ORNL-6366

OAK RIDGE NATIONAL LABORATORY

MARTIN MARIETTA

# An Eddy-Current Laboratory Test System Using Commercial Equipment

C. V. Dodd
L. M. Whitaker
W. E. Deeds



OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A07 Microfiche A01

ORNL-6366
Distribution
Categories UC-23,
UC-25, UC-38

Metals and Ceramics Division

# AN EDDY-CURRENT LABORATORY TEST SYSTEM USING COMMERCIAL EQUIPMENT

C. V. Dodd
L. M. Whitaker
W. E. Deeds

Date Published: April 1987

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under Contract DE-AC05-84OR21400

CONTENTS

# AN EDDY-CURRENT LABORATORY TEST SYSTEM USING COMMERCIAL EQUIPMENT

C. V. Dodd, L. M. Whitaker,* and W. E. Deeds†

ABSTRACT

The combination of a Hewlett Packard impedance analyzer and a laboratory computer has produced a versatile and effective eddy-current laboratory test instrument. This combination has allowed a wide range of eddy-current measurements to be quickly, accurately, and easily performed. Also, this combination has been programmed to make absolute resistivity measurements, multiple-frequency/multiple-property measurements, normalized impedance measurements, and Bode plots (magnitude and phase vs frequency). These measurements are performed for a variety of coil and conductor combinations, including reflection, through-transmission, and single-coil configurations. The intelligence of the computer, combined with the ease of operation and programming of the impedance analyzer over the IEEE-488 bus, can make quick work of jobs that formerly took much longer.

## INTRODUCTION

Theoretical studies in eddy-current testing show that many different types of measurements are possible. However, the time, expense, and effort to set up these measurements often result in their not being used. Although it is quite easy to run an eddy-current test as a computer simulation, the effort and detail involved in performing the actual test are considerable. Now, however, combining a Hewlett Packard (HP) 4192A LF Impedance Analyzer with an IBM System 9000 computer enables us to make experimental eddy-current measurements, record the results, and analyze them with little more effort than running a computer simulation. The computer and analyzer are connected over the IEEE-488 bus, and all operating parameters can be controlled and varied over the bus. The eddy-current coil and the test samples must be furnished, and the probe must be moved

*Summer research participant, Eckerd College, St. Petersburg, Fla.
†Adjunct research participant, University of Tennessee, Knoxville.

from sample to sample in the prescribed manner; the data gathering, analysis, and presentation, however, can all be done by the computer. The probe motion over the respective samples can also be computer controlled by using the Modulynx Motion Controllers, or the probe can be positioned manually. The analyzer can make either impedance measurements or send-receive measurements. The sensitivity of the analyzer is high enough (0.01° phase shift and 0.001 dB magnitude) to make most eddy-current measurements except small-defect measurements. The stability of the analyzer is good, and the noise can be reduced to a very small level by averaging; the speed, though, is not adequate for many production tests. However, these tests can be simulated on a computer and then verified by using the impedance analyzer before investing in the construction of an expensive production instrument. This report describes some of the eddy-current measurements that can be made with the impedance analyzer connected to the IBM and the programs to run these measurements. The measurements include normalized impedance plots, through-transmission absolute conductivity measurements, multiple-frequency measurements of multiple-property problems, Bode plots, and Bode difference plots.

## EQUIPMENT

The HP 4192A LF Impedance Analyzer is used to perform the measurements. The IBM System 9000 computer is used to program the frequency, output voltage level, and measurement mode. The programs are written in FORTRAN 77 and run under the CSOS 2.0 operating system. The system subroutines are preceded by an underscore: _GETKEY, _SYSFUNC, etc. All of the system subroutines are described in the *Interface Library Reference Manual*, "Graphics Routines" (Chap. 7) and "System Library" (Chap. 16). The computer can also control the motion of the probe by using the Modulynx Mechanical Scanner, which in turn controls the Velmex Mechanical Scanners. These general purpose scanners can control motion in up to six independent axes, with a resolution of 0.006 mm (0.00025 in.).

## COIL SETUP

The apparatus setup and circuit diagrams are given for each program. Through-transmission coils and reflection coils (Fig. 1) are used in the

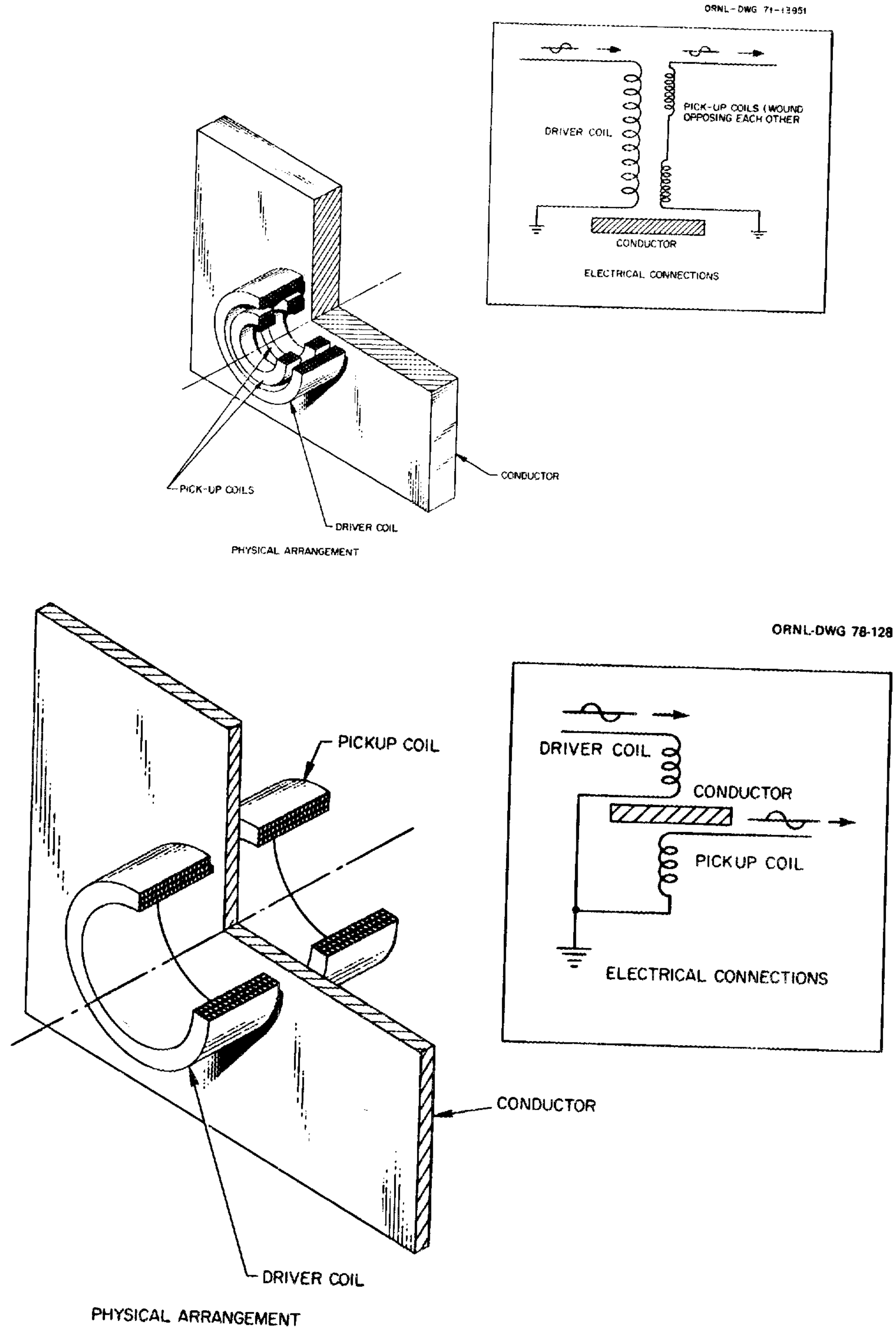



Fig. 1. Coil and conductor arrangement for reflection and through-transmission probes.

programs that make phase and magnitude readings. The program NORIMP makes normalized impedance measurements of a single absolute coil (Fig. 2). The Bode plots and Bode difference plots can be made with any type of coil system.

## PROGRAM CONDTF

The program CONDTF makes an absolute measurement of the resistivity (the reciprocal of electrical conductivity) by using the through-transmission technique. The coil configuration for through-transmission measurements is shown in Fig. 1, and the apparatus to hold the samples is shown in Fig. 3. A discussion of the errors caused by coil construction, sample preparation, sample mounting, temperature effects, and circuit layout is given in an earlier report.[1] Although the apparatus was designed for use with an oil bath, accurate measurements can be made in air. The program will correct for temperature if the sample temperature coefficient and the ambient temperature are given. For the higher-resistivity materials, small temperature variations have a small effect on the resistivity. The major error probably will be due to the inhomogeneities of the sample itself. In general, if reasonable care is taken, accuracies of ~1% can be achieved.

The program will make a series of "air" readings of the magnitude and phase at different frequencies. Then the operator is instructed to place the sample between the coils and repeat the readings. The program then starts with a "guess" for the resistivity and computes the magnitude and phase for the three assumed resistivities. (A nominal resistivity, a small decrease from nominal, and a small increase are used.) If the nominal assumed resistivity produces a phase shift within 0.005° from air to conductor, the program proceeds to the next frequency. If not, a correction on the resistivity is made using a polynomial in the inverse of the measured phase shift of the form

$$\text{Resistivity} = C1/(\text{Phase}) + C2/(\text{Phase})^2 + C3/(\text{Phase})^3 \ .$$

The constants for the polynomial are calculated from the three computed values of phase shift for the assumed values of resistivity. Using this

ORNL-DWG 86-15954

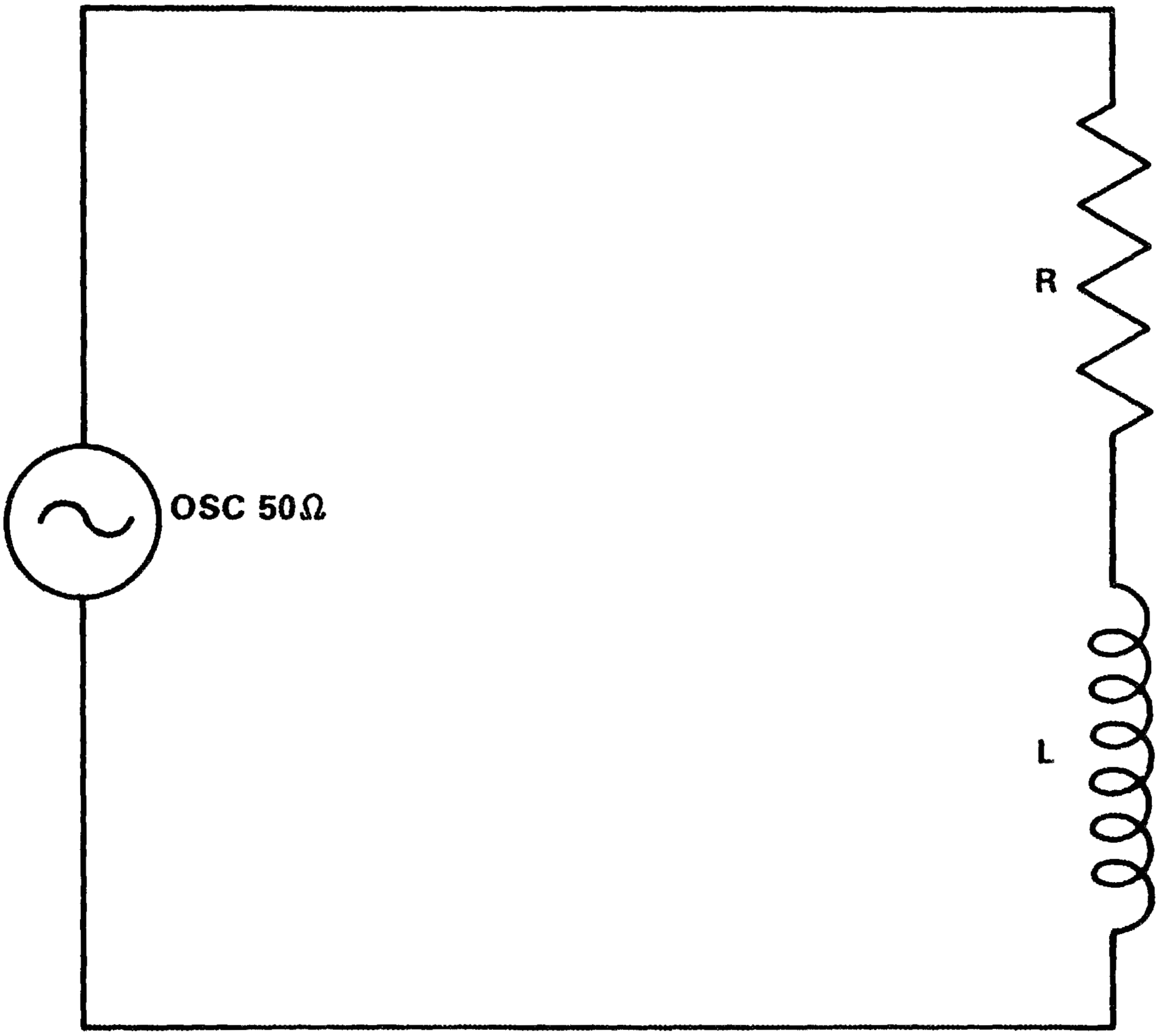



Fig. 2. Circuit for a single absolute coil.

ORNL-DWG 86-15950

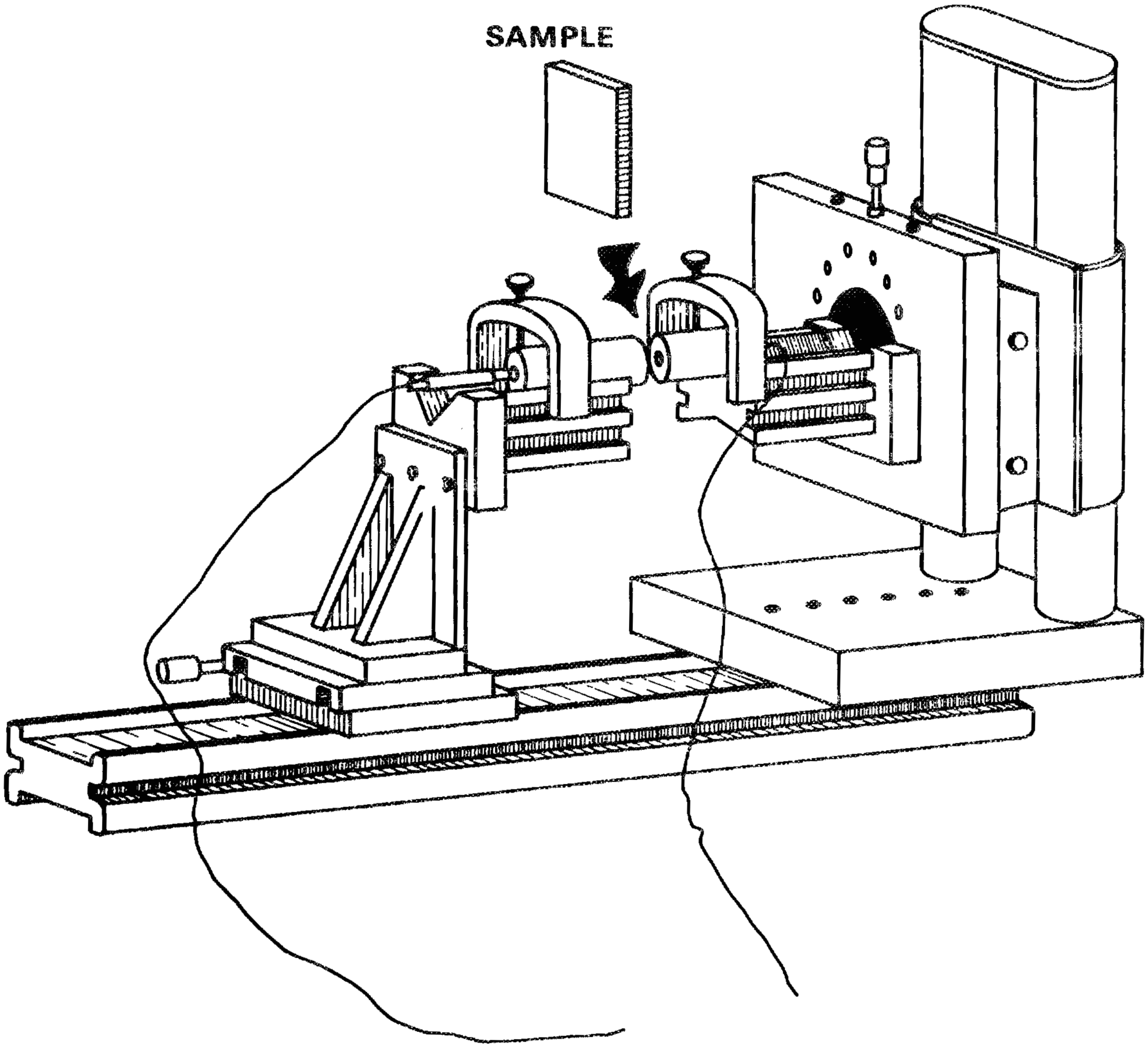



Fig. 3. Coil holder for through-transmission resistivity measurements.

new value of resistivity, a new set of phases and magnitudes is calculated. The standard deviation of the resistivity, calculated from the standard deviation of the phase measurements, is printed at each frequency. If the frequencies chosen are too low, there will be very little phase shift when the sample is inserted, and this standard deviation will be large. If the frequencies are too high, the magnitude of the received signal will be too small to make an accurate phase measurement on the signal, and the standard deviation will also be too large. In addition, if the phase goes through 360°, the corrections used to determine the resistivity will not work, and the program will hang up in a loop. As the coil approaches resonance, both the magnitude and phase vary by a large amount because of parasitic capacitances, which are only approximately known. This contributes a large error at these higher frequencies, and they should be avoided. The resistivity calculated at one frequency is used as the "guess" for the next frequency. The choice of coil size, number of turns, and frequency must be matched to the sample size, thickness, and approximate resistivity for optimum accuracy. The calculations are performed assuming a flat plate several times larger than the coil diameter, but slight curvatures can be accommodated without too much error. Sample diameters at least 50 times the coil diameter will produce less than 1% error.

The through-transmission coils are connected to the impedance analyzer (Fig. 4). Attenuators can be added to the reference input leg of the circuit to make its amplitude match the sample amplitude. The analyzer will give a higher resolution reading on the magnitude if the signal amplitude ratio is <20 dB, but the computed amplitude will differ from the measured value by the value of the in-line attenuators. However, the amplitude difference, which the program computes, is unaffected by the insertion of in-line attenuators.

## INPUT DATA

The input data for this program have a large effect on the accuracy of the calculations. Therefore, the data must be as precise as possible.

ORNL-DWG 86-15952

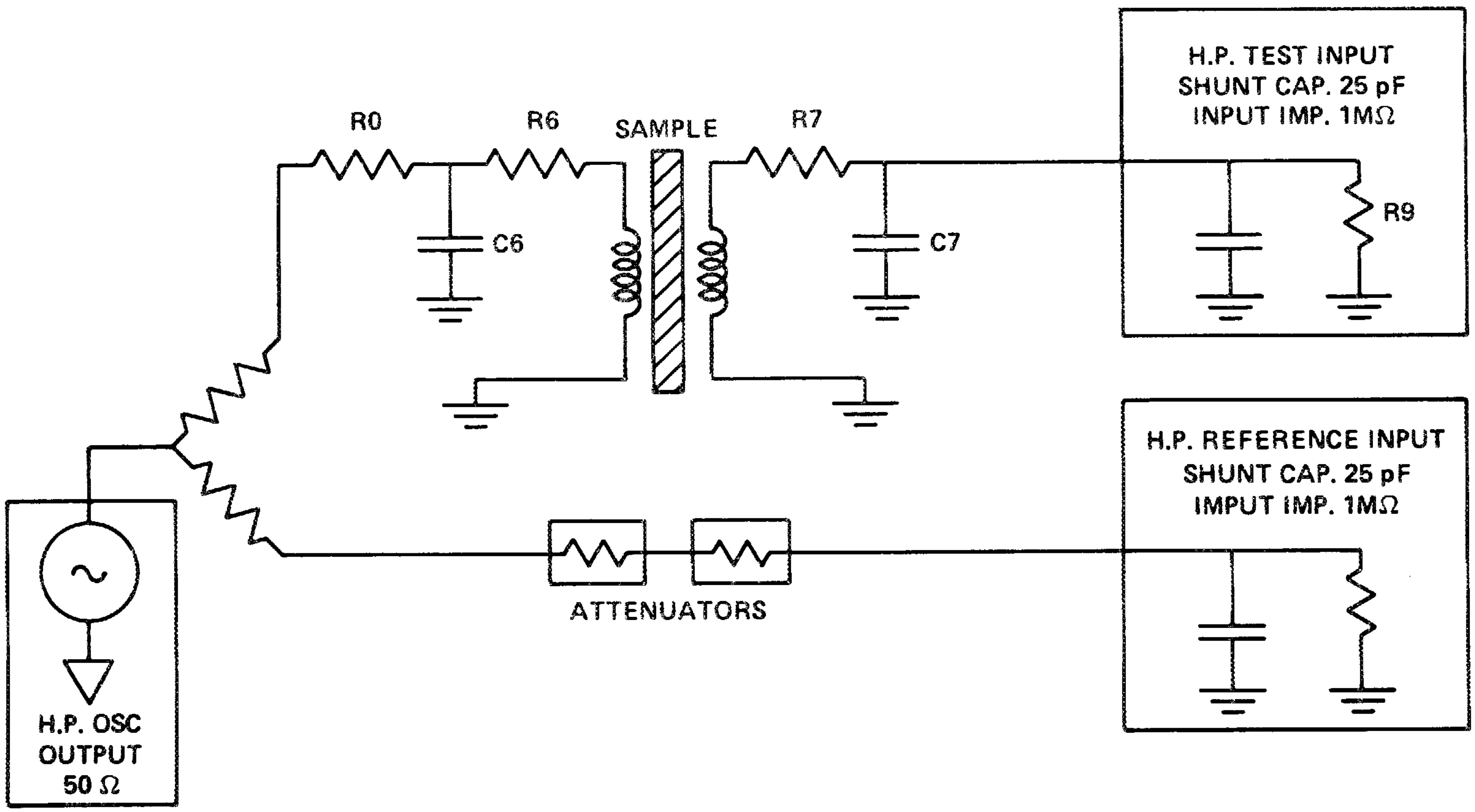



Fig. 4. Impedance analyzer connections for through-transmission measurements.

Figure 5 shows a view of the coil and conductor arrangement. The input data are as follows (all lengths are in inches, temperatures in Celsius).

DRIVER COIL VALUES

DATA R1 - inner radius, R2 - outer radius, L3 - length

TNDR - number of turns, RDCDR - dc resistance of coil

L6 - driver to conductor spacing

RECEIVER COIL VALUES

DATA R3 - inner radius, R4 - outer radius, L4 - length

TNPU - number of turns, RDCPU-dc resistance of coil

DATA RESISI - initial guess at resistivity, needs no accuracy

THICKN - sample thickness, COILDS - distance between two coils

DATA PERM/4*1 /, relative permeability of the conductor; usually equals 1

NMES - number of measurements taken, NFT - number of frequencies

DATA GAIN - voltage gain, left at unity for impedance analyzer

TC - temperature coefficient of material, TEMP - room temperature

SUBROUTINES

There are only three subroutines in this program: _GETTIME, BESSEL, and RFLCKT. _GETTIME is a systems subroutine (refer to the systems library). BESSEL and RFLCKT, along with the other subroutines, are described and listed in the Appendix.

MAIN PROGRAM

A listing of CONDTF follows.

```
      PROGRAM CONDTF
C     CONDTF VERSION -  8 OCTOBER 1986
C     SPECIAL VERSION OF THROUGH TRANSMISSION PROGRAM TO MAKE ABSOLUTE
C     CONDUCTIVITY MEASUREMENTS.MEASURES THE MAGNITUDE AND PHASE USING THE
C     HP 4192A IMPEDANCE ANALYZER , ADDRESS 3=C ON GPIB BUSS. THE
C     CONDUCTIVITY IS VARIED AND THE MAGNITUDE AND PHASE RECALCULATED
C     UNTIL THE MEASURED PROPERTIES MATCH THE CALCULATED PROPERTIES.
C     THIS PROGRAM MAKES MULTIPLE MEASUREMENTS AT MULTIPLE FREQUENCIES.
```

