NASA
Technical
Memorandum

NASA TM -103514

EDDY CURRENT INSPECTION OF GRAPHITE FIBER COMPONENTS

By G.L. Workman and C.C. Bryson

Materials and Processes Laboratory
Science and Engineering Directorate

October 1990

National Aeronautics and
Space Administration
George C. Marshall Space Flight Center

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.
NASA TM-103514

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Eddy Current Inspection of Graphite Fiber Components

5. Report Date
October 1990

6. Performing Organization Code

7. Author(s)
Dr. G.L. Workman* and C.C. Bryson

8. Performing Organization Report No.

10. Work Unit No.

9. Performing Organization Name and Address

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

11. Contract or Grant No.

13. Type of Report and Period Covered
Technical Memorandum

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, DC 20546

14. Sponsoring Agency Code

15. Supplementary Notes

Prepared by Materials and Processes Laboratory, Science and Engineering Directorate.

*Senior Research Scientist, University of Alabama in Huntsville.

16. Abstract

The recognition of defects in materials properties still presents a number of problems for nondestructive testing in aerospace systems. This project attempts to utilize current capabilities in eddy current instrumentation, artificial intelligence, and robotics in order to provide insight into defining geometrical aspects of flaws in composite materials which are capable of being evaluated using eddy current inspection techniques. The unique capabilities of E-probes and horseshoe probes for inspecting probes for inspecting graphite fiber materials have been WERE evaluated and appear to hold great promise once the technology development matures.

This preliminary phase report describes the initial results of modeling eddy current interactions with certain flaws in graphite fiber samples. ARE DESCRIBED

17. Key Words (Suggested by Author(s))

Nondestructive Evaluation, Expert Systems, Defect Recognition, Eddy Current Testing of Composites, Eddy Current Transducers

18. Distribution Statement

Unclassified – Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	52	NTIS

For sale by the National Technical Information Service, Springfield, VA 22161-2171

ACKNOWLEDGMENTS

This work has benefited from a number of helpful discussions with several researchers in the field including S. Vernon and J. Warren of the Naval Surface Weapons Laboratory, and B. Lempriere of Boeing Aerospace. Also, we are indebted to R. Holmes, W. De Weese and the Thiokol filament winding group for preparing our specimens and to Magnetics Corporation for the ferrite samples. Gratitude is also expressed to the students from the University of Alabama, Huntsville (UAH) who worked on this project, M. Wang and S. Lee.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

TECHNICAL MEMORANDUM

EDDY CURRENT INSPECTION OF GRAPHITE FIBER COMPONENTS

I. INTRODUCTION

Nondestructive testing of aerospace components provides a challenging research activity for the evaluation of new technologies. Graphite-based composites are becoming more accepted in critical structures which require a high strength-to-weight ratio. The primary disadvantages, at this time, are the lack of sensitivity of most eddy current transducers and the inability to characterize a fabricated part as to its intended service capability; i.e., given known defect information, what will be the service life of the part? Attempts to answer these questions can be approached in two steps. First one needs to characterize defects in the material using some nondestructive evaluation (NDE) technique and then be able to predict the service life of the component with the known defects. Metallic structures have traditionally been handled very well using NDE and fracture mechanics; however, there is still a lot of work to be done to be able to accomplish similar tasks with composite structures.

Since graphite is conducting to a small degree, i.e., the resistivity can vary anywhere from 500 to 1,000 $\mu\Omega$-cm, eddy current inspection techniques are applicable to determining certain flaws in graphite fiber-based structures. Uniformity in resistivity in the normally manufactured products currently presents difficulties; however, it is anticipated that these can be overcome.

A. Approaches to Eddy Current Signal Interpretation

Although the use of expert systems to interpret eddy current signals is very new, concepts using feature extraction for defect identification have been around for some time. The use of statistical pattern recognition, for example, has shown merit. An excellent report on this technique is given in reference 1. Their work was concerned with characterizing flaws in austenitic steels used in power plant environments, which is a significantly different material from graphite fibers. However, the analysis performed in that work shows the utility of feature abstraction and statistical techniques based on signal characterization of known defect types. Some of these ideas are also useful in developing knowledge-based reasoning systems for defect characterization.

This research effort has focused on the use of eddy current techniques for characterizing flaws in graphite-based, filament-wound cylindrical structures. A major emphasis has also been placed on incorporating artificial intelligence techniques into the signal analysis portion of the inspection process. Developing an eddy current scanning system using a commercial robot for inspecting graphite structures (and others) has been a goal in the overall concept and is essential for final implementation for expert systems interpretation. Manual scans, as performed in the preliminary work here, do not provide sufficiently reproducible eddy current signatures to be easily built into a real time expert system.

The expert systems approach to eddy current signal analysis requires that a suitable knowledge base exist in which correct decisions, as to the nature of a flaw, can be performed. In eddy current, or any other expert systems used to analyze signals in real time in a production environment, it is important to simplify computational procedures as much as possible. For that reason, we have initially chosen to utilize the measured resistance and reactance values for the preliminary aspects of this work. A simple computation, such as phase angle of the signal, is certainly within the real time processing capability of the computer system. In the work described here, there has been a balance between physical measurements and finite-element calculations of those measurements. As described earlier, the goal is to evolve into the most cost-effective procedures for maintaining the correctness of the knowledge base.

B. Eddy Current Transducer Characterization

Another major activity performed in this work was to develop eddy current transducers specifically designed to inspect graphite fiber components. A number of relevant papers have been published in the last few years defining some of the problems in using eddy current probes to inspect components containing graphite [2–10]. The E-probes and horseshoe probes developed here provide large signals for both cut tows and impact damage. Major increases in the coupling efficiency with carbon fibers and in the sensitivity of flaw detection are desirable. The use of finite element models to predict eddy current signatures has also been a major activity within this work. Using the Ansoft Maxwell® software, values of the impedance changes for various defects are computed using both the E-probe and the horseshoe probe. It is anticipated that verification of this procedure as heuristic inputs into the expert system being developed for interpretation of the eddy current signals will be a continuing activity, however, logistically it is still more economical than preparing physical models of all possible flaws for signature characterization.

II. EXPERIMENTAL

Several filament-wound graphite cylinders with prefabricated defects, as shown in figure 1, were prepared by Marshall Space Flight Center (MSFC) contractors in the Materials and Processes (M&P) Laboratory. Eddy current scans are then able to produce a representative signal for this type of defect. The types of eddy current probes currently available at MSFC were normally pancake type probes, which are useful for finding surface and subsurface cracks in metallic specimens. Initial measurements with the available probes indicated that very little coupling between the probe and the specimen was occurring. This observation motivated us to experiment with the finite element model to better understand the physics of the eddy current measurements.

The Ansoft Maxwell® software allows axisymmetric solution of electromagnetic fields in various geometries. The normal pencil probe used so often in eddy current testing does not work well with graphite fibers. Several factors are responsible for this ineffectiveness. Firstly, the overlap between the toroidal field emanating from the transducer is extremely small, resulting in a very small impedance change between air and the graphite fibers. Even a number of fibers located parallel to each in the filament-wound cylinder do not provide sufficient coupling to differentiate between the resistivity changes in going from air to the graphite sample.

Two other types of probes were used in this work. E-probes and horseshoe probes were also used with much better success. The E-probe can be modeled axis-symmetrically with the finite element programs so that a solution can be obtained. More importantly, the electromagnetic field is unidirectional and, in the proper orientation, provides excellent coupling with the fibers. The horseshoe probe is not axis-symmetric, but it has been modeled as half of an E-probe. Experiments verifying that half an E-probe can model a horseshoe probe allows the use of defect signatures to be computed using the Maxwell® software. After some experimentation with the software, we were able to model horseshoe probes by placing the excitation coils axisymmetrically and obtain solutions in that manner. At the time this report was written, most of the solutions obtained were for E-probes. We are now able to model horseshoe probes.

In order to verify the finite element model, several types of impedance plots were generated using the values of resistance and reactance calculated by the Ansoft software. For instance, table 1 shows the results from varying inspection frequency from 50 kilohertz to 5 megahertz, and figure 2 shows the corresponding impedance plane plot. This plot behaves normally, as would be expected from reference 2. Note that these calculations assume that the thickness of the material is much greater than the standard depth of penetration. Similar calculations of the materials points were made for ferrite and graphite. These results are shown in figure 3. The normal display for such a plot is rotated 90° counterclockwise. Again, the results fit the theory fairly well.

Samples of cylindrical components fabricated from resin 55A and fiber AS4-W-12K were provided by the Thiokol contractors who work at the filament winding facilities at MSFC. Eddy current measurements were obtained from these samples using standard pancake probes and the E-probe described below with the SmartEddy system and are plotted according to figure 4. The maximum specimen thickness available was only 0.168 in. Thus, at the lower frequencies, the penetration was too deep for the impedance plot to go to 1 on the vertical axis. Otherwise the physical measurements fit the calculated data from figure 2 very well.

The layout of defects embedded in the graphite samples is given in figure 1. Two different thicknesses were provided, 0.168 in and 0.085 in. Due to the low conductivity of the graphite, only the larger thickness was suitable for most of our measurements. The phase angle relationships expected between two different types of defects are shown in figure 5. Note that the phase angles represent unique signatures for a particular type of defect and can be used to perform defect identification.

A. Probe Design Considerations

The design and fabrication of an eddy current transducer requires consideration of the following parameters:

1. Probe type

2. Inspection frequency

3. Optimum size

4. Cross-section of the winding area

5. Optimum balance between wire size and number of turns.

Very useful discussions of these parameters have been found in the work by Vernon [10], Dodd [11], and the Chalk River Training Manual [12]. For overall performance, the inspection frequency of the transducer plus cable must be less than the resonant frequency, and the skin depth should be about 1.67 times the material thickness. The calculations in the following sections have been performed for the horseshoe probe designed in this research.

1. Probe Self-Inductance

Determination of the optimal operating frequency for the eddy current inspection of graphite fiber components is assisted by knowing the actual impedance as seen by the alternating current generator for the test. In general, the operating frequency of the inspection transducer should be around 20 percent of the resonance frequency. In order to come up with this information, we were able to calculate the self-inductance using Ampere's Law and then measure it for confirmation. Our transducers are not round or cylindrical, as normally considered in such calculations, so there was some question about possible errors in the calculation. Being able to measure the inductance of the final assembly was beneficial. Capacitance of the transducer-cable arrangement is even more difficult; however, we were able to measure that parameter also.

The typical values for our transducers are given by:

$$L = 4\ \pi\ k_m\ N^2\ A*10^{-10}/1$$

$$= 4*3.1416*3*7^2*80.6*10^{-10}/5.1$$

$$= 2.92 \text{ microhenrys}$$

$$k_m = \text{geometrical factor} = 3\quad ,$$

$$\text{Capacitance (measured)} = 87.6 \text{ picofarads} = 8.7*10^{-11} \text{ farads}$$

Note that the measured value for self inductance was determined to be 3.35 microhenrys or $3.35*10^{-6}$ henrys, which turned out to be reasonably close to the calculated value. That value is then used to determine the resonance frequency, which is given by:

$$f_r = 1/2\pi\sqrt{LC}$$

$$= \frac{1}{2*3.1416*\sqrt{3.35*10^{-6}*8.76*10^{-11}}}$$

$$= 9.3 \text{ Mhz}\quad .$$

Inspection frequencies up to 2 Mhz would then be acceptable for these transducers.

2. Calculation of Depth of Penetration for Graphite Fiber Components

The empirical relationship between resistivity and frequency derived by Vernon [10] for eddy current cup probes gives:

$$\rho = \frac{195.6*10^{-6}*r^2*f}{(\tan\,\theta_L + 0.158)^2}$$

$$= 17{,}303\ \mu\Omega\text{-cm} \quad ,$$

when $f = 4$ Mhz, $\theta_L = 60°$, and $r = 0.125$ mm. For 2-Mhz excitation, the depth of penetration would then be

$$\delta = 1.98\ \sqrt{\rho/f} = 1.98\ \sqrt{17{,}300/2*10^6}$$

$$= 0.18 \text{ inches}$$

For thinner samples, higher frequencies are required, and redesign of the probe, to have higher resonance frequency, will have to be done. Since our transducers are not cup probes, some of these calculations may require correction, although to date excellent correlation has been obtained.

Based upon these calculations, the eddy current E-probe and horseshoe probe were wound using seven turns of wire. The E-probe transducer was constructed using transformer cores supplied by Magnetics. Unfortunately, we were not given the magnetic permeability of the material. A major disadvantage of the E-probe is that two signals are normally obtained since there are actually two different electromagnetic fields. This means that the expert system analyzing the signals has to account for both signals. With this analysis requirement in mind, a horseshoe probe was fabricated to generate only one signal for each change in impedance occurring in scanning the part.

In order to fabricate our own horseshoe probe, an E-probe was cut in half and reassembled as a horseshoe probe, again using seven turns of wire. The signal obtained from scanning the samples should theoretically be identical to just one of the lobes of the E-probe. However, the coupling efficiency is proportional to the footprint (area) of the transducer, and the horseshoe probe is about 80 percent the area of half of the E-probe. Hence, the coupling efficiency is not quite as good, and correspondingly smaller signals are obtained.

III. RESULTS

Signals obtained by scanning the graphite samples fabricated at MSFC are presented in figures 6 through 10. These data were obtained using the SmartEddy for the data acquisition. The SmartEddy system provides decomposition into resistance and reactance for the time amplitude plots. Both impedance plane plots and amplitude displays of resistance and reactance are shown here. Impedance plane displays of the signals can be more informative in that the phase angle changes which occur for a given defect are consistent with the type of defect and the defect depth. However, for the purposes of the expert system analysis, the final determination for whether reactance and resistance changes alone can be used for rapid interpretation of the data has not been concluded at this time.

Two major points need to be discussed here. First, the resistivity of graphite fibers is not consistent throughout the sample. Hence, the materials point keeps changing throughout a scan. This will have to be taken care of during the expert system analysis. Secondly, the theory and most eddy current specialists do not believe that eddy currents can provide an indication of delaminations. Our samples were fabricated using a graphite knot for a delamination (figs. 4 and 6), and we were able to observe the embedded delamination (although the signal was small). A kissing debond, which is obviously more difficult to fabricate, would not have given such an obvious indication of a defect. We currently feel that we were able to see a volume change since the delamination was caused by inserting a dielectric material between two graphite layers.

An ultrasonic C-scan of one of the samples provided (Fig. 11) was also taken in order for comparison with the eddy current data. Note that the delamination is quite evident in the through transmission scan. The other defects are easily distinguished and there is a lot of noise in the image. The eddy current image of this specimen would be much more useful once all systems are available for use.

A. Finite Element Modeling

The Ansoft Maxwell® software was used to compute the interaction between the various eddy current transducers and the materials. The program allows one to generate a three-dimensional model and solve for the electromagnetic fields in an axisymmetric environment. A schematic of the user interface to the program is shown in figure 12. The results can only be displayed in two dimensions. However, modules were developed for simulation of scans across a particular defect. For instance, figures 13 and 14 show simulated values for resistance and reactance when the probe is displaced away from the defect as compared with when sitting over the defect for two different flaws. Note that the impedance changes are small, nonetheless they are observable.

In correlation with the finite element modeling, it is informative to consider the physical measurements made with an E-probe and a commercially acquired horseshoe probe. The lift-off lines shown in figure 15 indicate the relative magnitudes of the signals obtained from each transducer. That difference in magnitude was a major driving force to build a horseshoe probe with good coupling characteristics and to work on a technique to rotate the transducer to time-average multidirectional orientation of the fibers in a part. Figure 16 shows a SmartEddy display obtained from the probe. Note the varying materials point due to manual rotation of the probe. Robotically scanning and spinning should provide a time-averaged material point and alleviate that problem.

It is anticipated that for the final expert system implementation, a series of these calculations would provide the simulated time-varying amplitude impedance of the eddy current transducer moving over the defect. This feature would then be added to the knowledge base to provide predictive capability for that type of flaw. The time-averaged data will also have to be correlated with the finite-element models.

B. Expert Systems Approach to Eddy Current Signal Analysis

The goal in using an expert system for interpetation of eddy current signals relies upon the use of heuristics and efficient strategies for sorting through the decision trees to identify the type of defect being observed. If these goals can be satisfied, then the arduous computation required for statistical pattern recognition can be reserved for the difficult interpretations, and the envisioned expert system can handle the real time inspection requirements.

