

3



Naval Research Laboratory

Washington, DC 20375-5000

NRL Memorandum Report 6559

Date: 15 November 1989

A TEST METHOD FOR MEASURING CORONA INCEPTION VOLTAGE FOR TRANSDUCER AUTOTRANSFORMERS

AD-A216 818

D.L. Diebel
and
A.C. Tims

Underwater Sound Reference Detachment
Naval Research Laboratory
P.O. Box 568337
Orlando, Florida 32856-8337

FILED
JAN 19 1990
8

9 0 0 1 1 8 0 0 4

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 15 November 1989	3. REPORT TYPE AND DATES COVERED FINAL - FY87/FY88	
4. TITLE AND SUBTITLE A TEST METHOD FOR MEASURING CORONA INCEPTION VOLTAGE FOR TRANSDUCER AUTOTRANSFORMERS			5. FUNDING NUMBERS WU - 59-0584	
6. AUTHOR(S) DEAN L. DIEBEL AND A.C. TIMS				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NAVAL RESEARCH LABORATORY UNDERWATER SOUND REFERENCE DETACHMENT P.O. BOX 568337 ORLANDO, FL 32856-8337			8. PERFORMING ORGANIZATION REPORT NUMBER NRL MEMORANDUM REPORT 6559	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) NAVAL SEA SYSTEMS COMMAND WASHINGTON, DC 20362-5101			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) An autotransformer corona inception test circuit has been developed and a prototype has been constructed and demonstrated. Circuit diagrams and component values are presented. Ancillary equipment for the test is identified and recommended, test procedures and precautions are presented, and measured corona inception voltage data as well as pictorial data are presented for autotransformers of different manufacture and type. Included is a computer program which is used to model and calculate values of test circuit filter components. <i>from records</i>				
14. SUBJECT TERMS Corona Transformers Autotransformer Transducer reliability (electroacoustic),			15. NUMBER OF PAGES 38	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

BLANK PAGE

Contents

INTRODUCTION..... 1
BACKGROUND 1
APPROACH AND OBJECTIVES..... 2
CIV TEST CIRCUIT DESCRIPTION..... 3
CIV TEST PROCEDURES.....10
CIV TEST PRECAUTIONS.....13
CIV TEST DATA.....13
CONCLUSIONS.....15
SUMMARY.....16
ACKNOWLEDGMENTS.....16
REFERENCES.....16
APPENDIX - Ladder Computer Program for EMI Filter.....17



Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Avail. and/or	
Dist. Statement	
A-1	

**A TEST METHOD FOR MEASURING CORONA INCEPTION VOLTAGE
FOR TRANSDUCER AUTOTRANSFORMERS**

INTRODUCTION

A small but significant percentage of failures in high-power fleet sonar transducers are due to breakdown of electrical insulating or dielectric materials. A physical phenomenon that contributes to these failures is corona which is a discharge of electricity caused by ionization of the surrounding medium when the voltage gradient exceeds a certain critical value. The frequency of discharge is above 75 kHz. At high operating voltages corona occurs before dielectric breakdown, but will, in time, deteriorate insulating materials and cause dielectric failure.

One of the components in a typical high-power transducer that is susceptible to corona is the autotransformer (tapped power inductor) which is used to augment tuning, transmit voltage response, and transmit impedance. These transformers usually operate at secondary voltages in the range of 1500 V, but may be as high as 5000 to 6000 V in some instances. A suitable corona test is valuable in determining if a particular transformer design or production transformer has corona at relatively low voltages. Such a situation may be indicative of poor terminal design, voids or air pockets in the coil coatings or potting compounds, and an indication of premature failure in the normal operating environment.

This report covers the development of a corona inception voltage (CIV) test that can be applied as a qualification, quality control, or quality assurance provision in a transducer autotransformer specification.

BACKGROUND

In 1987, General Electric (GE) and Raytheon were awarded production contracts to fabricate TR-317() sonar transducers according to a government developed Fabrication Specification Package (FSP). The FSP contains requirements that are deemed necessary for the composite transducer to achieve a 15-year service life. One such requirement was a corona specification for the autotransformer. As a part of the FSP development, certain critical assemblies of the transducer, which included the transformer, were procured by sample buys (in small quantities) to proof the FSP drawings. Because of limited financial resources in the years prior to the TR-317() production contract award, some of the sample buys were still in progress at the time of contract award.

During the transformer sample buy, a transformer contractor (Harder, Inc.) expressed concern about the validity of the corona requirements and the test method specified in the transformer drawings [1]. The concerns were: (1) the validity of a quantitative limit and measure of corona; and (2) the effectiveness of the specified filter circuit in attenuating the fundamental frequencies and passing the frequencies associated with corona. After consultation with the Naval Sea Systems Command (NAVSEA) it was determined that Harder's concerns were valid. These concerns led the TR-317() production contractors, with NAVSEA concurrence, to submit a Deviation/Waiver requesting modifications to the transformer drawing with respect to corona measurement, corona suppression, and abatement. In order to prevent delays in receiving First-Article transducers, the waiver (applicable to these transducers only) was allowed.

Since the waiver was allowed for First-Article transducers only, and at the beginning of production NAVSEA would again have to address the problem, NRL-USRD accepted the task of developing a technically defensible corona test for the TR-317 transformer that could be applied to the production transformers.

Several references are made in this report, to the TR-330A transducer (NAVSEA Drawing 53711-5517085), and TR-330A transformer corona data are presented. Corona requirements were not included in the original TR-330A FSP because the corona specifications for the TR-317() transformer were indefensible, and there were problems associated with corona testing the TR-317() transformer. The TR-317() FSP was used as a template for the TR-330A, thus the solution for the problem in the TR-317() also becomes the basis for a validated corona test for the TR-330A transformer.

APPROACH AND OBJECTIVES

The approach to the problem was to use the corona test circuit and specification in the TR-317() transformer drawing and make the necessary changes to provide a technically defensible test. That is, "defensible" in the sense that if a transformer failed to pass the test, the transformer would be questionable -- not the measurement, methodology, or test circuit.

The first step in the approach was to eliminate the *quantitative* test procedure and replace it with a *qualitative* test procedure. Therefore, instead of specifying a *quantitative* measure of corona current at some test voltage, the approach would be based on the *qualitative* detection of corona inception. That is, the goal was simply to be able to detect when corona first occurs, and measure the voltage at which it occurs.

The hardware for the test must be specified and tested to determine if corona can be detected reliably, repeatedly, and economically at the transformer assembly level.

The items needed to accomplish a CIV test, not including ancillary equipment, are:

- . A circuit to resonate with the autotransformer at the approximate center of the operating frequency range.
- . A high pass filter to attenuate the lower fundamental frequencies, yet allow the higher corona frequencies to be passed for detection.
- . A properly constructed Faraday shield to provide protection from outside electromagnetic interference which can mask the corona signal.

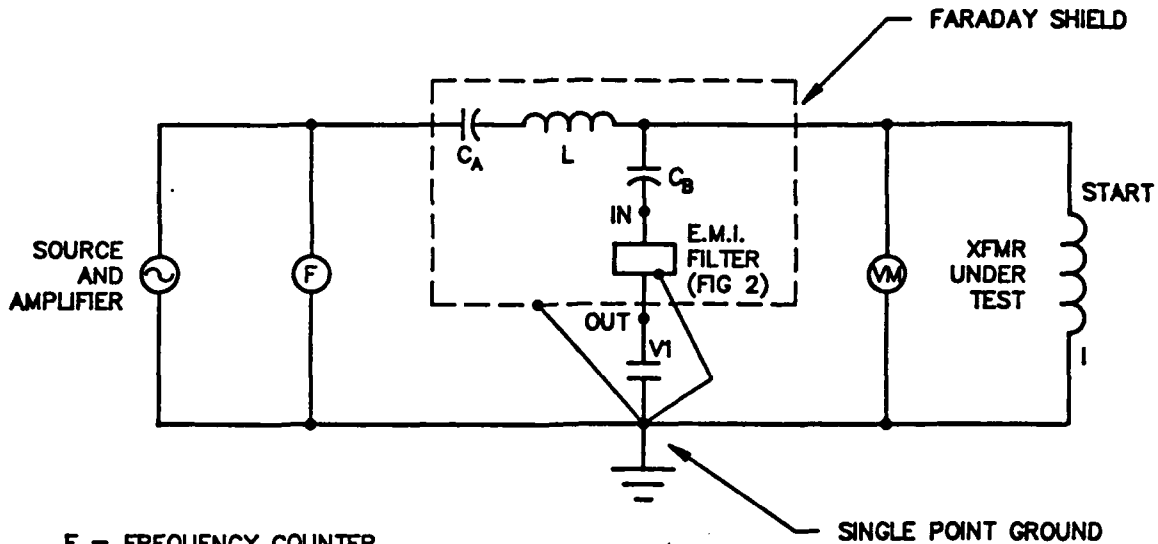
CIV TEST CIRCUIT DESCRIPTION

The corona test circuit and test procedure originally developed by the TR-317R transducer design team was inadequate for its intended purpose. Figure 1 depicts an improved CIV test circuit, including the ancillary test equipment, that has been developed from the original test circuit and validated. In Fig. 1, capacitor C_A , inductor L , and capacitor C_B are the essential parts of the autotransformer resonant circuit: C_A resonates with the transformer under test, C_B isolates the EMI filter from the rest of the circuit to prevent loading, and L enhances the corona detection by preventing the source from shunting the corona signal. C_A is calculated from the formula for the resonance frequency of an ideal parallel LC circuit which is,

$$f_r = \frac{1}{2\pi\sqrt{L_T C_A}} \quad (1)$$

If, in Eq. (1), we define L_T as the inductance of the transformer under test, and f_r as the approximate center of the transformers operating frequency band, then

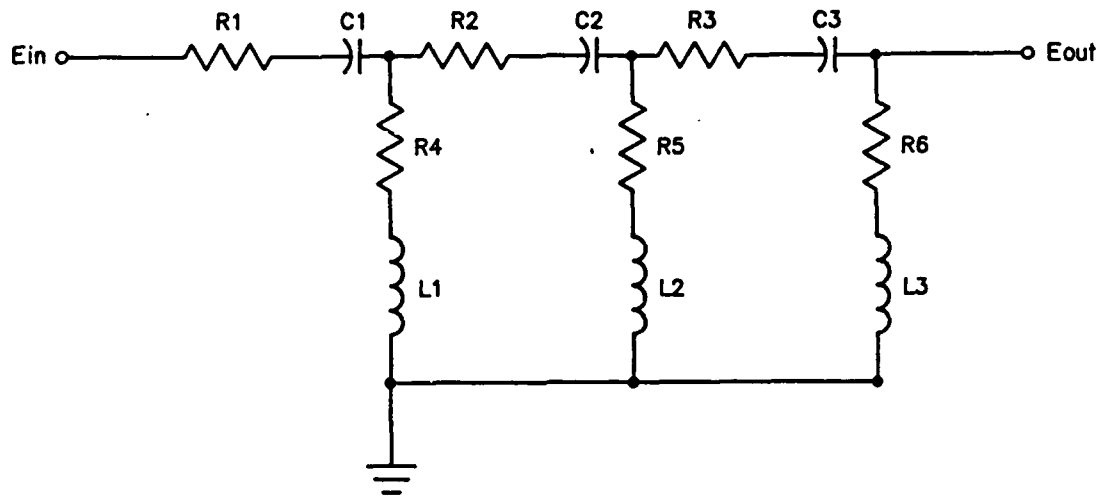
$$C_A = \frac{1}{4\pi^2 f_r^2 L_T} \quad (2)$$



F = FREQUENCY COUNTER
 C = 5000pF 5000 WVDC .1%DF
 C = 150 pF CAPACITOR 5000 WVDC .1%DF
 VI = INPUT TO OSCILLOSCOPE
 VM = VOLTMETER, FLUKE 8922A TRUE RMS OR EQUIVALENT (USE 100X OR 1000X PROBE).
 L = 2.4mH INDUCTOR PIE WOUND LOW CAPACITANCE
 SOURCE AND AMPLIFIER = WAVETEK 182A FUNCTION GENERATOR AND INSTRUMENTS INC.,
 LDV2-6, 10KVA POWER AMPLIFIER OR EQUIVALENT.