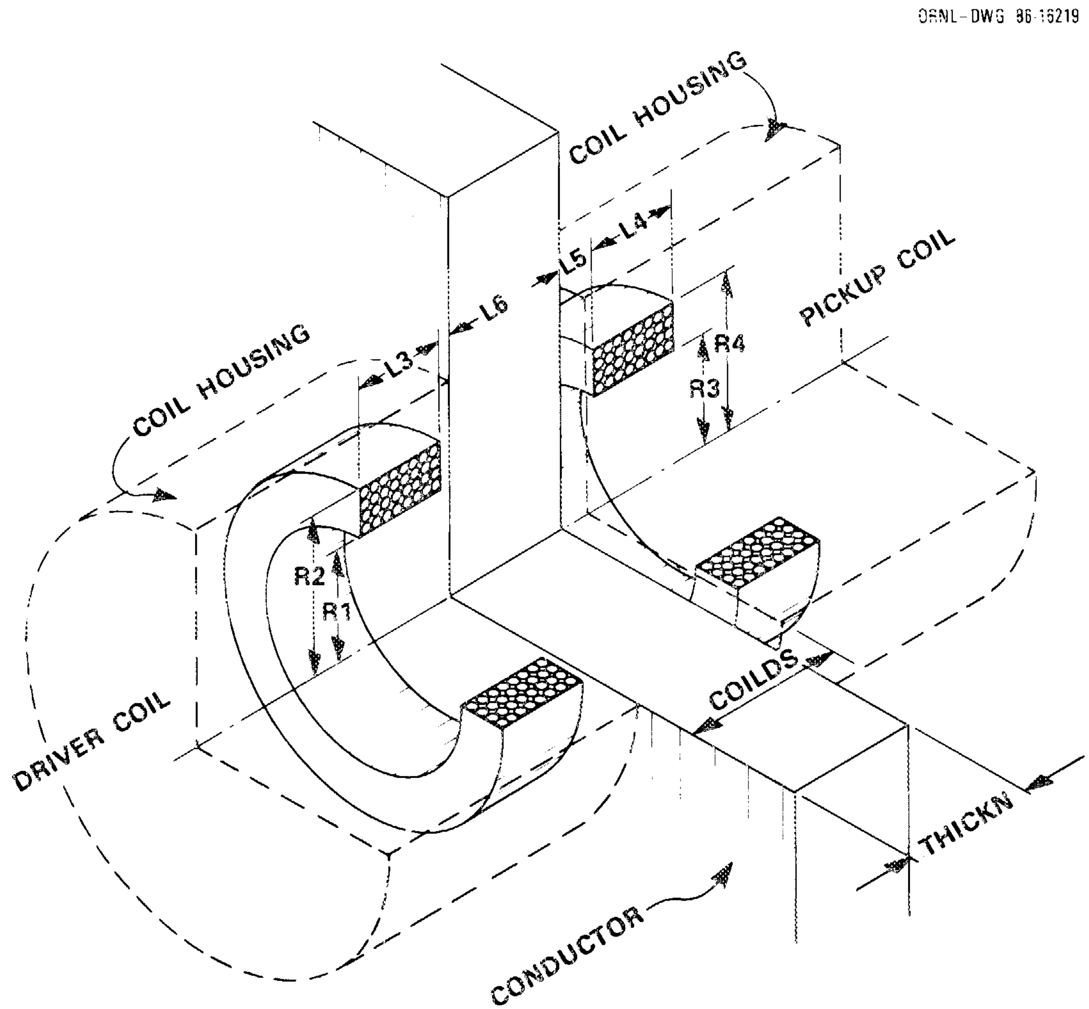



Fig. 5. Coil and conductor dimensions for through-transmission measurements.

```
C      COILDS=COIL FACE-TO-FACE DISTANCE, MEASURED WITH INSULATION ON COILS
C      GAP=CONDUCTOR-TO-PICKUP SPACING, MEASURED WITH INSULATION
C      LOU=LOGICAL OUTPUT HARD COPY UNIT
C      L6=DRIVER-TO-CONDUCTOR SPACING,(EQUALS FACE INSULATION ON EITHER COIL)
C      L5=GAP+L6=PICKUP-TO-CONDUCTOR SPACING,INCLUDING COIL INSULATION
C      GAP IS ASSUMED ON PICKUP SIDE,CONDUCTOR IS AGAINST DRIVER
C      MAGN=(MAGNITUDE WITH SAMPLE)/(MAGNITUDE IN AIR)
C      NSTOP=1,SEARCH FOR MEASURED RHO VALUE;NSTOP=2,CALCULATE AIR VALUE
C      DPHA(NF)=MEASURED PHASES - AIR PHASES
C      RESISI ASSUMED RESISTIVITY OF SAMPLE,PRINTED AS DC RESIS,MICROHM-CM
C      THICKN=THICKNESS OF SAMPLE IN INCHES
       IMPLICIT REAL*8 (A-H,O-Z)
       REAL*8 L3,L4,L5,L6
       CHARACTER CHR1*4,CHR2*5
       DIMENSION WUSR(4),PERM(4),RHO(4),R(3,3)
       DIMENSION TMUTRE(4),TMUTIM(4)
       DIMENSION DRIVRE(4),DRIVIM(4)
       DIMENSION PICKRE(4),PICKIM(4)
       DIMENSION TMAG(4),PHASE(4),FREQ(6),SDVR(6)
       DIMENSION SPHA(6),SMAG(6),AMAG(6),APHA(6)
       DIMENSION AVMA(6),AVPA(6),SDMA(6),SDPA(6),DMAG(6),DPHA(6)
       DIMENSION AVMM(6),AVPM(6),SDMM(6),SDPM(6),SSMAG(6),SSPHA(6)
       DATA TWOPI, RAD /6.28318531, 57.2957795/
C      DRIVER COIL VALUES
       DATA R1/.0750/,R2/.123/,L3/.050/,TNDR/112./,RDCDR/4.36/,L6/0.004/
C      RECEIVER COIL VALUES
       DATA R3/.0750/,R4/.1230/,L4/.050/,TNPU/112./,RDCPU/4.26/
C      DRIVER COIL VALUES  - 200W
C      DATA R1/.1475/,R2/.251/,L3/.100/,TNDR/193./,RDCDR/5.8/,L6/0.004/
C      RECEIVER COIL VALUES - 200W
C      DATA R3/.1485/,R4/.2505/,L4/.100/,TNPU/193./,RDCPU/5.8/
       DATA RESISI/1187.4/,THICKN/.2018/,COILDS/.205/
       DATA PERM/4*1./,NMES/20/,NFT/4/,FREQ/5.E4,1.E5,2.E5,5.E5,0.,0./
       DATA GAIN/1./,TC/0.10800/,TEMP/23.0/
```

```
      DATA LOU/6/,LANA/43/,NSTOP/1/
      OPEN(LOU,FILE='#PR')
C     HP 4192A IMPEDANCE ANALYZER IS ADDRESS 3=C ON GPIB BUSS
      OPEN(LANA,FILE='#BUSC?')
      WRITE(0,*)'THROUGH TRANSMISSION MEASUREMENT OF ELECTRICAL
     *CONDUCTIVITY'
C     CONNECT HP POWER SPLITTER TO 50 OHM OSCILLATOR OUTPUT
C     CONNECT ONE OUTPUT TO CHANNEL A AND THE OTHER TO THE DRIVER
C     COIL.CONNECT THE PICKUP COIL TO CHANNEL B.A RESISTOR MAY BE
C     PLACED IN SERIES WITH THE DRIVER COIL.BE SURE TO PUT THE RIGHT
C     VALUE IN THE PROGRAM FOR ALL COMPONENTS AND MEASUREMENTS.
C     WRITE(0,*)'ENTER ',NFT,' FREQUENCIES THAT YOU WISH TO TEST IN HERTZ'
C     READ(0,*)(FREQ(NF),NF=1,NFT)
      CALL_GETTIME(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
      WRITE(LOU,2)IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT(1X,/,'CONDTF   TIME ',I2,':',I2,':',I2,'  DATE ',I2,'/',I2,
     *'/',I2)
      DCRES=RESISI
C
C     THE NORMALIZED COIL DIMENSIONS ARE CALCULATED
C
      RBAR=.5*(R1+R2)
      GAP=COILDS-THICKN
      L5=GAP+L6
C
      WRITE(LOU,9)COILDS,THICKN,GAP
    9 FORMAT(' ACTUAL DIMENSIONS(IN.) ',/, ' COIL FACE TO FACE ',F7.5,
     C' SAMPLE THICK ',F7.5,' GAP ',F7.5)
      WRITE(LOU,50)RBAR
      WRITE(LOU,60)
      WRITE(LOU,70)R1,R2,L3,TNDR,L6
      WRITE(LOU,80)R3,R4,L4,TNPU,L5
C     WRITE(LOU,85)
   50 FORMAT(' MEAN RADIUS ',F12.5,' INCHES ')
   60 FORMAT('    COIL    INN. RAD    OUT. RAD    LENGTH    TURNS EA
     1  0 L.O./M L.O')
```

```
   70 FORMAT(' DRIVER  ',3(F11.4),F11.1,F13.4)
   80 FORMAT(' PICK-UP ',3(F11.4),F11.1,F13.4)
   85 FORMAT(1X)
      R1=R1/RBAR
      R2=R2/RBAR
      R3=R3/RBAR
      R4=R4/RBAR
      L3=L3/RBAR
      L4=L4/RBAR
      L5=L5/RBAR
      L6=L6/RBAR
      THICKN=THICKN/RBAR
C     THE CIRCUIT PARAMETERS ARE NOW GIVEN.
C     R0=DRIVER SERIES RESISTANCE(OHMS)
C     R9=PICKUP SHUNT RESISTANCE(OHMS)
C     CAPDR=DRIVER SHUNT CAPACITANCE(FARADS)
C     CAPPU=PICKUP SHUNT CAPACITANCE(FARADS)
C     V0=DRIVER OUTPUT VOLTAGE(RMS VOLTS)
C
      R0=3184.1
      R9=1.0E6
      CAPDR=1.22E-10
      CAPPU=9.15E-11
      V0= 1.0
C     MAKE READINGS ON SAMPLE IN AIR
      WRITE(0,*)'             START AIR READINGS WITHOUT SAMPLE'
      WRITE(0,65)
      WRITE(0,66)
      WRITE(LOU,65)
      WRITE(LOU,66)
65    FORMAT('          |      AIR READINGS       ||     MATERIAL READINGS
     *   ||  DIFFERENCE   | ')
66    FORMAT(' FREQ(HZ)  MAG:AVG  SDV PHA:AVG  SDV  MAG:AVG  SDV PHA:AVG
     * SDV      MAG      PHA')
```

```
C     PARAMETERS ARE SET IN IMPEDANCE ANALYZER OVER GPIB BUSS
   75 DO 90 NF=1,NFT
      FREK=FREQ(NF)/1000.
      WRITE(LANA,*)'A5B2V1T3F0'
      WRITE(LANA,81)FREK
   81 FORMAT(1X,'FR',F9.3,'EN')
      WRITE(LANA,82)V0
   82 FORMAT(1X,'OL',F5.3,'EN')
      SPHA(NF)=0.
      SMAG(NF)=0.
      AVMA(NF)=0.
      SSMAG(NF)=0.
      AVPA(NF)=0.
      SSPHA(NF)=0.
      WRITE(LANA,*)'EX'
      READ(LANA,86)CHR1,XMAG,CHR2,XPHA,CHR4
   84 AMAG(NF)=0.0
      APHA(NF)=0.0
      DO 89 II=1,NMES
      WRITE(LANA,*)'EX'
      READ(LANA,86)CHR1,XMAG,CHR2,XPHA
   86 FORMAT(A4,E11.2,A5,E11.2)
      AMAG(NF)=AMAG(NF)+XMAG/FLOAT(NMES)
      APHA(NF)=APHA(NF)+XPHA/FLOAT(NMES)
   87 FORMAT(4X,F12.3,3X,F12.3,3X,F12.3,3X,F12.3)
      AVMA(NF)=AVMA(NF)+XMAG
      SSMAG(NF)=SSMAG(NF)+XMAG*XMAG
      AVPA(NF)=AVPA(NF)+XPHA
      SSPHA(NF)=SSPHA(NF)+XPHA*XPHA
   89 CONTINUE
      AVMA(NF)=AVMA(NF)/FLOAT(NMES)
      SDMA(NF)=SQRT(ABS(SSMAG(NF)-AVMA(NF)*AVMA(NF)*FLOAT(NMES))
     *                  /FLOAT(NMES-1))
      AVPA(NF)=AVPA(NF)/FLOAT(NMES)
```

```
      SDPA(NF)=SQRT(ABS(SSPHA(NF)-AVPA(NF)*AVPA(NF)*FLOAT(NMES))
     *               /FLOAT(NMES-1))
      WRITE(0,91)FREQ(NF),AVMA(NF),SDMA(NF),AVPA(NF),SDPA(NF)
   91 FORMAT(  1PE9.3,1X,0P2(F8.3,F5.3),1X,2(F8.3,F5.3),1X,2(F8.3))
   90 CONTINUE
      WRITE(0,*)'1.PLACE SAMPLE BETWEEN COILS AND RUN 2.REPEAT AIR RDGS
     *3.STOP?'
      READ(0,*)NEXT
      GO TO(94,75,810),NEXT
   94 DO 98 NF=1,NFT
      FREK=FREQ(NF)/1000.
      WRITE(LANA,*)'A5B2V1T3F0'
      WRITE(LANA,81)FREK
      AVMM(NF)=0.
      SSMAG(NF)=0.
      AVPM(NF)=0.
      SSPHA(NF)=0.
C
      DO 99 II=1,NMES
      WRITE(LANA,*)'EX'
      READ(LANA,86)CHR1,XMAG,CHR2,XPHA
      SMAG(NF)=SMAG(NF)+XMAG/FLOAT(NMES)
      SPHA(NF)=SPHA(NF)+XPHA/FLOAT(NMES)
C     WRITE(0,87)SMAG(NF),SPHA(NF),XMAG,XPHA
      AVMM(NF)=AVMM(NF)+XMAG
      SSMAG(NF)=SSMAG(NF)+XMAG*XMAG
      AVPM(NF)=AVPM(NF)+XPHA
      SSPHA(NF)=SSPHA(NF)+XPHA*XPHA
   99 CONTINUE
      AVMM(NF)=AVMM(NF)/FLOAT(NMES)
      SDMM(NF)=SQRT(ABS(SSMAG(NF)-AVMM(NF)*AVMM(NF)*FLOAT(NMES))
     *               /FLOAT(NMES-1))
      AVPM(NF)=AVPM(NF)/FLOAT(NMES)
      SDPM(NF)=SQRT(ABS(SSPHA(NF)-AVPM(NF)*AVPM(NF)*FLOAT(NMES))
     *               /FLOAT(NMES-1))
```

```
      DPHA(NF)=AVPM(NF)-AVPA(NF)
      IF(DPHA(NF).GT.0.)DPHA(NF)=DPHA(NF)-360.
      DMAG(NF)=AVMM(NF)-AVMA(NF)
      WRITE(0,91)FREQ(NF),AVMA(NF),SDMA(NF),AVPA(NF),SDPA(NF),AVMM(NF)
     *,SDMM(NF),AVPM(NF),SDPM(NF),DMAG(NF),DPHA(NF)
      WRITE(LOU,91)FREQ(NF),AVMA(NF),SDMA(NF),AVPA(NF),SDPA(NF),AVMM(NF)
     *,SDMM(NF),AVPM(NF),SDPM(NF),DMAG(NF),DPHA(NF)
C     CONDUCTIVITY VALUES
   98 CONTINUE
      DO 300 NF=1,NFT
      DCRES=RESISI
      NSTOP=1
   95 RHO(1)=.95*RESISI
      RHO(2)=RESISI
      RHO(3)=1.05*RESISI
      RHO(4)=1.E20
C
C           THE INTEGRATION IS PERFORMED BY THE MIDPOINT METHOD,
C       EVALUATING AT THE CENTER OF THE INTERVAL; FOR X LARGE
C       THE INTEGRAL CONVERGES RAPIDLY,SO LARGER INTERVALS
C       ARE TAKEN.
C           IN THE INTEGRATION TMUT, DRIVER, PICKUP, AIR1, AND
C       AIR2 ARE CALCULATED.
C
      S1 = 0.005
      S2 = 5.0
      B1 = 0.0
      B2 = S2
C
C       INITIALIZE ALL SUMS TO ZERO AND CALCULATE THE VALUES OF
C       WUSR(NP).
C
```

```
      RCON=.509397921*RBAR*RBAR*FREQ(NF)
      DO  100 NP=1,4
          TMUTRE(NP)=0.
          TMUTIM(NP)=0.
          DRIVRE(NP)=0.
          DRIVIM(NP)=0.
          PICKRE(NP)=0.
          PICKIM(NP)=0.
          WUSR(NP)=RCON*PERM(NP)/RHO(NP)
  100 CONTINUE
      AIR1=0.
      AIR2=0.
C
  110 I1 = (B2 - B1)/S1
      X = B1 - S1*0.5
      DO 210 I=1,I1
        X = X + S1
        XX = X*X
        XXXX = XX*XX
C
C       CALCULATION OF THE COIL PART OF THE INTEGRAND. SUBPROGRAM FOR
C       THROUGH TRANSMISSION TYPE COILS.
C
C        SUBROUTINE BESSEL EVALUATES THE INTEGRAL OF
C        THE PRODUCT OF THE BESSEL FUNCTION J1(X) AND ITS
C        ARGUMENT, X.
C
       CALL BESSEL(XJR2,X,R2)
       CALL BESSEL(XJR1,X,R1)
       CALL BESSEL(XJR4,X,R4)
       CALL BESSEL(XJR3,X,R3)
       D21 = XJR2 - XJR1
       D43 = XJR4 - XJR3
```

```
      S3=S1*D21*D43
      S4=S1*D21*D21
      S5=S1*D43*D43
      WD=-1.
      IF(X*L3.GT.40.)GO TO 120
      WD=DEXP(-X*L3)-1.
  120 WP=-1.
      IF(X*L4.GT.40)GO TO 130
      WP=DEXP(-X*L4)-1.
C
C     UPDATE OF AIR VALUES
C
  130 AIR1=AIR1+(S4+S4)*(X*L3+WD)
      AIR2=AIR2+(S5+S5)*(X*L4+WP)
C
C       BYPASS THE UPDATE OF MUTUAL INDUCTANCE QUANTITIES
C       FOR LARGE X.
C
      IF (X .GT. 40.0) GO TO 200
      EX6=DEXP(-X*L6)
      EX5=DEXP(-X*L5)
      TMUT = S3*EX5*EX6*WD*WP
      DVR = S4*EX6*EX6*WD*WD
      PIC = S5*EX5*EX5*WP*WP
C     LOOP OVER ALL FOUR SETS OF CONDUCTIVITIES
C
      DO 155 NP=1,4
      BR=.70710678*SQRT(XX+SQRT(XXXX+WUSR(NP)*WUSR(NP)))/PERM(NP)
      BI=.5*WUSR(NP)/(BR*PERM(NP)*PERM(NP))
      XTH=THICKN*PERM(NP)
      XPON=DEXP(-BR*XTH)
      COSF=DCOS(-BI*XTH)*XPON
      SINF=DSIN(-BI*XTH)*XPON
      COSF2=COSF*COSF-SINF*SINF
```

```
      SINF2=2*SINF*COSF
      DRL=(X+BR)*(X+BR)-BI*BI+(BI*BI-(X-BR)*(X-BR))*COSF2-2*(X-BR)*BI*
     1SINF2
      DIM=2*(X+BR)*BI+2*(X-BR)*BI*COSF2+(BI*BI-(X-BR)*(X-BR))*SINF2
      ZNR=(XX-BR*BR+BI*BI)*(1.-COSF2)-2*BR*BI*SINF2
      ZNI=(BR*BR-XX-BI*BI)*SINF2-2*BR*BI*(1-COSF2)
      ZMNR=4.*X*(BR*COSF-BI*SINF)
      ZMNI=4.*X*(BI*COSF+BR*SINF)
      Q4=DIM/DRL
      Q3=1./(DRL+DIM*Q4)
      Q4=-Q3*Q4
      DRIVRE(NP)=DRIVRE(NP)+DVR*(ZNR*Q3-ZNI*Q4)
      DRIVIM(NP)=DRIVIM(NP)+DVR*(ZNR*Q4+ZNI*Q3)
      PICKRE(NP)=PICKRE(NP)+PIC*(ZNR*Q3-ZNI*Q4)
      PICKIM(NP)=PICKIM(NP)+PIC*(ZNR*Q4+ZNI*Q3)
      TMUTRE(NP)=TMUTRE(NP)+TMUT*(ZMNR*Q3-ZMNI*Q4)
      TMUTIM(NP)=TMUTIM(NP)+TMUT*(ZMNR*Q4+ZMNI*Q3)
  155 CONTINUE
  200 CONTINUE
  210 CONTINUE
      B1 = B2
      B2 = B2 + S2
      S1 = 0.05
      IF (X .LT. 9.) GO TO 110
      S1 = 0.1
      IF (X .LT. 29.) GO TO 110
      S1 = 0.2
      IF (X .LT. 39.) GO TO 110
      S1 = 0.5
      IF (X .LT. 199.) GO TO 110
C
C     THE NORMALIZED MAGNITUDE AND PHASE ARE CALCULATED
C
```

```
      T1=TNDR/((R2-R1)*L3)
      T2=TNPU/((R4-R3)*L4)
      W=TWOPI*FREQ(NF)
      CALL RFLCKT(TMAG,PHASE,RAD,TMUTRE,TMUTIM,DRIVRE,DRIVIM
     1,PICKRE,PICKIM,GAIN,RBAR,V0,RDCDR,RDCPU,R0,R9,CAPDR,CAPPU
     2,T1,T2,W,AIR1,AIR2)
      ZLDR=1.0027581E-7*RBAR*T1*T1*AIR1
      ZLPU=1.0027581E-7*RBAR*T2*T2*AIR2
C     THE CIRCUIT VALUES ARE PRINTED OUT
C
      WLCDR=1./SQRT(ZLDR*CAPDR)
      WLCPU=1./SQRT(ZLPU*CAPPU)
      IF(NSTOP.NE.0.AND.NF.EQ.1)WRITE(LOU,150)
      IF(NSTOP.NE.0.AND.NF.EQ.1)WRITE(LOU,160)R0,RDCDR,CAPDR,ZLDR,WLCDR,V0
      IF(NSTOP.NE.0.AND.NF.EQ.1)WRITE(LOU,170)R9,RDCPU,CAPPU,ZLPU,WLCPU
C     IF(NSTOP.NE.0.AND.NF.EQ.1)WRITE(LOU,85)
  150 FORMAT('            SER/SHT RES COIL DC RES  SHUNT CAP. COIL INDUCT.
     1  RES.FREQ  DRIV VOLT')
  160 FORMAT(' DVR CKT ',2(F12.3),3(1PE12.4),0PF11.4)
  170 FORMAT(' PICK CKT',2(F12.3),3(1PE12.4),0PF12.4)
C
C     THE NORMALIZED MAGNITUDES AND PHASES ARE PRINTED OUT.
C
      IF(NSTOP.NE.0)WRITE(LOU,180)
      WRITE(0,180)
  180 FORMAT(' NP        RHO      PHASE    MAGNITUDE')
      DO 250 NP=1,3
      TMAG(NP)=20.*DLOG10(TMAG(NP)/TMAG(4))
      PHASE(NP)=PHASE(NP)-PHASE(4)
      IF(PHASE(NP).GT.0.)PHASE(NP)=PHASE(NP)-360.
      IF(NSTOP.EQ.1)WRITE(LOU,240) NP,RHO(NP),PHASE(NP),TMAG(NP)
C     WRITE(LOU,240) NP,RHO(NP),PHASE(NP),TMAG(NP)
      WRITE(0,240) NP,RHO(NP),PHASE(NP),TMAG(NP)
  240 FORMAT(1X,I2,F11.4,F10.3,F10.5)
  250 CONTINUE
```

```
C
      IF(NSTOP.NE.0)WRITE(LOU,260)NP,PHASE(4),TMAG(4)
  260 FORMAT(1X,I2,4X,'AIR',3X,F10.3,F10.5)
C
      IF(NSTOP.EQ.2)GO TO 800
c     RESISI=RESISI+(DPHA(NF)-PHASE(2))*(RHO(3)-RHO(1))/(PHASE(3)-PHASE(1))
C     CALCULATION OF RESISTIVITY BY FITTING TO POLYNOMIAL EQUATION OF:
C     RESISI= C1/PHASE +C1/(PHASE**2)+C3/(PHASE**3)
      DO 265 J=1,3
      R(J,1)=1/(PHASE(J))
      R(J,2)=R(J,1)*R(J,1)
      R(J,3)=R(J,1)*R(J,2)
  265 CONTINUE
      D=R(1,1)*R(2,2)*R(3,3)-R(1,1)*R(3,2)*R(2,3)-R(2,1)*R(1,2)*R(3,3)+
     *R(2,1)*R(3,2)*R(1,3)+R(3,1)*R(1,2)*R(2,3)-R(3,1)*R(2,2)*R(1,3)
      C1=(RHO(1)*R(2,2)*R(3,3)-RHO(1)*R(3,2)*R(2,3)-RHO(2)*R(1,2)*R(3,3)+
     *RHO(2)*R(3,2)*R(1,3)+RHO(3)*R(1,2)*R(2,3)-RHO(3)*R(2,2)*R(1,3))/D
      C2=(R(1,1)*RHO(2)*R(3,3)-R(1,1)*RHO(3)*R(2,3)-R(2,1)*RHO(1)*R(3,3)+
     *R(2,1)*RHO(3)*R(1,3)+R(3,1)*RHO(1)*R(2,3)-R(3,1)*RHO(2)*R(1,3))/D
      C3=(R(1,1)*R(2,2)*RHO(3)-R(1,1)*R(3,2)*RHO(2)-R(2,1)*R(1,2)*RHO(3)+
     *R(2,1)*R(3,2)*RHO(1)+R(3,1)*R(1,2)*RHO(2)-R(3,1)*R(2,2)*RHO(1))/D
      RESISI= (C1+(C2+C3/DPHA(NF))/DPHA(NF))/DPHA(NF)
C
      NSTOP=0
      IF(ABS(DPHA(NF)-PHASE(2)).GT.0.0025)GO TO 95
      PERCT=100.*(RESISI-DCRES)/DCRES
C     TMAG(2)=20.*DLOG10(TMAG(2))
      SDVR(NF)=SQRT(ABS(SDPA(NF)*SDPA(NF)+SDPM(NF)*SDPM(NF))
     *         /FLOAT(NMES))*(RHO(3)-RHO(1))/(PHASE(3)-PHASE(1))
      WRITE(LOU,270)RESISI,SDVR(NF),DCRES,PERCT,FREQ(NF)
  270 FORMAT(' RESIS,AC=',F10.4,'+-',F7.4,'SDV  ASS=',F10.4,F8.3,
     1' %DIF  FREQ=',1PE12.4)
      RESIS=RESISI
      WRITE(LOU,275)TMAG(2),DMAG(NF),PHASE(2),DPHA(NF)
```

```
  275 FORMAT(' MAG=',F9.4,'CAL',F9.4,'DB MES  PHASE=',F9.3,'CAL',F9.3,
     *'DEG MES')
      IF(TC.EQ.0)GO TO 800
      RESIS=RESIS+TC*(20.-TEMP)
      WRITE(LOU,280)RESIS,TEMP,TC
  280 FORMAT(' RESIS AT 20 C=',F10.4,' MEASURED AT',F6.2,' C   CORR FAC=
     *',F7.4)
  800 WRITE(LOU,85)
  300 CONTINUE
  810 WRITE(LOU,85)
      STOP 'JOB'
      END
C
      SUBROUTINE RFLCKT(TMAG,PHASE,RAD,TMUTRE,TMUTIM,DRIVRE,DRIVIM
     1,PICKRE,PICKIM,GAIN,RBAR,VO,RDCDR,RDCPU,R0,R9,CAPDR,CAPPU
     2,T1,T2,W,AIR1,AIR2)
C
      SUBROUTINE BESSEL(XJ1,X,R)
```

A sample run of the program CONDTF follows.

```
CONDTF   TIME 11: 7:52  DATE 10/ 9/86
 ACTUAL DIMENSIONS(IN.)
 COIL FACE TO FACE  .20500 SAMPLE THICK  .20180 GAP  .00320
 MEAN RADIUS        .09900 INCHES
   COIL     INN. RAD   OUT. RAD    LENGTH    TURNS EA   0 L.O./M L.O
 DRIVER        .0750      .1230      .0500      112.0         .0040
 PICK-UP       .0750      .1230      .0500      112.0         .0072
          |       AIR READINGS        ||   MATERIAL READINGS    ||  DIFFERENCE  |
  FREQ(HZ)  MAG:AVG  SDV PHA:AVG  SDV  MAG:AVG  SDV PHA:AVG  SDV      MAG     PHA
 5.000E+04  -52.505 .011 -90.520 .042  -52.947 .010-107.568 .050    -.441 -17.048
 1.000E+05  -46.468 .004 -90.943 .023  -47.686 .007-121.555 .042   -1.217 -30.612
 2.000E+05  -40.364 .005 -91.891 .015  -43.189 .004-143.850 .016   -2.825 -51.959
 5.000E+05  -31.820 .000 -94.737 .005  -39.050 .000 167.311 .013   -7.230 -97.952
          SER/SHT RES COIL DC RES  SHUNT CAP. COIL INDUCT.  RES.FREQ  DRIV VOLT
  DVR CKT    3184.100      4.360 1.2200E-10  6.4426E-05  1.1279E+07     1.0000
```

| NP | RHO | PHASE | MAGNITUDE |
|---|---|---|---|
| 1 | 1128.0300 | -17.846 | -.52946 |
| 2 | 1187.4000 | -17.066 | -.49168 |
| 3 | 1246.7700 | -16.353 | -.45796 |
| 4 | AIR | 89.618 | .00024 |

RESIS,AC= 1188.8303+- 1.1708SDV ASS= 1187.4000 .120 %DIF
*FREQ= 5.0000E+04
MAG= -.4908CAL -.4410DB MES PHASE= -17.048CAL -17.048DEG MES
RESIS AT 20 C= 1188.5063 MEASURED AT 23.00 C CORR FAC= .1080

| NP | RHO | PHASE | MAGNITUDE |
|---|---|---|---|
| 1 | 1129.3888 | -31.771 | -1.34489 |
| 2 | 1188.8303 | -30.494 | -1.26076 |
| 3 | 1248.2717 | -29.319 | -1.18487 |
| 4 | AIR | 89.235 | .00048 |

RESIS,AC= 1183.1084+- .5154SDV ASS= 1188.8303 -.481 %DIF
*FREQ= 1.0000E+05
MAG= -1.2685CAL -1.2175DB MES PHASE= -30.612CAL -30.612DEG MES
RESIS AT 20 C= 1182.7844 MEASURED AT 23.00 C CORR FAC= .1080

| NP | RHO | PHASE | MAGNITUDE |
|---|---|---|---|
| 1 | 1123.9530 | -54.106 | -3.04413 |
| 2 | 1183.1084 | -52.117 | -2.87889 |
| 3 | 1242.2638 | -50.282 | -2.72856 |
| 4 | AIR | 88.456 | .00097 |

RESIS,AC= 1188.0294+- .1497SDV ASS= 1183.1084 .416 %DIF
*FREQ= 2.0000E+05
MAG= -2.8658CAL -2.8250DB MES PHASE= -51.959CAL -51.959DEG MES
RESIS AT 20 C= 1187.7054 MEASURED AT 23.00 C CORR FAC= .1080

| NP | RHO | PHASE | MAGNITUDE |
|---|---|---|---|
| 1 | 1128.6279 | -101.793 | -7.62970 |
| 2 | 1188.0294 | -98.438 | -7.28013 |
| 3 | 1247.4308 | -95.333 | -6.95986 |
| 4 | AIR | 85.873 | .00274 |

RESIS,AC= 1197.0251+- .0591SDV ASS= 1188.0294 .757 %DIF FREQ=
*5.0000E+05
MAG= -7.2298CAL -7.2300DB MES PHASE= -97.952CAL -97.952DEG MES
RESIS AT 20 C= 1196.7011 MEASURED AT 23.00 C CORR FAC= .1080

## MULTIPLE-PROPERTY PROGRAMS

The next three programs are designed to be run as a set and to allow multiple-property/multiple-frequency eddy-current problems[2,3] to be run with a minimum of effort. The multiple-property type of eddy-current problem is the most common and the most difficult to solve. Eddy-current readings are affected by a large number of different test property variations, such as wall thickness, conductivity, permeability, lift-off, and defects, in addition to geometry effects such as multiple conductive regions. The large number of items that affect eddy-current readings allows eddy currents to be used for many different problems but also limits the applications to those where the signals from unwanted properties do not overwhelm the signals from the properties of interest. The use of multiple-property solutions allows a tremendous increase in the signal-to-noise ratio and allows the accurate eddy-current testing of many problems that were normally considered impossible. Ten multiple-property problems are estimated for every single-property eddy-current problem. These programs are very general and can be applied to a very wide range of eddy-current problems, partly because of the wide frequency range and versatility of the Impedance Analyzer. The first program, RDGANA, will make magnitude and phase readings at different frequencies for a given set of properties. The second program, FITANA, will take these readings and perform least-squares fits of the readings to the properties that produced the readings, giving an "inversion" equation. By using the inversion equation, the third program will perform back plots (plots of the properties computed by multiplying the instrument readings by the least-squares coefficients) of the standards or other samples.

### PROGRAM RDGANA

The first of the three programs, RDGANA, reads and records the magnitude and phase readings produced by a set of standards with different property values. The values of the different properties should span the range of variations in the different properties that are expected in the actual test. For example, if conductivity, lift-off, and wall thickness are to vary in an eddy-current inspection, the standards may include 3 different thickness values, 3 different conductivities, and 3 different lift-off values for a total of 27 different property combinations. If an accurate measurement of a particular property, such as thickness, is desired, then the thickness of the standards must be accurately measured. If the elimination of a property's effects is desired, then the actual value is not important as long as it spans the desired range. For instance, the lift-off standards in the above test need not be extremely accurate if the only goal is to eliminate their effect on the thickness measurements. The selection, construction, and use of standards is often the most difficult part of the multiple-property problem. The program will make multiple-frequency readings by using either reflection or through-transmission type coils. Impedance measurements can be made on coils with only a small change in format statements and a change in the instructions to the analyzer. The probe may be set in the Modulynx Mechanical Scanner or placed manually on the sample.

A file, with FORTRAN unit number LID, contains the information on the probe and cables, the frequencies to be used, the property values of the standard, and the position data for each property value if the mechanical scanner is to be used. The variable NAXIS determines whether the readings will be read from the Modulynx Scanner or manual placement will be used. A value of zero for the number of axes will allow manual placement of the standards. A maximum of three axes can be used with the Modulynx Scanner. The subroutine POSIT moves the scanner to the desired locations. The subroutine ANARDG takes the multifrequency readings in this program and most of the others in this report. These subroutines are listed in the Appendix.

