AD-A101 053 TEXAS TECH UNIV LUBBOCK INST FOR ELECTRONIC SCIENCE F/G 9/3
NONLINEAR FAULT DIAGNOSIS.(U)
MAY 81 R - LIU, K NAKAJIMA, P OLIVIER

UNCLASSIFIED NL

NONLINEAR FAULT DIAGNOSIS

R.-w. Liu

K. Nakajima

P. Olivier

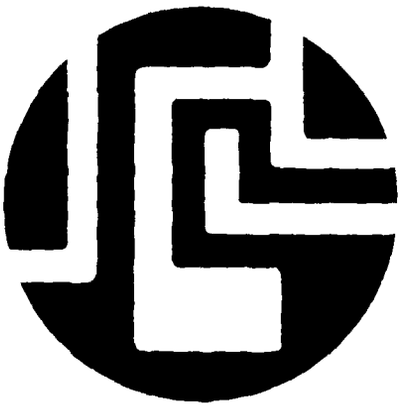
Quoc Ding Ngo

R. Saeks

A. Sangiovanni-Vincentelli

C.L. Wey

C.-c. Wu



Institute for
Electronic Science

TEXAS TECH UNIVERSITY
Lubbock, Texas 79409

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A101053	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Nonlinear Fault Diagnosis		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) R.-w. Liu, K. Nakajima, P. Olivier, Quoc Ding Ngo, R. Saeks, A. Sangiovanni-Vincentelli, C.L. Wey, and C.-c. Wu.		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Texas Tech University Dept. of Electrical Engineering Lubbock, Texas 79409		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research 800 N. Quincy Street Arlington, VA. 22217		12. REPORT DATE May 1981
		13. NUMBER OF PAGES 200
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Fault Analysis, Automated Maintenance, Fault Diagnosis, Fault Isolation, Test Points, Test Vectors

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A summary of several research projects in the nonlinear fault diagnosis is given. Several alternative algorithms for the solution of the nonlinear fault diagnosis problem are presented, together with a diagnosibility theory, and a set of criteria which an "ideal" fault diagnosis problem should strive to meet.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

NONLINEAR FAULT DIAGNOSIS

CONTENTS

FAULT DIAGNOSIS IN ELECTRONIC CIRCUITS

R. SAEKS AND R.-W. LIU

Fault Diagnosis in Electronic Circuits

R. Saeks
Department of Electrical Engineering
Texas Tech University
Lubbock, Texas 79409

R.-w. Liu
Department of Electrical Engineering
Notre Dame University
Notre Dame, IN. 46556

During the past quarter century the engineering community has been witness to tremendous strides in the art of electronics design. The graphical algorithms of the previous generation have given way to the modern CAD package, the breadboard has been subsumed by the simulator. Indeed, even the universal building block has become a reality. To the contrary electronics maintenance has changed little since the day of the vacumm tube, remaining the responsibility of the experienced technician with scope and multimeter. As such, our ability to design a complex electronic circuit is quickly out-distancing our ability to maintain it. In turn, the price reductions which have accompanied modern electronics technology have been paralleled by increasing maintenance and operations costs. Indeed, many industries are finding that the life cycle maintenance costs for their electronic equipment now exceed their original capitol investment.

Given the above, it is quickly becoming apparent that the electronics maintenance process, like the design process, must be automated. Unfortunately, the 50 years of progress in circuit theory, on which our electronics design automation has been predicated, does not exist in the maintenance area. As such, the past decade has witnessed the inauguration of a basic research program to lay the foundations for a theory of electronic maintenance and a parallel effort to develop operational electronic maintenance codes.

Thus far the greatest success has been achieved in the digital electronic area, wherein the finite state nature of the UUT (unit under test) may be exploited.[3] Typically, one assumes that all failures manifest themselves in the form of component outputs which are either "stuck-at-one" or "stuck-at-zero" and/or shorts and opens.[5] Under such an assumption a theory for digital system maintenance has been developed and practical fault diagnosis algorithms are in the formative stages of development. Typically, one hypothesizes some limit on the number of simultaneous faults and then simulates the responses of the UUT to a family of test vectors for each allowed combination of faults. The actual responses of the UUT are then compared with the simulated responses to locate the failure. Although lacking in asthetic appeal the above approach, termed *fault simulation*, is ideally suited for the maintenance environment, wherein, the actual simulation process need only be done once at the factory or a maintenance depot with the simulated response data being distributed via magnetic tape to the various field locations where the actual test is conducted. As such, with the aid of some sophisticated software engineering, this apparently "brute force" approach to the fault diagnosis problem has slowly evolved into a workable concept.[4] Indeed, at the present time a number of automatic test program generators which classify faults, choose test vectors, and carry out the appropriate simulation (often in a parallel processing mode), are commercially available and, as such, the automated maintenance of digital electronic circuits is becoming a reality.[4]

Unfortunately, the above described success in the digital world has not been paralleled by progress in the analog world. Indeed, test engineers

complain that while 80% of the boards are digital, 80% of their headaches are analog and hybrid. This difficulty arises from a number of characteristics of the analog problem which are not encountered in digital circuits. Indeed, in an analog circuit:

(i) there is a continuum of possible failures,

(ii) a component may be "in tolerance" but not nominal,

(iii) complex feedback structures are encountered,

(iv) simulation is slow and costly,

(v) post-fault component characteristics may not be known,

(vi) and a fault in one component may induce an apparent fault in another.

Items (i) and (ii) imply that an extremely large number of simulations will be required for analog testing. Items (iii) and (iv) suggest that these simulations will be far more expensive than similar digital simulations. Finally, items (v) and (vi) indicate that the simulation of a post-fault circuit by itself may not be a tractable problem. As such, it is by no means clear that the kind of "brute force" fault simulation algorithm associated with the digital problem will be applicable to the analog or hybrid case.

As an alternative to fault simulation, a number of academic researchers have proposed a variety of "post test" fault diagnosis algorithms, wherein, an "equation solving like" algorithm is used to locate the faulty component given the test data from UUT.[2,8] Although these algorithms are, in some sense, "smarter" than the simulation algorithms, most of the required computing must be done in the field after the UUT has been tested. Moreover, these computational requirements must be replicated each time a unit fails.

As such, the success of such "post test" algorithms is contingent on reducing their computational requirements to a bare minimum. Although no system is yet operational, with the aid of the powerful linear circuit theory developed over the past half century, a computationally efficient solution to the fault diagnosis problem for linear analog circuits appears to be within reach.[1,2] Unfortunately, no such light exists at the end of the nonlinear tunnel, wherein progress appears to be limited by a "computational complexity/test point" bound.

Not surprisingly, the computational cost of an analog fault diagnosis algorithm is an inverse function of the number of test points at which measurements of the UUT may be made. Indeed, if one lets n be a measure of UUT complexity (which may loosely be taken to be the total number of terminals for all of the circuit components), then if one has access to $O(n)(\dagger^1)$ test points the fault diagnosis problem can be resolved using linear algorithms.[7,10] Moreover, by combining such algorithms with the above mentioned linear algorithms, acceptable computational efficiency can be obtained with O(m) test points where m is a measure of the complexity of the "nonlinear subsystem" of the UUT.[6,7] Although such algorithms can be effective on the typical academic example a "real world" PC (printed circuit) board does not have terminal space for the 20 or 30 test points which are required even for a routine board made up of discrete components and/or SSI (Small Scale Integration). Although the problem can be partially alleviated by making internal measurements with the aid of a "bed-of-nails" tester it has been our experience that such testers cause as many failures as they locate while their applicability to two-sided, multilayer, and

† $f(n) = O(n)$ means f increases in the order of n; more precisely, $|f(n)| \leq c|n|$ for some $c > 0$.

coated boards is severly limited. As such, we would like to limit the number of test points required by an analog fault diagnosis algorithm should inc ease at a rate of no greater than $O(n^{1/2})$. A further study of the possible tradeoff between test points and computational cost appears in references 11 and 12.

Unfortunately, all computationally acceptable "post test" algorithms which have thus far been proposed have test point requirements which grow linearly with UUT complexity (assuming that m grows linearly with n). As such, many researchers are looking at the classical fault simulation algorithms with renewed vigor. Indeed, these algorithms have minimal on-line computational costs, while the number of test points employed can easily be kept below $O(n^{2/1})$. The difficulty lies with the required number of simulations and the development of decision algorithms which will allow us to "interpolate" between simulated data points.

Thus, while the state-of-the-art in digital diagnosis is fast maturing, a serious investigation of analog fault diagnosis problems is only just beginning. Indeed, a satisfactory fault diagnosis code for linear analog circuits has yet to be demonstrated while the nonlinear problem has yet to progress beyond the basic research stage.

References

1. Chen, H.S.M. and R. Saeks,"A Search Algorithm for the Solution of the Multifrequency Fault Diagnosis Equations", IEEE Trans. on Circuits and Systems, CAS-26, 589-594 (1979).

2. Duhamel, P. and J.C. Rault, "Automatic Test Generation Techniques for Analog Circuits - A Review", IEEE Trans on Circuits and Systems, 25, 411-439, (1979).

3. Friedman, A.D. and Memon, P.R., Fault Detection in Digital Circuits, New York, Prentice Hall (1971).

4. Greenbaum, J.R., "Computer-Aided Fault Analysis - Today, Tomorrow, or Never", in Rational Fault Analysis (ed. R. Saeks, and S.R. Liberty), New York, Marcel Dekker, 95-111 (1977).

5. Hayes, J.P., "Modeling Faults in Digital Circuits", in Rational Fault Analysis, (ed. R. Saeks, and S.R. Liberty), New York, Marcel Dekker, 78-95 (1977).

6. Hsieh, M., Ph.D. Dissertation, Texas Tech Univ., 1980.

7. Ngo, Q.-D., M.S. Thesis, Texas Tech Univ. (1980).

8. Plice, W.A., "Automatic Generation of Fault Isolation Tests for Analog Circuit Boards-A Survey", presented at ATEX EAST 78, Boston, Sept. (1978).

9. Saeks, R. Singh, S.P. and R.W. Liu, "Fault Isolation via Components Simulation", IEEE Trans. on Circuit Theory, CT-19, 634-640 (1972).

10. Trick, T.N., Mayeda, W., and A.A. Sakla, "Calculation of Parameter Values from Node Measurements", IEEE Trans. on Circuits and Systems, CAS-26, 466-474 (1979).

11. R.W. Liu and V. Visvanathan, "Sequentially Linear Fault Diagnosis: Part I - Theory", IEEE Trans. on Circuits and Systems, 490-496, July (1979).

12. V. Visvanathan and R.W. Liu, "Sequentially Linear Diagnosis: Part II - The Design of Diagnosable Systems", IEEE Trans. on Circuits and Systems, 558-564, July (1979).

FAULT DIAGNOSIS - A NONLINEAR SYSTEMS PROBLEM

R. SAEKS

Fault Diagnosis - A Nonlinear Systems Problem

R. Saeks
Department of Electrical Engineering
Texas Tech University
Lubbock, Texas 79409

Summary

Conceptually, the fault analysis problem for an analog circuit or system amounts to the measurement of a set of externally accessible parameters of the system from which one desires to determine the internal system parameters or equivalently* locate the failed components as illustrated in Figure 1. Here, the

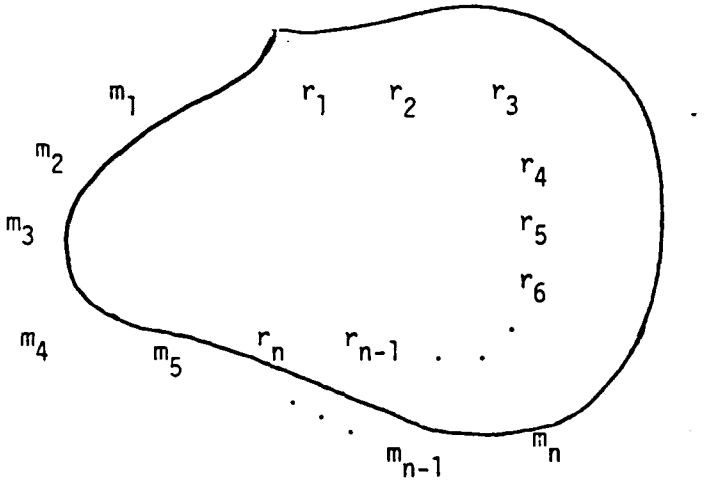


Figure 1. Conceptual Model of Fault Diagnosis Problem.

measurements, m_i, may represent data taken at distinct test points or alternatively, data taken at a fixed test point under different stimuli.

*Since the problem of determining the values of the failed components is usually straightforward, once the failures have been located, the exact determination of all internal component parameters is essentially equivalent to the problem of "simply" locating the failed components.

Similarly, the r_i represent parameters characterizing the various internal system components. Here, a single parameter may characterize an entire component, say a resistance, capacitance or inductance. Alternatively, a component may be represented by several parameters: the h-parameters of a transistor, the poles and gain of an op-amp, etc. In general, one models a system component by the minimum number of parameters which will allow the failure to be isolated up to a "shop replaceable assembly" with all "allowed" system failures manifesting themselves in the form of some parameter change.

To solve the fault diagnosis problem, one then measures $m = \text{col}(m_i)$ and solves a nonlinear algebraic equation

$$m = F(r) \tag{1}$$

for $r = \text{col}(r_i)$ to diagnose the fault. Note, the function, F, is nonlinear even for linear systems, however, for linear time-invariant systems the function, F, can be expressed analytically. More generally, in the nonlinear case, one can evaluate $F(r)$ for any given parameter vector, r, with a simulator, and thus solve 1. numerically, even though F has no analytic expression.

Although one does not usually formulate the fault diagnosis problem in terms of the above described equation solving notation, this formulation is equivalent to the classical fault simulation concept. Indeed, fault simulation is simply a search algorithm for solving 1. Here, one precomputes $\hat{m} = f(\hat{r})$ for each allowable[#] faulty parameter vector $\hat{r}$ and then compares the measured m with the simulated $\hat{m}$'s, stored in a fault dictionary, to solve equation 1.

[#]By allowable faults we mean all possible parameter vectors, $\hat{r}$, which satisfy a specified set of fault hypotheses. These typically restrict the maximum number of component parameters which are simultaneously out of tolerance and the type of failure (open circuit, short circuit, small change, etc.)

Although the above described approach to fault simulation has been successful** when applied to digital system, there is considerable question surrounding its applicability to analog circuits and systems. The problem is two-fold. First, rather than simply failing as a one or zero, an analog parameter has a continuum of possible failures. Secondly, unlike a digital system wherein a component is either good or bad, in an analog system, a component parameter is either in tolerance or out of tolerance. As such, for each hypothesized failure, it may prove necessary to do an entire family of Monte Carlo simulations in which the values of the good components are randomly chosen within their tolerance limits. Although, at the present time we have insufficient practical experience to determine the precise number of fault simulations required for analog fault diagnosis, it is estimated that the number of simulations required for an analog system will exceed the number of simulations required for a digital system of similar complexity by a factor ranging between two and six order of magnitude. As such, the fault simulation concept which has proven to be so successful for a digital system may not be applicable in the analog case.

As an alternative to fault simulation, one may adopt one of the more classical equation solving algorithms for the solution of 1. Here, one first measures m and on the basis of this measurement, makes an initial guess r^0 (usually taken to be nominal parameter vector) at the solution of the equations. One then evaluates $m^0 = F(r^0)$ and compares it with m. If $m^0 = m$, r^0 is the solution to the fault diagnosis equation. If not, one makes a new "educated" guess at the solution, r^1, (usually based on the deviation between m and m^0)

**Most industrial users of ATE obtain satisfactory fault detection in digital circuits via fault simulation techniques but require guided probe techniques in addition to the fault dictionary data for fault diagnosis (isolation).

and one repeats the process by evaluating $m^1 = F(r^1)$ and comparing it with m. Hopefully, sequence of component parameter vectors, r^i, simulated data vectors, $m^i = F(r^i)$, is obtained, which "quickly" converges to r and m, respectively. Since the evaluation of $F(r^i)$ is essentially equivalent to the simulation of the system with the faulty parameter values, r^i, this technique is really another form of fault simulation. In this case, however, one simulates the system after the data vector has been measured and uses this data to make an educated guess at a (hopefully) small number of parameter vectors at which the system should be simulated. As such, the approach has been termed simulation after test to distinguish it from the classical approach, wherein all simulation is done before test.

At the time of this writing, both approaches are under study, neither of which have been shown to be superior. Fault "simulation after test" requires that one include an efficient simulator in the ATE itself, which can be used for on-line computation of $m^i = F(r_i)$ after the UUT has been measured. On the ohter hand, simulation after test eliminates the requirement of searching a large fault dictionary for the (approximate) data matches required by "simulation before test". In addition, the complex ATPG requirement for "simulation before test" is eliminated.

To make "simulation after test" feasible, however, an efficient equation solving algorithm is required to obtain convergence of the r^i sequence in a reasonable amount of time. Moreover, since "real world" failures in analog circuits and systems often take the form of open and short circuited components or large parameter diviations from nominal, the classical perturbational algorithms a-la Newton-Raphson are inapplicable. Fortunately, in the context

of the fault diagnosis problem, one can reasonably assume that relatively few component parameters have failed. As such, even though it is not valid to assume that $r-r^0$ (the deviation of r from nominal) is small in norm, it is reasonable to assume that it is small in "rank".

CRITERIA FOR ANALOG FAULT DIAGNOSIS

R. SAEKS

Criteria for Analog Fault Diagnosis*

R. Saeks
Department of Electrical Engineering
Texas Tech University
Lubbock, Texas 79409

Introduction

After a half century of neglect by the electronics community the past decade has witnessed an expanding effort in the analog fault diagnosis area. Indeed, the ever increasing complexity of electronic circuits combined with the decreasing availability of trained maintenance technicians has pushed *computer-aided testing* (CAT) to the forefront of electronics research. Unfortunately, the tremendous strides which have been made in digital test technology have not been paralleled by equal progress in the analog area. As such, even though "80% of the boards are digital 80% of the problems are analog".

The lack of progress in analog CAT vis-a-vis digital CAT may be attributed to four factors:

i). the cost of analog circuit simulation,

ii). the continuous nature of analog failure phenomena,

iii). tolerances on the "good" components in analog circuit,

iv). and the lack of viable models for the components in a faulty circuit.

Moreover, these difficulties have been exaggerated by the economics of the maintenance environment which limits the degree to which many of the classical tools of analog circuit design can be used in a CAT package.

The purpose of the present paper is to describe a set of criteria which we believe a practical analog CAT algorithm should achieve and to indicate the degree to which they are met by the various algorithms which have thus

* This research supported in part by the Joint Services Electronic Program at Texas Tech University under ONR Contract 76-C-1136.

far been proposed.[1] These criteria include computational requirements, numbers of test points and test vectors employed, robustness to tolerance effects, availability of models, and the degree to which the algorithm is amenable to parallel processing. Although many specific algorithms have been proposed they may naturally be classified into three categories:

i). simulation-before-test,

ii). simulation-after-test with a single test vector,

iii). and simulation-after-test with multiple test vectors.

Each of these three approaches to the analog CAT problem is compared against our criteria, and, interestingly, each approach fails to meet at least one of the proposed criteria.

Criteria

A. Computational Requirements: Unlike a CAD algorithm which is used only in the initial design of a circuit or system, a CAT algorithm lives in an operational environment and thus must be used repeatedly each time a system fails. As such, a viable measure for the computational cost of a CAT algorithm must distinguish between on-line computation which is done in the field and must be repeated for each unit under test (UUT) and off-line computation which is independent of the unit under test and thus need only be done once at the factory or a maintenance depot. Indeed, the distinction between on-line and off-line computation is further exaggerated by high cost of computing and the dearth of trained personnel in a field maintenance environment vis-a-vis that is available at a maintenance depot. Thus in a CAT algorithm a *great priority must be placed on reducing the on-line computational requirements* even at the cost of significantly increasing the off-line computation. As such, an algorithm which is viable in a design environment might not be acceptable in a maintenance environment and vice-versa. Indeed, in a

CAT algorithm one would be happy to accept the cost of generating a complex data base in an off-line environment to achieve a reduction in on-line computational requirements.

B. Test Points: Historically, analog circuits have been tested with the aid of a "bed of nails" tester which allows one to make use of test data which is not accessible via the input and output terminals of the circuit board. Unfortunately, modern circuit boards are often multilayered and/or coated, thereby limiting the applicability of the "bed of nails" concept. As such, a modern CAT algorithm must be designed to work with the test data which is available at the externally accessible terminals of a printed circuit board. In practice, this proves to be a dominating factor in the design of a CAT package, which precludes the use of some of the more attractive algorithms with test point requirements which grow linearly with circuit complexity. In fact, circuit complexity is proportional to the area of a printed circuit board (if not a power thereof) while the number of accessible test points is proportional to the edge length of the board. As such, in a practical CAT package it is reasonable to require that the *number of test points grow with the square root of circuit complexity* (or less).

C. Robustness: Unlike a digital system wherein a device is either good or bad in an analog circuit a device is either "in-tolerance" or "out-of-tolerance" and, as such, an analog CAT algorithm must be able to cope with the effects of *components which are in-tolerance but not nominal*. Although, at the time of this writing, there is insufficient experimental data to determine the import of robustness in an analog CAT algorithm it is, at minimum, a factor of which one must be cognizent and may, in fact, prove to be a dominating factor in the design of a viable CAT package.

D. Models: Since most CAT algorithms presuppose some form of circuit simulation in their operation and design of such an algorithm must consider the type and availability of circuit models which are required and/or available. In particular, does the algorithm use nominal circuit models or *faulted circuit models?* Indeed, even if nominal circuit models are used do they operate in their normal range? Finally, one must consider whether or not the algorithm is capable of dealing with "fuzzy" components which do not admit viable simulation models.

E. Module vs. Parameter Testing: Most analog fault diagnosis algorithms can be catagorized as either module oriented or parameter oriented. In the former case the algorithm tests the input-output performance of the individual modules or subsystems which make up the UUT while in the latter case the algorithm estimates a set of parameter values which determine the performance of a given circuit component. Although one can often formulate a circuit model for a given module thereby permitting one to use a parameter oriented algorithm to test modules, such a process may unnecessarily complicate the test procedure. As such, *a module oriented CAT algorithm is preferred* over a parameter oriented algorithm if it can be formulated without compromising other factors.

F. In-Situe Testing: Although secondary to the above considerations the ideal CAT algorithm should allow for *in-situe testing*. Since one cannot control the input signals applied to the UUT in-situe such an algorithm must work with an arbitrary input signal rather than a fixed set of test vectors.

G. Parallel Processing: Since the CAT problem is inherently a large scale systems problem it is essential to exploit whatever computational

power is available to reduce both on-line and off-line computational requirements. In particular, digital CAT algorithms often use some degree of parallel processing in their implementation. Given the additional computational problems associated with an analog CAT algorithm *the degree to which an algorithm can be implemented in parallel becomes a significant factor* in determining its viability and should therefore be included among our criteria for an analog CAT package.

In the above paragraphs we have described seven aspects of the CAT problem which must be considered in judging an analog CAT algorithm. Although we would ideally like to formulate an algorithm with minimal computational requirements a *moderate amount of off-line computation* is acceptable since the off-line computation need only be done once and is carried out in a depot environment where good computational facilities and high level personnel are available. On the other hand since the *on-line computation* associated with a CAT algorithm is replicated for each UUT and carried out in a field environment it must be kept to a minimum. Likewise the test point requirements for an analog CAT algorithm must be kept to a minimum. Although the requirement that the number of test points used by a CAT algorithm grow with the square root of circuit complexity is open to debate it is indicative of a fundamental limitation to the effect that the number of test points should grow at less than a linear rate with circuit complexity. Concerning the remaining criteria we want an algorithm that is robust though the significance of this requirement is not fully understood at this time. Similarly, the availability of circuit models to implement an algorithm must be considered. Finally, but secondary to the above requirements, it would be desirable to have a module oriented algorithm which is

amenable to in-situe testing and parallel processing. These criteria are summarized in table 1 along with a set of goals which one would wish to achieve in an "ideal analog fault diagnosis algorithm".

CAT Algorithms

A. Simulation-Before-Test: Although it is essentially a brute force search algorithm simulation-before-test is well suited to the depot/field computational environment of the CAT problem and, as such, it predominates in most state-of-the-art digital CAT packages.[3] On the other hand its weaknesses become more pronounced in the analog problem wherein it has yet to be successfully implemented. Basically, a simulation-before-test algorithm is a search algorithm in which one simulates the expected test data which would result from various hypothesized failures in an off-line environment. Then when the actual test data is obtained in the field it is compared with the simulated results to determine the failure. Needless to say the technique requires immense amounts of off-line computer time to generate the required data base but is extremely efficient on-line, wherein one need only compare the test results with the simulated data base.

Unfortunately, the cost of an analog simulation is much greater than that of a digital simulation. Moreover, one requires a much larger data base in the analog problem than in the digital problem to cope with the continuous nature of the analog failure phenomena and the robustness problem. As such, there is considerable doubt about the applicability of the simulation-before-test concept in an analog CAT package.

Vis-a-vis our criteria for analog fault diagnosis simulation-before-test requires extremely large amounts of off-line computer time but only a minimum of on-line computer time. Additionally, the test point requirements

for the algorithm are minimal. On the other hand the technique has no inherent robustness and uses faulted simulation models for all components. With regard to the secondary factors the algorithm is module oriented and amenable to parallel processing but not in-situe testing. These considerations are summarized in Table 1.

B. Simulation-After-Test with a Single Test Vector: Rather than using a search algorithm for fault diagnosis one can attempt to model the analog fault diagnosis problem as a nonlinear equation in which one solves for the internal variables or component parameters in terms of the test data. Although this may, at first, seem to totally bypass the repetitive simulation-before-test algorithm, a careful analysis will reveal that each iteration of the required numerical equation solver amounts to a simulation of the UUT. In this case, however, the particular simulations which one carries out are based on known test data rather than a-priori fault hypotheses. As such, the simulations are done on-line after the test data has been obtained and the technique is thus termed simulation-after-test.[2]

In the case where only a single test vector is employed the resultant fault diagnosis equations are "almost linear" and may be solved with the aid of a single (off-line) sparse matrix inversion.[4,5] The test point requirements for the algorithm, however, grow linearly with circuit complexity. Interestingly, this class of algorithms have been discovered independently by a number of authors over the years, most of whom thought that they had found the "ideal algorithm" until they fully appreciated the significance of the test point requirement which severely limits its applicability. From the point of view of our other criteria, however, the algorithm is,

indeed, "ideal". Off-line computational requirements are moderate while on-line computational requirements are minimal. Moreover, the algorithm is inherently robust and requires no simulation models of any kind, it tests modules, and it is amenable to in-situe testing. Finally, the computational requirements associated with the algorithm are sufficiently low so as to render the parallel processing question moot.