To demonstrate the foundation upon which statistical interpetations are based, it is interesting to reconsider the procedures described in reference 1. Feature extraction, as described in that reference are also useful considerations for applying expert systems. In considering an alternative methodology, figure 17 displays the expert systems model planned here. There are several features contained in this concept which are significant. In the DATA INPUT section of figure 17, note that the use of a robotic scanning system will enable consistent time-varying amplitudes to be obtained. Manual scanning, as has been performed in this phase of the work, does not provide consistent results. The implementation is being performed in several phases as instrumentation becomes available. An anticipated application of neutral networks is obvious to the defect identification problems in eddy current inspection [13]. Figure 18 shows the expert system architecture for eddy current analysis. The build-up of the knowledge base is also occurring in stages using finite element models and fabricated defects to generate decision criteria for the expert system. The only other work reported previously dealt with metallic components and primarily developed a heuristic for determining crack depths and widths [14,15].

Inputs into the knowledge base are defined by calibrated defects and finite element models. We anticipate that finite element modeling, once complete verification that the models are as good in predicting impedance changes as real defects, will be a much more economical and faster methodology for construction of the knowledge base.

The MacIvory® computer was chosen for the expert system platform, providing both user-friendly interfaces and symbolic processing. The interface between the Symbolics engine and the MacIntosh® is the Remote Procedural Call (RPC) interface. The two processors operate independently of each other and provide remote calls to the other processor to handle a particular function which can only be handled by the processor. Lisp is the language used by the MacIvory, and Genera is the operating environment. Of course, the MacIntosh environment is standard to that computer system. Evaluation of this platform will obviously be more strenuous in the second phase.

In the first few months of working on the project, several programs and interfaces used to call up eddy current data; i.e., to display several types of plots (i.e., emulate the SmartEddy to a degree) and to perform some smoothing routines have been developed. Examples of these displays are given in figures 19 through 21. Due to the nature of the manually scanned data obtained from early measurements of the test specimens, we are not able yet to uniquely characterize the signals from the defects. Once the robot scanning system is in place, that task will become more easily accomplished. A major effort will be needed to overcome the variable resistivity problem; however, we currently feel that a neutral network on the front end of the data acquisition will be very beneficial. That is also a task to be developed soon.

IV. CONCLUSIONS

This work has resulted in the development of eddy current transducers for the inspection of carbon filament materials with improved sensitivity, and the development of preliminary software modules for the interpretation of eddy current signals. Improved coupling efficiencies achieved with the E-probes and horseshoe probes are exceptional for graphite fibers [11,16]. The eddy current supervisory system and expert system are being developed on a MacIvory® system. Since the robot scanning system was not available during this phase of the research, only the basic foundations for the artificial intelligence were laid. Proof of the concept will be performed in the next phase. Utilization of finite element models for predetermining eddy current signals has been shown to be useful in this work, both for understanding how electromagnetic fields interact with graphite fibers, but also for use in determining how to develop the knowledge base.

Sufficient data have been taken to indicate that the E-probe and the horseshoe probe can be useful eddy current transducers for inspecting graphite fiber components. The lacking component at this time is a spinning mechanism for achieving a time-averaged impedance value for a multi-directional orientation of the fibers. Other applications include a sensor for determining proper lay-up of a graphite fiber during manufacturing.

Appendices A and B contain the software currently developed to display the eddy current signals on both the IBM PC (SmartEddy type instrument) and the MacIvory workstation. This software also shows the intent toward user-friendly displays for ease-of-use. This pattern will continue.

REFERENCES

1. Doctor, P.G., et al.: "Pattern Recognition in Methods for Characterizing and Sizing Flaws Using Eddy Current Data." Eddy Current Characterization of Materials and Structures, ASTM STP 722, 1981, pp. 461–483.

2. Vernon, S.N.: "The Universal Impedance Diagram of the Ferrite Pot Core Eddy Current Transducers." IEEE Transactions on Magnetics, vol. 25, 1989, pp. 2639–2645.

3. Sabbagh, H.A., and Vernon, S.N.: "Description and Verification of a Model of Eddy Current Probes With Ferrite Cores." Review of Progress in Quantitative NDE, vol. 3A, 1984, Plenum Press, New York, pp. 653–662.

4. Vernon, S.N.: "Parametric Eddy Current Defect Depth Model and Its Application to Graphite Epoxy." NDT International, vol. 22, 1989, pp. 139–148.

5. Gammell, P.M., and Vernon, S.N.: "Eddy Current Inspection of Broken Fiber Flaws in Non-Metallic Fiber Composites." Review of Progress in Quantitative NDE, vol. 4, 1985, pp. 1229–1237.

6. Vernon, S.N.: "Probe Properties Affecting the Eddy Current NDI of Graphite Epoxy." Review of Progress in Quantitative NDE, vol. 7, 1987, pp. 1113–1123.

7. Vernon, S.N.: "Eddy Current Inspection of Thick Carbon Fiber Reinforced Composites." Review of Progress in Quantitative NDE, vol. 8, 1988, pp. 1543–1550.

8. Bowler, J.R.: "Electromagnetic Fields in Advanced Composites," pp. 736–741.

9. Valleau, A.R.: "Eddy Current Nondestructive Testing of Graphite Composite Materials." Materials Evaluation, vol. 48, 1990, pp. 230–239.

10. Vernon, S.: "A Single-Sided Eddy-Current Method to Measure Electrical Resistivity." Materials Evaluation, vol. 46, 1988, pp. 1581–1587.

11. Dodd, C.V., and Deeds, W.E.: "Absolute Eddy Current Measurements of Electrical Conductivity." Review of Progress in Quantitative NDE, vol. 2, Plenum Press, New York, 1982, pp. 387–393.

12. Cecco, V.S., Van Drunen, G., and Sharp, F.L.: "Eddy Current Manual, Volume 1." Chalk River Laboratories, Chalk River, Ontario.

13. Upda, L., and Upda, S.S.: "Eddy Current Defect Characterization Using Neural Networks." Materials Evaluation, vol. 48, 1990, pp. 342–347.

14. Upda, L., and Lord, W.: "A Search-Based Imaging System for Electromagnetic Nondestructive Testing." IEEE Expert, vol. 4, 1989, pp. 18–26.

15. Upda, L., and Lord, W.: "An AI Approach to the Eddy Current Defect Characterization Problem." Review of Progress in Quantitative NDE, vol. 8, 1988, pp. 899–906.

16. Shull, P.J., Capobianca, T.E., and Moulder, J.C.: "Design and Characterization of Uniform Field Eddy Current Probes." Review of Progress in Quantitative NDE, vol. 6A, 1986, pp. 695–703.

Table 1. Normalized eddy current signals for graphite fiber using Ansoft Maxwell® finite element model.

FREQUENCY (Hz)	RESISTANCE	REACTANCE
50 k	0.099	0.996
100 k	0.152	0.942
200 k	0.18	0.882
500 k	0.21	0.790
800 k	0.225	0.730
1 M	0.23	0.670
1.5 M	0.214	0.570
2 M	0.214	0.570
2.5 M	0.197	0.530
3 M	0.179	0.498
3.5 M	0.158	0.469
4 M	0.135	0.445
5 M	0.026	0.342

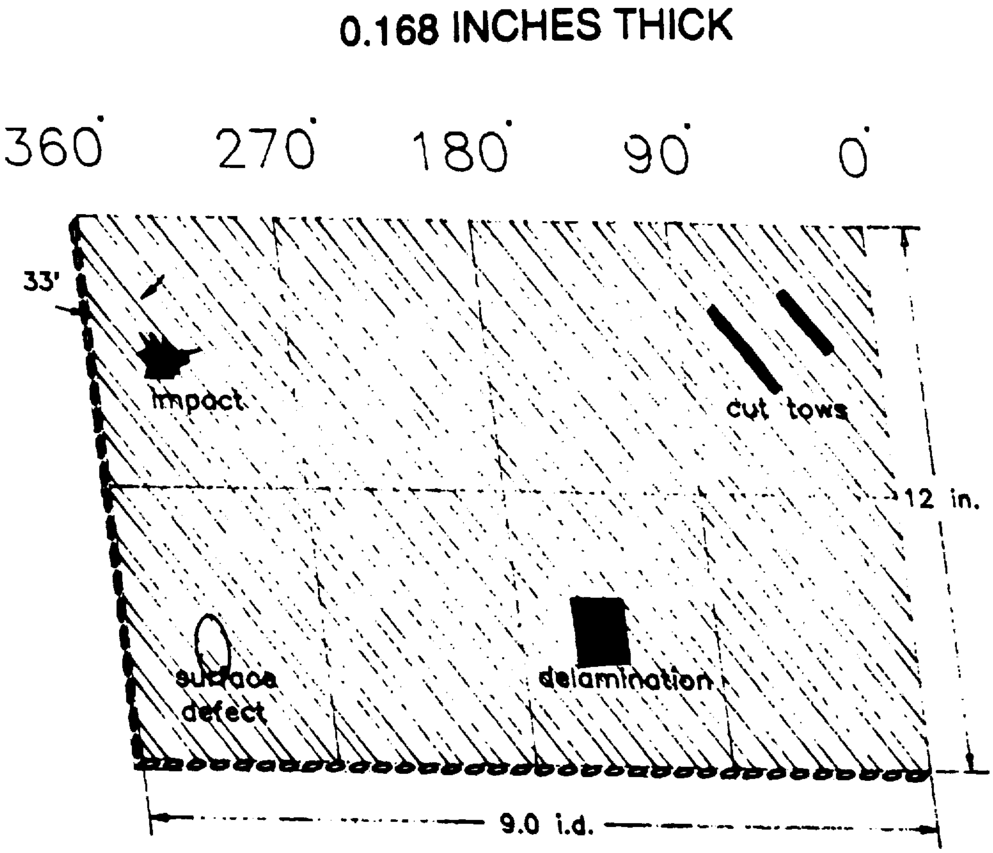


Figure 1. Graphite fiber test cylinder layout.

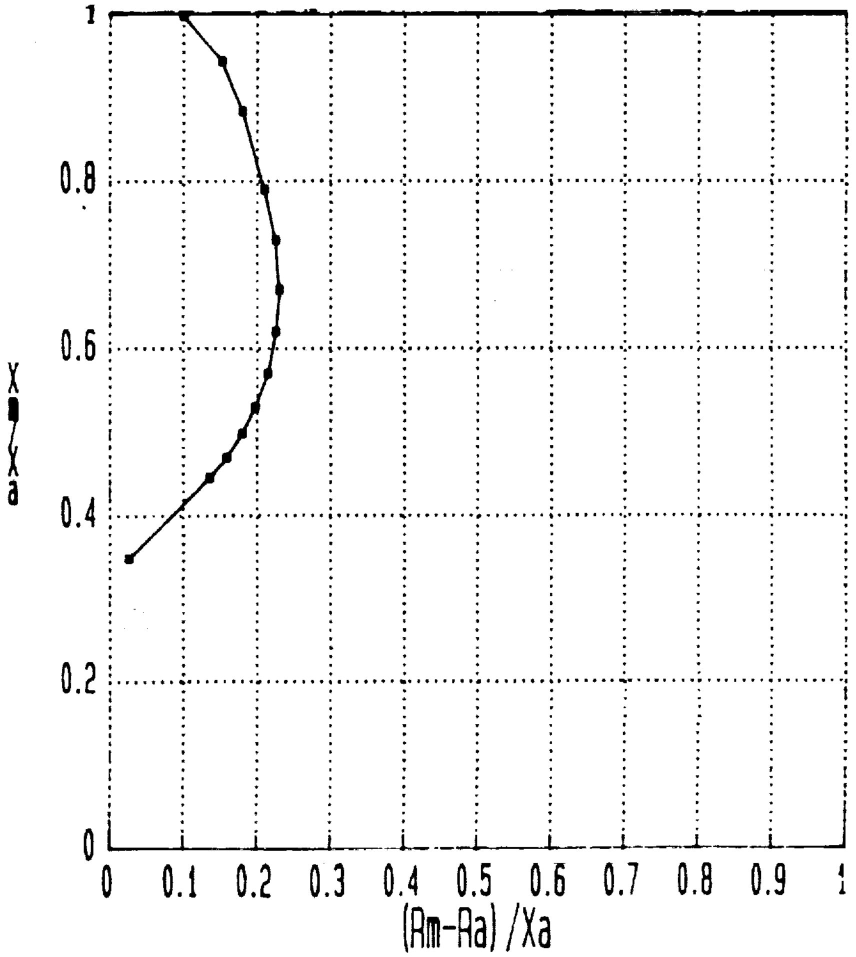


Figure 2. Normalized impedance plane diagram for graphite fibers.

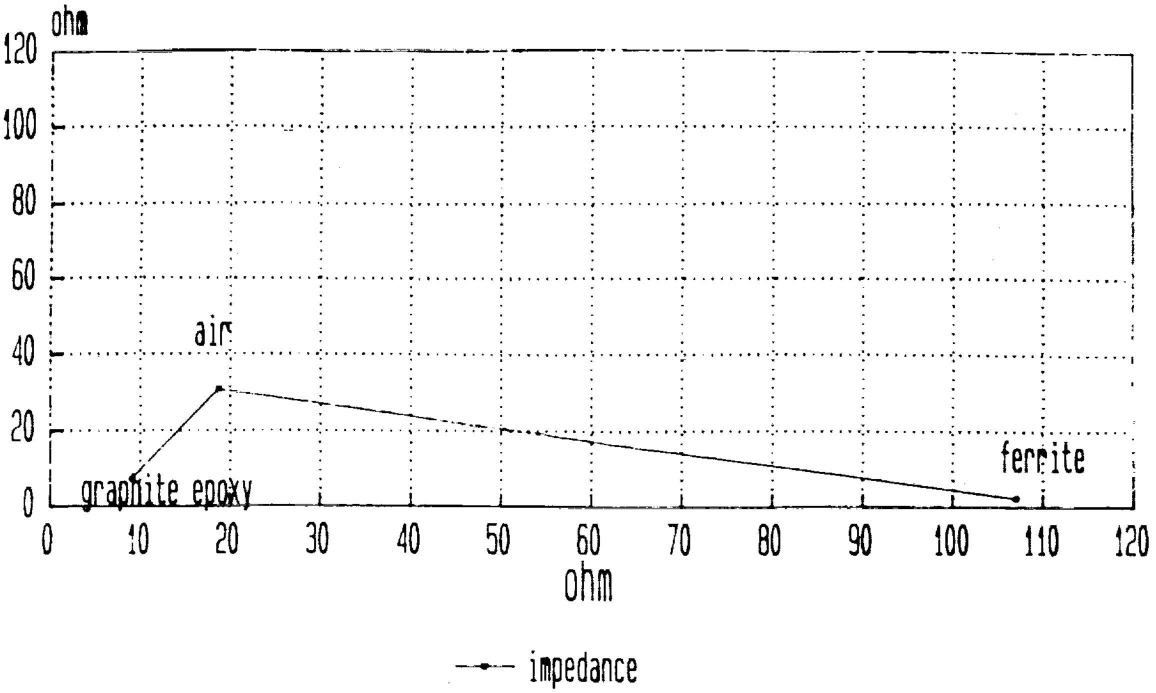


Figure 3. Calculated materials points for graphite epoxy and ferrite.

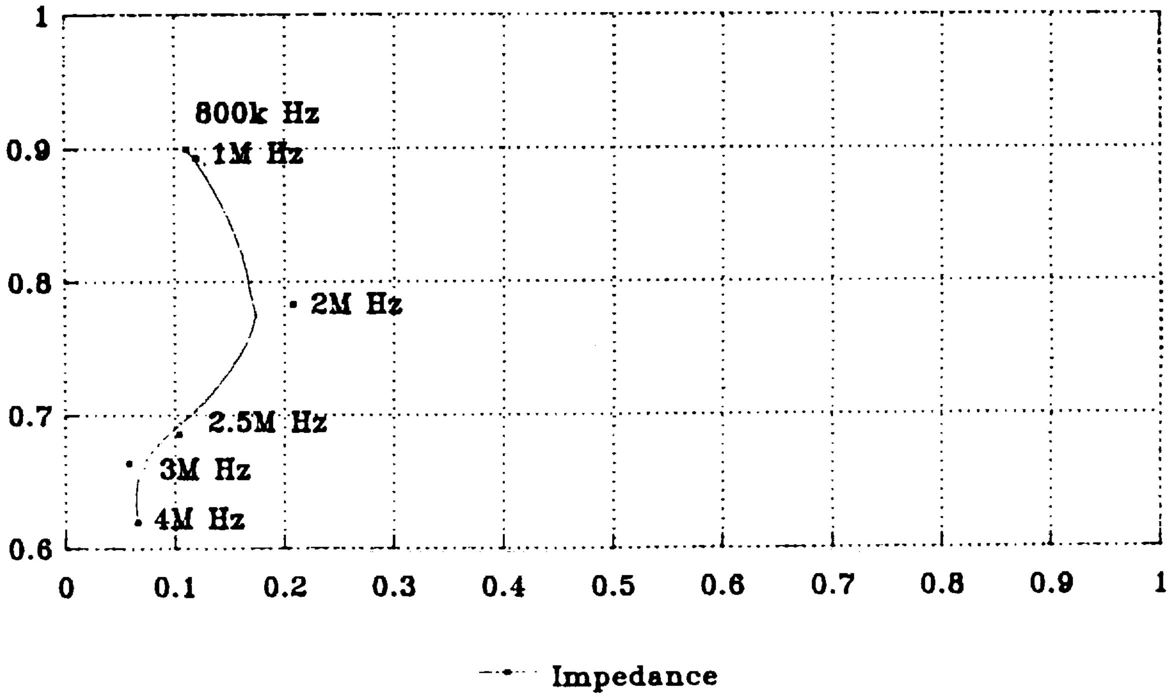


Figure 4. Impedance plane diagram of eddy current measurements on graphite fiber sample.