## Coil and Equipment Setup

The coil setup can be for reflection, through-transmission, or absolute-impedance measurements. The reflection coil connections to the impedance analyzer are shown in Fig. 6, and the through-transmission connections are shown in Fig. 4. Impedance measurements can also be made using a single absolute coil (Fig. 2). The Modulynx Position Controller, along with the Impedance Analyzer, can be connected to the IEEE-488 bus (Fig. 7), or the probe and samples can be positioned manually (Fig. 8). The address of the position controller is 6 on the bus, and the address of the impedance analyzer is 3. The Modulynx Position Controller can be used to control the position of the *xyz* scanner (Fig. 9) or any of the several tubing scanners, which are not pictured.

## Input Data

Input data for program RDGANA are contained in the program and also in an input file. The data arrays in RDGANA must be large enough to hold the data as they are taken and stored. To show the array sizes needed, a set of sample dimension statements is given before the actual dimension statements. Some of the input data that are not described in the program are explained below.

PRONAM – names of different properties being measured
TMAGS(NFT),PHASES(NFT) – intermediate variables from subroutine
AXISNM – array of axis names
FNAME – general file name
XNEW – current position coordinate array
FREQ – frequency array
POST – location data for positioners read from device LID
XMAG – array of magnitudes, read from impedance analyzer
XPHA – array of phases, read from impedance analyzer
VOUT – array of output voltage levels
PROP – property value array
TMAG – magnitude summation array
PHASE – phase summation array
NOLD – number of steps taken to get to current position

ORNL-DWG 86-15951

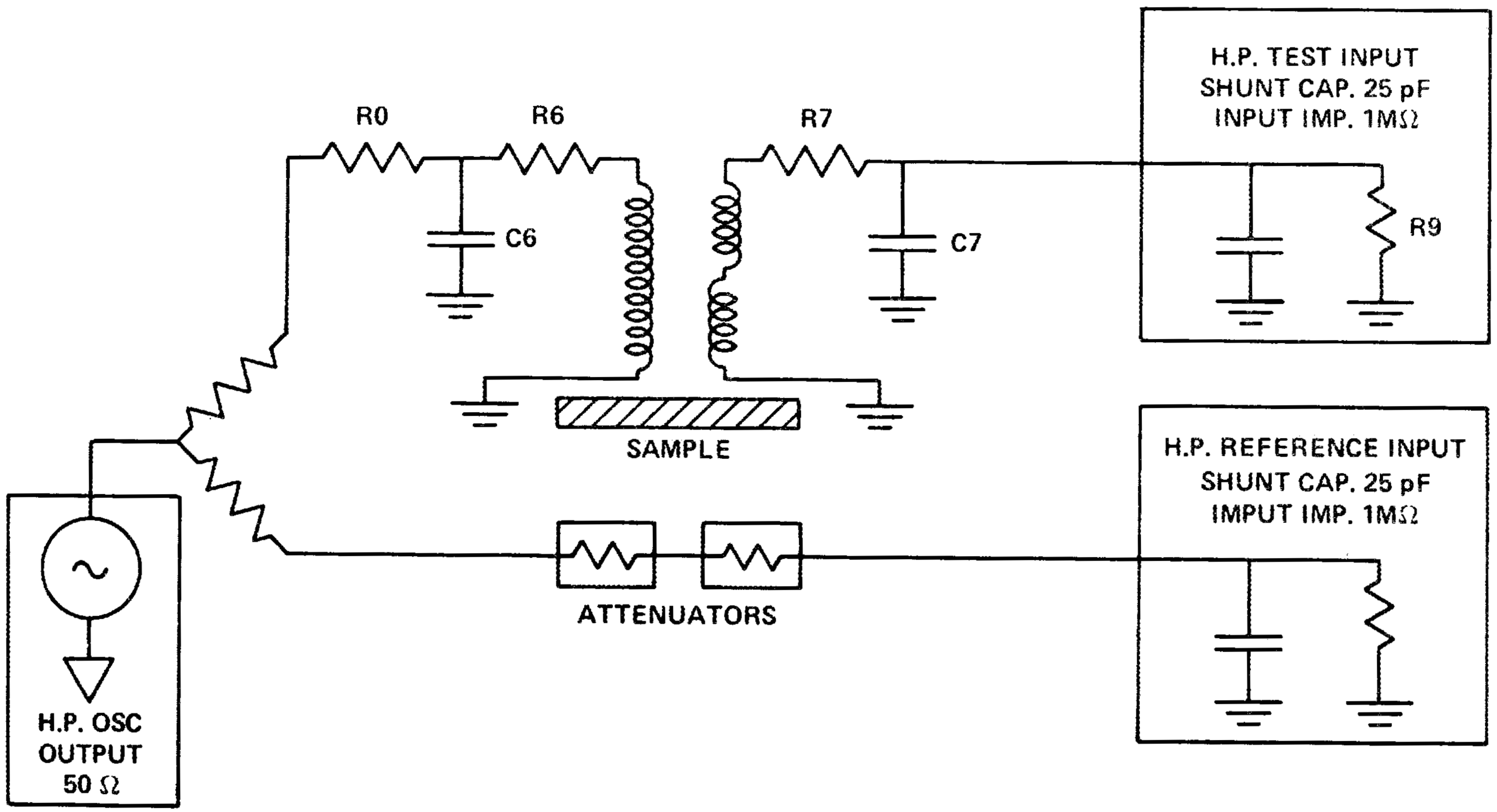



Fig. 6. Impedance analyzer connections for reflection-probe measurements.

ORNL-DWG 86-15953

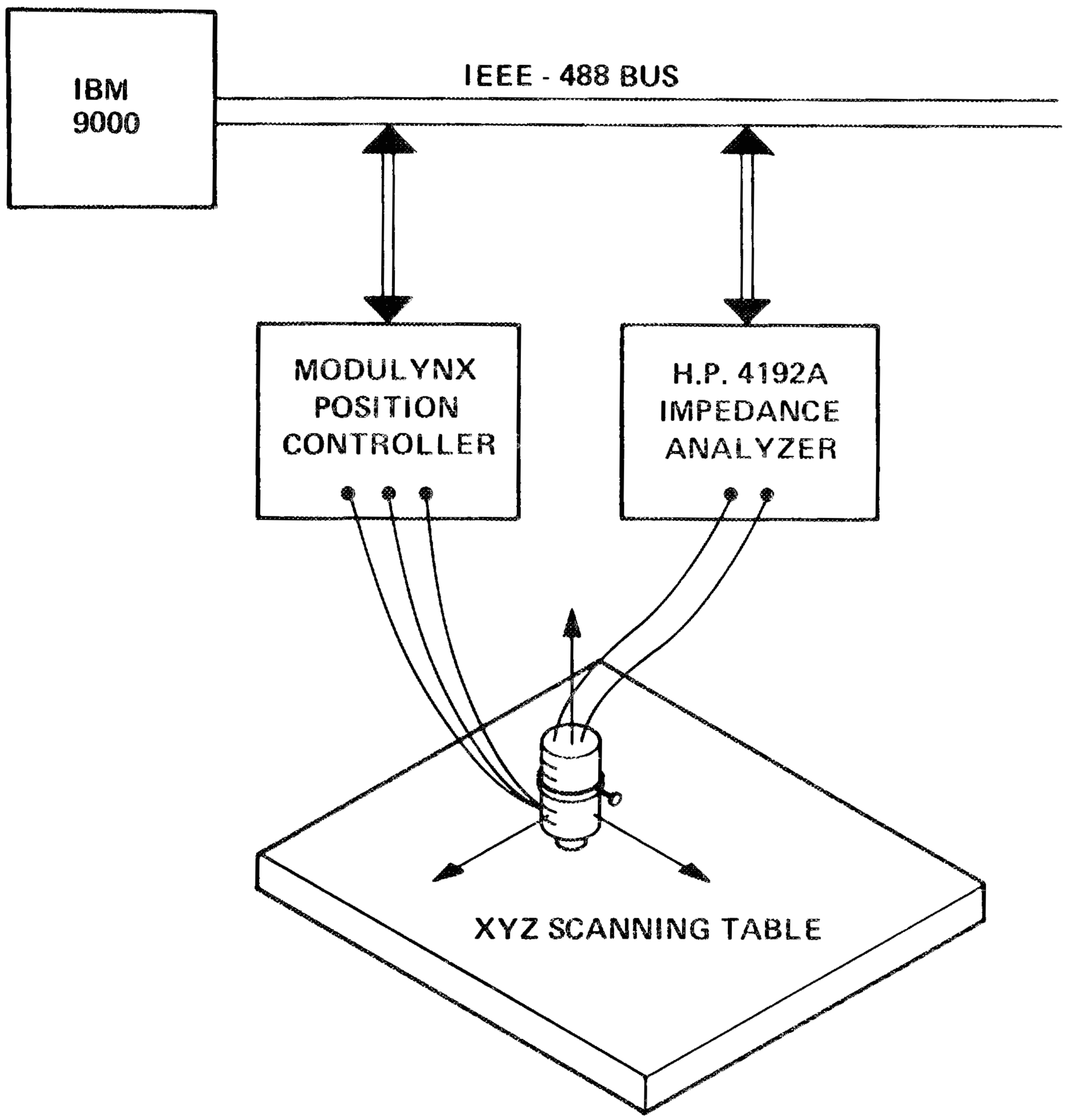



Fig. 7. Connections of the IBM System 9000, the impedance analyzer, and the Modulynx position controller to the IEEE-488 bus.

ORNL-DWG 86-15955

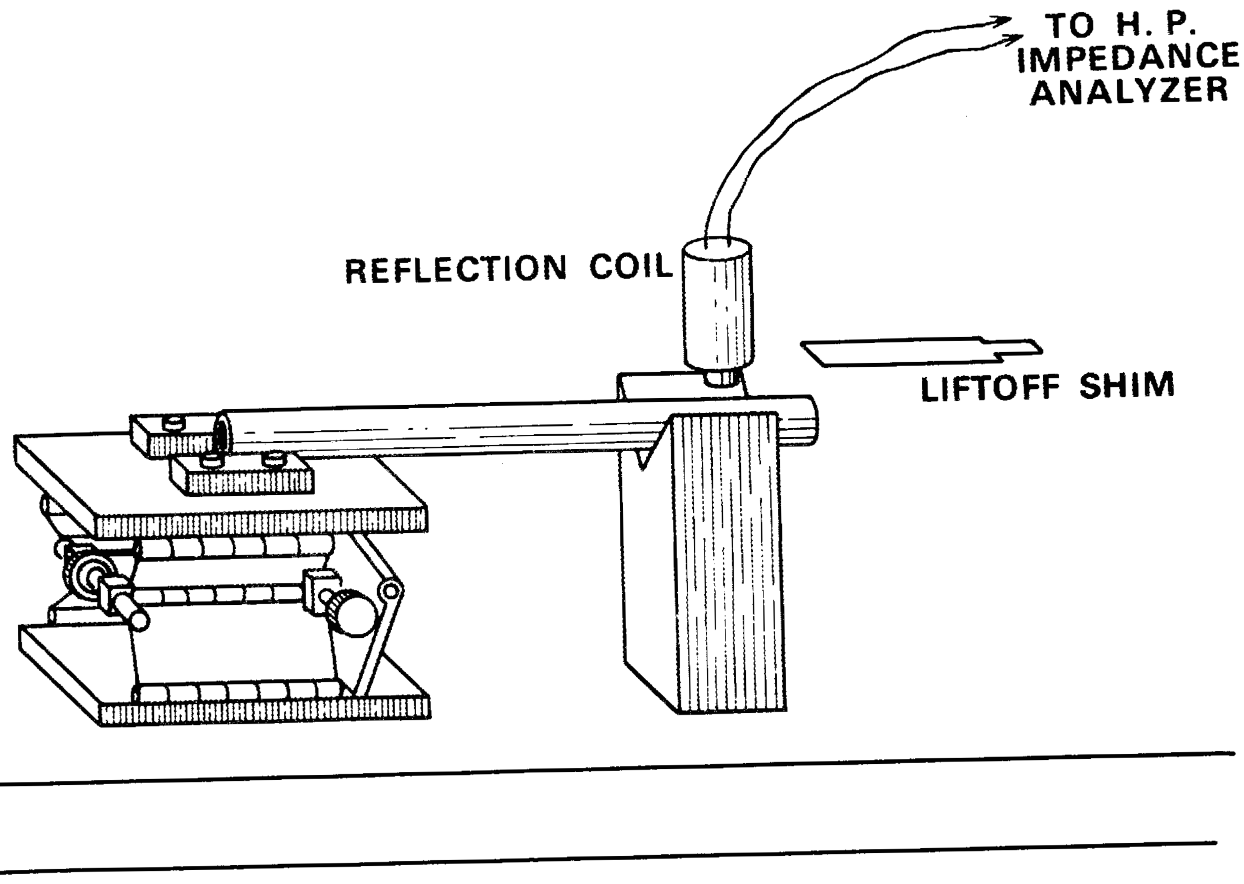



Fig. 8. Manual positioning of a reflection probe on tubing samples.

ORNL-DWG 86-15947

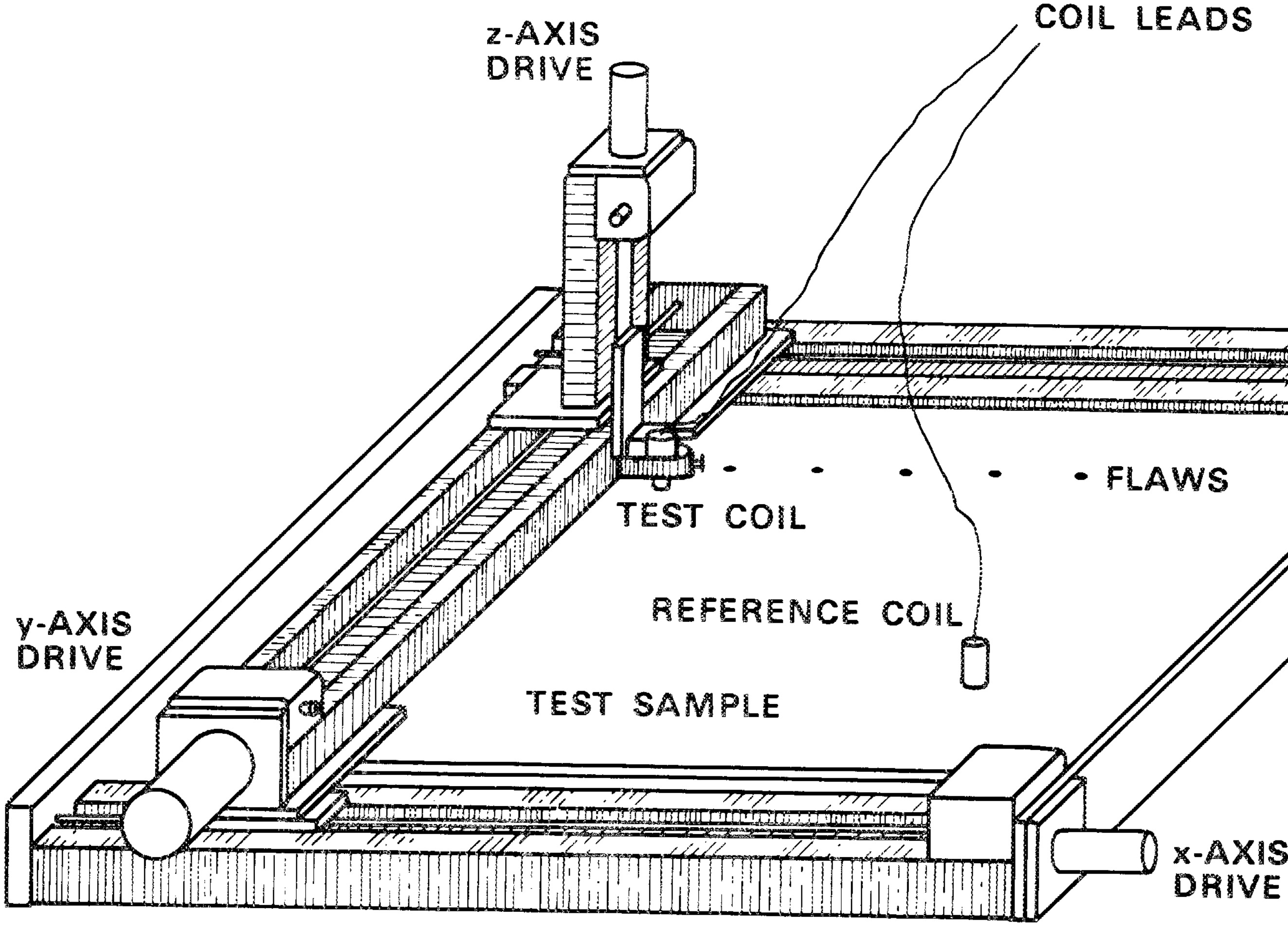



Fig. 9. An *xyz* computer-controlled eddy-current scanner.

NCL - input data file for reflection coil

MBUS - logical output number for Modulynx Axis Position Controller

LOU - logical output device number for printer

LANA - device number for impedance analyzer

LOD - logical output device number for data output file

INDAT/'NRCDIN'/ - names the input data file

OTDAT/'NRCDOT'/ - names the output data file, input for FITANA

The data file, INDAT, contains most of the data used in the program. The first section is a description of the experimental apparatus and is used only for documentation. It can be removed or modified as desired without affecting the program results. The next line (10) contains the number of frequencies (NFT), the number of total sets of properties (NPT), and the number of different types of properties that vary (NPROPM). The next line (11) contains the values of the frequencies, and line 12 contains the output voltage levels for each of the frequencies. Line 13 contains the name for the NPROPM properties that vary and then the names of the axes, if any. Starting with line 14 are the property value and position for the property NP for the next NPT lines. A sample of an input data file, named CONDIN.DAT, is listed after the listing of RDGANA.

<u>Subroutines</u>

This program contains many systems subroutines. They have been referenced previously and are as follows.

1. _GETTIME
2. _STIMEOUT
3. _SETOS
4. _SETOPTIONS
5. _GETKEY

Two subroutines, POSIT and ANARDG, are listed in the Appendix for this program.

Main Program

A listing of the program RDGANA, along with a data file, NRCDIN, follows.

```
      PROGRAM RDGANA
C     (14 JULY 1986)
C
C      THIS PROGRAM READS MAGNITUDE AND PHASE DATA AT DIFFERENT
C     FREQUENCIES,USING THE HEWLETT PACKARD 4192A IMPEDANCE ANALYZER AND
C     THE MODULYNX MECHANICAL SCANNER.IT IS DESIGNED FOR LARGE ARRAYS OF
C     READINGS.THE MAGNITUDE AND PHASE DATA IS STORED ALONG WITH EACH SET
C     OF PROPERTIES.THE PROPERTY AND POSITIONING DATA IS READ FROM DEVICE
C     LID.
C     LI=LOGICAL INPUT TERMINAL(FROM OPERATOR)
C     LID=LOGICAL INPUT DEVICE(MAY BE  A DISK FILE OR TERMINAL)
C     LOT=LOGICAL OUTPUT OPERATOR TERMINAL.
C     LPT=LINE PRINTER(OUTPUT)TERMINAL
C     MSET=NUMBER OF SETS OF READINGS THAT WILL BE TAKEN
C     NCH=NUMBER OF CHANNELS TO BE READ,USUALLY 2*NFT
C     NFT=NUMBER OF FREQUENCIES
C     NF=FREQUENCY INDEX
C     NAXIS=NUMBER OF AXES THAT ARE POSITIONED
C     NP=PROPERTY INDEX
C     NPROPM=MAX NUMBER OF PROPERTIES THAT MAY VARY
C     NPT=TOTAL NUMBER OF SETS OF PROPERTY VALUES FOR A COMPLETE
C         SET OF READINGS=NUMBER SAMPLES+NUMBER DEFECTS
C
      CHARACTER*6 NPROBE,NCABLE,INSTNO,COIL
      CHARACTER INDAT*8,OTDAT*8,FNAME*12,BLANKS*4,PRONAM(7)*4,AXISNM(3)*4
C     DIMENSION FREQ(NFT),VOUT(NFT),TMAGS(NFT),PHASES(NFT)
C     DIMENSION TMAG(NPT,NFT),PHASE(NPT,NFT),PROP(NPT,NPROPM)
C     DIMENSION XMAG(NFT),XPHA(NFT)
C     DIMENSION XNEW(NAXIS),NOLD(NAXIS),POST(NPT,NAXIS)
C
```

```
      DIMENSION FREQ(4),VOUT(4),TMAGS(4),PHASES(4)
      DIMENSION TMAG(27,4),PHASE(27,4),PROP(27,4)
      DIMENSION XMAG(4),XPHA(4)
      DIMENSION XNEW(3),NOLD(3),POST(27,3)
C
      DATA LI/0/,LOT/0/,LOU/6/,NCH/6/,LID/9/,NCL/28/,LOD/37/,MSET/3/
      DATA LANA/43/,MBUS/46/,NOLD/3*0/,NAXIS/0/
      DATA BLANKS/'    '/,INDAT/'CONDIN'/,OTDAT/'CONDOT'/
      DATA NPT/27/,NPROPM/2/
C
C PRINT TITLE AND DATE
      OPEN(LOU,FILE='#PR')
      CALL_GETTIME(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
      WRITE(LOU,2)IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT(' RDGANA    TIME ',I2,':',I2,':',I2,'  DATE ',I2,'/',I2,'/',I2)
C
   30 FORMAT(1X)
C
C     OPEN FILE FOR INSTRUMENT DESCRIPTION INPUT - ASSUMED LOCATION ON
C     DEFAULT DISK
   40 FORMAT(A8)
      FNAME=INDAT//'.DAT'
      OPEN(LID,FILE=FNAME,STATUS ='OLD')
      WRITE(0,*)'INPUT FILE OPENED'
C     OPEN FILE FOR OUTPUT DATA STORAGE - ASSUMED LOCATION ON DEFAULT DISK
      FNAME=OTDAT//'.DAT'
      OPEN(LOD,FILE=FNAME,STATUS ='NEW')
      WRITE(0,*)'OUTPUT FILE OPENED'
      OPEN(NCL,FILE='REF.DAT',STATUS ='OLD')
      WRITE(0,*)'COIL FILE OPENED'
C
C     HEWLETT PACKARD 4192A LF IMPEDANCE ANALYZER ADDR=3 ON GPIB BUSS
C     ADDR=C? ON IBM BUSS DRIVER, FORTRAN LOGICAL UNIT = 43
      OPEN(LANA,FILE='#BUSC?')
      WRITE(0,*)'IMPEDANCE ANALYZER  FILE OPENED'
```

```
C       MODULYNX AXIS POSITION CONTROLLER ADDR=6 ON GPIB BUSS,ADDR=F? ON
C       IBM BUSS DRIVER,FORTRAN LOGICAL UNIT = 46
        OPEN(MBUS,FILE='#BUSF?')
        WRITE(0,*)'MOTOR FILE OPENED'
        CALL_STIMEOUT(MBUS,1000)
        CALL_SETEOS(MBUS,$00,$0A)
        CALL_SETOPTIONS(MBUS,0,0,0,0,0,1,0)
C       WRITE(MBUS,*)'XA2000B160H2000'
        WRITE(MBUS,*)'XD YD ZD'
        WRITE(MBUS,*)'<CAH>'
        WRITE(MBUS,*)'XE YE ZE'
C
C INPUT DESCRIPTION OF EXPERIMENTAL APPARATUS
C
        WRITE(LOT,60)
60      FORMAT(' PROBE #:  ')
        READ(LID,70)NPROBE
        WRITE(LOT,70)NPROBE
70      FORMAT(A6)
        WRITE(LOT,80)
80      FORMAT(' SERIAL #:  ')
        READ(LID,*)NSER
        WRITE(LOT,*)NSER
        WRITE(LOT,90)
90      FORMAT(' DRIVER SERIES RESISTANCE:  ')
        READ(LID,*)R0, RATT
        WRITE(LOT,*)R0, RATT
        WRITE(LOT,100)
100     FORMAT(' DRIVER SHUNT CAP:  ')
        READ(LID,*)CAPDR
        WRITE(LOT,*)CAPDR
        WRITE(LOT,110)
```

```
110   FORMAT(' PICK-UP SHUNT RESISTANCE:  ')
      READ(LID,*)R9
      WRITE(LOT,*)R9
      WRITE(LOT,120)
120   FORMAT(' PICK-UP SHUNT CAP:  ')
      READ(LID,*)CAPPU
      WRITE(LOT,*)CAPPU
      WRITE(LOT,130)
130   FORMAT(' CABLE I.D. #:  ')
      READ(LID,70)NCABLE
      WRITE(LOT,70)NCABLE
      WRITE(LOT,140)
140   FORMAT(' LENGTH OF CABLE:  ')
      READ(LID,*)CABLEL
      WRITE(LOT,*)CABLEL
      WRITE(LOT,150)
150   FORMAT(' CAPACITANCE OF CABLE:  ')
      READ(LID,*)CCABLE
      WRITE(LOT,*)CCABLE
C
C INPUT FREQUENCY VALUES
C
      WRITE(LOT,190)
190   FORMAT(' NUMBER OF FREQUENCIES,NUMBER OF PROP SETS,NUMBER OF
     *PROPERTIES:')
      READ(LID,*)NFT,NPT,NPROPM
      WRITE(LOT,*)NFT,NPT,NPROPM
      WRITE(LOT,200)
200   FORMAT(' INPUT THE VALUE OF EACH FREQ SEPARATED BY A SPACE:',/)
      READ(LID,*)(FREQ(NF),NF=1,NFT)
      WRITE(LOT,*)(FREQ(NF),NF=1,NFT)
      WRITE(LOT,210)
210   FORMAT(' INPUT THE OUTPUT VOLTAGE LEVELS FOR EACH FREQUENCY:',/)
      READ(LID,*)(VOUT(NF),NF=1,NFT)
      WRITE(LOT,*)(VOUT(NF),NF=1,NFT)
```

```
      READ(LID,270)(PRONAM(NPR),NPR=1,NPROPM),(AXISNM(NAX),NAX=1,NAXIS)
      WRITE(LOT,280)(PRONAM(NPR),NPR=1,NPROPM),(AXISNM(NAX),NAX=1,NAXIS)
270   FORMAT(A4,2X,A4,2X,A4,2X,A4,2X,A4,2X,A4,2X,A4,2X,A4,2X,A4,2X,A4)
280   FORMAT(10(3X,A4))
C
C READ PROPERTY VALUES AND LOCATION DATA FOR THE POSITIONERS FROM LID
C
      DO 300 NP=1,NPT
      READ(LID,*)(PROP(NP,NPR),NPR=1,NPROPM),(POST(NP,NAX),NAX=1,NAXIS)
      WRITE(LOT,290)(PROP(NP,NPR),NPR=1,NPROPM),(POST(NP,NAX),NAX=1,NAXIS)
290   FORMAT(10(F9.3))
300   CONTINUE
C
       WRITE(MBUS,*)'XD YD ZD'
C
C WRITE SET UP VALUES FOR IMPEDANCE ANALYZER,FIRST FREQUENCY AND VOLTAGE
C OUT
C

      WRITE(LANA,*)'A5B2H1V1T3F0'
      WRITE(LANA,81)VOUT(1),FREQ(1)
   81 FORMAT(1X,'OL',F5.3,'ENFR',F9.3,'EN')
C
C TAKE READINGS ON NOMINAL STANDARD FOR VOLSTD(I) READINGS
C
      XNEW(1)=0.0
      XNEW(2)=0.0
      XNEW(3)=0.0
      CALL POSIT(XNEW,NOLD,NAXIS,MBUS)
       WRITE(MBUS,*)'XE YE ZE'
773   WRITE(LOT,775)
775   FORMAT(' TAKE STD RDGS ----- HIT ANY KEY TO STOP ')
      WRITE(LOT,920)(II,II, II=1,NFT)
      WRITE(LOT,30)
```

```
780    CALL ANARDG(TMAGS,PHASES,FREQ,VOUT,LANA,NFT)
       WRITE(LOT,950)((TMAGS(NF),PHASES(NF)),NF=1,NFT)
       CALL_GETKEY(IKY,IS,KBF1,KBF2,IRR)
       IF(IRR.NE.0)GO TO 780
        WRITE(MBUS,*)'XD YD ZD'
       WRITE(LOT,785)
785    FORMAT(' SET UP SCANNER AND CALIB STANDARD,HIT ANY KEY TO GO')
787    CALL ANARDG(XMAG,XPHA,FREQ,VOUT,LANA,NFT)
       WRITE(LOT,950)((XMAG(NF),XPHA(NF)),NF=1,NFT)
       CALL_GETKEY(IKY,IS,KBF1,KBF2,IRR)
       IF(IRR.NE.0)GO TO 787
C       ZERO ARRAYS THAT WILL CONTAIN SUMS OF THE PHASE & MAG READINGS.
C
       DO 800 NF=1,NFT
       DO 800 NP=1,NPT
       TMAG(NP,NF)=0.
       PHASE(NP,NF)=0.
800    CONTINUE
C
C THIS SECTION TAKES THE ACTUAL PHASE & MAGNITUDE DATA READINGS
C
       DO 1150 MSE=1,MSET
       DO 980 NP=1,NPT
       WRITE(LOT,900)NP,(PROP(NP,NPR),NPR=1,NPROPM),(POST(NP,NAX),NAX=1,
      *NAXIS)
900    FORMAT(I4,10(f9.3))
C
C      POSITION SAMPLES BEFORE TAKING READINGS - USE MANUAL POSITIONING IF
C      NAXIS=0
C
       IF(NAXIS.EQ.0)GO TO 914
       DO 910 NAX=1,NAXIS
       XNEW(NAX)=POST(NP,NAX)
910    CONTINUE
       CALL POSIT(XNEW,NOLD,NAXIS,MBUS)
        WRITE(MBUS,*)'XE YE ZE'
```

```
C     IF(MOD((NP-1),11).NE.0)GO TO 918
 914  WRITE(0,*)'INSERT LIFT-OFF SHIM MANUALLY-HIT ANY KEY WHEN READY'
 915  CALL_GETKEY(IKY,IS,KBF1,KBF2,IRR)
      IF(IRR.NE.0)GO TO 915
918   WRITE(LOT,920)(II,II, II=1,NFT)
920   FORMAT(5('  MAG(',I1,')   PH(',I1,')'))
930   CALL ANARDG(XMAG,XPHA,FREQ,VOUT,LANA,NFT)
      WRITE(MBUS,*)'XD YD ZD'
      WRITE(MBUS,*)'XD YD ZD'
       WRITE(MBUS,*)';'
      WRITE(LOT,950)((XMAG(NF),XPHA(NF)),NF=1,NFT)
950   FORMAT(5(F8.3,F8.2))
      DO 980 NF=1,NFT
      TMAG(NP,NF)=TMAG(NP,NF)+XMAG(NF)
      PHASE(NP,NF)=PHASE(NP,NF)+XPHA(NF)
980   CONTINUE
1150  CONTINUE
C
C     RETURN POSITIONER TO XNEW(II)=0.0 SO THAT POSITIONS CAN BE CHECKED
C     THEN TURN MOTOR CURRENT OFF
C
       WRITE(MBUS,*,ERR=1160)'<CAH>'
       GO TO 1170
 1160  WRITE(0,*)'ERROR IN FINAL HOME POSITION'
 1170  WRITE(MBUS,*,ERR=1180)' XE YE ZE '
       GO TO 1190
 1180  WRITE(0,*)' ERROR IN CURRENT SHUT-DOWN'
C
C CALCULATE AVERAGES OF THE READINGS
C
 1190 DO 1260 NF=1,NFT
      DO 1260 NP=1,NPT
      TMAG(NP,NF)=TMAG(NP,NF)/(FLOAT(MSET))
      PHASE(NP,NF)=PHASE(NP,NF)/(FLOAT(MSET))
```

```
1260  CONTINUE
C
C STORE ALL INFORMATION IN DIRECT ACCESS FILE LOD ON DISK
C
      WRITE(LOD,1270)IHR,IMN,ISE,IMO,IDA,IYR
1270  FORMAT(6(1X,I4))
      WRITE(LOD,1275) NPROBE,NSER,R0,RATT,CAPDR,R9,CAPPU,NCABLE,
     *               CABLEL,CCABLE
1275  FORMAT(A6,1X,I5,5(E13.4),1X,A6,2(E13.4))
      WRITE(LOD,1277) NFT,NPT,NPROPM
1277  FORMAT(3(1X,I5))
      WRITE(LOD,1280)(FREQ(NF),NF=1,NFT)
      WRITE(LOD,1280)(VOUT(NF),NF=1,NFT)
1280  FORMAT(6(E13.4))
      WRITE(LOD,1285)(PRONAM(NPR),NPR=1,NPROPM)
1285  FORMAT(7(1X,A4))
      WRITE(LOD,1290)((TMAGS(NF),PHASES(NF)),NF=1,NFT)
1290  FORMAT(6(1X,F7.3,1X,F7.2))
      DO 1300 NP=1,NPT
      WRITE(LOD,1297)(TMAG(NP,NF),PHASE(NP,NF),NF=1,NFT),(PROP(NP,NPR)
     *,NPR=1,NPROPM)
1297  FORMAT(14(F9.3,1X))
1300  CONTINUE
C
C END OF SECTION WHICH WRITES DIRECT ACCESS FILE
C PRINT SUMMARY OF JOB STATISTICS ON LOU
C
      WRITE(LOU,1350) NPROBE,NSER
1350  FORMAT(' PROBE NO.:',A6,5X,' SERIAL NO.:',I5)
      WRITE(LOU,1360) R0,RATT,CAPDR
1360  FORMAT(' DR SER RES:',F10.1,2X,'REF ATTN',F5.1,'  DRIVER SHUNT CAP.:',
     *        E12.4)
      WRITE(LOU,1370) R9,CAPPU
```

```
1370  FORMAT(' PICK-UP SHUNT RESISTANCE:',F10.1,5X,'PICK-UP SHUNT CAP.:'
     *          ,E12.4)
      WRITE(LOU,1380) NCABLE,CABLEL,CCABLE
1380  FORMAT(' CABLE I.D. NO.:',A6,5X,'LENGTH:',F10.1,5X,'CAP.:',
     *          E12.4)
      WRITE(LOU,1410) (FREQ(NF),NF=1,NFT)
1410  FORMAT('FREQUENCY:',10(1PE12.4,5X))
      WRITE(LOU,1420) (VOUT(NF),NF=1,NFT)
1420  FORMAT('VOLTAGE OUT:',10(1PE12.4,5X))
C
C     READINGS FROM NOMINAL SAMPLE
C
      WRITE(LOU,1745)
1745  FORMAT(' READINGS FROM NOMINAL TUBE SAMPLE')
      WRITE(LOU,1290)((TMAGS(NF),PHASES(NF)),NF=1,NFT)
C
C     PRINT OUT READINGS AND PROPERTIES
C
      WRITE(LOU,1750)(NF,NF,NF=1,NFT),(PRONAM(NPR),NPR=1,NPROPM)
C     THE NEXT FORMAT STATEMENT SHOULD HAVE NFT('  MAG',I1,1X,'
     *PHA',I1,1X)...
1750  FORMAT('PSET',3('  MAG',I1,1X,'  PHA',I1,1X),1X,4(1X,A4,1X),3(1X,A4))
      DO 1800 NP=1,NPT
      WRITE(LOU,1760)NP,(TMAG(NP,NF),PHASE(NP,NF),NF=1,NFT),(PROP(NP,
     *NPR),NPR=1,NPROPM)
C     THE NEXT FORMAT STATEMENT SHOULD HAVE NFT(F7.3,F7.2) ...
1760  FORMAT(I4,3(F7.3,F7.2),1X,2(F10.4))
1800  CONTINUE
      STOP
      END
C
      SUBROUTINE POSIT(XNEW,NOLD,NAXIS,MBUS)
C
      SUBROUTINE ANARDG(XMAG,XPHA,FREQ,VOUT,LANA,NFT)
```