C. Simulation-After-Test with Multiple Test Vectors: One approach to reducing the test point requirements of the simulation-after-test algorithm is to use multiple test vectors to increase the number of equations obtained from a given set of test points, thereby rendering the fault diagnosis equation soluable with a restricted number of test points. The most common form of the multiple test vector algorithm is the multi-frequency algorithm used in linear fault diagnosis, though the concept extends to the nonlinear case via the use of multiple test vectors of any type.[1,2]

The reduced test point requirement obtained via the use of multiple test vectors is, however, achieved at the cost of greatly increasing the complexity of the resultant fault diagnosis equations. Indeed, the "almost linear" equations of the single test vector algorithm are replaced by an extremely complex set of nonlinear equations (even for linear systems) in the multiple test vector algorithm. Although these equations can be made trackable in the linear case they appear to be totally untrackable in the nonlinear case and, as such, most of the advantages of the simulation-after-test concept are lost when multiple test vectors are employed.

With regard to our criteria the multiple test vector algorithms require large amounts of on-line computer time though relatively little off-line

computer time is required. In its most obvious form the technique is robust, though this robustness is compromised by most of the "tricks" which have been proposed to make the multiple test vector fault diagnosis equations trackable. Faulted simulation models are required and the algorithm is inherently parameter oriented. Finally, it is not suited to either in-situe testing or parallel implementation.

Conclusions

The above concepts are summarized in Table 1, wherein the various criteria, by which an analog CAT algorithm should be measured are tabulated, the goals for an ideal algorithm are described, and the degree to which the various algorithms achieve these goals is indicated. From the table it is apparent that none of the algorithms is fully acceptable. Indeed, even if one neglects the secondary considerations regarding modules vs. parameters, in-situe testing, and parallel processing all three approaches fail to meet one or more of the primary criteria (indicated by capital leters in the table). As such, the proper approach to the solution of the analog CAT problem remains an open question.

References

1. Duhamel, P., and J.C. Rault, "Automatic Test Generation for Analog Circuits and Systems: A Review", IEEE Trans. on Circuits and Systems, Vol. CAS-26, pp. 411-440, (1979).

2. Plice, W.A., "Automatic Generation of Fault Isolation Tests for Analog Circuit Boards, A Survey", Presented at ATEX East '78, Boston, pp. 26-28, Sept. 1978.

3. Saeks, R., and S.R. Liberty, Rational Fault Analysis, New York, Marcel Dekker, 1977.

4. Saeks, R., Singh, S.P., and R.-w. Liu, "Fault Isolation via Components Simulation", IEEE Trans. on Circuit Theory, Vol. CT-19, pp. 634-640, (1972).

5. Trick, T.N. Mayeda, W., and A. Sakla, "Calculation of Parameter Values from Node Voltage Measurements", IEEE Trans. on Circuits and Systems, Vol. CAS-26, pp. 466-474, (1979).

Criteria	Goal	Simulation-Before Test	Simulation-After-Test with a Single Test Vector	Simulation-After-Test with Multiple Test Vectors
Off-line Comp.	Moderate	VERY HIGH	Moderate	Moderate
On-line Comp.	Minimal	Minimal	Minimal	HIGH
Test Points	Less than linear growth	Less than linear growth	LINEAR GROWTH	Less than linear growth
Robustness	Yes	NO	Yes	Yes (sometimes)
Models	Nominal	FAULTED	None	FAULTED
Modules/Param.	Modules	Modules	Modules	Parameters
In-Situe Test	Yes	No	Yes	No
Parallel Proc.	Yes	Yes	———	No

Table 1: Performance of Analog Fault Diagnosis Algorithms. Unacceptable performance factors are indicated by capital letters.

ANALOG FAULT DIAGNOSIS WITH FAILURE BOUNDS

C.-C. Wu

K. Nakajima

C.-L. Wey

AND

R. Saeks

Analog Fault Diagnosis with Failure Bounds*

C.-c. Wu, K. Nakajima, C.-L. Wey, and R. Saeks
Department of Electrical Engineering
Texas Tech University
Lubbock, Texas 79409

Abstract

A simulation-after-test algorithm for the analog fault diagnosis problem is proposed in which a bound on the maximum number of simultaneous failures is used to minimize the number of test points required. The resultant algorithm is applicable to both linear and nonlinear systems and can be used to isolate a fault up to an arbitrarily specified "replaceable module".

* This research supported in part by the Joint Services Electronic Program of Texas Tech University under ONR Contract 76-C-1136.

I. Introduction

Conceptually, analog fault diagnosis algorithms can be subdivided into three classes;[3] simulation-before-test, simulation-after-test with a single test vector, and simulation-after-test with multiple test vectors. The former is commonly employed in digital testing and is characterized by minimal on-line computational requirements. Unfortunately, the high cost of analog circuit simulation coupled with the large number of potential fault modes which must be simulated in an analog circuit limits the applicability of simulation-before-test algorithms in an analog test environment. As an alternative to simulation-before-test, a number of researchers have proposed simulation-after-test algorithms, in which the internal system variables or component parameters are computed from the test data via a "nonlinear equation solver - like" algorithm. Indeed, in the case where sufficiently many test points are available only a single test vector is required and the fault diagnosis problem reduces to the solution of a linear equation.[8,9] Except for the large number of test points required, this approach is ideally suited to the analog fault diagnosis problem and, as such, a considerable research effort has been directed towards the problem of reducing its test point requirements.[3] One such approach uses multiple test vectors to increase the number of equations obtained from a given set of test points. Unfortunately, this is achieved at the cost of greatly complicating the set of simultaneous equations which must be solved and, as such, the applicability of the approach is limited.

The purpose of the present paper is to describe an alternative simulation-after-test algorithm in which a bound on the maximum number of simultaneous

failures is used to reduce the test point requirements while still retaining the computational simplicity inherent in a single test vector algorithm. Indeed, even though a given circuit may contain several hundred components it is reasonable to assume that at most two or three have failed simultaneously. As such, rather than solving a set of simultaneous equations in n-space the solution to our fault diagnosis problem actually lies in a two or three dimensional submanifold which should yield a commensurate reduction in test point requirements. Unfortunately, even though we may assume that at most two or three components have failed we do not know which two or three, and as such, some type of search is still required. Fortunately, with the aid of an appropriate decision algorithm the required seach can be implemented quite simply.

Consider the circuit or system which is illustrated abstractly in figure 1.

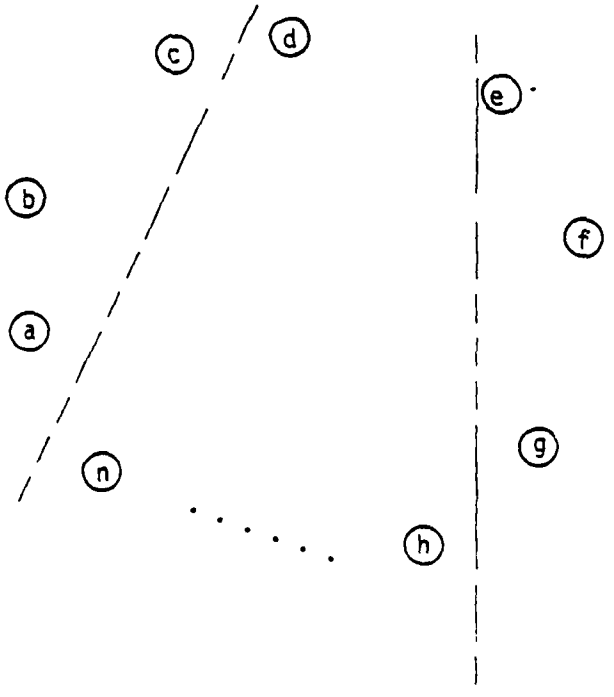


Figure 1. Test algorithm for abstract circuit or system.

Here, the individual circuit components or sub-systems are denoted by circles indexed from a to n. These components are subdivided into two groups, at each step of the test algorithm, as indicated by the dashed lines in figure 1. At each step we assume that one group; say, d through n; is composed of good components and we use the known characteristics of these components together with the test data to determine whether or not the remaining components; a, b and c in this case; are good. Of course, if components d through n are actually good then the resultant test results for components a, b and c will be reliable. On the other hand, if any one of the components d through n is faulty the test data on a, b and c will be unreliable. As such, we repeat the process at the next step of the test algorithm with a different subdivision of components. For instance, we may assume that a through d and h through n are good and use their characteristics to test components, e, f and g. Finally, after a number of such repetitions the test results obtained at the various steps are analyzed to determine the faulty components.

Of course, the number of components which may be tested at any one step is dependent on the number of test points available while the number of steps required is determined by the number of components which may be tested at any one step and the bound on the maximum number of simultaneous failures. As such, the procedure yields a natural set of tradeoffs between the numbers of test points, simultaneous failures and steps required by the algorithm. Indeed, since the computational cost associated with each step of the algorithm is essentially the cost of a single system simulation the latter parameter is a natural measure of the computational cost.

In the following section we describe the simulation model used to test

one set of components under the assumption that the remaining components are good. The model is formulated in both the linear and nonlinear cases and can be used as readily to test modules and subsystems as individual components. Moreover, the requirement that an appropriate matrix be invertible determines the maximum number of components which can be simultaneously tested from a given set of test points as well as the allowable component subdivisions. In section three two decision algorithms for analyzing the resultant test data are described. Indeed the required theory is reminiscent, though not identical to, the t-diagnosibility theory developed for digital system testing over the past decade.[4,6] In the context of our application we give an exact decision algorithm for the case of a single failure together with an analysis of the possible tradeoffs between test points and algorithm steps (read computer costs). Although an exact decision algorithm for the multifailure case has yet to be developed an heuristic algorithm which is applicable to both the single and multifailure case is presented. The algorithm, which is based on an inherently analog heuristic[7], to the effect that two analog errors will never cancel, has proven to be highly reliable while simultaneously reducing the number of steps required from that of the exact (single failure) algorithm. Finally, section four is devoted to a number of examples. These examples include linear circuits with 12 and 22 components which were run on a desktop calculator and a 16 bit mini, respectively.

Although we have yet to implement the algorithm in the nonlinear case the nonlinear algorithm is identical to the linear algorithm except for the requirement that a nonlinear simulator be used in lieu of the linear simulator. As such, we believe that the increase in computational costs for the nonlinear case vis-a-vis the linear case will be in proportion th the increased running time for the nonlinear simulator.

II. The Simulation Model

Although our test algorithm can be formulated in terms of any of the standard system models for the purpose of this exposition we will assume a *component connection model* for the circuit or system under test.[2] In the nonlinear case the *unit under test* is represented by a set of decoupled state models characterizing its components and/or subsystems together with an algebraic connection equation as follows.

$$\begin{aligned} \dot{x}_i &= f_i(x_i, a_i) \\ b_i &= g_i(x_i, a_i) \end{aligned} \quad ; \; x_i(0) = 0, \; i=1,2, \ldots ,n \tag{2.1}$$

and

$$a = L_{11}b + L_{21}u \tag{2.2}$$

$$y = L_{21}b + L_{22}u \tag{2.3}$$

Here, $a = \text{col}(a_i)$ is the column vector composed of the *component input variables*, $b = \text{col}(b_i)$ is the column vector composed of *component output variables*, u is the vector of *external test inputs* applied to the system and y is the vector of *system responses measured at the various test points*. Although the component connection model is not universal it is quite general and subsumes most of the classical topological connection models commonly used in circuit and system theory.[2] Moreover, its inherently decoupled nature is ideally suited to the test problem wherein we desire to distinguish between the characteristics of the individual system components. Although these components may be taken to be elementary RLC components and/or discrete semiconductor devices, in practice the "components" are taken to be the "replaceable modules" within the circuit or system, under test; say, an IC or a "throw-away" circuit board.

At each step of the test algorithm we subdivide the "components" into two groups denoted by "1" and "2" with the components in group "1" assumed to be good and used together with the known values of u and y to compute the component input and output variables, a_i and b_i, for the components in group "2". Although computationally we prefer to work with the decoupled component equations for notational brevity we combine the equations for the components in each group into a single equation

$$\begin{aligned} \dot{x}^1 &= f^1(x^1, a^1) \\ & \qquad ; \quad x^1(0) = 0 \\ b^1 &= g^1(x^1, a^1) \end{aligned} \tag{2.4}$$

and

$$\begin{aligned} \dot{x}^2 &= f^2(x^2, a^2) \\ & \qquad ; \quad x^2(0) = 0 \\ b^2 &= g^2(x^2, a^2) \end{aligned} \tag{2.5}$$

Here, x^1, a^1 and b^1 are the vectors of group "1" component state variables, inputs and outputs; and similarly for x^2, a^2 and b^2. To retain notational compatibility with 2.4 and 2.5 we reorder and partition the connection equations of 2.2 and 2.3 to be conformable with 2.4 and 2.5 as follows

$$a^1 = L_{11}^{11} b^1 + L_{11}^{12} b^2 + L_{12}^1 u \tag{2.6}$$

$$a^2 = L_{11}^{21} b^1 + L_{11}^{22} b^2 + L_{12}^2 u \tag{2.7}$$

$$y = L_{21}^1 b^1 + L_{21}^2 b^2 + L_{22} u \tag{2.8}$$

Given equations 2.4 through 2.8 our goal is to compute the group "2" component variables, a^2 and b^2, given the *test input*, u, the *measured test responses*, y, and an *assumption to the effect that the group "1" componets are*

not faulty. To this end we assume that L_{21}^2 admits a left inverse, $[L_{21}^2]^{-L}$, which, in turn, determines the allowable component subdivisions. Under this assumption one may then formulate a component connection model for a "*pseudo circuit*" composed of the group "1" components with external input vector $u^p = \text{col}(u\ ,\ y)$ and external output vector $y^p = \text{col}(a^2,\ b^2)$ in the form

$$\begin{aligned} \dot{x}^1 &= f^1(x^1, a^1) \\ & \qquad ;\quad x^1(0) = 0 \\ b^1 &= g^1(x^1, a^1) \end{aligned} \tag{2.9}$$

$$a^1 = K_{11} b^1 + K_{21} u^p \tag{2.10}$$

$$y^p = K_{21} b^1 + K_{22} u^p \tag{2.11}$$

Indeed, some algebraic manipulation of equations 2.6 through 2.8 together with the assumption that $[L_{21}^2]^{-L}$ exists will yield

$$K_{11} = \left[L_{11}^{11} - L_{11}^{12} [L_{21}^2]^{-L} L_{21}^1 \right] \tag{2.12}$$

$$K_{12} = \left[L_{12}^1 - L_{11}^{12} [L_{21}^2]^{-L} L_{22} \ \vdots\ L_{11}^{12} [L_{21}^2]^{-L} \right] \tag{2.13}$$

$$K_{21} = \begin{bmatrix} L_{11}^{21} - L_{11}^{22} [L_{21}^2]^{-L} L_{21} \\ \hdashline - [L_{21}^2]^{-L} L_{21}^1 \end{bmatrix} \tag{2.14}$$

$$K_{22} = \left[\begin{array}{c:c} L_{12}^2 - L_{11}^{22} [L_{21}^2]^{-L} L_{22} & L_{11}^{22} [L_{21}]^{-L} \\ \hdashline - [L_{21}^2]^{-L} L_{22} & [L_{21}^2]^{-L} \end{array}\right] \tag{2.15}$$

Since, in our test problem both u and y are known, the above equations can be solved via any standard circuit analysis code to compute $y^p = a^2, b^2)$. Now, under our assumption that the group "1" componenta sre not faulty $y^p = (a^2, b^2)$ represents the inputs and outputs which actually appeared at the terminals of the group "2" components during the test. As such, we may determine which of the group "2" components are faulty by solving equation 2.5 with input a^2 and checking to determine whether or not the resultant output coincides with b^2. Of course, since our assumption to the effect that the group "1" components are not faulty may not be valid *the results of this test are not reliable.* As such, we repeat the process a number of times with different choices for the subdivision of the components into group "1" and group "2". Here, the only constraint on the choice of subdivisions is the requirement that $[L_{21}^2]^{-L}$ exist while the number of combinations employed is limited only by the cost of the required simulations. The results of the several steps in the test algorithm are then analyzed via the techniques described in the following section to determine those components which are actually faulty. To this end the results of each step of the test algorithms are tabulated as follows

"2" \ "1"	a	b	c	...	k
0	x				
1	y				
	⋮				
0	z				

Here, a,b,c, ... ,k denote the group "1 components for a given step of the

test algorithm, x,y, ... ,z denote the corresponding group "2" components while the binary annotation associated with the group "2" components indicates whether this step of the test algorithm indicated that they were good (0) or bad (1). Although this tabular notation is somewhat cumbersome we will eventually generate a binary array indexed by the group "1" and group "2" components in the process of our decision algorithm in which case the tabular notation probes to be convenient.

For linear systems one may formulate an identical algorithm in which the component equations 2.1 are modeled in the frequency domain via

$$a_i = Z_i b_i \quad ; \quad i=1,2, \ldots ,n \tag{2.16}$$

where we have suppressed the s-variable for notational brevity. Then upon subdividing the components into two groups characterized by the equalities $b^1 = Z^{a^1}$ and $b^1 = Z^{a^2}$ and solving the resultant equations under the assumption that $[L_{21}^2]^{-L}$ exists one obtains an equation in the form $y^p = Mu^p$. Specifically,

$$a^2 = M_{11}u + M_{12}y \tag{2.17}$$

$$b^2 = M_{21}u + M_{22}y \tag{2.18}$$

where

$$M_{11} = ([L_{11}^{21}-L_{11}^{22}[L_{21}^2]^{-L}L_{11}^1]\{1-Z^2[L_{11}^{11}-L_{12}^{12}[L_{21}^2]^{-L}L_{21}^1\}^{-1}Z^2[L_{21}^1-L_{11}^{12}[L_{21}^2]^{-L}L_{22}]) + [L_{12}^2-L_{11}^{22}[L_{21}^2]^{-L}L_{22} \tag{2.19}$$

$$M_{12} = ([L_{11}^{21}-L_{11}^{22}[L_{21}^2]^{-L}L_{11}^1]\{1-Z^1[L_{11}^{11}-L_{12}^{12}[L_{21}^2]^{-1}L_{21}^1\}^{-1}Z^1[L_{11}^{12}[L_{21}^2]^{-L}) + [L_{11}^{22}[L_{21}^2]^{-L} \tag{2.20}$$

$$M_{21} = -([L_{21}^2]^{-L}L_{21}\{1-Z^1[L_{11}^{11}-L_{12}^{12}[L_{21}^2]^{-L}L_{21}^1\}^{-1}Z^2[L_{21}^1-L_{11}^{12}[L_{21}^2]^{-L}L_{22}]) - [L_{21}^2]^{-L}L_{22} \tag{2.21}$$

and

$$M_{22} = -([L_{21}^2]^{-L}L_{21}^1(1-Z^1[\cdot_{11}^{11}-L_{12}^{12}[L_{21}^2]^{-L}L_{21}^1)^{-1}Z^1[L_{11}^{12}[L_{21}^2]^{-L}) - [L_{21}^2]^{-L} \qquad (2.22)$$

Although these expressions appear to be foreboding they may all be computed with the aid of only a single matrix inversion. Moreover, since the M_{ij} are independent of the test data and computed in terms of the nominal values of the group "1" components they may be computed off-line and stored in a data base to be retreived at the time a test is conducted. Furthermore, since only a single test vector is required, single frequency testing can be employed in which case the M_{ij} need only be computed at a single frequency. As such, the only on-line computation required for the fault diagnosis of a linear system is the matrix-vector multiplication indicated by equations 2.17 and 2.18 together with the computation of Z^2a^2.

Unlike the linear case, if one is working with a nonlinear circuit or system, the simulations required to compute a^2 and b^2 require a-priori knowledge of y and $\dot{y}$ and thus must be carried out on-line. In practice, however, relatively few time steps are required by these simulations, thereby minimizing their running time. Moreover, all simulations are carried out with nominal components allowing one to use standard CAD circuit models. Indeed, since the group "2" component models are only invoked at the final step of the analysis one can avoid simulating "troublesome" components by always including them in group "2" though this usually means that additional test points will be required. As such, one can avoid simulating "fuzzy" components which do not admit a viable simulation model and/or nonlinear components. Indeed, if sufficiently many test points are available to permit all nonlinear components to be included in group "2" a linear simulation model such as that of 2.17 and 2.18 may be employed even for a nonlinear system.

III. Decision Algorithms

Since the results of the tests described in the preceding section are dependent on our assumption that the group "1" components are not faulty they are not immediately applicable. Following the philosophy initiated by Preparata, Metze, and Chein[6] in their study of self testing computer networks, however, if one assumes a bound on the maximum number of faulty components it is possible to determine the actual fault(s) from an analysis of the test results obtained at the various steps in the algorithm. To this end we will give a complete analysis of the theory required to locate a single fault together with an heuristic which is applicable to the multiple fault case.

Let us assume that at most one circuit component is faulty and that the test results obtained from a given step of the algorithm indicate that all group "2" components are good as indicated in the following table.

"2"	"1"	a	b	c	...	k
0	x					
0	y					
⋮	⋮					
0	z					

In this case we claim that the group "2" components are, in fact, good. Indeed, if a group two component were actually faulty then our test results are incorrect, which could only happen if one of the group "1" components was faulty. As such, the system would have two faulty components contradicting our assumption to the effect that at most one component is faulty.

Now, consider the case where the results from a given step of the test algorithm indicate that exactly one group "2" component is faulty; say, x

"2" \ "1"	a	b	c	...	k
1	x				
0	y				
⋮	⋮				
0	z				

In this case the same argument we used above will guarantee that the components which test good; say, y through z; are, in fact, good. On the other hand we have no information about x. It may be faulty or, alternatively, the test result may be due to a faulty group "1" component.

Finally, consider the case where two or more group "2" components test bad in a given step as indicated in the following table.

"2" \ "1"	a	b	c	...	k
1	x				
1	y				
⋮	⋮				
0	z				

Since, under our assumption of a single failure, it is impossible for two or more group "2" components to be faulty, this test result implies that at least one of the group "1" components is bad. On the other hand since we have assumed that there is at most one faulty component the faulty group "1" component is the only faulty component and, as such, the group "2" components are all good.

Consistent with the above, at each step of the test algorithm, either all or all but one of the group "1" components are found to be good. As such, if we choose our subdivisions so that the components which are found to be good at one step of the algorithm are included in group "1" in all succeeding steps

we will eventually arrive at a group "1", all of whose components are known to be good. As such, the test results obtained at that step will be reliable, thereby allowing us to accurately determine the faulty components in group "2". Although the number of components in group "1" and group "2" may vary from step to step (especially if we work with multivariate components) if we assume that group "1" contains n-m components n-m components and group "2" contains "m" components at each step of the algorithm then the process will terminate in approximately n/m steps. Since the computational cost of the algorithm is proportional to the number of steps (essentially the cost of one simulation per step) while m is determined by the number of allowable test points the ratio n/m represents a natural measure of the possible tradeoffs between test point and computer requirements when employing the algorithm in a single fault made.

Unlike the single fault case, at the time of this writing, we do not yet have an exact decision algorithm for the multiple fault case. Following Liu, however, theproblem can be greatly simplified if one adopts an "analog heuristic" to the effect that two independent analog failures will never cancel.[5] Needless to say, this is an inherently analog heuristic since two binary failures have a fifty-fifty chance of canceling one another. In the analog case, however, two independent failures are highly unlikely to cancel one and another (as long as one works with reasonably small tolerances).

Recall from our discussion of the single fault case that whenever a test result indicates that a component is good then it is, in fact, good. Although this is not rigorously true in the multiple failure case it is true under the

assumption of our heuristic. For instance, consider the test results indicated in the following table in which x is found to be good.

"2" \ "1"	a	b	c	...	k
0	x				
1	y				
⋮	⋮				
0	z				

Now, if x is actually faulty there must be a faulty group "1" component whose effect is to cancel the error in x as observed during this step of the test algorithm. This is, however, forbidden by our heuristic and, as such, we conclude that x is actually good.

Interestingly, our heuristic can be carried a step further than indicated above since, under our heuristic, a bad group "1" component would normally yield erroneous test results. An exception would, however, occur if some of the group "1" components are totally decoupled from some of the group "2" components. As such, if prior to our test we generate a coupling table (by simulation or a sensitivity analysis) which indicates whether or not a faulty group "1" component will effect the test results on a group "2" component, our heuristic may be used to verify that certain group "1" components are good whenever a good group "2" component is located. Consider for example the following table

"2" \ "1"	a	b	c	...	k
0	x	1	0	1	1
1	y	1	1	0	0
⋮	⋮	⋮	⋮	⋮	⋮
0	z	0	1	1	0

in which a "1" in the i-j position indicates that the test results for component i are affected by component j while a "0" in the i-j position indicates that component j does not affect the test results for component i. Now, since component x has been found to be good in this test our heuristic implies that those group "1" components which are coupled to x in this test shows that z is good the heuristic implies that b and c are also good. This, with a single test we have verified that x,z,a,b,c, and k are all good.

Since in any practical circuit the coupling table is composed mostly of 1's it has been our experience that relatively few steps of the algorithm will yield a complete diagnosis. To implement the heuristic, however, one must assume that the maximum number of faulty components is strictly less than the number of group two components. If not, the test results at each step may show that all group "2" components are faulty. in which case no reliable test information is obtained. Moreover, the degree to which the number of group "2" components exceeds the maximum number of faulty components determines the number of algorithm steps which will be required to fully diagnose a circuit.

Although no exact decision algorithm for the multiple failure case presently exists it is noteworthy that the underlying combinatorial decision problem is quite similar to the t-diagnosibility problem usually associated with self testing computer networks, wherein the multiple fault problem has been resolved.[5,6] In that problem, however, one computer tests another with the rest of the network being decoupled, whereas in our problem a subset of components test all remaining components since there exists no practical mechanism for decoupling components in an analog circuit or system. In any

event the problems are similar and, in fact, Amin[1] has already formulated a generalization of the t-diagnosibility problem in which one subset of computers on a network tests another. As such, we believe that an exact decision algorithm for our problem can be formulated in the multiple failure case.

IV. Examples

To illustrate the exact decision algorithm for the single fault case consider a system composed of eight components; a,b, ... ,h; in which any five may test the remaining three. Initially, we let a through e represent the group "1" components and f,g and h represent the group "2" components and assume that the test results for this first step are as indicated in the following table.