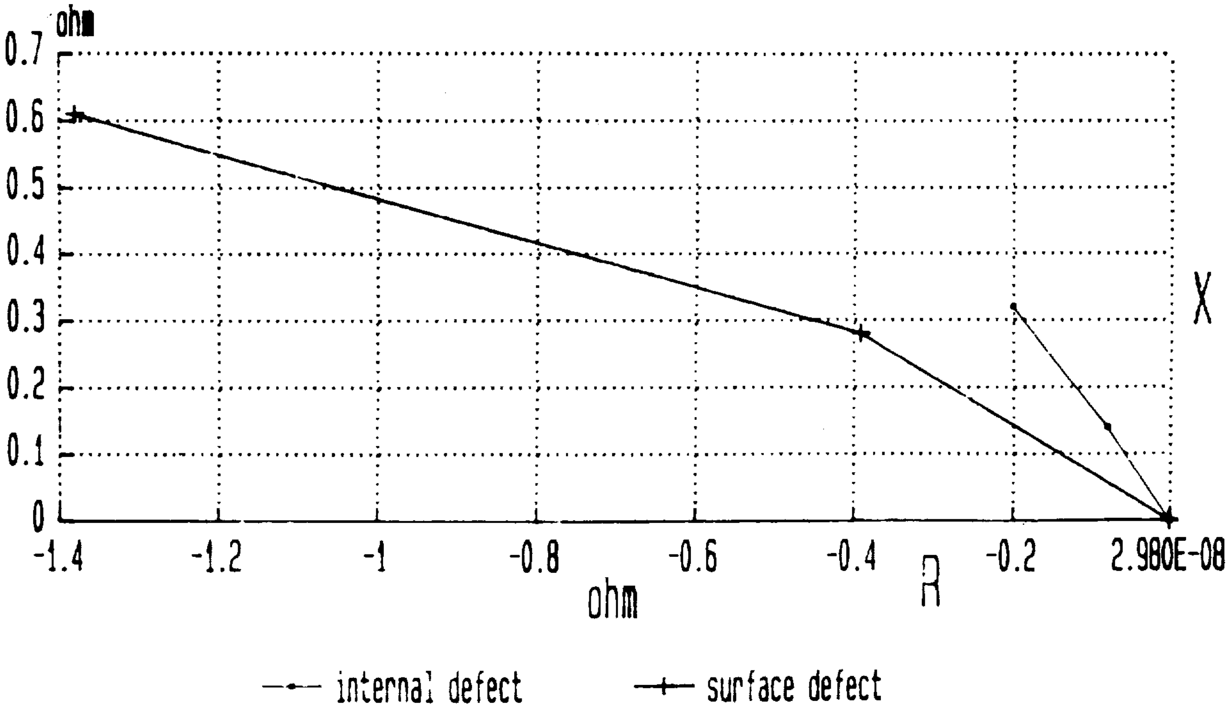
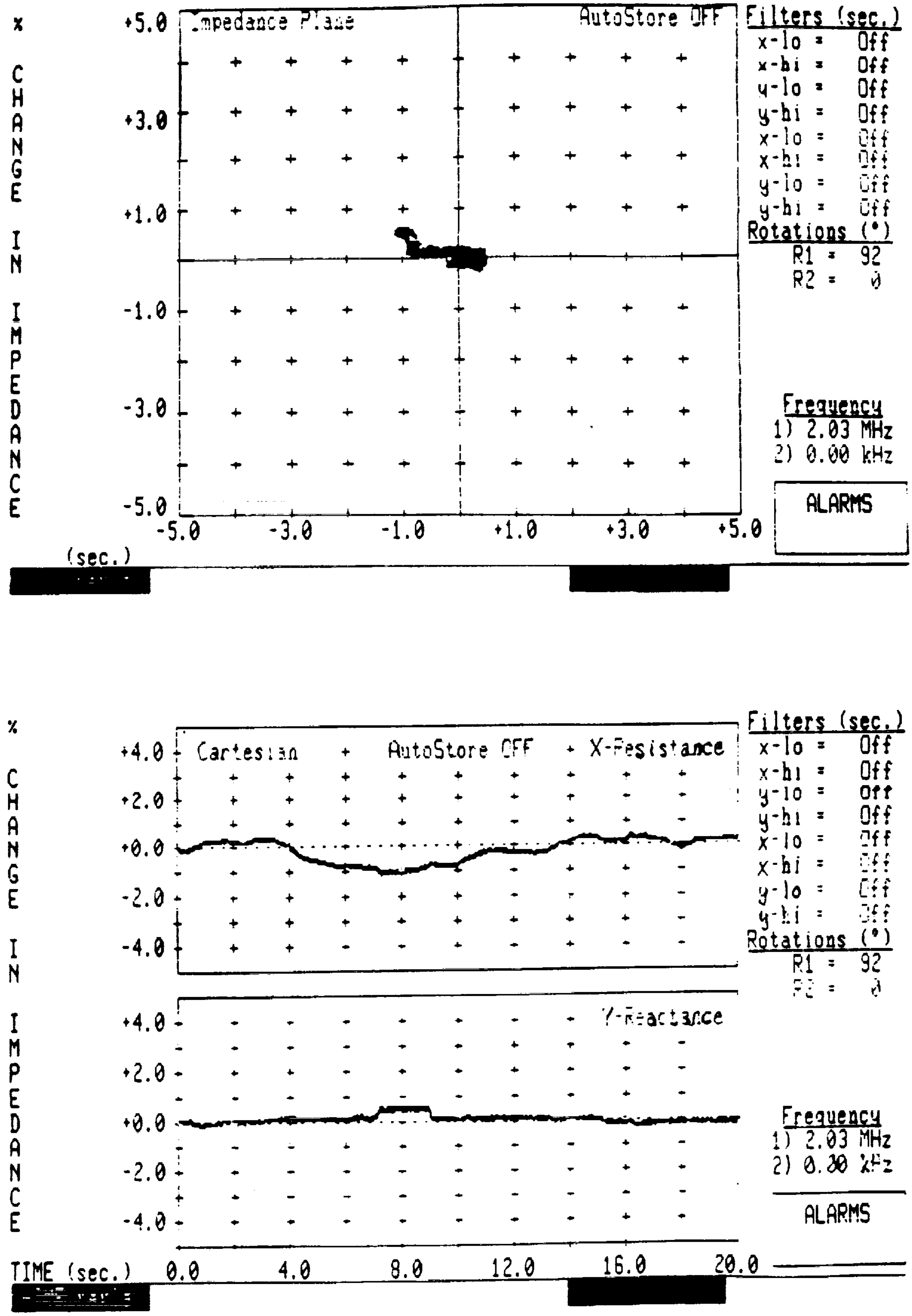


Figure 5. Phase angle relationships for defects in graphite fiber material.



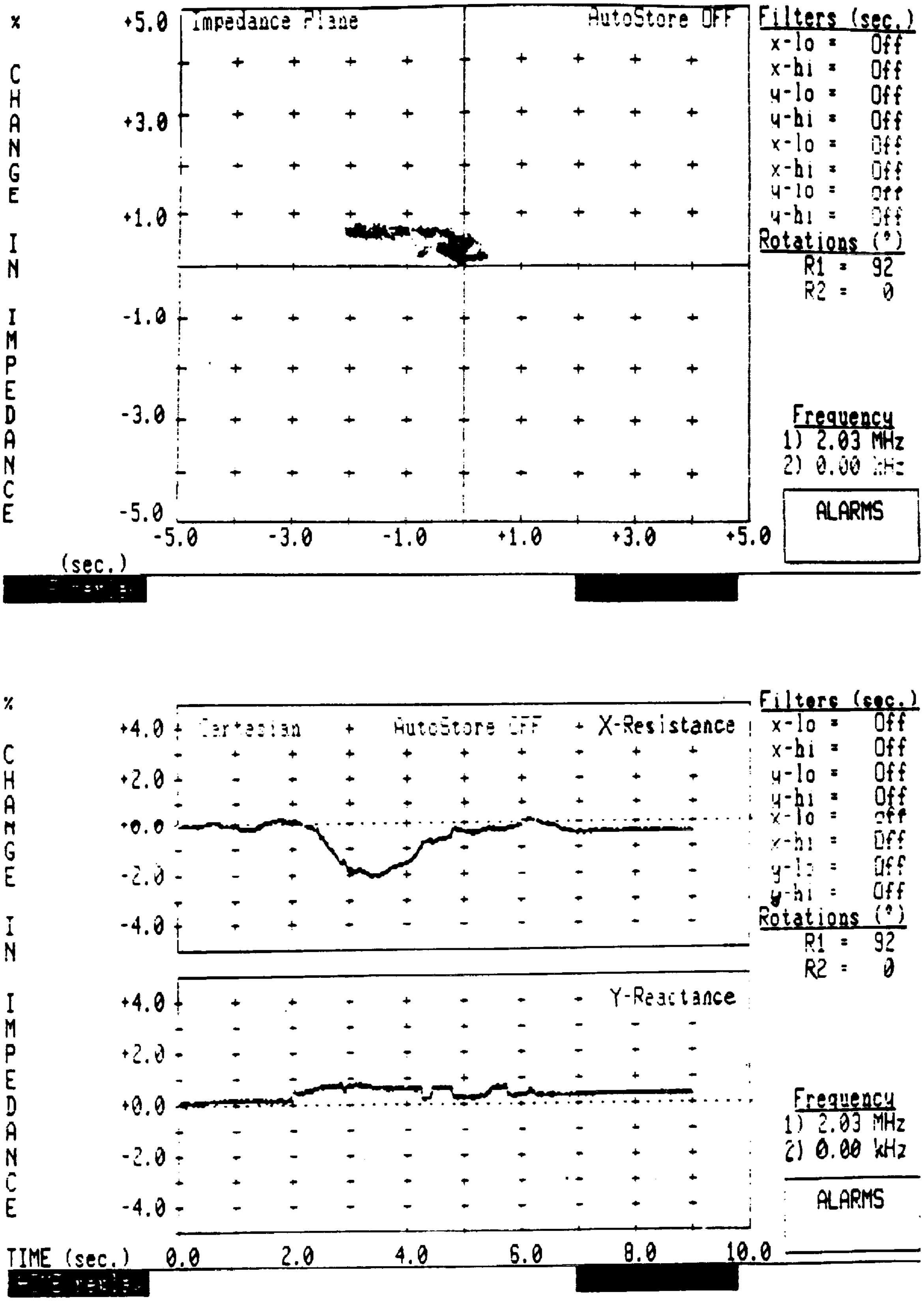
Filters (sec.)
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
Rotations (°)
R1 = 92
R2 = 0

Frequency
1) 2.03 MHz
2) 0.00 kHz

Filters (sec.)
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
Rotations (°)
R1 = 92
R2 = 0

Frequency
1) 2.03 MHz
2) 0.00 kHz

Figure 6. SmartEddy displays of delamination in test sample.



Filters (sec.)
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off

Rotations (°)
R1 = 92
R2 = 0

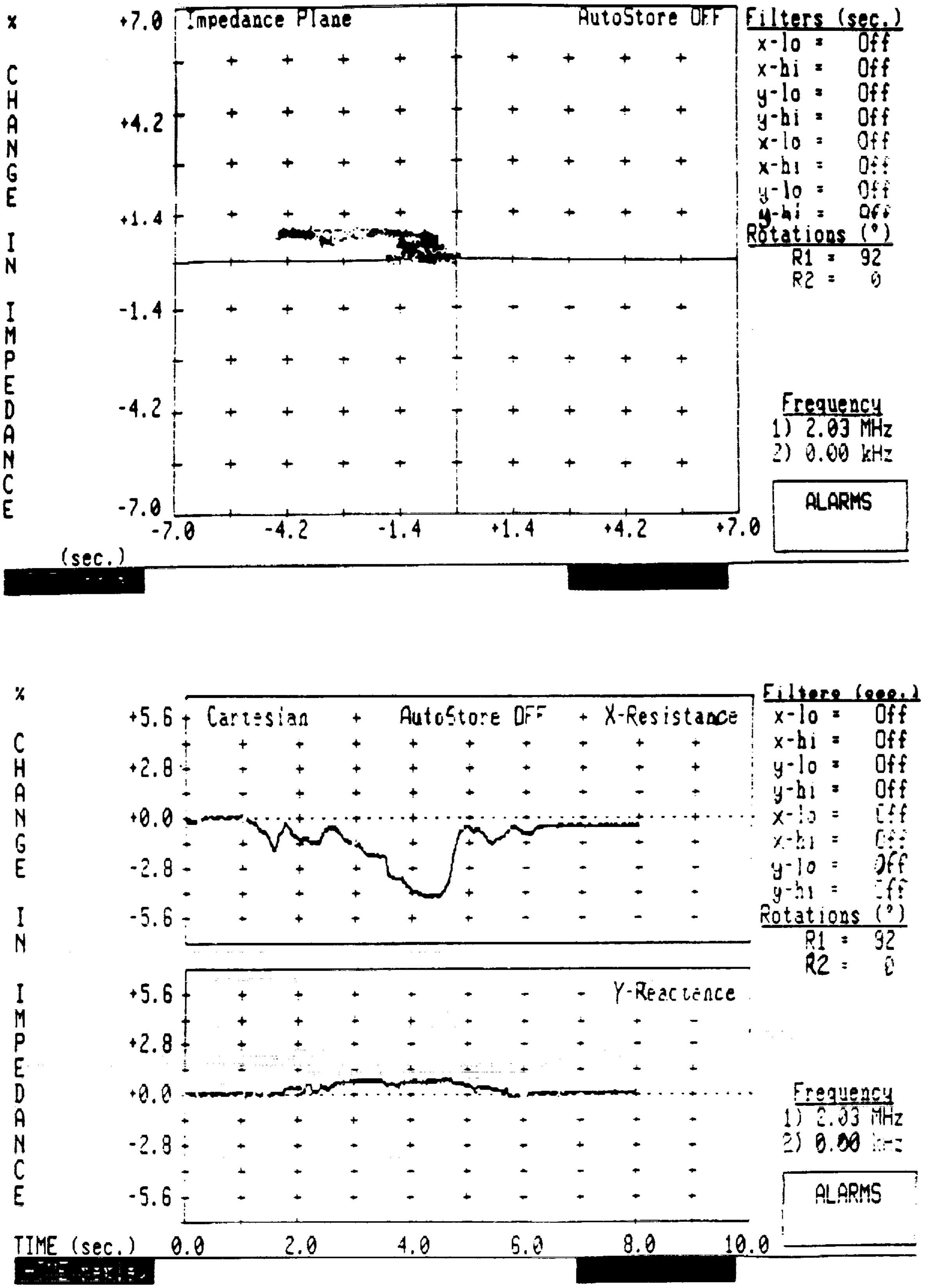
Frequency
1) 2.03 MHz
2) 0.00 kHz

Filters (sec.)
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off

Rotations (°)
R1 = 92
R2 = 0

Frequency
1) 2.03 MHz
2) 0.00 kHz

Figure 7. SmartEddy displays of near surface internal defect in test sample using graphite knot.