Fig. 1 - CIV test circuit.

Figure 2 is a schematic diagram of the EMI filter section of the circuit shown in Fig. 1 and depicts the three stage ladder, R-C, R-L, high-pass filter which passes the corona signal to an appropriate detector. The circuit component values shown in Fig. 2 were determined by using equations derived from the model of a three-stage, high pass, L section filter. The circuit values were verified on a computer program, developed by the Naval Ocean Systems Center (NOSC) for an L section filter. The computer program for the circuit is contained in the appendix.



R1, R2, R3 = 4000 OHMS, 5W
 R4 = 400 OHMS 5W
 R5 = 300 OHMS 5W
 R6 = 200 OHMS 5W

C1 = 0.5 μ F 3KVDC
 C2 = 0.2 μ F 3KVDC
 C3 = 0.1 μ F 3KVDC

L1, L2, L3 = 0.02H 0.1AMP

COMPONENTS TOLERANCE = \pm 5%

Fig. 2 - EMI filter for CIV test circuit.

Several different circuits were evaluated to determine the five most optimum values shown in Fig. 3, although all the circuits evaluated were, to some degree, acceptable in that the fundamental frequency would be attenuated while allowing the corona frequency to be passed through for detection. Circuit value set #1 was chosen because at no time did E_{out}/E_{in} exceed 0 dB, and set #1 exhibited a steep cutoff slope. Set #2 also exhibited a steep cutoff slope, but exceeded 0 dB E_{out}/E_{in} . The other circuit values would allow attenuation of the fundamental frequency, but not to the same degree as set #1. The calculated values (along with actual tests) verified that the circuit would be acceptable for testing both TR-317() and TR-330 transformers with only a change in the value of capacitor C_A shown in Fig. 1.

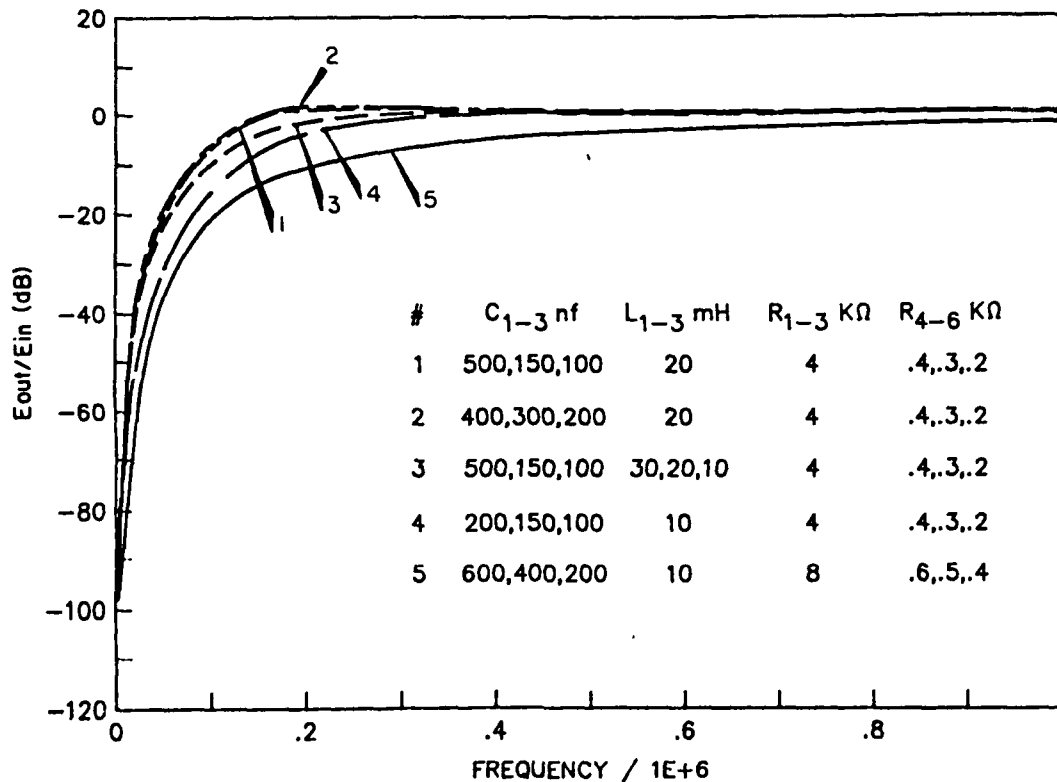


Fig. 3 - EMI filter characteristics versus frequency.

The quality of the components used in the circuit is very important to eliminate the possibility of corona in any other part of the circuit except the transformer under test. All capacitors are oil-filled polypropylene capacitors and have a dielectric dissipation factor of 0.001 (0.1%) or less.

The inductors for the CIV test circuit were designed and fabricated at NRL-USRD. Inductor L, in Fig. 1, consists of 18 turns of #25 polythermaleze insulated copper wire, pi wound on a three section bobbin. The bobbin assembly is placed into a Ferroxcube #2616 P 3C8 ferrite cup core, and the Q, measured on a RLC bridge, is 68. Inductors L1, L2, and L3 in Fig. 2 consist of 223 turns of #25 polythermaleze insulated copper wire in a Ferroxcube #3622 PA400 3B7 ferrite cup core; and the measured Q is approximately 270.

Figures 4a through 4f are photographs of the test circuit chassis and enclosure. Excluding the ancillary equipment, only the transformer under test is not within the shielded enclosure.

The Faraday shield for the circuit consists of the aluminum chassis bottom, the chassis faceplate, and the copper wire mesh attached by screws to the chassis as shown in Fig. 4c. Shielding the circuit could also be accomplished by enclosure in a metal box or cabinet instead of using a copper mesh. The mesh was used during the development phase to visually determine the existence of any arcing. The circuit components and wiring should be isolated from the enclosure to minimize any interaction between the return current path of the circuit and the grounded case shield which should not have any current flow. This condition is implied by the single point ground connection shown in Fig. 1.

The electrical connections between circuit components on the chassis are made with 15 kV dc rated, silicone insulated, 20 AWG stranded copper wire.



Fig. 4a - Front view of faceplate.

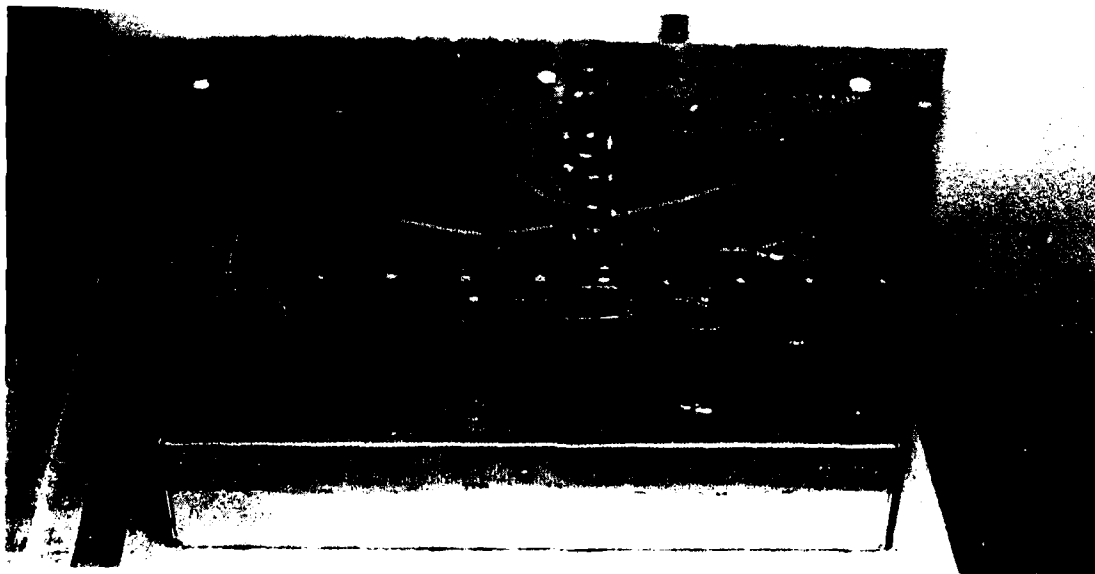


Fig. 4b - Rear view shield mesh installed.

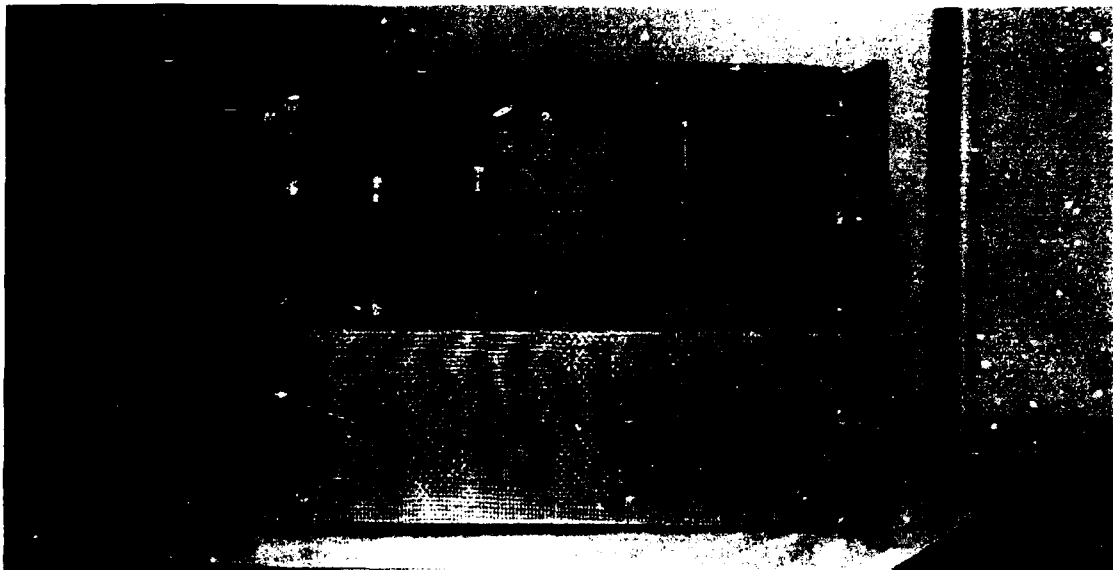


Fig. 4c - Side view Faraday shield/chassis.

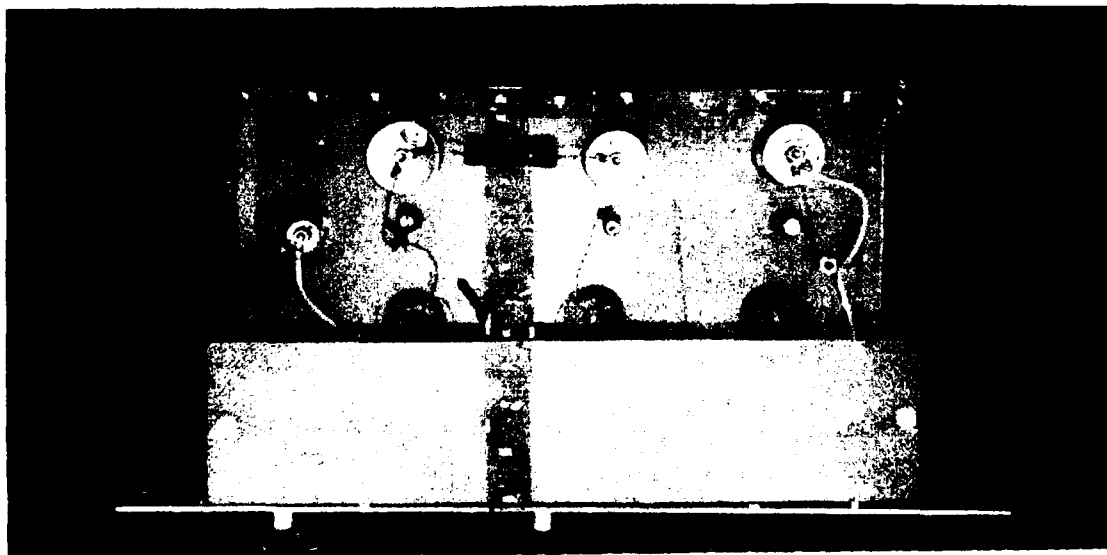


Fig. 4d - Top view mesh installed.