"2" \ "1"	a	b	c	d	e
0 f					
0 g					
1 h					

Employing our exact algorithm for the single fault case the above table indicates that components f and g are good and, as such, we move them into group "1" for the second step of the algorithm obtaining

"2" \ "1"	f	g	a	b	c
0 d					
1 e					
1 h					

Since this test indicates that two group "2" components are bad which contradicts our single fault assumption the faulty component must be in group "1" implying that d,e, and h are all good. We therefore move these components into group "1" and implement the final step of our algorithm in the form

"2" \ "1"	h	e	d	f	g
0 a					
1 b					
0 c					

Since all group "1" components are known to be good this final test is reliable and indicative of the fact that b is the faulty component.

Note, that the requirement that L_{21}^{2} be left invertible may make it impossible to use some component subdivisions in which case an alternative sequence of steps may be required in the above process. For instance, if h,e,d,f, and g is not an allowable subdivision the last step in the above process might be replaced by

"2" \ "1"	e	d	f	g	a
1 b					
0 c					
0 h					

indicating that c and h are good. Now, a final step in which c,e,d,f, and g make up group "1" will be reliable as indicated below.

"2" \ "1"	c	e	d	f	g
0 a					
1 b					
0 h					

Now, consider the same single fault example in which our heuristic algorithm is applied using the coupling table indicated below

"2" \ "1"	a	b	c	d	e
0 f	1	0	0	1	1
0 g	0	0	1	1	0
1 h	1	1	1	0	1

According to our heuristic f and g are good and, moreover, everything in group "1" which is coupled to either f or g is good. As such, we conclude from this

first step that f,g,a,c,d, and e are all good. Thus, taking group "1" to be e,d,f,g and a in the next step will yield a reliable test for b,d and h as above.

Finally, consider the case where at most two failures are assumed with the first step in our test algorithm yielding:

"1"	"2"	a	b	c	d	e
0	f	1	0	0	1	1
1	g	0	0	1	1	0
1	h	1	1	1	0	1

Consistent with our heuristic f,a,d, and e are found to be good in this step. Incorporating these components into group "1" for the following step we obtain:

"2"	"1"	f	a	c	d	e
1	b	0	1	1	1	0
1	g	1	0	1	1	0
1	h	1	1	1	0	1

which gives us no information in the multiple failure case*. As such, we try another allowable combination obtaining the following table

"2"	"1"	f	g	a	d	e
1	b	0	1	1	1	0
0	h	1	1	1	0	1
1	c	0	1	1	1	0

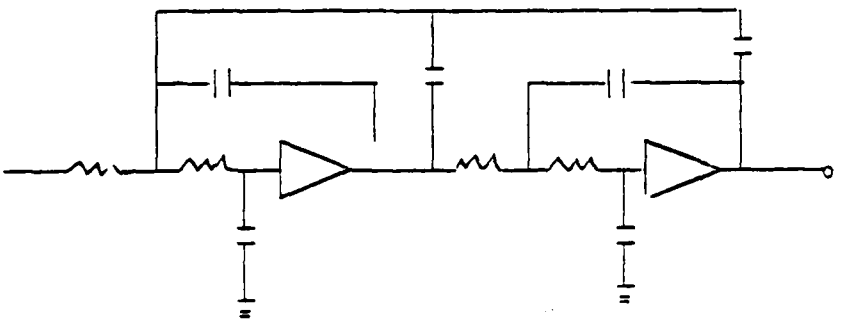
indicating that h,f,g,a, and e are good. Coupled with our previous knowledge

* Actually one can deduce that at least one of the group "1" components is bad since all three group "2" components cannot be bad via our two fault assumption. This, in turn, implies that at most one of the group "2" components is bad.

hat d is good this implies that all group "1" components are good and hence this last step in our algorithm reliably indicates that b and c are the faulty components.

To obtain more realistic examples the above techniques are applied to the 12 and 22 component linear amplifier circuits shown in figure 2 using simulated test data for various numbers of simultaneous failures, choices of test point locations, and both decision algorithms. All analysis for the 12 component circuit was done on an HP 9825 desktop calculator while the 22 component examples were run on a TI 990/20 minicomputer. The results of some 150 simulations of the algorithm are tabulated in table 1. where the number of test points, simultaneous faults, and the decision algorithm employed are indicated. The results of the various simulations are indicated by the ambiguity of the resultant diagnosis. For instance, in our simulation of the 12 component circuit with 3 test points, one failure and the exact algorithm

a)



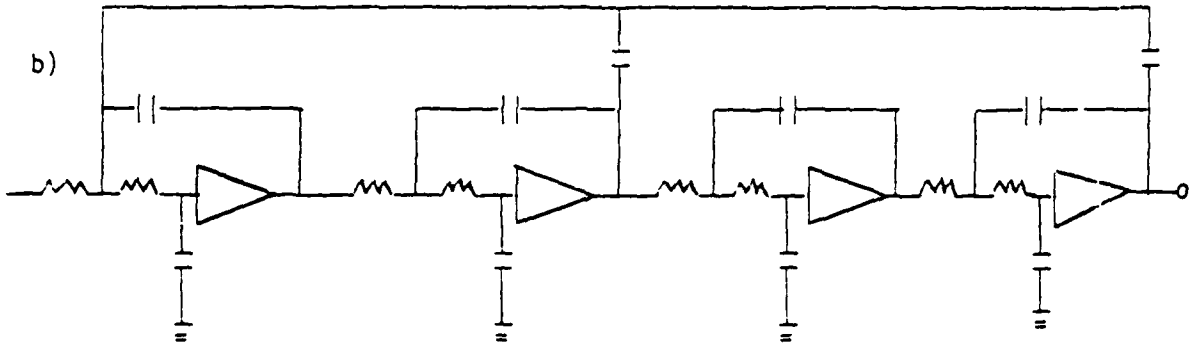


Figure 2. a) 12 component amplifier and b) 22 component amplifier. All stages of the amplifier circuits have nominal op-amp gains of 1.6, nominal resistances of 10K ohms, and nominal capacitance values of .001μf while the feedback capacitors have nominal values of 100pf.

12 runs were made (one with each component faulty). On 10 occasions the fault was located exactly while the fault was located exactly while the fault was located up to an ambiguity set composed of two components on 2 occasions. Finally, we note that the 5th run of the 12 component circuit indicated by an asterisk in the table represents a simulation in which the good components were set at +/-2% off of nominal to test the robustness of the algorithm.

Circuit/Computer	#Test Points	#Faults	Dec. Alg	Ambiguity set 1	2	4	6
12 component circuit simulated on an HP 9825 desktop calculator	4	1	Exact	12			
	4	2	Heuristic	12			
	3	1	Exact	10	2		
	3	1	Heuristic	12			
	3	1	Exact	10*	2*		
22 component circuit simulated on a TI 990/20 minicomputer	8	1	Exact	22			
	6	1	Exact	18		4	
	5	1	Exact	16			6
	5	1	Heuristic	22			

Table 1. Simulated test data. * indicates a simulated test in which the good components were taken to be +/- 2 off of nominal.

V. Conclusions

Although the proposed algorithm is still new and we are just beginning to investigate its performance in the nonlinear case the algorithm promises to meet most of the criteria set in reference 7. Although the *on-line computational requirements* for the algorithm do not compare with a simulation-before-test algorithm they can be kept within reasonable bounds. Indeed, unlike most simulation-after-test algorithms no iterative on-line computation is required. Moreover, one can limit the on-line computation by restricting the number of algorithm steps (at the price of increasing the ambiguity in the resultant diagnosis). Furthermore, in the linear case and/or in the case where there are sufficiently many test points available to permit all nonlinear and fuzzy components to be included in group "2" the major part of the computation required by the algorithm can be done *off-line*.

In general the proposed algorithm permits one to tradeoff between on-line computational requirements and *test points*. Indeed, as indicated in table 1., one can reduce the *test point requirements* to quite reasonable levels though this is usually achieved at the cost of increasing the number of steps in the algorithm (and hence its on-line computational requirements). In particular, our simulations indicate that the algorithm comes close to achieving the $\sqrt{n}$ test point goal set in reference 7.

With respect to the remainder of the criteria specified in reference 7 the algorithm "looks good". In particular, all simulations are carried out using *nominal component models*, it can test *linear and nonlinear modules* of arbitrary size, and is amenable to *in-situe* testing and *parallel processing* techniques (since several steps of the algorithm can be carried out simultaneously).

At the present time the major open question with respect to the performance of the algorithm is its *robustness*. Indeed, there is nothin in the algorithm to make it inherently robust though our initial test for robustness indicated by the asterisk in table 1. proved to be favorable.

VI. References

1. Amin, T., unpublished notes, Bell Laboratories, 1980.

2. DeCarlo, R.A., and R. Saeks, Interconnected Dynamical Systems, New York, Marcell Dekker, (to appear).

3. Duhamal, P., and Rault, J.C., "Automatic Test Generation Techniques for Analog Circuits and Systems: A Review", IEEE Trans. on Circuits and Systems, Vol. CAS-26, pp. 411-440, (1979).

4. Hakimi, S.L., "Fault Analysis in Digital Systems - A Graph Theoretic Approach", in Rational Fault Analysis, New York, Marcel Dekker, 1977, pp. 1-12

5. Liu, R.-w. unpublished notes, Univ. of Notre Dame, 1980.

6. Preparata, F.P., Metze, G., and R.T. Chein, "On the Connection Assignment Problem of Diagnosible Systems", IEEE Trans. on Electronic Computers, Vol. EC-16, pp, 448-454, (1967).

7. Saeks, R., "Criteria for Analog Fault Diagnosis", Proc. of the 1981 European Conf. on Circuit Theory and Design, The Hague, Aug 1981, (to appear).

8. Saeks, R., Singh, S.P., and R.-w. Liu, "Fault Isolation via Components Simulation", IEEE Trans. on Circuit Theory, Vol. CT-19, pp. 634-640, (1972).

9. Trick, T.N., Mayeda, W., and A. Sakla, "Calculation of Parameter Values from Node Voltage Measurements", IEEE Trans. on Circuits and Systems, Vol. CAS-26, pp. 466-474, (1979).

A DIFFERENTIAL-INTERPOLATIVE APPROACH TO ANALOG FAULT SIMULATION

BY

C.-c. WU

A. SANGIOVANNI-VINCENTELLI

AND

R. SAEKS

A Differential-Interpolative Approach to Analog Fault Simulation

C.-c. Wu**, A. Sangiovani-Vincentelli*, and R. Saeks**

I. Introduction

After a half centry of neglect by the circuits and systems community the past decade has witnessed the emergence of a research effort in the analog circuit maintenance area. The various algorithms which have been thus far proposed for the analog fault diagnosis problem may naturally be subdivided into two classes termed "simulation-before-test" and "simulation-after-test". The former are commonly used in digital system test algorithms and require an automatic test program generator (ATPG) which simulates the responses of "all possible" failures. This is typically done at a maintenance depot with the simulated responses being recorded and shipped to the field where the response of the unit under test (UUT) is compared with the simulated responses to determine the failure. The major advantage of simulation-before-test is that it is ideally matched to the depot/field maintenance environment with the largest part of the computation done only once. As such, the technique is ideally suited for digital testing where the binary nature of the problem keeps the number of failures to be simulated within bounds and eliminates tolerence problems. Unfortunately, in the analog problem we must cope with a continuum of possible failures and simultaneously deal with good components which are in tolerance but not nominal. As such, a tremendous number of simulations are required by a simulation-before-test algorithm, while some type of decision algorithm is required to cope with the tolerance effects.

* Dept. of Elec. Engrg. and Comp. Science, Univ. of California at Berkeley, Berkeley, CA 90024.
** Dept. of Elec. Engrg., Texas Tech Univ., Lubbock, TX 79409. This research supported in part by the Joint Services Electronics Program at Texas Tech University under ONR Contract 76-C-1136.

Unlike simulation-before-test, simulation-after-test uses an "equation solver-like" algorithm to compute the parameters of the UUT components in the field. Since most such algorithms require iterative evaluation of the equation to be solved, the UUT is effectively simulated at each iteration, though the simulation is based on actual test data rather than hypothesized failure data. The simulation process is, thus, carried out after testing the UUT and hence the choice of terminology. The advantage to such an approach is that the faulty component parameters are computed explicitly, thereby, eliminating the ambiguity caused by the use of discrete simulation-before-test data and tolerance effects. Although relatively few simulations are required for each UUT, they must be carried out in the field rather than the depot and they must be repeated for each UUT.

The purpose of the present paper is to describe a research effort directed at alleviating some of the difficulties in developing a simulation-before-test algorithm for analog fault diagnosis. The underlying philosophy and motivation for our formulation is discussed in section 2, along with a derivation of the required differential-interpolative fault diagnosis formula. Finally, section 3 is devoted to a number of illustrative examples of the approach. These include both linear and nonlinear examples formulated in the frequency and time domains, respectively.

II. A Differential-Interpolative Algorithm

Although any practical fault diagnosis algorithm must be able to handle systems with a hundred or more components, from an intuitive point of view our algorithm is best illustrated in the two component cases where the parameter space can be displayed graphically. Say, we are dealing with an RC circuit for which the parameter space is illustrated in figure 1.

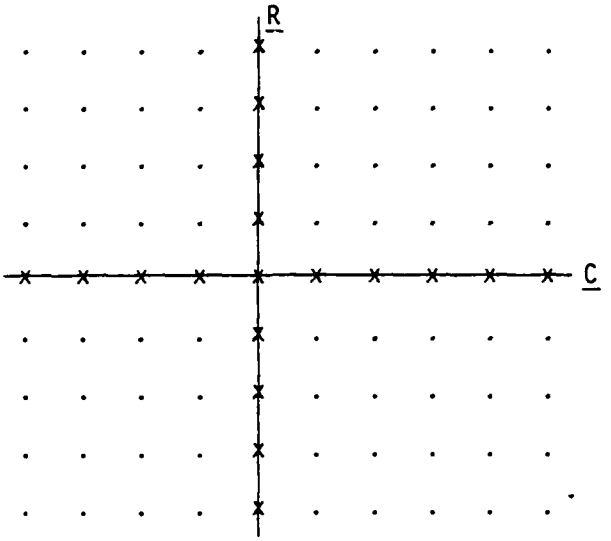


Figure 1: Parameter space for RC circuit.

Here, R and C represent normalized parameter values, wherein, the nominal parameter values are transformed to the origin. In the most general simulation-before-test algorithms one assumes that the faulty parameter values may lie anywhere in the R-C plane and therefore carries out simulations along a two dimensional discrete array spread over the entire plane.

Fortunately, in a "real world" testing environment one can assume that only a "limited number of components" fail simultaneously. In our two component example we may therefore assume that either R or C has failed with the other remaining nominal in which case the circuit need only be simulated at a discrete

set of points along the coordinate axes in the R-C plane denoted by x's in figure 1. As such, the number of simulations required is significantly decreased. Indeed, this is one of the major advantages of the simulation-before-test concept as compared to simulation-after-test algorithms which typically fail to exploit a "limited number of failures" assumption.

While the above described approach has been used with considerable success in digital system testing, wherein, the axes are binary and no tolerance problems are encountered, it is not well suited to the analog test problem. First, an analog failure may occur anywhere along the axis and hence some type of approximation scheme is required to interpolate between the discrete simulations. Secondly, a "good" component is assumed to be in-tolerance though it may not be nominal. As such, in an analog environment the "limited number of failures" assumption implies that the solution to our fault diagnosis problem lies near, but not necessarily on, the coordinate axes as indicated by the shaded regions in figure 2a.

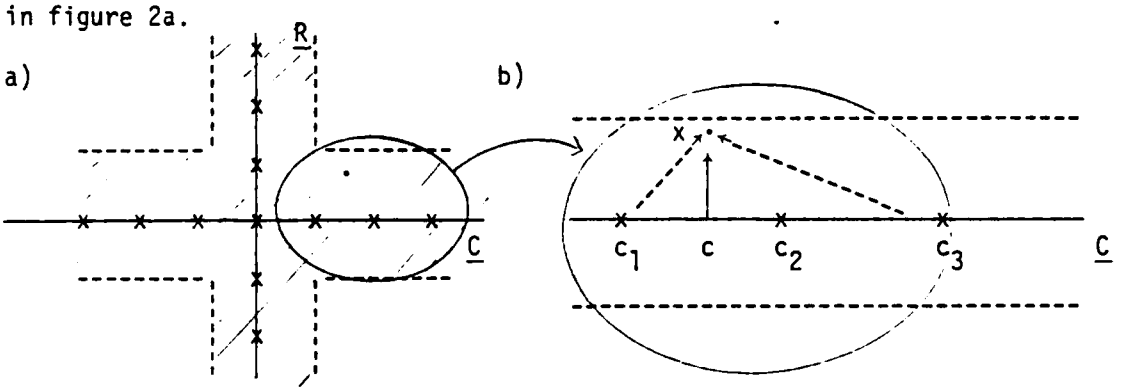


Figure 2: a) Solution space under a single failure assumption.
b) Illustration of the differential-interpolative diagnosis algorithm.

While we might choose to simply fill the shaded region with additional simulations, the cost of such a process may prove to be excessive. Rather, we

exploit the fact that the deviations of good component parameters from nominal are small and use a 1st order Taylor series approximation to approximate the deviation. We note that such an approach cannot be used to locate the faulty parameter values which may be far from nominal; indeed, it is often infinite or zero; though it can be used to cope with the tolerance effects.

Our differential-interpolative approach thus uses a classical minimum distance algorithim to locate the general region of the faulty parameter values indicated by the circle in figure 2a (which is magnified in figure 2b). Now, it is assumed that the simulated values of the system responses; f_1, f_2, and f_3; corresponding to the points; c_1, c_2, and c_3; are available along with the associated inverse sensitivity matrices; J_1^{-1}, j_2^{-1}, and J_3^{-1}. We then interpolate these data points to approximate the system responses and the associated inverse sensitivity matrice along the axis by functions $f(c)$ and $J(c)^{-1}$. Although any interpolation can be employed we have had our best results using a bilinear interpolation for f (which gives exact results in the linear case) and a second order polynomic interpolation of J^{-1}. Now, if x denotes the faulty parameter vector and m denotes the measured system responses then a 1st order Taylor series approximation combined with our interpolation will yield the (approximate) equality

$$m = f(c) + J(c)[x - c] \qquad 1.$$

for those values of c near x. Equivalently,

$$[x - c] = J(c)^{-1}[m - f(c)] \qquad 2.$$

Interestingly, by invoking the Projection theorem one can reduce the above vector equation to a scalar equation and simultaneously eliminate the requirement for storing the inverse sensitivity matrices. Indeed, the vector [x-c] will be

perpendicular to the axis at the point c which makes the closest approach to the fault. As such, if e_c denotes the unit vector in the direction of the axis then

$$0 = e_c^t[x - c] = e_c^t J(c)^{-1}[m - f(c)] \qquad 3.$$

which can be solved for the faulty parameter value, c. Note, our goal is to solve for c, not x, since we are interested in locating the faulty parameter value in the presence of the tolerance problem, but we really do not care to compute the deviations from nominal in the good parameters.

To summarize, if rather than simply storing the simulated circuit responses, f_i, we also store the vectors $e_c^t J_i^{-1}$ then the tolerance effects associated with the good components can be completely removed from our fault diagnosis algorithm - at least up to the approximation error induced by the interpolation process and Taylor series expansion. Since most good circuit simulation codes include a package for generating sensitivity matrices at little additional cost over and above that involved in simply simulating the circuit responses the approach can be implemented with only a minimal increase in simulation costs. As such, the major expense associated with the approach lies with the storage requirements (for the f_i and $e_c^t J_i^{-1}$ vectors) which are approximately double that of a classical fault simulation algorithm.

Although the above derivation has been illustrated in the two dimensional case with a single faulty parameter it can be readily extended to a general setting, say with several hundred components and three or four simultaneous faults. If one assumes p simultaneous faults then p inner products are required to apply the Projection theorem yielding p equations and p unknowns to be solved for the faulty parameter values. Otherwise the formulation for the general case is identical to the single fault case described above.

III. Examples

In this section, three examples are given, two of them for linear systems and one for the nonlinear case. All of these examples were simulated on an HP9825A programmable calculator, and yielded fairly good results.

Our first example is a second order low pass filter. The filter contains five components, K, R_1, R_2, C_1, and C_2, while, the circuit diagram is shown in figure 3-1

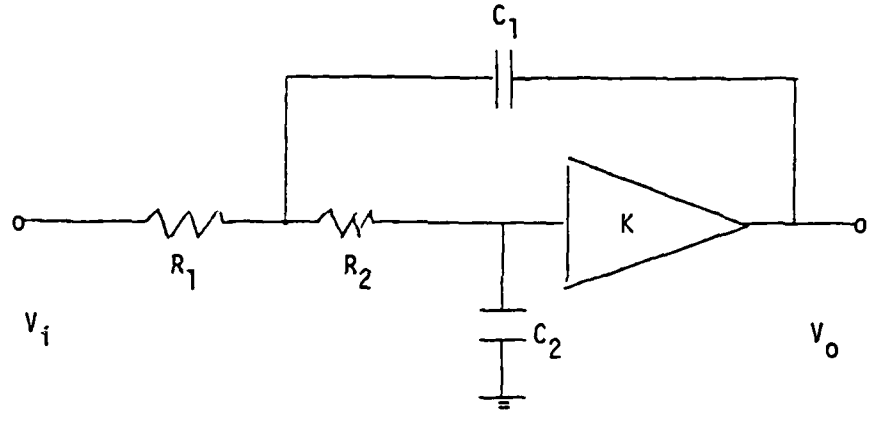


Figure 3-1.

The transfer function for this circuit is given by

$$f(\hat{r},s) = \frac{K}{S^2C_1C_2R_1R_2 + s[R_2C_2 + R_1C_2 + R_1C_1(1-K)]+1} \qquad (3\text{-}1)$$

The partial derivatives of the transfer function with respect to each parameter take the form

$$\frac{\delta f}{\delta K}(\hat{r},s) = \frac{D + SC_1R_1K}{D^2} \qquad (3\text{-}2)$$

$$\frac{f}{R_1}(\hat{r},s) = \frac{-K[S^2C_1C_2R_2 + SC_2 + SC_1(1-K)]}{D^2} \qquad (3\text{-}3)$$

$$\frac{\delta f}{\delta R_2}(\hat{r},s) = \frac{-K[S^2C_1C_2R_1 + SC + SC_2]}{D^2} \tag{3-4}$$

$$\frac{\delta f}{\delta C_1}(\hat{r},s) = \frac{-K[S^2C_1R_2R_1 + SR_1(1-K)]}{D^2} \tag{3-5}$$

$$\frac{\delta f}{\delta C_2}(\hat{r},s) = \frac{-K[S^2C_1R_1R_2 + SR_2 + SR_1]}{D^2} \tag{3-6}$$

where $D = S^2C_1C_2R_1R_2 + S[R_2C_2 + R_1C_2 + R_1C_1(1-K)] + 1$

Since we have five parameters in the transfer function, five distinct test frequencies are required to provide sufficient information for diagnosis. The $f(\hat{r})$ vector is then formed by the transfer functions corresponding to each frequency,

$$f(\hat{r}) = \begin{bmatrix} f(\hat{r},s_1) \\ f(\hat{r},S_2) \\ f(\hat{r},S_3) \\ f(\hat{r},S_4) \\ f(\hat{r},S_5) \end{bmatrix} \tag{3-7}$$

Similarly, the sensitivity matrix is given by

$$J(\hat{r}) = \begin{bmatrix} \frac{\delta f}{\delta k}(\hat{r},S_1) & \frac{\delta f}{\delta R_1}(\hat{r},S_1) & \cdots & \frac{\delta f}{\delta C_2}(\hat{r},S_1) \\ \frac{\delta f}{\delta K}(\hat{r},S_2) & \frac{\delta f}{\delta R_1}(\hat{r},S_2) & \cdots & \frac{\delta f}{\delta C_2}(\hat{r},S_2) \\ \vdots & \vdots & & \\ \frac{\delta f}{\delta K}(\hat{r},S_5) & \frac{\delta f}{\delta R_1}(\hat{r},S_5) & \cdots & \frac{\delta f}{\delta C_2}(\hat{r},S_5) \end{bmatrix} \tag{3-8}$$

Next, we generate a data base in which $f(\hat{r})$ and $e_r^t J(\hat{r})$ are evaluated at selected points on each axis. Since the bilinear interpolation gives exact results in the linear case, we need not choose too many points on each axis, in fact, for this simulation example 15 points were selected along each axis.

Given the measured data for a faulty circuit, the processor searches the data base, to locate the point α_2 which minimizes $\|m-f(\alpha)\|$. We then choose the two points α_1, α_3 on both sides of α_2 along the same axis, and use these three points to evaluate the coefficients of the bilinear polynomial $\frac{ar+b}{r+c}$ in the interval (α_1, α_3) also, we use $e_r^t J^{-1}(\alpha_1)$, $e_r^t J^{-1}(\alpha_2)$ and $e_r^t J^{-1}(\alpha_3)$ to compute the coefficients of the second order polynomial approximate to $e_r^t J^{-1}(\alpha)$ Finally, a golden section linear search is used to solve

$$e_r^t J^{-1}(\alpha)(f(\alpha)-m) = 0 \qquad (3\text{-}9)$$

The faulty diagnosis results are listed in table 3-1. Here, the nominal values of K, R_1, R_2, C_1, C_2 are 1.6, 1KΩ, 1KΩ, 0.16μF and 0.16μF respectively and the faulty parameter is underlined in the table

Table 3-1

	1	2	3	4	5	6
K	<u>0.6</u>	1.62	1.62	1.58	1.62	<u>4</u>
R_1	1090	<u>2500</u>	1090	1070	1050	1050
R_2	930	1040	<u>10</u>	930	930	1045
C_1	0.163μ	0.161μ	0.157μ	<u>0.25μ</u>	0.157μ	0.162μ
C_2	0.162μ	0.162μ	0.162μ	0.157μ	<u>0.25μ</u>	0.162μ
Result	K 0.591	R_1 2492	R_2 19.4	C_1 0.239μ	C_2 0.238μ	K 4.01

In the first simulation, K is the faulty component with a value of 0.6, the other four components are 5% or so off their nominal values, the simulation result shows that K failed, and locates it at K=0.591. The same remarks apply to the other five simulations.

