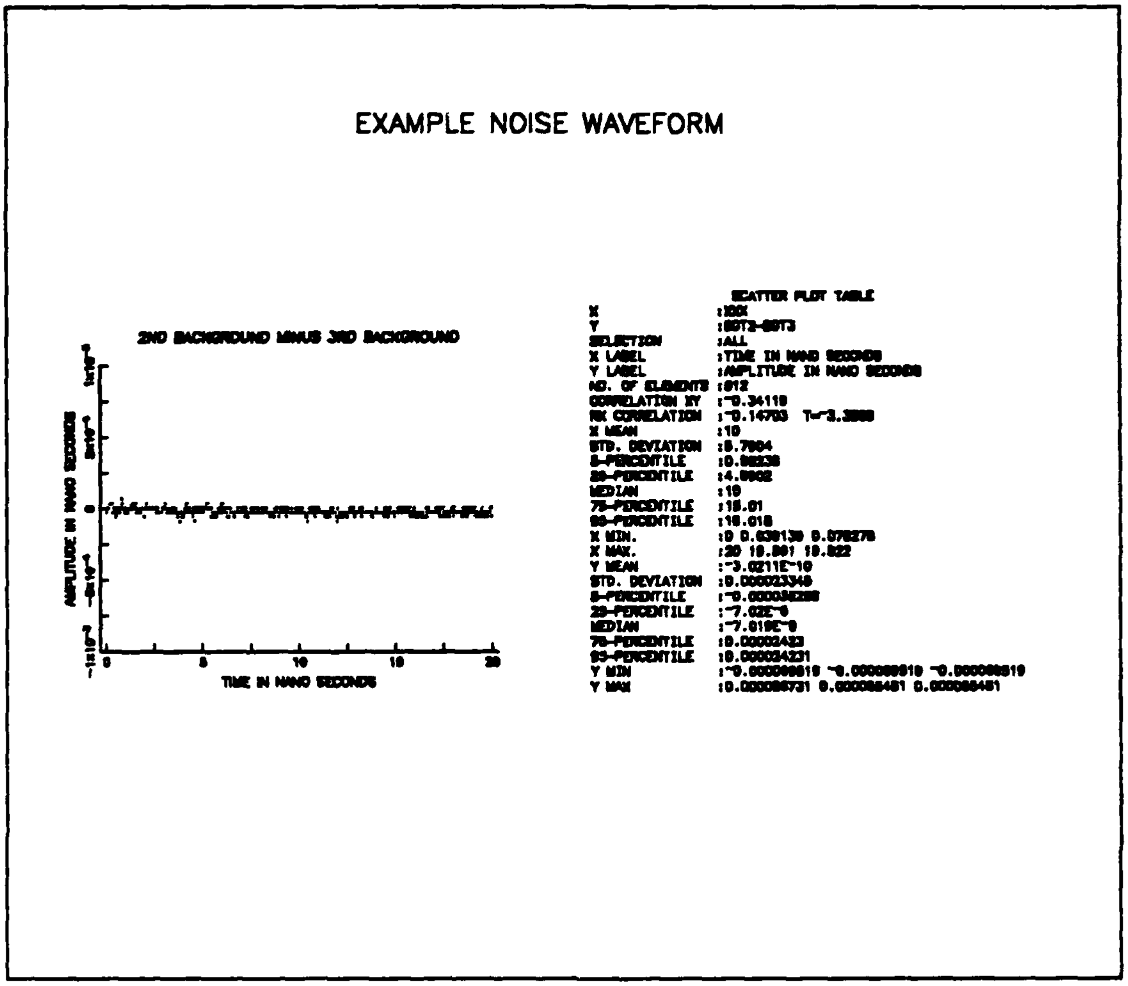
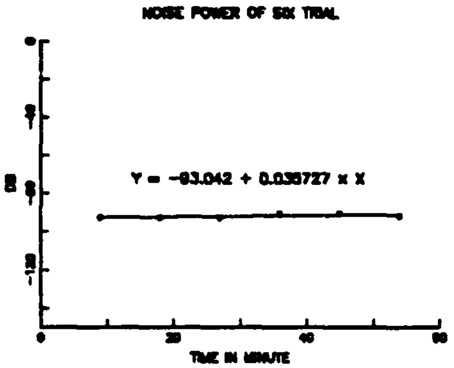


Figure 16. Noise waveform statistics

Table 18. STATISTICS OF NOISE WAVEFORM

Mean	-3.0211E-10
Std. deviation	0.000023345
Median	-7.019E-8
Maximum	0.000066731

NOISE ANALYSIS



SCATTER PLOT TABLE

X	:TIME
Y	:10x10*POWERS
SELECTION	:ALL
X LABEL	:TIME IN MINUTE
Y LABEL	:DB
NO. OF ELEMENTS	:6
CORRELATION XY	:0.60163
RK CORRELATION	:0.63775 T=1.656
X MEAN	:31.5
STD. DEVIATION	:16.837
5-PERCENTILE	:9
25-PERCENTILE	:18
MEDIAN	:31.5
75-PERCENTILE	:45
95-PERCENTILE	:54
X MIN.	:9 18 27
X MAX.	:54 45 36
Y MEAN	:¯91.917
STD. DEVIATION	:0.99986
5-PERCENTILE	:¯92.76
25-PERCENTILE	:¯92.716
MEDIAN	:¯92.343
75-PERCENTILE	:¯90.669
95-PERCENTILE	:¯90.669
Y MIN	:¯92.76 ¯92.716 ¯92.636
Y MAX	:¯90.669 ¯90.669 ¯92.049

Figure 17. Noise power

Table 19. STATISTICS OF NOISE MODEL

Mean	-91.917
Std. deviation	0.99986
Median	-92.343
Maximum	-90.669

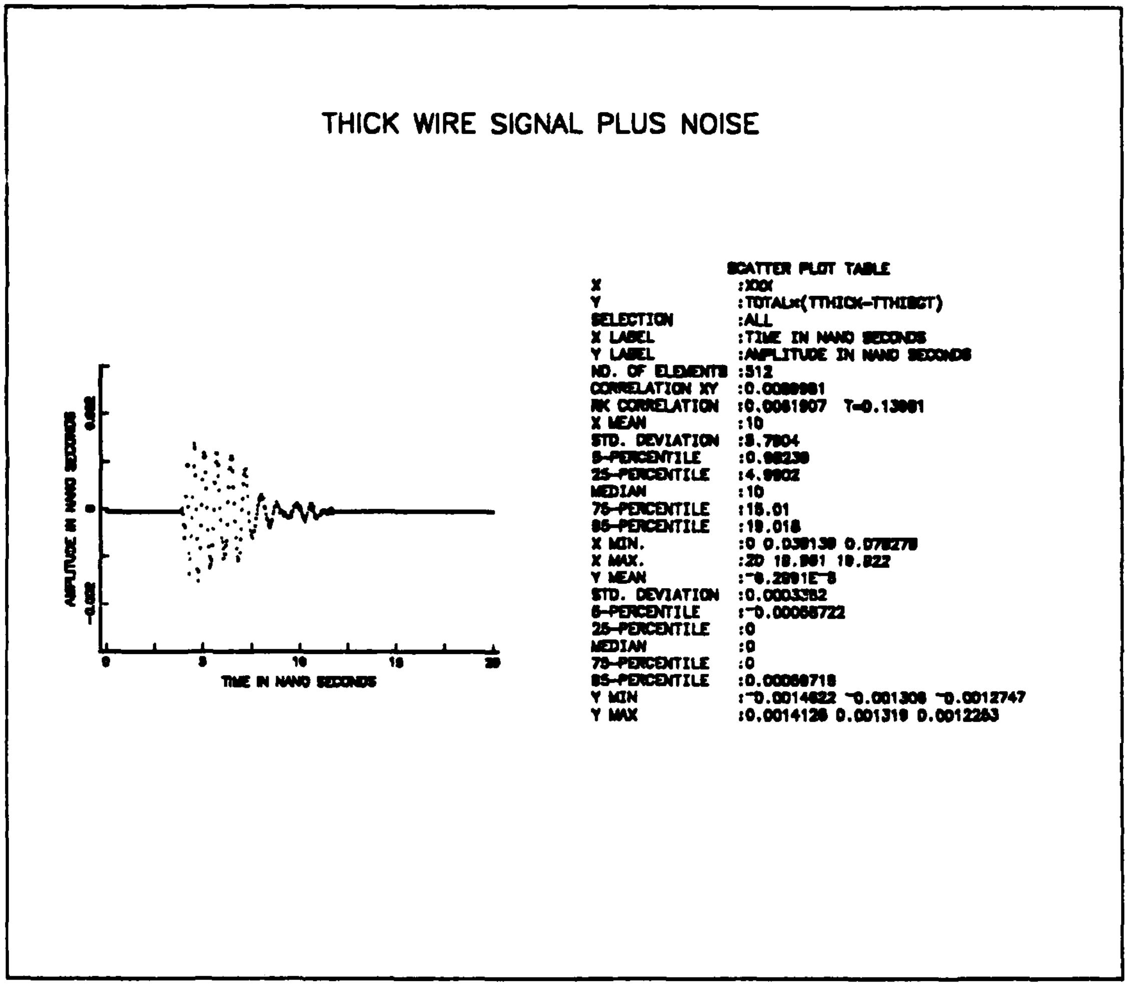


Figure 18. Signal plus noise waveform statistics

Table 20. STATISTICS OF THICK WIRE SIGNAL PLUS NOISE WAVEFORM

Mean	-6.2991E-6
Std. deviation	0.0003362
Median	0
Maximum	0.0014128

$$E(X^2) = Variance + E^2(X) \quad (5.2)$$

where $E(X)$ is mean of the noise and $E(X^2)$ is the mean square value of the noise (noise power). Note that the second term on the right side of equation 5.2 was close to zero, so it can be neglected. As a result, the expected power is equal to the variance. The signal plus noise power was equal to 0.0003382^2 or -69.41 dB. The noise expected power was calculated by equation 5.1 and is given by -92.64 dB for the average time of 10 minutes. Since the noise power was so small compared to the signal plus noise power, the subtraction between the signal plus noise and the noise was not necessary. The noise power when time windowed was reduced by a factor of 200/512, or 4.08 dB. The result for the SNR is 27.31 dB in the case of the thick wire measurement. Note that this SNR is about 10 dB better than that provided for the same target using the Tektronix equipment, as carried out by McDaniel [Ref. 11].

VI. CONCLUSIONS

A. SUMMARY

This thesis describes the theory and implementation of a newly updated transient scattering measurement facility. An overview of the historical precedents and motivations for this development was given in Chapter I. This was followed in Chapter II by a detailed description of the new equipment being used as well as the physical structure of the anechoic chamber.

Building upon this overview of the hardware aspects, the conceptual and mathematical modeling of the scattering range is undertaken in Chapter III. Using the physical model of the radiating and receiving antennas in the anechoic chamber, the various electromagnetic interactions between the scattering target and the chamber are represented by a comprehensive linear system model. The topology of the various frequency domain transfer functions in this model is then used to justify the measurement and pre-processing steps required to estimate the smoothed scattering impulse response, due to a specified double-Gaussian incident plane wave. There are three measurements required for each target impulse response estimation. Aside from the actual target measurement, a background measurement is needed to subtract out the directly received signal from the transmitting antenna, in addition to that scattered from the various chamber surfaces. A third measurement, for a metallic sphere, is needed to allow for compensation of the transfer function characteristics of the transmitting and receiving antennas in the chamber. By effecting a deconvolution comparison with a highly accurate numerical calculation of the transient scattering response of the sphere target, this final measurement also permits the estimation of the target's response to the same excitation as is used in the sphere computation.

Chapter IV considers the details of the actual software that performs the signal acquisition, signal averaging and deconvolution post-processing of the measured waveforms. The acquisition and signal averaging software is a blend of a commercial IEEE-488 bus controller (GURU-II) and a custom driver, designed as part of this thesis effort. The deconvolution algorithm provides, first of all, graphically displayed iterative background subtractions from both the target and sphere measurements to allow compensation for time drifts. Secondly, the final deconvolution is performed via a variant on Riad's method, as discussed in Chapter III.

The validations shown in Chapter V speak for themselves as to the expected measurement quality obtained using this new transient scattering range. Both low-Q and high-Q type targets are considered by the respective comparisons of sphere and thin-wire scattering measurements with computed smoothed impulse responses. Further illustrations of the type of measurements to be performed with this range are shown in the scale model aircraft backscattering signatures. A simple software modification will allow the display of very broadband frequency domain radar cross sections (RCS) for scale model targets.

To quantitatively investigate the fidelity of the measurement system, a noise estimation of the new implementation was investigated and was modelled. The particular signal-to-noise ratio (SNR) of a thick wire scattering measurement was calculated. A 27 dB SNR of this case demonstrated the relatively high fidelity of the measurement system for the case small RCS targets, including silver coated 1/72 scale model aircraft.