Filters (sec.)
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
Rotations (°)
R1 = 92
R2 = 0

Frequency
1) 2.03 MHz
2) 0.00 kHz

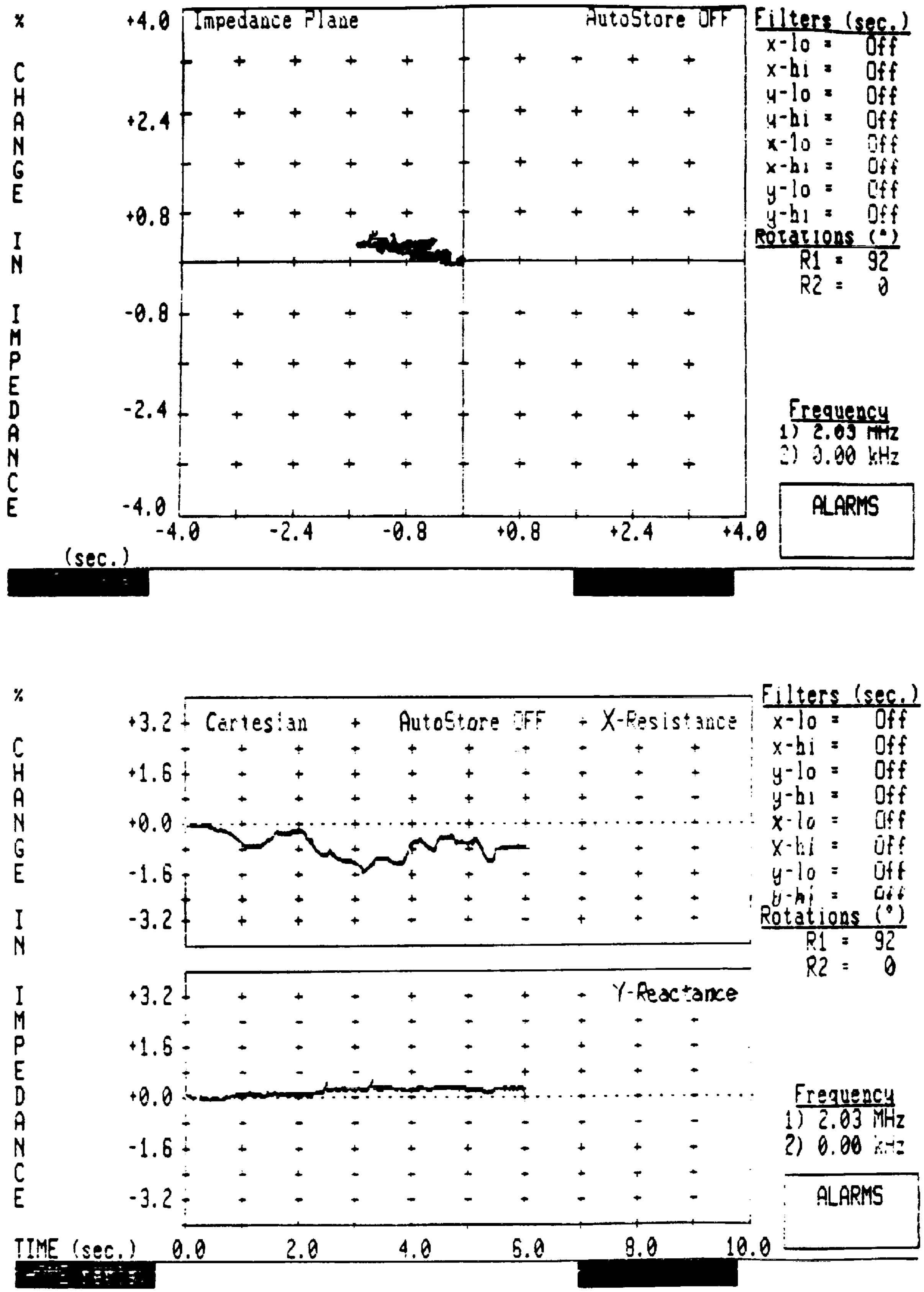
ALARMS

Filters (sec.)
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
Rotations (°)
R1 = 92
R2 = 0

Frequency
1) 2.03 MHz
2) 0.00 kHz

ALARMS

Figure 8. SmartEddy displays of surface defect in test sample using graphite knot.



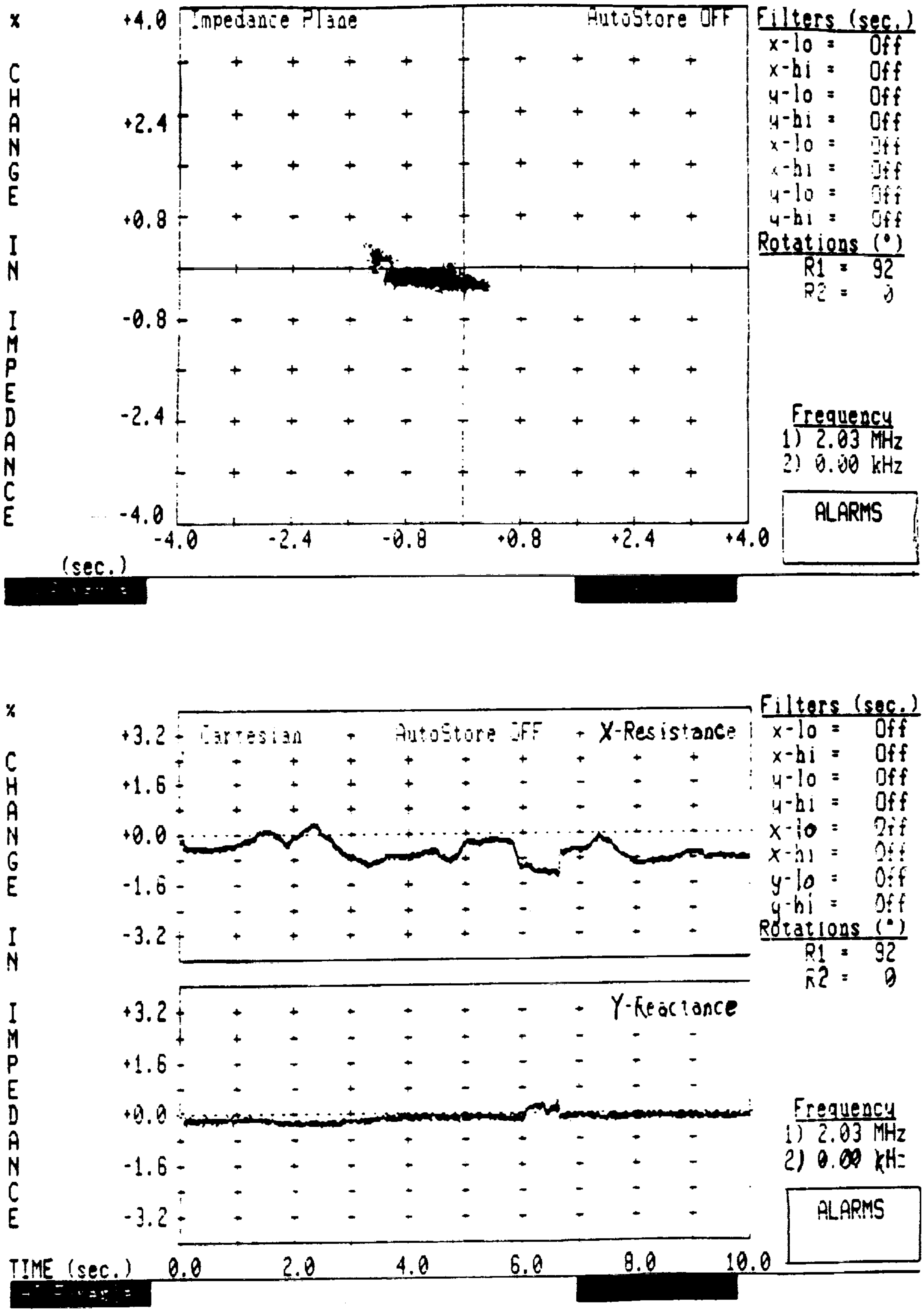
Filters (sec.)
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
Rotations (°)
R1 = 92
R2 = 0

Frequency
1) 2.03 MHz
2) 0.00 kHz

Filters (sec.)
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
Rotations (°)
R1 = 92
R2 = 0

Frequency
1) 2.03 MHz
2) 0.00 kHz

Figure 9. SmartEddy displays of impact damage in test sample.



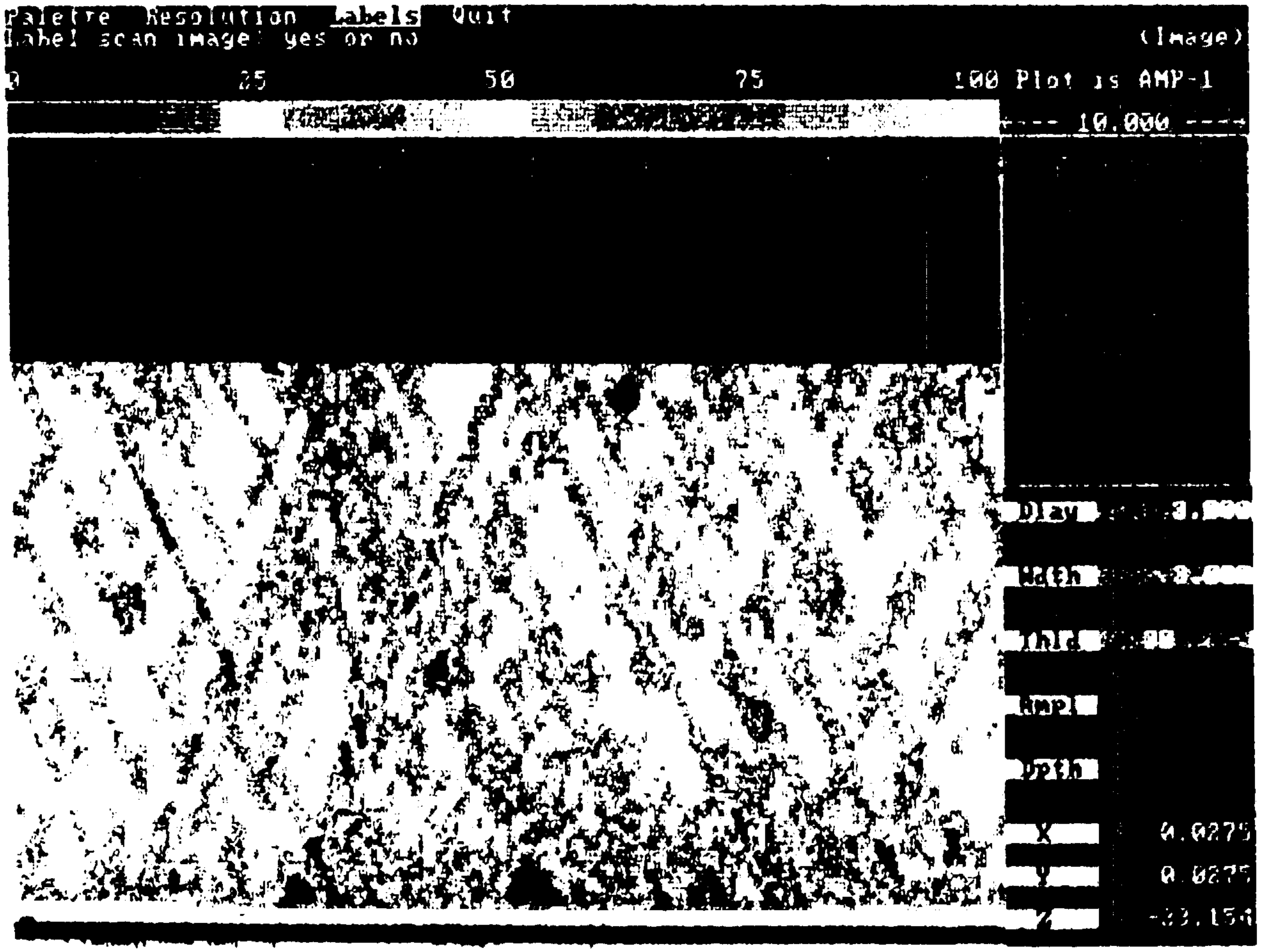
Filters (sec.)
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
Rotations (°)
R1 = 92
R2 = 0

Frequency
1) 2.03 MHz
2) 0.00 kHz

Filters (sec.)
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
Rotations (°)
R1 = 92
R2 = 0

Frequency
1) 2.03 MHz
2) 0.00 kHz

Figure 10. SmartEddy displays of internal cut tows in test sample.



NOTE THAT IN COMPARISON WITH FIGURE 4, ONLY THE DELAMINATION DEFECT IS EASILY OBSERVED.

Figure 11. Ultrasonic C-scan display of graphite filament-wound cylinder.

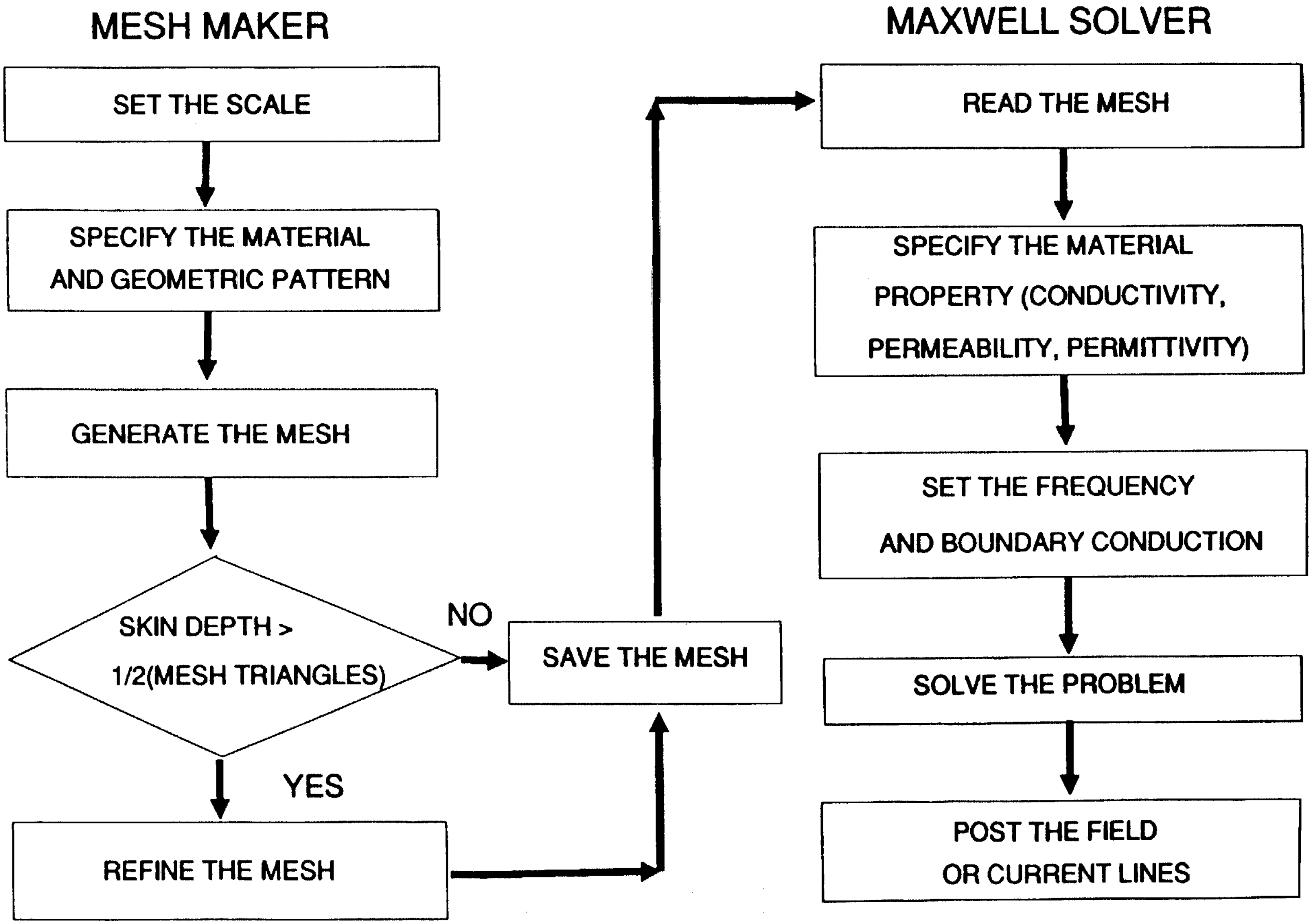

MESH MAKER
SET THE SCALE
SPECIFY THE MATERIAL AND GEOMETRIC PATTERN
GENERATE THE MESH
SKIN DEPTH > 1/2(MESH TRIANGLES)
NO
YES
REFINE THE MESH
SAVE THE MESH
MAXWELL SOLVER
READ THE MESH
SPECIFY THE MATERIAL PROPERTY (CONDUCTIVITY, PERMEABILITY, PERMITTIVITY)
SET THE FREQUENCY AND BOUNDARY CONDUCTION
SOLVE THE PROBLEM
POST THE FIELD OR CURRENT LINES

Figure 12. Electromagnetic field solver.