Although the technique generally yields satisfactory results occasional errors occur when the good components are too far out of tolerance. For instance, the following parameter values K=1.62, R_1=1070, R_2=910, C_1=0.5μ, and C_2=0.172μ led to an erroneous result. The simulation shows that C_2 is failed with the value of 0.179μ. However, the faulty component, in this simulation, is actually C_1. If we sketch a two dimensional representation of the C_1, C_2 plane the difficulty becomes clear. Figure 3-2 shows that C_2 is too far away from its own nominal value, and thus instead of locating the error at α we expect the simulation result locates the failure at β, with the differential term still pointing toward the actual failure denoted by x.

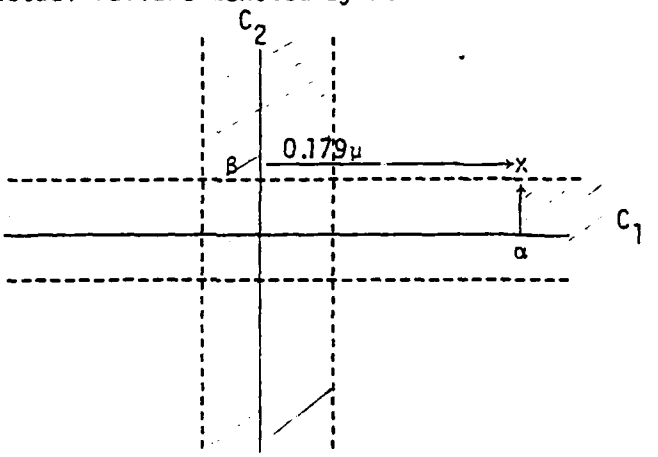


Figure 3-2

Our second example is a fourth order low pass filter, comprised of two cascade second order low pass filters, the circuit diagram is shown in Figure 3-3.

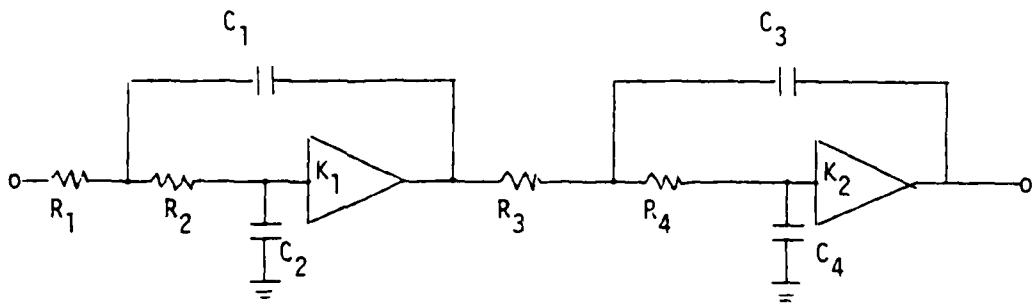


Figure 3-3

The procedure for generating the data base is similar to the one described above, except here, ten different frequencies are required. The fault diagnosis results are described in table 3-2. Here, the nominal parameter values are $K_1 = 1.6$, $R_1 = R_2 = 1K$, $C_1 = C_2 = 0.16\mu$, $K_2 = 1.2$, $R_3 = R_4 = 1.5K$, $C_3 = C_4 = 0.2\mu$, while the faulty parameter is again underlined in the table.

Our final nonlinear example is composed of a diode loaded by a shunt RC circuit as illustrated in Figure 3-4. The diode is modeled by the characteristic function

$$I = I_o(e^{V_o/V_t} - 1) \tag{3-10}$$

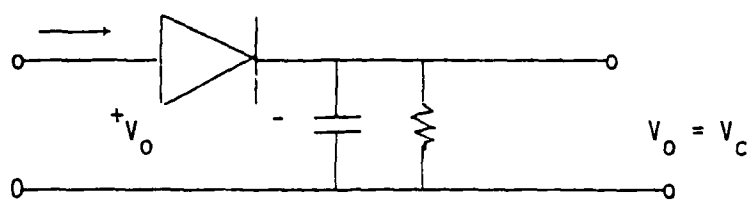


Figure 3-4

Now, instead of working with frequency domain transfer functions, we work in the time domain. A state equation for this circuit is given by

$$\dot{V}_c = \frac{1}{C} I_o(e^{(V_i - V_c)/V_t} - 1) - \frac{V_c}{RC}. \tag{3-11}$$

Table 3-2

	1	2	3	4	5	6
K_1	1.62	1.62	1.57	1.57	0.36	1.61
R_1	1070	1070	1055	1050	1070	1043
R_2	920	920	1030	300	930	1010
C_1	0.162μ	0.157μ	0.157μ	0.162μ	0.161μ	0.11μ
C_2	0.161μ	0.162μ	0.162μ	0.157μ	0.158μ	0.157μ
K_2	1.17	3.45	1.22	1.22	1.21	1.18
R_3	600	1550	1560	1430	1430	1535
R_4	1560	1570	1450	1570	1590	1478
C_3	0.21μ	0.21μ	0.195μ	0.205μ	0.195μ	0.202μ
C_4	0.205μ	0.195μ	0.12μ	0.203μ	0.203μ	0.198μ
Result	R_3	K_2	C_4	R_2	K_1	C_1
	711	3.546	0.125μ	339.7	0.357	0.118

The goal is to integrate this differential equation so as to build a $V_c(\hat{r})$ vector and $f(\hat{r})$ vectors as in the previous examples.

Numerical techniques can be used to compute $V_c(t)$ at any instant t. In this example, the $V_c(\hat{r})$ vector was elevated by applying the fourth order Runge-Kutta method. Note that since there are four independent parameters, R, C, I_o, and V_T; in equation (3-11) $V_v(\hat{r}, t)$ should be evaluated at four different time instants to build a $V_c(\hat{r})$ vector.

The sensitivity matrix is generated via the linear differential equations

$$\dot{\frac{\delta V_c}{\delta I_o}} = \frac{1}{C}(e^{(v_i - V_c)/V_t} - 1) + E \frac{\delta V_c}{\delta I_o} \tag{3-12}$$

$$\dot{\frac{\delta V_c}{\delta V_t}} = \frac{I_o}{V_T^2 C}(V_c - V_i)e^{(V_i - V_c)/V_T} + E \frac{\delta V_c}{\delta V_T} \tag{3-13}$$

$$\dot{\frac{\delta V_c}{\delta R}} = \frac{V_c}{R^2 C} + E \frac{\delta V_c}{\delta R} \tag{3-14}$$

$$\dot{\frac{\delta V_c}{\delta C}} = \frac{V_c}{RC^2} + \frac{\delta V_c}{\delta C} \tag{3-15}$$

Where $E = -\frac{1}{V_T C} I_o e^{(V_i - V_c)/V_T} - \frac{1}{RC}$

Again, a fourth order Runge-Kutta method is applied to obtain $\frac{\delta V_c}{\delta I_o}(\hat{r}, t)$, $\frac{\delta V_c}{\delta V_t}(\hat{r}, t)$, $\frac{\delta V_c}{\delta R}(\hat{r}, t)$, and $\frac{\delta V_c}{\delta C}(\hat{r}, t)$ at the four specified instants.

The simulation results are summerized in table 3-3. The nominal values of I_o, V_T, R, C are 0.2, 0.1, 1K, and 0.25 respectively.

Table 3-3

	1	2	3	4	5
I_o	0.035	0.21	0.2	0.22	0.002
V_T	0.11	5.3	0.11	0.11	0.09
R	920	930	6250	1070	1090
C	0.23	0.27	0.255	2.4	0.23
Result	I_o 0.0312	K 5.449	R 6001	C 2.678	I_o 0.003

IV. Conclusions

For the simulation-before-test approach to fault diagnosis, we gain from the fact that most computation can be done by off-line computation, thus greatly reducing the repetitive on-line computation associated with many fault diagnosis algorithms. From a practical point of view, the economics of such an approach are extremely attractive. Unfortunately, the simulation-before-test approach is subject to a certain degree of ambiguity introduced by good components which are in-tolerance but not nominal.

In this paper, we have proposed a simulation-before-test algorithm for analog fault diagnosis, in which a differential-interpolative technique is used to eliminate the ambiguity caused by tolerance effects. Our approach has been tested with satisfactory results in both the linear and nonlinear cases. In fact, for the linear case, the approach provides an exact interpolation for f(c) on the axes, and thus reduces the amount of simulation-before-test data required on each axis. Although this is not true for nonlinear case, the diagnosis results are still very attractive. Of course, occasional errors may occur when the good components are too far out of tolerance. This phenomena is, however, expected and well understood. Indeed, the difficulty occurs only when the 1st order Taylor series approximation is too good. Because this phenomena will rarely occur in the real world, we believe that it may be neglected in a practical algorithm.

MULTITEST DIAGNOSIBILITY OF NONLINEAR CIRCUITS AND SYSTEMS

A. SANGIOVANNI-VINCENTELLI

AND

R. SAEKS

Multitest Diagnosibility of Nonlinear Circuits and Systems

A. Sangiovanni-Vincentelli* and R. Saeks**

I. Introduction

During the past decade a considerable research effort has been devoted to the analog fault diagnosis problem wherein one desires to locate faulty circuit components given the overall circuit response to one or more test vectors. Conceptually the process may be described by a nonlinear equation

$$y - f(\alpha,u) \qquad 1.$$

where y represents the measured response to the test vector u given the faulty parameter vector, α. Since u is known and y is a measureable quantity the fault diagnosis problem may be resolved by simply solving 1. for α given u and y. Unfortunately, in practice, the dimension of y is limited by the number of accessible test points in the circuit and is typically smaller than the dimension of the parameter vector, thereby precluding the direct solution of 1. To alleviate this difficulty a set of test vectors; $\{u_1,u_2, \ldots ,u_n\}$; is employed yielding the set of simultaneous equations

$$y_i = f(\alpha,u_i) \; ; \; i=1,2, \ldots ,m \qquad 2.$$

Since the parameter vector, α, is independent of the choice of test vector this process effectively increases the number of available equations without

*Dept. of Elec. Engrg. and Comp. Sci., Univ. of California at Berkeley, Berkeley, Ca. 90024.

**Dept. of Elec. Engrg., Texas Tech Univ., Lubbock, TX. 79409. This research supported in part by the Joint Services Electronics Program at Texas Tech Univ. under ONR contract 76-C-1136.

increasing the number of unknowns. More Concisely, if we let $\underline{y} = \text{col}(y_i)$ and $F(\alpha) = \text{col}(f(\alpha,u_i))$ the "multitest" fault diagnosis problem reduces to the solution of

$$\underline{y} = F(\alpha) \qquad 3.$$

Needless to say once equation 3. has been formulated its solution is amenable to standard algorithms. The problem, however, is to determine whether or not there exists a set of test vectors $\{u_1,u_2, \ldots ,u_m\}$ such that 3. is solvable in an appropriate sense. To this end we will formulate a diagnosibility criterion directly in terms of the function f which determines the degree to which the equation $\underline{y} = F(\alpha)$ will be solvable, given an "optimal" choice of test vectors. Since this criterion is a property of the circuit rather than the test algorithm it can therefore be used as a design aid with which to choose test points and/or to aid in designing "testable circuits".

The authors have previously formulated such criteria for linear circuits[3] and for memoryless nonlinear circuits,[1] respectively. The present formulation for dynamical nonlinear circuits is obtained by simply reformulating the memoryless nonlinear theory in an infinite dimensional setting. Surprisingly, however, since the parameter vector, r, remains finite dimensional the infinite theory yields a *finite dimensional test matrix.*

Let U denote a (topological) space of admissible test inputs and let Y denote a Hilbert space of test outputs. We then model a general nonlinear dynamical system as a function $f:U\times A \to Y$ where $y = f(u,\alpha)$ denotes the response of the system to an input u given the parameter vector $\alpha \in A$. Here, A is an open subset of R^k which defines our parameter space and f is assumed to be

continuous in u and continuously (Frechet) differentiable in α.[2] The problem at hand is to solve for α given a set of input/output pairs (u_i, y_i), i=1,2, ... ,m. We say that the system is *locally diagnosable at a point* $\alpha_0 \in A$ if there exists an open neighborhood, V, of α_0 such that for every $\alpha \in V$ $f(u,\alpha) \neq f(u,\alpha_0)$ for some $u \in U$. Similarly, we say that the system is locally *diagnosable* if it is locally diagnosable for almost all $\alpha_0 \in A$ (in the sense of Lebesque measure). Finally, if $M(\alpha)$ is a matrix valued function of $\alpha \in A$ we say that a point α_0 is a *regular point* if there exists an open neighborhood, V, of α_0 such that $M(\alpha)$ has constant rank in V.

To define our test matrix we let $J(u,\alpha) = \frac{\partial f}{\partial \alpha}(u,\alpha)$ denote the Frechet derivative[2] of f with respect to α evaluated at a fixed u and α. With $u \in U$ fixed, f maps A to Y and hence $J(u,\alpha)$ is a linear transformation mapping from R^k to Y[2]. Moreover, for k = 1 such a mapping may be represented by an element of Y hence $\frac{\partial f}{\partial \alpha_i}(u,\alpha)$ may be identified with an element of Y and

$$J(u,\alpha) = \text{row}\left[\frac{\partial f}{\alpha_i}(u,\alpha)\right] \in Y^k \cdot \qquad 4.$$

Now let w denote a positive measure defined on the Borel sets of U such that $w(v) > 0$ for every open set V contained in U.[2] Finally, we introduce the test matrix

$$R_w(\alpha) = \int_U J^*(u,\alpha)J(u,\alpha)dw(u) \qquad 5.$$

Recall that $J(u,\alpha)$ is a mapping for R^k to Y hence its adjoint is a mapping from Y to R^k and thus $R_w(\alpha)$ is a matrix valued function even though U and Y may be infinite dimensional.[2] Also note that the self duality of Y is necessary for $R_w(\alpha)$ to be well defined.

Theorem 1: Let w be any admissible measure for which $R_w(\alpha)$ exists for all $\alpha \in A$. Then the system is locally diagnosable at a regular point $\alpha_0 \in A$ if and only if $R_w(\alpha)$ is nonsingular.

Proof: By invoking the integral form of the mean value theorem[2] we have

$$f(u,\alpha) - f(u,\alpha_0) = \int_0^1 J(u,t\alpha_0 + (1-t)\alpha)dt\,[\alpha - \alpha_0] \qquad 6.$$

for any α in a neighborhood of α_0 and $u \in U$. Now, assume that the system is not locally diagnosable at α_0 in which case there exists a sequence $\{\alpha_i\} \in A$ approaching α_0 for which

$$f(u,\alpha_i) = f(u,\alpha_0) \qquad 7.$$

Letting $\alpha = \alpha_i$ in 6. yields

$$0 = \int_0^1 J(u,t\alpha_0 + (1-t)\alpha_i)dt\,[\alpha_i - \alpha_0] \qquad 8.$$

$$= \int_0^1 J(u,t\alpha_0 + (1-t)\alpha_i)dt\,[a_i] \qquad 9.$$

where $a_i = [\alpha_i - \alpha_0]/\|[\alpha_i - \alpha_0]\|$. Since a_i is normalized to lie in the unit sphere of R^k which is a compact set a_i admits a convergent subsequence a_{i_k} whose limit, a, also has unit norm.[2] Using the convergent subsequence in 9. we have

$$0 = \lim_{k \to \infty} \int_0^1 J(u,t\alpha_0 + (1-t)\alpha_{i_k})dt\,[a_{i_k}] = \int_0^1 J(u,\alpha_0)dt\,[a] = J(u,\alpha_0)a \qquad 10.$$

Since a has unit norm it is non-zero while

$$a^*R_w(\alpha_0)a = \int_U a^*J^*(u,\alpha_0)J(u,\alpha)a\,dw(u) = 0 \qquad 11.$$

implying that $R_w(\alpha_o)$ is singular since it is a positive semidefinite matrix which is not positive definite.[2]

Conversely, if $R_w(\alpha_o)$ is singular it follows from the assumption that α_o is a regular point and lemma 1 of reference 1 that there exists an open neighborhood V of α_o and a continuous R^k valued function $c(\alpha) \neq 0$ defined on V such that

$$0 = c^*(\alpha)R_w(\alpha)c(\alpha) = \int_U c^*(\alpha)J^*(u,\alpha)J(u,\alpha)c(\alpha)\, dw(u) \qquad 12.$$

Since $w(v) > 0$ and $J(u,\alpha)$ is continuous in α this implies that

$$J(u,\alpha)c(\alpha) = 0 \quad \alpha \in V \qquad 13.$$

Finally, we define a curve $\alpha(s) \in V$ by the differential equation

$$\frac{\partial \alpha}{\partial s} = c(\alpha) \quad ; \quad \alpha(0) = \alpha_o \qquad 14.$$

Substituting $\alpha(s)$ into $f(u,\alpha)$ and computing its derivative with respect to s via the chain rule[2] we obtain

$$\frac{\partial f}{\partial s}(u,\alpha(s)) = J(u,\alpha(s))\frac{\partial \alpha}{\partial s} = J(u,\alpha(s))c(\alpha) = 0 \qquad 15.$$

showing that $f(u,\alpha(s))$ is constant along a curve emanating from α_o. Since $\alpha(s)$ is independent of u this implies that the system is not locally diagnosable at α_o thereby completing the proof. □

Note, the proof uses the finite dimensionality of the parameter space but does not require that U and Y be finite dimensional. As such, the theory is equally valid for memoryless and dynamical systems. Furthermore, the result is independent of the choice of the measure w.

To use the theorem as a test for local diagnosability we may invoke Theorem 2 of reference 1. Alternatively, one may make the mild assumption that $f(u,\alpha)$ is (real) analytic in α in which case the hypotheses of that theorem are automatically satisfied and the following result is obtained.

Theorem 2. Assume that $f(u,\alpha)$ is analytic in α and let w be any admissable measure for which $R_w(\alpha)$ exists for all $\alpha \in A$. Then the system is locally diagnosable if and only if $R_w(\alpha_o)$ is nonsingular for some $\alpha_o \in A$.

Proof: Let

$$\rho = \max_{\alpha \in A} \{\text{rank}[R_w(\alpha)]\} \tag{16.}$$

which must be achieved, say at $\underline{\alpha}$, since $\text{rank}[R_w(\alpha)]$ takes on only finitely many values. Now, let $M(\alpha)$ be a ρ by ρ submatrix of $R_w(\alpha)$ which has rank ρ at $\underline{\alpha}$ and consider $\det[M(\alpha)]$. Since $\text{rank}[M(\underline{\alpha})] = \rho$ $\det[M(\underline{\alpha})] \neq 0$ verifying that $\det[M(\alpha)]$ is not identically zero. This function is, however, analytic in α (since $f(u,\alpha)$ is analytic in α) and since it is not identically zero, its zero set is nowhere dense in A.[2] Letting

$$B^c = \overline{\{\alpha \in A,\ \det[M(\alpha)] = 0\}} \tag{17.}$$

denote the closure of the zero set and B denote the complement of B^c the hypotheses of theorem 2 of reference 1 are satisfied. Indeed, since B^c is the closure of a nowhere dense set it has Lebesque measure zero[2] while the fact that $\det M[(\alpha)] \neq 0$ for $\alpha \in B$ implies that

$$\rho \geq \text{rank}\ [R_w(\alpha)] \geq \text{rank}[M(\alpha)] = \rho\ ;\ \alpha \in B \tag{18.}$$

hence

$$\text{rank } [R_w(\alpha)] = \rho \ ; \ \alpha \in B \qquad 19.$$

Since the hypotheses of Theorem 2 of reference 1 are satisfied in our case the conclusion of the theorem to the effect that the system is locally diagnosable if and only if $R_w(\alpha_o)$ is nonsingular for some α_o follows.

Consistant with the proof of Theorem 2 rank$[R_w(\alpha)]$ is a generic property (i.e., constant almost everywhere) and hence we may refer to ρ as the *generic rank* of $R_w(\alpha)$ when $f(u,\alpha)$ is analytic in α. As such, one may check the non-singularity of $R_w(\alpha)$ with a randomly chosen α_o and the result of the theorem may be restated via:

Corollary 1: Assume that $f(u,\alpha)$ is analytic in α and let w be any admissible measure for which $R_w(\alpha)$ exists for all $\alpha \in A$. Then the system is locally diagnosable if and only if the generic rank of $R_w(\alpha)$ is k.

Once it has been verified that a system is locally diagnosable it still remains to pick a set of test signals $\{u_1, u_2, \ldots, u_m\}$ and to solve the resultant set of equations

$$\underline{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} f(u_1,\alpha) \\ f(u_2,\alpha) \\ \vdots \\ f(u_m,\alpha) \end{bmatrix} = F(\alpha) \qquad 20.$$

for $\alpha \in A$. For this purpose we require that

$$H_m = \frac{\partial F}{\partial \alpha}(\alpha_0) = \begin{bmatrix} J(u_1,\alpha_0) \\ J(u_2,\alpha_0) \\ \vdots \\ J(u_m,\alpha_0) \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial \alpha_1}(u_1,\alpha_0) & \frac{\partial f}{\partial \alpha_2}(u_1,\alpha_0) & \cdots & \frac{\partial f}{\partial \alpha_k}(u_1,\alpha_0) \\ \frac{\partial f}{\partial \alpha}(u_2,\alpha_0) & & & \\ & & & \\ \vdots & & & \\ \frac{\partial f}{\partial \alpha_1}(u_m,\alpha_0) & & \cdots & \frac{\partial f}{\partial \alpha_k}(u_m,\alpha_0) \end{bmatrix} \quad 21.$$

have column rank k in which case a Newton-Rapson type of algorithm will be assured of converging to α_0 from a sufficiently good initial guess. More generally, if this Jacobian matrix has a generic column rank of k such an algorithm will converge to almost any solution of 17. given a sufficiently good initial guess. Recall from equation 1. that each $\frac{\partial f}{\partial \alpha_i}(u_j,\alpha_0)$ is an element of Y and hence we are dealing with an m by k matrix whose entries are elements of Y.

Theorem 3: Let w be an admissible measure for which $R_w(\alpha)$ exists for all $\alpha \in A$ and assume that $R_w(\alpha_0)$ is nonsingular at a regular point $\alpha_0 \in A$. Then there exists a sequence $u_i \in U$; i=1,2, ... , $m \leq k$; for which $\frac{\partial F}{\partial \alpha}(\alpha_0)$ has column rank k.

Proof: If $\frac{\partial f}{\partial \alpha_1}(u,\alpha_0) = 0$ for all u then the first column $J(u,\alpha_0)$ is zero for all u implying that the first row and the first column of $R_w(\alpha_0)$ is zero. This, however, contradicts the assumption that $R_w(\alpha_0)$ is nonsingular and we may therefore assume that there exists a $u_1 \in U$ for which $\frac{\partial f}{\partial \alpha_1}(u_1,\alpha_0) \neq 0$. As such, there exists $u_1 \in U$ such that H_1 has a column rank of at least 1. Using this fact as the starting point for an inductive hypothesis we assume that there exists u_i; i = 1,2, ... n; such that the matrix H_n has column rank $p < k$ where $n \leq p$. We now desire to verify the existence of a vector $u_{n+1} \in U$

for which the corresponding matrix H_{n+1} has column rank greater than or equal to $p + 1$. To this end we let T be a nonsingular matrix of scalars which operate on the columns of H_n in such a way that the $(p + 1)$st column through the kth column of H_nT is zero. Since T is nonsingular H_{n+1} will have column rank greater than, or equal to, $p + 1$ for some u_{n+1} if and only if $H_{n+1}T$ has column rank greater than, or equal to, $p + 1$. Because of the special form of H_nT, however, this will be the case if, and only if, the bottom row of $H_{n+1}T$ given by

$$\frac{\partial f}{\partial \alpha}(u_{n+1},\alpha_o)T = J(u_{n+1},\alpha_o)T \tag{22.}$$

is non-zero in columns $p + 1$ through k for some u_{n+1}. If this is not the case we may let t denote the $(p+1)$st column of T which is non-zero since T is nonsingular, in which case we have

$$J(u,\alpha_o)t = 0 \tag{23.}$$

for all $u \in U$. This, however, implies that

$$t^*R_w(\alpha_o)t = \int_U t^*J^*(u\alpha_o)J(u\alpha_o)t \, dw(u) = 0 \tag{24.}$$

which contradicts the assumption that $R_w(\alpha_o)$ is nonsingular. As such, there must exist a u_{n+1} for which H_{n+1} has column rank greater than, or equal to $p + 1$. Repeating the argument inductively until an $H_m = \frac{\partial F}{\partial \alpha}(\alpha_o)$ with column rank k is obtained now completes the proof of the theorem. Note that since $n \leq p$ at each step, $m \leq k$.

Our purpose in the above has been to indicate a mechanism by which the existing diagnosibility theory for linear systems[3] and memoryless nonlinear systems can be extended to the general nonlinear dynamical case. Indeed, the

same tools can be used to extend the theory in any number of directions. For instance, one might use the rank of R_w as a measure of the degree to which a circuit fails to be diagnosable. Indeed, the resultant *measure of testability* is a natural generalization of the frequency domain measure of testability introduced in reference 4. Moreover, the frequency domain criterion for choosing test signals introduced by Priester and Clary[5] may also be extended to the case of general nonlinear systems via the above described formulation. Finally, a vector space criterion for diagnosability similar to that of reference 3. may be formulated.

1. Visvanathan, V., and A. Sangiovanni-Vincentelli, "Fault Diagnosis of Nonlinear Memoryless Systems", unpub. notes, Univ. of Calif. Berkeley(1980).

2. Lang, S., Analysis II, Reading, Addison-Wesley, 1969.

3. Sen, N., and R. Saeks, "Fault Diagnosis for Linear Systems Via Multi-frequency Measurements", IEEE Trans. on Circuits and Systems, Vol. CAS-26, pp. 457-465, (1979).

4. Sen, N., and R. Saeks, "A Measure of Testability and its Application to Test Point Selections - Theory", Proc. of the 20th Midwest Symp. on Circuits and Systems, Lubbock, TX, Aug 1977, pp 576-583.

5. Priester, R., and J. Cleary, Unpublished Notes, Research Triangle Inst., Raliegh, NC, 1980

A DATA BASE FOR SYMBOLIC NETWORK ANALYSIS

C.-c. Wu

AND

R. Saeks

A Data Base for Symbolic Network Analysis*

C.-C. Wu and R. Saeks
Texas Tech University
Lubbock, Texas 79409

I. Introduction

Historically, symbolic network analysis has been motivated by the problems of circuit design and, as such, the emphasis has been placed on quickly and efficiently obtaining a symbolic transfer function from a given set of circuit specifications.[2,3] In an operational or maintenance environment, however, one is typically given a prescribed nominal circuit and desires determine the effect of various (possibly large) perturbations thereon. This is the case in a power system where one is given a fixed network and desires to determine the effect of proposed modifications thereto. Alternatively, in the problem of analog circuit fault diagnosis one desires to simulate the effect of a number of alternative failures to compare the simulated data with the observed failure data.[4]

In such an operational or maintenance environment numerous perturbations of the nominal circuit are studied and, as such, significant computational efficiencies can be obtained if one first generates a data base in terms of the nominal circuit parameters and then extracts the appropriate symbolic transfer function from the data base each time a different symbolic transfer is required. Of course the benefit to be achieved via such an approach is dependent on the size of the data base and the ease with which a symbolic transfer function may be retrieved therefrom.