B. FUTURE CONSIDERATIONS

Even though the demonstrated system performance was excellent, there is no such thing as too high an SNR in scattering measurements. In this regard, the power-bandwidth product has the potential to be enhanced by future efforts.

The 1 to 7 GHz passband of the scattering system was limited by the transmitting GaAS power amplifier. On the other hand, the bandwidth of the HP DPO system can go as high as 20 Ghz, while the current antennas have a 1 to 12 GHz passband. Future use of broader bandwidth antennas and power amplifier, having a rated RMS output power of 2 watts or more, would increase the overall system performance. More accurate smoothed impulse responses could be achieved, as well as wider bandwidth RCS measurements.

Because of the limited scope of this thesis, the noise power estimation was only an approximation. A more comprehensive noise analysis should be carried out in order to quantify and minimize the noise power sources.

The final consideration is the future investigation of alternate mechanisms for providing broadband scattering measurements. One such mature technique is through a stepped frequency continuous wave system, employing a network analyzer front-end as the coherent receiver, perhaps preceded by a low-noise amplifier to reduce the overall system noise figure. A somewhat radical technique, one that is currently being investigated at NPS, is to use a very broadband amplified noise source as the transmitting source, while employing the HP DPO as a two-channel correlation receiver. This method is akin to a dynamic matched filter, offering enhanced bandwidth, at relatively low cost vis-a-vis the stepped frequency setup. Additionally, the potential exists for both tactical and strategic employment of such random source systems, either as active "pseudo-jammers" or as totally passive radars relying upon either natural or man-made random signal sources for the illumination of targets.

APPENDIX A. THE IMPULSE RESPONSE DECONVOLUTION

The following computer program calculates the Impulse Response from the measurement data as described in chapter 4. The program is written in RM/FORTRAN.