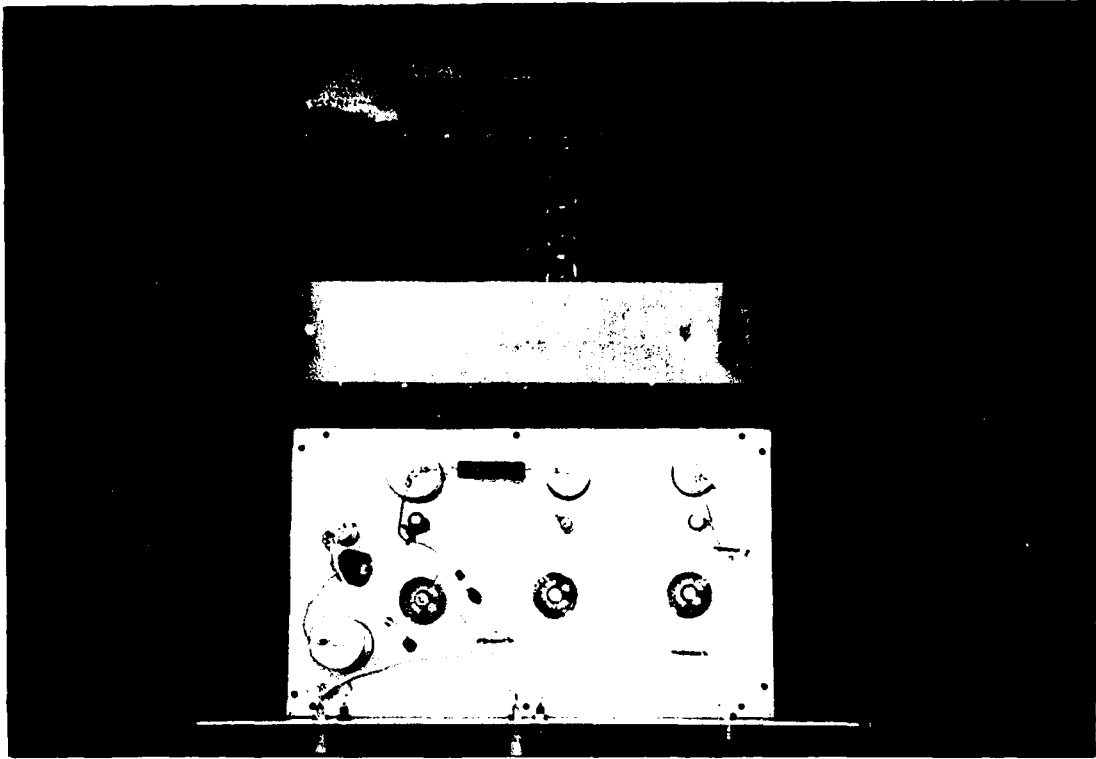


Fig. 4e - Top view internal components/mesh shield.

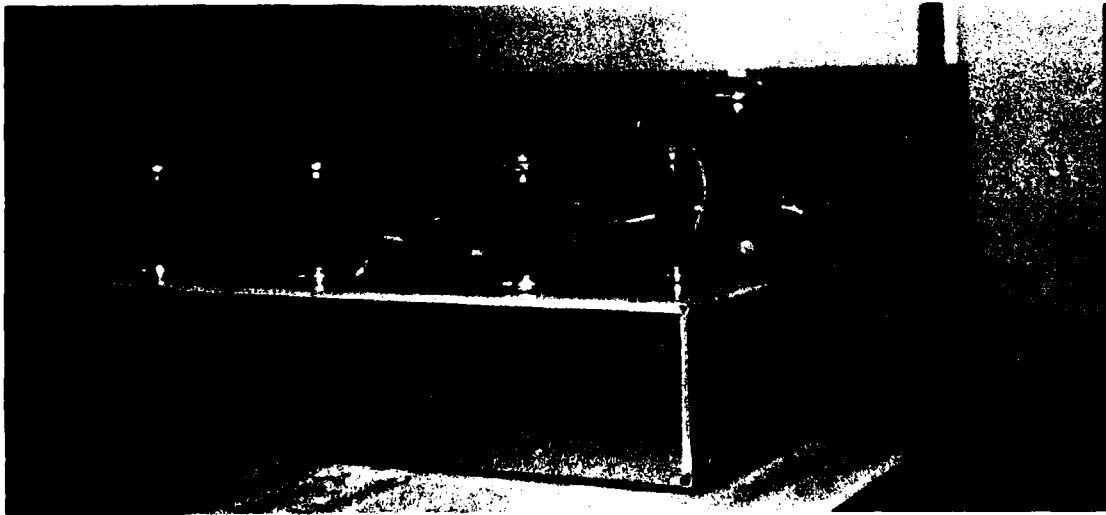


Fig. 4f - Rear view component mounting.

CIV TEST PROCEDURES

An oscilloscope with a measurement capability of dc to at least 20 MHz should be used. The output connector V1 is connected to the vertical input of the oscilloscope. The oscilloscope should be preset to a vertical input sensitivity of 2 V/division and a horizontal sweep time of 20 μ s/division. The rest of the setup is done as shown in Fig. 1.

The hookup wire from the power amplifier output should be a single twisted pair. All other circuit connections are made with RG-58/U coaxial cable (which will withstand 2 kV) to further shield the circuit from EMI.

In evaluating test procedures, measurements were made with and without transformer shielding and the results indicated that a shield was not necessary for the transformers tested (TR-317 and TR-330A autotransformers). In extremely noisy environments, transformer shielding can easily be accomplished, if necessary, but all shielded ground connections to the circuit should be made at a single point on the chassis.

To make the CIV test:

1. Adjust the input voltage as read on the VM to the transformer to approximately 5 to 10V at the approximate resonant frequency [fr, Eq.(1)] of the autotransformer under test. Then carefully adjust the frequency generator to the frequency that produces the maximum voltage on the voltmeter, or the maximum amplitude waveform on the oscilloscope.
2. After the frequency is adjusted to resonance, carefully increase the applied voltage as read on the VM until the corona inception "hash" is sporadically observed on the oscilloscope. Corona inception is evidenced by the sporadic high-frequency "hash" type oscillations on the oscilloscope waveform and by an increase in the voltmeter reading. Record the frequency and the voltage at which corona inception is observed.
3. Repeat the first two steps two more times to insure repeatability.

Figures 5 and 6 serve to illustrate how the corona "hash" appears on the oscilloscope waveform. Figure 5a shows the scope wave form for a TR-330A transformer under normal drive conditions and Fig. 5b under higher drive conditions that has produced corona. Note that a small amount of the drive frequency is evident in the oscilloscope display; but, as seen in the high-drive condition, Fig. 5b, is not detrimental to the observance of the corona "hash" in Fig. 5b. Figures 6a and 6b show the same drive conditions just described, respectively, for a TR-317 transformer; Fig. 6a shows a clean waveform; and Fig. 6b shows a waveform with high-frequency corona "hash." If desired, corona detection can be augmented if an AM radio receiver is available, placed near the transformer under test, and tuned to 550-560 kHz; sporadic noise (loud static) will be heard from the receiver at the same time that corona "hash" (as illustrated in Figs. 5b and 6b) appears on the oscilloscope waveform.

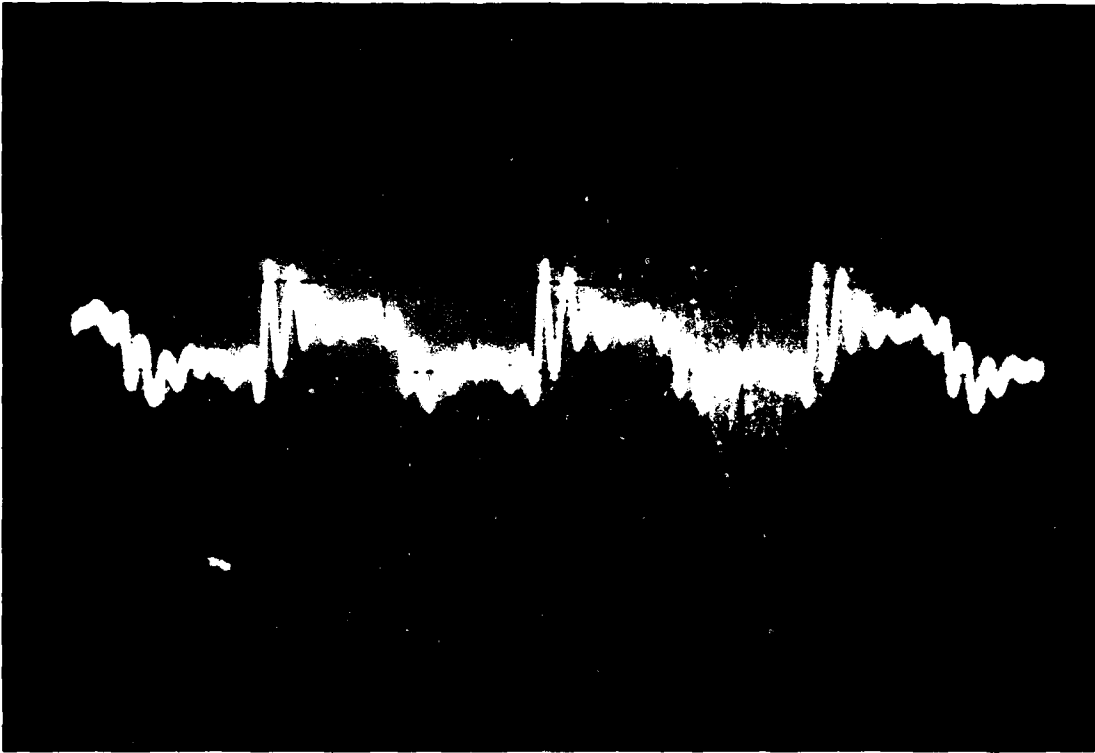


Fig. 5a - Output of the CIV test circuit for a TR-330A autotransformer under normal drive conditions.

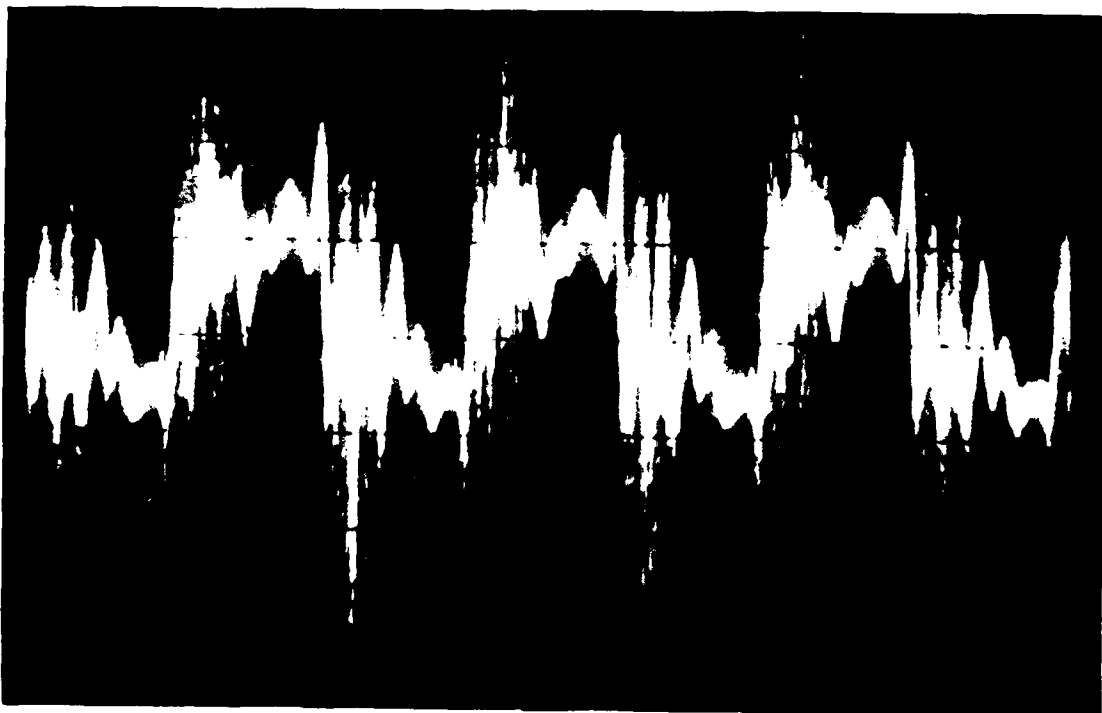


Fig. 5b - Output of the CIV test circuit for a TR-330A autotransformer under high-drive conditions with corona "hash."