*This research supported in part by the Joint Services Electronics Program at Texas Tech University under ONR Contract 76-C-1136.

The obvious manner in which to generate such a data base is to simply pre-compute the coefficients of all required symbolic transfer functions and store them in the data base. Retreival from such a data base is, of course, immediate but the data base may become overly large. Indeed, the number of transfer functions which must be stored is $O(\underline{k}^{\underline{p}})$ where $\underline{k}$ is the total number of potentially variable circuit parameters and $\underline{p}$ is the maximum number of circuit parameters which may vary simultaneously. An alternative approach is to store the nominal transfer function information and then use Householder's formula[1] to compute the required symbolic transfer functions. In such a data base we need only store $O(\underline{n}^2)$ transfer functions where $\underline{n}$ is the total number of component output terminals but retreival requires $O(\underline{n}^3+p^3)$ multiplications where p is the actual number of circuit parameters which vary simultaneously. Since, in practice, $\underline{n} >> p$ the retreival process requires approximately $O(\underline{n}^3)$ multiplications and is dominated by the large dimensional matrix multiplication required by Householder's formula rather than the low dimensional inverse.

In the present paper we will formulate an alternative data base for the symbolic transfer functions which also requires $O(\underline{n}^2)$ entries, but for which retreival requires only $O(p^3)$ multiplications. Since p is typically small this is tantamount to immediate retreival.

In the remainder of this introduction we will review the properties of the component connection model for a large scale circuit or system[1] which serves as the starting point for our theory. The data base and retreival formulae for the case where $p \leq 2$ are formulated in section 2. while the general retreival formula is derived in section 3. Section 4. is devoted

to the problem of retreiving sensitivity formulae from the data base while section 5. deals with the problem of updating the data base when the nominal circuit parameters are changed. Finally, section 6. is devoted to an example illustrating the theory.

The component connection model is an algebraic model for an interconnected dynamical system which subsumes the classical topological models but is more readily manipulated both analytically and computationally. The motivation and justification of the model are discussed in detail in reference 1 and will not be repeated here. The component connection model takes the form of the set of simultaneous equations

$$b = Z(j\omega)a \tag{1.1}$$

$$a = L_{11}b + L_{12}u \tag{1.2}$$

and

$$y = L_{21}b + L_{22}u \tag{1.3}$$

Here, Z (=Z(jω)) is a frequency dependent matrix characterizing the decoupled system components with composite component input and output vectors a and b, respectively. On the other hand the L_{ij}; i,j = 1,2; matrices are frequency independent connection matrices characterizing the coupling between the composite component vectors, a and b, and the composite system input and output vectors, u and y, respectively.

A little algebra with the component connection equations will readily reveal that

$$S = L_{22} + L_{21}(1 - ZL_{11})^{-1}ZL_{12} \tag{1.4}$$

where S (= S(jω)) is the composite system transfer function matrix[1] characterizing the external behavior of the system via

$$y = S(j\omega)u \tag{1.5}$$

Often, rather than working with the entire S matrix we find it convenient to work with its individual entries; s^{qv}, q = 1,2, ... ,$\underline{q}$ and v = 1,2, ...,$\underline{v}$: which are related to the component connection model via

$$s^{qv} = L_{22}^{qv} + L_{21}^{q}(1 - ZL_{11})^{-1}ZL_{12}^{v} \tag{1.6}$$

Here L_{22}^{qv} is the q-v entry in L_{22}; q = 1,2, ... ,$\underline{q}$ and v = 1,2, ...,$\underline{v}$; L_{21}^{q} is the qth row of L_{21}; q = 1,2, ... ,$\underline{q}$; and L_{12}^{v} is the vth column of L_{12}; v = 1,2, ... , $\underline{v}$.

Finally, since we are interested in analyzing the effects of perturbing one or more components from their nominal values, we decompose Z into nominal and perturbation terms in the form

$$Z = Z_0 + Z_1 \tag{1.7}$$

where

$$Z_1 = \sum_{k=1}^{\underline{p}} c^k \delta^k r^k \tag{1.8}$$

Here, c^k (= $c^k(j\omega)$) is a column vector, r^k (= $r^k(j\omega)$) is a row vector, and δ^k is the scalar perturbation for the kth potentially variable component parameter. In a typical application one is given c^k, r^k, and δ^k ; k = 1,2, ... ,$\underline{k}$; characterizing $\underline{k}$ potentially variable component parameters though at most $\underline{p}$ such parameters vary in any given analysis; $p \leq \underline{p} \leq \underline{k}$. Indeed, $p << \underline{k}$ in most applications. Finally, we note that Z_1 can be expressed more concisely in the form

$$Z = C\Delta R \tag{1.9}$$

where

$$C = [c^1 \,;\, c^2 \,;\, \ldots \,;\, c^p] \tag{1.10}$$

$$R = \begin{bmatrix} \underline{r^1} \\ \underline{r^2} \\ \vdots \\ \underline{r^p} \end{bmatrix} \tag{1.11}$$

and

$$\Delta = \begin{bmatrix} \delta^1 & & & \\ & \delta^2 & & \\ & & \ddots & \\ & & & \delta^p \end{bmatrix} \tag{1.12}$$

The above described notation formulated for the component connection model is summarized in table 1.

Matrix	Type	Dimension	Index
a	composite component input vector	$\underline{m} \times 1$	-
b	composite component output vector	$\underline{n} \times 1$	-
u	composite system input vector	$\underline{v} \times 1$	-
y	composite system output vector	$\underline{q} \times 1$	-
L_{11}	connection matrix	$\underline{m} \times \underline{n}$	-
L_{21}	connection matrix	$\underline{q} \times \underline{n}$	-
L_{21}^{q}	qth row of L_{21}	$1 \times \underline{n}$	$q = 1,2, \ldots ,\underline{q}$
L_{12}	connection matrix	$\underline{m} \times \underline{v}$	-
L_{12}^{v}	vth column of L_{12}	$\underline{m} \times 1$	$v = 1,2, \ldots ,\underline{v}$
L_{22}	connection matrix	$\underline{q} \times \underline{v}$	-
L_{22}^{qv}	q-v entry in L_{22}	1×1	$q = 1,2, \ldots ,\underline{q}$; $v = 1,2, \ldots ,\underline{v}$
S	composite system transfer function matrix	$\underline{q} \times \underline{v}$	-
S^{qv}	q-v entry in S	1×1	$q = 1,2, \ldots ,\underline{q}$; $v = 1,2, \ldots ,\underline{v}$
Z	composite component transfer	$\underline{n} \times \underline{m}$	-
Z_o	nominal composite component	$\underline{n} \times \underline{m}$	-
Z_1	composite component transfer function perturbation matrix	$\underline{n} \times \underline{m}$	-
c^k	column vector characterizing perturbation of kth parameter	$\underline{n} \times 1$	$k = 1,2, \ldots ,\underline{k}$
C	array of the c^k vectors for the parameters chich actual vary (row$[c^k]$)	$\underline{n} \times p$	-
r^k	row vector characterizing	$1 \times \underline{m}$	$k = 1,2, \ldots ,k$
R	array of r^k vectors for the parameters which actually vary (col$[r^k]$)	$p \times \underline{m}$	-
δ^k	kth variable parameter	1×1	$k = 1,2, \ldots ,\underline{k}$
Δ	array of δ^k's for parameters which actually vary (diag$[\delta^k]$)	$p \times p$	-

Table 1. Summary of Component Connection Model

II. The Data Base

Our data base is composed of the following family of (frequency dependent) scalar transfer functions

$$s_o^{qv} = L_{22}^{qv} + L_{21}^{q}(1 - Z_o L_{11})^{-1} Z_o L_{21}^{v} \quad ; \quad q = 1,2, \ldots ,\underline{q}; \; v = 1,2, \ldots ,\underline{v} \qquad 2.1$$

$$b^{qj} = L_{21}^{q}(1 - Z_o L_{11})^{-1} c^{j} \quad ; \qquad q = 1,2, \ldots ,\underline{q}; \; j = 1,2, \ldots ,\underline{k} \qquad 2.2$$

$$d^{kv} = r^{k}[1 + L_{11}(1 - Z_o L_{11})^{-1} Z_o] L_{12}^{v}; \quad k = 1,2, \ldots ,\underline{k} \quad v = 1,2, \ldots ,\underline{v} \qquad 2.3$$

and

$$e^{kj} = r^{k} L_{11}(1 - Z_o L_{11})^{-1} c^{j} \quad ; \quad k,j = 1,2, \ldots ,\underline{k} \qquad 2.4$$

Here, $\underline{q}$ and $\underline{v}$ denote the number of external system inputs and outputs which are typically few in number. As such, the e^{kj} array composed of $\underline{k}^2$ entries dominates the data base. Also note that all of the entries in the data base are formulated in terms of the nominal component values and, as such, the data base may be generated off-line without a priori knowledge of the perturbations to be analyzed. Finally, the entire data base may be generated with the aid of only a single $\underline{n}$ by $\underline{n}$ (sparse) matrix inverse.

Now, if we assume that only a single parameter is perturbed, i.e.

$$Z_1 = c^{k} \delta^{k} r^{k} \qquad 2.5$$

for some fixed $k = 1,2, \ldots ,\underline{k}$, to retrieve s^{qv} from the data base we must evaluate

$$s^{qv} = L_{22}^{qv} + L_{21}^{q}(1 - [Z_o + c^{k} \delta^{k} r^{k}] L_{11})^{-1} [Z_o + c^{k} \delta^{k} r^{k}] L_{12}^{v} \qquad 2.6$$

in terms of the elements of our data base and the variable parameter, δ^{k}. To this end we invoke Householder's formula[1]

$$(W + XY)^{-1} = W^{-1} - W^{-1}X(1 + YW^{-1}X)^{-1}YW^{-1} \qquad 2.7$$

with $W = (1 - Z_oL_{11})$, $X = -c^k\delta^k$, and $Y = r^kL_{11}$ obtaining

$$(1 - [Z_o + c^k\delta^kr^k]L_{11})^{-1} = [(1 - Z_oL_{11}) + (-c^k\delta^k)(r^kL_{11})]^{-1}$$

$$= (1-Z_oL_{11})^{-1} + (1-Z_oL_{11})^{-1}c^k\delta^k(1-4^kL_{11}(1-Z_oL_{11})^{-1}c^k\delta^k)^{-1}r^kL_{11}(1-Z_oL_{11})^{-1}$$

$$= (1-Z_oL_{11})^{-1} = \frac{(1-Z_oL_{11})^{-1}c^k\delta^kr^kL_{11}(1-Z_oL_{11})^{-1}}{1 - \delta^ke^{kk}} \qquad 2.8$$

Now, upon substitution of 2.8 into 2.6 we obtain

$$s^{qv} = L_{22}^{qv} + L_{21}^{q}(1-[Z_o+c^k\delta^kr^k]L_{11})^{-1}[Z_o+c^k\delta^kr^k]L_{12}^{v}$$

$$= L_{22}^{qv} + L_{21}^{q}(1-Z_oL_{11})^{-1}[Z_o+c^k\delta^kr^k]L_{12}^{v}$$

$$+ \frac{L_{21}^{q}(1-Z_oL_{11})^{-1}c^k\delta^kr^kL_{11}(1-Z_oL_{11})^{-1}[Z_o+c^k\delta^kr^k]L_{12}^{v}}{1 - \delta^ke^{kk}}$$

$$= s_o^{qv} + \delta^kb^{qk}r^kL_{12}^{v} + \frac{\delta^kb^{qk}r^kL_{11}(1-Z_oL_{11})^{-1}Z_oL_{12}^{v} + (\delta^k)^2b^{qk}e^{kk}r^kL_{12}^{v}}{1 - \delta^ke^{kk}}$$

$$= s_o^{qv} + \frac{\delta^kb^{qk}d^{kv} + (\delta^k)^2[-b^{qk}e^{kk}r^kL_{12}^{v} + b^{qk}e^{kk}r^kL_{12}^{v}]}{1 - \delta^ke^{kk}} = s_o^{qv} + \frac{\delta^kb^{qk}d^{kv}}{1 - \delta^ke^{kk}} \qquad 2.9$$

which is the desired symbolic transfer function.

If we assume that two parameters are perturbed; that is

$$Z_1 = c^k\delta^kr^k + c^j\delta^jr^j \qquad 2.10$$

a similar formula can be obtained wherein Householder's formula is applied twice. Since this formula is subsumed by the general retreival formula derived in the following section, we simply state the result without proof. In particular,

$$s^{qv} = L_{22}^{qv} + L_{21}^{q}(1 - [Z_o + c^k \delta^k r^k + c^j \delta^j r^j] L_{11})^{-1} [Z_o + c^k \delta^k r^k + c^j \delta^j r^j] L_{12}^{v}$$

$$= s_o^{qv} + \frac{\delta^k{}_b q^k{}_d k^v + \delta^j{}_b q^j{}_d j^v + \delta^k \delta^j (-e^{kk}{}_b q^j{}_d j^v - e^{jj}{}_b q^k{}_d k^v + e^{kj}{}_b q^j{}_d j^v + e^{jk}{}_b q^j{}_d k^v)}{1 - \delta^k e^{kk} - \delta^j e^{jj} + \delta^k \delta^j (e^{kk} e^{jj} - e^{kj} e^{jk})} \qquad 2.11$$

III. Retreival Theorem

As is apparent from equation 2.11, our retreival formulas are quite complex, even for the case p = 2 and, as such, a more compact notation is required if they are to be tractable. To this end, we assume that δ^k; k = 1,2, ... ,p: denote the potentially variable parameters and that

$$Z_1 = \sum_{k=1}^{p} c^k \delta^k r^k = C \Delta R \qquad 3.1$$

Of course, the same expression applies to any set of p potentially variable parameters given an appropriate change of the index set. To obtain the required symbolic transfer function for

$$S = L_{22}+L_{21}(1-[Z_o+Z_1]L_{11})^{-1}[Z_o+Z_1]L_{12} \qquad 3.2$$

with the above specified Z_1 we now define the following matrices made up of elements from our data base

$$S_o = \begin{bmatrix} s_o^{11} & s_o^{12} & \cdots & s_o^{1v} \\ s_o^{21} & s_o^{22} & \cdots & s^{2v} \\ \vdots & \vdots & & \vdots \\ s_o^{q1} & s_o^{q2} & \cdots & s_o^{qv} \end{bmatrix} \qquad 3.3$$

$$B = \begin{bmatrix} b^{11} & b^{12} & \cdots & b^{1p} \\ b^{21} & b^{22} & \cdots & b^{2p} \\ \vdots & \vdots & & \vdots \\ b^{q1} & b^{q2} & \cdots & b^{qp} \end{bmatrix} \qquad 3.4$$

$$D = \begin{bmatrix} d^{11} & d^{12} & \dots & d^{1\underline{v}} \\ d^{21} & d^{22} & \dots & d^{2\underline{v}} \\ \vdots & \vdots & & \vdots \\ d^{\underline{p}1} & d^{\underline{p}2} & & d^{\underline{p}\underline{v}} \end{bmatrix} \qquad 3.5$$

$$E = \begin{bmatrix} e^{11} & e^{12} & \dots & e^{1p} \\ e^{21} & e^{22} & \dots & e^{2p} \\ \vdots & \vdots & & \vdots \\ e^{\underline{p}1} & e^{\underline{p}2} & \dots & e^{\underline{p}p} \end{bmatrix} \qquad 3.6$$

while Δ is defined as per equation 1.12.

THEOREM: Using the above notation

$$S = L_{22}+L_{21}(1-[Z_oZ_1]L_{11})^{-1}[Z_o+Z_1]L_{12} = S_o + B(1 - \Delta E)^{-1}\Delta D$$

Proof: First, we observe that

$$S_o = L_{22} + L_{21}(1-Z_oL_{11})^{-1}Z_oL_{12} \qquad 3.7$$

is just the nominal system transfer function matrix while

$$B = L_{21}(1-Z_oL_{11})^{-1}C \qquad 3.8$$

and

$$D = R[1 + L_{11}(1-Z_oL_{11})^{-1}Z_o]L_{12} = R(1-L_{11}Z_o)^{-1} \qquad 3.9$$

via Householder's formula. Finally,

$$E = RL_{11}(1-Z_oL_{11})^{-1}C$$

where R and C are as defined by equations 1.10 and 1.11. As such,

$$(1 - \Delta E)^{-1} = (1 - \Delta RL_{11}(1-Z_0L_{11})^{-1}C)^{-1}$$

$$= [1 + \Delta RL_{11}(1 - (1-Z_0L_{11})^{-1}C\Delta RL_{11})^{-1}(1-Z_0L_{11})^{-1}C]$$

$$= [1 + \Delta RL_{11}(1 - Z_0L_{11} - Z_1L_{11})^{-1}C]$$

$$= [1 + \Delta RL_{11}(1-ZL_{11})^{-1}C] \qquad 3.10$$

where we have invoked Householder's formula with $Z = 1$, $X = \Delta RL_{11}$, and $Y = (1-Z_0L_{11})^{-1}C$; and equation 1.9. As such,

$$S_0 + B(1 - \Delta E)^{-1}\Delta D = S_0 + L_{21}(1-Z_0L_{11})^{-1}C[1 + \Delta RL_{11}(1-ZL_{11})^{-1}C]\Delta R(1-L_{11}Z_0)^{-1}L_{12}$$

$$= S_0 + L_{21}(1-Z_0L_{11})^{-1}[Z_1 + Z_1L_{11}(1-ZL_{11})^{-1}Z_1](1-L_{11}Z_0)^{-1}L_{12}$$

$$= S_0 + L_{21}(1-Z_0L_{11})^{-1}\{[(1-ZL_{11})+Z_1L_{11}](1-ZL_{11})^{-1}\}Z_1(1-L_{11}Z_0)^{-1}L_{12}$$

$$= S_0 + L_{21}(1-Z_0L_{11})^{-1}(1-Z_0L_{11})(1-ZL_{11})^{-1}Z_1(1-L_{11}Z_0)^{-1}L_{12}$$

$$= S_0 + L_{21}(1-ZL_{11})^{-1}Z_1(1-L_{11}Z_0)^{-1}L_{12}$$

$$= L_{22} + L_{21}(1-Z_0L_{11})^{-1}Z_0L_{12} + L_{21}(1-ZL_{11})^{-1}Z_1(1-L_{11}Z_0)^{-1}L_{12}$$

$$= L_{22} + L_{21}Z_0(1-L_{11}Z_0)^{-1}L_{12} + L_{21}(1-ZL_{11})^{-1}Z_1(1-L_{11}Z_0)^{-1}L_{12}$$

$$= L_{22} + L_{21}[Z_0 + (1-ZL_{11})^{-1}Z_1](1-L_{11}Z_0)^{-1}L_{12}$$

$$= L_{22} + L_{21}(1-ZL_{11})^{-1}[(1-ZL_{11})Z_0 + Z_1](1-L_{11}Z_0)^{-1}L_{12}$$

$$= L_{22} + L_{21}(1-ZL_{11})^{-1}[Z - ZL_{11}Z_0](1-L_{11}Z_1)^{-1}L_{12}$$

$$= L_{22} + L_{21}(1-ZL_{11})^{-1}Z[1-L_{11}Z_0](1-L_{11}Z_0)^{-1}L_{12}$$

$$= L_{22} + L_{21}(1-ZL_{11})^{-1}ZL_{12} = S$$

as required. ////////

IV. Sensitivity Formulae

If one is working directly with the component connection model, it is well known[4] that the sensitivity of S with respect to a parameter, δ^i, can be computed via the formula

$$\left[\frac{dS}{d\delta^i}\right] = L_{21}(1-ZL_{11})^{-1}\left[\frac{dZ}{d\delta^i}\right][1 + L_{11}(1-ZL_{11})^{-1}Z]L_{12} \qquad 4.1$$

and hence it is appropriate to ask whether or not such a sensitivity matrix can also be computed from our data base. Since the expression

$$S = S_o + B(1 - \Delta E)^{-1}\Delta D \qquad 4.2$$

is formally identical to 1.4, if $1 \leq i \leq p$ we may write

$$\left[\frac{dS}{d\delta^i}\right] = B(1-\Delta E)^{-1}M_i[1 + E(1-\Delta E)^{-1}\Delta]D \qquad 4.3$$

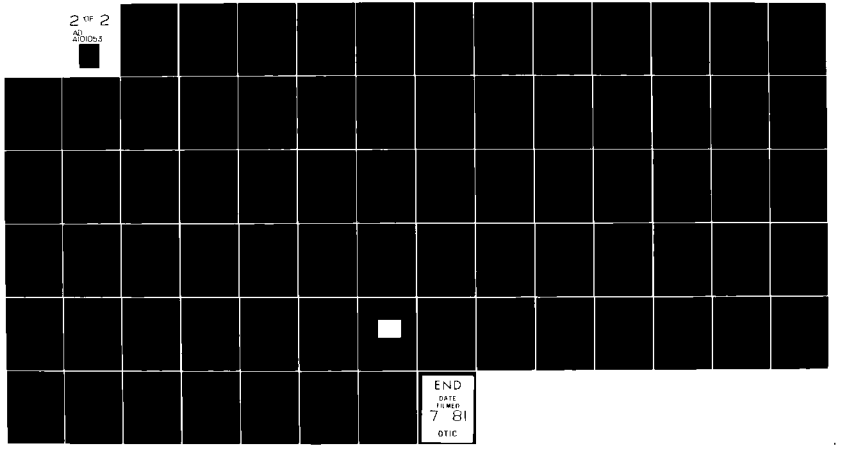
where

$$M_i = \frac{d\Delta}{d\delta^i} = \begin{bmatrix} 0 & & & & & \\ & 0 & & & & \\ & & 0 & & & \\ & & & \ddots & & \\ & & & & 1 & \\ & & & & & \ddots \\ & & & & & & 0 \end{bmatrix} \qquad 4.4$$

with the one appearing in the ith diagonal entry. Clearly, the expression can be computed directly from the data base with the same level of computational effort as required for the retreival formula.

In the case where δ^i is not included in the [illegible]

AD-A101 053 TEXAS TECH UNIV LUBBOCK INST FOR ELECTRONIC SCIENCE F/G 9/3
NONLINEAR FAULT DIAGNOSIS.(U)
MAY 81 R - LIU, K NAKAJIMA, P OLIVIER
UNCLASSIFIED NL



deviate from nominal, i > p in our notation, we must first augment the B, E, D, and Δ matrices to include δ^i and then apply equation 4.3 to the augmented system. To this end we let

$$B^i = \begin{bmatrix} b^{11} & b^{12} & \cdots & b^{1p} & b^{1i} \\ b^{21} & b^{22} & \cdots & b^{2p} & b^{2i} \\ \vdots & \vdots & & \vdots & \vdots \\ b^{q1} & b^{q2} & & b^{qp} & b^{qi} \end{bmatrix} \qquad 4.5$$

$$D^i = \begin{bmatrix} d^{11} & d^{12} & \cdots & d^{1v} \\ d^{21} & d^{22} & \cdots & d^{2v} \\ \vdots & \vdots & & \vdots \\ d^{p1} & d^{p2} & \cdots & d^{pv} \\ d^{i1} & d^{i2} & \cdots & d^{iv} \end{bmatrix} \qquad 4.6$$

$$E^i = \begin{bmatrix} e^{11} & e^{12} & \cdots & e^{1p} & e^{1i} \\ e^{21} & e^{22} & \cdots & e^{2p} & e^{2i} \\ \vdots & \vdots & & \vdots & \vdots \\ e^{p1} & e^{p2} & \cdots & e^{pp} & e^{pi} \\ e^{i1} & e^{i2} & \cdots & e^{ip} & e^{ii} \end{bmatrix} \qquad 4.7$$

and

$$\Delta^a = \left[\begin{array}{c|c} \Delta & 0 \\ \hline 0 & 0 \end{array}\right] \qquad 4.8$$

The we obtain the retreival formulae

$$S = S_o + B^i(1 - \Delta^a E^i)^{-1}\Delta^a D^i \qquad 4.9$$

and

$$\left[\frac{dS}{d\delta^i}\right] = B^i(1-\Delta^a E^i)^{-1}M_{p+1}[1+E^i(1-\Delta^a E^i)^{-1}\Delta^a]D^i \qquad 4.10$$

V. Updating the Data Base

In many applications one uses a data base such as that described above, as a design tool to aid in simulating the effects of various proposed modifications to the system. When such a modification is finally implemented it is then necessary to update the data base to reflect the new nominal parameter values

$$\tilde{Z}_o = Z_o + \sum_{k=1}^{p} c^k \delta^k r^k = Z_o + C\Delta R \qquad 5.1$$

with the aid of Householder's formula we may compute

$$\begin{aligned}(1-\tilde{Z}_o L_{11}) &= [(1-Z_o L_{11}) - C\Delta RL_{11}]^{-1} = (1-Z_o L_{11})^{-1} \\ &+ (1-Z_o L_{11})^{-1} C[1 - \Delta RL_{11}(1-Z_o L_{11})^{-1} C]^{-1} \Delta RL_{11}(1-Z_o L_{11})^{-1} \\ &= (1-Z_o L_{11})^{-1} + (1-Z_o L_{11})^{-1} C(1-\Delta E)^{-1} \Delta RL_{11}(1-Z_o L_{11})^{-1} \qquad 5.2\end{aligned}$$

which upon substitution into equation 2.4 yields

$$\tilde{e}^{kj} = e^{kj} + [e^{k1}\ e^{k2}\ \ldots\ e^{kp}](1-\Delta E)^{-1}\Delta \begin{bmatrix} e^{1j} \\ e^{2j} \\ \vdots \\ e^{pj} \end{bmatrix} \qquad 5.3$$

Similarly,

$$\tilde{b}^{qj} = b^{qk} + [b^{q1}\ b^{q2}\ \ldots\ b^{qp}](1-\Delta E)^{-1}\Delta \begin{bmatrix} e^{1j} \\ e^{2j} \\ \vdots \\ e^{pj} \end{bmatrix} \qquad 5.4$$

$$\tilde{d}^{kv} = d^{kv} + [e^{k1}\ e^{k2}\ \ldots\ e^{kp}](1-\Delta E)^{-1}\Delta \begin{bmatrix} d^{1v} \\ d^{2v} \\ \vdots \\ d^{pv} \end{bmatrix} \qquad 5.5$$

and

$$\tilde{s}_o^{qv} = s_o^{qv} + [b^{q1}\ b^{q2}\ \ldots\ b^{qp}](1-\Delta E)^{-1}\Delta \begin{bmatrix} d^{1v} \\ d^{2v} \\ \vdots \\ d^{pv} \end{bmatrix} \qquad 5.6$$

As such, the entries in our data base can be updated with a computational effort which is commensurate with that required by the retreival formula.