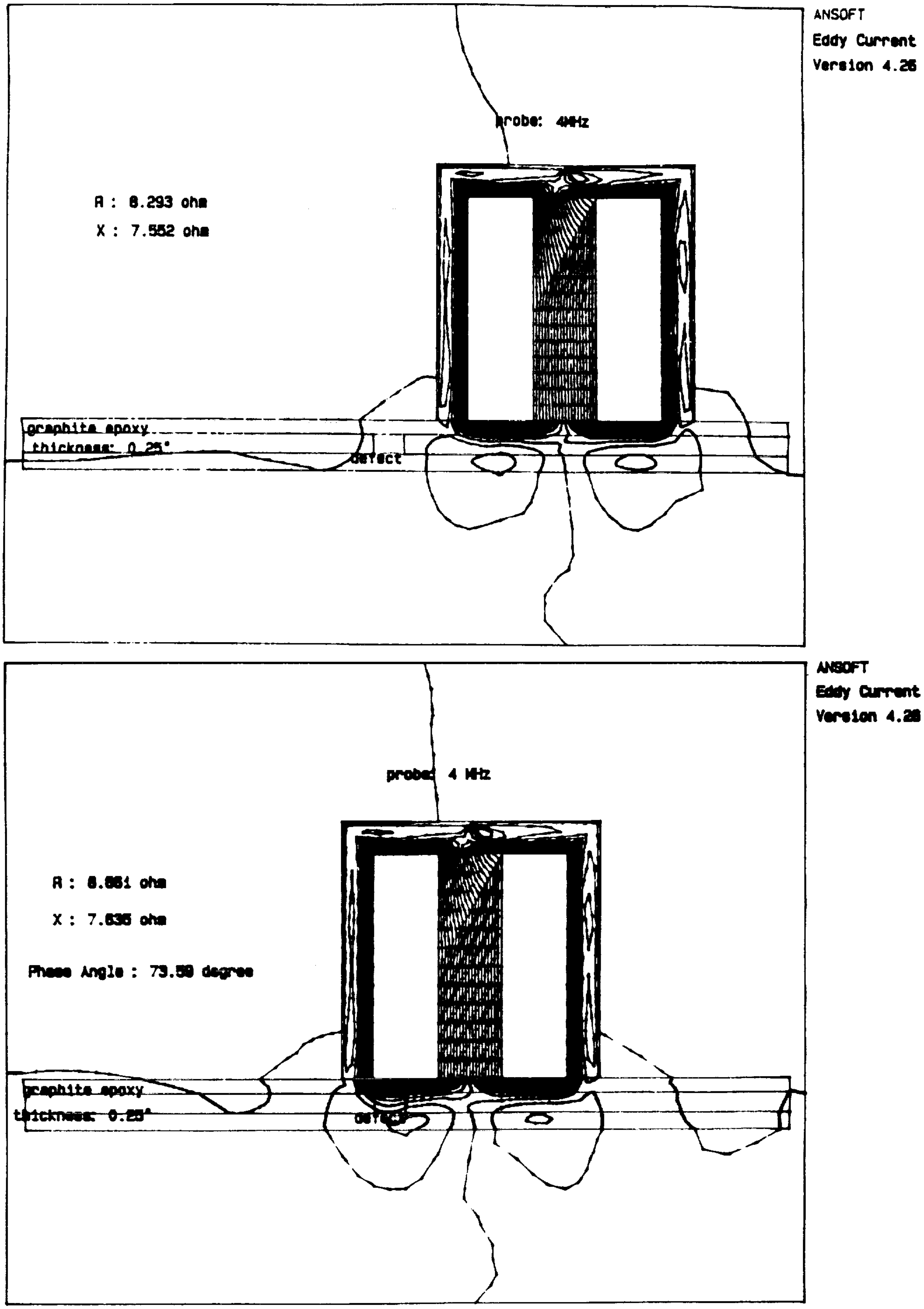


Figure 13. Finite element model of graphite fiber scan over internal defect.

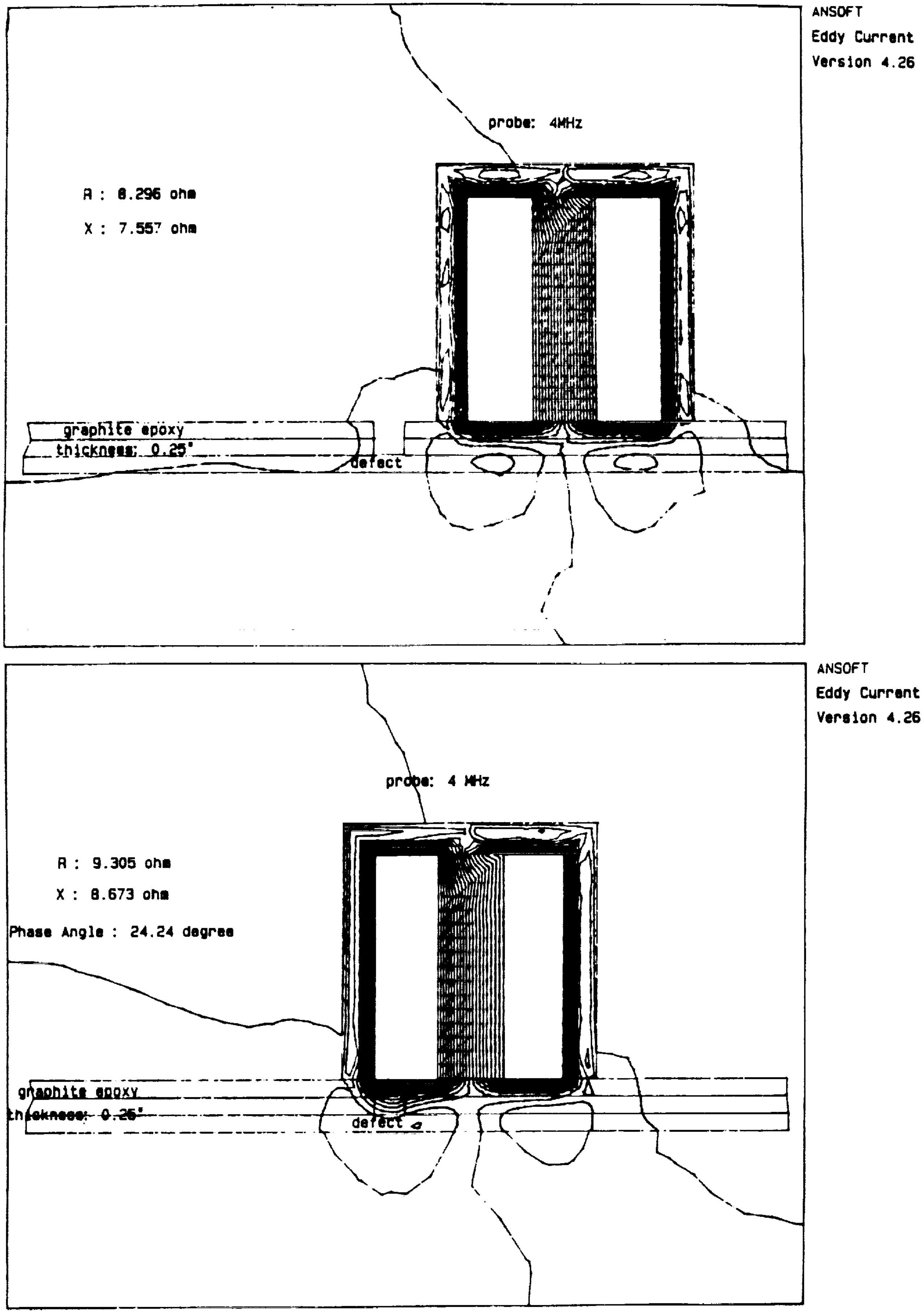
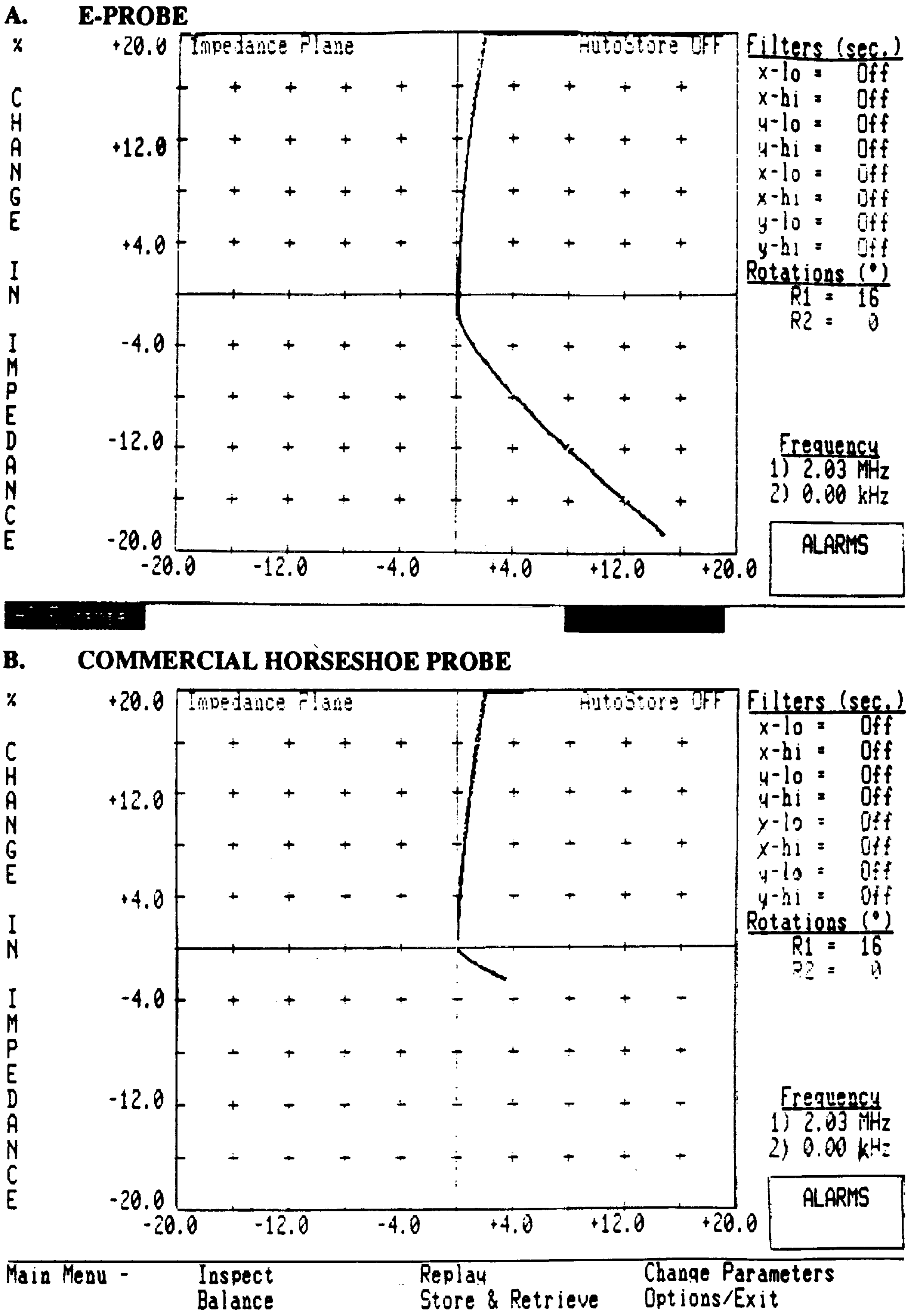


Figure 14. Finite element model of graphite fiber scan over surface defect.



A. E-PROBE

Filters (sec.)
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off

Rotations (°)
R1 = 16
R2 = 0

Frequency
1) 2.03 MHz
2) 0.00 kHz

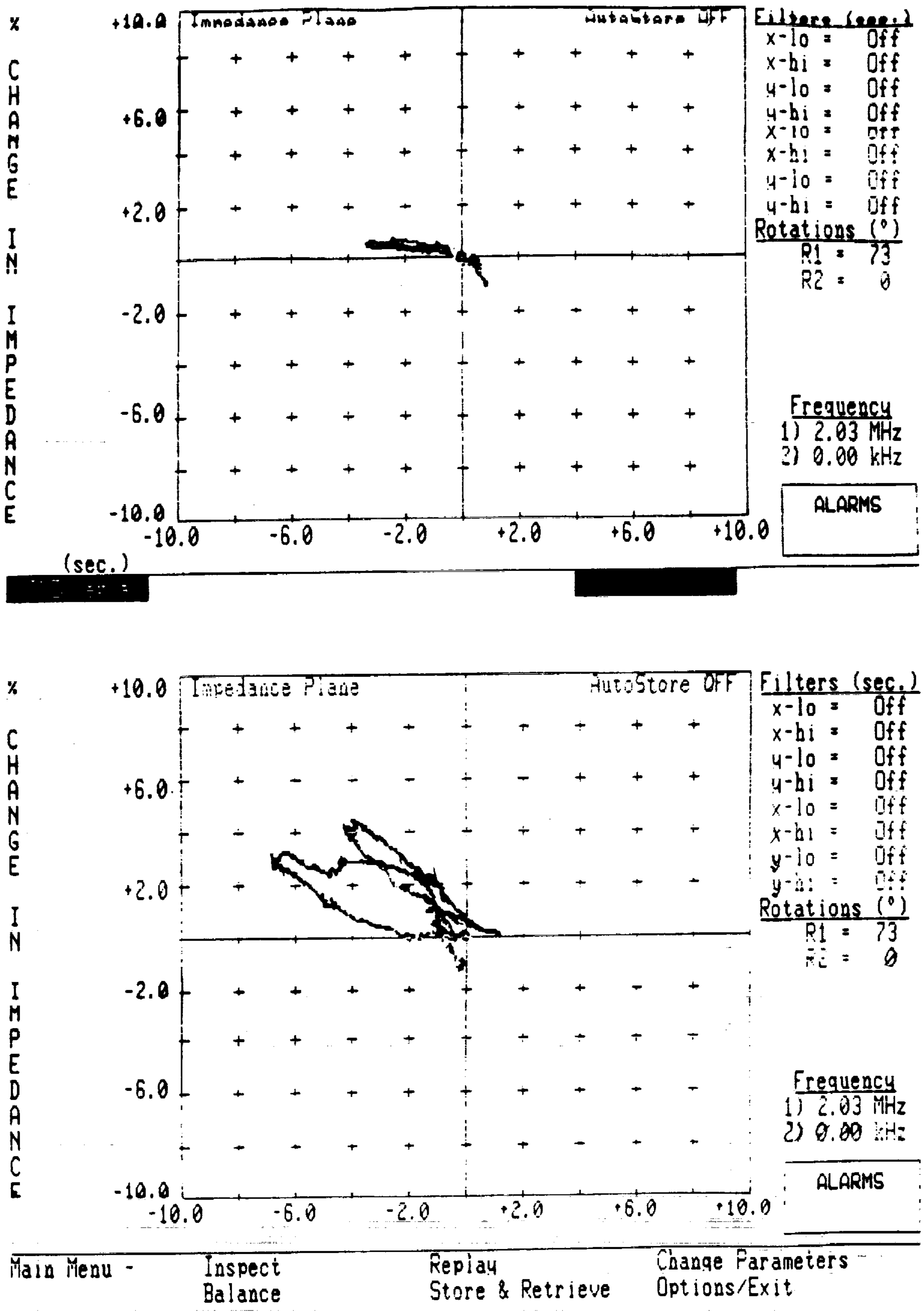
B. COMMERCIAL HORSESHOE PROBE

Filters (sec.)
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off

Rotations (°)
R1 = 16
R2 = 0

Frequency
1) 2.03 MHz
2) 0.00 kHz

Figure 15. Comparison of measured lift-off lines for E-probe and commercially available horseshoe probe.



Filters (sec.)
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
Rotations (°)
R1 = 73
R2 = 0

Frequency
1) 2.03 MHz
2) 0.00 kHz

ALARMS

(sec.)

Filters (sec.)
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
x-lo = Off
x-hi = Off
y-lo = Off
y-hi = Off
Rotations (°)
R1 = 73
R2 = 0

Frequency
1) 2.03 MHz
2) 0.00 kHz

ALARMS

Main Menu - Inspect Balance Replay Store & Retrieve Change Parameters Options/Exit

Figure 16. SmartEddy displays of horseshoe probe scans over surface defect.