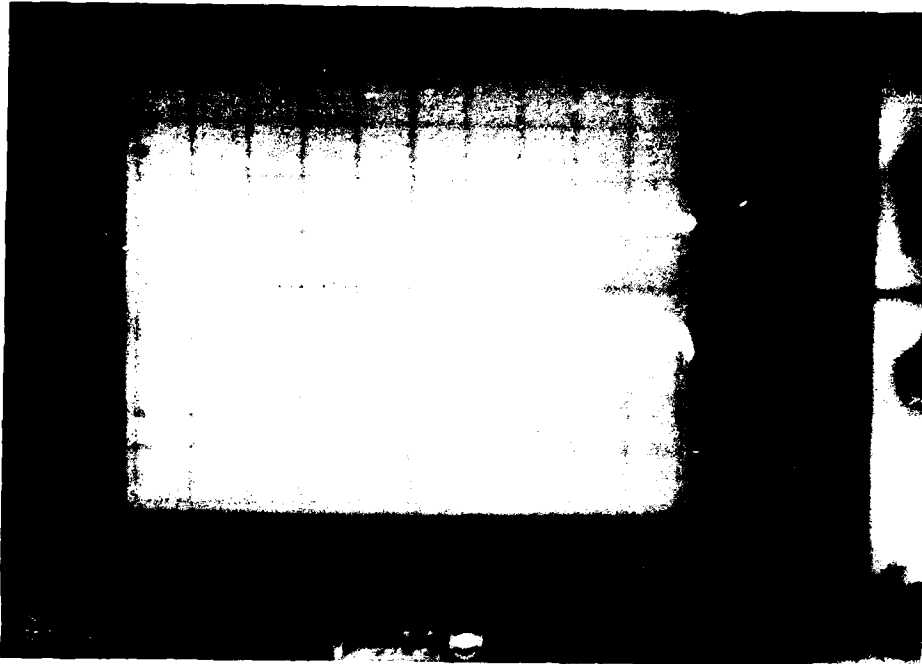


Fig. 6a - Output of the output of the CIV test circuit for a TR-317 autotransformer under normal drive conditions.



Fig. 6b - Output of the CIV test circuit for a TR-317 autotransformer under high-drive conditions with corona "hash."

CIV TEST PRECAUTIONS

The amplifier chosen for the CIV test must be adequate for the intended purpose. For the TR-317 transformer test circuit, shown in Fig. 1, the power amplifier should be an Instruments Inc. Model LDV 2-6, 10 kVA or an equivalent that will supply the current and voltages necessary for the test. For the TR-330A transformer, a McIntosh Model MC-2500, 1 kVA amplifier or equivalent is adequate. Since a high impedance output may make the test circuit susceptible to high-frequency noise pickup, one should use the minimum impedance setting on the amplifier compatible with providing the required test voltage across the transformer.

For the TR-317 corona test circuit, capacitor C_A should be the value shown in Fig. 1. For the TR-330A transformer corona test, the value of C_A should be 3000 pF.

It should be emphasized that corona occurs in the presence of high voltages; therefore, high voltage is required to make the test. The measurements should be made with care and respect for the operating conditions to prevent serious electrical shock to the operator.

CIV TEST DATA

The CIV test circuit and procedures previously described have been used to determine the CIV for several TR-317 and TR-330A production transformers and for experimental TR-330A toroidal autotransformers. The TR-317 group contains transformers salvaged from TR-317 autopsies, transformers from the TR-317R sample buy, and GE TR-317() production transformers -- the latter two made by Harder, Inc. and Chloride, respectively. The transformers are fabricated in a variety of ways; i.e., varnish coated but not potted, potted, etc., these conditions are noted with the measured data in Tables 1 and 2.

The TR-330A group contains transformers which are potted except for two experimental toroidal transformers which are unpotted. The unpotted transformers were included in the test to demonstrate the difference in CIV between unpotted and potted transformers.

Table 1 provides data from the CIV tests on the TR-317 transformers. The serial numbers shown in the table are the actual serial numbers shown on the transformers, except for those that have an A prefix (the A prefix indicates the serial number of the transducer from which the transformer was removed). The table indicates, in the column headed "Type," certain conditions and materials used in the fabrications of the transformer. Table 1a is corona test data for TR-317 transformers, some of which were salvaged from autopsies. Table 1b is test data for 6 GE TR-317() production transformers (Chloride). Table 1c shows the test data from 6 TR-317R Sample Buy (Harder) transformers. The frequency data shown in Table 1 is relative to the electrical resonance of the transducer in-water (peak of free-field voltage sensitivity). The table indicates three independent measurements on each transformer. Measurements were taken at approximately 1 minute intervals.

DIEBEL and TMS

Table 1a - Corona Test Data for TR-317 Autotransformers.

Ser #	CIV	Relative	Relative	Relative	Type	CIV			
		Freq (Hz)	Freq (Hz)	Freq (Hz)		Mean	Std Dev		
013	2731	790	2522	858	2561	852	Coated	2604	111
405	2915	586	2665	583	2596	597	Coated	2725	168
H-1	3115	462	2557	499	2550	473	Potted	2741	324
H-2	3007	518	2825	532	3110	519	Potted	2981	144
H-3	2731	578	2899	613	2756	570	Coated	2795	90
A31108	3372	587	3169	634	3125	604	(Coated	3222	132
A30433	2549	586	2778	605	2882	596	From	2736	170
A22719	3196	596	2954	569	3020	572	TR-317()	3057	125
A24394	2998	593	3142	579	3057	772	Autopsies)	3066	72

Table 1b - TR-317() Autotransformers from GE (made by Chloride).

Ser #	CIV	Relative	Relative	Relative	Type	CIV			
		Freq (Hz)	Freq (Hz)	Freq (Hz)		Mean	Std Dev		
1	3152	509	3199	596	3267	597	Coated	3208	58
2	3428	501	3137	617	3203	616	with corona	3256	153
3	3056	515	3200	609	3175	608	suppressant	3144	77
4	3247	620	3105	619	3050	619	*	3134	102
5	3007	624	3109	623	3135	624		3084	68
6	3108	620	3013	621	3135	618		3085	64

* Hi Temp 221, Hi Temp Resins Inc.

Table 1c - TR-317R Sample Buy Autotransformers from Harder Inc.

Ser #	CIV	Relative	Relative	Relative	Type	CIV			
		Freq (Hz)	Freq (Hz)	Freq (Hz)		Mean	Std Dev		
037	2581	576	2564	577	2581	577	Coated	2575	10
041	2522	570	2479	569	2526	569	with	2509	26
054	3003	571	3100	568	3044	569	epoxy	3049	49
058	2570	568	2605	569	2500	568	resin	2558	53
060	2570	582	2500	582	2580	582	as per	2550	44
061	2700	579	2816	578	2703	579	dwg pkg	2760	66

NRL MEMORANDUM REPORT 6559

Table 2 provides corona inception data for several TR-330A transformers. The test circuit shown in Fig. 1 was used to take the data with the following modifications: Capacitor CA was changed to 3000 pF and the power amplifier was a McIntosh Model 2500. The frequencies shown in Table 2 are, in this case, the actual resonance frequencies.

Table 2 - TR-330A Autotransformer Corona Test Data.

Ser #	CIV	Freq (Hz)	CIV	Freq (Hz)	CIV	Freq (Hz)	Type	CIV	
								Mean	Std Dev
1	1504	6925	1500	6922	1513	6922	Ferrite	1506	7.0
2	1526	6944	1568	6948	1540	6950	Ferrite	1545	21
3	1500	6942	1517	6945	1522	6942	Ferrite	1513	12
194	1559	6583	1587	6582	1590	6582	Ferrite	1579	17
567	1583	7152	1548	7148	1597	7150	Ferrite	1576	25
617	1559	7088	1556	7087	1560	7086	Ferrite	1558	2.1
MPP 1*	530	6543	600	6531	596	6534	Unpotted	575	39
MPP 2*	1659	6302	1644	6282	1652	6288	Potted	1652	7.5
MPP 3*	545	6533	585	6526	562	6527	Unpotted	564	20
MPP 4*	1655	6235	1650	6234	1659	6235	Potted	1655	4.5

* MPP 1 = Unpotted toroid 55251-W4 core.

MPP 2 = Potted toroid 55251-W4 core.

MPP 3 = Unpotted toroid 55248-A2 core.

MPP 4 = Potted toroid 55248-A2 core.

MPP = Molypermalloy Powder core manufactured by Magnetics Inc.

Potting compound = Eccobond 45 black with 19M catalyst. Autotransformer was vacuum potted.

CONCLUSIONS

The data from both the TR-330A, TR-317(), TR-317R, and the TR-317 autotransformer samples indicate that the TR-330A representatives have a smaller standard deviation than representatives from the TR-317 group. In particular, serial #H-1 of the TR-317 group, with a standard deviation of 324 V, presents an interesting problem. Operator error is the most probable cause of the first excessively high CIV reading for serial #H-1, as the last two readings are much closer to each other (within 7 V). Standard deviations were below 6% of the mean CIV recorded. For this type of measurement, the standard deviations are acceptable. An AM receiver was added to the rest of the ancillary equipment to gather the TR-330A data where the standard deviations were less than 2% of the mean CIV. The AM receiver has the advantage of providing the operator with an audio reference which compliments the visual reference.

SUMMARY

An autotransformer corona test circuit has been developed and tested, ancillary equipment for the test has been identified and recommended, test procedures and precautions have been presented, and measured corona data are presented for autotransformers of different manufacture and type. As a result, recommendations have been made to NAVSEA to incorporate the test into appropriate drawing packages.

ACKNOWLEDGMENTS

The authors wish to express appreciation to Homer Ding of NOSC, San Diego, CA, for his work on the computer program and for his guidance, and to Leo Johnson, also of NOSC, for his guidance.

REFERENCES

1. R.W. Timme, "SONAR TRANSDUCER RELIABILITY IMPROVEMENT PROGRAM (STRIP), FY87 Fourth Quarter Progress Report," NRL Memorandum Report No. 6128 (Sep 1987).
2. R.W. Timme and J.F. Cartier, "NAVSEA TRANSDUCER IMPROVEMENT PROGRAM (NTIP), FY88 Second Quarter Progress Report," NRL Memorandum Report No. 6235 (Apr 1988).
3. R.W. Timme and J.F. Cartier, "NAVSEA TRANSDUCER IMPROVEMENT PROGRAM (NTIP), FY88 Fourth Quarter Progress Report," NRL Memorandum Report No. 6371 (Oct 1988).

Appendix

LADDER COMPUTER PROGRAM FOR EMI FILTER

Figure A1 contains the circuit identifiers used to operate the computer program which follows. This program was used successfully on a Hewlett Packard 9000, model 200 computer using HP BASIC 5.0. Lines 80-120 in the program describe the initial procedure of loading the program.