A sample run of RDGANA follows.

```
 RDGANA     TIME 15: 4: 2  DATE 10/ 3/86
 PROBE NO.:20F          SERIAL NO.:55617
 DR SER RES:       75.0  REF ATTN 50.0  DRIVER SHUNT CAP.:    .0000E+00
 PICK-UP SHUNT RESISTANCE: 1000000.0     PICK-UP SHUNT CAP.:    .0000E+00
 CABLE I.D. NO.:CABLE       LENGTH:      18.6      CAP.:   .4650E-10
FREQUENCY:  2.0000E+03        5.0000E+03        1.3000E+04
VOLTAGE OUT:  1.1000E+00        1.1000E+00        1.1000E+00
 READINGS FROM NOMINAL TUBE SAMPLE
 -60.970  -44.12 -47.820  -46.12 -31.740  -66.99
PSET  MAG1    PHA1    MAG2    PHA2    MAG3    PHA3     RHO            LOFF
   1-60.690 -42.69-47.390 -45.15-31.243 -66.07   617.9000     .0000
   2-63.683 -51.38-51.387 -48.86-34.953 -63.53  1191.0000     .0000
   3-65.120 -58.60-53.773 -55.59-37.680 -66.04  1975.0000     .0000
   4-61.700 -46.63-48.860 -49.00-32.927 -69.42   617.9000    1.0000
   5-64.233 -54.63-52.363 -52.86-36.193 -67.22  1191.0000    1.0000
   6-65.510 -61.96-54.657 -60.28-38.903 -71.02  1975.0000    1.0000
   7-62.283 -49.36-49.660 -51.21-33.847 -71.44   617.9000    2.0000
   8-64.620 -57.59-53.103 -56.31-37.170 -70.55  1191.0000    2.0000
   9-65.717 -63.30-55.030 -62.69-39.470 -73.65  1975.0000    2.0000
```

The input data file used for this run was named CONDIN.DAT by the main program. A listing of this file follows.

```
20F
55617
75. 50.
0.00E+00
1.00E+06
0.00E+00
CABLE
18.6
.0465E-09
3 9 2
2.0E3 5.0E3 1.3E4
1.1 1.1 1.1 1.1
```

| RHO | LOFF |
|---|---|
| 617.9 | 0.000 |
| 1191. | 0.000 |
| 1975. | 0.000 |
| 617.9 | 1.000 |
| 1191. | 1.000 |
| 1975. | 1.000 |
| 617.9 | 2.000 |
| 1191. | 2.000 |
| 1975. | 2.000 |

PROGRAM FITANA

The program FITANA takes the data that RDGANA records and makes a least-squares fit of nonlinear functions of the readings to the properties. The function type can be linear, logarithmic, exponential, or inverse. The operator can select the type of function and the degree of polynomial for the magnitude and phase of each function at each frequency. The magnitude and phase at each frequency or the entire frequency can be omitted by selecting a polynomial of degree zero. For initial investigations, it is preferable to run RDGANA at a large number of frequencies and determine the ones that give the best fit from FITANA. The program will print the "fit," which is a measure of how well the polynomial was able to fit the measured readings to the property values that produced the measured data, and a "drift" calculated from the maximum possible change caused by a typical variation in the magnitude (0.005 dB) and the phase (0.01°). These two parameters give a good estimation of how accurately the property can be measured. When the set of coefficients that gives the best, or, at least, an adequate, fit and drift is found, the operator can save the coefficients in memory and on a disk file, COEDAT.DAT. The operator can then proceed to the next portion of the program, which will take new readings of the magnitude and phase of whatever samples are placed near the probe. New property calculations can be made by using the stored coefficients. The probe can be positioned on the sample by hand or by using the Modulynx Mechanical Scanner, operated in a manual mode. In addition to omitting the frequencies, the operator can also eliminate certain sets of the properties by jumping around them as they are read in from the data file. However, this requires editing and recompiling FITANA. The jump points are located at lines 215 and 255, and examples of property selections are given in the comment statements. This technique is much faster than rerunning RDGANA.

## Coil and Equipment Setup

After calculating the least-squares fit, FITANA takes new readings. The connections for the new readings should be done the same way that the connections for the original readings were done. Either the Modulynx Mechanical Scanner or a manual method can be used. Please refer to the diagrams and illustrations for RDGANA because the procedures are essentially the same.

## Input Data

The input data for this program are complicated, and they must be put in correctly. In particular, incorrect values for the dimension of some of the arrays will cause incorrect fits. Input data that are not described in the program or were not described previously in other programs are listed below.

NPROBE – probe number, read as character data

NCABLE – cable identification number, read as character data

ADUM*82 – dummy character variable, used to position after a REWIND

PROPTY – array of property names, character data

INDAT – name of input data file, named OTDAT in RDGANA

COEDAT – name of output data file for coefficients

ARRAY DIMENSIONS THAT MUST BE CHANGED:

READNG(NPT+1,IRDPRM+1) – array storing the computed functional values of instrument readings for least-squares function fitting

PRO(NPT) – array of property values read from input file

ARRAY DIMENSIONS THAT MAY NEED TO BE CHANGED:

TMDFT, PHDFT – magnitude and phase drift, respectively

COE,COEF – arrays of coefficients computed in least-squares fits

NPOL,JPOL - arrays storing identification of least-squares function and polynomial type

JOFSET - offset control parameter for least-squares fitting function

JRDPR - number of terms in a least-squares fitting function

PHASE1 - phase readings, to be expanded into nonlinear readings in RDGEXP

TMAG1 - magnitude readings, to be expanded into nonlinear readings in RDGEXP

PRONAM - array of property names

PROPTY - intermediate array of property names

POLARY - array of alphanumeric representation of least-squares fitting function terms; includes cross terms and polynomial degree

DATA IRDPRM - maximum number of coefficients allowed in least-squares fitting expansion

DATA NUNIT - dimensional unit control 1-English, 2-metric

Subroutines

The systems subroutines used for program FITANA are _GETTIME and _GETKEY. ANARDG, ALSQS, POLTYP, and RDGEXP are used in this program and listed in the Appendix.

Main Program

A listing of the program FITANA follows.

```
      PROGRAM FITANA
C     VERSION 3 OCTOBER 1986
C     PROGRAM TO PERFORM A LEAST SQUARES FIT TO DATA READ INTO A DISK
C     FILE BY RDGANA PROGRAM AND THEN PERFORM CONTINUOUS BACK CALCULATIONS
C     USING INSTRUMENT READINGS THAT ARE MADE BETWEEN EACH DISPLAYED SET.
C
C     IRDPRM=MAXIMUM NUMBER OF COEFFICIENTS IN EXPANSION
C     LITEK=LOGICAL INPUT UNIT
C     LOTEK=OPERATOR OUTPUT UNIT FOR PROMPTING AND DISPLAY
C     LOU=LOGICAL OUTPUT UNIT FOR PERMANENT RECORD
C     NF=FREQUENCY INDEX
C     NFT=NUMBER OF FREQUENCIES
C     NP=PROPERTY INDEX
```

```
C     NPROPM=MAXIMUM NUMBER OF PROPERTIES CALCULATED(=6 NOW)
C     NPT=TOTAL NUMBER OF POINTS READ (LINES OF DATA) IN THE DATA SET.
C        =(SAMPLES)*(FLAWS)*(INTERFACES)*(L.O.)*(READINGS)*( - - - -
C     NPTT=VALUE OF NPT READ FROM FILE 30
C     NPRINT=PRINT AND TRANSFER INDEX.
C     NSTOP=INDEX TO STOP THE INSTRUMENT READINGS
C
      REAL L2,L4,L3,L5,L6
      CHARACTER*6 NPROBE,NCABLE,COIL
      CHARACTER COEDAT*8,FNAME*12,BLANKS*4,PRONAM(7)*4,AXISNM(3)*4,ADUM*82
      CHARACTER PROPTY(7)*4,STOP*4,INDAT*8,POLARY(16,5)*4
C  DIMENSIONS THAT ARE NOT CHANGED:
      DIMENSION NCONV(4),VOLSTD(12)
C
C  DIMENSIONS THAT ARE CHANGED:
C **  DIMENSION READNG(NPT+1,IRDPRM+1),PRO(NPT),POLARY(IRDPRM+1,5)
C     DIMENSION PROP(1,NPROPM),TMDFT(NFT),PHDFT(NFT),JPOL(6,NFT,NPROPM)
C     DIMENSION COE(IRDPRM),COEF(IRDPRM,NPROPM)
C     DIMENSION RDG1(1,IRDPRM),,NPOL(6,NFT)
C     DIMENSION JOFSET(NPROPM),JRDPR(NPROPM)
C     DIMENSION PROPTY(NPROPM),PRONAM(NPROPM),FREQ(NFT),VOUT(NFT)
C     DIMENSION TMAG1(1,NFT),PHASE1(1,NFT),XMAG(NFT),XPHA(NFT)
C
C  THE APPROPRIATE NUMBERS SHOULD BE INSERTED IN THE FOLLOWING
C  DIMENSION STATEMENTS; COMMENTED STATEMENTS MARKED ** MANDATORY
C
      DIMENSION READNG(10,16),PRO(10),PROP(1,7)
      DIMENSION TMDFT(4),PHDFT(4),JPOL(6,4,7),COE(15),COEF(15,7)
      DIMENSION RDG1(1,15),NPOL(6,4),JOFSET(7),JRDPR(7)
      DIMENSION FREQ(4),VOUT(4),SUMPRO(7),SSPRO(7),SDVPRO(7)
      DIMENSION TMAG1(1,4),PHASE1(1,4),XMAG(4),XPHA(4)
C
```

```
C  DATA THAT MAY NEED TO BE CHANGED:
      DATA NPT/9/,NPRINT/0/,LOU/6/,LITEK/0/,LOTEK/0/,NRDG/30/
      DATA NCL/28/,LID/37/,LOD/38/,LANA/43/,MBUS/46/
      DATA IRDPRM/15/,IR/1/,NCHS/6/,NPROPM/7/,NPROPT/1/
      DATA STOP/'STOP'/,BLANKS/'    '/,NUNIT/1/
      DATA NLINES/7/,INDAT/'CONDOT'/,COEDAT/'CONDCOE '/
C
C PRINT TITLE AND DATE
      OPEN(LOU,FILE='#PR',ERR=990)
      CALL_GETTIME(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
    2 FORMAT(' FITANA    TIME ',I2,':',I2,':',I2,'  DATE ',I2,'/',I2,'/',I2)
      WRITE(LOU,2,ERR=990)IHR,IMN,ISE,IMO,IDA,IYR
C
   30 FORMAT(1X)
C
C     OPEN  FILE FOR INPUT DATA FROM RDGANA,ASSUMED ON DEFAULT DISK
   40 FORMAT(A8)
      FNAME=INDAT//'.DAT'
   41 FORMAT(A12)
      OPEN(LID,FILE=FNAME,STATUS ='OLD',ERR=991)
C     OPEN FILE FOR OUTPUT COEFICIENT DATA STORAGE - ASSUMED ON DEFAULT DISK
      FNAME=COEDAT//'.DAT'
      OPEN(LOD,FILE=FNAME,STATUS ='NEW',ERR = 992)
C
C     HEWLETT PACKARD 4192A LF IMPEDANCE ANALYZER ADDR=3 ON GPIB BUSS
C     ADDR=C? ON IBM BUSS DRIVER, FORTRAN LOGICAL UNIT = 43
      OPEN(LANA,FILE='#BUSC?')
      WRITE(0,*)'IMPEDANCE ANALYZER  FILE OPENED'
      WRITE(LANA,*)'A5B2H1V1T3F0'
C
C READ INITIAL INFORMATION IN DIRECT ACCESS FILE LID ON DISK
C
      READ(LID,55,ERR=42)IHR,IMN,ISE,IMO,IDA,IYR
```

```
  55  FORMAT(6(1X,I4))
  42  WRITE(LOU,140)IHR,IMN,ISE,IMO,IDA,IYR
      WRITE(0,140)IHR,IMN,ISE,IMO,IDA,IYR
      READ(LID,60,ERR=62) NPROBE,NSER,R0,RATT,CAPDR,R9,CAPPU,NCABLE,
     *               CABLEL,CCABLE
  60  FORMAT(A6,1X,I5,5(E13.4),1X,A6,2(E13.4))
  62    WRITE(0,60) NPROBE,NSER,R0,RATT,CAPDR,R9,CAPPU,NCABLE,
     *               CABLEL,CCABLE
      READ(LID,65) NFT,NPTT,NPROPM
      WRITE(0,65) NFT,NPTT,NPROPM
  65  FORMAT(3(1X,I5))
      READ(LID,70)(FREQ(NF),NF=1,NFT)
  70  FORMAT(6(E13.4))
      WRITE(0,70)(FREQ(NF),NF=1,NFT)
      READ(LID,70)(VOUT(NF),NF=1,NFT)
      WRITE(0,70)(VOUT(NF),NF=1,NFT)
      READ(LID,75)(PRONAM(NPR),NPR=1,NPROPM)
  75  FORMAT(7(1X,A4))
      WRITE(0,75)(PRONAM(NPR),NPR=1,NPROPM)
C     READ NOMINAL  STANDARD VOLTAGES
C
      LNF=2*NFT
      READ(LID,80)(VOLSTD(NF),NF=1,LNF)
  80  FORMAT(6(1X,F7.3,1X,F7.2))
      WRITE(0,80)(VOLSTD(NF),NF=1,LNF)
C
C END OF SECTION WHICH READS INITIAL DIRECT ACCESS FILE
C
C      THE DATE AND TIME THE DATA WAS TAKEN ARE PRINTED
C
  140 FORMAT(' CALIBRATION DATA TAKEN ',I2,':',I2,':',I2,'  DATE ',I2,'/'
     *,I2,'/',I2)
  160 FORMAT(1X)
C     THE FUNDAMENTAL FREQUENCY IS WRITTEN TO THE IMPEDANCE ANALYZER
      WRITE(LANA,81)VOUT(1),FREQ(1)
```

```
   81 FORMAT(1X,'OL',F5.3,'ENFR',F9.3,'EN')
C
C     COIL DATA IS READ FROM  REF.DAT OR CIR.DAT FILE
C
C     FILE TO INSERT INTO REFLECTION OR CIRCULAR PROGRAMS FOR COIL
C     DATABASE REFERENCES
C
C       THE INPUT DATA FOR THE REFLECTION COIL IS READ FROM FILE 28
C       SEE FIG.2,P.7 AND FIG.4, P.7, ORNL-TM-4107, FOR DEFINITIONS.
C
      OPEN(28,FILE='REF.DAT',STATUS ='OLD',ERR=994)
   10 READ(28,11)COIL,RBAR,R1,R2,L3,R3,R4,L4,L5,L6
     *,RDCDR,RDCPU,TNDR,TNPU
   11 FORMAT(A6,9F8.4,F10.4,F11.4,2F8.1)
      IF(COIL.EQ.'END   ')WRITE(0,*)'REF COIL NOT FOUND'
      IF(COIL.EQ.'END   ')GO TO 14
      IF(COIL.NE.NPROBE)GO TO 10
      GO TO 630
C
C     THE INPUT DATA FOR CIRCULAR COILS IS READ FROM FILE 29
C     SEE TABLE 3,PAGE 10 OF ORNL/NUREG/TM-335 FOR DEFINITIONS
C
   14 OPEN(29,FILE='CIR.DAT',STATUS ='OLD',ERR=995)
   15 READ(29,20)COIL,RBAR,R1,R2,XL,ZL,RLIM,RDCDR
     *,RDCPU,TNDR,TNPU,ZLDR,ZLPU,X
   20 FORMAT(A6,6F8.4,2F10.4,2F8.1,2E11.3,F8.1)
      IF(COIL.EQ.'END   ')WRITE(0,*)' CIR COIL NOT FOUND'
      IF(COIL.EQ.'END   ')GO TO 630
      IF(COIL.NE.NPROBE)GO TO 15
      GO TO 630
C
C     LEAST SQUARES DESIGN SECTION.
C
```

```
C     SELECT PROPERTY TO BE FITTED AND SET UP PROPERTY ARRAY.
C
  300 MSET=NPT
      MSET1=MSET+1
      IF(NPROPT.GT.NPROPM) GO TO 860
  310 WRITE(LOTEK,320)(NPR,PRONAM(NPR),NPR=1,NPROPM)
  320 FORMAT(' SELECT NUMBER OF THE PROPERTY TO BE FITTED:',
     */,7(I3,1X,A4),' ? ')
      READ(LITEK,*)NPROP
      IF(NPROP.GT.NPROPM)NPROP=NPROPM
  350 WRITE(LOTEK,360)
  360 FORMAT(' TYPE 1 IF THERE IS OFFSET; 0 IF NO OFFSET:',/)
      READ(LITEK,*) JOFSET(NPROPT)
      IOFSET=JOFSET(NPROPT)
      IRDPR=IOFSET
  370 WRITE(LOTEK,380)
  380 FORMAT(' SELECT THE NUMBER OF THE FUNCTION TYPE, POLYNOMIAL',
     *' DEGREE, & # OF '/,
     *' CROSS TERMS FOR EACH MAGNITUDE & PHASE')
      WRITE(LOTEK,390)
  390 FORMAT(' FUNCTION TYPE:1=LINEAR 2=LOG 3=EXP 4=INV ')
  400 WRITE(LOTEK,160)
      WRITE(LOTEK,410)
  410 FORMAT(25X,'FCTN POL # CROSS'/,25X'TYPE DEG TERMS')
      DO 450 NF=1,NFT
      DO 440 NC=1,2
      NCC=NC*3
      NCP=NCC-1
      NCF=NCP-1
      IF(NC.EQ.1) WRITE(LOTEK,420) FREQ(NF)
      IF(NC.EQ.2) WRITE(LOTEK,430) FREQ(NF)
  420 FORMAT(' MAG AT ',1PE12.6,' KHZ  ',¼)
```

```
  430 FORMAT(' PHA AT ',1PE12.6,' KHZ  ',¼)
      READ(LITEK,*)
     *JPOL(NCF,NF,NPROPT),JPOL(NCP,NF,NPROPT),JPOL(NCC,NF,NPROPT)
C     JPOL(3,NF,NPROPT)=0
      IRDPR=IRDPR+JPOL(NCP,NF,NPROPT)+JPOL(NCC,NF,NPROPT)
      JRDPR(NPROPT)=IRDPR
      NPOL(NCF,NF)=JPOL(NCF,NF,NPROPT)
      NPOL(NCP,NF)=JPOL(NCP,NF,NPROPT)
      NPOL(NCC,NF)=JPOL(NCC,NF,NPROPT)
  440 CONTINUE
  450 CONTINUE
      IRDPR1=IRDPR+1
      IF(IRDPRM.LT.JRDPR(NPROPT))WRITE(LOTEK,460)
      IF (IRDPRM.LT.JRDPR(NPROPT))GO TO 630
  460 FORMAT(' ERROR: # OF TERMS IN POLARY EXCEEDS DIMENSION')
      JROW=IRDPRM+1
      CALL POLTYP(POLARY,JROW,IRDPR,NPOL,6,NFT,2,IOFSET,LOTEK)
C
C      EXPAND THE RAW READINGS INTO IRDPR READINGS.
C
  470 DO 480 NF=1,NFT
      TMDFT(NF)=0.
      PHDFT(NF)=0.
  480 CONTINUE
      REWIND(LID)
      DO 481 IREC=1,NLINES
      READ(LID,483)ADUM
  483 FORMAT(A2)
  481 CONTINUE
      NR=1
      DO 490 NP=1,NPTT
C     WRITE(0,*)NP,NPTT
      READ(LID,*)(TMAG1(1,NF),PHASE1(1,NF),NF=1,NFT),
     *(PROP(1,NPR),NPR=1,NPROPM)
```

```
 482  FORMAT(8(F7.3,1X),F9.3,1X,F7.3)
C
C     THE PROPERTIES CAN BE SET AND MODIFIED IN THIS SECTION
C
C     IF THIS PROPERTY IS NOT TO BE USED TRANSFER TO 490
C
C     IF(PROP(1,1).GT.10000)GO TO 490
C     IF(PROP(1,1).EQ.1217.250)GO TO 490
C     IF(PROP(1,2).GT.0.003)GO TO 490
C
      CALL RDGEXP(RDG1,TMAG1,PHASE1,NPOL,IOFSET,TMDFT,PHDFT,
     *1,IRDPR1,1,NFT,1,1,1,1)
      PRO(NR)=PROP(1,NPROP)
      DO 485 IRD=1,IRDPRM
      READNG(NR,IRD)=RDG1(1,IRD)
 485  CONTINUE
      NR=NR+1
C     WRITE(0,*)IRDPR
  490 CONTINUE
C
C     DO THE LEAST SQUARES FIT OF THE READINGS TO THE PROPERTIES.
C
      CALL ALSQS(READNG,PRO,COE,RSOS,MSET,IRDPR,MSET1)
C
C     CALCULATE THE DIFFERENCES IN THE FIT AND THE MAXIMUM DRIFTS.
C
  500 SSDRIF=0.
      SSDIFF=0.
      IF(NPRINT.EQ.2)WRITE(LOTEK,510)PRONAM(NPROP)
      IF(NPRINT.EQ.2)WRITE(LOU,510)PRONAM(NPROP)
  510 FORMAT(' PSET',8X,A4,9X,'CAL',8X,'DIFF',7X,'DRIFT')
      REWIND(LID)
```

```
      DO 515 IREC=1,NLINES
      READ(LID,483)ADUM
  515 CONTINUE
      NR=1
      DO 570 NP=1,NPTT
      DRIFT=0.
      READ(LID,*)(TMAG1(1,NF),PHASE1(1,NF),NF=1,NFT),
     *(PROP(1,NPR),NPR=1,NPROPM)
C
C      TRANSFER TO 570 IF PROPERTY IS NOT THE ONE WE WANT
C
C     IF(PROP(1,1).GT.10000)GO TO 570
C     IF(PROP(1,1).EQ.1217.250)GO TO 570
C     IF(PROP(1,2).GT.0.003)GO TO 570
C
      DO 540 NF=1,NFT
      DO 530 NC=1,2
C
C     ONE MAGNITUDE OR PHASE DRIFT IS SET ON AT A TIME.
C
      IF(NC.EQ.1)TMDFT(NF)=0.005
      IF(NC.EQ.2)PHDFT(NF)=.01
      CALL RDGEXP(RDG1,TMAG1,PHASE1,NPOL,IOFSET,TMDFT,PHDFT,
     *1,IRDPR1,1,NFT,1,1,1,1)
C
C     THE POLYNOMIAL IS CALCULATED
C
      SUM=0.
      DO 520 IR=1,IRDPR
      SUM=SUM+COE(IR)*RDG1(1,IR)
  520 CONTINUE
      DRIFT=DRIFT+ABS(READNG(NR,IRDPR1)-SUM)
```

```
      TMDFT(NF)=0.
      PHDFT(NF)=0.
  530 CONTINUE
  540 CONTINUE
      DIFF=PRO(NR)-READNG(NR,IRDPR1)
      SSDIFF=SSDIFF+DIFF*DIFF
      SSDRIF=SSDRIF+DRIFT*DRIFT
      IF(NPRINT.NE.2)GO TO 565
C
C       THE ENTIRE FIT IS PRINTED OUT
C
      WRITE(LOU,560)NP,PRO(NR),READNG(NR,IRDPR1),DIFF,DRIFT
      WRITE(LOTEK,560)NP,PRO(NR),READNG(NR,IRDPR1),DIFF,DRIFT
  560 FORMAT(1X,I4,4F12.5)
  565 NR=NR+1
  570 CONTINUE
      SDRIF=SQRT(SSDRIF/FLOAT(MSET))
      SDIFF=SQRT(SSDIFF/FLOAT(MSET))
      WRITE(LOU,580)PRONAM(NPROP),SDIFF,SDRIF,(NPOL(1,NF),NPOL(4,NF),NF=1,
     *NFT)
      WRITE(LOU,585)IOFSET,(NPOL(2,NF),NPOL(5,NF),NF=1,NFT)
      WRITE(LOU,587)(NPOL(3,NF),NPOL(6,NF),NF=1,NFT)
      WRITE(LOU,30)
      WRITE(LOTEK,575)PRONAM(NPROP),SDIFF,SDRIF
  575 FORMAT(' RMS DIF IN ',A4,'=',F10.5,2X,'DRIFT=',F10.5)
  580 FORMAT(' RMS DIF IN ',A4,'=',F10.5,2X,'DRIFT=',F10.5,3X,'FCTN'12I2)
  585 FORMAT(I2,' CONSTANT',37X,'POL ',12I2)
  587 FORMAT(48X,'XTRM',12I2)
  590 IF (NPRINT.NE.3)GO TO 630
      IF(NPROPT.NE.1)GO TO 597
      WRITE(LOD,594)NFT
      WRITE(LOD,595)(FREQ(NF),NF=1,NFT)
      WRITE(LOD,595)(VOUT(NF),NF=1,NFT)
  594 FORMAT(I2)
```

```
  595 FORMAT(6(E13.4))
      WRITE(LOD,*)(VOLSTD(NF),NF=1,LNF)
  597 WRITE(LOU,160)
      WRITE(LOTEK,160)
      NCOED1=NCOED+4*IRDPRM+1
      WRITE(LOD,*)IRDPR
      DO 610 I=1,IRDPR
C     WRITE(LOU,600)I,COE(I),(POLARY(I,J),J=1,5)
      WRITE(LOTEK,600)I,COE(I),(POLARY(I,J),J=1,5)
  600 FORMAT(' COEF(',I2,')=',1PE15.7,4X,5A4)
      COEF(I,NPROPT)=COE(I)
      DO 610 NCO=1,4
C     WRITE(LOD,*)NCONV(NCO)
  610 CONTINUE
      NCOED=NCOED1
C
C     THE COEFFICIENT,OFFSET,NPOL AND IRDPR ARE WRITTEN ON THE DISC
C
      WRITE(LOD,615)PRONAM(NPROP),IRDPR,JOFSET(NPROPT)
  615 FORMAT(1X,A4,2I4)
      WRITE(LOD,*)(COEF(IR,NPROPT),IR=1,IRDPR)
      DO 620 NF=1,NFT
      WRITE(LOD,*)(NPOL(I,NF),I=1,6)
  620 CONTINUE
      PROPTY(NPROPT)=PRONAM(NPROP)
      NPROPT=NPROPT+1
C
  630 WRITE(LOTEK,640)
  640 FORMAT(' 1 FIT PROP 2 PRT ENTIRE FIT 3 PRT/SV COEF 4 CHG FCTN/POL'
     *,'TYP 5 RUN TEST',/)
      READ(LITEK,*)NPRINT
      GO TO(300,500,590,350,650),NPRINT
```

```
  650 NPROPT=NPROPT-1
      WRITE(LOU,160)
      GO TO 880
C     CALCULATES PROPERTIES FROM MAGNITUDES AND PHASES AND CONTINUOUSLY
C     DISPLAYS THE VALUES ON THE CRT TERMINAL.
C
  680 DO 690 NPRO=1,NPROPT
      SUMPRO(NPRO)=0.0
      SSPRO(NPRO)=0.0
  690 CONTINUE
      IF(NPRINT.EQ.2)WRITE(LOTEK,710)(PROPTY(NPRO),NPRO=1,NPROPT)
      IF(NPRINT.EQ.3)WRITE(LOTEK,715)(II,II,II=1,NFT)
      DO 825 NR=1,NRDG
  700 CONTINUE
  710 FORMAT(1X,6(8X,A4,1X))
  715 FORMAT(1X,6('       MAG(',I1,')       PHA(',I1,')'))
C
C     NEW READINGS ARE MADE FROM THE EDDY CURRENT INSTRUMENT.
C
  720 CALL ANARDG(XMAG,XPHA,FREQ,VOUT,LANA,NFT)
      DO 745 NF=1,NFT
      TMAG1(1,NF)=XMAG(NF)
      PHASE1(1,NF)=XPHA(NF)
  745 CONTINUE
C
C     EXPANSION OF TMAG1(1,NF) AND PHASE1(1,NF) INTO READNG(1,IRDPRM)
C
  725  DO 800 NPRO=1,NPROPT
      DO 790 NF=1,NFT
      DO 780 I=1,6
      NPOL(I,NF)=JPOL(I,NF,NPRO)
  780 CONTINUE
C     WRITE(0,721)(NPOL(I,NF),I=1,6)
```

```
C 721 FORMAT(6I3)
      TMDFT(NF)=0.
      PHDFT(NF)=0.
  790 CONTINUE
      IOFSET=JOFSET(NPRO)
      IRDPR=JRDPR(NPRO)
      IRDPR1=IRDPR+1
      CALL RDGEXP(RDG1,TMAG1,PHASE1,NPOL,IOFSET,TMDFT,PHDFT,
     *1,IRDPR1,1,NFT,1,1,1,1)
      PRO(NPRO)=0.
      DO 795 IR=1,IRDPR
      PRO(NPRO)=PRO(NPRO)+COEF(IR,NPRO)*RDG1(1,IR)
  795 CONTINUE
      SUMPRO(NPRO)=SUMPRO(NPRO)+PRO(NPRO)
      SSPRO(NPRO)=SSPRO(NPRO)+PRO(NPRO)*PRO(NPRO)
  800 CONTINUE
C     NSTART=1
      IF(NPRINT.EQ.2)WRITE(LOTEK,805)(PRO(NPRO),NPRO=1,NPROPT)
  805 FORMAT(1X,7(F13.5))
  820 FORMAT(1X,6(F12.4))
      CALL_GETKEY(IKY,IS,KBF1,KBF2,IRR)
      IF(IRR.EQ.0)GO TO 827
C
C     PROGRAM WILL STAY IN THIS LOOP UNTIL ANY KEY IS STRUCK.
C
  825 CONTINUE
  827 IF(NLABL+NPRINT.EQ.3)WRITE(LOU,710)(NPRO,NPRO=1,NPROPT)
      NLABL=NPRINT
      IF(NPRINT.EQ.2)WRITE(LOU,830)(PRO(NPRO),NPRO=1,NPROPT)
      IF(NPRINT.EQ.3)WRITE(LOU,830)((TMAG1(1,NF),PHASE1(1,NF)),NF=1,NFT)
  830 FORMAT(3X,6(F12.5))
  835 FORMAT('SDV',6(F12.5))
      IF(NPRINT.EQ.2.OR.NPRINT.EQ.3)GO TO 880
```

```
      DO 850 NPRO=1,NPROPT
      SUMPRO(NPRO)=SUMPRO(NPRO)/FLOAT(NRDG)
      SDVPRO(NPRO)=SQRT(ABS(SSPRO(NPRO)-SUMPRO(NPRO)*SUMPRO(NPRO)*FLOAT
     *(NRDG))
     *               /FLOAT(NRDG-1))
  850 CONTINUE
      WRITE(LOTEK,830)(SUMPRO(NPRO),NPRO=1,NPROPT)
      WRITE(LOTEK,835)(SDVPRO(NPRO),NPRO=1,NPROPT)
      WRITE(LOTEK,160)
      WRITE(LOU,830)(SUMPRO(NPRO),NPRO=1,NPROPT)
      WRITE(LOU,835)(SDVPRO(NPRO),NPRO=1,NPROPT)
      WRITE(LOU,160)
      GO TO 880
  855 WRITE(LOTEK,857)
  857 FORMAT(' LIMIT OF FILE 37 IS EXCEEDED.')
      GO TO 880
  860 WRITE(LOTEK,870)
  870 FORMAT(' PROP ARRAY IS FILLED.')
  880 WRITE(LOTEK,890)
  890 FORMAT(' PRINT :1.MEAS VOLT & PROPS 2.CAL&DISPLAY PROPS'
     *,' 3.RAW RDGS. 4.AVG NRDGS 5.STOP',/)
      READ(LITEK,*)NPRINT
      GO TO (1300,680,680,680,900),NPRINT
C
C     PRINT SUMMARY OF DATA ON FILE 37
C
1300  WRITE(LOU,1350) NPROBE,NSER
1350  FORMAT(' PROBE NO.:',A6,5X,' SERIAL NO.:',I5)
      WRITE(LOU,1360) R0,CAPDR
1360  FORMAT(' DRIVER SERIES RESISTANCE:',F10.1,5X,'DRIVER SHUNT CAP.:',
     *        E12.4)
      WRITE(LOU,1370) R9,CAPPU
1370  FORMAT(' PICK-UP SHUNT RESISTANCE:',F10.1,5X,'PICK-UP SHUNT CAP.:'
     *         ,E12.4)
      WRITE(LOU,1380) NCABLE,CABLEL,CCABLE
```

```
1380  FORMAT(' CABLE I.D. NO.:',A6,5X,'LENGTH:',F10.1,5X,'CAP.:',
     *         E12.4)
      WRITE(LOU,1410) (FREQ(NF),NF=1,NFT)
1410  FORMAT('FREQUENCY:',10(1PE12.4,5X))
      WRITE(LOU,1420) (VOUT(NF),NF=1,NFT)
1420  FORMAT('OUTPUT VOLTAGE:',10(1PE12.4,5X))
C
1700  CONTINUE
C
C     PRINT OUT READINGS AND PROPERTIES
C
      WRITE(LOU,160)
      WRITE(LOU,1750)(NF,NF,NF=1,NFT),(PRONAM(NPR),NPR=1,NPROPM)
1750  FORMAT(1X,'PSET',3('     MAG',I1,'     PHA',I1),7(4X,A4,1X))
      REWIND(LID)
      DO 1755 IREC=1,NLINES
      READ(LID,483)ADUM
 1755 CONTINUE
      DO 1800 NP=1,NPTT
      READ(LID,482)(TMAG1(1,NF),PHASE1(1,NF),NF=1,NFT),(PROP(1,NPR)
     *,NPR=1,NPROPM)
      WRITE(LOU,1760)NP,(TMAG1(1,NF),PHASE1(1,NF),NF=1,NFT),(PROP(1,
     *NPR),NPR=1,NPROPM)
1760  FORMAT(1X,I4,13(F10.4))
1800  CONTINUE
C
      GO TO 880
  990 WRITE(LOTEK,*)' ERROR IN OPENING PRINTER FILE-CHECK FOR OFF-LINE'
      GO TO 900
  991 WRITE(LOTEK,*)'ERROR IN OPENING INPUT DATA FILE'
      GO TO 900
  992 WRITE(LOTEK,*)'ERROR IN OPENING OUTPUT COEF DATA FILE'
      GO TO 900
```

```
  993 WRITE(LOTEK,*)'ERROR IN OPENING INSTRUMENT DATA FILE'
      GO TO 900
  994 WRITE(LOTEK,*)'ERROR IN OPENING REF DATA FILE'
      GO TO 900
  995 WRITE(LOTEK,*)'ERROR IN OPENING CIR DATA FILE'
  900 STOP
      END
C
      SUBROUTINE RDGEXP(READNG,TMAG,PHASE,NPOL,IOFSET,TMDFT,PHDFT,MSET1
     1,IRDPR1,M,NFT,NL,NLT,NP,NPTT)
C
      SUBROUTINE POLTYP(POLARY,JROW,MROW,NPOL,IROW,NFT,NC,IOFSET,IDEV)
C
      SUBROUTINE ALSQS(A,Y,B,R2,NN,MM,NA)
C
      SUBROUTINE ANARDG(XMAG,XPHA,FREQ,VOUT,LANA,NFT)
C
```