The subroutine PLO8 and P11 require the GRAFMATIC routine library. The procedure to compile, link and run this source code refers to the RM/FORTRAN User's Guide Version 2.11 (DOS).

```
C**************************************************
C     THIS IS THE IMPULSE RESPONSE DECONVOLUTION *
C     WRITTEN BY LT.JG SOONPUEN SOMAPEE  Feb 88  *
C     ADAPTATION OF ORIGINAL TEKTRONIX PROGRAM   *
C     BY PROF.  M.A.MORGAN  Circa 1983-85        *
C**************************************************
C**************************************************
C     WHERE:                                     *
C     DATE      IS THE DATE OF MEASUREMENT       *
C     DATAB     IS THE NUMBER OF DATA BLOCKS     *
C     F1        IS THE DUMMY STRING              *
C     ID        IS THE IDENTIFICATION STRING     *
C     K         IS THE NUMBER OF POINTS          *
C     L9        IS THE RELATIVE SMOOTHING        *
C     MAG       IS THE MAGNITUDE OF Xo*Ht(f)     *
C     NAME4     IS THE NAME OF THE TARGET DATA   *
C     N         IS THE NUMBER OF POINTS          *
C     NUMPT     IS THE SAME AS K                 *
C     NEXPT     IS INDICATEES POWER-OF-2 EXPONENT *
C               WILL BE GENERATE BY PREFFT       *
C               SET TO -1 INDICATES ERROR        *
C     R         IS THE REAL PART OF THE INVERSE  *
C               FOURIER TRANSFORM OF Xo*Ht(f)    *
C     SUBAV     IS THE NUMBER OF SUB-AVERAGES    *
C     T         IS THE TIME INTERVAL             *
C     TIME      IS THE TIME INTEVAL              *
C     TITLE     IS THE TITLE NAME FOR PLOTTING   *
C     TUT       IS THE FREQUENCY DOMAIN OF Xo*H(f)*
C     TYPE      IS THE WAVEFORM TYPE             *
C     W         IS THE COMPLEX EXPONENTIAL ARRAY *
C               GENERATED BY PREFFT SUBROUTINE   *
C     Xo*Ht(f)  IS THE GAUSSIAN WEIGHTED COSINE  *
C     XS        IS THE STRING FOR PLOTTING ROUTINE*
C               FOR THE HEADER                   *
C               DRIVER                           *
C     Y4        IS THE CALIBRATION SPHERE        *
C               SUBTRACTED WAVEFORM (FREQUENCY)  *
C                                                *
C     Y5        IS THE TARGET SUBTRACTED WAVEFORM *
C               FREQUENCY DOMAIN                 *
C                                                *
C     YSC       IS THE COMPUTED SCATTERED ELECTRIC*
```

```
C               FIELD FOR THE CANONICAL TARGET    *
C     Y1        IS THE CALIBRATION MEASUREMENT    *
C     Y2        IS THE BACKGROUND MEASUREMENT     *
C     Y3        IS THE TARGET MEASUREMENT         *
C     YS        IS THE SAME AS XS                 *
C     YN        IS THE YES OR NO STRING           *
C     XMI1      IS THE MINIMUM VALUE              *
C     XMA1      IS THE MAXIMUM VALUE              *
      CHARACTER*64 TITLE,XS,ID,YS,DATE,TYPE,NAME
      CHARACTER*64 TYPE1,TYPE2,TYPE3,F1
      CHARACTER*16 NAME1,NAME2,NAME3
      CHARACTER*64 ID1,ID2,ID3
      CHARACTER*1  YNS,YNS1
      COMPLEX YSC(1025),Y4(1025),Y5(1025),W(1025)
      REAL T,T2 ,TIME
      REAL Y1(1024),Y2(1024),Y3(1024)
      INTEGER N,SUBAV,DATAB,NUMPT
C**************************************************
      YNS='N'
C**************************************************
C     READ DATA FOR COMPUTED SPHERE THEN
C     CALLING SUBROUTINE FFF TO DRAW THE TIME
C     SERIES AND SPECTRAL PLOT
C**************************************************
      WRITE(*,*) '****************************************************'
      WRITE(*,*) '*                                                  *'
      WRITE(*,*) '*  DECONVOLUTION  PROGRAM  BY DR. M.A. MORGAN      *'
      WRITE(*,*) '*  ADAPTED TO IBM BY LT.JG. SOONPUEN SOMAPEE       *'
      WRITE(*,*)
      WRITE(*,*) '*               FORMAT OF DATA FILE:               *'
      WRITE(*,*)
      WRITE(*,*) '*  1. READ ID              (FORMAT CHARACTER*64) *'
      WRITE(*,*) '*  2. READ TYPE            (FORMAT CHARACTER*64) *'
      WRITE(*,*) '*  3. READ DATE            (FORMAT CHARACTER*64) *'
      WRITE(*,*) '*  4. READ F1              (FORMAT CHARACTER*64) *'
      WRITE(*,*) '*  5. READ SUB-AVERAGE     (FORMAT INTEGER)      *'
      WRITE(*,*) '*  6. READ TIME WINDOW     (FORMAT REAL(F5.2))   *'
      WRITE(*,*) '*  7. READ DATA BLOCK      (FORMAT INTEGER)      *'
      WRITE(*,*) '*  8. READ NUMBER OF POINTS (FORMAT INTEGER)     *'
      WRITE(*,*) '****************************************************'
      WRITE(*,*) 'FILE NAME FOR COMPUTED CALIBRATION SPHERE:  '
      READ(*,173) NAME
C************************************************************
C
C     YSC = COMPUTED SPHERE COMPLEX FREQUENCY DOMAIN DATA
C
C     NUMPT = NUMBER OF POINTS IN THE OUTPUT OF THIS ROUTINE
C
C     TIME = OUTPUT TIME INTERVAL
C
C************************************************************
      CALL FFF1(NAME,YSC,W,NEXPT,TIME)
C************************************************************
9110  CONTINUE
C************************************************************
C     READ THE DATA FILES FOR CALIBRATE AND BACKGROUND
```

```
C     THEN OVERLAY THESE ONTO THE SAME AXIS.
C     PLOT77 ALLOWS THE OPERATOR TO SHIFT ONE WAVEFORM
C     PRIOR TO SUBTRACTING THE BACKGROUND.
C
C     THE RESULT FROM SUBTRACTION IS TRANSFORMED TO
C     FREQUENCY DOMAIN AND SAVED TO THE COMPLEX ARRAY Y4
C
C     NAME1 = NAME OF THE FILE TO SENT TO PLOT77 ROUTINE
C             (CALIBRATE DATA)
C     NAME2 = NAME OF THE BACKGROUND DATA
C     ID1   = IDENTIFICATION  CHR$ FOR CALIBRATE
C     Y4    = COMPLEX ARRAY OUTPUT (CALIBRATE )
C*************************************************************
      WRITE(*,*) '****************************************'
      WRITE(*,*) 'FILE NAME FOR CALIBRATION SPHERE:  '
      READ(*,173) NAME1
      CALL INPP(NAME1,ID1,TYPE1,DATE,F1,SUBAV,T2,DATAB,NUMPT,Y1)
      WRITE(*,*) '****************************************'
      WRITE(*,*) 'FILE NAME FOR SPHERE BACKGROUND WAVEFORM: '
      READ(*,173) NAME2
C**************************************************
      CALL INPP(NAME2,ID2,TYPE2,DATE,F1,SUBAV,T2,DATAB,NUMPT,Y2)
      CALL PLC77(Y1,Y2,T2,NUMPT,ID1)
C**************************************************
9112  CONTINUE
      WRITE(*,*) '****************************************'
      WRITE(*,*) 'FILE NAME FOR TARGET WAVEFORM:     '
      READ(*,173) NAME3
C**************************************************
      CALL INPP(NAME3,ID3,TYPE3,DATE,F1,SUBAV,T2,DATAB,NUMPT,Y3)
C**************************************************
      WRITE(*,*)
      WRITE(*,*) 'USE PREVIOUS BACKGROUND FOR THE TARGET ? (Y/N):    '
      READ(*,173) YNS1
      IF(YNS1.EQ.'Y'.OR.YNS1.EQ.'y') GO TO 9222
      WRITE(*,*) '****************************************'
      WRITE(*,*) 'FILE NAME FOR TARGET BACKGROUND WAVEFORM: '
      READ(*,173) NAME2
C**************************************************
      CALL INPP(NAME2,ID2,TYPE2,DATE,F1,SUBAV,T2,DATAB,NUMPT,Y2)
9222  CONTINUE
C
C     REPEATING SHIFT AND SUBTRACT OPERATIONS ON TARGET AND BACKGROUND
C     NAME1= TARGET DATA FILE
C     NAME2= BACKGROUND DATA FILE
C     OUTPUT
C     ID1 = IDENTIFICATION OF THE TARGET
C     Y5  = COMPLEX ARRAY OF FREQUENCY DOMAIN TARGET DATA
C**************************************************
      CALL PLO77(Y3,Y2,T2,NUMPT,ID3)
           DO 987 I=1,NUMPT
                Y4(I)=CMPLX(Y1(I),0.0)
                Y5(I)=CMPLX(Y3(I),0.0)
987        CONTINUE
      MODE=0
      CALL FFT(NUMPT,MODE,TIME,NEXPT,W,Y4)
```

```
      CALL FFT(NUMPT,MODE,TIME,NEXPT,W,Y5)
      WRITE(*,*) '********************************************'
      CALL DECON3(NUMPT,YSC,Y4,Y5,ID3,TYPE3,DATE,F1,SUBAV,T2,DATAB)
      WRITE(*,*) '************************************************'
      WRITE(*,*) 'WOULD YOU LIKE TO PROCESS ANOTHER TARGET ? (Y/N): '
      READ(*,173) YNS
      IF(YNS.EQ.'Y' .OR. YNS.EQ.'y' ) GO TO 9112
      WRITE(*,*) '*********************************'
      WRITE(*,*) '* eeeee    eeeeee     e         *'
      WRITE(*,*) '*   eee    eee    eeee          *'
      WRITE(*,*) 'eeeeeeeeeeeeeeeeeeeeeeeeeeeee   *'
      WRITE(*,*) '     eeeeeeeeeeeeeeeeeeeeeeee   *'
      WRITE(*,*) '      eeeeeeeeeeeeeeeeeeeeee    *'
      WRITE(*,*) '*********************************'
      WRITE(*,*) '*********************************'
173   FORMAT(A)
      STOP
      END

      SUBROUTINE INPP( NAME,ID,TYPE,DATE,F1,SUBAV,T2,DATAB,NUMPT,Y)
C***********************************************************************
C
C     INPUT  NAME
C     OUTPUT ID,TYPE,DATE,F1,SUBAV,T2,DATAB,NUMPT,Y
C     NAME  = CHR$ NAME OF THE DATA FILE
C     ID    = CHR$ TARGET IDENTIFICATION
C     TYPE  = CHR$ WAVE FORM IDENTIFICATION
C     DATE  = CHR$ DATE
C     INTEGER
C     SUBAV = NUMBER OF SUB-AVERAGES USED TO ACQUIRE THE DATA
C     DATAB = DATA BLOCK
C     F1    = DUMMY
C     NUMPT = NUMBER OF POINTS
C     REAL
C     T2    = TIME WINDOW
C     Y     = Y ARRAY OUTPUT
C***********************************************************************
       CHARACTER*16 NAME
       CHARACTER*64 ID,F1
       CHARACTER*64 TYPE,DATE
       REAL*4 Y(1500)
       INTEGER SUBAV,DATAB,NUMPT,K
       PI=3.141592654
       K=1
       OPEN (UNIT=10,FILE=NAME,STATUS='OLD')
C      READ THE HEADER
           READ(10,5) ID
           READ(10,5) TYPE
           READ(10,5) DATE
           READ(10,5) F1
           READ(10,2030) SUBAV
           READ(10,2010) T2
           READ(10,2030) DATAB
           READ(10,2030) NUMPT
10         READ(10,2000,END=20) Y(K)
```

```
              K=K+1
17          GOTO 10
20       CLOSE(10)
2000     FORMAT(E12.6)
2010     FORMAT(F5.2)
2030     FORMAT(I5)
5        FORMAT(A)
            K=K-1
         IF(K.EQ.NUMPT) GOTO 3000
         WRITE(*,*) 'CHECK THE DATA - # POINTS MISMATCH'
         STOP
3000     RETURN
         END

        SUBROUTINE FFF1( NAME,YY,W,NEXPT,TIME)
C*************************************************
C       WHERE                                    *
C       DATE      IS THE DATE OF MEASUREMENT     *
C       DATAB     IS THE NUMBER OF DATA BLOCKS   *
C       F1        IS THE DUMMY STRING            *
C       ID        IS THE IDENTIFICATION STRING   *
C       K         IS THE NUMBER OF POINT         *
C       L9        IS THE RELATIVE SMOOTHING      *
C       MODE      IS EQUAL TO 0 MEAN FORWARD     *
C                 FOURIER TRANSFORM
C       MAG       IS THE MAGNITUDE OF Xo*Ht(f)   *
C       NAME4     IS THE NAME OF THE TARGET DATA *
C       N         IS THE NUMBER OF POINTS        *
C       NUMPT     IS THE SAME AS K               *
C       NEXPT     IS THE POWER-OF-2 EXPONENT     *
C                 WILL BE GENERATED BY PREFFT    *
C                 SET TO -1 INDICATES ERROR      *
C       R         IS THE REAL PART OF THE INVERSE *
C                 FOURIER TRANSFORM OF Xo*Ht(f)  *
C       SUBAV     IS THE NUMBER OF SUB AVERAGE   *
C       T         IS THE TIME INTERVAL           *
C       TIME      IS THE TIME INTEVAL            *
C       TITLE     IS THE TITLE NAME FOR PLOTTING *
C       TUT       IS THE FREQUENCY DOMAIN OF Xo*H(f)*
C       TYPE      IS THE WAVEFORM TYPE           *
C       W         IS THE COMPLEX EXPONENTIAL ARRAY *
C                 GENERATE BY PREFFT SUBROUTINE  *
C       Xo*Ht(f)  IS THE GAUSSIAN WEIGHTED COSINE *
C       XS        IS THE STRING FOR PLOTTING ROUTINE*
C                 FOR THE HEADER                 *
C                 DRIVER                         *
C       Y4        IS THE CALIBRATION SPHERE      *
C                 SUBTRACTED WAVEFORM (FREQUENCY) *
C                                                *
C       Y5        IS THE TARGET SUBTRACTED WAVEFORM *
C                 FREQUENCY DOMAIN               *
C                                                *
C       YSC       IS THE COMPUTED SCATTERED ELECTRIC*
C                 FIELD FOR THE CANONICAL TARGET *
C       Y1        IS THE CALIBRATION MEASUREMENT *
C       Y2        IS THE BACKGROUND MEASUREMENT  *
```

```
C     Y3        IS THE TARGET MEASURED            *
C     YS        IS THE SAME AS XS                 *
C     YN        IS THE YES OR NO STRING           *
C               METER                             *
C     XMI1      IS THE MINIMUM VALUE              *
C     XMA1      IS THE MAXIMUM VALUE              *
C**************************************************
      CHARACTER*16 NAME
      CHARACTER*64 ID,TYPE,DATE,TITLE,XS,YS,F1
      COMPLEX W(1025),YY(1025)
      REAL T
      REAL Y(1024),TIME,T2,XMINS,MAG(1024),XMAG
      INTEGER NUMPT,DATAB,SUBAV,MODE,N
C**************************************************
      CALL INPP( NAME,ID,TYPE,DATE,F1,SUBAV,T2,DATAB,NUMPT,Y)
      TIME=T2/(FLOAT(NUMPT-1))*1E-9
      XMINS=0.0
      N=NUMPT
      CALL  PRETIM(TITLE,XS,YS)
      CALL  P11(TITLE,N,XMINS,T2,Y,XS,ID,YS)
C     SET MODE EQUAL TO 0 TO FIND FOURIER SERIES
      MODE=0
      CALL  PREFFT (N,MODE,NEXPT,W)
             DO 1234 I =1 ,NUMPT
                    YY(I)=CMPLX( Y(I),0.0  )
1234         CONTINUE
      T=TIME
      CALL FFT (N,MODE,T,NEXPT,W,YY)
C     CHANGING THE OUTPUT  YY TO MAGNITUDE AND PHASE
      N=NUMPT
         CALL PREFRE(TITLE,XS,YS)
         CALL MAGNITUD(YY,N,MAG)
         XMAG=1.0/TIME/2.0*1E-9
         CALL P11(TITLE,N,XMINS,XMAG,MAG,XS,ID,YS)
      RETURN
      END

C***************************************************************
C                                                              *
C       FFT PROGRAM   FROM THE DIGITAL SPECTRAL ANALYSIS       *
C       WITH APLICATIONS  PRENTICE-HALL,INC..                  *
C       ENGLEWOOD CLIFF,NEW JERSEY 07632 1987                  *
C          S.LAWRENCE MARPLE                                   *
C                                                              *
C***************************************************************
      SUBROUTINE PREFFT (N,MODE,NEXP,W)
C
C   Input Parameters:
C
C     N     - Number of data samples to be processed (integer-must be a
C             power of two)
C     MODE  - Set to 0 for discrete-time Fourier series (Eq. 2.C.1) or
C             1 for inverse (Eq. 2.C.2)
C
C     Output Parameters:
C
```

```
C     NEXP  - Indicates power-of-2 exponent such that N=2**NEXP .
C             Will be set to -1 to indicate error condition if N
C             is not a power of 2 (this integer used by sub. FFT)
C     W     - Complex exponential array
C
C     Notes:
C
C     External array W must be dimensioned .GE. N by calling program.