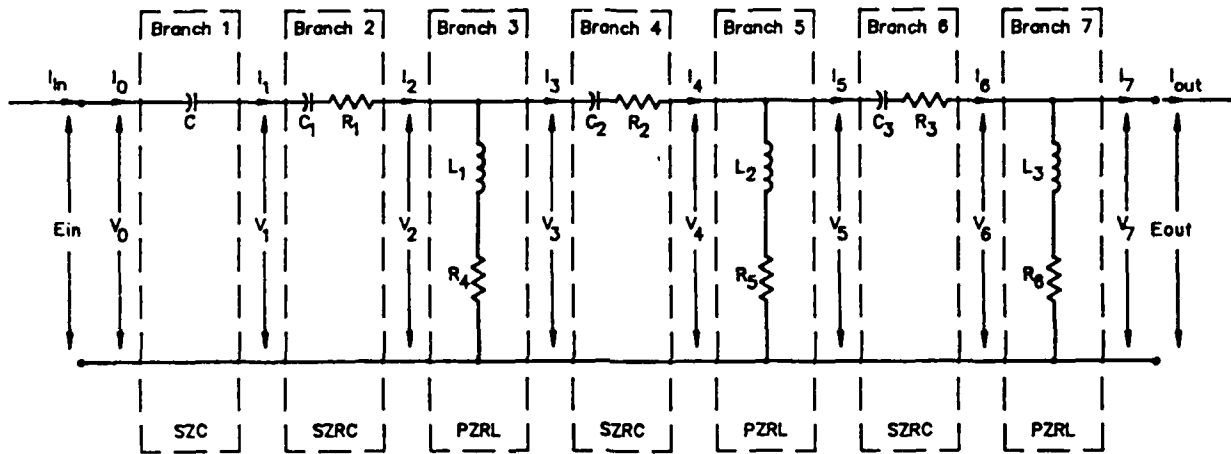


Fig. A1 - Circuit identifiers.

```

10      ! LADDER PROGRAM VERSION 1.0, OCTOBER 1988
20      ! AUTHOR: HOMER DING
30      !           NAVAL OCEAN SYSTEMS CENTER
40      !           CODE 711, BLDG 132
50      !           SAN DIEGO, CA. 92152
60      !           (619) 553-1443
70      !
80      ! THIS PROGRAM WAS WRITTEN IN HP BASIC 4.0.
90      ! A COMPLEX EXTENSION FROM STRUCTURED SOFTWARE SYSTEMS
100     ! WAS ALSO USED.
110     ! THIS PROGRAM SHOULD WORK WITH HP BASIC 5.0 IF THE GET STATEMENT
120     ! IS USED TO PLACE THE ASCII FILE IN THE COMPUTER.
130     !
140     ! THIS PROGRAM WILL COMPUTE THE OUTPUT/INPUT VOLTAGE TRANSFER FUN
CTION
150     ! OF A LADDER NETWORK.
160     ! Nb = TOTAL NUMBER OF BRANCHES
170     ! THE FIRST LETTER OF A BRANCH TYPE IS S, P, OR T
180     ! S = SERIES BRANCH
190     ! P = PARALLEL BRANCH
200     ! T = IDEAL TRANSFORMER
210     ! IF T THEN THERE ARE NO OTHER LETTERS WHICH DESCRIBES THE BRANCH
TYPE
220     ! Z = SERIES COMBINATION OF BRANCH COMPONENTS
230     ! Y = PARALLEL COMBINATION OF BRANCH COMPONENTS
240     ! C, L, AND R = TYPE OF BRANCH COMPONENTS
250     !
260     COM /Menu/ Menu$(20)[80],Ch$(20)[6],Mark0$(1),Mark1$(1),INTEGER N
ent,Nchr,Xtab,Ytab,Ch(20)
270     COM /N/ @Path1,Netfile$(20),Cmnt$(2)[80],Br$(20)[6],INTEGER Nb,RE
AL R(20),L(20),C(20),Clk$(80)
280     COM /C/ REAL F,W,F1,F2,Fs,Supsegs(6,4),INTEGER Nseg,Ns,Tns,Logflg
,Logflags(6),COMPLEX Eb(20),Ib(20),Imb(20),Yb(20),Anet(2,2)
290     COM /Lj/ Lf$(1),Cr$(1),Ff$(3),Esc$(1),Portrait$(6),Landscape$(6)
,Font$(4)[80],INTEGER Psc,Page,Npages,Fontnum
300     INTEGER Chose,M,K1,K2,I
310     REAL X,Y,G,B,Z,Mag,Ang
320     DIM S$(80),K$(2)
330     LOCAL 7
340     Lf$=CHR$(10)
350     Cr$=CHR$(13)
360     Esc$=CHR$(27)
370     Ff$=Esc$&CHR$(7)&CHR$(12)
380     Portrait$=Esc$&"&100"
390     Portrait$=Esc$&"&110"
400     Font$(1)=Portrait$&Esc$&"(10U"&Esc$&"(s0p10.00h12.00v0s0B"&Esc$&"
&16D"
410     Font$(2)=Landscape$&Esc$&"(10U"&Esc$&"(s0p10.00h12.00v0s0B"&Esc$&"
"&16D"
420     Font$(3)=Portrait$&Esc$&"(10U"&Esc$&"(s0p16.66h8.50v0s-3B"&Esc$&"
&15.6667C"
430     Font$(4)=Landscape$&Esc$&"(10U"&Esc$&"(s0p16.66h8.50v0s-3B"&Esc$&"
"&15.6667C"
440     REPEAT

```

```

450      CALL Menu(1)
460      CALL Cls
470      SELECT Ch$(1)
480      CASE "G"
490          CALL Getnfile
500      CASE "E"
510          CALL Entnet
520      CASE "S"
530          CALL Recnfile
540      CASE "P"
550          CALL Entint("PRINTER SELECT CODE(1=CRT, 2=THINKJET, 9=LASER
JET)",Psc)
560          Clk$(1,80)=FNClck$
570          CALL Prtnet
580      CASE "F"
590          CALL Entswpsegs
600      CASE "R"
610          CALL Entint("PRINTER SELECT CODE(1=CRT, 2=THINKJET, 9=LASER
JET)",Psc)
620          Clk$(1,80)=FNClck$
630          CALL Prtrat
640      CASE "I"
650          CALL Entint("PRINTER SELECT CODE(1=CRT, 2=THINKJET, 9=LASER
JET)",Psc)
660          Clk$(1,80)=FNClck$
670          CALL Prtlev
680      CASE "A"
690          CALL Entint("PRINTER SELECT CODE(1=CRT, 2=THINKJET, 9=LASER
JET)",Psc)
700          Clk$(1,80)=FNClck$
710          CALL Prtanet
720      END SELECT
730      UNTIL Ch(1)=Nent
740      CALL Cls
750      STOP
760      END
770      !
780      SUB Beep(OPTIONAL INTEGER Nbeep,Freq,REAL Secon,Secof)
790          INTEGER I,F,N
800          REAL S1,S0
810          N=1
820          F=2700
830          S1=.03
840          S0=.04
850          IF NPAR>0 THEN N=Nbeep
860          IF NPAR>1 THEN F=Freq
870          IF NPAR>2 THEN S1=Secon
880          IF NPAR>3 THEN S0=Secof
890          FOR I=1 TO N
900              BEEP F,S1
910              WAIT S0
920          NEXT I
930      SUBEND ! Beep
940      !

```

```

950     SUB Fkeys(OPTIONAL INTEGER Keyon) !0=OFF, 1=ON
960         INTEGER K
970         K=1
980         IF NPAR=1 THEN K=NOT Keyon
990         CONTROL CRT,12;K
1000    SUBEND ! Fkeys
1010    !
1020    SUB Curs(OPTIONAL INTEGER Col,Row,Curson)
1030        INTEGER C,R,S
1040        C=1
1050        R=19
1060        S=1 ! CURS ON
1070        IF NPAR>0 THEN C=Col
1080        IF NPAR>1 THEN R=Row
1090        IF NPAR>2 THEN S=Curson
1100        CONTROL CRT,0;C
1110        CONTROL CRT,1;R
1120        CONTROL CRT,10;S
1130    SUBEND ! Curs
1140    !
1150    SUB Cls(OPTIONAL INTEGER Col,Row,Keyof)
1160        INTEGER C,R,K
1170        C=1
1180        R=19
1190        K=0
1200        IF NPAR>0 THEN C=Col
1210        IF NPAR>1 THEN R=Row
1220        IF NPAR>2 THEN K=Keyof
1230        OUTPUT CRT;CHR$(128);
1240        OUTPUT KBD;CHR$(255)&"K";
1250        CALL Fkeys(K)
1260        CALL Curs(C,R)
1270    SUBEND ! Cls
1280    !
1290    DEF FNclock$
1300        RETURN UPC$(DATE$(TIMEDATE)&"", "&TIME$(TIMEDATE))
1310    FNEND ! FNclock$
1320    !
1330    SUB Continue
1340        INTEGER K1,K2
1350        DISP "TO CONTINUE HIT ANY KEY"
1360        CALL Kbscan(K1,K2)
1370        DISP
1380    SUBEND ! Continue
1390    !
1400    DEF FNAtn2(X,Y,OPTIONAL INTEGER Posang) ! 4 QUADRANT ARC TANGENT
1410        INTEGER P
1420        DEG
1430        P=0
1440        IF NPAR=3 THEN P=Posang
1450        RETURN ARG(CMPLX(X,Y))+(P<>0)*360
1460    FNEND ! FNAtn2
1470    !
1480    SUB Rec2pol(Xr,Xi,Mag,Ang,OPTIONAL INTEGER Posang)

```

```

1490         INTEGER P
1500         REAL M
1510         DEG
1520         P=0
1530         IF NPAR=5 THEN P=Posang
1540         M=ABS(CMPLX(Xr,Xi))
1550         Ang=FNAtn2(Xr,Xi,P)
1560         Mag=M
1570     SUBEND ! Rec2pl
1580     !
1590     SUB Pol2rec(Mag,Ang,Xr,Xi)      !POLAR TO RECTANGULAR
1600         REAL Txr
1610         Txr=Mag*COS(Ang)
1620         Xi=Mag*SIN(Ang)
1630         Xr=Txr
1640     SUBEND ! Pol2rec
1650     !
1660     DEF FNDb(X)
1670         RETURN 20*LGT(X)
1680     FNEND ! FNDb
1690     !
1700     DEF FNAdb(X)
1710         RETURN 10^(.05*X)
1720     FNEND ! FNAdb
1730     !
1740     SUB Entstr(Prompt$,Str$)
1750         DIM Temp$(80)
1760         DISP Prompt$&" = ";
1770         OUTPUT 2;Str$;
1780         OUTPUT KBD;CHR$(255);"H";
1790         LINPUT Temp$
1800         IF Temp$<>" " THEN Str$=Temp$
1810         PRINT Prompt$&" = ";Str$
1820     SUBEND ! EntSTR
1830     !
1840     SUB Entint(Prompt$,INTEGER I,OPTIONAL INTEGER Noprt)
1850         DIM Temp$(80)
1860         INTEGER Prtflg
1870         Prtflg=1
1880         IF NPAR=3 THEN Prtflg=NOT Noprt
1890         DISP Prompt$&" = ";
1900         OUTPUT 2;VAL$(I);
1910         OUTPUT KBD;CHR$(255);"H";
1920         LINPUT Temp$
1930         IF Temp$<>" " THEN I=VAL(Temp$)
1940         IF Prtflg THEN PRINT Prompt$&" = ";I
1950     SUBEND ! Entint
1960     !
1970     SUB Entreal(Prompt$,REAL U,OPTIONAL REAL Unitmult)
1980         REAL U
1990         DIM Temp$(80)
2000         U=1
2010         IF NPAR=3 THEN U=Unitmult
2020         DISP Prompt$&" = ";