VI. Examples

Consider the simple RC op-amp circuit shown in figure 1. The component connection

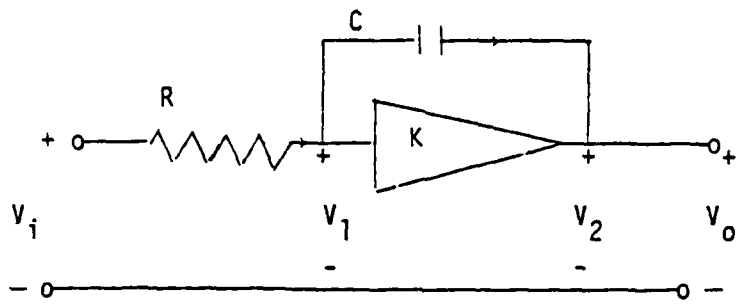


Figure 1: RC Op-amp circuit.

model for this circuit takes the form

$$\begin{bmatrix} i_c \\ v_r \\ v_2 \end{bmatrix} = \begin{bmatrix} sC & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & K \end{bmatrix} \begin{bmatrix} v_c \\ i_r \\ v_1 \end{bmatrix} \qquad 6.1$$

$$\begin{bmatrix} v_c \\ i_r \\ v_1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_c \\ v_r \\ v_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} v_i \qquad 6.2$$

$$\begin{bmatrix} v_o \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_c \\ v_r \\ v_2 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} v_i \qquad 6.3$$

Thus if all components taken to have nominal values of 1 we obtain

$$(1-Z_o L_{11}) = \begin{bmatrix} 1 & s & s \\ -1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \qquad 6.4$$

$$(1-Z_oL_{11})^{-1} = \begin{bmatrix} 1 & 0 & -s \\ 1 & 1 & -s \\ -1 & -1 & s+1 \end{bmatrix} \qquad 6.5$$

$$(1-Z_oL_{11})^{-1}Z_o = \begin{bmatrix} s & 0 & -s \\ s & 1 & -s \\ -s & -1 & s+1 \end{bmatrix} \qquad 6.6$$

$$L_{11}(1-Z_oL_{11})^{-1} = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & -s \\ -1 & -1 & s \end{bmatrix} \qquad 6.7$$

and

$$L_{11} + L_{11}(1-Z_oL_{11})^{-1}Z_o = \begin{bmatrix} 1 & 0 & -1 \\ s & 1 & -s \\ -s & -1 & s+1 \end{bmatrix} \qquad 6.8$$

Now, we may represent perturbations in the parameters C, R, and K via the matrices

$$c^1\delta^1r^1 = \begin{bmatrix} s \\ 0 \\ 0 \end{bmatrix} \delta^1 \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \qquad 6.9$$

$$c^2\delta^2r^2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \delta^2 \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \qquad 6.10$$

and

$$c^3\delta^3 r^3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \delta^3 \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \qquad 6.11$$

Combining the appropriate c^k and r^j matrices with the above expressions as per equations 2.1 thru 2.4 we obtain the data base

$$s_o = 1 \qquad 6.12$$

$$b^1 = -s \quad b^2 = -1 \quad b^3 = s+1 \qquad 6.13$$

$$d^1 = 1 \quad d^2 = 0 \quad d^3 = 1 \qquad 6.14$$

and

$$\begin{aligned} e^{11} &= 0 \quad e^{12} = 0 \quad e^{13} = -1 \\ e^{21} &= s \quad e^{22} = 0 \quad e^{23} = -s \\ e^{31} &= -s \quad e^{32} = -1 \quad e^{33} = s \end{aligned} \qquad 6.15$$

where we have deleted the q and v indices since we are dealing with a single-input single-output system.

Now, if one desires to compute the symbolic transfer function with respect to perturbations in the op-amp gain we have

$$S(s,\delta^3) = s_o + \frac{b^3\delta^3 d^3}{1-\delta^3 e^{33}} = \frac{1+\delta^3}{1-\delta^3 s} \qquad 6.16$$

Recalling that δ^3 represents a perturbation from a nominal parameter value of $K_o = 1$ our actual gain is $K = K_o + \delta^3 = 1+\delta^3$, which upon substitution into 6.16 yields

$$S(s,K) = \frac{K}{(1-K)s + 1} \qquad 6.17$$

which is the classical gain formula for such a circuit.

Finally, if we desire to update our data base to reflect a new nominal value for the circuit parameters of C = 1, R = 1, and K = 2 we invoke equations 5.3 thru 5.6 with $\delta^3 = 1$ yielding

$$\tilde{s}_o = s_o + \left.\frac{b^3\delta^3 d^3}{1 - \delta^3 e^{33}}\right|_{\delta^3=1} = \frac{2}{1 - s} \qquad 6.18$$

$$\tilde{e}^{11} = e^{11} + \frac{e^{13}\delta^3 e^{31}}{1 - \delta^3 e^{33}} = 0 + \left.\frac{(-1)\delta^3(-s)}{1 - \delta^3(s)}\right|_{\delta^3 = 1} = \frac{s}{1 - s} \qquad 6.19$$

and similarly for the other elements of the data base.

VII. Conclusions

The preceeding development has been motivated by operational and maintenance considerations rather than the design considerations. In such an environment one typically deals with a fixed nominal system, but carries out repeated analyses thereon. As such, the cost of generating the required data base is secondary compared to the cost of storing the data base and retreiving information therefrom. In these respects we believe that our data base is near optimal. Since the number of system inputs and outputs is typically small our data base contains approximately $\underline{k}^2$ elements (actually $\underline{k}^2 + \underline{k}(\underline{v}+\underline{q}) + \underline{v}\underline{q}$) where $\underline{k}$ is the total number of parameters which are potentially variable. This data base, however, contains sufficient information to permit one to retreive symbolic transfer functions for any number $p \leq \underline{k}$ of variable parameters. Indeed, the number of variable parameters in a symbolic transfer function is reflected only in the cost of retreival which is on the order of p^3 multiplications (actually $p^3 + p^2\underline{v} + p\underline{v}(\underline{q}+1)$). Since p is typically small, say five or less, this is minimal.

VIII. References

1. DeCarlo, R.A., and R. Saeks, Interconnected Dynamical Systems, New York, Marcel Dekker, (to appear).

2. Lin, P.-M., "Symbolic Network Functions by a Single Path Finding Algorithm", Proc. of the 7th Alertown Conf. on Circuits and Systems, Univ. of Illinois, Oct. 1969, pp. 196-205.

3. Lin,P.-M., et al."SNAP - A computer Program for Generating Symbolic Network Functions", Sch. of Elec. Engrg., Purdue Univ., Report TR-EE70-16, Aug 1970.

4. Puri, N.N. "Symbolic Fault Diagnosis Techniques", In Rational Fault Analysis (ed. R. Saeks and S.R. Liberty), New York, Marcel Dekker, 1977.

ON LARGE NONLINEAR PERTURBATIONS OF LINEAR SYSTEMS

P.D. Olivier and R. Saeks

Proceedings of the 12th Asilomar Conference
on Circuits, Systems and Computers

Pacific Grove, CA., pp. 473-477, Nov. 1978

On Large Nonlinear Perturbations of Linear Systems

Philip D. Olivier

R. Saeks

Department of Electrical Engineering
Texas Tech University
Lubbock, Texas 79409

Abstract

This paper generalizes the classical Householder's formula to certain nonlinear operators. This class of nonlinear operators is shown to be common in circuit theory. Several examples are provided that show where these operators occur and the result is applied.

I. INTRODUCTION

The purpose of this paper is to present a technique for analyzing lumped analog systems with some linear and some nonlinear elements. It is shown that such a system is described by an operator of the form

$$B+YoD \tag{1.1}$$

where B and D are linear and Y is nonlinear. Since no assumptions are made about the nature of the nonlinearities, it is impossible to view the operator YoD as small in any sense, hence, YoD has to be viewed as a large nonlinear perturbation of the linear operator B.

The technique to be presented is based on a theorem that allows us to invert (1.1) in two steps. First, invert the linear operator B. If there are N_L linear elements and N_N nonlinear elements, B will be an $(N_L+N_N)x(N_L+N_N)$ matrix. Second, invert a nonlinear operator of rank N_N. That such a result exists, is not surprising. Those experienced in solving equations involving such operators apply Gaussian elimination until there are N_N nonlinear equations in N_N unknowns. Another way to see that this segregation can be accomplished is to view the nonlinear elements as a "load" on an appropriate linear circuit, in much the same way as a circuit where one nonlinear element is analyzed by viewing that element as the load and finding the Thevenin's Equivalent circuit that it sees.

The main result of this paper is obtained by generalizing to operators the form B+YoD, a classical theorem concerning linear operators known as Householder's Formula. This classical result and its generalization are stated and proven in section 2. In section 3, we show such operators do,

indeed, occur in circuit theory and then two examples are presented. The results are summarized in section 4.

Section 2.

The classical Householder's formula [1] provides a means of calculating the inverse of the matrix B+CD in terms of B^{-1} and $(I+DB^{-1}C)^{-1}$. If B^{-1} is known and if the dimensions of C and Dare appropriate, then a great savings in time and effort can be realized using this technique.

Theorem 1: (Classical Householder's Formula)

If B is an NxN matrix, C is an NxP matrix, and D is a PXN matrix, then $(B+CD)^{-1} = B^{-1}-B^{-1}C(I+DB^{-1}C)^{-1}DB^{-1}$.

In the nonlinear extension, the linear operator C is replaced by the nonlinear operator Y. The proof of this extension looks, at first glance, like the proof of a linear rather than a nonlinear theorem. To see that this is indeed a nonlinear result, the differences between the nonlinear and linear operator algebra will be reviewed by giving two basic definitions.

Definition 1: (Operator Addition) Let f and g be two operators (linear or nonlinear) with the same domain, then the operator f+g is by the following

$$(f+g)(x)\triangleq f(x)+g(x) \qquad (2.1)$$

Definition 2: (Linearity) an operator f is linear if for all x and y in its domain and all scalars α and β

$$f(\alpha x+\beta y) = f(\alpha x)+f(\beta y)$$

$$=\alpha f(x)+\beta f(y).$$

The argument distributes to the left for all operators, but the operator distributes to the right only for linear operators. With this distinction in mind, we are ready to state and prove our main result which is a closed form expression for $(B+YD)^{-1}$ in terms of B^{-1} and $(I+DB^{-1}Y)^{-1}$ (of course, operator multiplication is to be interpreted as composition, i.e., $YD \triangleq Y \circ D$.

Theorem 2: If i)B and D are linear operators, ii)B^{-1} exists, iii)y is an arbitrary operator and iv)B+YD is defined, then

$$(B+YD)^{-1} = B^{-1}-B^{-1}Y(I+DBY)^{-1}DB^{-1}. \qquad (2.3)$$

Proof: Consider the operator X+XYX where X is linear and Y is possibly nonlinear.

$$X+XYX = X(I+YX) = I+XY)X.$$

If (I+YX) and (I+YX) are both invertible (it can be shown [2] that one is invertible if and only if the other is) we have

$$(I+XY)^{-1}X = X(I+YX)^{-1} \qquad (2.4)$$

Now consider the identity

$$I = (I+YX)(I+XY)^{-1} = I(I+YX)^{-1}+YX(I+YX)^{-1}$$

$$= (I+YX)^{-1}+Y(I+XY)^{-1}X$$

where we have used (4). Solving for $(I+YX)^{-1}$ yields

$$(5)\quad (I+YX)^{-1} = I-Y(I+XY)^{-1}X.$$

Finally, consider the operator B+YD.

$$(B+YD)^{-1} = [(I+YDB^{-1})B]^{-1} = B^{-1}(I+YDB^{-1})^{-1}$$

Letting $X = DB^{-1}$ in (5) yields

$$(B+YD)^{-1} = B^{-1}[I-Y(I+DB^{-1}Y)^{-1}DB^{-1}]$$
$$= B^{-1}-B^{-1}Y(I+DB^{-1}Y)^{-1}DB^{-1}.$$

To see how this result is useful, consider the case where B+YD is an N^{th} order nonlinear operator, D a linear operator that maps $|R^N \rightarrow |R^P$, P<N and Y a nonlinear operator that maps $|R^P \rightarrow |R^N$. This result allows the solution of N nonlinear equation in N unknowns to be replaced by the solution of N linear equations in N unknowns and also the solution of P (recall P<N) nonlinear equations in P unknowns. Thus, we have, via a closed form expression, ordered our equations and unknowns properly to make maximum use of linear techniques and minimum use of nonlinear techniques.

It should be noted that the proof of Theorem 2 relied on the fact that B and D were linear operators and allowed Y to be arbitrary. B and D were not assumed to be matrices and Y was not assumed to map $|R^P \rightarrow |R^N$. Any or all of the operators could be differential operators and the result would still be valid. Regardless of whether the operators aredifferential or functional, we have succeeded in breaking it up into a linear portion and a nonlinear portion. If there are few nonlinear components in comparison with the number of linear components, the nonlinearities can be viewed as a perturbation on the linear system.

Section 3

The purpose of this section is to show that operators of the form

B+YD occur in circuit analysis problems and to apply Theorem 2 to two examples.

This type of operator arises naturally in nonlinear network analysis. Consider the Node analysis of a network with reduced incidence matrix A, [3]. Kirchoff's Laws are

$$(KCL)\ A\underline{j} = 0$$

$$(KVL)\ \underline{v} = A^T\underline{e}.$$

The branch equations might be

$$\underline{j} = G\underline{v}+\underline{j}_s-G\underline{v}_s+f(\underline{v})$$

Where

$\underline{j} \triangleq$ the branch current vector;

$\underline{v} \triangleq$ the branch voltage vector;

$\underline{j}_s \triangleq$ the current source vector;

$\underline{v}_s \triangleq$ the voltage source vector;

$\underline{e} \triangleq$ the mode-to-datum voltage vector;

G is assumed to be an "invertible" matrix of differential operators, f is a nonlinear differential operator, and all branches are voltage controlled. If

$$\underline{i} \triangleq AGV_s-A\underline{j}_s$$

then Kirchoff's Laws and the branch equations can be combined to yield

$$(AGA^T)\underline{e}+Af(A^T\underline{e}) = \underline{i}. \qquad (3.1)$$

Letting $AGA^T=B$, and $Af(\cdot)=Y$, we see that this operator is of the desired form. The typical situation is for f to be a function of only a few ($p<n$) linear combinations of the components of $\underline{v}$ then (3.1) can be rewritten in the form

$$B+Y(CA^T)\ \underline{e}=i \tag{3.2}$$

which is precisely the type of operator that is amenable to the results of Theorem 2.

We now apply Theorem 2 to the problem of solving two nonlinear simultaneous equations in two unknowns. Consider the following nonlinear equations

$$f(X) = z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}X_1 - 1X_2 + X_1^3 \\ \frac{1}{2}X_2 + \frac{1}{2}X_2 \end{bmatrix} = \begin{bmatrix} 24 \\ 3 \end{bmatrix} .$$

In order to apply Theorem 2, $f(x)$ must be put into the form

$$(B+YD)(X)$$

where B is an invertible matrix. One way to do this is

$$f(X) = (B+YD)x = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + Y([1,0]X)$$

where

$$Y(.) = \begin{bmatrix} (.)^3 - (.) \\ 0 \end{bmatrix} .$$

Theorem 2 says that

$$X = B^{-1}z - B^{-1}Y(I+DB^{-1}Y)^{-1}DB^{-1}z = X_L - X_{NL}$$

where

$$X_L = B^{-1}z = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 24 \\ 3 \end{bmatrix} \quad \begin{bmatrix} 27 \\ -21 \end{bmatrix}$$

and

$$\begin{aligned} X_{NL} &= B^{-1}Y(1+DB^{-1}Y)^{-1}DB^{-1}Z \\ &= B^{-1}Y(1+DB^{-1}Y)^{-1}DB_L \\ &= B^{-1}Y(I+DB^{-1}Y)^{-1}27. \end{aligned}$$

Now

$$(I+DB^{-1}Y)^{-1}27 = u$$

is equivalent to

$$(I+DB^{-1}Y)y = 27.$$

$$(I+DB^{-1}Y)u = u+[1,0]\begin{bmatrix}1 & 1\\ -1 & 1\end{bmatrix}\begin{bmatrix}u^3 & -u\\ 0 & \end{bmatrix}$$

$$= u+[1 \quad 1]\begin{bmatrix}u^3 & -u\\ 0 & \end{bmatrix} = u+u^3-u=u^3=27$$

which implies

$$u = 3$$

Now

$$X_{NL} = B^{-1}Y(3) = \begin{bmatrix}1 & 1\\ -1 & 1\end{bmatrix}\begin{bmatrix}(3)^3 & -3\\ 0 & \end{bmatrix} = \begin{bmatrix}24\\ -24\end{bmatrix}$$

So

$$X = X_L - X_{NL} = \begin{bmatrix}27-24\\ -21+24\end{bmatrix} \quad \begin{bmatrix}3\\ 3\end{bmatrix} .$$

The reason for choosing a functional example is that for large circuits or systems, the differential equations are solved numerically so at each iteration an operator of the form B+YD must be inverted. To see that this is indeed the case, consider discretizing the differential equations obtained from the component connection model [2] of a system. The component equations are assumed (here) to be given in state form

$$\dot{X} = f(X,a)$$

$$b = g(X,a).$$

where a is the vector of component inputs, b is the vector of component outputs, and x is the state vector of the components. The connection equations

(KVL and KCL equations) are given by

$$\begin{bmatrix} a \\ y \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} b \\ u \end{bmatrix}$$

where u is the vector of system inputs and y is the vector of system outputs. If we order the entries in all of the vectors correctly, we can partition the vectors in the following manner

$$a = \begin{bmatrix} a^N \\ a^L \end{bmatrix}, \quad b = \begin{bmatrix} b^N \\ b^L \end{bmatrix}, \text{ and } x = \begin{bmatrix} x^N \\ x^L \end{bmatrix}$$

where the superscript N(L) denotes entries associated with the nonlinear (linear) components. The discretized equations that the computer is to solve have the form

$$\sum_{i=0}^{r} d_i X_{k-i}^N = f^N(X_k^N, a_k^N),$$

$$\sum_{i=0}^{r} d_i X_{k-i}^L = AX_k^L + Ba_k^L,$$

$$b_k^N = g^N(X_k^N, a_k^N),$$

$$b_k^L = CX_k^L + Da_k^L,$$

$$a_k^N = L_{11}^{NN} b_k^N + L_{11}^{NL} b_k^L + L_{12}^N u_k,$$

$$a_k^L = L_{11}^{LN} b_k^N + L_{11}^{LL} b_k^L + L_{12}^L u_k,$$

$$Y_K = L_{21}^N b_k^N + L_{22}^L b_k^L + L_{22} u_k.$$

The last equation is just the output equation and is not used during the iterations. These equations can be put in the following form

$$TW_k + Y(DW_k) =$$

$$\begin{bmatrix} -Ido & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A-doI & B & 0 & 0 \\ 0 & 0 & C & D & 0 & -I \\ 0 & I & 0 & 0 & -L_{11}^{NN} & -L_{11}^{NL} \\ 0 & 0 & 0 & I & -L_{11}^{NN} & -L_{11}^{LL} \end{bmatrix} \begin{bmatrix} X_k^N \\ a_k^N \\ X_k^L \\ a_k^L \\ b_k^N \\ b_k^L \end{bmatrix}$$

$$\begin{bmatrix} f^N(X_k^N, a_k^N) \\ g^N(X_k^N, a_k^N) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{r} d_i X_{r-i}^N \\ \sum_{i=1}^{r} d_i X_{r-i}^L \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where D= I,0 , and I is conformable with

$$\begin{bmatrix} X_k^N \\ a_k^N \end{bmatrix}$$

4. Conclusion

The classical Householder's Formula has been generalized to certain nonlinear operators. It was shown that these nonlinear operators occur in

circuit theory, both in the differential equations that describe the circuit and in the discretized equations that are used in the computer aided analysis of these circuits. It is hoped that this result will be as useful a tool in the fault analysis of nonlinear circuits as the classical result turned out to be in the fault analysis of linear circuits.

References

1. A.S. Householder, "A Survey of Some Closed Methods for Inverting Matrices", SIAM Jour. on Appl. Math., Vol. 5, pp. 155-169, (1957).
2. Saeks, R., and R.A. DeCarlo, Interconnected Dynamical Systems, New York, Marcel Dekker, (to appear).
3. Desoer, C.A. and S.E. Kuh, Basic Circuit Theory, McGraw-Hill, New York, 1969, pp. 423-425.

NONLINEAR OBSERVER'S AND FAULT ANALYSIS

P.D. Olivier and R. Saeks

22nd Midwest Symposium on Circuits and Systems

University of Pennsylvania, Philadelphia, PA.
pp. 535-536, June 1979

NONLINEAR OBSERVER'S AND FAULT ANALYSIS

P. D. Olivier and R. Saeks
Department of Electrical Engineering
Texas Tech University
Lubbock, Texas 79409

Abstract

A fault analysis algorithm appropriate for time varying and nonlinear systems, is developed. The algorithm essentially constructs an observer for a nonlinear system that is intimately related to the system under test.

INTRODUCTION

Given enough time and computing capability brute force searches will identify possible fault sets. The real problem in fault analysis is to construct algorithms that, in some sense, locate the fault sets "efficiently". "Efficiently", in this context, means that the fault isolation must be done relatively quickly and with limited on site computing. Such techniques have been developed to handle linear time invariant and digital systems.[2,3,4] These, however, make heavy use of the defining properties for these systems, and do not generalize. The purpose of this paper is to show that an observer for an appropriate nonlinear differential equation can be utilized, on line, to determine the values of the system parameters. A technique, based on optimal control theory, for constructing such observers is also presented.

OBSERVERS AND FAULT ANALYSIS

Consider testing a system that is described by the nonlinear state equations

$$\dot{x}_1 = f(x_1, a, u, t)$$
$$y = g(x_1)$$

where x_1 is the dynamical state vector, a is the vector of parameters to be estimated (they are assumed constant over the test time), and u is the input used in the test procedure. If we want to estimate a we need to include it in the state vector, i.e., we want to build an observer for the augmented differential equation

$$\begin{bmatrix} \dot{x}_1 \\ \dot{a} \end{bmatrix} = \begin{bmatrix} f(x_1,a,u,t) \\ 0 \end{bmatrix}$$

$$y = g(x_1).$$

If it is possible to build an observer that will observe the subvector a we have solved the fault analysis problem. It would then be necessary to justify our solution in terms of time and computation requirements.

AN OBSERVER DESIGN

We chose to design an observer with the following structure

$$\begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{a}} \end{bmatrix} = \begin{bmatrix} f(\hat{x},\hat{a},u,t) \\ 0 \end{bmatrix} + H(\hat{y} - y) \quad (H \text{ time invariant})$$

$$\hat{y} = g(\hat{x}_1).$$

we term such an observer as a Model reference linear time invariant observer. The term "linear time invariant" is used because the residuals enter in a linear time invariant fashion. The problem is now that of choosing H. To avoid involved stability considerations (at least initially) we choose H so that it minimizes the following function

$$J(H) = \int_{t_o}^{t_1} [(x_1-x)^2+(a-a)^2] \, dt$$

and hope that the stability takes care of itself. The construction of H can be done by solving the following optimal control problem[1]

$$\min_{H \in \mathbb{R}^2} J(H) \quad \left(J(H) = \int_{t_0}^{t_1} X^T Q X dt, \quad Q = \begin{bmatrix} I & -I \\ -I & I \end{bmatrix} \right)$$

subject to the differential equations constraints

$$\dot{X} = \begin{bmatrix} x_1 \\ a \\ \dot{\hat{x}}_1 \\ \dot{\hat{a}} \end{bmatrix} = \begin{bmatrix} f(x_1,a,u,t) \\ 0 \\ f(\hat{x}_1,\hat{a},u,t) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ H \\ \end{bmatrix} (\hat{y}-y)$$

$$X(t_0) = [x_1(t_0)^T, a(t_0)^T, \hat{x}_1(t_0)^T, \hat{a}(t_0)^T]^T$$

Note. 1) H will be dependent on the $X(t_0)$ used in its construction, so when it is used to estimate a (when a differs from $a(t_0)$) it has little chance of being the optimal H. So even though we use optimization techniques to construct H, it will not, in general, be optimal. 2) Several observers may need to be constructed, each one convergent for a in a different region.

Experience indicates that only a few components fail at a time. Because of this a reasonable approach is to construct an observer for each component, (thereby minimizing the dimension of the augmented state vector) and estimate the parameters for each component in parallel. Observers can also be built for the common two and three element faults.

REFERENCES

1. D.E. Kirk, Optimal Control Theory, Prentice Hall, Englewood Cliff's, New Jersey, 1970.

2. R. Saeks, and S.R. Liberty, eds., Rational Fault Analysis, Marcel Dekker, Inc., New York, 1977.

3. R. Saeks, N. Sen, H.M.S. Chen, K.S. Lu, S. Sangani, and R.A. DeCarlo, "Fault Analysis in Electronic Circuits and Systems II". Technical Report, Texas Tech University, 1978.

4. N. Sen, M.S. Thesis, Texas Tech University, Lubbock, Texas, 1975.

A DUAL-MODE FAULT-DIAGNOSIS TECHNIQUE FOR ANALOG NON-LINEAR ELECTRONIC SYSTEMS

Quoc Dinh Ngo

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The advent of microelectronics, the ever-increasing complexity and compactness of electronic circuits, together with a need for higher reliability in space, military, and even commercial projects, has brought new problems to industry; test and diagnosis of electronic circuits is one of them.

Presently, a printed circuit board may include several hundred components; modular construction as well as small geometrical dimensions make impractical, even infeasible, and certainly uneconomical, the conventional test methods based on classical laboratory equipment such as signal generators, meters, oscilloscopes, and probes.

In either the analog or digital case, fault detection and location in electronic systems is generally performed via measurements at a limited number of input and output connections. These measurements are then executed by computer test programs to provide diagnosis.