C
      COMPLEX W(1024),C1,C2
      NEXP=1
5     NT=2**NEXP
      IF (NT .GE. N)  GO TO 10
      NEXP=NEXP+1
      GO TO 5
10    IF (NT .EQ. N)  GO TO 15
      NEXP=-1
      RETURN
15    S=8.*ATAN(1.)/FLOAT(NT)
      C1=CMPLX(COS(S),-SIN(S))
      IF (MODE .NE. 0)  C1=CONJG(C1)
      C2=(1.,0.)
      DO 20 K=1,NT
        W(K)=C2
20      C2=C2*C1
      RETURN
      END

C***********************************************************

      SUBROUTINE FFT (N,MODE,T,NEXP,W,X)
C
C     Input Parameters:
C
C     N,MODE,NEXP,W - See parameter list for subroutine PREFFT
C     T             - Sample interval in seconds
C     X             - Array of N complex data samples, X(1) to X(N)
C
C     Output Parameters:
C
C     X - N complex transform values replace original data samples
C         indexed from k=1 to k=N, representing the frequencies
C         (k-1)/NT hertz
C
C     Notes:
C
C     External array X must be dimensioned .GE. N by calling program.
C
      COMPLEX X(1024),W(1024),C1,C2
      MM=1
      LL=N
      DO 70 K=1,NEXP
        NN=LL/2
        JJ=MM+1
        DO 40 I=1,N,LL
          KK=I+NN
```

```
          C1=X(I)+X(KK)
          X(KK)=X(I)-X(KK)
40        X(I)=C1
        IF (NN .EQ. 1) GO TO 70
        DO 60 J=2,NN
          C2=W(JJ)
          DO 50 I=J,N,LL
            KK=I+NN
            C1=X(I)+X(KK)
            X(KK)=(X(I)-X(KK))*C2
50          X(I)=C1
60        JJ=JJ+MM
        LL=NN
        MM=MM*2
70      CONTINUE
      NV2=N/2
      NM1=N-1
      J=1
      DO 90 I=1,NM1
        IF (I .GE. J)  GO TO 80
        C1=X(J)
        X(J)=X(I)
        X(I)=C1
80      K=NV2
85      IF (K .GE. J)  GO TO 90
        J=J-K
        K=K/2
        GO TO 85
90      J=J+K
      IF (MODE .EQ. 0)  S=T
      IF (MODE .NE. 0)  S=1./(T*FLOAT(N))
      DO 100 I=1,N
100     X(I)=X(I)*S
      RETURN
      END

C***********************************************
C
      SUBROUTINE MAGNITUD (Y,NUMPT,MAG)
C
C***********************************************
       COMPLEX Y(1024),H
       REAL*4  MAG(1024)
       INTEGER N,K,NP,NUMPT
           DO 10 I=1 ,NUMPT/2
                       H=Y(I)
                       MAG(I)=CABS(H)
10     CONTINUE
       NUMPT=NUMPT/2
       RETURN
       END

C*************************************************

      SUBROUTINE PRETIM(TITLE,XS,YS)
      CHARACTER*64 TITLE,XS,YS
```

```
      TITLE='TIME SERIES PLOT'
      XS   ='X-AXIS TIME IN NS '
      YS   ='Y-AXIS MAGNITUDE IN VOLT'
      RETURN
      END

C**************************************************

      SUBROUTINE PREFRE(TITLE,XS,YS)
      CHARACTER*64 TITLE,XS,YS
      TITLE='SPECTRAL PLOT'
      XS='X-AXIS FREQUENCY IN GHZ'
      YS='Y-AXIS MAGNITUDE IN VOLT'
      RETURN
      END

      SUBROUTINE IFFF(N,T,X)
C***********************************************************
      COMPLEX W(1025),X(1025)
      REAL T
      INTEGER MODE
      MODE=1
      CALL PREFFT(N,MODE,NEXP,W)
      CALL FFT(N,MODE,T,NEXP,W,X)
      RETURN
      END

C**********************************************************************

      SUBROUTINE DECON3(K,YSC,Y4,Y5,ID,TYPE,DATE,F1,SUBAV,T2,DATAB)
C*****************************************************
C     COMPUTING THE TARGET RESPONSE TO THE INCIDENT *
C     WAVESHAPE FOR THE COMPUTED SPHERE. USING      *
C     RIAD'S METHOD OF OPTIMAL DECONVOLUTION.       *
C                                                   *
C     Xo*Ht(f)=Y5*CONJ(Y4)*Ysc/(Y4*CONJ(Y4)+L)      *
C                                                   *
C     WHERE                                         *
C     Xo(f)     IS THE INCIDENT WAVEFORM            *
C                                                   *
C     Ht(f)     IS THE TARGET TRANSFER FUNCTION     *
C                                                   *
C     Y4        IS THE CALIBRATION SPHERE           *
C               SUBTRACTED WAVEFORM (FREQUENCY)     *
C                                                   *
C     Y5        IS THE TARGET SUBTRACTED WAVEFORM   *
C               FREQUENCY DOMAIN                    *
C                                                   *
C     YSC       IS THE COMPUTED SCATTERED ELECTRIC  *
C               FIELD FOR THE CANONICAL TARGET      *
C     Y1        IS THE CALIBRATION MEASUREMENT      *
C     Y2        IS THE BACKGROUND MEASUREMENT       *
C     Y3        IS THE TARGET MEASURED              *
C     K         IS THE NUMBER OF POINT              *
C     NUMPT     IS THE SAME AS K                    *
C     R         IS THE REAL PART OF THE INVERSE     *
```

```
C               FOURIER TRANSFORM OF Xo*Ht(f)            *
C     MAG       IS THE MAGNITUDE OF Xo*Ht(f)             *
C     ID        IS THE IDENTIFICATION STRING             *
C     SUBAV     IS THE NUMBER OF SUB AVERAGE             *
C     DATAB     IS THE NUMBER OF DATA BLOCK              *
C     F1        IS THE DUMMY STRING                      *
C     NAME4     IS THE NAME OF THE TARGET DATA           *
C     XS        IS THE STRING FOR PLOTTING ROUTINE       *
C               FOR THE HEADER                           *
C     YS        IS THE SAME AS XS                        *
C     TITLE     IS THE TITLE NAME FOR PLOTTING           *
C     YN        IS THE YES OR NO STRING                  *
C     L9        IS THE SMOOTHING PARAMETER               *
C     TUT       IS THE FREQUENCY DOMAIN OF Xo*H(f)       *
C     TYPE      IS THE WAVEFORM TYPE                     *
C     XMI1      IS THE MINIMUM VALUE                     *
C     XMA1      IS THE MAXIMUM VALUE                     *
C     DATE      IS THE DATE OF MEASUREMENT               *
C*********************************************************
      COMPLEX       Y5(1025),Y4(1025),YSC(1024),TUT(1024),SUM
      COMPLEX       L1,L2,L3
      REAL          R(1024),L9,XMA1,XMI1,MAG(1024)
      REAL          TIME,T,T2
      INTEGER       K,NUMPT,SUBAV,DATAB
      CHARACTER*16 NAME4
      CHARACTER*64 ID,YS,XS,TITLE,TYPE,DATE,F1
      CHARACTER*1  YN
C*****************************************************
      T=T2/FLOAT(K-1)*1E-9
      SUM=(0.0,0.0)
      DO 111 I=1,K
            SUM=SUM+Y4(I)*CONJG(Y4(I))
111   CONTINUE
438   WRITE(*,*) '*******************************************'
      WRITE(*,*) ' ENTER RELATIVE SMOOTHING PARAMETER :     '
      READ (*,*) L9
      L1=CMPLX(L9,0.0)
      L2=SUM/CMPLX(FLOAT(K+1),0.0)*L1
      DO 1 I=1,K
            TUT(I)=Y5(I)*CONJG(Y4(I))*YSC(I)/(Y4(I)*CONJG(Y4(I))+L2)
1     CONTINUE
      TITLE='SPECTRAL PLOT'
      YS='DECONVOLUTION'
      XS='X-AXIS FREQUENCY IN GHZ'
      NUMPT=K
C******************************************************
      CALL MAGNITUD(TUT,NUMPT,MAG)
C******************************************************
      XMI1=0.0
      XMA1=1.0/T/2.0*1E-9
C******************************************************
      CALL P12(TITLE,NUMPT,XMI1,XMA1,MAG,XS,ID,YS,L9)
C******************************************************
      CALL IFFF(K,T,TUT)
C******************************************************
      DO 222 I=1,K
```

```
      R(I)=REAL(TUT(I))
222   CONTINUE
C**************************************************
      XMI1=0.0
      XMA1=T2
      XS='X-AXIS TIME IN NSEC'
      TITLE='TIME SERIES PLOT'
      YS='DECONVOLUTION'
C**************************************************
      CALL P12(TITLE,K,XMI1,XMA1,R,XS,ID,YS,L9)
C**************************************************
      YN='N'
      WRITE(*,*) '*****************************************************'
      WRITE(*,*) 'WOULD YOU LIKE TO SAVE THE DECONVOLUTION RESULT ?:    '
      READ(*,999) YN
      IF ( YN.EQ.'Y'.OR.YN.EQ.'y') GO TO 4376
      GOTO  4388
C**************************************************
4376  WRITE(*,*)'****************************************'
      WRITE(*,*)' ENTER FILE NAME FOR THE TIME DOMAIN DATA:  '
      READ(*,999) NAME4
      CALL OUT(NAME4,ID,TYPE,DATE,F1,SUBAV,T2,DATAB,K,R)
4388  YN='Y'
      WRITE (*,*) '****************************************************'
      WRITE (*,*) 'REPEAT WITH NEW SMOOTHING PARAMETER ? (Y/N): '
      READ(*,999) YN
      IF ( YN.EQ.'Y'.OR.YN.EQ.'y') GO TO 438
999   FORMAT(A)
      RETURN
      END

C*************************************************
C      F = ARRAY OF Y VARIABLE
C      XS= $ X-SCALE (STRING)
C      YS= $ Y-SCALE (STRING)
C      ID= IDENTIFICATION OF THIS GRAPH
C
                 SUBROUTINE P11(TITLE,NPTS,XMINN,XMAXX,FFF,XS,ID,YS)
c
C               RM-FORTRAN Subroutine to Plot a Solid Line with TITLE
C               Based on Program PLOT.FOR.    Dec 1987 by M.A. Morgan
C
C                   INPUT DATA FORMAT
C
C                                            TITLE - 64 Space Title
C                                            NPTS  - # Data Points
C                                            XMIN  - Real Min X value
C                                            XMAX  - Real Max X value
C                                            F(N)  - Input Data Array
C                                            IHC   - Hardcopy (1=Yes)
C
                              CHARACTER*1 YN,DUM
                           CHARACTER*64 TITLE,XS,ID,YS
                REAL X(1025),F(1025),XMIN,XMAX,FFF(1025),XMINN,XMAXX
                INTEGER*2 N,JROW,JCOL,ISYMBL,ITYPE,CYAN,WHITE,YELLOW
        IHC=0
```

```
      WRITE(*,*) ' WOULD YOU LIKE TO HAVE A HARD COPY: '
      READ(*,100) YN
      IF(YN.EQ.'Y'.OR.YN.EQ.'y') IHC=1
                         WHITE=7
                         CYAN=11
      DO 1111 I=1,NPTS
        F(I)=FFF(I)
 1111 CONTINUE
      XMIN=XMINN
      XMAX=XMAXX
                         YELLOW=14
                         N=NPTS
                DX=(XMAX-XMIN)/(NPTS-1.0)
                         FMIN=0.0
                         FMAX=0.0
                      DO 22 K=1,NPTS
                  X(K)=XMIN+(K-1.0)*DX
                IF(F(K).LT.FMIN) FMIN=F(K)
                IF(F(K).GT.FMAX) FMAX=F(K)
   22                    CONTINUE
                 IF(FMIN.GT.0.0) FMIN=0.0
                 IF(FMAX.LT.0.0) FMAX=0.0
C            Computing Scale Factors for Vertical Axis
                     ABSMIN=ABS(FMIN)
                     ABSMAX=ABS(FMAX)
                 YMAX=AMAX1(ABSMIN,ABSMAX)
                  NSCL=INT(LOG10(YMAX))
               IF (YMAX.LT.1.0) NSCL=NSCL-1
                      YSCL=10.**NSCL
                      FMIN=FMIN/YSCL
                      FMAX=FMAX/YSCL
                    ABSMIN=ABSMIN/YSCL
                    ABSMAX=ABSMAX/YSCL
                      DO 33 K=1,NPTS
   33                 F(K)=F(K)/YSCL
                         YMIN=0.0
                 IF(FMIN.EQ.0.0) GO TO 37
   35                  YMIN=YMIN+0.5
               IF(ABSMIN.GT.YMIN) GO TO 35
                  YMIN=YMIN*FMIN/ABSMIN
   37                    CONTINUE
                         YMAX=0.0
                 IF(FMAX.EQ.0.0) GO TO 41
   39                  YMAX=YMAX+0.5
               IF(ABSMAX.GT.YMAX) GO TO 39
                  YMAX=YMAX*FMAX/ABSMAX
   41                    CONTINUE
                        CALL QBEEP
                      CALL QSMODE(2)
         WRITE(*,*) 'Press RET to View Plot...RET Again to Clear Screen'
                  IF(IHC.NE.1) GO TO 42
      WRITE(*,*) 'A Hardcopy Will Be Made...Check that Printer is ON'
   42                    CONTINUE
                      READ(*,100) DUM
C         Calling GRAFMATIC Routines and Plotting Solid Line Graph
                         ITYPE=1
```

```
                        ISYMBL=-2
                    CALL QSMODE(16)
              CALL QPTXT(64,TITLE,YELLOW,16,23)
      CALL QPTXT(64,ID,CYAN,16,19)
      CALL QPTXT(64,XS,CYAN,16,20)
      CALL QPTXT(64,YS,CYAN,16,21)
         CALL QPLOT(110,540,20,240,XMIN,XMAX,YMIN,YMAX,0.,0.,0,1.,1.5)
              CALL QSETUP(0,CYAN,ISYMBL,CYAN)
      XMAJOR=XMAX/5.0
              CALL QXAXIS(XMIN,XMAX,XMAJOR,1,1,2)
      YMAJOR=1.0
              CALL QYAXIS(YMIN,YMAX,YMAJOR,1,1,2)
          JROW=14+(ABS(YMIN)/(ABS(YMAX)+ABS(YMIN)))*220
                JCOL=80-430*XMIN/(XMAX-XMIN)
             CALL QGTXT(3,'0.0',WHITE,JCOL,JROW,0)
                CALL QPTXT(1,'s',YELLOW,5,16)
                CALL QPTXT(1,'c',YELLOW,5,15)
                CALL QPTXT(1,'a',YELLOW,5,14)
                CALL QPTXT(1,'l',YELLOW,5,13)
                CALL QPTXT(1,'e',YELLOW,5,12)
                CALL QPTXT(1,'X',YELLOW,5,10)
                CALL QPTXT(2,'10',YELLOW,4,8)
                CALL QPTXT(1,'*',YELLOW,5,7)
                CALL QPTXT(1,'*',YELLOW,5,6)
                    CALL QCMOV(0,6)
                    WRITE(*,150) NSCL
                 CALL QTABL(ITYPE,N,X,F)
                    READ(*,100) DUM
                  IF(IHC.NE.1) GO TO 44
                      CALL PRTSC
 44                    CONTINUE
                    CALL QSMODE(2)
100                     FORMAT(A)
150                   FORMAT(4X,I3)
                         RETURN
                          END

C**************************************************
C
C     THIS SUBROUTINE PLOTS TWO GRAPHS ON THE SAME AXIS
C
C     F  =  Y1 ARRAY VARIABLE ( BLUE)
C     F1 =  Y2 ARRAY VARIABLE (GREEN)
C     AFTER PLOT THIS PROGRAM BE AWARE OF THE F1,F
C     IT WILL SENT BACK A SCALE F,F1
C
C     XS = $  X-SCALE
C     YS = $  Y-SCALE
C     ID = IDENTIFICATION
C**************************************************
           SUBROUTINE PLO8(TITLE,NPTS,XMINN,XMAXX,PUM,XS,YS,PUM1,ID)
C
C              RM-FORTRAN Subroutine to Plot a Solid Line with TITLE
C              Based on Program PLOT.FOR.    Dec 1987 by M.A. Morgan
C
C                   INPUT DATA FORMAT
```

```
C
C                                                     TITLE - 64 Space Title
C                                                     NPTS  - # Data Points
C                                                     XMIN  - Real Min X value
C                                                     XMAX  - Real Max X value
C                                                     F(N)  - Input Data Array
C                                                     IHC   - Hardcopy (1=Yes)
C
                                          CHARACTER*1 YN,DUM
                                      CHARACTER*64 TITLE,XS,ID,YS
                     REAL X(1025),F(1025),F1(1025),PUM(1025),PUM1(1025),XMINN,XMAXX
                       INTEGER*2 N,JROW,JCOL,ISYMBL,ITYPE,CYAN,WHITE,YELLOW,GREEN
C**************************************************