```

```

2030         OUTPUT 2;VAL$(V/U);
2040         OUTPUT KBD;CHR$(255);"H";
2050         LINPUT Temp$
2060         IF Temp$<>" " THEN V=VAL(Temp$)*U
2070         PRINT Prompt$&" = ";V/U
2080     SUBEND ! Entreal
2090     !
2100     SUB Prtmenu(INTEGER Nmenu)
2110         COM /Menu/ Menu$(20)[80],Ch$(20)[6],Mark0$(1),Mark1$(1),INTEGE
R Nent ,Nchr ,Xtab ,Ytab ,Ch(20)
2120         PRINTER IS CRT
2130         SELECT Nmenu
2140         CASE 1
2150             RESTORE Menu1
2160         CASE 2
2170             RESTORE Menu2
2180         CASE 3
2190             RESTORE Menu3
2200         END SELECT ! Nmenu
2210         CALL C1s
2220         ALPHA OFF
2230         READ Menu$(0)
2240         READ Nent
2250         READ Nchr
2260         Mxlen=0
2270         FOR I=1 TO Nent
2280             READ Menu$(I)
2290             Mxlen=MAX(Mxlen,LEN(Menu$(I)))
2300         NEXT I
2310         Xtab=INT(.5*(80-Mxlen))+1
2320         Ytab=18-Nent
2330         PRINT TABXY(Xtab+Nchr+2,Ytab-1);Mark0$&Menu$(0)
2340         FOR I=1 TO Nent
2350             PRINT TABXY(Xtab,Ytab+I);Mark0$&Menu$(I)
2360         NEXT I
2370         SUBEXIT
2380         !
2390 Menu1:   DATA "LADDER NETWORK PROGRAM",9,1
2400         DATA G GET DATA FROM DISK
2410         DATA E ENTER NETWORK
2420         DATA S SAVE DATA TO DISK
2430         DATA P PRINT NETWORK
2440         DATA F ENTER FREQUENCY SWEEP
2450         DATA A PRINT A MATRIX
2460         DATA R PRINT I/O RATIOS
2470         DATA I PRINT INTERNAL LEVELS
2480         DATA Q QUIT
2490         !
2500 Menu2:  DATA MAIN PLOT MENU,9,1
2510         DATA I INITIALIZE FOR PLOTTING
2520         DATA T ENTER TYPE OF PLOT
2530         DATA N ENTER NAMES FOR PLOT AND AXES
2540         DATA S ENTER SCALES
2550         DATA L ENTER LEGEND

```

```

2560          DATA A PLOT AXES
2570          DATA D PLOT DATA
2580          DATA + ADD TO FILE NUMBERS
2590          DATA Q QUIT
2600          !
2610          SUBEND ! Prtmenu
2620          !
2630          SUB Kbscan(INTEGER K1,K2,OPTIONAL K$)
2640              DIM Ch$(2)
2650              K1=0
2660              ON KBD GOSUB Kbintr
2670              REPEAT
2680              UNTIL K1<>0
2690              OFF KBD
2700              SUBEXIT
2710              !
2720 Kbintr:    Ch$=KBD$
2730            K1=NUM(Ch$(1,1))
2740            IF K1=255 THEN
2750                K2=NUM(Ch$(2,2))
2760                Ch$=Ch$(2,2)
2770            ELSE
2780                K2=0
2790            END IF
2800            IF NPAR=3 THEN K$=Ch$
2810            RETURN
2820          SUBEND ! Kbscan
2830          !
2840          SUB Menu(INTEGER Nmenu)
2850              COM /Menu/ Menu$(20)[80],Ch$(20)[6],Mark0$(1),Mark1$(1),INTEGE
R Nent,Nchr,Xtab,Ytab,Ch(20)
2860              INTEGER I,K1,K2,Wdth,Chose,Xtm1
2870              DIM K$(2)
2880              Mark0$=CHR$(128)
2890              Mark1$=CHR$(129)
2900              CALL Prtmenu(Nmenu)
2910              Chose=Ch(Nmenu)
2920              Xtm1=Xtab-1
2930              DISP "SELECT WITH ALPHA-NUMERIC OR UP-DOWN KEYS, THEN HIT RETU
RN"
2940              REPEAT
2950 Scan:      IF Chose<1 THEN Chose=Nent
2960              IF Chose>Nent THEN Chose=1
2970              PRINT TABXY(Xtm1,Ytab+Chose);Mark1$&" "&Menu$(Chose)&" ";
2980              CALL Kbscan(K1,K2,K$)
2990              PRINT TABXY(Xtm1,Ytab+Chose);Mark0$&" "&Menu$(Chose)&" ";
3000              IF K1<>255 THEN
3010                  GOSUB Search
3020              ELSE
3030                  IF K2<>69 THEN Chose=Chose+(K2=86)-(K2=94)
3040              END IF
3050              UNTIL K1=255 AND K2=69
3060              Ch(Nmenu)=Chose
3070              Ch$(Nmenu)=Menu$(Chose)[1,Nchr]

```

```

3080         DISP
3090         SUBEXIT
3100         !
3110 Search:   FOR I=1 TO Nent
3120             IF K$=Menu$(I)[1,1] THEN
3130                 Chose=I
3140                 I=Nent
3150             END IF
3160         NEXT I
3170         RETURN
3180         !
3190         SUBEND ! Menu
3200         !
3210         SUB Tboxon
3220             Lj$="CALL 25CE,1"&CHR$(13)
3230             OUTPUT 9;"QUIET ON"&CHR$(13);
3240             OUTPUT 9;Lj$;
3250         SUBEND ! Tboxon
3260         !
3270         SUB Tboxoff
3280             Quit$=CHR$(27)&CHR$(127)&"DONE"&CHR$(13)&CHR$(13)
3290             OUTPUT 9;Quit$;
3300             WAIT 1
3310             OUTPUT 9;"QUIET OFF"&CHR$(13);
3320         SUBEND ! Tboxoff
3330         !
3340         SUB Preprint
3350             COM /Lj/ Lf$(1),Cr$(1),Ff$(3),Esc$(1),Portrait$(6),Landscape$
[6],Font$(4)[80],INTEGER Psc,Page,Npages,Fontnum
3360             PRINTER IS Psc
3370             IF Psc=9 THEN
3380                 OUTPUT Psc;Esc$&CHR$(7)&"Q0";! GO TO LASERJET SERIES II EM
ULATION
3390                 WAIT 1.5
3400                 OUTPUT Psc;Font$(Fontnum);
3410                 OUTPUT Psc;Esc$&"&a10L";! SET LEFT MARGIN 10 SPACES
3420                 OUTPUT Psc;Esc$&"&s0C";! SET LINE WRAP
3430             END IF
3440         SUBEND ! Preprint
3450         !
3460         SUB Postprint
3470             COM /Lj/ Lf$(1),Cr$(1),Ff$(3),Esc$(1),Portrait$(6),Landscape$
[6],Font$(4)[80],INTEGER Psc,Page,Npages,Fontnum
3480             IF Psc=CRT THEN
3490                 CALL Continue
3500             ELSE
3510                 OUTPUT Psc;Ff$;
3520             END IF
3530             CALL Beep(5)
3540         SUBEND ! Postprint
3550         !
3560         SUB Llist
3570             COM /Lj/ Lf$(1),Cr$(1),Ff$(3),Esc$(1),Portrait$(6),Landscape$
[6],Font$(4)[80],INTEGER Psc,Page,Npages,Fontnum

```

```

3580          CALL Entint("PRINTER SELECT CODE(1=CRT, 2=THINKJET, 9=LASERJET
) ",Psc)
3590          File$="LADDER"
3600          Page=0
3610          Npages=0
3620          Fontnum=3
3630          CALL Preprint
3640          CAT File$
3650          CALL Postprint
3660          CALL Continue
3670          CALL Preprint
3680          LIST #Psc
3690          CALL Postprint
3700          CALL Beep(5)
3710          SUBEND ! Llist
3720          !
3730          SUB Record(OPTIONAL INTEGER A)
3740          File$="LADDER"
3750          IF NPAR>0 THEN
3760              RE-SAVE File$&"":,702,0".
3770              DISP """"&File$&"" PROGRAM RE-MAVED IN ASCII FORMAT";TAB(8
0)
3780          ELSE
3790              RE-STORE File$&"":,1400"
3800              DISP """"&File$&"" PROGRAM RE-STORED IN INTERNAL FORMAT";T
AB(80)
3810          END IF
3820          CALL Beep(5)
3830          SUBEND ! Record
3840          !
3850          DEF FExist(Name$)
3860              INTEGER Existflg
3870              Existflg=1
3880              ON ERROR RECOVER Er
3890              ASSIGN @P TO Name$
3900 R:          IF Existflg THEN ASSIGN @P TO *
3910              OFF ERROR
3920              RETURN Existflg
3930              !
3940 Er:         Existflg=0
3950              GOTO R
3960              !
3970          FNEED ! FExist
3980          !
3990          SUB Mknet
4000              COM /N/ @Path1,Netfile$[20],Cmnt$(2)[80],Br$(20)[6],INTEGER Nb
,REAL R(20),L(20),C(20),Clk$(80)
4010              CREATE BDAT Netfile$,4
4020          SUBEND ! Mknet
4030          !
4040          SUB Getnfile
4050              COM /N/ @Path1,Netfile$[20],Cmnt$(2)[80],Br$(20)[6],INTEGER Nb
,REAL R(20),L(20),C(20),Clk$(80)
4060              INTEGER Flg

```

```

4070         REPEAT
4080             CALL Entstr("NETWORK FILE NAME",Netfile$)
4090             Flg=FNExist(Netfile$)
4100             IF Flg=0 THEN
4110                 BEEP
4120                 PRINT """;Netfile$;"" DOESN'T EXIST, TRY AGAIN"
4130             END IF
4140         UNTIL Flg
4150         ASSIGN @Path1 TO Netfile$
4160         ENTER @Path1;Cmnt$(*),Nb,Br$(*),R(*),C(*),L(*)
4170         ASSIGN @Path1 TO *
4180     SUBEND ! Getnfile
4190     !
4200     SUB Recnfile
4210         COM /N/ @Path1,Netfile$[20],Cmnt$(2)[80],Br$(20)[6],INTEGER Nb
,REAL R(20),L(20),C(20),Clk$[80]
4220         CALL Entstr("NETWORK FILE NAME",Netfile$)
4230         IF FNExist(Netfile$)=0 THEN CALL Mknet
4240         ASSIGN @Path1 TO Netfile$
4250         OUTPUT @Path1;Cmnt$(*),Nb,Br$(*),R(*),C(*),L(*)
4260         ASSIGN @Path1 TO *
4270     SUBEND ! Recnfile
4280     !
4290     SUB Entnet
4300         COM /N/ @Path1,Netfile$[20],Cmnt$(2)[80],Br$(20)[6],INTEGER Nb
,REAL R(20),L(20),C(20),Clk$[80]
4310         INTEGER I
4320         DIM S$(40),T$(12)
4330     Format1:    IMAGE "BRANCH ",K,#
4340             CALL Cls
4350             CALL Entstr("REM 1",Cmnt$(1))
4360             CALL Entstr("REM 2",Cmnt$(2))
4370             CALL Entint("NUMBER OF BRANCHES",Nb)
4380             FOR I=1 TO Nb
4390                 OUTPUT S$ USING Format1;I
4400                 T$=Br$(I)
4410                 CALL Entstr(S$&" TYPE",T$)
4420                 Br$(I)=T$
4430                 IF T$[1,1]="X" OR T$[1,1]="T" THEN CALL Entreal("TRANSFORME
R TURNS RATIO (NS/NP)",R(I))
4440                 IF POS(T$,"R")<>0 THEN CALL Entreal(S$&" R (KOHM)",R(I),1.E
+3)
4450                 IF POS(T$,"C")<>0 THEN CALL Entreal(S$&" C (nF)",C(I),1.E-9
)
4460                 IF POS(T$,"L")<>0 THEN CALL Entreal(S$&" L (mH)",L(I),1.E-3
)
4470             NEXT I
4480     SUBEND ! Entnet
4490     !
4500     SUB Seg2f(INTEGER S)
4510         COM /C/ REAL F,W,F1,F2,Fs,Supsegs(6,4),INTEGER Nseg,Ns,Tns,Log
flg,Logflags(6),COMPLEX Eb(20),Ib(20),Imb(20),Yb(20),Anet(2,2)
4520         F1=Supsegs(S,1)
4530         F2=Supsegs(S,2)