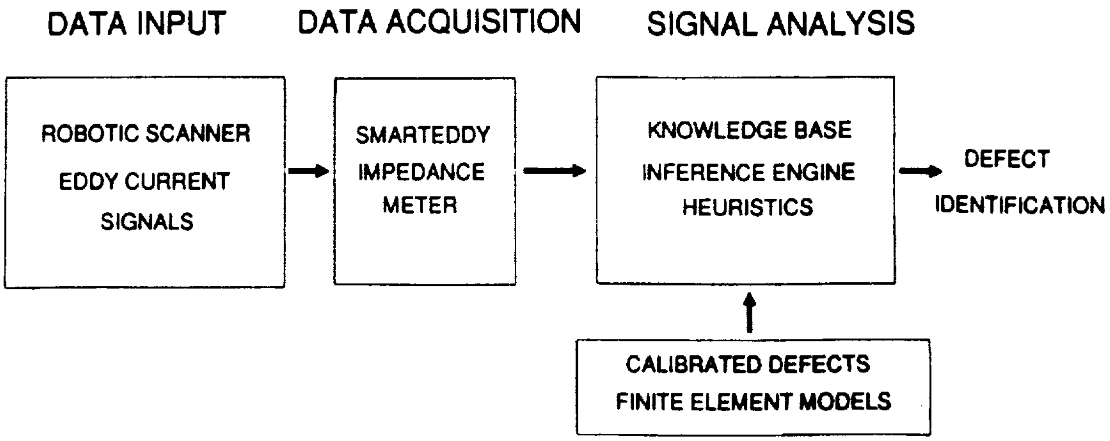


Figure 17. Model of expert system for automated eddy current analysis.

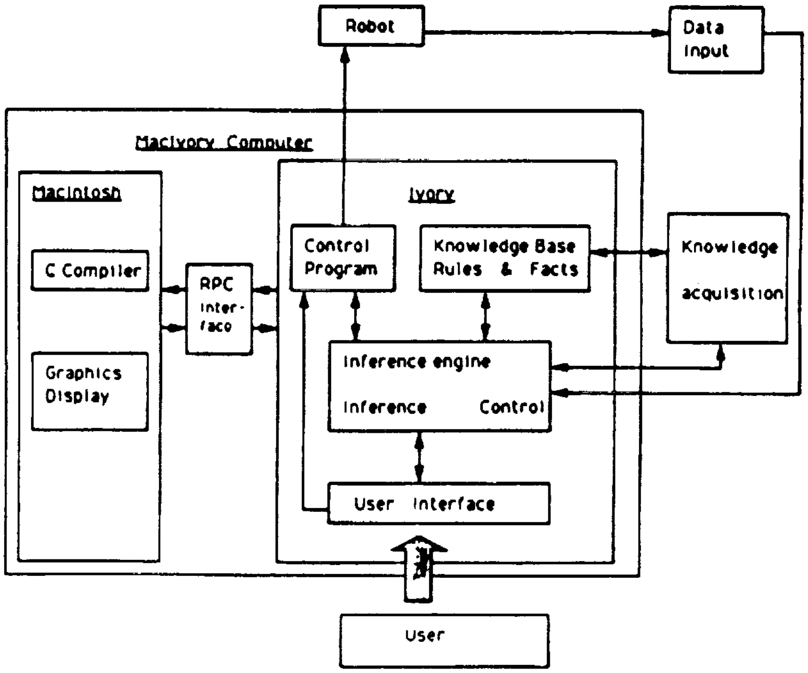
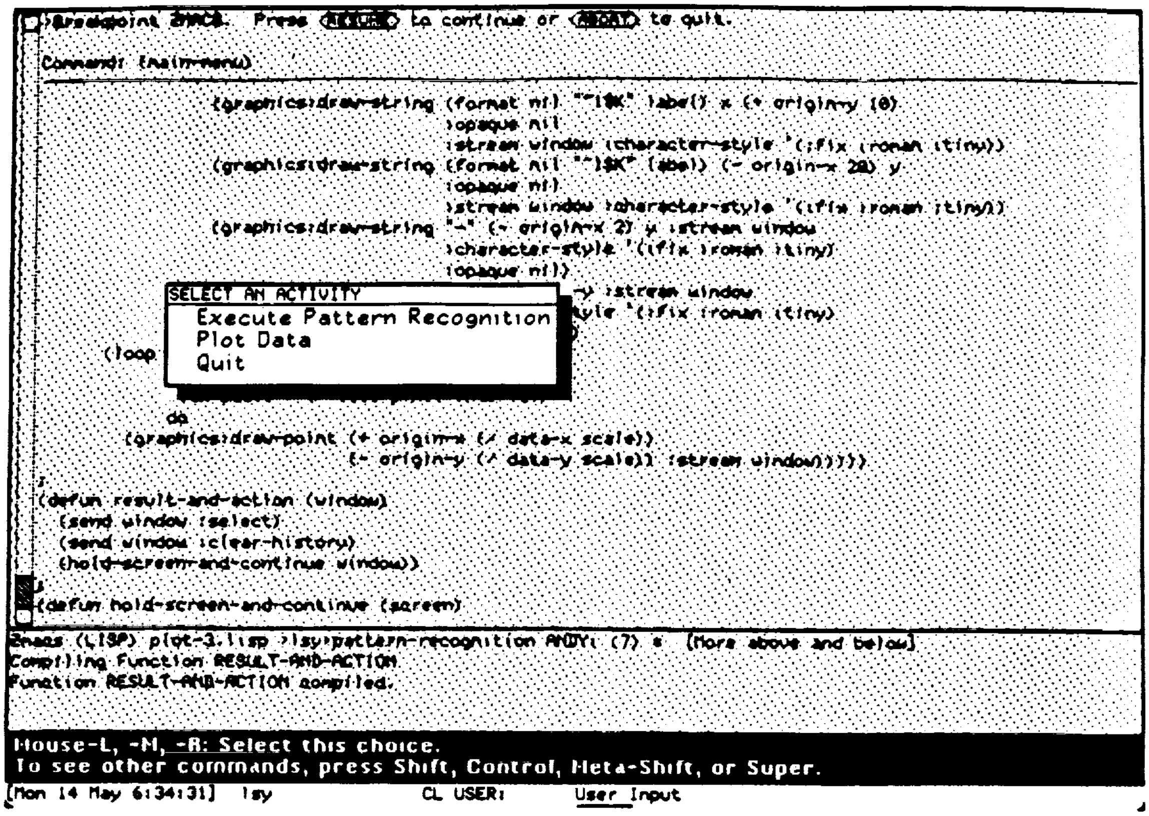
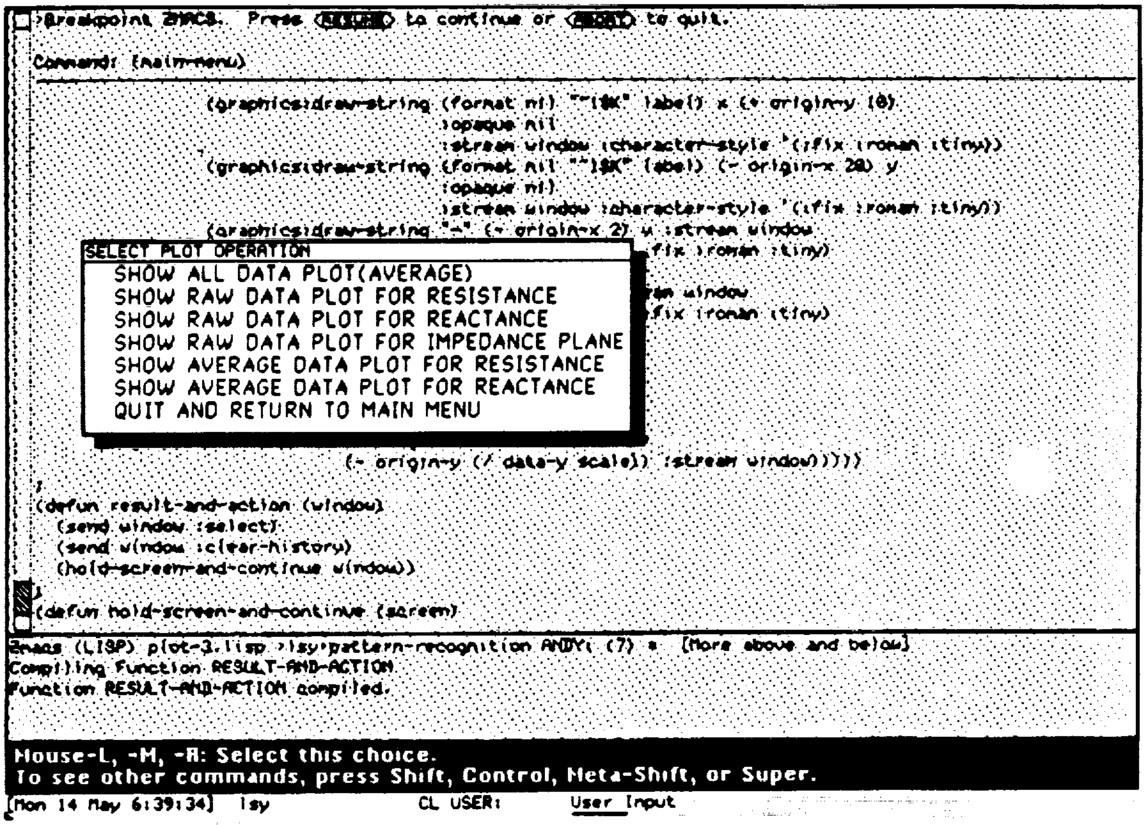


Figure 18. Expert system architecture for eddy current analysis.

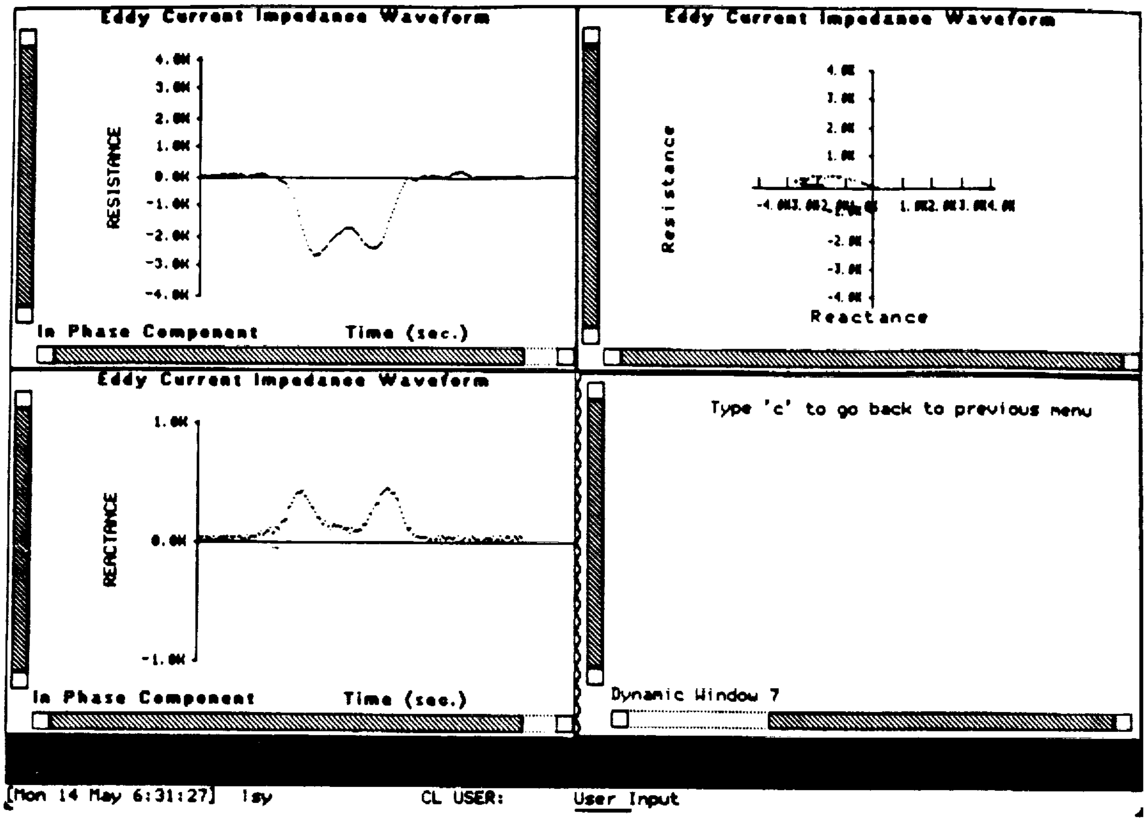


```
>Breakpoint ZMACS.  Press RESUME to continue or ABORT to quit.

Command: (main-menu)
              (graphics:draw-string (format nil "~1$K" label) x (+ origin-y 10)
                                    :opaque nil
                                    :stream window :character-style '(:fix :roman :tiny))
              (graphics:draw-string (format nil "~1$K" label) (- origin-x 20) y
                                    :opaque nil)
                                    :stream window :character-style '(:fix :roman :tiny))
              (graphics:draw-string "-" (- origin-x 2) y :stream window
                                    :character-style '(:fix :roman :tiny)
                                    :opaque nil))
                                                  -y :stream window
                                                  yle '(:fix :roman :tiny)
      (loop
            do
        (graphics:draw-point (+ origin-x (/ data-x scale))
                             (- origin-y (/ data-y scale)) :stream window))))
;
(defun result-and-action (window)
  (send window :select)
  (send window :clear-history)
  (hold-screen-and-continue window))
;
(defun hold-screen-and-continue (screen)
Zmacs (LISP) plot-3.lisp >lsy>pattern-recognition ANDY: (7) *  [More above and below]
Compiling Function RESULT-AND-ACTION
Function RESULT-AND-ACTION compiled.

Mouse-L, -M, -R: Select this choice.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Mon 14 May 6:34:31]   lsy              CL USER:       User Input
```


SELECT PLOT OPERATION
SHOW ALL DATA PLOT(AVERAGE)
SHOW RAW DATA PLOT FOR RESISTANCE
SHOW RAW DATA PLOT FOR REACTANCE
SHOW RAW DATA PLOT FOR IMPEDANCE PLANE
SHOW AVERAGE DATA PLOT FOR RESISTANCE
SHOW AVERAGE DATA PLOT FOR REACTANCE
QUIT AND RETURN TO MAIN MENU

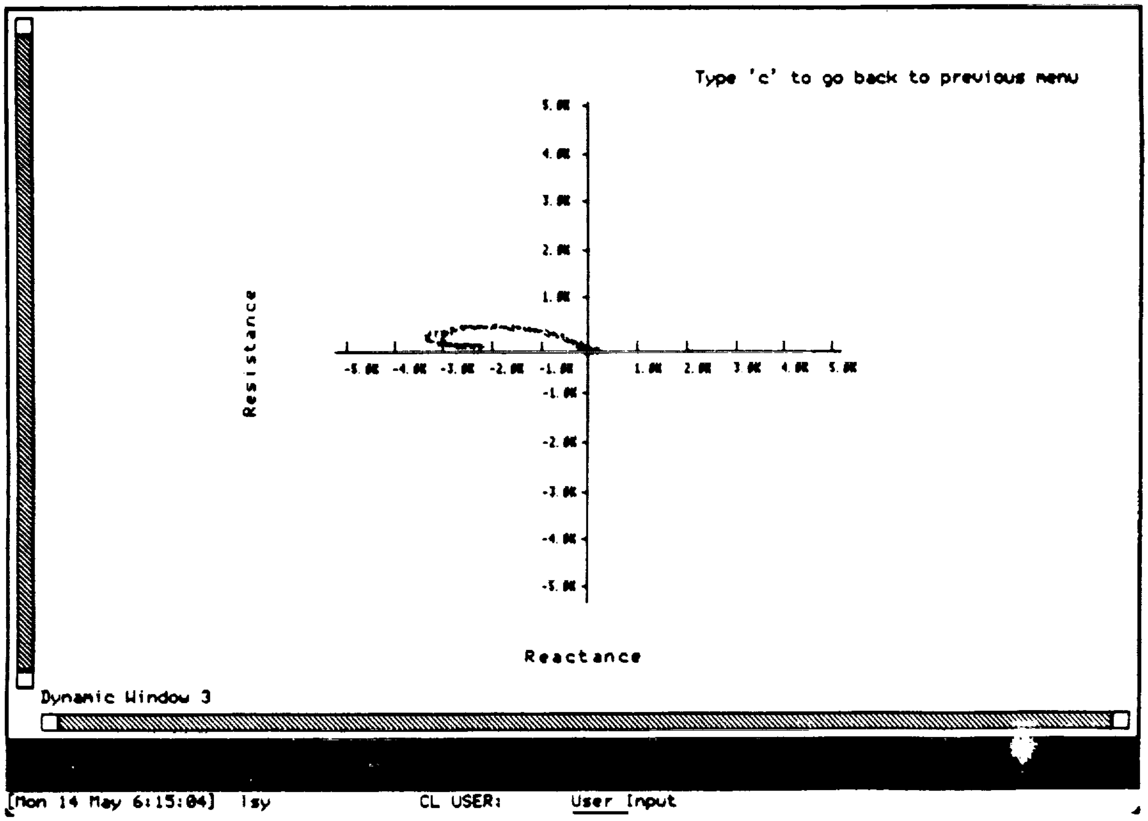
```
>Breakpoint ZMACS.  Press RESUME to continue or ABORT to quit.

Command: (main-menu)
              (graphics:draw-string (format nil "~1$K" label) x (+ origin-y 10)
                                    :opaque nil
                                    :stream window :character-style '(:fix :roman :tiny))
              (graphics:draw-string (format nil "~1$K" label) (- origin-x 20) y
                                    :opaque nil)
                                    :stream window :character-style '(:fix :roman :tiny))
              (graphics:draw-string "-" (- origin-x 2) y :stream window
                                                   fix :roman :tiny)
                                                   am window
                                                   fix :roman :tiny)
                             (- origin-y (/ data-y scale)) :stream window))))
;
(defun result-and-action (window)
  (send window :select)
  (send window :clear-history)
  (hold-screen-and-continue window))
;
(defun hold-screen-and-continue (screen)
Zmacs (LISP) plot-3.lisp >lsy>pattern-recognition ANDY: (7) *  [More above and below]
Compiling Function RESULT-AND-ACTION
Function RESULT-AND-ACTION compiled.

Mouse-L, -M, -R: Select this choice.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Mon 14 May 6:39:34]   lsy              CL USER:       User Input
```

Figure 19. Screens for user selection of options on MacIvory system.