      WRITE(*,*) 'WOULD YOU LIKE TO HAVE A HARD COPY ?:  '
      READ(*,100) YN
      IHC=0
      IF(YN.EQ.'Y'.OR.YN.EQ.'y') IHC=1
      DO 159 I=1,NPTS
          F(I)=PUM(I)
          F1(I)=PUM1(I)
  159 CONTINUE
      XMIN=XMINN
      XMAX=XMAXX
C**************************************************
                                              WHITE=7
                                              CYAN=11
                                             YELLOW=14
      GREEN=2
                                              N=NPTS
                                     DX=(XMAX-XMIN)/(NPTS-1.0)
                                              FMIN=0.0
                                              FMAX=0.0
      FMIN1=0.0
      FMAX1=0.0
                                           DO 22 K=1,NPTS
                                        X(K)=XMIN+(K-1.0)*DX
                                     IF(F(K).LT.FMIN) FMIN=F(K)
                                     IF(F(K).GT.FMAX) FMAX=F(K)
      IF(F1(K).LT.FMIN1) FMIN1=F1(K)
      IF(F1(K).GT.FMAX1) FMAX1=F1(K)
   22 CONTINUE
      FMIN=AMIN1(FMIN,FMIN1)
      FMAX=AMAX1(FMAX,FMAX1)
                                      IF(FMIN.GT.0.0) FMIN=0.0
                                      IF(FMAX.LT.0.0) FMAX=0.0
C                             Computing Scale Factors for Vertical Axis
                                         ABSMIN=ABS(FMIN)
                                         ABSMAX=ABS(FMAX)
                                     YMAX=AMAX1(ABSMIN,ABSMAX)
                                       NSCL=INT(LOG10(YMAX))
                                    IF (YMAX.LT.1.0) NSCL=NSCL-1
                                          YSCL=10.**NSCL
                                          FMIN=FMIN/YSCL
                                          FMAX=FMAX/YSCL
                                        ABSMIN=ABSMIN/YSCL
                                        ABSMAX=ABSMAX/YSCL
```

```
                                   DO 33 K=1,NPTS
      F1(K)=F1(K)/YSCL
   33                              F(K)=F(K)/YSCL
                                     YMIN=0.0
                                IF(FMIN.EQ.0.0) GO TO 37
   35                               YMIN=YMIN+0.5
                               IF(ABSMIN.GT.YMIN) GO TO 35
                                 YMIN=YMIN*FMIN/ABSMIN
   37                                CONTINUE
                                     YMAX=0.0
                                IF(FMAX.EQ.0.0) GO TO 41
   39                               YMAX=YMAX+0.5
                               IF(ABSMAX.GT.YMAX) GO TO 39
                                 YMAX=YMAX*FMAX/ABSMAX
   41                                CONTINUE
                                    CALL QBEEP
                                   CALL QSMODE(2)
      WRITE(*,*) 'Press RET to View Plot...RET Again to Clear Screen'
                                 IF(IHC.NE.1) GO TO 42
      WRITE(*,*) 'A Hardcopy Will Be Made...Check that Printer is ON'
   42                                CONTINUE
                                   READ(*,100) DUM
C             Calling GRAFMATIC Routines and Plotting Solid Line Graph
                                      ITYPE=1
                                     ISYMBL=-2
                                   CALL QSMODE(16)
                             CALL QPTXT(64,TITLE,YELLOW,16,23)
      CALL QPTXT(64,ID,GREEN,16,19)
      CALL QPTXT(64,XS,WHITE,16,20)
      CALL QPTXT(64,YS,CYAN,16,21)
         CALL QPLOT(110,540,20,240,XMIN,XMAX,YMIN,YMAX,0.,0.,0,1.,1.5)
                              CALL QSETUP(0,CYAN,ISYMBL,CYAN)
                                   XMAJOR=XMAX/5.0
      CALL QXAXIS(XMIN,XMAX,XMAJOR,1,1,2)
                                     YMAJOR=1.0
                            CALL QYAXIS(YMIN,YMAX,YMAJOR,1,1,2)
                        JROW=14+(ABS(YMIN)/(ABS(YMAX)+ABS(YMIN)))*220
                               JCOL=80-430*XMIN/(XMAX-XMIN)
                           CALL QGTXT(3,'0.0',WHITE,JCOL,JROW,0)
                              CALL QPTXT(1,'s',YELLOW,5,16)
                              CALL QPTXT(1,'c',YELLOW,5,15)
                              CALL QPTXT(1,'a',YELLOW,5,14)
                              CALL QPTXT(1,'1',YELLOW,5,13)
                              CALL QPTXT(1,'e',YELLOW,5,12)
                              CALL QPTXT(1,'X',YELLOW,5,10)
                              CALL QPTXT(2,'10',YELLOW,4,8)
                              CALL QPTXT(1,'*',YELLOW,5,7)
                              CALL QPTXT(1,'*',YELLOW,5,6)
                                   CALL QCMOV(0,6)
                                  WRITE(*,150) NSCL
                                CALL QTABL(ITYPE,N,X,F)
C***************************************
C     PLOT ANOTHER GRAPH
C***************************************
      GREEN=10
                                     ISYMBL=-2
```

```
      ITYPE=1
      CALL QSETUP(0,GREEN,ISYMBL,GREEN)
      CALL QTABL(ITYPE,N,X,F1)
      READ(*,100) DUM
      IF(IHC.NE.1) GO TO 44
      CALL PRTSC
 44   CONTINUE
      CALL QSMODE(2)
 100  FORMAT(A)
 150  FORMAT(4X,I3)
      RETURN
      END

C***************************************************
C      F = ARRAY OF Y VARIABLE
C      XS= $ X-SCALE (STRING)
C      YS= $ Y-SCALE (STRING)
C      ID= IDENTIFICATION OF THIS GRAPH
C
      SUBROUTINE P12(TITLE,NPTS,XMINN,XMAXX,FFF,XS,ID,YS,L9)
c
C      RM-FORTRAN Subroutine to Plot a Solid Line with TITLE
C      Based on Program PLOT.FOR.    Dec 1987 by M.A. Morgan
C
C          INPUT DATA FORMAT
C
C                 TITLE  - 64 Space Title
C                 NPTS   - # Data Points
C                 XMIN   - Real Min X value
C                 XMAX   - Real Max X value
C                 F(N)   - Input Data Array
C                 IHC    - Hardcopy (1=Yes)
C
      CHARACTER*1 YN,DUM
      CHARACTER*64 TITLE,XS,ID,YS,RELAT
      REAL X(1025),F(1025),XMIN,XMAX,FFF(1025),XMINN,XMAXX,L9
      INTEGER*2 N,JROW,JCOL,ISYMBL,ITYPE,CYAN,WHITE,YELLOW
      IHC=0
      WRITE(*,*) ' WOULD YOU LIKE TO HAVE A HARD COPY: '
      READ(*,100) YN
      IF(YN.EQ.'Y'.OR.YN.EQ.'y') IHC=1
      WHITE=7
      CYAN=11
      DO 1111 I=1,NPTS
      F(I)=FFF(I)
 1111 CONTINUE
      XMIN=XMINN
      XMAX=XMAXX
      YELLOW=14
      N=NPTS
      DX=(XMAX-XMIN)/(NPTS-1.0)
      FMIN=0.0
      FMAX=0.0
      DO 22 K=1,NPTS
      X(K)=XMIN+(K-1.0)*DX
      IF(F(K).LT.FMIN) FMIN=F(K)
```

```
      IF(F(K).GT.FMAX) FMAX=F(K)
22    CONTINUE
      IF(FMIN.GT.0.0) FMIN=0.0
      IF(FMAX.LT.0.0) FMAX=0.0
C     Computing Scale Factors for Vertical Axis
      ABSMIN=ABS(FMIN)
      ABSMAX=ABS(FMAX)
      YMAX=AMAX1(ABSMIN,ABSMAX)
      NSCL=INT(LOG10(YMAX))
      IF (YMAX.LT.1.0) NSCL=NSCL-1
      YSCL=10.**NSCL
      FMIN=FMIN/YSCL
      FMAX=FMAX/YSCL
      ABSMIN=ABSMIN/YSCL
      ABSMAX=ABSMAX/YSCL
      DO 33 K=1,NPTS
33    F(K)=F(K)/YSCL
      YMIN=0.0
      IF(FMIN.EQ.0.0) GO TO 37
35    YMIN=YMIN+0.5
      IF(ABSMIN.GT.YMIN) GO TO 35
      YMIN=YMIN*FMIN/ABSMIN
37    CONTINUE
      YMAX=0.0
      IF(FMAX.EQ.0.0) GO TO 41
39    YMAX=YMAX+0.5
      IF(ABSMAX.GT.YMAX) GO TO 39
      YMAX=YMAX*FMAX/ABSMAX
41    CONTINUE
      CALL QBEEP
      CALL QSMODE(2)
      WRITE(*,*) 'Press RET to View Plot...RET Again to Clear Screen'
      IF(IHC.NE.1) GO TO 42
      WRITE(*,*) 'A Hardcopy Will Be Made...Check that Printer is ON'
42    CONTINUE
      READ(*,100) DUM
C     Calling GRAFMATIC Routines and Plotting Solid Line Graph
      ITYPE=1
      ISYMBL=-2
      CALL QSMODE(16)
      CALL QPTXT(64,TITLE,YELLOW,16,23)
      CALL QPTXT(64,ID,CYAN,16,19)
      CALL QPTXT(64,XS,CYAN,16,20)
      CALL QPTXT(64,YS,CYAN,16,21)
      RELAT='RELATIVE SMOOTH ='
      CALL QPTXT(64,RELAT,CYAN,16,18)
      CALL QQNPUT(265,253,L9,3)
      CALL QPLOT(110,540,20,240,XMIN,XMAX,YMIN,YMAX,0.,0.,0,1.,1.5)
      CALL QSETUP(0,CYAN,ISYMBL,CYAN)
      XMAJOR=XMAX/5.0
      CALL QXAXIS(XMIN,XMAX,XMAJOR,1,1,2)
      YMAJOR=1.0
      CALL QYAXIS(YMIN,YMAX,YMAJOR,1,1,2)
      JROW=14+(ABS(YMIN)/(ABS(YMAX)+ABS(YMIN)))*220
      JCOL=80-430*XMIN/(XMAX-XMIN)
      CALL QGTXT(3,'0.0',WHITE,JCOL,JROW,0)
```

```
      CALL QPTXT(1,'s',YELLOW,5,16)
      CALL QPTXT(1,'c',YELLOW,5,15)
      CALL QPTXT(1,'a',YELLOW,5,14)
      CALL QPTXT(1,'l',YELLOW,5,13)
      CALL QPTXT(1,'e',YELLOW,5,12)
      CALL QPTXT(1,'X',YELLOW,5,10)
      CALL QPTXT(2,'10',YELLOW,4,8)
      CALL QPTXT(1,'*',YELLOW,5,7)
      CALL QPTXT(1,'*',YELLOW,5,6)
      CALL QCMOV(0,6)
      WRITE(*,150) NSCL
      CALL QTABL(ITYPE,N,X,F)
      READ(*,100) DUM
      IF(IHC.NE.1) GO TO 44
      CALL PRTSC
 44   CONTINUE
      CALL QSMODE(2)
 100  FORMAT(A)
 150  FORMAT(4X,I3)
      RETURN
      END

      SUBROUTINE PLO77(Y1,Y5,T2,NUMPT,ID1)
C************************************************
C     Y2 = Y5 = BACKGROUND DATA ARRAY          *
C     Y1      = SCATTERED DATA ARRAY           *
C************************************************
      INTEGER NUMPT,DATAB,SUBAV,F1,NSHIF,SET
      REAL Y1(1024),Y2(1024),Y3(1024),Y4(1024),Y5(1024)
      REAL XMIN,XMAX
      CHARACTER*16 NAME1,NAME2,NAME3,QN
      CHARACTER*1 YN
      CHARACTER*64 YS,XS,ID,TITLE,DATE
      CHARACTER*64 ID1,TYPE1
      CHARACTER*64 ID2,TYPE2
      CHARACTER*64 ID3,TYPE3
      REAL TIME,T2
      WRITE(*,*) '*****************************************************'
      WRITE(*,*) 'SHIFTING SCATTERED WAVEFORM & SUBTRACTING BACKGROUND'
      WRITE(*,*) '*****************************************************'
      XMIN=0.0
      XMAX=T2
      DO 22 K=1,NUMPT
 22   Y2(K)=Y5(K)
      SET=0
C************************************************
 33   CALL SHIFT(TITLE,YS,XS,ID)
C************************************************
      CALL PLO8(TITLE,NUMPT,XMIN,XMAX,Y1,XS,YS,Y2,ID)
C************************************************
C     SHIFT THE CURVE
C************************************************
      CALL NSHIFT(NSHIF)
      IF(NSHIF.EQ.0) GO TO 40
      CALL GOOD(Y1,Y2,NSHIF,NUMPT,SET)
      GO TO 33
```

```
   40 CONTINUE
C*************************************************
C     SUBTRACT BACKGROUND FROM THE WAVEFORM  ****
C*************************************************
      DO 44 I=1,NUMPT
   44 Y3(I)=Y1(I)-Y2(I)
      CALL SUBTR(TITLE,XS,YS)
      CALL P11(TITLE,NUMPT,XMIN,XMAX,Y3,XS,ID1,YS)
      WRITE(*,*) 'DO YOU WANT TO RE-SHIFT AND SUBTRACT ? (Y/N):    '
      READ(*,300) YN
      IF(YN.EQ.'y'.OR.YN.EQ.'Y') GO TO 33
      DO 77 I=1,NUMPT
   77 Y1(I)=Y3(I)
  300 FORMAT(A)
      RETURN
      END

C*************************************************

      SUBROUTINE NSHIFT(NSHIF)
      INTEGER NSHIF
      WRITE(*,*) '*******************************************'
      WRITE(*,*)
      WRITE(*,*) 'SCATTERED WAVEFORM SHIFTING:  '
      WRITE(*,*)
      WRITE(*,*) '+n shifts scat waveform to the RIGHT n-points'
      WRITE(*,*) '-n shifts scat waveform to the LEFT  n-points'
      WRITE(*,*) ' 0 DISPLAYS subtracted waveform  '
      WRITE(*,*)
      WRITE(*,*) 'ENTER SHIFT (+/-INTEGER):  '
      READ(*,*) NSHIF
      RETURN
      END

C*************************************************

      SUBROUTINE SUBTR(TITLE,XS,YS)
      CHARACTER*64 TITLE,XS,ID,YS
      TITLE='SUBTRACTED WAVEFORM'
      XS='X-AXIS TIME IN NS'
      YS='Y-AXIS MAGNITUDE IN VOLT'
      RETURN
      END

C*************************************************

      SUBROUTINE SHIFT(TITLE,YS,XS,ID)
      CHARACTER*64 TITLE,YS,XS,ID
      TITLE='TIME SERIES PLOT'
      YS='BLUE-TARGET'
      ID='GREEN-BACKGROUND'
      XS='SHIFT-TARGET ONLY'
      RETURN
      END

C*************************************************
```

```
      SUBROUTINE GOOD(Y1,Y2,NSHIF,NUMPT,SET)
C                                                    *
C     PROGRAM TO SHIFT Y2 CURVE                      *
C
C     -   SHIFT  Y1 TO THE LEFT                      *
C     +   SHIFT  Y1 TO THE RIGHT                     *
C
C     Y1        IS THE SHIFTING  CURVE               *
C     Y2        IS THE BACKGROUND CURVE              *
C     NSHIF     IS THE NUMBER OF POINT TO SHIFT      *
C     SET       IS THE SUM OF THE POINT TO SHIFT     *
C               IN ORDER TO ZERO PADDING             *
C     NUMPT     IS THE NUMBER OF POINT               *
C                                                    *
C*****************************************************
      REAL Y1(1024),Y2(1024),Y3(1024)
      INTEGER   NSHIF,NUMPT,SET
 5    IF( NSHIF.LT.0 ) THEN
              NSHIF=-1*(NSHIF)
              DO 20 I=NSHIFT,NUMPT-NSHIF-SET
                    Y3(I)=Y1(I+NSHIF)
 20           CONTINUE
              DO 10 I=NUMPT-NSHIF-SET,NUMPT
                    Y2(I)=0.0
                    Y3(I)=0.0
 10           CONTINUE
              DO 15 I=1,NSHIF+SET+1
                    Y2(I)=0.0
                    Y3(I)=0.0
 15           CONTINUE
         ELSE IF( NSHIF.GT.0 ) THEN
              N=NSHIF+SET+1
              DO 40 I=N,NUMPT
                     Y3(I)=Y1(I-NSHIF)
 40           CONTINUE
              DO 30 I=1,N
                    Y2(I)=0.0
                    Y3(I)=0.0
 30           CONTINUE
              DO 35 I=NUMPT-NSHIF-SET,NUMPT
                    Y2(I)=0.0
                    Y3(I)=0.0
 35           CONTINUE
       ELSE
       END IF
       SET=SET+NSHIF
       DO 50 I=1,NUMPT
            Y1(I)=Y3(I)
 50    CONTINUE
       RETURN
       END

      SUBROUTINE OUT(NAME,ID,TYPE,DATE,F1,SUBAV,T2,DATAB,NUMPT,Y)
C
C     ID $      = CHARACTER $ TARGET IDENTIFICATION
```

```
C      TYPE $    = TYPE OF WAVEFORM
C      DATE $    = DATE OF EXPERIMENT TAKE PLACE
C      F1        = DUMMY STRING
C      SUBAV     = SUB AVERAGE
C      NUMPT     = NUMPT OF POINT
C      DATAB     = DATA BLOCK
C      Y         = Y ARRAY
C      T2        = TIME WINDOW
         CHARACTER*64 ID,TYPE,DATE,F1
         CHARACTER*16 NAME
         REAL Y(1024),T2
         INTEGER  SUBAV,NUMPT,DATAB
         OPEN(UNIT=10,FILE=NAME,STATUS='UNKNOWN')
         WRITE(10,*) ID
         WRITE(10,*) TYPE
         WRITE(10,*) DATE
         WRITE(10,*) F1
         WRITE(10,22) SUBAV
         WRITE(10,21) T2
         WRITE(10,22) DATAB
         WRITE(10,22) NUMPT
                 DO 10 I=1,NUMPT
                     WRITE(10,20) Y(I)
  10             CONTINUE
  20    FORMAT(E12.6)
  21    FORMAT(F5.2)
  22    FORMAT(I5)
  30    FORMAT(A)
        CLOSE(10)
        RETURN
        END
```