A sample run of FITANA follows. The input data for this run are the output data file from RDGANA. The file is very similar to the printout for RDGANA and will not be repeated here. In practice, this file is not even examined.

```
FITANA    TIME  8:19:24  DATE 10/ 6/86
CALIBRATION DATA TAKEN 15: 4: 2  DATE 10/ 3/86
RMS DIF IN  RHO=  23.56744  DRIFT=  24.72452   FCTN 1 1 1 1 1 1
1 CONSTANT                                     POL  1 1 1 1 1 1
                                               XTRM 0 0 0 0 0 0

RMS DIF IN  RHO=   3.71595  DRIFT=   3.13650   FCTN 1 1 1 1 1 1
1 CONSTANT                                     POL  0 0 0 0 3 2
                                               XTRM 0 0 0 0 0 1
```

| PSET | RHO | CAL | DIFF | DRIFT |
|---|---|---|---|---|
| 1 | 617.90000 | 616.69380 | 1.20618 | 1.30762 |
| 2 | 1191.00000 | 1190.27900 | .72095 | 2.61230 |
| 3 | 1975.00000 | 1977.73600 | -2.73584 | 4.41992 |
| 4 | 617.90000 | 623.69220 | -5.79218 | .36646 |
| 5 | 1191.00000 | 1188.83900 | 2.16052 | 2.85791 |
| 6 | 1975.00000 | 1969.23100 | 5.76929 | 4.66504 |
| 7 | 617.90000 | 613.03020 | 4.86981 | .29456 |
| 8 | 1191.00000 | 1194.53300 | -3.53308 | 3.03857 |
| 9 | 1975.00000 | 1977.66700 | -2.66748 | 4.59180 |

```
RMS DIF IN  RHO=   3.71595  DRIFT=   3.13650   FCTN 1 1 1 1 1 1
1 CONSTANT                                     POL  0 0 0 0 3 2
                                               XTRM 0 0 0 0 0 1

RMS DIF IN LOFF=    .04770  DRIFT=    .00803   FCTN 1 1 1 1 1 1
1 CONSTANT                                     POL  0 0 0 0 3 2
                                               XTRM 0 0 0 0 0 1
```

| | | |
|---|---|---|
| | 1967.52800 | 1.92042 |
| SDV | 4.72703 | .01745 |
| | 1968.78000 | 1.91472 |
| SDV | 4.06838 | .01790 |
| | 1967.77400 | 1.91792 |
| SDV | 4.45669 | .01813 |

Two different fits were run on the resistivity (RHO), one using three frequencies, the other using polynomials in the magnitude and phase at the highest frequency, 13 MHz. The latter gave a better fit, and these coefficients were saved on the disk. A comparison of the "correct" value of RHO and the value calculated from the polynomial for the entire property set is given. The lift-off is then fitted, using the same polynomial for the resistivity. This process is not necessary but turns out to be convenient and tends to run faster when back readings are being made. Next, three sets of resistivity readings are made using option 4 for average readings. The standard deviation of these readings agrees approximately with the calculated drifts. This technique has been demonstrated on a simple problem with a small data set but will work just as well for data sets ranging up to several thousand different readings.

## PROGRAM PLTANA

The final program of these three, PLTANA, is designed to work with a mechanical scanner and will make plots of raw magnitudes and phases or plots of the computed values. The computed values use the coefficient data computed by FITANA and stored in the file named COEDAT. The main difference between the readings made by FITANA and PLTANA is that the latter has complete control of the scanner, using the Modulynx Motion Controller. The starting location and step size in both the $x$ and $y$ axes can be specified, and the readings can be smoothed by an averaging scheme. This program will simulate the action of a scanner that would be used in an actual inspection and is also used to locate defects and other property variations before the more exact readings and fits are run. The program has features, such as automatic computation of the offset, that will help obtain the raw readings on scale.

### Coil and Equipment Setup

Program PLTANA will always be set up using the Modulynx Motion Controller to control the position of the *xyz* scanner or one of the two-axis tubing scanners. The apparatus and electrical connections are the same as for program RDGANA, except that manual positioning is not used. Please refer to those diagrams and illustrations.

Input Data

Below is a partial glossary of the input data, some the same as programs RDGANA and FITANA but many different. Everything that has not been described is given here.

FF – form-feed character (a form feed is sent after some plots)

PHASE – array of phase readings

TMAGM – array of magnitude readings

READNG – array storing the nonlinear functions of instrument readings for construction of properties using least-squares coefficients

COEF – array for storage of coefficients for computatiom of properties

PROP – array for storage of computed properties

VOLSTD – array for storage of standard values read by RDGANA

VOLTS – magnitude and phase arrays are converted to a single array

NSTEPS – number of steps needed to reach new location to scan

DELTA – change in distance to XNEW between each set of readings

NOLD – number of steps taken (in three axes) to get to current position

NDACH – store data for plotting in integer array, raw readings or calculated properties computation

NCHS – number of reading channels, = 2 × (number of frequencies)

NPR – used to store prior values of NDACH for smoothing

NTT,NTT1 – used for smoothing data, NTT = no readings averaged, NTT1 = NTT-1

POFSET,PGAIN – arrays of property offsets and gains, used to stay on grid

ROFSET,RGAIN – arrays of raw reading offsets and gains, used on graph

OFSET, GAIN – offset and gain to correct raw readings to standard value

IXX – variable that keeps count of the number of sets of readings made

MBUS – logical output number for Modulynx Axis Position Controller

NCUR – number of curves to be plotted, = 2 × number of frequencies or number of properties

NTIM – variable used to instruct GRAPRT to initialize if .EQ. 0

NS – number of channels to be smoothed

INTER – integer to control frequency of values printed on CRT

COEDAT – names file that coefficients are read from, .DAT extension added

RAWDAT – names file for output storage of raw data values

XMAX – maximum value for coordinates to be scanned

XNEW – new coordinate value for scanning

## Subroutines

Program PLTANA uses the following systems subroutines.

1. _GETTIME
2. _STIMEOUT
3. _SETOS
4. _SETOPTIONS
5. _GETKEY
6. _SETPRMODE
7. _SETCOLOR
8. _SWRITE

The program subroutines are listed below.

1. POSIT
2. ANARDG
3. CORRDG
4. GRAPRT
5. RDGEXP

## Main Program

A listing of the program PLTANA follows.

```
      PROGRAM PLTANA
C     (30 SEPTEMBER 1986)
C
C     PROGRAM WILL PLOT RAW READINGS OR CALCULATED PROPERTIES
```

```
C     FROM READINGS MADE BY THE HP 4192A IMPEDANCE ANALYZER ON THE
C     IBM.THE DATA IS READ OVER THE IEEE-488 BUSS,ADDRESS = 3
C     MODIFIED FOR 2.0 SOFTWARE
C
      CHARACTER PRONAM(7)*4,COEDAT*8,FNAME*13,MAPDAT*6
      CHARACTER FF*1
      DIMENSION NSTEPS(3),VOUT(6),FREQ(6)
      DIMENSION JPOL(6,6,6),JRDPR(6),JOFSET(6)
      DIMENSION READNG(16),NPOL(6,6),TMAGM(6),PHASE(6),TMDFT(6),PHDFT(6)
      DIMENSION COEF(16,6),PROP(6)
      DIMENSION VOLSTD(12),VOLTS(12),Y1(2),Y2(2),YVAL(2,12)
      DIMENSION OFSET(12),GAIN(12)
      DIMENSION DELTA(3),XNEW(3),XMAX(3),NOLD(3)
      DIMENSION NPR(10),NDACH(12),ROFSET(12),RGAIN(12),POFSET(6),PGAIN(6)
      DATA MAG/0/,LOU/6/,LITEK/0/,LOTEK/0/,NCHS/6/,NRDGS/1/,NRET/1/,NFT/6/
      DATA NTT/8/,NTT1/7/,NS/2/,NPR/10*700/,NAXIS/3/,NOLD/3*0/
      DATA NCUR/12/,IXX/1/,XFAC/1.19048/,NTIM/0/
      DATA INTER/10/,ICOEF/1/,NPROPM/1/
      DATA LANA/43/,MBUS/46/,LOD/38/,COEDAT/'ALWCOE '/,TMDFT/6*0./,PHDFT/
     *6*0./
      DATA FREQ/.1,.2,.5,1.0,2.0,5.0/
      DATA VOUT/6*1.1/,MAPDAT/'RAWDAT '/,XMAX/25.0,16.5,0.5/
      DATA ROFSET/22.3,179.3,14.0,-162.5,3.0,-136.2,-5.1,-109.,-11.8,-78.,
     *-15.,42.3/
      DATA RGAIN/12*100/
      DATA POFSET/0.00,1.00,0.04,3*0./,PGAIN/2.0E3,1.0E3,1.0E4,3*1./
      DATA OFSET/12*0.0/,GAIN/12*1.0/
      FF=CHAR(12)
C
C     READINGS FROM A STANDARD TUBE ARE SET
      OPEN(LOU,FILE='#PR')
      CALL_GETTIME(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
```

```
      WRITE(LOU,2,ERR=990)IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT(' PLTANA    TIME ',I2,':',I2,':',I2,'  DATE ',I2,'/',I2,'/',
     *I2)
C     HEWLETT PACKARD IMPEDANCE ANALYZER ADDR=3 ON GPIB BUSS
C     ADDR=C? ON IBM BUSS DRIVER, FORTRAN LOGICAL UNIT = 43
      OPEN(LANA,FILE='#BUSC?')
      WRITE(LANA,*)'A5B2H1V1T3F0'
      WRITE(LANA,81)VOUT(1),FREQ(1)
   81 FORMAT(1X,'OL',F5.3,'ENFR',F9.3,'EN')
C     MODULYNX AXIS POSITION CONTROLLER ADDR=6 ON GPIB BUSS,ADDR=F? ON
C     IBM BUSS DRIVER,FORTRAN LOGICAL UNIT = 46
      OPEN(MBUS,FILE='#BUSF?',ERR=991)
      CALL_STIMOUT(MBUS,1000)
      CALL_SETEOS(MBUS,$00,$0A)
      CALL_SETOPTIONS(MBUS,0,0,0,0,0,1,0)
      WRITE(MBUS,*)'XD YD ZE'
      WRITE(MBUS,*)'<CAH>'
      WRITE(MBUS,*)'XA20000XB640XH4000'
      FNAME=COEDAT//'.DAT'
      OPEN(LOD,FILE=FNAME,STATUS ='OLD',ERR = 992)
C     READ COEFICIENT DATA WRITTEN BY FITANA
      NPRO=0
      LNF=2*NFT
   20 CONTINUE
      READ(LOD,24,ERR=992)NFT
      READ(LOD,25)(FREQ(NF),NF=1,NFT)
      READ(LOD,25)(VOUT(NF),NF=1,NFT)
      LNF=2*NFT
   24 FORMAT(I2)
   25 FORMAT(6(E13.4))
      READ(LOD,*)(VOLSTD(NT),NT=1,LNF)
      WRITE(0,*)(VOLSTD(NT),NT=1,LNF)
C     WRITE(LOU,*)'VOLSTD'
```

```
C       WRITE(LOU,*)(VOLSTD(NT),NT=1,LNF)
   30 NPRO=NPRO+1
      READ(LOD,*,END=60)IRDPR
      WRITE(0,*)IRDPR
C       WRITE(LOU,*)'IRDPR'
C       WRITE(LOU,*)IRDPR
      READ(LOD,40)PRONAM(NPRO),JRDPR(NPRO),JOFSET(NPRO)
   40 FORMAT(1X,A4,2I4)
      WRITE(0,40)PRONAM(NPRO),JRDPR(NPRO),JOFSET(NPRO)
      READ(LOD,*)(COEF(IR,NPRO),IR=1,IRDPR)
      WRITE(0,*)(COEF(IR,NPRO),IR=1,IRDPR)
      DO 50 NF=1,NFT
      READ(LOD,*)(JPOL(L,NF,NPRO),L=1,6)
      WRITE(0,*)(JPOL(L,NF,NPRO),L=1,6)
   50 CONTINUE
      GO TO 30
   60 CONTINUE
      NPROPM=NPRO-1
   80 WRITE(LOTEK,90)
      JSET=0
      IXX=1
   90 FORMAT(' WHAT NEXT 1.RAW RDG 2.CAL PROPS '
     *,'3.SCAN/RAW 4.SCAN/CAL 5.RST 6.STOP? ')
      READ(0,*)NPRINT
      IF(NPRINT.EQ.3.OR.NPRINT.EQ.4)WRITE(LOTEK,100)
  100 FORMAT(' TYPE XNEW(1),XNEW(2),DELTA(1),DELTA(2) FOR POSITIONER ')
      IF(NPRINT.EQ.3.OR.NPRINT.EQ.4)READ(0,*)XNEW(1),XNEW(2),DELTA(1),
     *DELTA(2)
      IF(NPRINT.EQ.3.OR.NPRINT.EQ.4)
     * WRITE(LOU,102)XNEW(1),XNEW(2),DELTA(1),DELTA(2)
  102 FORMAT('X0=',F7.3,'    Y0=',F7.3,'    DX=',F7.3,'    DY=',F7.3)
      LNF=NFT*2
      GO TO (105,200,105,200,400,500),NPRINT
```

```
  105 CONTINUE
C     OPEN FILE FOR OUTPUT STORAGE OF RAW DATA POINTS TO FILE 'RAWDAT'
      FNAME=MAPDAT//'.DAT'
      OPEN(LOD,FILE=FNAME,STATUS ='NEW',ERR = 994)
      WRITE(LOTEK,110)(NF,NF,NF=1,NFT)
      WRITE(LOD,110)(NF,NF,NF=1,NFT)
      WRITE(LOTEK,120)
      WRITE(LOD,120)
  110 FORMAT('LOCATION     ',3('MAG(',I1,')     PHA(',I1,')     '))
  120 FORMAT(1X)
  130 IF(NPRINT.EQ.1)GO TO 140
      XNEW(1)=XNEW(1)+DELTA(1)
      XNEW(2)=XNEW(2)+DELTA(2)
      IF(XNEW(1).LT.0.0.OR.XNEW(1).GT.XMAX(1))GO TO 163
      IF(XNEW(2).LT.0.0.OR.XNEW(2).GT.XMAX(2))GO TO 163
       WRITE(MBUS,*)'XD YD ZE'
      CALL POSIT(XNEW,NOLD,NAXIS,MBUS)
      IF(MOD(IXX,84).EQ.0)WRITE(MBUS,*)';<CFA>'
      IF(MOD(IXX,84).EQ.0)WRITE(MBUS,*)';'
       WRITE(MBUS,*)'XE YE ZE'
  140 CALL ANARDG(TMAGM,PHASE,FREQ,VOUT,LANA,NFT)
      DO 141 NF=1,NFT
      VOLTS(2*NF)=PHASE(NF)
      VOLTS(2*NF-1)=TMAGM(NF)
  141 CONTINUE
  142 FORMAT(6(F11.3))
      WRITE(LOD,142)(VOLTS(NF),NF=1,LNF)
      IF(MOD(NCOUNT,INTER).EQ.0)WRITE(LOTEK,150)XNEW(1),(VOLTS(NF),
     *NF=1,LNF)
      NCOUNT=NCOUNT+1
  150 FORMAT(F6.3,6(F7.3,F7.2))
      DO 160 NF=1,LNF
      NDACH(NF)=RGAIN(NF)*(ROFSET(NF)+VOLTS(NF))
```

```
  160 CONTINUE
      CALL GRAPRT(NDACH,IXX,LNF,LOU,NTIM)
      CALL_GETKEY(IKY,IS,KBF1,KBF2,IER)
      IF(IKY.EQ.83.AND.IER.EQ.0)GO TO 163
      GO TO 130
  163 CALL_SETPRMODE(LOU,0)
      NTIM=0
      IXX=1
      IF(NPRINT.EQ.3)WRITE(LOU,170)(FREQ(NF),NF=1,NFT)
      IF(NPRINT.EQ.3)WRITE(LOU,180)(VOLTS(NF),NF=1,LNF)
      DO 165 NF=1,LNF
      VOLTS(NF)=VOLTS(NF)+ROFSET(NF)
  165 CONTINUE
      IF(NPRINT.EQ.3)WRITE(LOU,180)(VOLTS(NF),NF=1,LNF)
      IF(NPRINT.EQ.3)WRITE(LOU,180)(ROFSET(NF),NF=1,LNF)
      IF(NPRINT.EQ.3)WRITE(LOU,190)(RGAIN(NF),NF=1,LNF)
      IF(NPRINT.EQ.3)WRITE(LOU,192)(NDACH(NF),NF=1,LNF)
      IF(XNEW(1).GT.XMAX(1).OR.XNEW(2).GT.XMAX(2))WRITE(LOU,195)FF
  170 FORMAT(6(1PE9.2,'KHZ',1X))
  180 FORMAT(6(F7.2,F6.1))
  190 FORMAT(6(F7.0,F6.0))
  192 FORMAT(6(I7,I6))
  195 FORMAT(A1)
      GO TO 80
C
C     PROPERTY CALCULATION AND DISPLAY SECTION
C
  200 WRITE(LOTEK,217)(PRONAM(NPRO),NPRO=1,NPROPM)
  217 FORMAT(1X,'LOCATION ',6(6X,A4,1X))
      WRITE(LOTEK,120)
  220 IF(NPRINT.EQ.2)GO TO 221
      XNEW(1)=XNEW(1)+DELTA(1)
      XNEW(2)=XNEW(2)+DELTA(2)
      IF(XNEW(1).LT.0.0.OR.XNEW(1).GT.XMAX(1))GO TO 275
```

```
      IF(XNEW(2).LT.0.0.OR.XNEW(2).GT.XMAX(2))GO TO 275
       WRITE(MBUS,*)'XD YD ZE'
      CALL POSIT(XNEW,NOLD,NAXIS,MBUS)
      IF(MOD(IXX,84).EQ.0)WRITE(MBUS,*)';<CFA>'
      IF(MOD(IXX,84).EQ.0)WRITE(MBUS,*)';'
       WRITE(MBUS,*)'XE YE ZE'
  221 CALL ANARDG(TMAGM,PHASE,FREQ,VOUT,LANA,NFT)
      DO 222 NF=1,NFT
      VOLTS(2*NF)=PHASE(NF)
      VOLTS(2*NF-1)=TMAGM(NF)
  222 CONTINUE
      CALL CORRDG(VOLTS,OFSET,GAIN,NCHS)
      DO 260 NPRO=1,NPROPM
      IRDPR1=JRDPR(NPRO)+1
      IOFSET=JOFSET(NPRO)
      DO 230 NF=1,NFT
      PHASE(NF)=VOLTS(2*NF)
      TMAGM(NF)=VOLTS(2*NF-1)
      DO 225 I=1,6
      NPOL(I,NF)=JPOL(I,NF,NPRO)
  225 CONTINUE
  230 CONTINUE
      CALL RDGEXP(READNG,TMAGM,PHASE,NPOL,IOFSET,TMDFT,PHDFT,1
     *,IRDPR1,1,NFT,1,1,1,1)
      PROP(NPRO)=0.
      IRDP=JRDPR(NPRO)
      DO 240 IR=1,IRDP
      PROP(NPRO)=PROP(NPRO)+READNG(IR)*COEF(IR,NPRO)
  240 CONTINUE
C     WRITE(LOU,245)(READNG(IR),IR=1,IRDP)
C 245 FORMAT(10F8.4,/,5F8.4)
      NDACH(NPRO)=PGAIN(NPRO)*(POFSET(NPRO)+PROP(NPRO))
  260 CONTINUE
```

```
C     IF(JSET.EQ.0)POFSET(3)=0.140-PROP(3)
      JSET=1
C
C     SECTION TO SMOOTH THE DEFECT INDICATION BY AVERAGING SUCCESSIVE
C     DEFECT CALCULATIONS . CHANNEL NS IS SMOOTHED
C
      DO 265 NT=1,NTT1
      NPR(NT)=NPR(NT+1)
  265 CONTINUE
      NPR(NTT)=NDACH(NS)
      NDACH(NS)=0
      DO 167 NT=1,NTT
      NDACH(NS)=NDACH(NS)+NPR(NT)
  167 CONTINUE
      NDACH(NS)=NDACH(NS)/NTT
C
C     END OF SECTION
C
      K=MOD(NCOUNT,INTER)
      NCOUNT=NCOUNT+1
      IF(K.EQ.0)WRITE(LOTEK,270)XNEW(1),(PROP(NPRO),NPRO=1,NPROPM)
  270 FORMAT(1X,F8.3,6(F11.4))
      CALL GRAPRT(NDACH,IXX,NPROPM,LOU,NTIM)
      CALL_GETKEY(IKY,IS,KBF1,KBF2,IER)
      IF(IKY.EQ.83.AND.IER.EQ.0)GO TO 275
      GO TO 220
  275 CALL_SETPRMODE(LOU,0)
      WRITE(LOU,290)(PRONAM(NPRO),NPRO=1,NPROPM)
      WRITE(LOU,293)(PROP(NPRO),NPRO=1,NPROPM)
      WRITE(LOU,295)(POFSET(NPRO),NPRO=1,NPROPM)
      WRITE(LOU,300)(PGAIN(NPRO),NPRO=1,NPROPM)
      WRITE(LOU,305)(NDACH(NPRO),NPRO=1,NPROPM)
      IF(XNEW(1).GT.XMAX(1).OR.XNEW(2).GT.XMAX(2))WRITE(LOU,195)FF
```

```
 290 FORMAT('PROPERTY',7(6X,A4))
 293 FORMAT(8X,7(F10.3))
 295 FORMAT('OFFSET  ',7(F10.3))
 300 FORMAT('GAIN    ',7(1PE10.2))
 305 FORMAT('PLOT INT',7(I10))
     NTIM=0
     IXX=1
     GO TO 80
 400 WRITE(LOTEK,401)
 401 FORMAT(' TYPE XNEW(1),XNEW(2),FOR STANDARD READING ')
     READ(0,*)XNEW(1),XNEW(2)
      WRITE(MBUS,*)'XD YD ZE'
     CALL POSIT(XNEW,NOLD,NAXIS,MBUS)
     IF(MOD(IXX,84).EQ.0)WRITE(MBUS,*)';<CFA>'
     IF(MOD(IXX,84).EQ.0)WRITE(MBUS,*)';'
      WRITE(MBUS,*)'XE YE ZE'
     DO 405 NF=1,LNF
     VOLTS(NF)=0.
 405 CONTINUE
     DO 430 NA=1,5
     CALL ANARDG(TMAGM,PHASE,FREQ,VOUT,LANA,NFT)
     DO 410 NF=1,NFT
     VOLTS(2*NF)=PHASE(NF)/5.+VOLTS(2*NF)
     VOLTS(2*NF-1)=TMAGM(NF)/5.+VOLTS(2*NF-1)
 410 CONTINUE
 430 CONTINUE
     DO 450 NOF=1,NCHS
     OFSET(NOF)=VOLSTD(NOF)-GAIN(NOF)*VOLTS(NOF)
450  CONTINUE
     DO 460 NF=1,LNF
     ROFSET(NF)=(FLOAT(800*NF/(LNF+1)))/RGAIN(NF)-VOLTS(NF)
 460 CONTINUE
     WRITE(0,180)(VOLTS(NF),NF=1,LNF)
     WRITE(0,180)(ROFSET(NF),NF=1,LNF)
     GO TO 80
```

```
 500 CONTINUE
     GO TO 1000
 990 WRITE(LOTEK,*)'ERROR IN OPENING PRINTER FILE-CHECK FOR OFF-LINE'
     GO TO 1000
 991 WRITE(LOTEK,*)'ERROR IN OPENING MOTOR DATA BUSS'
     GO TO 1000
 992 WRITE(LOTEK,*)'ERROR IN OPENING INPUT COEF DATA FILE'
     GO TO 80
 993 WRITE(LOTEK,*)'ERROR IN OPENING INSTRUMENT DATA BUSS'
     GO TO 1000
 994 WRITE(LOTEK,*)'ERROR IN OPENING MOTOR POSITIONER FILE'
1000 CONTINUE
      WRITE(MBUS,*)'XD YD ZE'
     WRITE(MBUS,*)'<CAH>'
     WRITE(MBUS,*)'XE YE ZE'
     STOP
     END
C
     SUBROUTINE RDGEXP(READNG,TMAG,PHASE,NPOL,IOFSET,TMDFT,PHDFT,MSET1
    1,IRDPR1,M,NFT,NL,NLT,NP,NPTT)
C
     SUBROUTINE CORRDG(VOLTS,COFSET,GAIN,NCHS)
C
     SUBROUTINE POSIT(XNEW,NOLD,NAXIS,MBUS)
C
     SUBROUTINE GRAPRT(IY,IXX,NCUR,LOU,NTIM)
C
     SUBROUTINE ANARDG(XMAG,XPHA,FREQ,VOUT,LANA,NFT)
C
```