```

```

4540          Fs=Swpsegs(S,3)
4550          Ns=Swpsegs(S,4)
4560          Logflg=Logflags(S)
4570      SUBEND ! Seg2f
4580      !
4590      SUB F2seg(INTEGER S)
4600          COM /C/ REAL F,W,F1,F2,Fs,Swpsegs(6,4),INTEGER Nseg,Ns,Tns,Log
4610          flg,Logflags(6),COMPLEX Eb(20),Ib(20),Imb(20),Yb(20),Anet(2,2)
4620          Swpsegs(S,1)=F1
4630          Swpsegs(S,2)=F2
4640          Swpsegs(S,3)=Fs
4650          Swpsegs(S,4)=Ns
4660          Logflags(S)=Logflg
4670      SUBEND ! F2seg
4680      !
4680      SUB Entswp(S$)
4690          COM /C/ REAL F,W,F1,F2,Fs,Swpsegs(6,4),INTEGER Nseg,Ns,Tns,Log
4700          flg,Logflags(6),COMPLEX Eb(20),Ib(20),Imb(20),Yb(20),Anet(2,2)
4710          CALL Entint(S$&"SWEEP TYPE (0=LINEAR, 1=LOG)",Logflg)
4720          CALL Entreal(S$&"FIRST FREQUENCY (Hz)",F1)
4730          CALL Entreal(S$&"LAST FREQUENCY (Hz)",F2)
4740          SELECT Logflg
4750          CASE 0 ! LINEAR SWEEP
4760              CALL Entreal(S$&"STEP FREQUENCY (Hz)",Fs)
4770              IF Fs=0 THEN
4780                  CALL Entint(S$&"NUMBER OF STEPS",Ns)
4790                  Fs=(F2-F1)/(N-1)
4800              ELSE
4810                  Ns=((F2-F1)/Fs)+1
4820              END IF
4830          CASE 1 ! LOG SWEEP
4840              CALL Entint(S$&"NUMBER OF STEPS",Ns)
4850              Fs=(F2/F1)^(1/(Ns-1))
4860          END SELECT
4870      SUBEND ! Entswp
4880      !
4880      SUB Compimb(INTEGER Bi)
4890          COM /N/ @Path1,Netfile$(20),Cmnt$(2)[80],Br$(20)[6],INTEGER Nb
4900          ,REAL R(20),L(20),C(20),Clk$(80)
4910          COM /C/ REAL F,W,F1,F2,Fs,Swpsegs(6,4),INTEGER Nseg,Ns,Tns,Log
4920          flg,Logflags(6),COMPLEX Eb(20),Ib(20),Imb(20),Yb(20),Anet(2,2)
4930          INTEGER Ps,Pp,Py,Pz,Pr,P1,Pc
4940          REAL Xr,Xi
4950          Ps=POS(Br$(Bi),"S")
4960          Pp=POS(Br$(Bi),"P")
4970          Pz=POS(Br$(Bi),"Z")
4980          Py=POS(Br$(Bi),"Y")
4990          Pr=POS(Br$(Bi),"R")
5000          P1=POS(Br$(Bi),"L")
5010          Pc=POS(Br$(Bi),"C")
5020          Xr=0
5030          Xi=0
5040          W=(PI+PI)*F
5050          IF Pz<>0 THEN

```

```

5040             IF Pr<>0 THEN Xr=R(Bi)
5050             IF Pl<>0 THEN Xi=Xi+W*L(Bi)
5060             IF Pc<>0 THEN Xi=Xi-1/(W*C(Bi))
5070             END IF
5080             IF Py<>0 THEN
5090                 IF Pr<>0 THEN Xr=1/R(Bi)
5100                 IF Pl<>0 THEN Xi=Xi-1/(W*L(Bi))
5110                 IF Pc<>0 THEN Xi=Xi+W*C(Bi)
5120             END IF
5130             Imb(Bi)=CMPLX(Xr,Xi)
5140             IF (Ps<>0 AND Py<>0) OR (Pp<>0 AND Pz<>0) THEN Imb(Bi)=1/Imb(Bi)
5150             SUBEND ! CompimbEDIT4530
5160             !
5170             SUB Cascade
5180                 COM /N/ @Path1,Netfiles$(20),Cmnts$(2)(80),Br$(20)(6),INTEGER Nb
,REAL R(20),L(20),C(20),Clk$(80)
5190                 COM /C/ REAL F,W,F1,F2,Fs,Swpsegs(6,4),INTEGER Nseg,Ns,Tns,Log
flg,Logflags(6),COMPLEX Eb(20),Ib(20),Imb(20),Yb(20),Anet(2,2)
5200                 INTEGER I
5210                 Anet(1,1)=CMPLX(1,0)
5220                 Anet(1,2)=CMPLX(0,0)
5230                 Anet(2,1)=Anet(1,2)
5240                 Anet(2,2)=Anet(1,1)
5250                 FOR I=Nb TO 1 STEP -1
5260                     CALL Compimb(I)
5270                     SELECT Br$(I)(1,1)
5280                         CASE "T","X"
5290                             Anet(1,1)=Anet(1,1)/R(I)
5300                             Anet(1,2)=Anet(1,2)/R(I)
5310                             Anet(2,1)=Anet(2,1)*R(I)
5320                             Anet(2,2)=Anet(2,2)*R(I)
5330                         CASE "S"
5340                             Anet(1,1)=Anet(1,1)+Imb(I)*Anet(2,1)
5350                             Anet(1,2)=Anet(1,2)+Imb(I)*Anet(2,2)
5360                         CASE "P"
5370                             Anet(2,1)=Anet(2,1)+Imb(I)*Anet(1,1)
5380                             Anet(2,2)=Anet(2,2)+Imb(I)*Anet(1,2)
5390                     END SELECT
5400                 NEXT I
5410             SUBEND ! Cascade
5420             !
5430             SUB Compely
5440                 COM /N/ @Path1,Netfiles$(20),Cmnts$(2)(80),Br$(20)(6),INTEGER Nb
,REAL R(20),L(20),C(20),Clk$(80)
5450                 COM /C/ REAL F,W,F1,F2,Fs,Swpsegs(6,4),INTEGER Nseg,Ns,Tns,Log
flg,Logflags(6),COMPLEX Eb(20),Ib(20),Imb(20),Yb(20),Anet(2,2)
5460                 INTEGER I,Im1
5470                 Yb(Nb)=CMPLX(0,0)
5480                 Ib(Nb)=Yb(Nb)
5490                 Eb(Nb)=CMPLX(1,0)
5500                 FOR I=Nb TO 1 STEP -1
5510                     Im1=I-1
5520                     CALL Compimb(I)

```

```

5530         SELECT Br$(I)[1,1]
5540         CASE "T","X"
5550             Eb(Im1)=Eb(I)/R(I)
5560             Ib(Im1)=Ib(I)*R(I)
5570         CASE "S"
5580             Eb(Im1)=Eb(I)+Ib(I)*Imb(I)
5590             Ib(Im1)=Ib(I)
5600         CASE "P"
5610             Eb(Im1)=Eb(I)
5620             Ib(Im1)=Eb(I)*Imb(I)+Ib(I)
5630         END SELECT
5640     NEXT I
5650 SUBEND ! Compeiy
5660     !
5670 SUB Prth
5680     COM /N/ @Path1,Netfile$(20),Cmnt$(2)[80],Br$(20)[6],INTEGER Nb
,REAL R(20),L(20),C(20),Cik$(80)
5690     COM /Lj/ Lf$(1),Cr$(1),Ff$(3),Esc$(1),Portrait$(6),Landscape$(
[6],Font$(4)[80],INTEGER Psc,Page,Npages,Fontnum
5700     IF Page>0 THEN
5710         IF Npages=0 THEN
5720             PRINT Cik$[1,61+10*(F>2)]&"PAGE "&VAL$(Page)
5730         ELSE
5740             PRINT Cik$[1,55+10*(F>2)]&"PAGE "&VAL$(Page)&" OF "&VAL$(
(Npages)
5750         END IF
5760     END IF
5770     IF NOT (Cmnt$(1)[1,1]=" " OR LEN(Cmnt$(1))=0) THEN PRINT Cmnt$(
(1)
5780     IF NOT (Cmnt$(2)[1,1]=" " OR LEN(Cmnt$(2))=0) THEN PRINT Cmnt$(
(2)
5790     PRINT
5800 SUBEND ! Prth
5810     !
5820 SUB Prtnet
5830     COM /N/ @Path1,Netfile$(20),Cmnt$(2)[80],Br$(20)[6],INTEGER Nb
,REAL R(20),L(20),C(20),Cik$(80)
5840     COM /Lj/ Lf$(1),Cr$(1),Ff$(3),Esc$(1),Portrait$(6),Landscape$(
[6],Font$(4)[80],INTEGER Psc,Page,Npages,Fontnum
5850     INTEGER I
5860 Fmt1:     IMAGE "BR   TYPE           R (KOHM)           L (mH)           C
(nF)  TURNS RATIO(S/P)"
5870 Fmt2:     IMAGE DD,3X,AAAAAA,#
5880 Fmt3:     IMAGE 6X,DDDDD.DDDDD,#
5890 Fmt4:     IMAGE "      -----",#
5900 Fmt5:     IMAGE 4X,DDDDD.DDDDD,#
5910 Fmt6:     IMAGE 4X,"-----",#
5920 Fmt7:     IMAGE "PRINTING PAGE ",K," OF ",K
5930     IF Psc=9 THEN
5940         Fontnum=3
5950     ELSE
5960         Fontnum=0
5970     END IF
5980     CALL Preprint

```

```

5990         Page=1
6000         Npages=1
6010         CALL Prth
6020         IF Psc<>CRT THEN OUTPUT CRT USING Fmt7;Page,Npages
6030         PRINT USING Fmt1
6040         FOR I=1 TO Nb
6050             PRINT USING Fmt2;I,Br$(I)
6060             IF POS(Br$(I),"R")<>0 THEN
6070                 PRINT USING Fmt3;R(I)*1.E-3
6080             ELSE
6090                 PRINT USING Fmt4
6100             END IF
6110             IF POS(Br$(I),"L")<>0 THEN
6120                 PRINT USING Fmt3;L(I)*1.E+3
6130             ELSE
6140                 PRINT USING Fmt4
6150             END IF
6160             IF POS(Br$(I),"C")<>0 THEN
6170                 PRINT USING Fmt3;C(I)*1.E+9
6180             ELSE
6190                 PRINT USING Fmt4
6200             END IF
6210             IF POS(Br$(I),"T")<>0 OR POS(Br$(I),"X")<>0 THEN
6220                 PRINT USING Fmt5;R(I)
6230             ELSE
6240                 PRINT USING Fmt6
6250             END IF
6260             PRINT
6270         NEXT I
6280         CALL Postprint
6290         SUBEND 1 Prtnet
6300         !
6310         SUB Prtrat
6320             COM /N/ @Path1,Netfile$(20),Cmnt$(2)[80],Br$(20)[6],INTEGER Nb
,REAL R(20),L(20),C(20),Clk$(80)
6330             COM /C/ REAL F,W,F1,F2,Fs,Swpsegs(6,4),INTEGER Nseg,Ns,Tns,Log
flg,Logflags(6),COMPLEX Eb(20),Ib(20),Imb(20),Yb(20),Anet(2,2)
6340             COM /Lj/ Lf$(1),Cr$(1),Ff$(3),Esc$(1),Portrait$(6),Landscapes$
[6],Font$(4)[80],INTEGER Psc,Page,Npages,Fontnum
6350             INTEGER I,K,S,Md,Im1
6360             REAL Zm,Za,Rm,Ra,Np
6370             COMPLEX T
6371         ! Addition by Mitch Bolling to store data for plotting by PLOTTER
6372         ! program - Dec 1988.
6373         INTEGER Nbytes,Numrec
6374         REAL Npts
6380 Fmt1:         IMAGE " #          FREQ(Hz)  EOUT/EIN(dB) EOUT/EIN(DEG)  ZINMAG(K
OHM)  ZINANG(DEG)"
6390 Fmt2:         IMAGE ODD,5(2X,ODDDDDDD.DDD)
6400 Fmt3:         IMAGE "PRINTING PAGE ",K," OF ",K
6401         !
6403         !   *** plot patch ***
6404 Askplot:     INPUT "Do you wish to create a PLOTTER file? (Y/N) ",Plots
6405         Pflag=0