APPENDIX B. THE ACQUISITION QUICK BASIC PROGRAM

The following computer program acquires the measurement data from the HP 54120T as described in chapter 5. The program is written in QUICK BASIC

```
1000 REM *******************************************
1020 REM *   MODIFY 4/22/88                        *
1040 REM *   WRITTEN BY LT.JG SOONPUEN SOMAPEE     *
1060 REM *   ACQUISITION                           *
1080 REM *                                         *
1240 REM *                                         *
1321 REM *******************************************
1322 D$=DATE$
1324 DUMMY$=TIME$:PRINT "DATE.....";D$:PRINT "TIME.....";DUMMY$
1325 COLOR 7,1
1326 PRINT "WELCOME TO ACQUISITION PROGRAM"
1327 PRINT "THERE ARE 3 TYPE OF MEASUREMENT"
1328 PRINT "1. BACKGROUND......WAVETYPE"
1329 PRINT "2. CALIBRATE.......WAVETYPE"
1330 PRINT "3. TARGET..........WAVETYPE"
1341 PRINT " PRESS THE FOLLOWING KEY "
1342 PRINT " L    ( USED DATA  LAST RESPONSE)............."
1343 PRINT " D    ( SYSTEM DEFAULT )................"
1344 V$=INKEY$:IF V$="" THEN 1344
1345 IF V$="L" OR V$ ="l" THEN 1960
1346 IF V$="d" OR V$ ="D" THEN 1360
1347 GOTO 1344
1360 REM *******************************************
1380 REM *    DEFAULT                              *
1400 REM *******************************************
1420 OPEN "DATA.DAT" FOR OUTPUT AS #1
1440   P1=512
1460   PRINT #1,P1
1480   N=1024
1500   PRINT#1,N
1520   ND=1
1540 PRINT#1,ND
1560 T$="4 inch Thin wire"
1580 PRINT#1,T$
1600 D$=DATE$
1620 PRINT#1,D$
1640 T2=1.563E-10
1660 PRINT#1,T2
1680 W$="Background Wave Type"
1700 PRINT#1,W$
1720 F1=20
1740 PRINT#1,F1
1760 S1=16
1780 PRINT#1,S1
1800 G1=S1
1820 DSE$="Backg"
1840 PRINT#1,DSE$
```

```
1860 DUMMY$=TIME$
1880 PRINT#1,DUMMY$
1900 DELAY1=38
1920 PRINT#1,DELAY1
1940 CLOSE#1
1960 REM ********************************************
1980      REM Optionally include the following declarations
2000      REM in your program.
2020      REM They provide appropriate mnemonics by which
2040      REM to reference commonly used values.
2060      REM Some mnemonics (GET%, ERR%,
2080      REM END%, ATN%) are preceded by "B" in order to
2100      REM distinguish them from
2120      REM BASICA keywords.
2140      REM
2160      REM GPIB Commands
2180      UNL% = &H3F      ' GPIB unlisten command
2200      UNT% = &H5F      ' GPIB untalk command
2220      GTL% = &H1       ' GPIB go to local
2240      SDC% = &H4       ' GPIB selected device clear
2260      PPC% = &H5       ' GPIB parallel poll configure
2280      BGET% = &H8      ' GPIB group execute trigger
2300      TCT% = &H9       ' GPIB take control
2320      LLO% = &H11      ' GPIB local lock out
2340      DCL% = &H14      ' GPIB device clear
2360      PPU% = &H15      ' GPIB ppoll unconfigure
2380      SPE% = &H18      ' GPIB serial poll enable
2400      SPD% = &H19      ' GPIB serial poll disable
2420      PPE% = &H60      ' GPIB parallel poll enable
2440      PPD% = &H70      ' GPIB parallel poll disable
2460      REM
2480      REM GPIB status bit vector
2500      REM global variable IBSTA% and wait mask
2520      BERR% = &H8000   ' Error detected
2540      TIMO% = &H4000   ' Timeout
2560      BEND% = &H2000   ' EOI or EOS detected
2580      SRQI% = &H1000   ' SRQ detected by CIC
2600      RQS% = &H800     ' Device needs service
2620      CMPL% = &H100    ' I/O completed
2640      LOK% = &H80      ' Local lockout state
2660      REM% = &H40      ' Remote state
2680      CIC% = &H20      ' Controller-In-Charge
2700      BATN% = &H10     ' Attention asserted
2720      TACS% = &H8      ' Talker active
2740      LACS% = &H4      ' Listener active
2760      DTAS% = &H2      ' Device trigger state
2780      DCAS% = &H1      ' Device clear state
2800       REM
2820      REM Error messages returned in global variable IBERR%
2840      EDVR% =  0       ' DOS error
2860      ECIC% =  1       ' Function requires GPIB-PC to be CIC
2880      ENOL% =  2       ' Write function detected no Listeners
2900      EADR% =  3       ' Interface board not addressed correctly
2920      EARG% =  4       ' Invalid argument to function call
2940      ESAC% =  5       ' Function requires GPIB-PC to be SAC
2960      EABO% =  6       ' I/O operation aborted
```

```
2980      ENEB% =  7       ' Non-existent interface board
3000      EOIP% = 10       ' I/O operation started before previous
3020      REM                operation completed
3040      ECAP% = 11       ' No capability for operation
3060      EFSO% = 12       ' File system operation error
3080      EBUS% = 14       ' Command error during device call
3100      ESTB% = 15       ' Serial poll status byte lost
3120      ESRQ% = 16       ' SRQ remains asserted
3140      REM
3160      REM EOS mode bits
3180      BIN% = &H1000    ' Eight bit compare
3200      XEOS% = &H800    ' Send EOI with EOS byte
3220      REOS% = &H400    ' Terminate read on EOS
3240      REM
3260      REM Timeout values and meanings
3280      TNONE% =   0     ' Infinite timeout (disabled)
3300      T10US% =   1     ' Timeout of 10 us (ideal)
3320      T30US% =   2     ' Timeout of 30 us (ideal)
3340      T100US% =  3     ' Timeout of 100 us (ideal)
3360      T300US% =  4     ' Timeout of 300 us (ideal)
3380      T1MS% =    5     ' Timeout of 1 ms (ideal)
3400      T3MS% =    6     ' Timeout of 3 ms (ideal)
3420      T10MS% =   7     ' Timeout of 10 ms (ideal)
3440      T30MS% =   8     ' Timeout of 30 ms (ideal)
3460      T100MS% =  9     ' Timeout of 100 ms (ideal)
3480      T300MS% = 10     ' Timeout of 300 ms (ideal)
3500      T1S% =    11     ' Timeout of 1 s (ideal)
3520      T3S% =    12     ' Timeout of 3 s (ideal)
3540      T10S% =   13     ' Timeout of 10 s (ideal)
3560      T30S% =   14     ' Timeout of 30 s (ideal)
3580      T100S% =  15     ' Timeout of 100 s (ideal)
3600      T300S% =  16     ' Timeout of 300 s (ideal)
3620      T1000S% = 17     ' Timeout of 1000 s (maximum)
3640      REM
3660      REM Miscellaneous
3680      S% = &H8         ' Parallel Poll sense bit
3700      LF% = &HA        ' Line feed character
3720      REM
3740      REM Application program variables passed to
3760      REM GPIB functions
3780      REM
3800      CMD$ = SPACE$(10)        ' command buffer
3820      RD$ = SPACE$(255)        ' read data buffer
3840      WRT$ = SPACE$(255)       ' write data buffer
3860      BNAME$ = SPACE$(7)       ' board name buffer
3880      BDNAME$ = SPACE$(7)      ' board or device name buffer
3900      FLNAME$ = SPACE$(50)     ' file name buffer
3920 REM ****************************************
3940 CLS
3960 DIM W(1500),RRR(1500)
3980 DIM RDSS$(1024)
4000 SCREEN 0
4020 COLOR  7,1,3
4040 OPEN "C:DATA.DATA"  FOR INPUT AS #1
4060 BEEP:CLS
4080 REM ****************************************
```

```
4100 REM *       BEGIN MAIN PROGRAM                 *
4120 REM ****************************************
4140 LOCATE 5,8:PRINT"  DATA ACQUISITION      "
4160 REM ************************************************************
4180 REM *          T$  = TARGET IDENTIFICATION                        *
4200 REM *          W$  = WAVEFORM TYPE                                *
4220 REM *          D$  = DATE                                         *
4240 REM *          N   = NUMBER OF SUB-AVERAGE / WAVEFORM OR COUNT    *
4260 REM *          T2  = DPO SAMPLING INTERVAL (nsec)                 *
4280 REM *          S1  = DPO MAXIMUM DPO VERTICAL SCALE(M)            *
4300 REM *          ND  = NUMBER OF DATA BLOCKS                        *
4320 REM *          P1  = NUMBER OF POINT / WAVEFORM                   *
4340 REM *          F1  = DPO TIME WINDOW                              *
4360 REM *         DSE$ = FILE NAME OF THE DATA OUTPUT FILE            *
4380 REM *         DEV$ = NAME OF THE HPDEV IN THE CONFIGURATION FILE*
4400 REM *         DSO% = THE NUMBER ASSOCIATED WITH THE DEVICE         *
4420 REM *         VEW$ = STRINGS$ VARIABLE FOR SENDING TO HP54120     *
4440 REM *       DUMMY$ = TIME SET                                     *
4500 REM ************************************************************
4520 REM
4540 INPUT#1,P1
4560 PRINT "NUMBER OF POINT....................";P1
4580 INPUT#1,N
4600 PRINT "NUMBER OF SUB-AVERAGE(count).......";N
4620 INPUT#1,ND
4640 PRINT "NUMBER OF DATA BLOCK...............";ND
4660 INPUT#1,T$
4680 PRINT "TARGET ID..........................";T$
4700 INPUT#1,D$
4720 PRINT "DATE...............................";D$
4740 INPUT#1, T2
4760 PRINT "DPO SAMPLING INTERVAL..............";T2
4780 INPUT#1, W$
4800 PRINT "WAVEFORM  TYPE.....................";W$
4820 INPUT#1,F1
4840 PRINT "DPO TIMEWINDOW( NSEC ).........";F1
4860 INPUT#1, S1
4880 PRINT "DPO VERTICAL SCALE(M)..............";S1
4900 G1=S1
4920 INPUT#1, DSE$
4940 PRINT "DATA FILE NAME OUTPUT..............";DSE$
4960 INPUT#1,DUMMY$
4980 PRINT "AUTO TIME SET......................";DUMMY$
5000 INPUT#1,DELAY1
5020 PRINT "DELAY ( NSEC )...................";DELAY1
5040 CLOSE#1
5060 PRINT "................................................       "
5080 COLOR 7,1:PRINT " PRESS THE FOLLOWING KEY "
5100 PRINT " Y     ( CHANGE THE DATA.DAT )......................"
5120 PRINT " N     ( NO CHANGE )...................................."
5140 V$=INKEY$:IF V$="" THEN 5140
5160 IF V$="Y" OR V$ ="y" THEN 11260
5180 IF V$="N" OR V$ ="n" THEN 5220
5200 GOTO 5100
5220 DEV$="HPDEV"
5240 REM *****************************************
```

```
5260 REM * OPEN DEVICE AND RETURN THE UNIT       *
5280 REM * DESCRIPTOR ASSOCIATED  WITH THE GIVEN *
5300 REM * NAME                                  *
5320 REM *****************************************
5340 CALL IBFIND(DEV$,DSO%)
5360 IF DSO% < 0 THEN PRINT "ERROR IN IBFIND"
5380 REM *****************************************
5400 REM *  CLEAR SPECIFIED DEVICE               *
5420 REM *****************************************
5440 CALL IBCLR(DSO%)
5460 REM *****************************************
5480 GOSUB 6080
5500 CLS:BEEP
5520 PRINT W$+"  DATA STORAGE COMPLETE"
5540 PRINT " MEASURE OTHER WAVEFORM TYPE(Y/N) "
5560 Q$=INKEY$:IF Q$=" " THEN GOTO 5560
5580 IF Q$="N" OR Q$="n" THEN GOTO 5640
5600 IF Q$="Y" OR Q$="y" THEN GOTO 4040
5620 GOTO 5560
5640 STOP
5660 REM******************************************
5680 REM*          SUBPROGRAM                    *
5700 REM*                                        *
5720 REM *****************************************
5740 REM *     TEXT REFERENCE                    *
5760 REM *     "GURU II GPIB USER'S RESOURCE"    *
5780 REM *     "UTILITY FOR IBM PC"              *
5800 REM *     "TEKTRONIX 1986"                  *
5820 REM *****************************************
5840 REM *****************************************
5860 REM *  WRITE DATA FROM STRING  SUBROUTINE   *
5880 REM *****************************************
5900 CALL IBWRT(DSO%,VEW$)
5920 RETURN
5940 REM *****************************************
5960 REM *     TEXT REFERENCE                    *
5980 REM *     "GETTING STARTED GUIDE"           *
6000 REM *     "HP 54120T DIGITIZING "           *
6020 REM *     "OSCILLOSCOPE"                    *
6040 REM *     "HP COMPANY 1987"                 *
6060 REM *****************************************
6080 REM *****************************************
6100 REM *          SETUP THE DISPLAY            *
6120 REM *****************************************
6140 VEW$="*RST"
6160 GOSUB 5840
6180 REM *****************************************
6200 REM * CRT FORMAT 1 PROVIDE ONE DISPLAY AREA *
6220 REM * AND USE 8 DIVISIONS FOR THE FULL      *
6240 REM * SCALE RANGE                           *
6260 REM *****************************************