Type 'c' to go back to previous menu

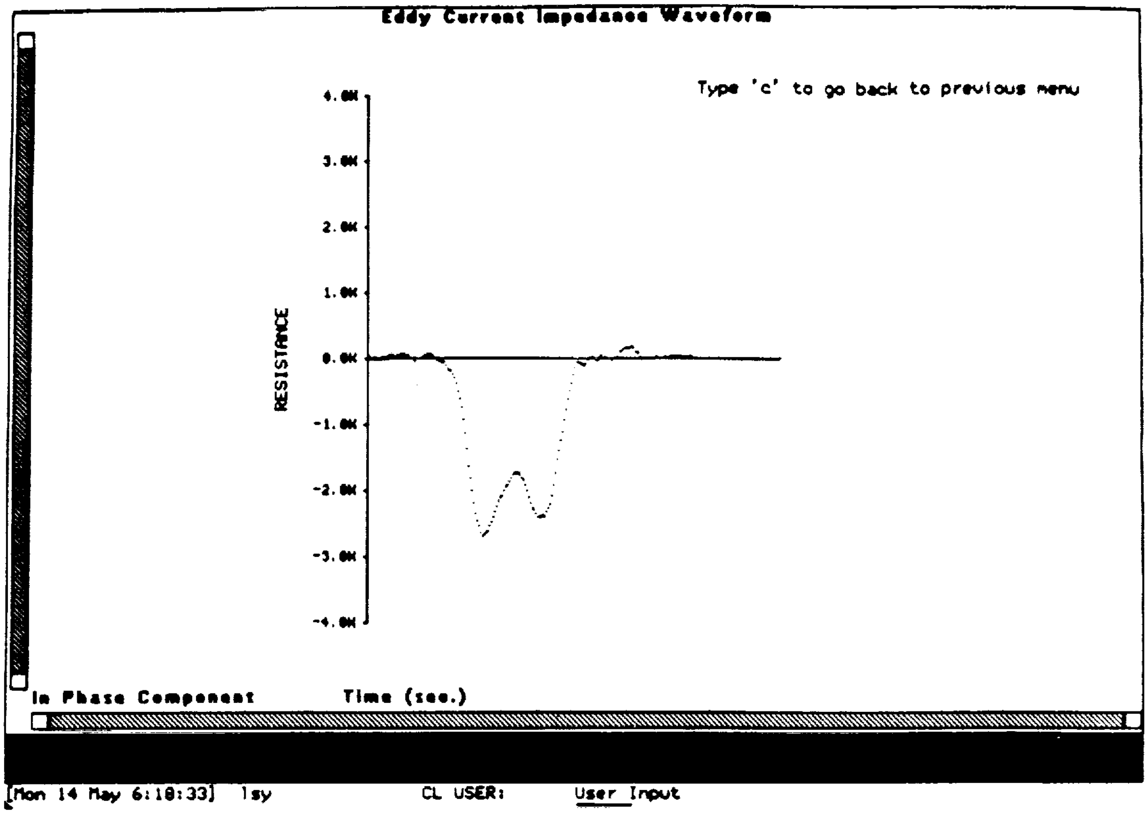
[Mon 14 May 6:31:27] lsy CL USER: User Input



Type 'c' to go back to previous menu

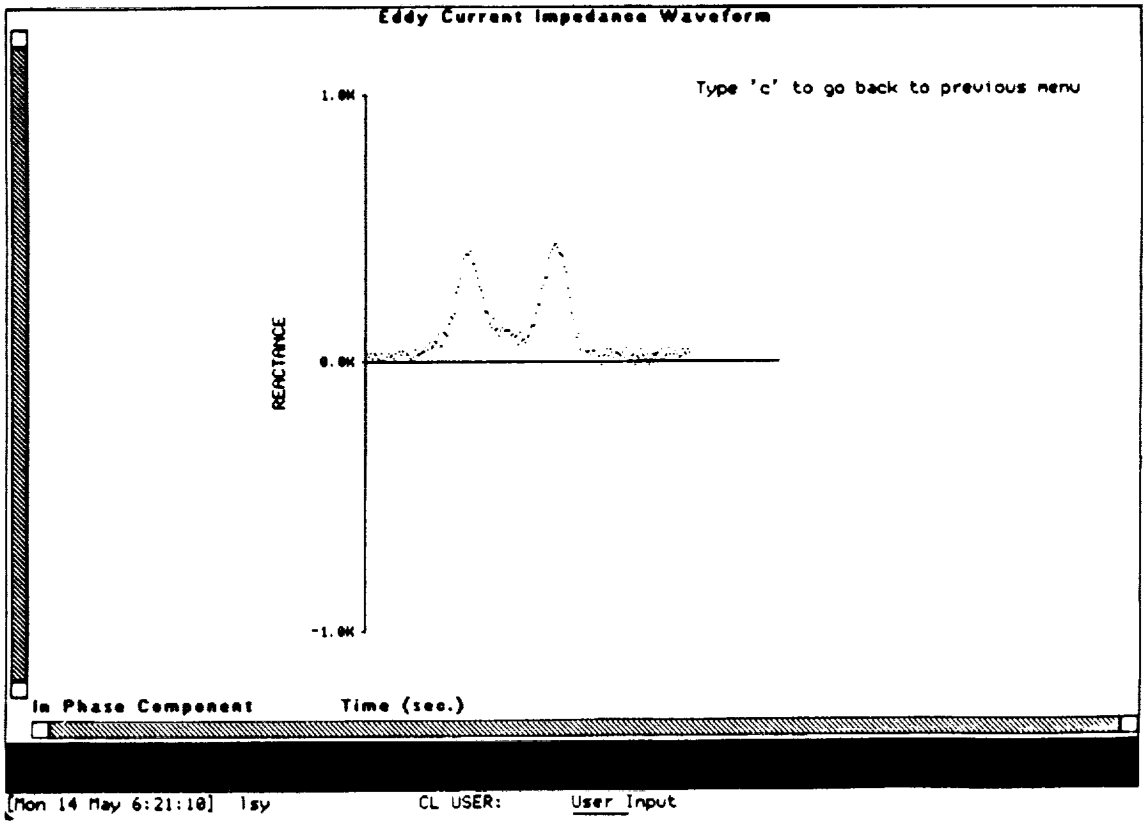
[Mon 14 May 6:15:04] lsy CL USER: User Input

Figure 20. Displays of eddy current data using MacIvory software.



Type 'c' to go back to previous menu

```
[Mon 14 May 6:18:33]  lsy            CL USER:        User Input
```



Type 'c' to go back to previous menu

```
[Mon 14 May 6:21:10]  lsy            CL USER:        User Input
```

Figure 21. Displays of eddy current data using MacIvory software (continued).

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX A

MACIVORY ROUTINES

```
 ;;; ;;; -*- Syntax: Common-Lisp; Package: COMMON-LISP-USER; Base: 10;
Mode: LISP -*-
;;;
;;;-Global Variables-
(defvar *window-size*)
(defvar *resistance* nil)
(defvar *reactance* nil)
(defvar *resistance-average* nil)
(defvar *reactance-average* nil)
(defvar *phase-angle* nil)
(defvar *pathname*)
(defvar *menu-window* (tv:make-window 'dw:dynamic-window
                   :blinker-p nil
                   :expose-p nil
                   ; :default-character-style '(:eurex :italic :large)
                   :more-p nil
                   :save-bits t))
;
(defvar *plot-detail-window* (tv:make-window 'dw:dynamic-window
                   :blinker-p nil
                   :expose-p nil
                   :more-p nil
                   :save-bits t
                    :margin-components
                    '((dw:margin-borders :thickness 1)
                     (dw:margin-label :margin :top
                                   :style (:swiss :bold :small)
                                   :string "Eddy Current Impedance
                                      Waveform"
                                   :centered-p t)
                      (dw:margin-scroll-bar :margin :bottom)
                      (dw:margin-scroll-bar :margin :left)
                      (dw:margin-label :margin :bottom
                                   :style (:swiss :bold :small)
                                   :string "In Phase Component        Time
                                      (sec.)"))))
;
(defvar *plot-detail-window-1* (tv:make-window 'dw:dynamic-window
                   :blinker-p nil
                   :expose-p nil
                   :more-p nil
                   :save-bits t))
```

```
;
(defvar *sample-window-1* (tv:make-window 'dw:dynamic-window
               :blinker-p nil
               :expose-p nil
               :more-p nil
               :save-bits t
               :margin-components
               '((dw:margin-borders :thickness 1)
                 (dw:margin-label :margin :top
                           :style (:swiss :bold :small)
                           :string "Eddy Current Impedance
                              Waveform"
                           :centered-p t)
                 (dw:margin-scroll-bar :margin :bottom)
                 (dw:margin-scroll-bar :margin :left)
                 (dw:margin-label :margin :bottom
                           :style (:swiss :bold :small)
                           :string "In Phase Component        Time
                              (sec.)"))))
;
(defvar *sample-window-2* (tv:make-window 'dw:dynamic-window
               :blinker-p nil
               :expose-p nil
               :more-p nil
               :save-bits t
               :margin-components
               '((dw:margin-borders :thickness 1)
                 (dw:margin-label :margin :top
                           :style (:swiss :bold :small)
                           :string "Eddy Current Impedance
                              Waveform"
                           :centered-p t)
                 (dw:margin-scroll-bar :margin :bottom)
                 (dw:margin-scroll-bar :margin :left)
                 (dw:margin-label :margin :bottom
                           :style (:swiss :bold :small)
                           :string "In Phase Component        Time
                              (sec.)"))))
```

```
;
(defvar  *sample-window-3*  (tv:make-window  'dw:dynamic-window
                    :blinker-p nil
                    :expose-p nil
                    :more-p nil
                    :save-bits t
                     :margin-components
                    '((dw:margin-borders  :thickness  1)
                       (dw:margin-label :margin :top
                                     :style (:swiss :bold :small)
                                     :string "Eddy Current Impedance
                                               Waveform"
                                     :centered-p  t)
                       (dw;margin-scroll-bar  :margin  :bottom)
                       (dw:margin-scroll-bar  :margin  :left))))
;
(defvar  *sample-window-4*  (tv:make-window  'dw:dynamic-window
                    :blinker-p nil
                    :expose-p nil
                    :more-p nil
                    :save-bits t))
;
(defun  configure-windows  ()
   (multiple-value-bind  (screen-left  screen-top  screen-right  screen-bottom)
   (send  tv:main-screen  :inside-edges)
   (let  ((width  (truncate  (/  (-  screen-right  screen-left)  2)))
       (height  (truncate  (/  (-  screen-bottom  screen-top)  2))))
       (send  *sample-window-1*  :set-edges  screen-left  screen-top
                                    width  height)
       (send  *sample-window-2*  :set-edges  screen-left  height
                                    width  screen-bottom)
       (send  *sample-window-3*  :set-edges  width  screen-top
                                    screen-right  height)
       (send  *sample-window-4*  :set-edges  width  height
                                    screen-right  screen-bottom))))
```

```
;;
;;-MAIN PROGRAM-
;
(defun  eddy-current  ()
    (send  *menu-window*  :select)
     (send  *menu-window*  :clear-history)
     (general-information)
     (configure-windows)
     (input-data-file)
   (data)
    (main-menu)
    (send  *terminal-io*  :select))


;
(defun  general-information  ()
  (format *menu-window* "~2%~10TPATTERN RECOGNITION FOR EDDY
                                                CURRENT")
   (format *menu-window* "~20%~10TPRESS 'C' TO CONTINUE")
    (loop  as  char  =  (send  *menu-window*  :any-tyi)
         until  (member  char  '(#\c  #\C)))
     (send  *menu-window*  :clear-history))
;
(defun  input-data-file  ()
  (format  *menu-window*  "~%please  type  data  file  name,")
  (format  *menu-window*  "~%~%example  data  file:  test.dat,  Press  Control
                                                Abort  to  exit  program.>>")
  (let  ((file  (zl:readline  *menu-window*)))
        (setq  *pathname*  (string-append  "andy:>lsy>pattern-recognition>")
                           (string  file)))))
;
(defun  main-menu  ()
  (loop
      as  selection  =  (dw:menu-choose
                  '(("Execute  Pattern  Recognition"  pattern-recognition)
                 ("Plot  Data"  plot-data)
                ("Quit"  quit))
              :prompt  "SELECT  AN  ACTIVITY"
                :momentary-p  nil)
     until  (equal  'quit  selection)
      do
     (execute-command  selection)
    (send  *terminal-io*  :select)))
;
```

```
(defun  execute-command  (command)
    (cond ((equal  command  'plot-data)  (plot-data-menu))
          ((equal  command  'pattern-recognition)  (pattern-recognition))))
;
(defun  plot-data-menu  ()
    (loop as selection = (dw:menu-choose
                  '(("SHOW ALL DATA PLOT(AVERAGE)" show-all)
                   ("SHOW RAW DATA PLOT FOR RESISTANCE" raw-
                        resistance)
                   ("SHOW RAW DATA PLOT FOR REACTANCE" raw-
                        reactance)
                   ("SHOW RAW DATA PLOT FOR IMPEDANCE PLANE"
                        impedance)
                  ("SHOW AVERAGE DATA PLOT FOR RESISTANCE"
                        average-resistance)
                   ("SHOW AVERAGE DATA PLOT FOR REACTANCE" average-
                        reactance)
                   ("QUIT AND RETURN TO MAIN MENU" quit))
                  :prompt "SELECT PLOT OPERATION"
                    :momentary-p nil)
        until (equal 'quit selection)
         do
       (cond ((equal  selection  'show-all)  (show-all))
             ((equal  selection  'raw-reactance)  (plot-full-reac  *plot-detail-
                                                                    window*))
             ((equal  selection  'raw-resistance)  (plot-full-resi  *plot-detail-
                                                                    window*))
           ((equal  selection  'impedance)  (plot-full-impedance  *plot-detail-
                                                                    window-1*))
             ((equal  selection  'average-resistance)  (plot-full-resi-avg  *plot-
                                                                    detail-window*))
             ((equal  selection  'average-reactance)  (plot-full-reac-avg  *plot-
                                                               detail-window*)))))
;
(defun  plot-full-reac  (window)
  (plot-wave-form  window  *reactance*  "REACTANCE")
   (hold-screen-and-continue  window))
;
(defun  plot-full-resi  (window)
  (plot-wave-form  window  *resistance*  "RESISTANCE")
   (hold-screen-and-continue  window))
;
```

```
(defun plot-full-impedance (window)
   (plot-impedance-plane window *resistance* *reactance*)
    (hold-screen-and-continue window))
;
(defun plot-full-resi-avg (window)
   (plot-wave-form window *resistance-average* "RESISTANCE")
    (hold-screen-and-continue window))
;
(defun plot-full-reac-avg (window)
   (plot-wave-form window *reactance-average* "REACTANCE")
    (hold-screen-and-continue window))
;
(defun show-all ()
     (plot-wave-form *sample-window-1* *resistance-average*
                                              "RESISTANCE")
    (plot-wave-form *sample-window-2* *reactance-average* "REACTANCE")
     (plot-impedance-plane *sample-window-3* *resistance-average*
                                              *reactance-average*)
     (result-and-action *sample-window-4*))
;
(defun plot-wave-form (window data string-name)
  (send window :select)
  (send window :clear-history)
  (let ((origin-x (* 3/10 (send window :inside-width)))
      (origin-y (* 1/2 (send window :inside-height)))
      (y-axis-length (- (* 9/10 (send window :inside-height))
                       (* 1/10 (send window :inside-height))))
      (x-axis-length (+ 50 (length data)))
      (max (apply #'max data))
      (min (apply #'min data))
            (label-num) (y-interval-pixel) (y-axis-bottom) (y-axis-begin)
      (y-axis-end))
     (setq y-axis-begin (- origin-y (* 1/2 y-axis-length))
           y-axis-end (+ origin-y (* 1/2 y-axis-length)))
     (graphics:draw-line origin-x y-axis-begin origin-x y-axis-end
                   :stream window)
     (graphics:draw-line origin-x origin-y (+ origin-x x-axis-length) origin-y
                   :stream window)
     (graphics:with-room-for-graphics (window origin-y)
       (graphics:draw-string-image string-name 0 0
                            :stream window
                            :translation (list (- origin-x 45) 0)
                            :attachment-x :center
```

```
                              :rotation (/ pi 2)))
    (cond ((> (abs max) (abs min))
                         (setq label-num (ceiling (abs max) 1000)))
          (t              (setq label-num (+ 1 (ceiling (abs min) 1000)))))
     (setq y-interval-pixel (floor y-axis-length (* 2 label-num)))
     (setq y-axis-bottom (+ origin-y (* y-interval-pixel label-num)))
    (loop for label from (- label-num) to  label-num
        for y      from y-axis-bottom by (- y-interval-pixel)
       do
      (graphics:draw-string (format nil "~1$K" label) (- origin-x 5) y
                           :stream window :opaque nil :attachment-y :center
                       :character-style '(:fix :roman :small)
                       :attachment-x  :right)
       (graphics:draw-line (- origin-x 2) y origin-x y :stream window))
    (loop for data-pt in data
        for x from (+ origin-x 1) by 1
         with scale = (/ 1000 y-interval-pixel)
       do
       (graphics:draw-circle x (- origin-y (/ data-pt scale)) 0.5
                          :stream window))))
;
(defun  plot-impedance-plane  (window  resi-data  reac-data)
   (send window :select)
   (send window :clear-history)
   (graphics:with-graphics-scale (window  1.25)
     (let ((origin-x (/ (* 1/2 (send window :inside-width)) 1.25))
          (origin-y (/ (* 1/2 (send window :inside-height)) 1.25))
          (x-axis-length (send window :inside-width))
          (y-axis-length (send window :inside-height))
          (max-resi (apply #'max resi-data))
          (min-resi (apply #'min resi-data))
          (max-reac (apply #'max reac-data))
          (min-reac (apply #'min reac-data))
           (x-axis-begin) (x-axis-end)(y-axis-begin) (y-axis-end)
           (label-num) (x-axis-start) (y-axis-start) (pixel-interval)
          (axis-length))
       (cond ((> y-axis-length x-axis-length)
              (setq axis-length (- (* 8/10 (send window :inside-width))
                                   (* 2/10 (send window :inside-width)))))
             (t  (setq axis-length (- (* 8/10 (send window :inside-height))
                                  (* 2/10 (send window :inside-height))))))
       (setq x-axis-begin (- origin-x (* 1/2 axis-length))
            x-axis-end   (+ origin-x (* 1/2 axis-length))
      y-axis-begin (- origin-y (* 1/2 axis-length))
      y-axis-end   (+ origin-y (* 1/2 axis-length)))
   (graphics:draw-line x-axis-begin origin-y x-axis-end origin-y :stream
       window)
   (graphics:draw-line origin-x y-axis-begin origin-x y-axis-end :stream
       window)
   (graphics:with-room-for-graphics (window origin-y)
 (graphics:draw-string-image "Resistance" 0 0
                             :stream window
                             :translation (list (- x-axis-begin 35) 0)
                             :attachment-x :center
                             :rotation (/ pi 2))
 (graphics:draw-string-image  "Reactance" 0 0
                             :stream window
                             :translation (list origin-x (+ (- origin-y) 0))
                             :attachment-x :center
                             :attachment-y :center))
 (cond ((> (abs max-resi) (max (abs max-reac) (abs min-resi) (abs min-
       reac)))
       (setq label-num (ceiling (abs max-resi) 1000)))
      ((> (abs max-reac) (max (abs max-resi) (abs min-resi) (abs min-
       reac)))
       (setq label-num (ceiling (abs max-reac) 1000)))
      ((> (abs min-resi) (max (abs max-reac) (abs max-resi) (abs min-
       reac)))
       (setq label-num (+ 1 (ceiling (abs min-resi) 1000))))
      ((> (abs min-reac) (max (abs min-resi) (abs max-reac) (abs max-
       resi)))
       (setq label-num (+ 1 (ceiling (abs min-reac) 1000)))))
 (setq pixel-interval (floor axis-length (* 2 label-num)))
 (setq x-axis-start (- origin-x (* pixel-interval label-num))
       y-axis-start (+ origin-y (* pixel-interval label-num)))
 (loop for label from (zl:minus label-num) to label-num
       for x from x-axis-start by pixel-interval
       for y from y-axis-start by (zl:minus pixel-interval)
       when (or (not (zerop label)) (not (zerop label)))
     do
       (graphics:draw-string (format nil "~1$K" label) x (+ origin-y 10)
                             :opaque nil
                              :stream window :character-style '(:fix :roman
                             :tiny))
       (graphics:draw-string (format nil "~1$K" label) (- origin-x 20) y
                             :opaque nil
```

```
                              :stream window :character-style '(:fix :roman
                             :tiny))
            (graphics:draw-string "-" (- origin-x 2) y :stream window
                              :character-style '(:fix :roman :tiny)
                             :opaque nil)
            (graphics:draw-string "|" x origin-y :stream window
                              :character-style '(:fix :roman :tiny)
                             :opaque nil))
        (loop for data-x in resi-data
              for data-y in reac-data
              with scale = (/ 1000 pixel-interval)
            do
        (graphics:draw-point (+ origin-x (/ data-x scale))
                             (- origin-y (/ data-y scale)) :stream window)))))
;
(defun result-and-action (window)
  (send window :select)
   (send window :clear-history)
    (hold-screen-and-continue window))
;
(defun hold-screen-and-continue (screen)
   (graphics:draw-string "Type 'c' to go back to previous menu"
                   (* 6/10 (send screen :inside-width))
                   (* 1/10 (send screen :inside-height))
                   :stream screen)
   (loop as char = (send screen :any-tyi)
     until (member char '(#\c #\C))))
(defun data ()
  (loop with data = (get-data)
      with resistance-sum = 0
      with reactance-sum = 0
      with resistance-averages = nil
      with reactance-averages = nil
      ;with phase-angle = nil
      with num-data = (accept 'integer
                      :stream *menu-window*
                      :prompt
                      "please type the number of data for calculating
                      average")
      for resistance in data by #'cddr
      for reactance in (cdr data) by #'cddr
      for count from 1
      collect resistance into resistance-list
```

```
       collect reactance into reactance-list
       ; do
       ;(setq phase-angle (cons (- 180 (* (atan reactance resistance) (/ 180
            (float pi 1.0)))) phase-angle))
       if (zerop (mod count num-data))
         do
            (setq resistance-averages (cons (/ resistance-sum num-data)
             resistance-averages)
               reactance-averages (cons (/ reactance-sum num-data)
                  reactance-averages)
              resistance-sum 0
              reactance-sum 0)
       else
         do
           (setq resistance-sum (+ resistance-sum resistance)
              reactance-sum (+ reactance-sum reactance))
      finally  (setq *resistance* resistance-list *reactance* reactance-list
                  *resistance-average* (reverse resistance-averages)
                *reactance-average* (reverse reactance-averages))))
              ; *phase-angle* (reverse phase-angle)))
;
(defun get-data ()
    (with-open-file (data-stream *pathname*
                    :direction :input
                    :error     :reprompt)
       (loop while (listen data-stream)
          collect (read data-stream))))
```