```

```

6406             IF (Plot$="Y" OR Plot$="y") THEN
6407                 Pflag=1
6408             ELSE
6409                 IF Plot$<>"N" AND Plot$<>"n" THEN Askplot
6410             END IF
6411 ! *****
6412 !
6414             IF Psc=9 THEN
6420                 Fontnum=3
6430             ELSE
6440                 Fontnum=0
6450             END IF
6460             CALL Preprint
6470             Page=1
6480             Np=Tns/(10+35*(Psc=2)+55*(Psc=9))
6490             IF Np<1 THEN Np=1
6500             Npages=INT(Np)
6510             IF Np MOD Npages<>0 THEN Npages=Npages+1
6520             K=1
6521 ! *** plot patch ***
6522 !
6523             IF Pflag THEN
6524                 Npts=0
6526                 FOR I=1 TO Nseg
6527                     CALL Seg2f(I)
6528                     Npts=Npts+Ns
6529                 NEXT I
6531                 Nbytes=(INT((Npts*2*8)/256)+1)*256
6533                 ASSIGN @Buffer TO BUFFER [Nbytes];FORMAT OFF
6534                 Num_rec=INT(((2*Npts+1)*8)/820)
6535                 IF (((2*Npts+1)*8) MOD 820)<>0 THEN Num_rec=Num_rec+1
6536                 INPUT "Enter the file name for the Plot: ",Pfile$
6537                 CREATE BDAT Pfile$,Num_rec,8200
6538                 ASSIGN @Path_1 TO Pfile$
6539                 OUTPUT @Path_1;Npts
6540                 TRANSFER @Buffer TO @Path_1;COUNT (Npts*2*8),CONT
6541             END IF
6542 ! *****
6543             FOR S=1 TO Nseg
6544                 CALL Seg2f(S)
6550                 F=F1
6560                 FOR I=1 TO Ns
6570                     ImI=I-1
6580                     Md=K MOD (10+40*(Psc=2)+55*(Psc=9))
6590                     IF Md=1 THEN
6600                         CALL Prth
6610                         IF Psc<>CRT THEN OUTPUT CRT USING Fmt3;Page,Npages
6620                         PRINT USING Fmt1
6630                         Page=Page+1
6640                     END IF
6650                 CALL Compey
6660                 T=Eb(0)/Ib(0)
6670                 Rec2pol(REAL(T),IMAG(T),Zm,Za)
6680                 T=Eb(Nb)/Eb(0)

```

```

6690                               Rec2pol(REAL(T),IMAG(T),Rm,Ra)
6700                               PRINT USING Fmt2;K,F,FNDb(Rm),Ra,1.E-3*Zm,Za
6701 !
6703 !       *** plotter patch ***
6704             IF Pflag THEN
6705                 OUTPUT @Buffer;F,FNDb(Rm)
6707             END IF
6709 !       *****
6710 !
6711             IF Logflg THEN
6720                 F=F+Fs
6730             ELSE
6740                 F=F+Fs
6750             END IF
6760             IF K MOD 5=0 THEN
6770                 PRINT
6780                 WAIT .5
6790             END IF
6800             IF Md=0 THEN
6810                 IF Psc=CRT THEN
6820                     CALL Continue
6830                 ELSE
6840                     OUTPUT Psc;Ff$;
6850                     WAIT 1
6860                 END IF
6870             END IF
6880             K=K+1
6890         NEXT I
6900     NEXT S
6901 !
6903 !       *** plotter patch ***
6904             CONTROL @Buffer,9;0
6905             WAIT FOR EOT @Path_1
6906             ASSIGN @Path_1 TO *
6907             ASSIGN @Buffer TO *
6908 !       *****
6909 !
6910             CALL Postprint
6920             SUBEND ! Prtrat
6930             !
6940             SUB Prtanet
6950             COM /N/ @Path1,Netfiles$(20),Cmnt$(2)(80),Br$(20)(6),INTEGER Nb
,REAL R(20),L(20),C(20),Cik$(80)
6960             COM /C/ REAL F,W,F1,F2,Fs,Swpsegs(6,4),INTEGER Nseg,Ns,Tns,Log
flg,Logflags(6),COMPLEX Eb(20),Ib(20),Imb(20),Yb(20),Anet(2,2)
6970             COM /Lj/ Lf$(1),Cr$(1),Ff$(3),Esc$(1),Portrait$(6),Landscap$(
6),Font$(4)(80),INTEGER Psc,Page,Npages,Fontnum
6980             DIM S$(80)
6990             INTEGER I,Indx,S,Md,Iml,J,K
7000             REAL Xm,Xa,Xr,Xi,Np
7010             COMPLEX T
7020 Fmt1:             IMAGE "F(Hz) = ",2X,DDDDDDDD.DD,"           MAG           ANG(DEG)
REAL                 IMAG"
7030 Fmt2:             IMAGE 14X,"A[" ,0," ,". ,0," ]",4(2X,SD.DDDDDDESZZ)

```

```

7040 Fmt3:      IMAGE "PRINTING PAGE ",K," OF ",K
7050            IF Psc=9 THEN
7060              Fontnum=3
7070            ELSE
7080              Fontnum=0
7090            END IF
7100            CALL Preprint
7110            Page=1
7120            Np=Tns/(3+8*(Psc=2)+12*(Psc=9))
7130            Npages=INT(Np)
7140            IF Np MOD Npages<>0 THEN Npages=Npages+1
7150            Indx=1
7160            FOR S=1 TO Nseg
7170              Seg2f(S)
7180              F=F1
7190              FOR I=1 TO Ns
7200                Md=Indx MOD (3+8*(Psc=2)+12*(Psc=9))
7210                IF Md=1 THEN
7220                  CALL Prth
7230                  IF Psc<>CRT THEN OUTPUT CRT USING Fmt3;Page,Npages
7240                  Page=Page+1
7250                END IF
7260                CALL Cascade
7270                PRINT USING Fmt1;F
7280                FOR J=1 TO 2
7290                  FOR K=1 TO 2
7300                    Xr=REAL(Anet(J,K))
7310                    Xi=IMAG(Anet(J,K))
7320                    Rec2pol(Xr,Xi,Xm,Xa)
7330                    PRINT USING Fmt2;J,K,Xm,Xa,Xr,Xi
7340                  NEXT K
7350                NEXT J
7360                IF Logflg THEN
7370                  F=F*Fs
7380                ELSE
7390                  F=F+Fs
7400                END IF
7410                IF Md=0 THEN
7420                  IF Psc=CRT THEN
7430                    CALL Continue
7440                    PRINT
7450                  ELSE
7460                    OUTPUT Psc;Ff$;
7470                  END IF
7480                END IF
7490                Indx=Indx+1
7500              NEXT I
7510            NEXT S
7520            CALL Postprint
7530            SUBEND ! Prtanet
7540            !
7550            SUB Prtlev
7560            COM /N/ @Path1,Netfile$(20),Cmnt$(2)[80],Brs(20)[6],INTEGER Nb
,REAL R(20),L(20),C(20),Clk$(80)

```

```

7570          COM /C/ REAL F,W,F1,F2,Fs,Swps(6,4),INTEGER Nseg,Ns,Tns,Log
flg,Logflags(6),COMPLEX Eb(20),Ib(20),Imb(20),Yb(20),Anet(2,2)
7580          COM /Lj/ Lf$(1),Cr$(1),Ff$(3),Esc$(1),Portrait$(6),Landscapes$
[6],Font$(4)[80],INTEGER Psc,Page,Npages,Fontnum
7590          INTEGER Indx,I,J,S,Md,Iml,Lpi,Ipp
7600          REAL Em,Ea,Im,Ia,Zm,Za,Np
7610          COMPLEX T
7620 Fmt1:     IMAGE "FREQ (Hz) = ",K
7630 Fmt2:     IMAGE " # BRANCH      EMAG(U)  EANG(DEG)   IMAG(mA)  IANG(DEG
) ZMAG(KOHM)  ZANG(DEG)"
7640 Fmt3:     IMAGE DD,2X,AAAAAA,2X,DDDDDD.DDD,2X,DDDDD.DDD,2X,DDDDDD.DDD,2X
,DDDDD.DDD,2X,DDDDDD.DDD,2X,DDDDD.DDD
7650 Fmt4:     IMAGE DD,2X,AAAAAA,2X,DDDDDD.DDD,2X,DDDDD.DDD,2X,DDDDDD.DDD,2X
,DDDDD.DDD
7660 Fmt5:     IMAGE "PRINTING PAGE ",K," OF ",K
7670          IF Psc=9 THEN
7680             Fontnum=3
7690          ELSE
7700             Fontnum=0
7710          END IF
7720          CALL Preprint
7730          Page=1
7740          Lpi=Nb+4
7750          Ipp=INT(((14+42*(Psc=2)+64*(Psc=9))/Lpi)
7760          Np=Tns/Ipp
7770          Npages=INT(Np)
7780          IF Np MOD Npages<>0 THEN Npages=Npages+1
7790          Indx=1
7800          FOR S=1 TO Nseg
7810             CALL Seg2f(S)
7820             F=F1
7830             FOR I=1 TO Ns
7840                Md=(Indx MOD Ipp)+(Ipp=1)
7850                IF Md=1 THEN
7860                   CALL Prth
7870                   IF Psc<>CRT THEN OUTPUT CRT USING Fmt5;Page,Npages
7880                   Page=Page+1
7890                END IF
7900                CALL Compeiy
7910                PRINT USING Fmt1;F
7920                PRINT USING Fmt2
7930                FOR J=0 TO Nb
7940                   Rec2pol(REAL(Eb(J)),IMAG(Eb(J)),Em,Ea)
7950                   Rec2pol(REAL(Ib(J)),IMAG(Ib(J)),Im,Ia)
7960                   IF Ib(J)<>CMLPX(0,0) THEN
7970                      T=Eb(J)/Ib(J)
7980                      Rec2pol(REAL(T),IMAG(T),Zm,Za)
7990                      PRINT USING Fmt3;J,Br$(J),Em,Ea,1.E+3*Im,Ia,1.E-3*
Zm,Za
8000                ELSE
8010                   PRINT USING Fmt4;J,Br$(J),Em,Ea,1.E+3*Im,Ia
8020                END IF
8030            NEXT J
8040          IF Logflg THEN

```

```

8050             F=F*Fs
8060             ELSE
8070             F=F+Fs
8080             END IF
8090             IF Ipp=1 THEN Md=0
8100             IF Md=0 THEN
8110                 IF Psc=CRT THEN
8120                     CALL Continue
8130                 ELSE
8140                     OUTPUT Psc;Ff$;
8150                 END IF
8160             ELSE
8170                 PRINT
8180             END IF
8190             Indx=Indx+1
8200         NEXT I
8210     NEXT S
8220     CALL Postprint
8230 SUBEND ! Prtlev
8240     !
8250 SUB Entsupsegs
8260     COM /C/ REAL F,W,F1,F2,Fs,Supsegs(6,4),INTEGER Nseg,Ns,Tns,Log
8270     flg,Logflags(6),COMPLEX Eb(20),Ib(20),Imb(20),Yb(20),Anet(2,2)
8280     INTEGER S
8290     DIM S$(80)
8300     IMAGE "SEGMENT ",D,X,#
8310     Tns=0
8320     CALL Entint("NUMBER OF SWEEP SEGMENTS",Nseg)
8330     FOR S=1 TO Nseg
8340         CALL Seg2f(S)
8350         OUTPUT S$ USING Fmt1;S
8360         CALL Entswp(S$)
8370         Tns=Tns+Ns
8380         CALL F2seg(S)
8390     NEXT S
8400 SUBEND ! Entsupsegs

```