6280 VEW$=":DISPLAY:GRATICULE FRAME;FORMAT 1"
6300 GOSUB 5840
6320 REM *****************************************
6340 REM *   TURN TDR STEP GENERATOR ON          *
6360 REM *****************************************
```

```
6380 VEW$=":NETWORK:REFLECTION:PRESET"
6400 GOSUB 5840
6420 REM ******************************************
6440 REM *   PAGE 6-13 TURN ON CHANNEL 4          *
6460 REM ******************************************
6480 VEW$=":VIEW CHANNEL4"
6500 GOSUB 5840
6520 REM ******************************************
6540 REM *    PAGE 6-4 TURN OFF CHANNEL1          *
6560 REM ******************************************
6580 VEW$=":BLANK CHANNEL1"
6600 GOSUB 5840
6620 REM ******************************************
6640 REM *   SET UP CHANNEL RANGE AND OFFSET      *
6660 REM *   PAGE 17-6                            *
6680 REM *   TO BE  20 NANOSECONDS                *
6700 REM ******************************************
6720 VEW$=":TIMEBASE:RANGE 20 NS"
6740 GOSUB 5840
6760 REM PRINT "**************************************************"
6780 REM PRINT "SET THE VERTICAL SCALE RANGE FROM 8 TO 640 MV     "
6840 VERT=S1
6860 VEW$=":CHANNEL4:RANGE "+STR$(VERT)+"M"
6880 GOSUB 5840
7020 REM ******************************************
7040 REM *   TURN ON THE GRATITUDE                *
7060 REM ******************************************
7080 VEW$=":DISPLAY:GRATICULE GRID"
7100 GOSUB 5840
7120 VEW$=":ACQ:BAND LOW"
7140 GOSUB 5840
7160 REM ******************************************
7180 REM *   AVERAGING ON                         *
7200 REM ******************************************
7220 VEW$=":ACQUIRE:TYPE AVERAGE"
7240 GOSUB 5840
7260 REM ******************************************
7280 REM *    ASKING FOR BANDWIDTH                *
7300 REM ******************************************
7320 PRINT "****************************************"
7340 PRINT "THE DEFAULT BANDWIDTH IS 12.4GHZ       "
7360 PRINT "WOULD YOU LIKE TO CHANGE TO 20 GHZ (Y/N)"
7380 Q$=INKEY$: IF Q$=" " THEN GOTO 7380
7400 IF Q$="N" OR Q$="n" THEN GOTO 7500
7420 IF Q$="Y" OR Q$="y" THEN GOTO 7460
7440 GOTO 7380
7460 VEW$=":ACQ:BAND HIGH"
7480 GOSUB 5840
7500 RD$=SPACE$(255)
7520 VEW$="ACQUIRE:COUNT "+STR$(N)
7540 GOSUB 5840
7560 VEW$=":ACQUIRE:POINTS "+STR$(P1)
7580 GOSUB 5840
7600 REM ******************************************
7620 REM *    SET TIME DELAY                      *
7640 REM *                                        *
```

```
7660 REM ****************************************
7680 REM
7700 REM
7720 REM
7740 DELAY=DELAY1
7760 DUM$=STR$(DELAY)
7780 VEW$=":TIM:DEL "+DUM$+"NS"
7800 GOSUB 5840
7820 REM
7840 REM
7860 REM
7880 REM
7900 REM
7920 REM
7940 REM
7960 REM ****************************************
7980 REM HP 54120T SENT THE LAST BYTE RESPONSE "LF" ASII DECIMAL 10
8000 VEW$=":SYSTEM:HEADER OFF;:EOI ON"
8020 GOSUB 5840
8040 PRINT "***************************************************"
8320 REM ****************************************
8340 PRINT "I AM GOING TO DIGITIZE CHANNEL(Y/N)"
8360 Q$=INKEY$:IF Q$=" " THEN GOTO 8360
8380 IF Q$="Y" OR Q$="y" THEN GOTO 8420
8400 GOTO 8360
8420 PRINT TIME$
8560 REM *****************************************
8600 REM *    GET THE NUMBER AND STORE IN THE    *
8620 REM *    FILE                               *
8640 REM *****************************************
8660 DSEW$="C:C"
8680 FOR I=1 TO ND
8700 IF I<10 THEN DSFILE$=DSEW$+RIGHT$(STR$(I),1)+".WFM"
8720 IF I>9 THEN DSFILE$=DSEW$+RIGHT$(STR$(I),2)+".WFM"
8740 REM
8760 REM *****************************************
8780 REM *  DIGITIZE  CHANNEL 4                   *
8800 REM *****************************************
8820 VEW$=":DIGITIZE CHANNEL4"
8840 REM
8860 GOSUB 5840
8880 VEW$=":WAVEFORM:SOURCE WMEMORY4;FORMAT ASCII"
8900 GOSUB 5840
8920 VEW$=":WAVEFORM:DATA?"
8940 GOSUB 5840
8960 REM
8980 CALL IBRDF(DSO%,DSFILE$)
9000 CLOSE#1
9020 NEXT I
9040 REM *****************************************
9060 REM *   FINISH DIGITIZING                   *
9080 REM *****************************************
9100 PRINT TIME$
9120 DSFILE$="HEADER"
9140 VEW$=":WAV:PRE?"
9160 GOSUB 5840
```

```
9180 CALL IBRDF(DSO%,DSFILE$)
9200 PRINT "HEADER COMPLETE"
9220 GOSUB 10200
9240 PRINT "*********************"
9260 PRINT "WOULD YOU LIKE TO GET A HARD COPY?(Y/N)"
9280 Q$=INKEY$: IF Q$=" " THEN 9280
9300 IF Q$="N" OR Q$="n" THEN GOTO 10080
9320 IF Q$="Y" OR Q$="y" THEN GOTO 9360
9340 GOTO 9280
9360 REM ****************************************
9380 REM *   PLOT THE LAST WAVEFORM             *
9400 REM ****************************************
9420 VEW$=":HARD:SOUR FACT,WMEM4"
9440 GOSUB 5840
9460 PRINT "PLOT ? (Y/N)"
9480 Q$=INKEY$: IF Q$=" " THEN 9480
9500 IF Q$="N" OR Q$="n" THEN GOTO 9600
9520 IF Q$="Y" OR Q$="y" THEN GOTO 9560
9540 GOTO 9480
9560 VEW$=":PLOT?;"
9580 GOSUB 5840
9570 GOTO 9780
9600 REM *****************************************
9620 REM *   PRINTING                            *
9640 REM *****************************************
9660 VEW$=":HARDCOPY:PRINTER DEFAULT"
9680 GOSUB 5840
9700 VEW$=":HARDCOPY:PAGE AUTOMATIC"
9720 GOSUB 5840
9740 VEW$=":PRINT?;"
9760 GOSUB 5840
9780 BOAD$="GPIBO"
9800 CALL IBFIND(BOAD$,BOA%)
9820 CALL IBSIC(BOA%)
9840 VEW$="?Q!"
9860 CALL IBCMD (BOA%,VEW$)
9880 V%=1
9900 CALL IBGTS (BOA%,V%)
9920 PRINT "****************************************"
9940 PRINT "PLEASE WAIT UNTIL THE PLOTTER HAD FINISHED"
9960 PRINT "PLEASE RELEASE THE DPO TO LOCAL(Y)"
9980 Q$=INKEY$: IF Q$=" " THEN 9280
10000 IF Q$="Y" OR Q$="y" THEN GOTO 10040
10020 GOTO 9980
10040 CALL IBLOC(DSO%)
10060 CALL IBLOC(BOA%)
10080 RETURN
10100 REM ***************************************
10120 REM  SUBROUTINE READ DATA TO STRING REF. GURU D-47
10140 CALL IBRD(DSO%,RD$)
10160 RETURN
10180 REM ***************************************
10200 REM *     PROGRAM TO ARRANGE FORMAT FILE    *
10220 REM *     SCALE THE DATA                    *
10240 REM *                                       *
10260 REM *                                       *
```

```
10280 REM ****************************************
10300 OPEN "C:HEADER" FOR INPUT AS#1
10320 INPUT#1,A%,B%,C%,D%,E,F,G%,H,I,J,K
10340 PRINT "FORMAT ";A%
10360 PRINT "TYPE   ";B%
10380 PRINT "POINT  ";C%
10400 PRINT "COUNT  ";D%
10420 PRINT "XINCRE ";E
10440 PRINT "XOR    ";F
10460 PRINT "XREF   ";G%
10480 PRINT "YINC   ";H
10500 PRINT "YOR    ";IY
10520 PRINT "YREF   ";J
10540 PRINT "RANGE  ";K
10560 CLOSE#1
10580 REM   DSE$="C:C"
10600 FOR K=1 TO ND
10620 IF K<10 THEN DSFILE$=DSEW$+RIGHT$(STR$(K),1)+".wfm"
10640 IF K>9  THEN DSFILE$=DSEW$+RIGHT$(STR$(K),2)+".wfm"
10660 IF K>99 THEN DSFILE$=DSEW$+RIGHT$(STR$(K),3)+".wfm"
10680 OPEN DSFILE$  FOR INPUT  AS #1
10700 REM
10720 REM
10740 REM
10760 REM
10780 REM
10800 REM
10820        FOR I=1  TO C%
10840          INPUT #1,WFM
10860          W(I)= ((WFM-J)*H+IY)
10880       NEXT I
10885 SUMM=0.0
10890        FOR I=1 TO C%
10892          SUMM=SUMM+W(I)
10896        NEXT I
10897 REM ****************************************
10898 REM *        REMOVE MEAN                   *
10899 REM ****************************************
10900        SMEAN=SUMM/C%
10901 PRINT "MEAN =";SMEAN
10902         FOR I=1 TO C%
10903           RRR(I)=W(I)-SMEAN
10904         NEXT I
10910 CLOSE#1
10920 DSFILE$=DSE$+".WFM"
10940 OPEN "A",#1,DSFILE$
10960 PRINT#1,T$
10980 PRINT#1,W$
11000 PRINT#1,D$
11020 PRINT#1,DUMMY$
11040 PRINT#1,D%
11060 PRINT#1,USING "###.##";F1
11080 PRINT#1,ND
11100 PRINT#1,P1
11120 FOR I=1 TO C%
11140 PRINT#1,USING"##.#####                  ";RRR(I)
```

```
11160 NEXT I
11180 REM
11200 CLOSE#1
11220 NEXT K
11221 BEEP:BEEP
11240 RETURN
11260 REM ************************************************
11280 REM *    P1    = NUMBER OF POINTS                     *
11300 REM *    N     = NUMBER OF SUBAVERAGES                *
11320 REM *    ND    = NUMBER OF DATA BLOCKS                *
11340 REM *    T$    = TARGET IDENTIFICATION                *
11360 REM *    D$    = DATE OF MEASUREMENT                  *
11380 REM *    T2    = DPO SAMPLING INTERVAL                *
11400 REM *    W$    = WAVEFORM TYPE                        *
11420 REM *    F1    = DPO TIME WINDOW                      *
11440 REM *    S1    = DPO MAXIMUM SCALE                    *
11460 REM *    DSE$ = OUTPUT DATA FILE NAME                 *
11480 REM *    DUMMY$ = DUMMY $                             *
11500 REM ************************************************
11520  TS$=SPACE$(64)
11540  DUMMY$=SPACE$(64)
11560 KEY OFF:SCREEN 0,0,0
11580 OPEN "DATA.DAT" FOR INPUT AS #1
11600 INPUT#1,P1:INPUT#1,N:INPUT#1,ND:INPUT#1,T$:INPUT#1,D$:INPUT#1,T2
11620 INPUT#1,W$:INPUT#1,F1:INPUT#1,S1:INPUT#1,DSE$:INPUT#1,DUMMY$
11640 INPUT#1,DELAY1:CLOSE#1
11660 COLOR 7,1,3:CLS
11680 X=4:Y=16
11700 PRINT"PROGRAM DATA ACQUISTION "
11720 LOCATE X,Y+15:PRINT"PRESS THE FOLLOWING KEY"
11740 LOCATE X+2,Y:PRINT " < F > FILE NAME FOR DATA OUTPUT...!...";DSE$
11760 LOCATE X+3,Y:PRINT " < P > NUMBER OF POINTS................";P1
11780 LOCATE X+4,Y:PRINT " < A > NUMBER OF SUB AVERAGES..........";N
11800 LOCATE X+5,Y:PRINT " < B > NUMBER OF DATA BLOCKS...........";ND
11820 LOCATE X+6,Y:PRINT " < T > TARGET ID. ($)..............!...";T$
11840 LOCATE X+7,Y:PRINT " < D > DATE  ($).......................";D$
11860 LOCATE X+8,Y:PRINT " < W > WAVEFROM TYPE...............!...";W$
11880 LOCATE X+9,Y:PRINT " < K > DPO TIME WINDOW ................";F1
11900 LOCATE X+10,Y:PRINT " < S > DPO SAMPLING INTERVAL..........";T2
11920 LOCATE X+11,Y:PRINT " < R > RUN THE PROGRAM................"
11940 LOCATE X+12,Y:PRINT " < U > DPO MAXIMUM VERT MV............";S1
11960 LOCATE X+13,Y:PRINT " < X > AUTO TIME SET..................";DUMMY$
11980 LOCATE X+14,Y:PRINT " < G > DELAY TIME.....................";38
12000 Q$=INKEY$:IF Q$=" " THEN 12000
12020 IF Q$="G" OR Q$="g" THEN 12900
12040 IF Q$="F" OR Q$="f" THEN 12300
12060 IF Q$="P" OR Q$="p" THEN 12340
12080 IF Q$="A" OR Q$="a" THEN 12560
12100 IF Q$="B" OR Q$="b" THEN 12600
12120 IF Q$="T" OR Q$="t" THEN 12640
12140 IF Q$="D" OR Q$="d" THEN 12680
12160 IF Q$="W" OR Q$="w" THEN 12700
12180 IF Q$="S" OR Q$="s" THEN 12740
12200 IF Q$="K" OR Q$="k" THEN 12780
12220 IF Q$="R" OR Q$="r" THEN 4040
12240 IF Q$="U" OR Q$="u" THEN 12820
```

```
12260 IF Q$="X" OR Q$="x" THEN 12860
12280 GOTO 11720
12300 PRINT"INPUT FILE NAME($) EX..C:F"
12320 INPUT DSE$:GOTO 12940
12340 CLS:LOCATE 4,8:PRINT"NUMBER OF POINT "
12360 PRINT "A)...1024...............POINT."
12380 PRINT "B)....512...............POINT."
12400 PRINT "C)....256...............POINT."
12420 PRINT "D)....128...............POINT."
12440 Q$=INKEY$: IF Q$="" THEN 12440
12460 IF Q$="A" OR Q$="a" THEN P1=1024
12480 IF Q$="B" OR Q$="b" THEN P1=512
12500 IF Q$="C" OR Q$="c" THEN P1=256
12520 IF Q$="D" OR Q$="d" THEN P1=128
12540 GOTO 12940
12560 PRINT"INPUT NUMBER OF SUB-AVERAGE"
12580 INPUT N:GOTO 12940
12600 PRINT "INPUT NUMBER OF DATA BLOCK"
12620 INPUT ND :GOTO 12940
12640 PRINT"INPUT TARGET ID($)"
12660 INPUT T$:GOTO 12940
12680 PRINT"SET DATE":D$=DATE$:GOTO 12940
12700 PRINT"INPUT WAVEFROM  TYPE($)"
12720 INPUT W$:GOTO 12940
12740 PRINT"INPUT DPO SAMPLING INTERVAL"
12760 INPUT T2:GOTO 12940
12780 PRINT"DPO TIME WINDOW          "
12800 INPUT F1:GOTO 12940
12820 PRINT"INPUT DPO VERTICAL SCALE"
12840 INPUT S1:GOTO 12940
12860 PRINT"AUTO TIME"
12880 DUMMY$=TIME$:GOTO 12940
12900 PRINT"INPUT TIME DELAY"
12920 INPUT DELAY1:GOTO 12940
12940 OPEN "DATA.DAT" FOR OUTPUT AS#1
12960 PRINT#1,P1:PRINT#1,N:PRINT#1,ND:PRINT#1,T$:PRINT#1,D$
12980 PRINT#1,T2:PRINT#1,W$:PRINT#1,USING "###.##";F1:PRINT#1,S1
13000 PRINT#1,DSE$:PRINT#1,DUMMY$:PRINT#1,DELAY1
13020 CLOSE#1 : CLS
13040 REM
13060 GOTO 11700
```