A sample run of PLTANA is shown in Fig. 10. In this figure, the third option, which scans and plots raw readings, was chosen.

ORNL-DWG 86-15956

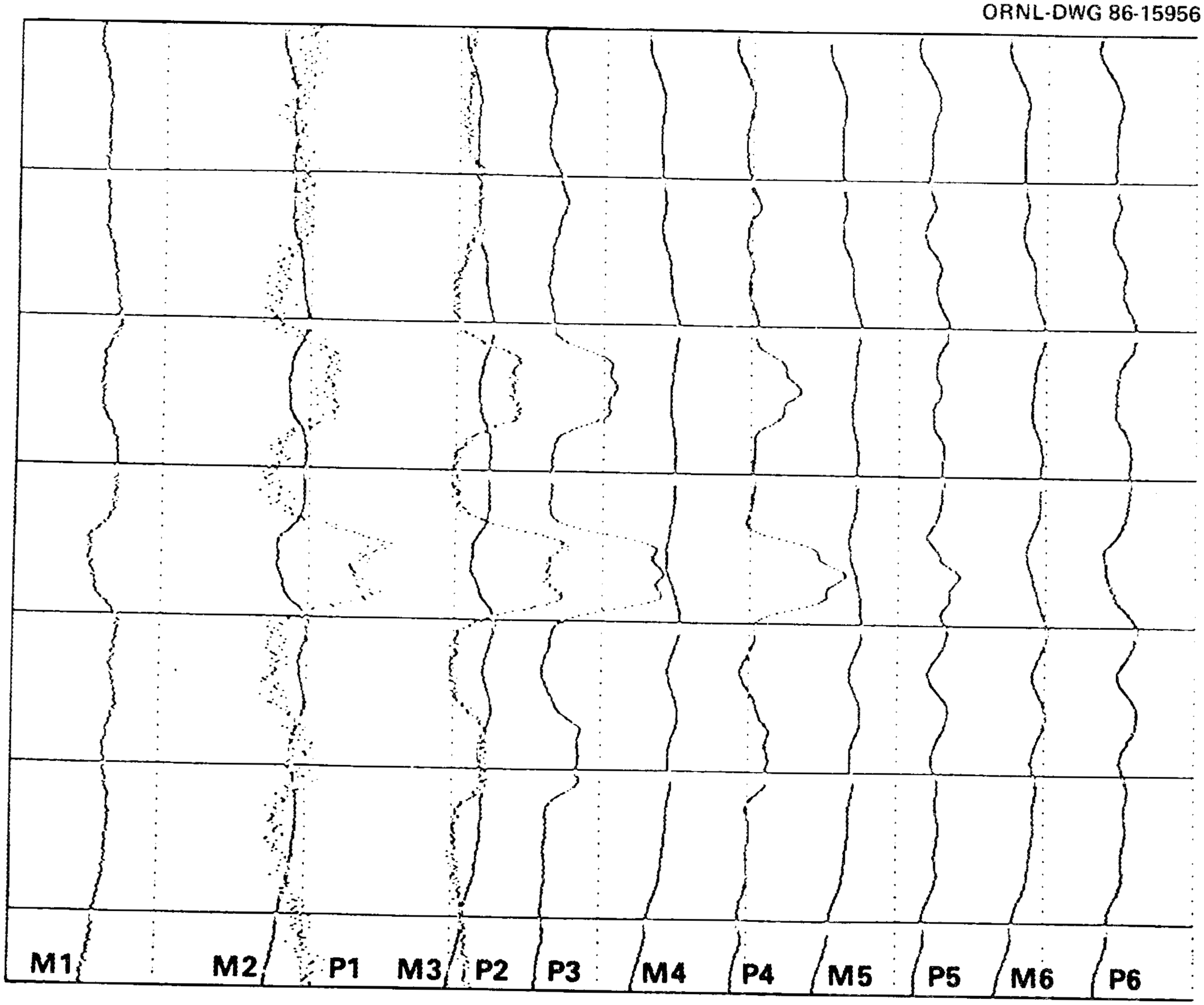



Fig. 10. Plot of raw data readings by PLTANA.

# PROGRAM NORIMP

This program will take impedance readings from a single absolute coil attached to the impedance analyzer. The real and imaginary parts of the impedance are read at different frequencies, first with the probe in air and then with the probe on a conductor. The number of frequencies, the minimum frequency, and the maximum frequency are set, and the intermediate frequencies are chosen exponentially between the minimum and maximum. The gain and offset of the real and imaginary axes of the plot can be set in data statements and varied on an interactive basis. In addition to the graphed normalized impedance, the coil inductance, the value of frequency times permeability times conductivity times mean radius squared, and the real and imaginary values can be displayed on the CRT. The coil inductance should stay constant with frequency until the coil approaches resonance. At this point, accurate eddy-current measurements are difficult to make, and the normalized impedance measurements will be inaccurate.

## DIAGRAM OF SETUP

Figure 2 shows a simplified impedance diagram for a single coil connected to the impedance analyzer by using the HP 16095A Probe Fixture. Refer to the HP Impedance Analyzer instruction manual for connection details.

## INPUT DATA

The following are the input data for NORIMP.

XRLA – array of real parts of impedance readings in air
XIMA – array of imaginary parts of impedance readings in air
XRLC – array of real parts of impedance readings on material
XIMC – array of imaginary parts of impedance readings on material
  arrays; dimension should be set equal to, or greater than, NFT
NFT – number of frequencies, VOLT – voltage used
NGR – device number for graphics driver
FMIN – minimum frequency, FMAX – maximum frequency (kHz)
GRL – gain, real axis, GIM – gain, imaginary axis
ORL – offset, real axis, OIM – offset, imaginary axis

SUBROUTINES

There are four systems subroutines in NORIMP

1. _GETKEY
2. _CLRWIN
3. _SETCOP
4. _VECTOR

The first subroutine, _GETKEY, is in Chap. 16 of the *Interface Library Reference Manual*; the other three are found in Chap. 7 and use the Graphics Driver. The program subroutines are listed below. Like the ones in previous programs, they are given in the Appendix.

1. ANARDG
2. GRID
3. GRAPH

MAIN PROGRAM

A listing of the program NORIMP follows.

```
      PROGRAM NORIMP
C     VERSION 23 SEPT 1986
C     MAKES A NORMALIZED IMPEDANCE PLOT ON THE CRT FROM DATA READ FROM
C     THE HEWLETT PACKARD IMPEDANCE ANALYZER.THE MAX AND MIN FREQUENCY,FMAX
C     & F MIN, THE REAL AND IMAGINARY AXIS GAINS,GIM & GRL ; AND THE
C     NUMBER OF FREQUENCIES NFT MAY BE SET.
C
      CHARACTER*3 SCALE(11)
      DIMENSION XRLA(51),XIMA(51),FREQ(51),WUSR(51)
      DIMENSION XRLC(51),XIMC(51),VOUT(51)
      DATA NFT/51/,FMIN/0.200/,FMAX/1000./,VOLT/1.1/
      DATA LANA/43/,NGR/39/,GRL/1000./,ORL/50./,GIM/500./,OIM/-50./
      DATA SCALE/'0.0','0.1','0.2','0.3','0.4','0.5','0.6','0.7','0.8',
     *'0.9'
     *,'1.0'/
C     SET UP GRAPHICS SUBROUTINE
      OPEN(NGR,FILE='#GR')
```

```
C
C     HEWLETT PACKARD 4192A LF IMPEDANCE ANALYZER ADDR=3 ON GPIB BUSS
C     ADDR=C? ON IBM BUSS DRIVER, FORTRAN LOGICAL UNIT = 43
      OPEN(LANA,FILE='#BUSC?')
      WRITE(0,*)'IMPEDANCE ANALYZER  FILE OPENED'
C
C WRITE SET UP VALUES FOR IMPEDANCE ANALYZER,FIRST FREQUENCY AND VOLTAGE
C     OUT
C
      WRITE(0,*)'PLACE COIL IN AIR'
      WRITE(0,*)'TYPE IN MIN ,MAX FREQ(KHZ),NO
     *  FREQS,RBAR(IN.),RHO(MICROHM CM)'
      READ(0,*)FMIN,FMAX,NFT,RBAR,RHO
      DELFR=EXP(ALOG(FMAX/FMIN)/FLOAT(NFT-1))
      WFACT=409.6*RBAR*RBAR/RHO
      VOUT(1)=VOLT
      FREQ(1)=FMIN
      WUSR(1)=WFACT*FMIN
      DO 50 NF=2,NFT
      FREQ(NF)=FREQ(NF-1)*DELFR
      VOUT(NF)=VOUT(NF-1)
      WUSR(NF)=WFACT*FREQ(NF)
   50 CONTINUE
      WRITE(LANA,*)'A2B3H1V1C2T3F0'
      WRITE(LANA,60)VOUT(1),FREQ(1)
   60 FORMAT(1X,'FR',F9.3,'ENOL',F5.3,'EN')
      CALL ANARDG(XRLA,XIMA,FREQ,VOUT,LANA,NFT)
      WRITE(0,*)'POSITION PROBE ON SAMPLE,HIT ANY KEY TO GO'
   70 CALL_GETKEY(IKY,IS,KBF1,KBF2,IRR)
      IF(IRR.NE.0)GO TO 70
      WRITE(0,*)'TYPE S TO STOP PROG,D TO PRINT DATA,O TO CHANGE OFFSET,
     *GAIN'
      CALL ANARDG(XRLC,XIMC,FREQ,VOUT,LANA,NFT)
   80 CALL_CLRWIN(NGR)
      F0=1000.*FREQ(1)
```

```
      CALL_GOTOXY(4,0)
      WRITE(0,90)RBAR,RHO,F0,NFT
   90 FORMAT('RBAR=',F6.3,' RHO=',F9.3,' F0=',F9.0,' NFT=',I2)
      CALL GRID(NGR)
      IM1=GIM*(XIMC(1)/XIMA(1))+OIM
      IR1=GRL*((XRLC(1)-XRLA(1))/XIMA(1))+ORL
      DO 100 NF=2,NFT
      IM2=GIM*(XIMC(NF)/XIMA(NF))+OIM
      IR2=GRL*((XRLC(NF)-XRLA(NF))/XIMA(NF))+ORL
      CALL GRAPH(IR1,IM1,IR2,IM2,NGR)
      IR1=IR2
      IM1=IM2
  100 CONTINUE
      CALL_SETCOP(NGR,150,15)
      CALL_CHARSTR(NGR,'NORMALIZED REAL IMPEDANCE')
      CALL_SETCOP(NGR,15,100)
      CALL_SETORIENT(NGR,1)
      CALL_CHARSTR(NGR,'NORMALIZED IMAGINARY IMPEDANCE')
      CALL_SETORIENT(NGR,0)
      DO 120 IXX=1,11
      IR1=GRL*0.1*FLOAT(IXX-1)+ORL -15
      IF(IR1.GT.450)GO TO 120
      CALL_SETCOP(NGR,IR1,30)
      CALL_CHARSTR(NGR,SCALE(IXX))
  120 CONTINUE
      DO 130 IXX=1,11
      IM1=GIM*0.1*FLOAT(IXX-1)+OIM -3
      IF(IM1.GT.450.OR.IM1.LT.40)GO TO 130
      CALL_SETCOP(NGR,18,IM1)
      CALL_CHARSTR(NGR,SCALE(IXX))
  130 CONTINUE
```

```
  199 CALL_GETKEY(IKY,IS,KBF1,KBF2,IRR)
      IF(IKY.EQ.68.AND.IRR.EQ.0)GO TO 299  ! PRINT DATA IF D IS TYPED
      IF(IKY.EQ.79.AND.IRR.EQ.0)GO TO 399  ! CHANGE OFFSET & GAIN IF O SI
     *                                       TYPED
      IF(IKY.EQ.83.AND.IRR.EQ.0)GO TO 999  ! STOP IF S IS TYPED
      GO TO 199
  299 CONTINUE
      CALL_GOTOXY(50,0)
      WRITE(0,*)'  WUSRR IND(MHY)   NRL   NIM'
      DO 320 NF=1,NFT
      CALL_GOTOXY(50,NF)
      ZNRL=(XRLC(NF)-XRLA(NF))/XIMA(NF)
      ZNIM=XIMC(NF)/XIMA(NF)
      XIND=1000.*XIMA(NF)/(6283.19*FREQ(NF))
      WRITE(0,310)WUSR(NF),XIND,ZNRL,ZNIM
  310 FORMAT(F8.2,F9.4,2(F6.3))
  320 CONTINUE
      GO TO 199
  399 CONTINUE
      WRITE(0,410)GRL,ORL,GIM,OIM
  410 FORMAT('PRESENT VALUES;GRL=',1F7.0,' ORL=',1F7.0,' GIM=',1F7.0,'
     *OIM=',1F7
.0)
      READ(0,*)GRL,ORL,GIM,OIM
      GO TO 80
  999 STOP
      END
C
      SUBROUTINE GRAPH(IX1,IY1,IX2,IY2,NGR)
C
      SUBROUTINE GRID(NGR)
C
      SUBROUTINE ANARDG(XMAG,XPHA,FREQ,VOUT,LANA,NFT)
C
```

The output from a sample run is shown in Fig. 11. The numerical data to the right of the plot are obtained by the D option.

## PROGRAMS BODEPL AND BODEDF

The next two similar programs both make Bode plots on the CRT. The first program, BODEPL, reads phase and magnitude data from the impedance analyzer at different frequencies and then plots the phase and magnitude of the eddy-current probe vs the frequency. The probe can be placed in air or on various parts of a sample. The impedance analyzer is usually operated in a send-receive mode although absolute coils can also be measured. Figures 4 and 6 show the connections for the through-transmission and the reflection coil measurements.

The second program, BODEDF, is basically the same as BODEPL except that it plots the difference between the magnitude and phase for the probe in one position and then the probe in another position. Thus, the difference plot represents the change in magnitude and phase and is used to show the change caused by tube supports, presence of wall thinning, presence of a defect, or even the presence of the conductor. These programs can be very useful in determining the best frequency to perform measurements of the variation of a given property and to ignore variations in other properties. Of course, the traditional use for Bode plots has been for analysis of the frequency response of electrical circuits, and these programs are well suited for that purpose.

### INPUT DATA FOR BODEPL

The input data used for this program are given below.

XMAG – magnitude values read by the impedance analyzer

XPHA – phase values read by the impedance analyzer

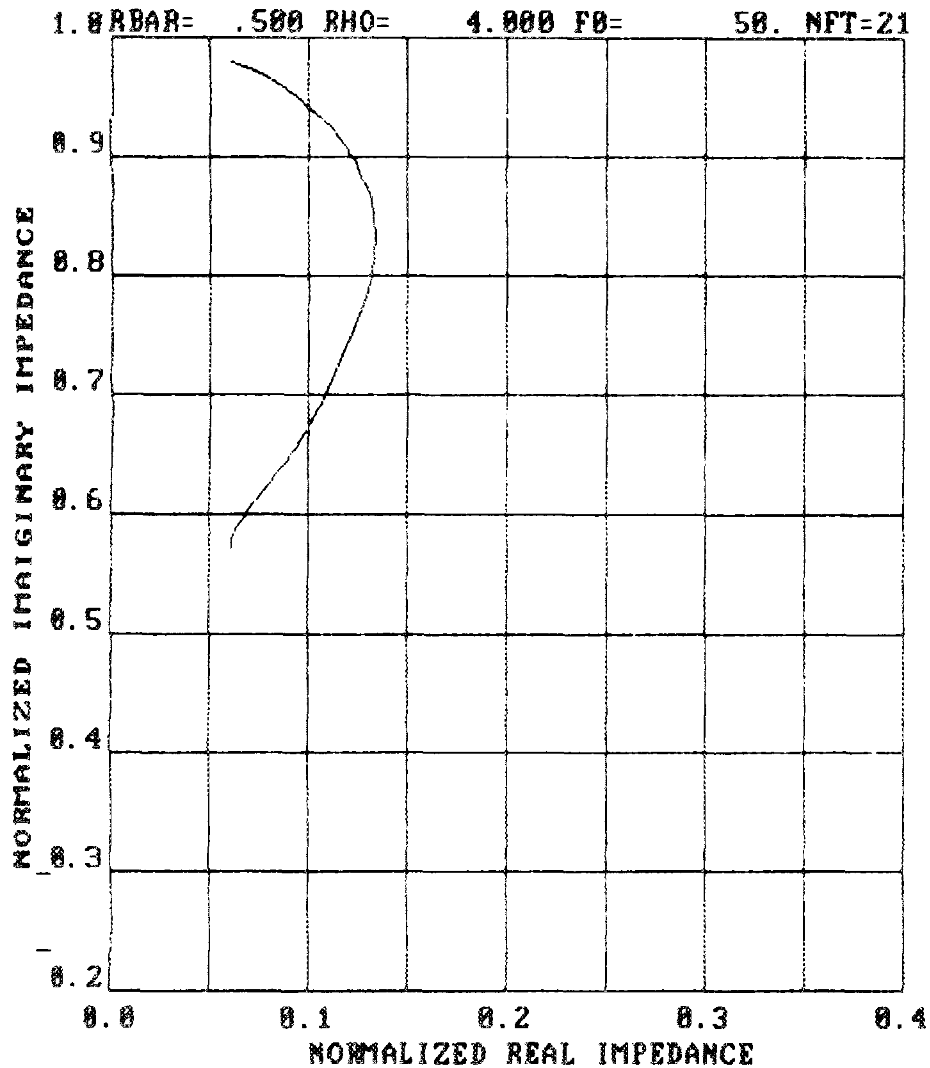
FREQ – frequencies, computed for NFT values, equally spaced on a log plot

VOUT – output voltage level array

NFT – number of frequencies; should be chosen from set listed in program

FMIN – starting frequency

ORNL-DWG 87-1829




| WUSRR | IND(MHY) | NRL | NIM |
|---|---|---|---|
| 1.28 | 8.8172 | .061 | .982 |
| 1.75 | 8.8185 | .079 | .968 |
| 2.38 | 8.8569 | .095 | .952 |
| 3.25 | 8.8723 | .112 | .927 |
| 4.44 | 8.8539 | .123 | .898 |
| 6.05 | 8.8518 | .132 | .866 |
| 8.26 | 8.8804 | .134 | .832 |
| 11.27 | 8.8661 | .133 | .800 |
| 15.37 | 8.8778 | .127 | .770 |
| 20.98 | 8.8665 | .121 | .745 |
| 28.62 | 8.8714 | .115 | .721 |
| 39.05 | 8.8671 | .109 | .698 |
| 53.28 | 8.8670 | .102 | .676 |
| 72.70 | 8.8660 | .095 | .656 |
| 99.19 | 8.8680 | .087 | .638 |
| 135.34 | 8.8686 | .079 | .623 |
| 184.67 | 8.8695 | .073 | .610 |
| 251.96 | 8.8696 | .067 | .598 |
| 343.78 | 8.8721 | .064 | .588 |
| 469.06 | 8.8755 | .062 | .580 |
| 640.00 | 8.8808 | .062 | .572 |

Fig. 11. Plot of normalized impedance of a coil by NORIMP.

FMAX - maximum frequency

VOLT - output voltage level

GMAG,OSMAG,GPH,OSPH -gain and offset of the magnitude and phase; may be varied on an interactive basis to get the best plot

CX1,CX2,CX5,CX10,MO,PO - used as labels on the Bode plots

SUBROUTINES

A list of the system subroutines is given below.

1. _GETKEY
2. _CLRWIN
3. _SETCOP
4. _VECTOR
5. _CLS
6. _CHARSTR

Only three subroutines are written in the program. ANARDG is used again, and two new routines, GRAPH and GRID, which both do what their names suggest, join it. Both are listed in the Appendix.