APPENDIX B

PROGRAM RSMED.C

PROGRAM RSMED.C

This program is used to perform the data analysis and plot various eddy current graphs for display purposes. Much of this program is based upon the software originally written in BASIC by Dr. Brian Lempriere and his group at Boeing Aerospace in Seattle.

```
# include <math.h>
# include <fcntl.h>
# include <stdio.h>
# include <alloc.h>
# include <graphics.h>
# include <stdlib.h>
# include <string.h>
# include <process.h>
# include "worlddr.h"
# include "segraph.h"
# include "asyncxx.h"
# include "hpplot.h"

      int err, i, n, grcolor, l;
      char xtitle[80];
      float *vi, *vq, *vph, *fdata;
      void ptime(float *vi, float *vq, float *vph, float *fdata, int l);
      void impedanc(float *vi, float *vq, int l);

main()
{
 int i=0, j=0, k, kk, jj, lth, hl;
 int lsb1, msb1,.lsb2, msb2;
 float *RO, gh, hg, dd=0., ee=0.,xyd, xye, f;
 float rr, A1, B1, C1, D1, xu, yu, xerr, yerr;
 double d, e, xd, ye;
 char dext[] = ".dat", a[1];
 unsigned char *string1, string2[80] ;
 FILE *dstream, *stream, *handle;
 char fname[10], dname[10], ename[6], degree[5];
 char y[4], yy[4], *de;
 string1 = (unsigned char *)malloc(35000);
 RO = (float *)malloc(80);
 fdata = (float *)malloc(165000);
 vi = (float *)malloc(165000);
 vq = (float *)malloc(165000);
 vph = (float *)malloc(165000);
 printf("  Please input data file name (e.g. HOLE2) : ");
 gets(dname);
```

```
stpcpy(ename,dname);
strcat(dname,dext);
printf("  Please input rotation degree (e.g. 72) : ");
de = gets(degree);
rr = atof(de);
stream = fopen(dname, "r");
while(fgets(string2, 8, stream)!=NULL)
{
  RO[j] = atof(string2);
  j = j + 1;
}
A1 = RO[34];
B1 = RO[36];
C1 = RO[38];
D1 = RO[40];
printf("  %f  ,  %f  ,  %f  ,  %f\n",A1,B1,C1,D1);
rr = rr * 3.1415927 / 180.;
printf("  Do you want to save the ascii file? (y/n) ");
gets(y);
if(y[0] == 89 | y[0] == 121)
{
  dstream = fopen("eddy.out", "w");
}
printf("  Do you want to view the signal graphics? (y/n) ");
gets(yy);
for (k = 1; k < 5; k++)
{
  char cext[5] = ".00";
  kk = k - 1;
  itoa(kk, a, 10);
  strcat(cext, a);
  gh = 0.;
  hg = 0.;
  xerr = 0.;
  yerr = 0.;
  stpcpy(fname, ename);
  strcat(fname, cext);
  puts(fname);
  handle = fopen(fname, "rb");
  if(handle == NULL)
  exit(0);
  hl = fileno(handle);
  lth = abs(filelength(hl));
  l = lth / 8;
```

```
fread(string1, 1, lth, handle);
for(j = 0; j < l; j++)
{
   vi[j] = 0.;
   vq[j] = 0.;
   vph[j] = 0.;
}
for(j = 0; j < l; j++)
{
   jj = j * 8;
   msb1 = string1[jj + 7];
   lsb1 = string1[jj + 8];
   vi[j] = 5. * ((256. * msb1) + lsb1 - 32767.) / 32767.;
   msb2 = string1[jj + 9];
   lsb2 = string1[jj + 10];
   vq[j] = 5. * ((256. * msb2) + lsb2 - 32767.) / 32767.;
   d = fabs(vi[j] - gh);
   e = fabs(vq[j] - hg);
   xd = abs(d - dd);
   ye = abs(e - ee);
   if( xd > 0.5){
   xyd = d;}
   if( ye > 0.5){
   xye = e;}
   if( d > 1.0 ){
   vi[j] = xyd + vi[j];}
   if( e > 1.0 ){
   vq[j] = xye + vq[j];}
   gh = vi[j];
   hg = vq[j];
   dd = d;
   ee = e;
   xu = A1 * vi[j] + B1 * vq[j];
   yu = C1 * vq[j] + D1 * vi[j];
   fdata[j] = (i + j) / 300.;
   vi[j] = (cos(rr) * xu - sin(rr) * yu - xerr) + 2400;
   vq[j] = (sin(rr) * xu + cos(rr) * yu - yerr) - 1800;
   if(xerr == 0. | yerr == 0.)
   {
      xerr = vi[0];
      yerr = vq[0];
      vi[0] = 0.000001;
      vq[0] = 0.000001;
   }
```

```
      vph[j] = atan2(vq[j], vi[j])*180./3.1415926;
      if(-750. < vi[j] & vi[j] < 750. | -750. < vq[j] & vq[j] < 750.)
      {
          vph[j] = 45.;
      }
    }
  i = j * k;
  if(y[0] == 121 | y[0] == 89)
  {
    for(j = 0; j < l; j++)
    {
      f = atan2(vq[j], vi[j]) * 180. / 3.1415927;
      fprintf(dstream," %f   %f  , %f\n", vi[j], vq[j], f);
    }
  }
  if(yy[0] == 121 | yy[0] == 89)
  {
     ptime(vi, vq, vph, fdata, l);
     impedanc(vi, vq, l);
  }
  fclose(handle);
 }
 fclose(stream);
 fclose(dstream);
}

void ptime(float *vi, float *vq, float *vph, float *fdata, int l)
{
    InitSEGraphics("c:\\turboc");
    SortDataX(fdata, vi+96, l-97, 1);
    SortDataX(fdata, vq+96, l-97, 1);
    SetCurrentWindow(7);
    SetAxesType(0, 0);
    SelectColor(4);
    ScalePlotArea(fdata[0], -8000., fdata[l-97], 8000.);
    SetXYIntercepts(fdata[0], 0.);
    DrawXAxis(1., 0);
    DrawYAxis(500., 0);
    LabelXAxis(2,0);
    LabelYAxis(2,0);
    BorderCurrentWindow(16);
    strcpy(xtitle,"       TIME (sec.)     ");
    TitleXAxis(xtitle);
    TitleWindow("  Eddy-Current Impedance Waveform  ");
    strcpy(xtitle," Resistance  ");
    TitleYAxis(xtitle);
    LinePlotData(fdata+96, vi+96, l-97, 4, 0);
    SetCurrentWindow(9);
    SetAxesType(0,0);
    SelectColor(4);
    ScalePlotArea(fdata[0], -8000., fdata[l-97], 8000.);
    SetXYIntercepts(fdata[0], 0.);
    DrawXAxis(1., 0);
    DrawYAxis(500.0, 0);
    LabelXAxis(2,0);
    LabelYAxis(2,0);
    BorderCurrentWindow(16);
    strcpy(xtitle,"     TIME (sec.)    ");
    TitleXAxis(xtitle);
    TitleWindow(" Eddy-Current Impedance Waveform  ");
    strcpy(xtitle," Reactance   ");
    TitleYAxis(xtitle);
    LinePlotData(fdata+96, vq+96, l-97, 4, 0);
    SetCurrentWindow(8);
    SetAxesType(0,0);
    SelectColor(4);
    ScalePlotArea(fdata[0], -300., fdata[l-97], 300);
    SetXYIntercepts(fdata[0], 0.);
    DrawXAxis(1., 0);
    DrawYAxis(100.0, 0);
    LabelXAxis(1,0);
    LabelYAxis(1,0);
    BorderCurrentWindow(16);
    strcpy(xtitle,"  TIME(sec.)   ");
    TitleXAxis(xtitle);
    TitleWindow(" Eddy-Current Phase Angle ");
    strcpy(xtitle," Phase Angle ");
    TitleYAxis(xtitle);
    LinePlotData(fdata+96, vph+96, l-97, 4, 0);
}

void impedanc(float *vi, float *vq, int l)
{
    int sy;
    SetCurrentWindow(10);
    SetAxesType(0,0);
    SelectColor(4);
    ScalePlotArea(-8000., -8000., 8000, 8000.);
```

```
   SetXYIntercepts(0., 0.);
   DrawXAxis(500., 0);
   DrawYAxis(500., 0);
   LabelXAxis(4, 0);
   LabelYAxis(4, 0);
   BorderCurrentWindow(16);
   strcpy(xtitle," Resistance ");
   TitleXAxis(xtitle);
   TitleWindow("  Eddy-Current Impedance Plane  ");
   strcpy(xtitle," Reactance ");
   TitleYAxis(xtitle);
   LinePlotData(vi+96, vq+96, l-97, 4, 0);
   sy = getch();
   if(sy == 121 | sy == 89)
   ScreenDump(3, 0, 3, 3, 2, 0, 0, &err);
   {
    PlotterOn();
   }
   ClearWindow();
   CloseSEGraphics();
}
```

APPROVAL

EDDY CURRENT INSPECTION OF GRAPHITE FIBER COMPONENTS

By G.L. Workman and C.C. Bryson

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

Paul H. Schuerer

PAUL H. SCHUERER
Director, Materials and Processes Laboratory

☆ U.S. GOVERNMENT PRINTING OFFICE 1990—531—081/20220