Until now, algorithms for automatically generating test programs have been concerned mainly with digital circuits. Analog circuits, on the other hand, have received far less attention, due to several reasons: Analog systems are frequently non-linear, and the values of the parameters of the elements exhibit large deviations [6]; analog signals are inherently more complex than digital signals. They occur continuously in time, rather than at discrete times, and their values have infinite resolution, instead of being truncated into a finite number of bits; most importantly, digital automatic test generation has been successful due to the simplified modeling at the logic gate or higher level, rather than the internal parameter level as in the case of analog systems. As a result, most analog automatic test generation and fault isolation techniques require a large computational capability on the ATE or off-line computers [5].

Several efforts have been made to attack the fault diagnosis problem in analog circuits. The multi-frequency technique for fault analysis in general linear dynamical systems was developed by N. Sen and R. Saeks [1],[7],[8],[9], was considered to be more efficient and advanced in terms of output selection and reduction. By varying the test frequency at the same test points, the number of test points can be reduced significantly compared to the case of single frequency measurements.

This technique has been generalized to non-linear analog systems by linearization of the non-linear components. Unfortunately, the linearization concept fails in many cases [10]. For example, considering a single-loop circuit consisting of a power supply, a resistor, and a tunnel diode. At a given bias, the non-linear characteristic of the tunnel diode has a particular slope. With the breakdown of the bias resistor, the linearization of the non-linear characteristic of the tunnel diode at the new biasing point will be different, and the diode will appear to be faulty if one ignores the fact that the diode is operating at a different bias.

Recognizing the linearization problem in multi-frequency test, the fault-diagnosis of non-linear analog systems in the D.C. case was studied by V. Visvanathan and A. Sangiovanni-Vincentelli [2], and N. David and A.N. Wilson [3] at the component parameter level. These techniques have these disadvantages: the required number of test points is more than required by the multiple frequency technique; the tremendous amount of computer time is required to solve very complicated non-linear equations. Since this is a D.C. test, these techniques can only be applied to memoryless systems (without reactive components such as capacitors and inductors).

The dual-mode technique for fault-diagnosis for non-linear analog systems is introduced as a compromise between the above approaches.

Mode 1 (A.C. Test): utilizing the multi-frequency technique to search for the faulty linear components, then the faulty linear component values are calculated. The non-linear components of the circuit under

diagnosis are replaced by small signal A.C. or linearization equivalent circuits. Therefore, this A.C. test is performed at the internal parameter level for both linear and non-linear components.

Mode 2 (D.C. Test): this test is used to diagnose a reduced non-linear problem after the linear components are determined in Mode 1. The nominal values or the faulty values given by Mode 1 are used for computations depending on whether the linear components are in tolerance or not. A non-linear device or element is treated as a blackbox with inputs and outputs. The task is to find the D.C. inputs and outputs or the operating conditions of the blackbox, not the internal parameters of the non-linear devices. This approach is compatible to previous successful techniques in digital systems performed at the logic gate level or higher. The fault-diagnosis algorithm is shown in Figure 1.

The Mode 2 formulation and examples are presented in this thesis together with a comparison with existing techniques in fault-diagnosis for non-linear analog systems. Finally, an experiment on a four-transistor amplifier circuit is conducted to verify the theorectical solutions and to determine the sensitivity of the dual-mode approach.

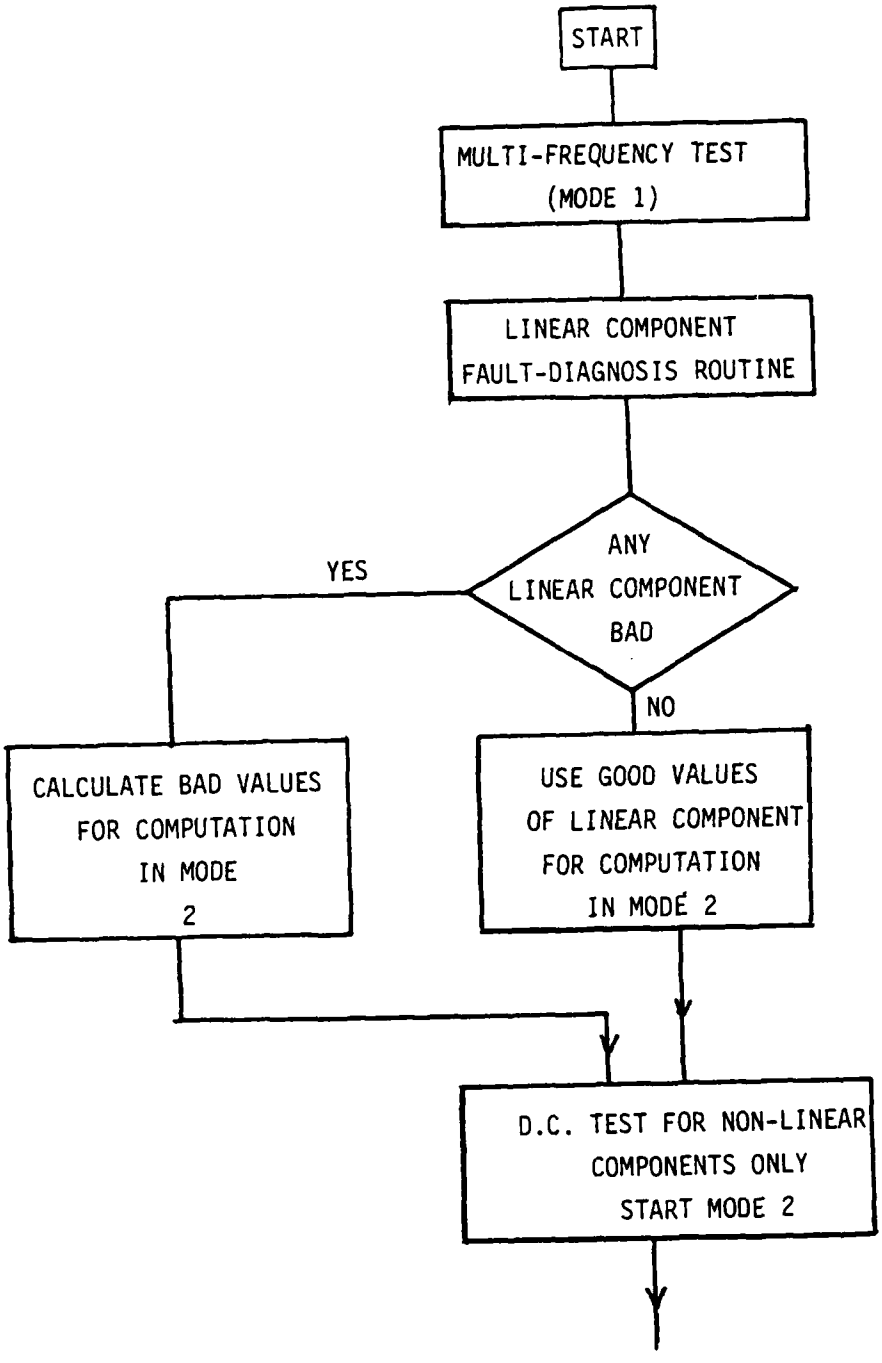


Fig. 1 Fault-Diagnosis Algorithm by the Dual-Mode Technique

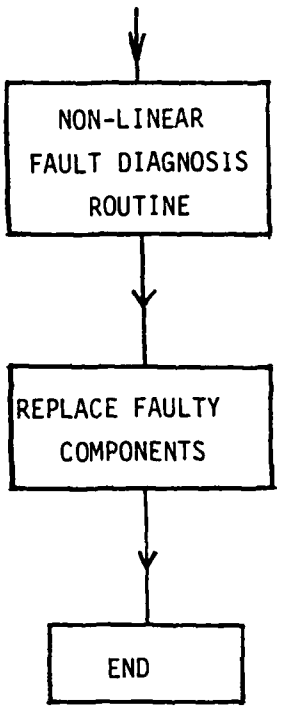


Fig.1 Fault-Diagnosis Algorithm by the Dual-Mode Technique (Continued)

CHAPTER II

THEORETICAL DEVELOPMENT

The theoretical formulation of the D.C. test for the non-linear problem is based on the component connection equations [4] :

$$a = M_{11}b + M_{12}u$$

$$b = M_{21}b + M_{22}u$$

Where u and y represent the vectors of accessible inputs and outputs which are available to the test systems, a and b represent the component input and output vectors, respectively. The relationship between a and b is:

$$b = Z\ a$$

Although the symbol Z is used, the components are not assumed to be represented by an impedance matrix. Indeed, hybrid models are used in most of our examples.

Before the circuit is analyzed in the D.C. test, the capacitors and inductors, which are assumed to be linear components, are replaced by open-circuits and short-circuits respectively.

The component vectors a and b are partitioned into:

$$a = \begin{bmatrix} a_L \\ a_N \end{bmatrix} \qquad b = \begin{bmatrix} b_L \\ b_N \end{bmatrix}$$

Where a_L and b_L are the linear component input and output vectors, and a_N and b_N are the non-linear component input and output vectors. Therefore:

$$\left[\begin{array}{c} b_L \\ \hline b_N \end{array}\right] = \left[\begin{array}{c|c} Z_L & \\ \hline & \dfrac{df_N}{da_N} \end{array}\right] \left[\begin{array}{c} a_L \\ \hline a_N \end{array}\right]$$

when a linearized model is employed.

The component connection matrix is partitioned accordingly:

$$\begin{bmatrix} a_L \\ a_N \\ \hline y \end{bmatrix} = \left[\begin{array}{cc:c} M_{11}^{LL} & M_{11}^{LN} & M_{12}^{L} \\ M_{11}^{L} & M_{11}^{NN} & M_{12}^{N} \\ \hline M_{21}^{L} & M_{21}^{N} & M_{22} \end{array}\right] \begin{bmatrix} b_L \\ b_N \\ \hline u \end{bmatrix}$$

Equation (1) and (2) can be solved simultanously to yield:

$$b_N = \left[M_{21}^{L}\left(Z_L^{-1} - M_{11}^{LL}\right)^{-1} M_{11}^{LN} + M_{21}^{N}\right]^{-1} \left[y - \left(M_{21}^{L}\left(Z_L^{-1} - M_{11}^{LL}\right)^{-1} M_{12}^{L} + M_{22}\right) u\right] \quad (3)$$

$$a_N = \left[M_{11}^{NL}\left(Z_L^{-1} - M_{11}^{LL}\right)^{-1} M_{11}^{LN} + M_{11}^{NN}\right] b_N + \left[M_{11}^{NL}\left(Z_L^{-1} - M_{11}^{LL}\right)^{-1} M_{12}^{L} + M_{12}^{N}\right] \quad (4)$$

Thus the inputs and outputs of the non-linear devices can be computed by a few simple matrix operations. These values are checked against the operating characteristics of the corresponding functional devices for fault isolation.

The coefficient matrices of (3) and (4) can be pre-computed if the linear components are not faulty. The matrix Z_L should be changed to incorporate the faulty values of the linear components, if any, to avoid the computational errors caused by the use of nominal values of the linear components.

For each pair of non-linear component input-output signals. one test point is required. For instance, a bipolar transistor can be modeled with two (2) input-output pairs:

$$a_N = \begin{bmatrix} I_B \\ I_C \end{bmatrix} \qquad b_N = \begin{bmatrix} V_{BE} \\ V_{CE} \end{bmatrix}$$

Therefore, two (2) measurements must be taken. Non-linear integrated circuits can be modeled in the same manner. The number of test points required in this mode goes up linearly with the number of non-linear devices in systems.

The matrix $\left[M_{21}^{L}(Z_L^{-1} - M_{11}^{LL})^{-1}M_{11}^{LN} + M_{21}^{N}\right]$ is singular if the test points are not chosen properly. The selection of test points to make the above matrix non-singular will be discussed in detail in the Appendix.

CHAPTER III

EXAMPLES

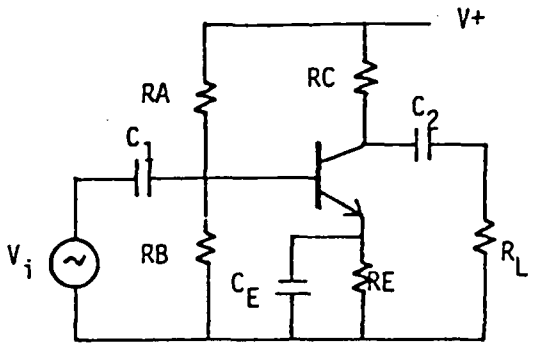


Fig. 2(a) Single-Transistor Amplifier

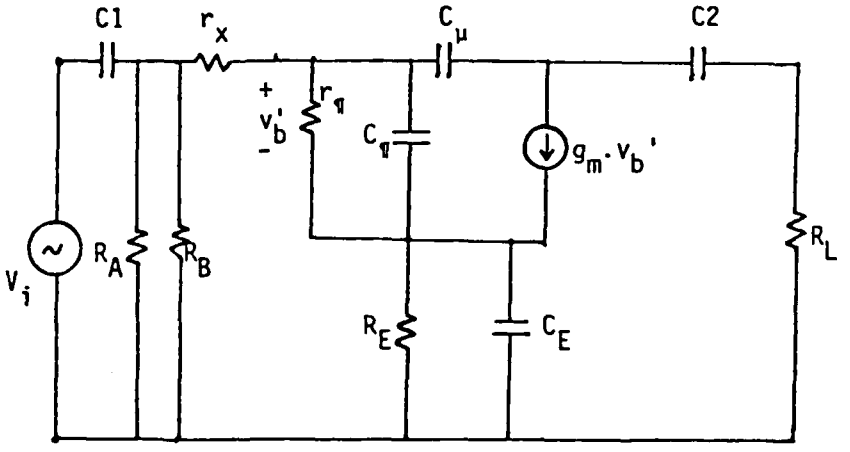


Fig. 2(b) Small Signal A.C. Equivalent Circuit

Multi-Frequency Test (Mode 1) [1], [12]

The small-signal A.C. equivalent circuit of the circuit of Fig. 1(a) is shown in Fig. 1(b). The connection equations associated with the A.C. or mode 1 test are as follows :

$$
\begin{bmatrix} IC1 \\ Ir_x \\ Ir_\pi \\ IC_\mu \\ IC2 \\ VR'_A \\ VRE \\ VC_\pi \\ VCE \\ Vg_m \\ VRC \\ VRL \\ \hline V_o \\ IC1 \\ VR'_A \\ IE \end{bmatrix} =
\left[\begin{array}{cccccccccccc:c}
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
-1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
-1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hdashline
-1 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}\right]
\begin{bmatrix} VC1 \\ Vr_x \\ Vr_\pi \\ VC_\mu \\ VC2 \\ IR'_A \\ IRE \\ IC_\pi \\ ICE \\ Ig_m \\ IRC \\ IR1 \\ \hline V_i \end{bmatrix}
$$

Here we initially allow V_o , IC1 , VR'_A ,and IE to be taken as test outputs. The measure of testability δmin is used to extract a reduced set of test outputs from these options. According to table I two (2) is the minimum number of test outputs in this example , which suffices to yield δmin = 0 (perfect testability) or to provide locally unique solutions for the fault-diagnosis equations .

The test measurements are taken at the two output VR'_A and IE at twelve (12 = n - δmin) distinct frequencies , where n is the dimension of the parameter vector r .The faulty parameters can be identified by using the Householder's Formula and the optimization algorithm[13] . Only the faulty parameter values for the linear components of the circuit need to be calculated to be included in the calculations in mode 2.

Table 1 Measure of Testability for the Amplifier Circuit of Figure 1

OUTPUT	$\delta_{MIN.}$
V_0	3
IC1	2
VR'_A	2
IE	3
V_0,IC1	0
V_0,VR'_A	1
V_0,IE	0
IC1,VR'_A	2
IC1,IE	1
VR'_A,IE	0
V_0 , IC1 , VR'_A	0
V_0 , IC1 , IE	0
V_0 , VR'_A , IE	0
IC1 , VR'_A , IE	0
V_0 , IC1 , VR'_A , IE	0

D.C. Analysis (Mode 2)

Assuming that the capacitors of the amplifier circuit are not shorted. They are treated as open-circuits in D.C. test, and removed from the circuit before Mode 2 analysis begins. The test point measurements are chosen at the same test points used in the multi-frequency test. The connection equations for this circuit in D.C. test are:

$$\begin{bmatrix} VRA \\ IRB \\ IRE \\ IRC \\ VBE \\ VCE \\ \hline VRB \\ IRE \end{bmatrix} = \left[\begin{array}{cccccc:c} 0 & -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ \hline 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{array}\right] \begin{bmatrix} IRA \\ VRB \\ VRE \\ VRC \\ IB \\ IC \\ \hline V+ \end{bmatrix}$$

and

$$\begin{bmatrix} IRA \\ VRB \\ VRE \\ VRC \\ \hline IB \\ IC \end{bmatrix} = \left[\begin{array}{ccc:c} 1/RA & & & \\ & RB & & \\ & & RE & \\ & & & RC \\ \hline & & & df_N \\ & & & da_N \end{array}\right] \begin{bmatrix} VRA \\ IRB \\ IRE \\ VRC \\ \hline VBE \\ VCE \end{bmatrix}$$

The non-linear component input and output vectors a_N and b_N can be found via (3) and (4):

$$b_N = \begin{bmatrix} IB \\ IC \end{bmatrix} = \begin{bmatrix} -\frac{RA+RB}{RA.RB} & 0 \\ \frac{RA+RB}{RA.RB} & 1 \end{bmatrix} \left(\begin{bmatrix} VRB \\ IRE \end{bmatrix} - \begin{bmatrix} \frac{RB}{RA+RB} \\ 0 \end{bmatrix} V+ \right)$$

$$a_N = \begin{bmatrix} VBE \\ VCE \end{bmatrix} = \begin{bmatrix} -RE - \frac{RA.RB}{RA+RB} & -RE \\ -RE & -RC-RE \end{bmatrix} \begin{bmatrix} IB \\ IC \end{bmatrix} + \begin{bmatrix} \frac{RB}{RA+RB} \\ 1 \end{bmatrix} V+$$

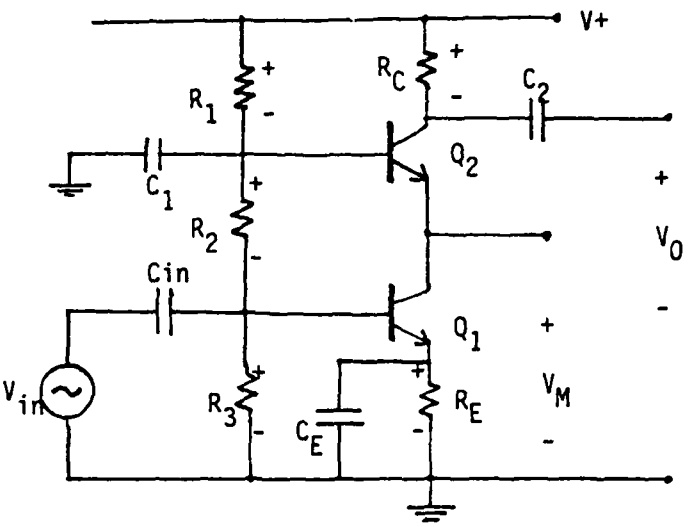


Fig. 3 Cascode Amplifier Circuit

The capacitors in the circuit are assumed to be good components. They are removed from the circuit before Mode 2 diagnosis begins. The connection equations for the circuit are:

$$
\begin{bmatrix} VR1 \\ VRC \\ IR2 \\ IR3 \\ IRE \\ VBE2 \\ VBE1 \\ IC2 \\ IC1 \\ \hline VRE \\ IR1 \\ IRC \\ VM \end{bmatrix} =
\left[\begin{array}{ccccccccc:c}
0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & -1 & 1 \\
1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & -1 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\hdashline
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0
\end{array}\right]
\begin{bmatrix} IR1 \\ IRC \\ VR2 \\ VR3 \\ VRE \\ IB2 \\ IB1 \\ VCE2 \\ VCE1 \\ \hline V+ \end{bmatrix}
$$

$$b = \begin{bmatrix} IR1 \\ IRC \\ VR2 \\ VR3 \\ VRE \\ IB2 \\ IB1 \\ VCE2 \\ VCE1 \end{bmatrix} = \left[\begin{array}{cccc|c} 1/R1 & & & & \\ & 1/RC & & & \\ & & R2 & & \\ & & & R3 & \\ & & & & RE \\ \hline & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{array}\right] \begin{bmatrix} VR1 \\ VRC \\ IR2 \\ IR3 \\ IRE \\ VBE2 \\ VBE1 \\ IC2 \\ IC1 \end{bmatrix}$$

The non-linear inputs and outputs are determined via (3) and (4):

$$\begin{bmatrix} IB2 \\ IB1 \\ VCE2 \\ VCE1 \end{bmatrix} = \frac{1}{RE.R2} \begin{bmatrix} -R3 & RE.R & R3.RE & 0 \\ R3+R2 & -RE.R & -(R2+R3)RE & 0 \\ 0 & 0 & -R2.RE.RC & -R2.RE \\ -R2.RE & 0 & 0 & R2.RE \end{bmatrix} \left[\begin{bmatrix} VRE \\ IR1 \\ IRC \\ VM \end{bmatrix} - \begin{bmatrix} RE/(RE+RC) \\ 1/R \\ 1/(RE+RC) \\ RE/(RE+RC) \end{bmatrix} V+ \right]$$

$$\begin{bmatrix} VBE2 \\ VBE1 \\ IC2 \\ IC1 \end{bmatrix} = \begin{bmatrix} \frac{-R1(R2+R3)}{R}-K & \frac{-R1.R3}{R}-K & \frac{K}{RC} & \frac{-K}{RE} \\ \frac{-R1.R3}{R}-K & \frac{-R3(R1+R2)}{R}-K & \frac{K}{RC} & \frac{K}{RC} \\ \frac{-K}{RC} & \frac{-K}{RC} & \frac{-K}{RE.RC} & \frac{-K}{RE.RC} \\ \frac{K}{RE} & \frac{-K}{RC} & \frac{-K}{RE.RC} & \frac{-K}{RE.RC} \end{bmatrix} \begin{bmatrix} IB2 \\ IB1 \\ VCE2 \\ VCE1 \end{bmatrix} + \begin{bmatrix} \frac{R2+R3}{R}-\frac{K}{RC} \\ \frac{R3}{R}-\frac{K}{RC} \\ \frac{K}{RE.RC} \\ \frac{K}{RE.RC} \end{bmatrix} V+$$

where $R = R1 + R2 + R3$

$$K = \frac{RE.RC}{RE+RC}$$

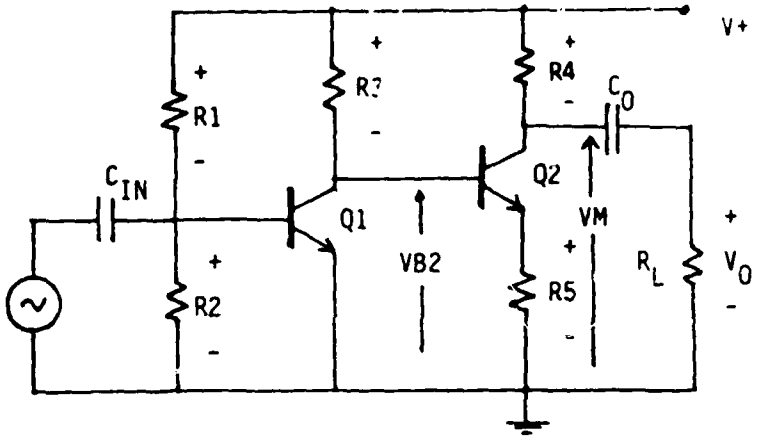


Fig. 4 Direct-Coupled Two-Stage Amplifier

The component-connection equations for Mode 2 analysis are:

$$
\begin{bmatrix} VR1 \\ VR2 \\ IR3 \\ IR4 \\ IR5 \\ IB1 \\ VBE2 \\ VCE1 \\ VCE2 \\ VM \\ I+ \\ VR5 \\ VB2 \end{bmatrix} = \left[\begin{array}{ccccccccc:c} 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1 \\ \hdashline 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} IR1 \\ IR2 \\ VR3 \\ VR4 \\ VR5 \\ VBE1 \\ IB2 \\ IC1 \\ IC2 \\ V+ \end{bmatrix}
$$

$$b = \begin{bmatrix} IR1 \\ IR2 \\ VR3 \\ VR4 \\ VR5 \\ \hline VBE1 \\ IB2 \\ IC1 \\ IC2 \end{bmatrix} = \left[\begin{array}{cccc|c} 1/R1 & & & & \\ & 1/R2 & & & \\ & & R3 & & \\ & & & R4 & \\ & & & & R5 \\ \hline & & & & \\ & & & & \\ & & & & \frac{df_N}{da_N} \\ & & & & \end{array}\right] \begin{matrix} VR1 \\ VR2 \\ IR3 \\ IR4 \\ IR5 \\ \hline IB1 \\ VBE2 \\ VCE1 \\ VCE2 \end{matrix}$$

The matrix $(M_{21}^{L} (Z_L^{-1} - M_{11}^{LL})^{-1} M_{11}^{LN} + M_{21}^{N})$ is non-singular:

$$\begin{bmatrix} -1/R1 & 0 & 0 & 0 \\ -1/R1 & 1 & 1 & 1 \\ 0 & R5 & 0 & R5 \\ 0 & -R3 & -R3 & 0 \end{bmatrix}$$

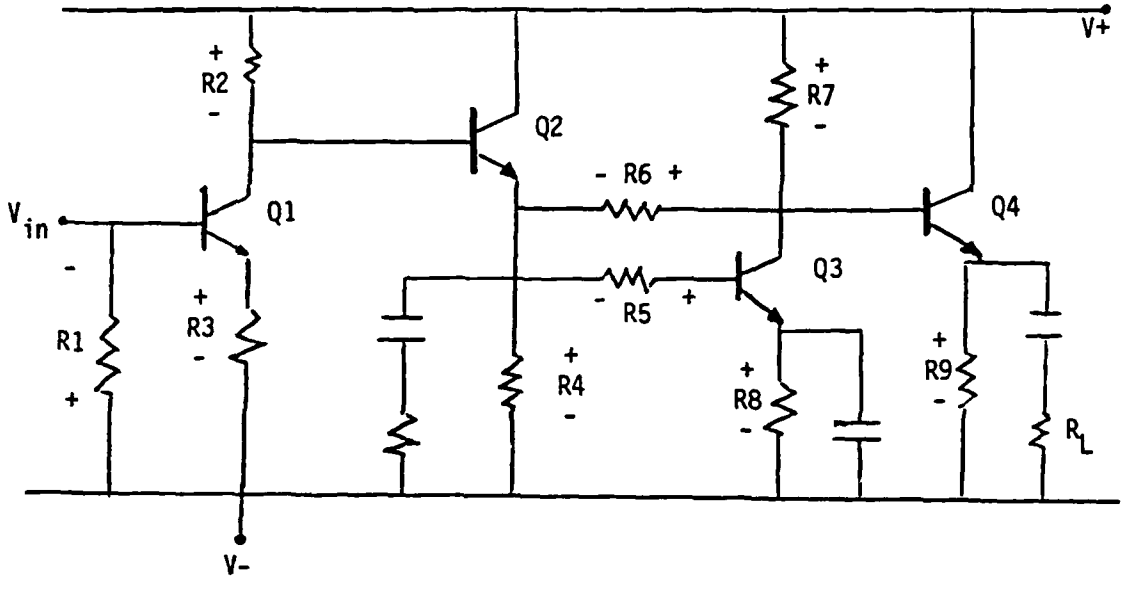


Fig. 5 Video Amplifier Circuit

The same circuit is used as an example in [3]. In [3] ten (10) test points are required besides the input terminals. The solutions or the values of the parameters are obtained by solving eighteen (18) non-linear equations. On the other hand, the dual-mode technique requires eight (8) test points besides the input terminals and the solutions can be obtained by straight-forward matrix manipulations.