APPENDIX C. CHARACTERISTICS OF LOW LOSS CABLE

Table 21. CHARACTERISTICS OF PORT RETURN CABLES

Impedance	50 ohms
Capacitance	26 pF/ft
Time delay	1.2 ns/ft
Velocity of propagation	85% of light velocity
Jacket withstand	1.0 kV
Center conductor	15 AWG solid
Minimum bend radius	1 inch
Dielectric constant	1.4
Dielectric withstand	1.0 kV

Table 22. INSERTION LOSS

0-4 GHz	less than 0.36 dB
4-8 GHz	less than 0.5 dB
8-12 GHz	less than 0.6 dB
12-16 GHz	less than 0.72 dB
16-18 GHz	less than 0.8 dB
18-26.5 GHz	less than 1.0 dB

LIST OF REFERENCES

1. E. M. Kennaugh and D. L. Moffatt, "Transient and Impulse Response Approximations," *Proc. IEEE,* 53, pp. 893-901, August 1965.

2. J. D. DeLorenzo, "Video Time-Domain Scattering Range," *NEREM Rec.*, Vol. 9, 1967.

3. J. M. Murchison and B.Falk, "A Range for Measuring the Short-Pulse Response of Scattering Objects," *NEREM Rec.*, Vol. 9, 1967.

4. C. E. Baum, *On the Singularity Expansion Method for the Solution of Electromagnetic Interaction Problems,* Air Force Weapons Laboratory Interaction Note 88, December 1971.

5. F. M. Tesche, "On the Analysis of Scattering and Antenna Problems using the Singularity Expansion Technique," *IEEE Trans. on Antennas Prop.* , Vol. AP-21, No. 1, pp. 53-62, January 1973.

6. M. L. Van Blaricum and R. Mittra, "A Technique for Extracting the Poles and Residues of a System Directly from Its Transient Response," *IEEE Trans. on Antennas Prop.*, Vol. AP-23, pp. 777-781, November 1975.

7. C. W. Hammond, *The Development of a Bistatic Electromagnetic Scattering Laboratory,* Master's Thesis, Naval Postgraduate School, Monterey, CA, 1980.

8. M. A. Morgan, "Time-Domain Scattering Measurements," *IEEE Antennas and Propagation Newsletter,* Vol. 6, pp. 5-9, August 1984.

9. M. A. Mariatequi, *Development, Calibration, and Evaluation of a Free Field Scattering Range,* Master's Thesis, Naval Postgraduate School, Monterey, CA, 1983.

10. J. R. Auton, T. L. Larry, and M. L. Van Blaricum, *Radar Target Identification and Characterization Using Natural Resonance Extraction*, Report CR-84-1309 on Office of Naval Research contract N00014-82-C-0079, General Research Corporation, Santa Barbara, CA, September 1984.

11. Brent W. McDaniel, *Calibration and Evaluation of A Free Field Scattering Range Using Wideband Pulse Amplification*, Master's Thesis, Naval Postgraduate School, Monterey, CA, 1985.

12. M. A. Morgan, B. W. McDaniel, "Transient Electromagnetic Scattering Data Acquisition and Signal Processing," *IEEE Trans. Inst. & Meas.*, Vol. 37, No. 2, pp. 263-267, June 1988.

13. *Getting Started Guide for HP 54120T Digitizing Oscilloscope*, Hewlett-Packard Company, Colorado Spring Division, 1987.

14. *Service Manual for HP 54120T 20 GHz Digitizing Oscilloscope*, Hewlett-Packard Company, Colorado Spring Division, 1987.

15. Alan Clements, *Microprocessor Systems Design 6800 Hardware, Software, and Interfacing*, PWS Computer Science Boston, MA, 1987.

16. *GURU II GPIB User's Resource Utility for the IBM PC*, Tektronix, INC. Beaverton, 1986.

17. S. M. Raid, "Impulse Response Evaluation Using Frequency Domain Optimal Compensation Deconvolution," *Midwest Symposium on Circuits and Systems*, University of Toledo, OH, August 1981.

18. S. Lawrence Marple, Jr., *Digital Spectral Analysis with Applications*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1987.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Department Chairman, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5100	1
4.	Professor M. A. Morgan, Code 62Mw Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5100	10
5.	Professor Richard W. Adler Code 62Ab Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5100	1
6.	Dr. John N. Entzminger Director, Tactical Technology Office Defense Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, VA 22209	1
7.	Dr. Dominick Giglio Tactical Technology Office Defense Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, VA 22209	1
8.	Dr. Rabiner Madan Code 1114SE Office of Naval Research 800 N. Quincy St. Arlington, VA 22209	1
9.	Professor K. M. Chen Department of Electrical Engineering and Systems Science Michigan State University East Lansing, MI 48824	1

10. Dr. M. L. Van Blaricum 1
General Research Corporation
5383 Hollister Ave.
Santa Barbara, CA 93160

11. Dr. David Lewis 1
AST Office
Defense Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

12. Dr. Bruce Z.Hollmann,Code F12 1
Naval Surface Warfare Center
Dahlgren, VA 22448

13. Royal Thai Navy 5
Bangkok
Thailand