MAIN PROGRAM

A listing of the program BODEPL follows.

```
      PROGRAM BODEPL
C     VERSION 2 OCTOBER 1986
C     MAKES A BODE PLOT ON THE CRT FROM DATA READ FROM THE HEWLETT PACKARD
C     IMPEDANCE ANALYZER.THE MAX AND MIN FREQUENCY,FMAX & FMIN IN KHZ ;
C     THE MAGNITUDE AND PHASE GAIN,GMAG & GPH ; AND THE NUMBER OF
C     FREQUENCIES NFT MAY BE SET.  PROGRAM WILL NOT END UNTIL AN 'S' IS
C     STRUCK TO ALLOW TIME TO PRINT SCREEN.  TYPE O TO CHANGE OFFSET AND
C     GAIN OF MAG & PHASE
C     GMAG=MAGNITUDE GAIN
C     OSMG=MAGNITUDE OFFSET (IN DB)
C     GPH =PHASE GAIN
C     OSPH=PHASE OFFSET (IN DEGREES)
C
      CHARACTER*2 CX1,CX2,CX5,CX10
```

```
      CHARACTER MO*3,PO*3,TITLE*50
      DIMENSION XMAG(701),XPHA(701),FREQ(701),VOUT(701)
      DATA NFT/51/,FMIN/1000./,FMAX/13000./,VOLT/0.6/
      DATA GPH/5./,OSPH/60./,GMAG/10./,OSMG/45./
C     TITLE LIMIT<12345678901234567890123456789012345678901234567890>LIMIT
      DATA TITLE/' RESISTIVITY MEASUREMENTS WITH A REFLECTION COIL  '/
      DATA CX1/'1 '/,CX2/'2 '/,CX5/'5 '/,CX10/'10'/,MO/'0 M'/,PO/'0 P'/
      DATA LSTOP/500/,LANA/43/,NGR/39/
C     THE SCALE WILL FIT A LOT BETTER IF NFT IS CHOSEN SO THAT
C     700/NX=INTEGER
C     NFT=701,351,141,71,51,36,29,21,15,8
C     SET UP GRAPHICS FILE
      OPEN(NGR,FILE='#GR')
C
C     HEWLETT PACKARD 4192A LF IMPEDANCE ANALYZER ADDR=3 ON GPIB BUSS
C     ADDR=C? ON IBM BUSS DRIVER, FORTRAN LOGICAL UNIT = 43
      OPEN(LANA,FILE='#BUSC?')
      WRITE(0,*)'IMPEDANCE ANALYZER  FILE OPENED'
      WRITE(0,*)'TYPE S TO STOP PROG,D TO PRINT DATA,O TO CHANGE OFFSET,
     *GAIN,T TO CHANGE TITLE'
C
C WRITE SET UP VALUES FOR IMPEDANCE ANALYZER,FIRST FREQUENCY AND
C     VOLTAGE OUT
C
      DELFR=EXP(ALOG(FMAX/FMIN)/FLOAT(NFT-1))
C     SCALE THE PLOT SO WE USE ALL 700 X PIXELS, OR
C     700=SCALE*ALOG(FMAX/FMIN)
      SCALE=FLOAT(700)/ALOG(FMAX/FMIN)
      IFR=INT(ALOG10(999.*FMIN))-2
      FLM=10.**IFR
      IXFL=50+SCALE*ALOG(FLM/FMIN)
      IDX10=SCALE*ALOG(10.)
      IDX2=SCALE*ALOG(2.)
      IDX5=SCALE*ALOG(5.)
      NX=700/(NFT-1)
      VOUT(1)=VOLT
```

```
      FREQ(1)=FMIN
      DO 50 NF=2,NFT
      FREQ(NF)=FREQ(NF-1)*DELFR
      VOUT(NF)=VOUT(NF-1)
   50 CONTINUE
      WRITE(LANA,*)'A5B2H1V1T3F0'
      WRITE(LANA,81)VOUT(1),FREQ(1)
   81 FORMAT(1X,'FR',F9.3,'ENOL',F5.3,'EN')
      CALL ANARDG(XMAG,XPHA,FREQ,VOUT,LANA,NFT)
   85 CALL_CLS
C     PUT LABELS ON GRAPH AND ADD GRID LINES
      CALL GRID(NGR)
      CALL_SETCOP(NGR,130,15)
      CALL_CHARSTR(NGR,TITLE)
      GPHL=100./GPH
      GMAGL=100./GMAG
      WRITE(0,90)GMAGL,GPHL,FMIN
   90 FORMAT('     MAGNITUDE=',F5.1,' DB/DIV   PHASE=',F5.1,' DEG/DIV   F0'
     *,F8.3' KHZ')
      IM0=(0.+OSMG)*GMAG+346
      CALL_SETCOP(NGR,5,IM0)
      CALL_CHARSTR(NGR,M0)
      IP0=(0.+OSPH)*GPH +235
      CALL_SETCOP(NGR,5,IP0)
      CALL_CHARSTR(NGR,P0)
  150 CONTINUE
C     DRAW VERTICAL GRID LINES ON THE GRAPH, WRITE FREQUENCY MARKERS
      CALL GRAPH(IXFL,50,IXFL,449,NGR)
      ICH=IXFL-5
      CALL_SETCOP(NGR,ICH,30)
      IF(IXFL.LT.LSTOP)CALL_CHARSTR(NGR,CX1)
      IXI=IXFL+IDX2
      CALL GRAPH(IXI,50,IXI,449,NGR)
```

```
      ICH=IXI-5
      CALL_SETCOP(NGR,ICH,30)
      IF(IXI.LT.LSTOP)CALL_CHARSTR(NGR,CX2)
      IXI=IXFL+IDX5
      CALL GRAPH(IXI,50,IXI,449,NGR)
      ICH=IXI-5
      CALL_SETCOP(NGR,ICH,30)
      IF(IXI.LT.LSTOP)CALL_CHARSTR(NGR,CX5)
      IXFL=IXFL+1
      CALL GRAPH(IXFL,50,IXFL,449,NGR)
      IXFL=IXFL-1+IDX10
      IF(IXFL.LT.751)GO TO 150
      IX1=50
      IM1=(XMAG(1)+OSMG)*GMAG+350
      IP1=(XPHA(1)+OSPH)*GPH +250
      DO 100 NF=2,NFT
      IM2=(XMAG(NF)+OSMG)*GMAG+350
      IP2=(XPHA(NF)+OSPH)*GPH +250
      IX2=IX1+NX
      CALL GRAPH(IX1,IM1,IX2,IM2,NGR)
      CALL GRAPH(IX1,IP1,IX2,IP2,NGR)
      IP1=IP2
      IM1=IM2
      IX1=IX2
  100 CONTINUE
      LSTOP = 751
      IXFL=50+SCALE*ALOG(FLM/FMIN)
  120 CALL_GETKEY(IKY,IS,KBF1,KBF2,IRR)
      IF(IKY.EQ.68.AND.IRR.EQ.0)GO TO 199 ! PRINT DATA IF D IS TYPED
      IF(IKY.EQ.79.AND.IRR.EQ.0)GO TO 299 ! CHANGE OFFSET & GAIN IF O IS
     *                                      TYPED
      IF(IKY.EQ.83.AND.IRR.EQ.0)GO TO 999 ! STOP IF S IS TYPED
      IF(IKY.EQ.84.AND.IRR.EQ.0)GO TO 399 ! CHANGE TITLE IF T IS TYPED
      GO TO 120
C     SECTION TO PRINT OUT READINGS
```

```
  199 CONTINUE
C       PRINTER GRAPHICS FILE OPENED,LOU=6
        OPEN(6,FILE='#PR')
        WRITE(6,*)'FREQ(KHZ)   MAG(DB)  PHA(DEG)   MAG(IN)  PHA(IN)'
        DO 200 NF=1,NFT
        IM2=(XMAG(NF)+OSMG)*GMAG+350
        IP2=(XPHA(NF)+OSPH)*GPH +250
        WRITE(6,210)FREQ(NF),XMAG(NF),XPHA(NF),IM2,IP2
  200 CONTINUE
  210 FORMAT(F10.3,2X,F8.3,2X,F8.2,6X,I4,6X,I4)
        GO TO 120
  299 CONTINUE
        CALL_CLRWIN(NGR)
        WRITE(0,310)GPH,OSPH,GMAG,OSMG
  310 FORMAT('OLD PHASE; GAIN ',F7.1,' OFFSET ',F7.1,' OLD MAG; GAIN ',
     *F7.1,' OFFSET ',F7.1, 'NEW VALUES ?')
        READ(0,*)GPH,OSPH,GMAG,OSMG
        GO TO 85
  399 CONTINUE
        CALL_CLRWIN(NGR)
        WRITE(0,*)' TYPE IN NEW TITLE, UP TO 50 CHARACTERS'
        READ(0,410) TITLE
  410 FORMAT(A50)
        GO TO 85
  999 CONTINUE
        CALL_CLRWIN(NGR)
        STOP
        END

C
        SUBROUTINE GRAPH(IX1,IY1,IX2,IY2,NGR)
C
        SUBROUTINE GRID(NGR)
```

```
C
      SUBROUTINE ANARDG (XMAG, FREG, VOUT, LANA, NFT)
C
```

The plot from a sample run of BODEPL is shown in Fig. 12. The abscissa units for the magnitude and phase, along with the starting frequency, are listed at the top of the plot. The plots are not labeled by the computer but can be identified by looking at the final value on the impedance analyzer.

BODEDF

BODEDF is the same as BODEPL except that it makes two sets of readings and takes the difference between the two. The program will take one set and then pause until the operator has moved the probe to its new location. The connections, input data, and subroutines are the same as for BODEPL. Like BODEPL, BODEDF can also be used in circuit analysis to show how small changes in the circuit can change the frequency response.

MAIN PROGRAM

A listing of the program BODEDF follows.

```
      PROGRAM BODEDF
C     VERSION 1 OCTOBER 1986
C     MAKES A BODE PLOT ON THE CRT FROM DATA READ FROM THE HEWLETT PACKARD
C     IMPEDANCE ANALYZER.THE MAX AND MIN FREQUENCY,FMAX & FMIN IN KHZ ;
C     THE MAGNITUDE AND PHASE GAIN,GMAG & GPH ; AND THE NUMBER OF
C     FREQUENCIES NFT MAY BE SET.THE DIFFERENCE IN MAGNITUDE AND PHASE
C     ARE PLOTTED.PROGRAM WILL NOT WAIT UNTIL AN 'S' IS STRUCK TO ALLOW
C     TIME TO PRINT SCREEN.TYPE 0 TO CHANGE OFFSET AND GAIN OF THE
C     THE MAGNITUDE & PHASE
C     GMAG=MAGNITUDE GAIN
C     OSMG=MAGNITUDE OFFSET (IN DB)
```

ORNL-DWG 86-16217

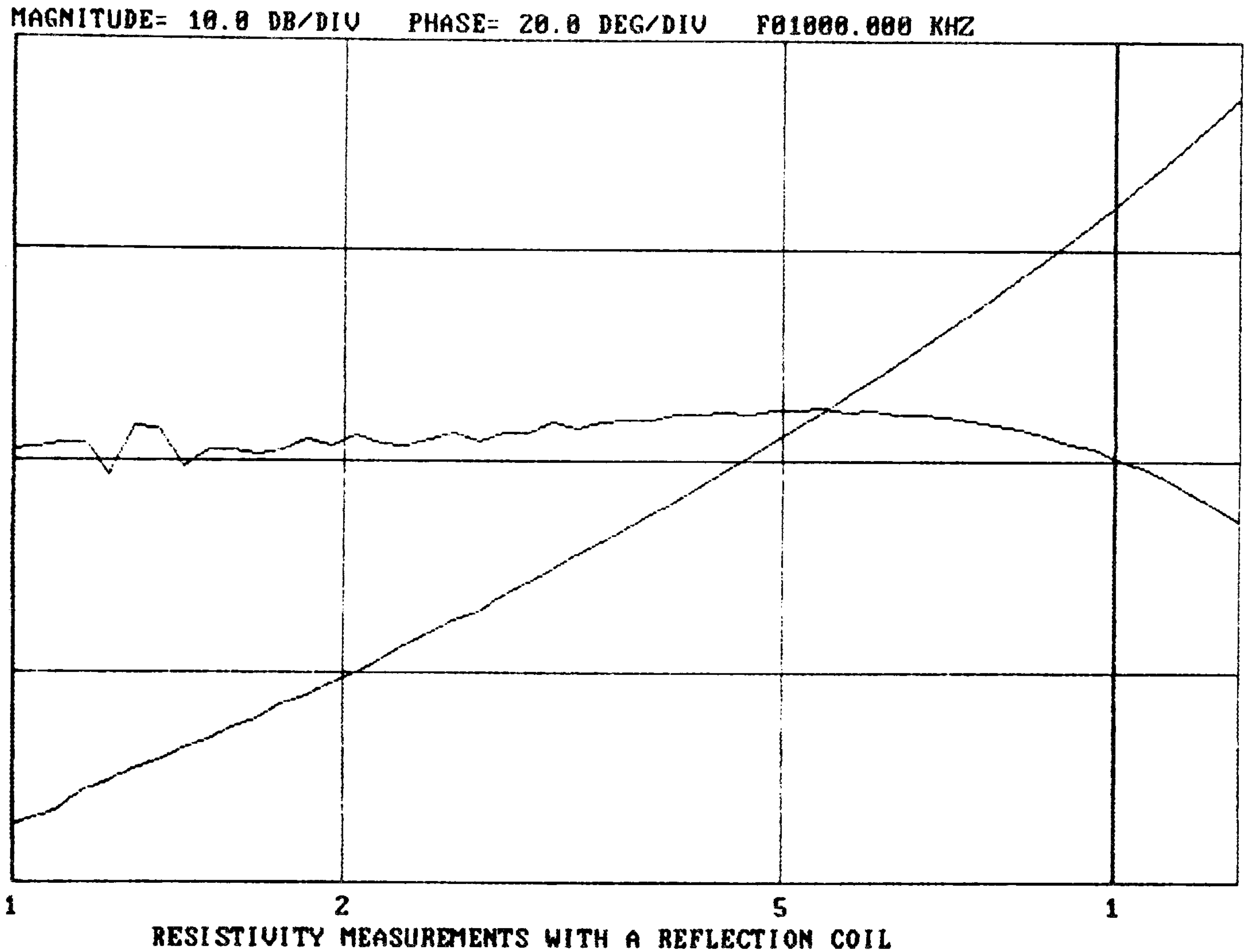



Fig. 12. Plot of magnitude and phase of an eddy-current probe by BODEPL.

```
C     GPH =PHASE GAIN
C     OSPH=PHASE OFFSET (IN DEGREES)
C
      CHARACTER*2 CX1,CX2,CX5,CX10
      CHARACTER M0*3,P0*3,TITLE*50
      DIMENSION XMAG1(701),XPHA1(701),XMAG2(701),XPHA2(701),FREQ(701),
     *VOUT(701)
      DATA NFT/51/,FMIN/1000./,FMAX/13000./,VOLT/0.6/
      DATA GPH/50./,OSPH/0.0/,GMAG/100./,OSMG/0.0/
C     TITLE LIMIT<12345678901234567890123456789012345678901234567890>LIMIT
      DATA TITLE/'CHANGE DUE TO A THICKNESS CHANGE IN THE RHO SAMPLE'/
      DATA CX1/'1 '/,CX2/'2 '/,CX5/'5 '/,CX10/'10'/,M0/'0 M'/,P0/'0 P'/
      DATA LSTOP/500/,LANA/43/,NGR/39/
C     THE SCALE WILL FIT A LOT BETTER IF NFT IS CHOSEN SO THAT
C     700/NX=INTEGER
C     NFT=701,351,141,71,51,36,29,21,15,8
C     SET UP GRAPHICS FILE
      OPEN(NGR,FILE='#GR')
C
C     HEWLETT PACKARD 4192A LF IMPEDANCE ANALYZER ADDR=3 ON GPIB BUSS
C     ADDR=C? ON IBM BUSS DRIVER, FORTRAN LOGICAL UNIT = 43
      OPEN(LANA,FILE='#BUSC?')
      WRITE(0,*)'IMPEDANCE ANALYZER  FILE OPENED'
      WRITE(0,*)'TYPE S TO STOP PROG,D TO PRINT DATA,O TO CHANGE OFFSET,
     *GAIN,T TO CHANGE TITLE'
C
C WRITE SET UP VALUES FOR IMPEDANCE ANALYZER,FIRST FREQUENCY AND
C     VOLTAGE OUT
C
      DELFR=EXP(ALOG(FMAX/FMIN)/FLOAT(NFT-1))
C     SCALE THE PLOT SO WE USE ALL 700 X PIXILS,
C     OR 700=SCALEALOG(FMAX/FMIN)
      SCALE=FLOAT(700)/ALOG(FMAX/FMIN)
      IFR=INT(ALOG10(999.*FMIN))-2
      FLM=10.**IFR
```

```
      IXFL=50+SCALE*ALOG(FLM/FMIN)
      IDX10=SCALE*ALOG(10.)
      IDX2=SCALE*ALOG(2.)
      IDX5=SCALE*ALOG(5.)
      NX=700/(NFT-1)
      VOUT(1)=VOLT
      FREQ(1)=FMIN
      DO 50 NF=2,NFT
      FREQ(NF)=FREQ(NF-1)*DELFR
      VOUT(NF)=VOUT(NF-1)
   50 CONTINUE
      WRITE(LANA,*)'A5B2H1V1T3F0'
      WRITE(LANA,81)VOUT(1),FREQ(1)
   81 FORMAT(1X,'FR',F9.3,'ENOL',F5.3,'EN')
      CALL ANARDG(XMAG1,XPHA1,FREQ,VOUT,LANA,NFT)
      WRITE(0,*)'REPOSITION PROBE,HIT ANY KEY TO GO'
   82 CALL GETKEY(IKY,IS,KBF1,KBF2,IRR)
      IF(IRR.NE.0)GO TO 82
      CALL ANARDG(XMAG2,XPHA2,FREQ,VOUT,LANA,NFT)
   85 CALL CLS
C     PUT LABLES ON GRAPH AND ADD GRID LINES
      CALL GRID(NGR)
      CALL SETCOP(NGR,130,15)
      CALL CHARSTR(NGR,TITLE)
      GPHL=100./GPH
      GMAGL=100./GMAG
      WRITE(0,90)GMAGL,GPHL,FMIN
   90 FORMAT('     MAGNITUDE=',F5.1,' DB/DIV    PHASE=',F5.1,' DEG/DIV    F0'
     *,F8.3' KHZ')
      IM0=(0.+OSMG)*GMAG+346
```

```
      CALL_SETCOP(NGR,5,IM0)
      CALL_CHARSTR(NGR,M0)
      IP0=(0.+OSPH)*GPH +235
      CALL_SETCOP(NGR,5,IP0)
      CALL_CHARSTR(NGR,P0)
  150 CONTINUE
C     DRAW VERTICAL GRID LINES ON THE GRAPH, WRITE FREQUENCY MARKERS
      CALL GRAPH(IXFL,50,IXFL,449,NGR)
      ICH=IXFL-5
      CALL_SETCOP(NGR,ICH,30)
      IF(IXFL.LT.LSTOP)CALL_CHARSTR(NGR,CX1)
      IXI=IXFL+IDX2
      CALL GRAPH(IXI,50,IXI,449,NGR)
      ICH=IXI-5
      CALL_SETCOP(NGR,ICH,30)
      IF(IXI.LT.LSTOP)CALL_CHARSTR(NGR,CX2)
      IXI=IXFL+IDX5
      CALL GRAPH(IXI,50,IXI,449,NGR)
      ICH=IXI-5
      CALL_SETCOP(NGR,ICH,30)
      IF(IXI.LT.LSTOP)CALL_CHARSTR(NGR,CX5)
      IXFL=IXFL+1
      CALL GRAPH(IXFL,50,IXFL,449,NGR)
      IXFL=IXFL-1+IDX10
      IF(IXFL.LT.751)GO TO 150
      IX1=50
      IM1=(XMAG2(1)-XMAG1(1)+OSMG)*GMAG+350
      IP1=(XPHA2(1)-XPHA1(1)+OSPH)*GPH +250
      DO 100 NF=2,NFT
      IM2=(XMAG2(NF)-XMAG1(NF)+OSMG)*GMAG+350
```

```
      IP2=(XPHA2(NF)-XPHA1(NF)+OSPH)*GPH +250
      IX2=IX1+NX
      CALL GRAPH(IX1,IM1,IX2,IM2,NGR)
      CALL GRAPH(IX1,IP1,IX2,IP2,NGR)
      IP1=IP2
      IM1=IM2
      IX1=IX2
  100 CONTINUE
      LSTOP = 751
      IXFL=50+SCALE*ALOG(FLM/FMIN)
  120 CALL_GETKEY(IKY,IS,KBF1,KBF2,IRR)
      IF(IKY.EQ.68.AND.IRR.EQ.0)GO TO 199 ! PRINT DATA IF D IS TYPED
      IF(IKY.EQ.79.AND.IRR.EQ.0)GO TO 299 ! CHANGE OFFSET & GAIN IF O IS
     *                                      TYPED
      IF(IKY.EQ.83.AND.IRR.EQ.0)GO TO 999 ! STOP IF S IS TYPED
      IF(IKY.EQ.84.AND.IRR.EQ.0)GO TO 399 ! CHANGE TITLE IF T IS TYPED
      GO TO 120
C     SECTION TO PRINT OUT READINGS
  199 CONTINUE
C     PRINTER GRAPHICS FILE OPENED,LOU=6
      OPEN(6,FILE='#PR')
      WRITE(6,*)'FREQ(KHZ)   MAG(DB)  PHA(DEG)   MAG(IN)  PHA(IN)'
      DO 200 NF=1,NFT
      IM2=(XMAG2(NF)-XMAG1(NF)+OSMG)*GMAG+350
      IP2=(XPHA2(NF)-XPHA1(NF)+OSPH)*GPH +250
      WRITE(6,210)FREQ(NF),XMAG1(NF),XPHA1(NF),IM2,IP2
  200 CONTINUE
  210 FORMAT(F10.3,2X,F8.3,2X,F8.2,6X,I4,6X,I4)
      GO TO 120
  299 CONTINUE
      CALL_CLRWIN(NGR)
      WRITE(0,310)GPH,OSPH,GMAG,OSMG
```

```
  310 FORMAT('OLD PHASE; GAIN ',F7.1,' OFFSET ',F7.1,' OLD MAG;
     *GAIN ',F7.1,' OFFSET ',F7.1, 'NEW VALUES ?')
      READ(0,*)GPH,OSPH,GMAG,OSMG
      GO TO 85
  399 CONTINUE
      CALL_CLRWIN(NGR)
      WRITE(0,*)' TYPE IN NEW TITLE, UP TO 50 CHARACTERS'
      READ(0,410) TITLE
  410 FORMAT(A50)
      GO TO 85
  999 CONTINUE
      CALL_CLRWIN(NGR)
      STOP
      END
C
      SUBROUTINE GRAPH(IX1,IY1,IX2,IY2,NGR)
C
      SUBROUTINE GRID(NGR)
C
      SUBROUTINE ANARDG(XMAG,XPHA,FREQ,VOUT,LANA,NFT)
C
```

A sample of a plot made by BODEDF is shown in Fig. 13. This plot shows the change in magnitude and phase caused by a thickness change in a resistivity sample, which is not thick enough to get an accurate resistivity measurement even at 13 MHz.

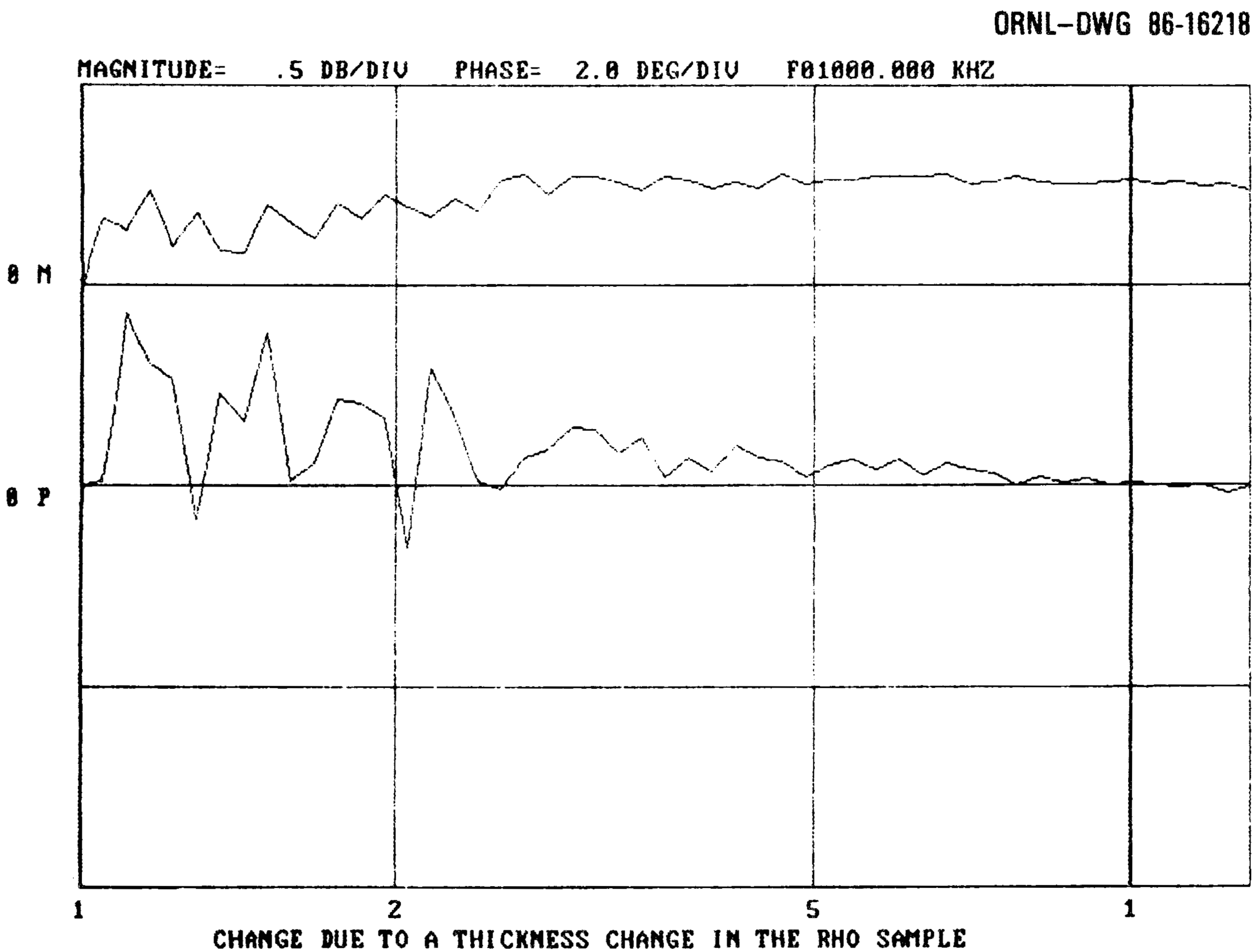



Fig. 13. Plot of the difference in magnitude and phase of an eddy-current probe by BODEDF.

## CONCLUSION

The combination of a small laboratory computer, such as the IBM System 9000, with the HP 4192A Impedance Analyzer has provided a quick, accurate, and fairly easy method of performing a number of different types of eddy-current laboratory measurements. The versatility and automation of these measurements have reduced considerably the cost of experimental measurements and allowed more complex and more accurate tests to be developed. Problems formerly considered impossible now may be solved and experimentally verified by using multiple-frequency/multiple-property methods.

## REFERENCES

1. C. V. Dodd and W. E. Deeds, "Absolute Eddy-Current Measurement of Electrical Conductivity," pp. 387–94 in *Review of Progress in Quantitative Nondestructive Evaluation*, Vol. 1, proceedings of AF/DARPA Review of Progress in Quantitative Nondestructive Evaluation, Boulder, Colorado, Aug. 2–7, 1981, ed. D. O. Thompson and D. E. Chimenti, Plenum, New York, 1982.

2. W. E. Deeds, C. V. Dodd, and G. W. Scott, *Computer-Aided Design of Multifrequency Eddy-Current Tests for Layered Conductors with Multiple Property Variations*, ORNL/TM-6858, October 1979.

3. C. V. Dodd and W. E. Deeds, "In-Service Inspection of Steam Generator Tubing Using Multiple-Frequency Eddy-Current Techniques," pp. 229–39 in *Eddy-Current Characterization of Materials and Structures*, ASTM Special Technical Publication 722, ed. G. Birnbaum and G. Free, American Society for Testing and Materials, Philadelphia, 1981.

# Appendix

## SUBROUTINES

### SUBROUTINE ALSQS (USED IN FITANA)

Subroutine ALSQS takes the READING array constructed by subroutine RGDEXP and finds the array of coefficients, COEF, that, when multiplied by the READING array, will give the best least-squares fit to the property array PRO. Upon return, the best-fitting property values are in the last column of the READING array, and RSOS is the residual sum of squares of the differences from actual property values.

```
      SUBROUTINE ALSQS(A,Y,B,R2,NN,MM,NA)
C
C     ALSQS IS A FORTRAN IV SUBROUTINE TO SOLVE THE LINEAR LEAST
C     SQUARES PROBLEM  NORM(AB - Y) = MIN.   CALLING SEQUENCE IS
C              CALL ALSQS(A,Y,B,R2,N,M,NA)
C     WHERE
C        A    IS AN ARRAY CONTAINING THE LEAST SQUARES MATRIX.
C             UPON RETURN THE (M+1)-TH COLUMN CONTAINS THE
C             APPROXIMATING VECTOR AB.
C        Y    IS THE VECTOR TO BE FIT
C        B    CONTAINS UPON RETURN THE COEFFICIENTS OF THE FIT.
C        R2   CONTAINS UPON RETURN THE RESIDUAL SUM OF SQUARES.
C        N    IS THE NUMBER OF ROWS IN THE LEAST SQUARES MATRIX.
C        M    IS THE NUMBER OF COLUMNS IN THE LEAST SQUARES
C             MATRIX.
C        NA   IS THE FIRST DIMENSION OF THE ARRAY A.
C
      DIMENSION A(NA,1),Y(1),B(1)
      N = NN
      N1 = N+1
      M = MM
      M1 = M+1
      MM1 = M-1
C
```

```
C     REDUCE THE LEAST SQUARES MATRIX TO UPPER TRIANGULAR FORM
C
      DO 60 L=1,M
         SS = 0.
         DO 10 I=L,N
   10       SS = SS + A(I,L)**2
         S2 = SS
         S =SQRT(S2)
         IF (A(L,L).LT.0.) S=-S
         D = S2 + S*A(L,L)
         A(L,L) = A(L,L) + S
         IF (L.EQ.M) GO TO 50
         L1 = L+1
         DO 30 J=L1,M
            PP = 0.
            DO 20 I=L,N
   20          PP = PP + A(I,L)*A(I,J)
   30       A(N1,J) = PP/D
         DO 40 J=L1,M
            DO 40 I=L,N
   40          A(I,J) = A(I,J) - A(I,L)*A(N1,J)
   50    A(N1,L) = -S
   60    CONTINUE
C
C     REDUCE THE VECTOR Y
C
      DO 80 I=1,N
   80    A(I,M1) = Y(I)
      DO 100 L=1,M
         PP = 0.
         DO 90 I=L,N
   90       PP = PP + A(I,L)*A(I,M1)
         D = PP/(-A(L,L)*A(N1,L))
         DO 100 I=L,N
  100       A(I,M1) = A(I,M1) - D*A(I,L)
```

```
C
C     CALCULATE THE COEFFICIENT VECTOR B
C
      B(M) = A(M,M1)/A(N1,M)
      IF (M.EQ.1) GO TO 130
      B(M) = A(M,M1)/A(N1,M)
      IF (M.EQ.1) GO TO 130
      DO 120 LL=1,MM1
         L = M-LL
         L1 = L+1
         PP = A(L,M1)
         DO 110 I=L1,M
  110       PP = PP - A(L,I)*B(I)
  120    B(L) = PP/A(N1,L)
C
C     CALCULATE R2
C
  130 SS = 0.
      MP1 = M+1
      DO 140 I=MP1,N
         SS = SS + A(I,M1)**2
  140    A(I,M1) = 0.
      R2 = SS
C
C     PERFORM THE BACK CALCULATIONS
C
      DO 170 LL=1,M
         L = M-LL+1
         PP = 0.
         DO 150 I=L,N
  150       PP = PP + A(I,L)*A(I,M1)
         D = PP/(-A(L,L)*A(N1,L))
         DO 160 I=L,N
  160       A(I,M1) = A(I,M1) - D*A(I,L)
  170    CONTINUE
      RETURN
      END
```

SUBROUTINE ANARDG (USED IN RDGANA, FITANA, PLTANA, NORIMP, BODEPL, AND BODEDF)

Subroutine ANARDG takes readings at the number of different frequencies (NFT) on the Hewlett Packard Impedance Analyzer. XMAG(NFT) is the magnitude array, and XPHA(NFT) is the phase shift array. FREQ(NFT) is the frequency values, and VOLT(NFT) is the voltages of the signals that are sent to the probe by the impedance analyzer. The output signal will not be changed if only one frequency is used. The mode of operation of the analyzer is controlled by the main program.

```
      SUBROUTINE ANARDG(XMAG,XPHA,FREQ,VOUT,LANA,NFT)
C
C     SUBROUTINE TO PERFORM MULTIFREQUENCY READINGS USING HP IMPEDANCE
C     ANALYZER
C
      CHARACTER CHR1*4,CHR2*5
      DIMENSION XMAG(NFT),XPHA(NFT),FREQ(NFT),VOUT(NFT)
      DO 100 NF=1,NFT
      IF (NFT.GT.1)WRITE(LANA,81)VOUT(NF),FREQ(NF)
   81 FORMAT(1X,'OL',F5.3,'ENFR',F9.3,'EN')
      WRITE(LANA,*)'EX'
      READ(LANA,83)CHR1,XMAG(NF),CHR2,XPHA(NF)
  100 CONTINUE
   83 FORMAT(A4,E11.2,A5,E11.2)
      RETURN
      END
```

SUBROUTINE BESSEL (USED IN CONDTF)

Subroutine BESSEL computes the integral of x*J1(x), where J1(x) is the Bessel function of the first kind and order 1 from 0 to x. A detailed analysis of this procedure and the associated errors are given elsewhere.[1] The error is less than 3.E-11 from 0 to 5 and 3.E-6 from 5 to infinity. Most of the value of the numerical integration has been accumulated by the time x reaches 5.

```
      SUBROUTINE BESSEL(XJ1,X,R)
C
C       COMPUTES THE INTEGRAL OF THE BESSEL FUNCTION J1(X)
C       TIMES X FROM ZERO TO Z. USES POWER SERIES SUMMATION FOR
C       Z .LE. 5, AND AN ASYMPTOTIC SERIES FOR Z .GT. 5.
C       SEE ORNL-TM-3295, PP. 258-261.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DATA PIO4 /.785398163/
      Z=X*R
      IF (Z .GT. 5.0) GO TO 1010
      K = Z + Z
      L = K + 3
      F1 = 0.5*R*R*R
      XJ1 = F1/3.0
      T = -.25*Z*Z
      D1 = 2.
      D2 = 5.
      E = 4.
      DO 1000 I=1,L
        F1 = F1*T/D1
        XJ1 = XJ1 + F1/D2
        D1 = D1 + E
        E = E + 2.
        D2 = D2 + 2.
 1000 CONTINUE
      GO TO 1020
C
C       ASYMPTOTIC SERIES FOR Z .GT. 5
C
 1010 T = 1./Z
      X0=(((((-188.1357*T + 109.1142)*T - 23.79333)*T + 2.050931)*T
     1     - 0.1730503)*T + 0.7034845)*T - 0.064109E-3
```

```
      X1 = ((((-5.817517*T + 2.105874)*T - .6896196)*T +.4952024)*T
     1    - 0.187344E-2)*T + 0.7979095
      ARG = Z - PI04
      XJ1 = (1. - DSQRT(Z)*(X1*DCOS(ARG) - X0*DSIN(ARG)))/(X**3)
 1020 RETURN
      END
```

SUBROUTINE CORRDG (USED IN PLTANA)

This short subroutine corrects the voltages to compensate for the gain and offset, which are input to the data statements or computed by a reference to a standard.

```
      SUBROUTINE CORRDG(VOLTS,COFSET,GAIN,NCHS)
      DIMENSION VOLTS(NCHS),COFSET(NCHS),GAIN(NCHS)
      DO 100 NCH=1,NCHS
      VOLTS(NCH)=COFSET(NCH)+GAIN(NCH)*VOLTS(NCH)
  100 CONTINUE
      RETURN
      END
```

SUBROUTINE GRAPH (USED IN NORIMP, BODEPL, AND BODEDF)

Subroutine GRAPH draws a line on the CRT from (IX1,IY1) to (IX2,IY2) by using two system subroutines, _SETCOP and _VECTOR. The subroutine _SETCOP sets the current operating point at the coordinates IX1 and IY1, and _VECTOR draws a straight line from the current operating point to the new coordinates IX2 and IY2.

```
      SUBROUTINE GRAPH(IX1,IY1,IX2,IY2,NGR)
C     PROGRAM TO DRAW A LINE IN FORTRAN
      CALL_SETCOP(NGR,IX1,IY1)
      CALL_VECTOR(NGR,IX2,IY2)
      RETURN
      END
```

SUBROUTINE GRAFRT (USED IN PLTANA)

Through the PRINTER DRIVER this subroutine prints a graph directly onto the printer. Please see *Interface Library Reference Manual*, Vol. 2, Chap. 14, and *The CSOS Operating System Programmer's Guide*, Chap. 14. With the data stored in integer form in the IY array, this subroutine will plot a number (NCUR) of different curves. The printer will advance 1/84 in. for every call to the subroutine, but actual printing is not done until the pixel array, GRAPH – 800 × 8 bits, or 800 bytes, wide – is full. This requires seven calls to the subroutine for one sweep of the printer. The graphics mode is mode 1 for 800 pixel resolution or mode 2 for 1600 pixel resolution. The printer must be set back to mode 0 for alphanumeric output.

```
      SUBROUTINE GRAPRT(IY,IXX,NCUR,LOU,NTIM)
C     PLOT A GRAPH ON PRINTER,EACH CALL WILL ADVANCE 1/84 OF INCH
C     IXX IS A COUNT OF THE NUMBER OF TIMES THAT SUBROUTINE IS CALLED
C     PRINTING IS DONE EVERY 7TH TIME SUBROUTINE IS CALLED.
C     Y VALUES OF THE CURVES MUST BE SCALED FROM 1 TO 800
      INTEGER*1 GRAPH(800),GRD1(800),GRD2(800)
      INTEGER*2 BIT(7)
      DIMENSION IBUF(10),IY(NCUR),IOY(40)
      DATA IOY/40*-1000/
      DATA GRD1/800*0/,GRD2/800*1/
      DATA BIT/1,2,4,8,16,32,64/,IMAX/800/,IMIN/800/,IX/0/,IBUF/10*0/,NA/0/
      IF(NTIM.GT.0)GO TO 20
      CALL_SETPRMODE(LOU,1)
C     SET COLOR  CALL;0=RED,1=GREEN,2=BLUE,3=BLACK(DEFAULT)
C     CALL_SETCOLOR(LOU,2)
C     SET UP GRID
      GRD1(1)=127
      GRD2(1)=85
      DO 5,I=100,800,100
      GRD1(I)=85
      GRD2(I)=85
```

```
  5 CONTINUE
    NTIM=1
    DO 10 I=1,IMAX
    GRAPH(I)=GRD2(I)
 10 CONTINUE
 20 CONTINUE
    IX=IX+1
    DO 30 NC=1,NCUR
    IF(IY(NC).GT.799)IY(NC)=799
    IF(IY(NC).LT.1)IY(NC)=1
    GRAPH(IY(NC))=GRAPH(IY(NC))+BIT(IX)
    GRAPH(IY(NC)+1)=GRAPH(IY(NC)+1)+BIT(IX)
 30 CONTINUE
    IF(IX.LT.7)GO TO 200
    IOFS=0
    CALL_SWRITE(LOU,GRAPH(1),IOFS,IMAX,IMIN,IMAX,0,IRR)
    IF(IRR.NE.0)WRITE(0,*)'ERROR= ',IRR
    IX=0
    IX1=IXX-7
    IF(MOD(IXX,84).EQ.0)GO TO 80
    DO 70 I=1,IMAX
    GRAPH(I)=GRD1(I)
 70 CONTINUE
    GO TO 100
 80 DO 90 I=1,IMAX
    GRAPH(I)=GRD2(I)
 90 CONTINUE
100 CONTINUE
200 IXX=IXX+1
    RETURN
    END
```

SUBROUTINE GRID (USED IN NORIMP, BODEPL, AND BODEDF)

This subroutine draws a grid on the screen so that different plots can be made. The numbers in the GRAPH calls are the coordinates of the points. See the subroutine GRAPH for a more detailed explanation. Because the programs BODEPL and BODEDF use a different grid than NORIMP, the programs use different versions of the GRID subroutine. The version for BODEPL and BODEDF follows.

```
SUBROUTINE GRID(NGR)
CALL GRAPH(50,50,750,50,NGR)
CALL GRAPH(50,150,750,150,NGR)
CALL GRAPH(50,250,750,250,NGR)
CALL GRAPH(50,350,750,350,NGR)
CALL GRAPH(50,449,750,449,NGR)
CALL GRAPH(50,50,50,449,NGR)
CALL GRAPH(750,50,750,449,NGR)
RETURN
END
```

The version of GRID for NORIMP now follows:

```
SUBROUTINE GRID(NGR)
CALL GRAPH(50,50,450,50,NGR)
CALL GRAPH(50,100,450,100,NGR)
CALL GRAPH(50,150,450,150,NGR)
CALL GRAPH(50,200,450,200,NGR)
CALL GRAPH(50,250,450,250,NGR)
CALL GRAPH(50,300,450,300,NGR)
CALL GRAPH(50,350,450,350,NGR)
CALL GRAPH(50,400,450,400,NGR)
CALL GRAPH(50,450,450,450,NGR)
CALL GRAPH(50,50,50,450,NGR)
CALL GRAPH(100,50,100,450,NGR)
CALL GRAPH(150,50,150,450,NGR)
CALL GRAPH(200,50,200,450,NGR)
CALL GRAPH(250,50,250,450,NGR)
CALL GRAPH(300,50,300,450,NGR)
```

```
      CALL GRAPH(350,50,350,450,NGR)
      CALL GRAPH(400,50,400,450,NGR)
      CALL GRAPH(450,50,450,450,NGR)
      RETURN
      END
```

SUBROUTINE POLTYP (USED IN FITANA)

Subroutine POLTYP constructs the POLARY array for printing a representation of the polynomial expansion. This array names the cross terms and coefficients. An example of this output follows the subroutine.

```
      SUBROUTINE POLTYP(POLARY,JROW,MROW,NPOL,IROW,NFT,NC,IOFSET,IDEV)
C
C     THIS SUBROUTINE CONSTRUCTS AN ARRAY FOR PRINTING
C
      CHARACTER*2 RDGTYP(2),INUM(9),IDEG(12)
      CHARACTER*4 FUNTYP(4),BLANKS,CONSTA(2),POLARY(JROW,5),ROLD(4)
      DIMENSION NPOL(IROW,NFT)
      DATA  RDGTYP/' M',' P'/,CONSTA/'CONS','TANT'/
      DATA FUNTYP/'(LIN','(LOG','(EXP','(INV'/
      DATA INUM/'1)','2)','3)','4)','5)','6)','7)','8)','9)'/
      DATA IDEG/'1 ','2 ','3 ','4 ','5 ','6 ','7 ','8 ','9 ','10','11','12'/
      DATA BLANKS/'    '/
C
C     BLANK OUT POLARY ARRAY
C
      DO 71 J=1,5
      DO 71 I=1,JROW
71    POLARY(I,J)=BLANKS
C
      NROW=1
      IF(IOFSET.NE.1) GO TO 70
      POLARY(NROW,1)=CONSTA(1)
```

```
      POLARY(NROW,2)=CONSTA(2)
      NROW=NROW+1
C
70    DO 200 NF=1,NFT
      DO 300 NTYP=1,NC
      NCC=NTYP*3
      NCP=NCC-1
      NCF=NCP-1
      ROLD(1)=POLARY(NROW-1,1)
      ROLD(2)=POLARY(NROW-1,2)
      NDEG=NPOL(NCP,NF)
      IF(NDEG.EQ.0) GO TO 300
      DO 400 I=1,NDEG
      POLARY(NROW,1)=FUNTYP(NPOL(NCF,NF))
      POLARY(NROW,2)=RDGTYP(NTYP)//INUM(NF)
      POLARY(NROW,3)=IDEG(I)
      NROW=NROW+1
400   CONTINUE
C
C    CREATE CROSS TERMS
C
      NCTERM=NPOL(NCC,NF)
      IF(NCTERM.EQ.0) GO TO 300
      IF(NF.EQ.1.AND.NTYP.EQ.1) GO TO 99
      J=NCTERM
      DO 500 I=1,NCTERM
      POLARY(NROW,1)=POLARY(NROW-1,1)
      POLARY(NROW,2)=POLARY(NROW-1,2)
      POLARY(NROW,3)=ROLD(1)
      POLARY(NROW,4)=ROLD(2)
      POLARY(NROW,5)=IDEG(J)
      J=J-1
      NROW=NROW+1
```

```
500   CONTINUE
300   CONTINUE
200   CONTINUE
C
C  PRINT RESULTS
C
      WRITE(IDEV,21)((POLARY(I,J),J=1,5),I=1,MROW)
21    FORMAT(' POLARY='/(1X,5A4))
      GO TO 1000
C
C  PRINT ERROR MESSAGES
C
99    WRITE(IDEV,31)
31    FORMAT(' ERROR: CANNOT HAVE A CROSS TERM ON 1ST ITERATION')
C
1000  RETURN
      END
```

SUBROUTINE POSIT (USED IN RDGANA AND PLTANA)

This subroutine is used with the Modulynx Mechanical Scanner. It is used in the program RDGANA, unless the measurements are taken manually, and it is used in PLTANA. NAXIS determines the number of axes over which the scanner will move; XNEW is the new location, NOLD is the number of steps from 0 to the prior position, and MBUS is the logical unit number for the Modulynx Mechanical Scanner.

```
      SUBROUTINE POSIT(XNEW,NOLD,NAXIS,MBUS)
C
C     DETERMINES THE NEW LOCATION FOR 3 DIFFERENT POSITIONERS AND FROM
C     THE OLD LOCATION,THE NUMBER OF STEPS NEEDED TO REACH THE NEW
C     LOCATION.THEN SENDS THE PULSES OVER THE IEEE-488 BUSS TO THE
     *CONTROLLER.
C     AFTER THE NEW POSITION IS REACHED THE PULSE COUNT IS COMPARED TO THE
C     ENCODER COUNT AND CORRECTED IF DIFFERENT.
C
```

```
      DIMENSION NOLD(3),XNEW(3),NSTEPS(3)
      CHARACTER*2 AXC(3),GC
      DATA AXC/'XM','YM','ZM'/,GC/'G '/,STEPSZ/0.00025/
      DO 20 IAXIS=1,NAXIS
      NSTP=(XNEW(IAXIS)+0.000125)/STEPSZ
      NSTEPS(IAXIS)=NSTP-NOLD(IAXIS)
      NOLD(IAXIS)=NSTP
   20 CONTINUE
       WRITE(MBUS,30,ERR=40)(AXC(IAXIS),NSTEPS(IAXIS),GC,IAXIS=1,NAXIS)
       WRITE(MBUS,*,ERR=50)';<CFA>'
       WRITE(MBUS,*,ERR=60)';'
      GO TO 70
   30 FORMAT(4(A2,I7,A2))
   40 WRITE(0,*)'ERROR,UNABLE TO WRITE POSITION DATA'
      GO TO 70
   50 WRITE(0,*)'ERROR,UNABLE TO WRITE HOME COMMAND'
      GO TO 70
   60 WRITE(0,*)'ERROR,WAIT TOO LONG FOR AXIS READY COMMAND'
   70 RETURN
      END
```

SUBROUTINE RDGEXP (USED IN FITANA AND PLTANA)

This subroutine will expand the magnitude and phase readings into polynomials, which can be in terms of linear, exponential, log, or inverse functions of the magnitude and phase. The integer array variable, NPOL(6,NFT), contains the information that instructs RDGEXP to perform a particular type of expansion. The first integer determines the function type, the second the polynomial degree for that particular function, and the third the number of cross terms between the magnitude and phase. IOFSET is 1 if there is a constant or offset in the polynomial, 0 otherwise. TMDFT and PHDFT are arrays that contain the magnitude and phase drift and are used to calculate new READNG values as one magnitude or one phase is varied. The calculated drift is the worst-case sum of each of the individual drifts.

```
      SUBROUTINE RDGEXP(READNG,TMAG,PHASE,NPOL,IOFSET,TMDFT,PHDFT,MSET1
     1,IRDPR1,M,NFT,NL,NLT,NP,NPTT)
C     REAL*8 READNG,RDG
      DIMENSION READNG(MSET1,IRDPR1),NPOL(6,NFT),TMDFT(NFT),PHDFT(NFT)
      DIMENSION TMAG(NLT,NPTT,NFT),PHASE(NLT,NPTT,NFT)
C
C     NPOL CONTAINS A NUMBER FOR THE FUNCTION TYPE, THE POLYNOMIAL
C     DEGREE, AND THE NUMBER OF CROSS TERMS
C     FOR THE MAGNITUDE AND PHASE AT EACH FREQUENCY,STORED AS NPOL
C     (NF;1-MAG FUN, 2-MAG POL,3-MAG #CROSS TERMS,4-PH FUN,5-PH POL,
C     6-PH # CROSS TERMS).  IF IOFSET=0, NO OFF-SET
C     WILL BE INCLUDED,=1 OFF-SET IS INCLUDED.THE VALUES OF TMDFT(NF)&
C     PHDFT(NF) GIVE THE AMOUNT OF DRIFT IN THE MAGNITUDE AND PHASE.IF
C      NPOL(NCP,NF) =0,THAT PARTICULAR MAGNITUDE AND PHASE FOR THAT
C     FREQUENCY WILL BE SKIPPED.
C
      READNG(M,1)=1.
      N=1
      IF(IOFSET.EQ.1)N=2
      DO 210 NF=1,NFT
      DO 200 NC=1,2
      NCC=NC*3
      NCP=NCC-1
      NCF=NCP-1
      ROLD=RDG
      IF(NPOL(NCP,NF).EQ.0) GO TO 200
      IF(NC.EQ.1) RDG=TMAG(NL,NP,NF)+TMDFT(NF)
      IF(NC.EQ.2) RDG=PHASE(NL,NP,NF)+PHDFT(NF)
C
C     THE TYPE OF FUNCTION IS SELECTED
C
      IF(NPOL(NCF,NF).EQ.1)RDG=RDG
      IF(NPOL(NCF,NF).EQ.2)RDG=ALOG(RDG)
```

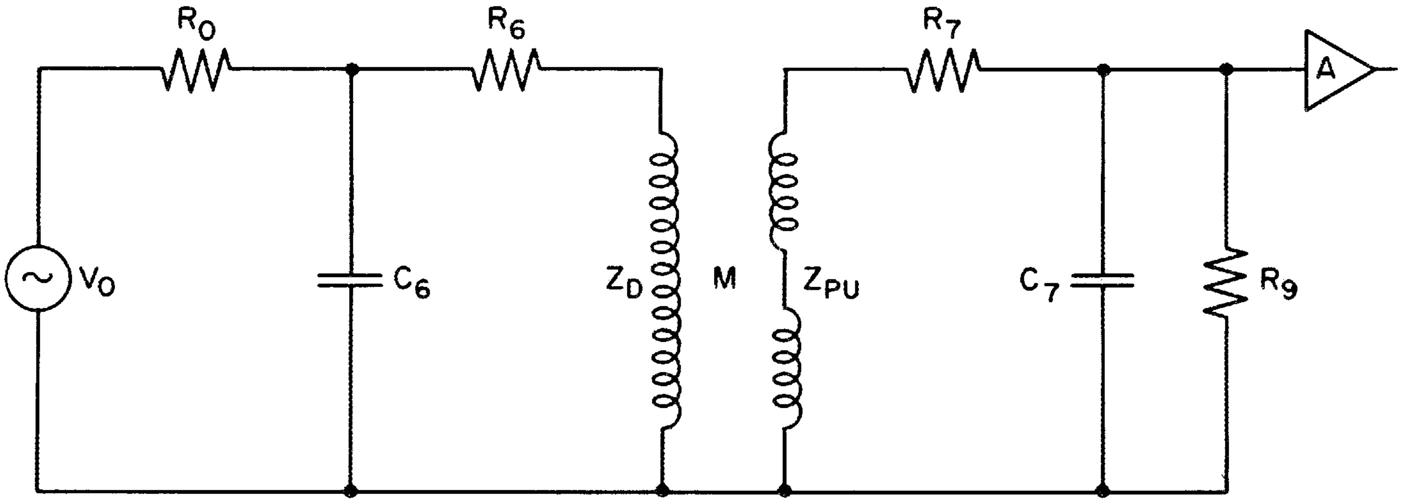
```
      IF(NPOL(NCF,NF).EQ.3)RDG=EXP(RDG)
      IF(NPOL(NCF,NF).EQ.4)RDG=1./RDG
C
C     THE TYPE OF POLYNOMIAL IS SELECTED
C     AND THE POLYNOMIAL VALUES ARE CONSTRUCTED.
C
      READNG(M,N)=RDG
      N=N+1
      NDEG=NPOL(NCP,NF)-1
      IF(NDEG.LT.1) GO TO 15
      DO 10 I=1,NDEG
      READNG(M,N)=RDG*READNG(M,N-1)
      N=N+1
   10 CONTINUE
C
C     CROSS TERMS ARE CONSTRUCTED
C
15    IF(NPOL(NCC,NF).EQ.0) GO TO 200
      RDY=ROLD
      NCTERM=NPOL(NCC,NF)
      DO 20 I=1,NCTERM
      RDY=ROLD*RDY
   20 CONTINUE
      IF(RDY.NE.0) RINV=RDG/ROLD
      IF(RDY.EQ.0) RINV=0.
      DO 30 I=1,NCTERM
      RDY=RDY*RINV
      READNG(M,N)=RDY
      N=N+1
   30 CONTINUE
C
  200 CONTINUE
  210 CONTINUE
      RETURN
      END
```

SUBROUTINE RFLCKT (USED IN CONDTF)

The subroutine RFLCKT is a circuit analysis subroutine for reflection or through-transmission coils. The circuit diagram is shown in Fig. A.1, with $Z_{PU}$ being the impedance of either a single or dual receiving coil. A detailed analysis of this circuit is found in ref. A.2. The magnitude and phase of the probe output voltage are calculated for the probe in the circuit shown.

```
      SUBROUTINE RFLCKT(TMAG,PHASE,RAD,TMUTRE,TMUTIM,DRIVRE,DRIVIM
     1,PICKRE,PICKIM,GAIN,RBAR,V0,RDCDR,RDCPU,R0,R9,CAPDR,CAPPU
     2,T1,T2,W,AIR1,AIR2)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION TMUTRE(4),TMUTIM(4)
      DIMENSION DRIVRE(4),DRIVIM(4)
      DIMENSION PICKRE(4),PICKIM(4)
      DIMENSION TMAG(4),PHASE(4)
      COLFAC=1.0027518E-7*RBAR
      DVRF=COLFAC*T1*T1*W
      PICF=COLFAC*T2*T2*W
      ZMUTF=COLFAC*T1*T2*W
      QT = V0*R9*GAIN
      X1 = W*R0*CAPDR
      X2 = W*R9*CAPPU
      Z1Z2RE = X1*X2 - 1.
      Z1Z2IM=-X1-X2
      DO 60  NP=1,4
          ZMUR = ZMUTF*TMUTRE(NP)
          ZMUI = ZMUTF*TMUTIM(NP)
          ZDRR = -DVRF*DRIVIM(NP)
         ZDRI=DVRF*(DRIVRE(NP)+AIR1)
          ZPUR = -PICF*PICKIM(NP)
          ZPUI=PICF*(PICKRE(NP)+AIR2)
          ZPR = ZDRR + RDCDR
          C1 = ZPR*X1 + ZDRI
```

ORNL-DWG 72-671




$V_O$ DRIVING VOLTAGE
$R_O$ SERIES RESISTANCE IN THE DRIVING CIRCUIT
$C_6$ SHUNT CAPACITANCE OF THE DRIVING CIRCUIT
$R_6$ D.C. RESISTANCE OF THE DRIVER COIL
$Z_D$ IMPEDANCE OF THE DRIVER COIL
M MUTUAL IMPEDANCE BETWEEN THE DRIVER AND PICK-UP COILS
$Z_{PU}$ IMPEDANCE OF THE PICK-UP COILS
$R_7$ D.C. RESISTANCE OF THE PICK-UP COILS
$C_7$ SHUNT CAPACITANCE OF THE PICK-UP CIRCUIT
$R_9$ AMPLIFIER INPUT IMPEDANCE

Fig. A.1. Instrument circuit diagram for both reflection and through-transmission probes.

```
      C2 = X1*ZDRI - R0 - ZPR
      ZPR = ZPUR + RDCPU
      C3 = ZPR*X2 + ZPUI
      C4 = X2*ZPUI - R9 - ZPR
      Z5SQRE = (ZMUR + ZMUI)*(ZMUR - ZMUI)
      Z5SQIM=2*ZMUR*ZMUI
      DENRE = Z1Z2RE*Z5SQRE - Z1Z2IM*Z5SQIM + C1*C3 - C2*C4
      DENIM = Z1Z2RE*Z5SQIM + Z1Z2IM*Z5SQRE + C1*C4 + C2*C3
      TNMRE = QT*ZMUI
      TNMIM = -QT*ZMUR
      TMAG(NP) =
     1  DSQRT((TNMRE**2 + TNMIM**2)/(DENRE**2 + DENIM**2))
      PHASE(NP) =RAD*(
     1  DATAN2(TNMIM*DENRE - TNMRE*DENIM,
     2        TNMRE*DENRE + TNMIM*DENIM))
   60 CONTINUE
      RETURN
      END
```

REFERENCES

1. W. A. Simpson et al., *Computer Programs for Some Eddy-Current Problems — 1970*, ORNL/TM-3295, June 1971.

2. W. E. Deeds, C. V. Dodd, and G. W. Scott, *Computer-Aided Design of Multifrequency Eddy-Current Tests for Layered Conductors with Multiple Property Variations*, ORNL/TM-6858, October 1979.

ORNL-6366
Distribution
Categories UC-23,
UC-25, UC-38

## INTERNAL DISTRIBUTION

1–2. Central Research Library
3. Document Reference Section
4–5. Laboratory Records Department
6. Laboratory Records, ORNL RC
7. ORNL Patent Section
8. A. Bleier
9. R. A. Buhl
10. D. F. Craig
11–15. W. E. Deeds
16–20. C. V. Dodd
21. J. P. Forester
22. R. S. Graves
23. R. W. McClung
24. R. K. Nanstad
25. R. E. Pawel
26. G. M. Slaughter
27. J. H. Smith
28–30. P. T. Thornton
31. J. R. Weir
32–36. L. M. Whitaker
37. G. Y. Chin (Consultant)
38. H. E. Cook (Consultant)
39. F. F. Lange (Consultant)
40. T. E. Mitchell (Consultant)
41. W. D. Nix (Consultant)
42. J. C. Williams (Consultant)

## EXTERNAL DISTRIBUTION

43. ARGONNE NATIONAL LABORATORY, 9700 S. Cass Avenue, Argonne, IL 60439

    O. K. Chopra

44. BABCOCK AND WILCOX COMPANY, Fossil Power Generation Division, 20 South Van Buren Avenue, Barberton, OH 44203

    M. Gold

45. BURNS AND ROE, INC., 700 Kinderkamack Road, Oradell, NJ 07469

    C. S. Ehrman

46. COMBUSTION ENGINEERING, INC., 911 W. Main Street, Chattanooga, TN 37402

    D. A. Canonico

47–48. COMBUSTION ENGINEERING, INC., 1000 Prospect Hill Road, Windsor, CT 06095

    P. E. C. Bryant
    A. L. Gaines

49. ELECTRIC POWER RESEARCH INSTITUTE, 3412 Hillview Avenue, P.O. Box 10412, Palo Alto, CA 94303

    R. I. Jaffee

50. ELECTRIC POWER RESEARCH INSTITUTE-CONSOLIDATED MANAGEMENT ORGANIZATION, Suite 220, One Energy Center, Naperville, IL 60566

    D. R. Riley

51. FOSTER WHEELER DEVELOPMENT CORPORATION, 12 Peach Tree Hill Road, Livingston, NJ 07039

    W. R. Apblett

52. GA TECHNOLOGIES, INC., P.O. Box 86508, San Diego, CA 92138

    D. I. Roberts

53–54. GENERAL ELECTRIC COMPANY, Nuclear Systems Technology Operations, 310 DeGuigne Drive, P.O. Box 3508, Sunnyvale, CA 94088

    P. J. Ring
    P. Roy

55. HANFORD ENGINEERING DEVELOPMENT LABORATORY, P.O. Box 1970, Richland, WA 99352

    L. D. Blackburn

56. MARTIN MARIETTA AEROSPACE, Department 3773, Michoud Division, P.O. Box 29304, New Orleans, LA 70189

    G. W. Scott

57–58. ROCKWELL INTERNATIONAL, Rocketdyne Division, 6633 Canoga Avenue, Canoga Park, CA 91304

    T. L. Anderson
    W. T. Lee

59. ROCKWELL INTERNATIONAL, Energy Technology Engineering Group, P.O. Box 1449, Canoga, Park, CA 91304

    H. C. Wieseneck

60. WESTINGHOUSE ELECTRIC CORPORATION, Advanced Energy Systems Division, P.O. Box 10864, Pittsburgh, PA 15236

    R. W. Buckman

61–64. DOE, Washington, DC 20545

Office of Fusion Energy
T. C. Reuther

Office of Reactor Systems Development and Technology
C. C. Bigelow
R. J. Neuhold
B. J. Rock

65. DOE, OAK RIDGE OPERATIONS OFFICE, P.O. Box E, Oak Ridge, TN 37831

Office of Assistant Manager for Energy Research and Development

66–299. DOE, TECHNICAL INFORMATION CENTER, Office of Information Services
P.O. Box 62, Oak Ridge, TN 37831

For distribution as shown in DOE/TIC-4500, Distribution Categories UC-23 (Radioisotope and Radiation Applications), UC-25 (Materials), and UC-38 (Engineering and Equipment).

☆U.S. GOVERNMENT PRINTING OFFICE 1987—748-168/60004