After the capacitors are removed, the connection equations are:

$$
\left[\begin{array}{c}
IR1 \\ IR3 \\ IR2 \\ IR4 \\ IR5 \\ IR6 \\ IR7 \\ IR8 \\ IR9 \\
VBE1 \\ VCE1 \\ VBE2 \\ VCE2 \\ VBE3 \\ VCE3 \\ VBE4 \\ VCE4 \\ \hline
VR2 \\ VR3 \\ VR7 \\ VR1 \\ VR9 \\ VR8 \\ VR4 \\ IC4
\end{array}\right]
=
\left[\begin{array}{ccccccccccccccccc|cc}
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&-1&0&-1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
-1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1\\
0&-1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&-1\\
0&0&-1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&1&1&0&0&-1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&-1&-1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&1&0&1&0&0&-1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&1&0\\ \hline
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0
\end{array}\right]
\left[\begin{array}{c}
VR1 \\ VR3 \\ VR2 \\ VR4 \\ VR5 \\ VR6 \\ VR7 \\ VR8 \\ VR9 \\
IB1 \\ IC1 \\ IB2 \\ IC2 \\ IB3 \\ IC3 \\ IB4 \\ IC4 \\ \hline
V+ \\ V-
\end{array}\right]
$$

The non-linear inputs and outputs can be found via (3) and (4):

$$b_N = \begin{bmatrix} IB1 \\ IC1 \\ IB2 \\ IC2 \\ IB3 \\ IC3 \\ IB4 \\ IC4 \end{bmatrix} = \begin{bmatrix} 0 & R2 & R2 & 0 & 0 & 0 & 0 & 0 \\ R3 & R3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & R7 & 0 & 0 \\ R1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & R9 & R9 \\ 0 & 0 & 0 & 0 & R8 & R8 & 0 & 0 \\ 0 & 0 & R4 & R4 & -R4 & 0 & -R4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} - 1 \begin{bmatrix} VR2 \\ VR3 \\ VR7 \\ VR1 \\ VR9 \\ VR8 \\ VR4 \\ IC4 \end{bmatrix}$$

$$a_N = \begin{bmatrix} VBE1 \\ VCE1 \\ VBE2 \\ VCE2 \\ VBE3 \\ VCE3 \\ VBE4 \\ VCE4 \end{bmatrix} = \begin{bmatrix} -R1-R3 & -R3 & 0 & 0 & 0 & 0 & 0 & 0 \\ -R3 & -R3-R2 & -R2 & 0 & 0 & 0 & 0 & 0 \\ 0 & -R2 & -R2-R4 & -R4 & R4 & 0 & R4 & 0 \\ 0 & 0 & -R4 & -R4 & R4 & 0 & R4 & 0 \\ 0 & 0 & R4 & R4 & -R4-R5 & -R8 & -R4 & 0 \\ 0 & 0 & 0 & 0 & -R8 & -R7-R8 & 0 & 0 \\ 0 & 0 & R4 & R4 & -R4 & 0 & -R4-R6-R9 & -R9 \\ 0 & 0 & 0 & 0 & 0 & 0 & -R9 & -R9 \end{bmatrix} b_N$$

$$+ \begin{bmatrix} 0 & -1 \\ 1 & -1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} V+ \\ V- \end{bmatrix}$$

CHAPTER IV

COMPARISON OF FAULT-DIAGNOSIS TECHNIQUES FOR NON-LINEAR SYSTEMS

Method of V. Visvanathan and A. Sangiovanni-Vincentelli [2]

Referring to the single-transistor amplifier circuit of Fig. 6(a) and its equivalent circuit of Fig. 6(b) using Ebers-Moll Model [11].

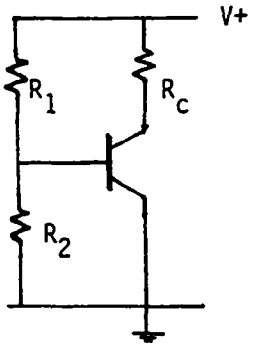


Fig. 6(a)

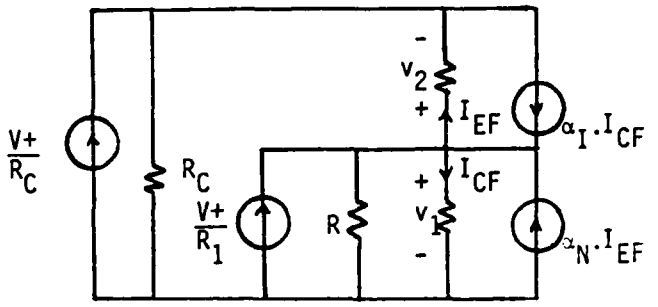


Fig. 6(b)

where $R = R1//R2$

$I_{EF} = K_1(\exp(\lambda_1 v_1)-1)$

$I_{CF} = K_2(\exp(\lambda_2 v_2)-1)$

The node equations at C and B are :

$$(V+/RC) - \alpha_I I_{CF} + I_{EF} = (v_1 - v_2)/RC \qquad (5)$$

$$(V+/R1) + \alpha_N I_{EF} - I_{CF} = v_1/R \qquad (6)$$

Substitute I_{EF} and I_{CF} into (5) and (6) to obtain:

$$V+/RC - \alpha_I K_2(\exp(\lambda_2 v_2)-1) + K_1(\exp(\lambda_1 v_1)-1) = (v_1 - v_2)/RC \qquad (7)$$

$$V+/R1 - \alpha_N K_1(\exp(\lambda_1 v_1)-1) - K_2(\exp(\lambda_2 v_2)-1) = v_1/R \qquad (8)$$

Suppose v_1 is chosen to be the test point measurement, by solving (7) and (8) simultanously, v_1 can be expressed as :

$$y = v_1 = f(u, R_1, R_2, R_C)$$

where u is the input voltage, V+ in this case .

The Jacobian matrix of y with respect to R_1, R_2, and R_C, which is the basis of V. Visvanathan and A. Sangiovanni-Vincentelli formulation, must be computed numerically. This example, on the most simple circuit, has thus shown the impracticality of the above method.

Method of Nasrollah David and A. N. Wilson [3]

Referring to the single transistor amplifier circuit in [3]. This circuit requires five (5) test points, excluding the input terminals. Also, the parameter values can be obtained by solving eight (8) non-linear equations.

Dual-Mode Fault-Diagnosis

Mode 1: referring to the single-transistor amplifier circuit in [1], the set of test point measurements V'_{Ra} and I_E yields the perfect testability ($\delta_{min} = 0$) implying that the fault diagnosis equations have locally unique solutions.

Mode 2: referring to the circuit in Figure 1 of Example 1 in the previous sections, the same test points can be used to measure V_{RB} and I_E. In general case, the set of test points for the two modes will overlap each other.

Summary

Computationally, the dual-mode fault-diagnosis technique uses only straightforward matrix manipulations. This is an advantage over solving non-linear equations. The required number of test points in the dual-mode technique is much less than that of Wilson's technique. Only one test point is required by Sangiovanni-Vincentelli's method but the trade-off is to solve a very complicated set of non-linear equations.

CHAPTER V

FAULT-DIAGNOSIS ALGORITHM FOR TRANSISTORS

The calculations in Mode 2 of the dual-mode fault-diagnosis technique are performed to provide the D.C. operating points of the non-linear devices. In the case of bipolar transistors, the operating conditions can be determined by the collector current I_C, the base current I_B, the base-emitter voltage V_{BE}, and the collector-emitter voltage V_{CE}. The fault-diagnosis algorithm for bipolar transistors is developed on the basis of their known operating charac

Table 2 Transistor Operating Modes

Modes	I_B	I_C	V_{BE}	V_{CE}
Active	I_B	$\beta_N . I_B$	$\simeq .6$ v	$V_{CE} > V_{BE}$
Saturation	I_B	$\beta_S . I_B$	$\simeq .7$ v	$V_{CE} < V_{BE}$
Cut-off	0	0	$< .45$v	$V_{CE} \simeq V_{CC}$

where β_N is the nominal current transfer ratio in active mode, β_S is the saturated current transfer ratio, and V_{CC} is the power supply voltage connected to the collector.

The algorithm for field-effect transistors is even simpler because the input or gate current I_G is always zero. Field-effect transistors are characterized by the gate voltage V_G, the drain voltage V_D, and the drain current I_D.

The fault-diagnosis algorithm for bipolar transistors shown in Figure 7, is used to analyze the experimental results in the following section.

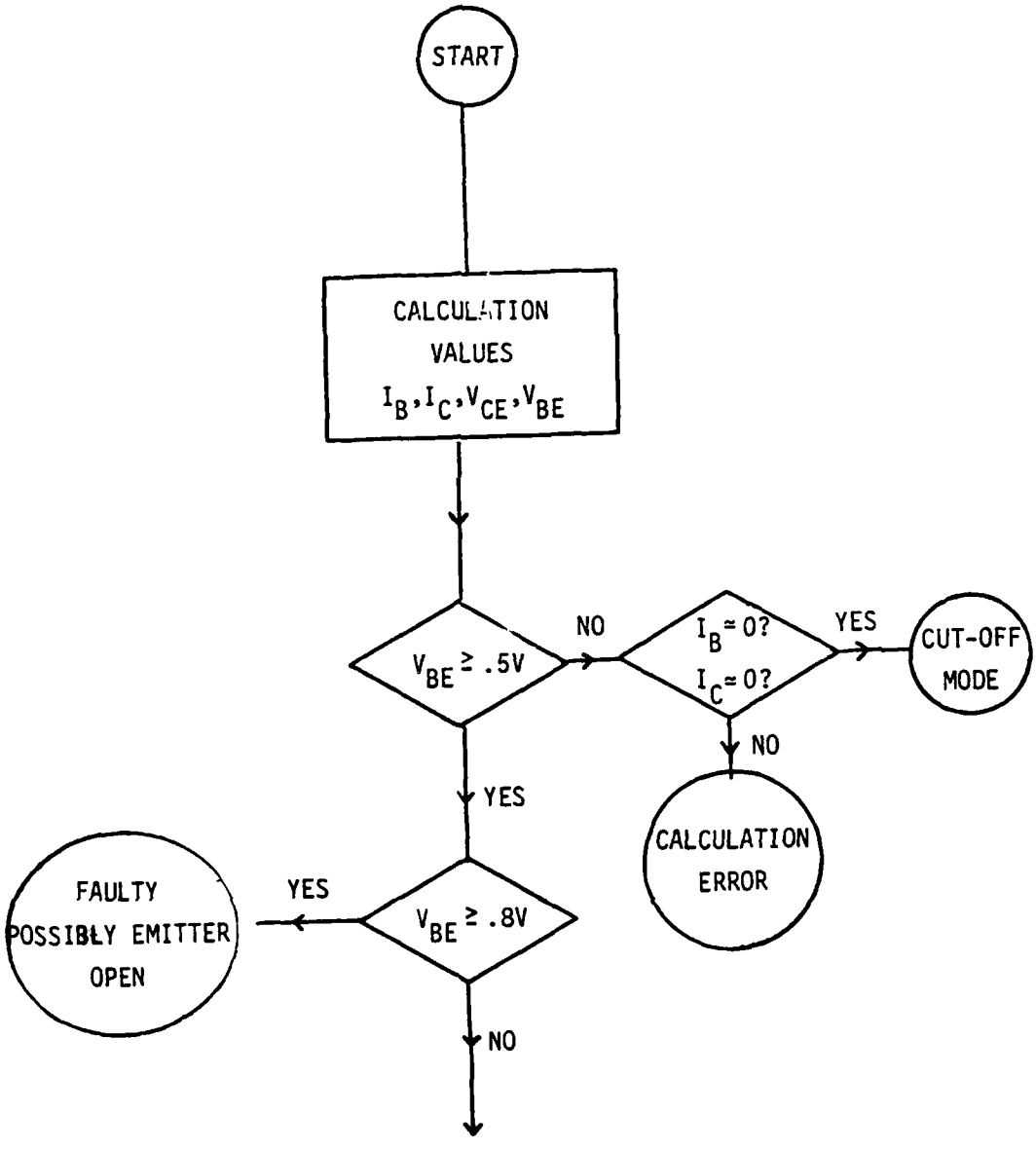


Fig. 7 Fault Diagnosis Algorithm for N-P-N Bipolar Transistors

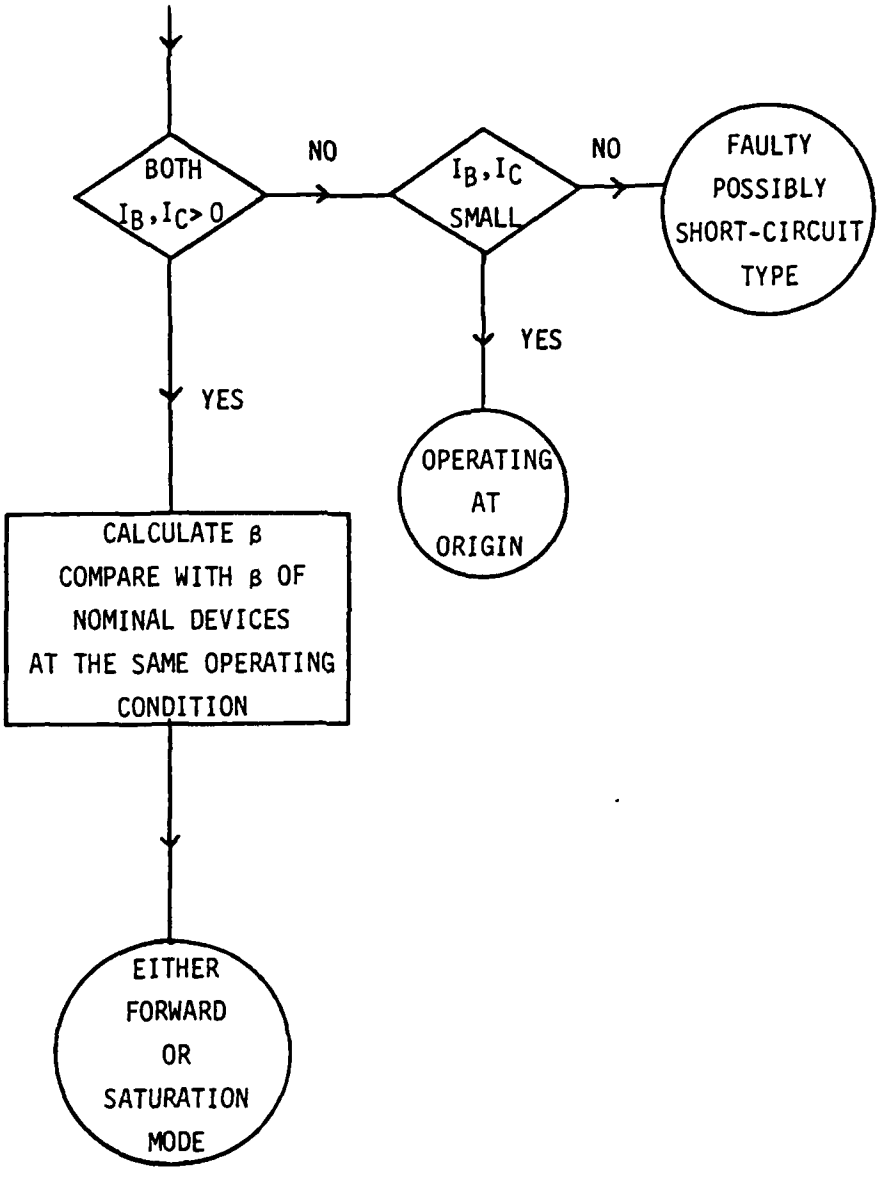


Fig. 7 Fault Diagnosis Algorithm for N-P-N Bipolar Transistors (Continued)

CHAPTER VI

EXPERIMENTAL RESULTS

The video amplifier circuit of Example 4 was built and tested at nominal operating conditions and at intentionally faulty conditions. The measured values of the non-linear devices'operating conditions are compared to those obtained by calculation. The computation in this particular example is simple enough to be carried out by a programmable hand-held calculator. The computational error are investigated experimentally in the following cases.

Case I. Nominal Operating Conditions

The components of the circuit, which consist of four (4) transistors and nine (9) resistors, were carefully analyzed before the experiment was started. The four transistors 2N2222A were checked on a curve tracer. Their betas or current gains varied between one hundred twenty (120) to two hundred sixty (260). Thus there is more than one hundred percent (100%) variation among the various transistors. Furthermore, these transistors are highly sensitive to temperature. For example, a transistor that carries twice the amount of collector current than another will generate more heat and change its characteristics. The temperature sensitivity has a major effect on the accuracy of the test results. All the resistors were within five percent (5%) tolerance. The measured values and the nominal values of the resistors are compared in Table 3. The experimental test-point mea surements for all cases of the experiment is tabulated in Table 4.

The experiment was performed in a temperature-controlled environment at twenty degrees Celcius (20° C). The measurements at the test points were taken and used to compute the transistors' operating conditions with two (2) sets of the resistor or linear component values. The two (2) sets of computed values for the transistors' operating conditions are tabulated in Table 5.

The errors that are produced by using the manufacturer's claimed values for the resistors are less than ten percent (10%) for the collector currents, the base-emitter and the collector-emitter voltages.

Table 3 Comparison of Measured and Nominal Values of Resistors

Resistor	Measured values	Claimed values (Nominal)	Percent error
R1	1.189 K	1.20 K	-.90
R2	3.298 K	3.30 K	-.06
R3	5.654 K	5.60 K	.96
R4	1.173 K	1.2 K	-2.25
R5	.316 K	.33 K	-4.24
R6	.325 K	.33 K	-1.50
R7	1.010 K	1.00 K	-1.00
R8	1.461 K	1.50 K	-2.60
R9	3.280 K	3.30 K	-.61

Table 4 Test Point Measurement for All Cases of the Experiment at V+ = 26.2 Volts and V- = -28.0 Volts

Test point	Units	Case I	Case II	Case III	Case IV
VR2	volt	16.04	28.52	15.87	15.86
VR3	volt	27.31	25.66	27.32	27.32
VR7	volt	6.04	0.00	9.827	5.336
VR1	volt	.0264	1.611	.0263	.0263
VR9	volt	8.83	0.00	0.00	0.00
VR8	volt	8.800	0.00	14.32	0.00
VR4	volt	9.492	0.00	15.04	16.44
IC4	mA	2.665	0.00	12.82	.00866

Table 5 Transistor Operating Conditions of Case I

Parameter	Computed operating points based on		Measured operating points(ii)	% Error (i) vs.(ii)
	Measured R	Claimed R (i)		
IB1	22.2	22.00	22.04 uA	.18
IB2	56.76	70.33	56.33 uA	24.90
IB3	47.11	-169.3	41.75 uA	-505.5
IB4	31.07	14.76	28.09 uA	-47.50
IC1	4.808	4.855	4.805 mA	10.40
IC2	8.113	7.748	8.094 mA	-4.27
IC3	5.978	6.038	5.993 mA	.75
IC4	2.665	2.665	2.665 mA	0.00
VBE1	.6636	.6636	.656 v	1.16
VBE2	.6640	.6640	.655 v	1.30
VBE3	.6740	.7449	.674 v	10.50
VBE4	.6520	.6571	.649 v	1.17
VCE1	10.85	10.85	10.82 v	.28
VCE2	16.71	16.71	16.70 v	.06
VCE3	11.36	11.36	11.36 v	0.00
VCE4	17.37	17.37	17.36 v	.06

The errors are, however, worse for the base currents due to the fact that the base currents are so small. Considering the transistor Q_3, the base-emitter voltage is positive; therefore, the base current cannot be negative or the transistor is not faulty. This is thus a case of calculation error.

Case II. A Bad Linear Component in the Circuit

The resistor R_2 is increased from 3.3K to 8.9K. A set of measurements is taken from the test points, and used to compute the transistors' operating conditions with the faulty value of R_2 taken or not taken into account. The results are tabulated in Table 6. The significance is that the transistor Q_2 appears to be faulty if the faulty value of R_2 is not taken into account. The base current, which is equal to the collector current in magnitude, suggests that the base and the collector of Q_2 are shorted together while the emitter is open. On the other hand, if the faulty value of R_2 is used for computation of the base and collector current of Q_2, these currents are almost zero (Ø), indicating that the transistor Q_2 is operating at the origin of the characteristic curve.

Case III. A Bad Transistor in the Circuit

Transistor Q_4 is replaced by a known faulty transistor, whose emitter is open-circuited and base to collector junction is almost short-circuited. This is a common type of failure when the emitter to base junction is forward-biased too much. The base-collector voltage-current characteristics of the above transistor is shown in Figure 8. The transistors' operating points are calculated and tabulated in Table 7.

Referring to the calculated operating points based on the measured resistor values of Table 7, the value of IB4 equals in magnitude to IC4, and VBE4 is much greater than point six volts (0.6V). These data indicate that the emitter-base junction is short-circuited, therefore, the transistor Q_4 is faulty.

Table 6 Transistor Operating Conditions for Case II

Transistor	Parameter	Faulty value of R_2 is used for operating point calculations.	Faulty value of R_2 not taken into account.
Q1	IB1	1.355	1.355 mA
	IC1	3.183	3.183 mA
	VBE1	.729	.729 v
	VCE1	.020	.020 v
Q2	IB2	.001	5.464 mA
	IC2	-.001	-5.464 mA
	VBE2	-2.32	-2.32 v
	VCE2	26.20	26.20 v
Q3	IB3	0.0	0.0 mA
	IC3	0.0	0.0 mA
	VBE3	0.0	-0.0 v
	VCE3	26.20	26.20 v
Q4	IB4	0.0	0.0 mA
	IC4	0.0	0.0 mA
	VBE4	0.0	0.0 v
	VCE4	26.20	26.20 v

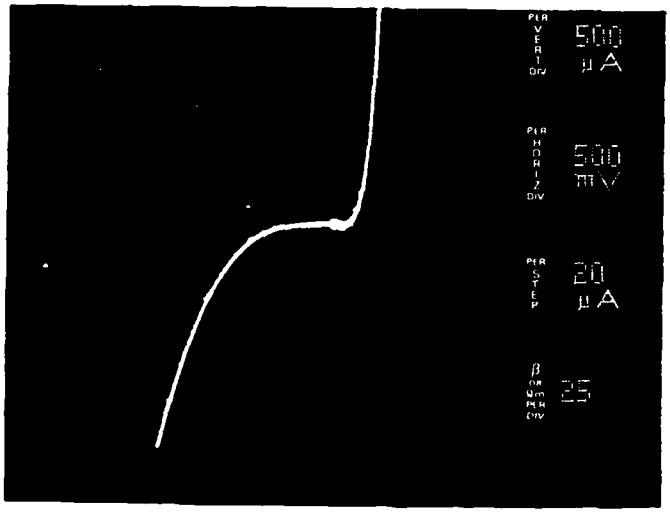


Fig. 8 Collector-Base I-V Characteristics of the Faulty Transistor Used in Case III and IV of the Experiment

Table 7 Calculated Operating Points and Fault-Diagnosis for Case III

Q	Parameter	Calculated value based on measured resistance.	Diagnosis	Calculated value based on claimed resistance	Diagnosis
Q1	IB1	22.12	Good	21.92 uA	Good
	IC1	4.81	β = 217	4.86 mA	β = 222
	VBE1	.6540		.654 v	
	VCE1	11.01		11.01 v	
Q2	IB2	2.15	At rest	-47.56uA	VBE>.5v ,
	IC2	73.47	IB $\simeq$ 0	-517.4 uA	IB,IC must
	VBE2	-4.71	IC $\simeq$ 0	-4.71 v	be zero :
	VCE2	11.16		11.16 v	Calculation error
Q3	IB3	71.80	Good	-280.3 uA	VBE >.6v
	IC3	9.73	β = 136	9.83 mA	IB must> 0
	VBE3	.6970		.813 v	:Cal.error
	VCE3	2.05		2.05 v	
Q4	IB4	-12.82	Bad	-12.82 mA	Bad
	IC4	12.82	IB=-IC	12.82 mA	IB= -IC
	VBE4	19.21	VBE>>.6v	19.27 v	VBE>>.6v
	VCE4	26.20	:B-E opened C-B short.	26.20 v	

Again, using the nominal values of the resistors for calculation produces significant errors. Consider the transistor Q_2, the base to emitter voltage is negative, therefore, the transistor is operating in the cut-off region or both the base and the collector currents have to be close to zero (Ø). However, the calculations show a relatively large negative value for both of these currents. The error also occurs in the base current calculation for transistor Q_3, which causes an ambiguous state for the above transistor.

Case IV. Transistors Q_3 and Q_4 are Faulty

Transistors Q_3 and Q_4 are replaced by known faulty transistors, whose base-collector characteristics are shown in Figure 8. The transistors' operating points are calculated and tabulated in Table 8. The calculation of operating points based on the measured resistor values is accurate, while those based on the nominal resistor values creates errors.

Table 8 The Calculated Operating Points and Fault-Diagnosis for Case IV

Q	Parameter	Calculated value based on measured resistor values	Diagnosis	Calculated value based on claimed resistor values	Diagnosis
Q1	IB1	22.12	Good	21.92 uA	Good
	IC1	4.81	β = 217	4.86 mA	β = 222
	VBE1	.654		.654 v	
	VCE1	11.02		11.02 v	
Q2	IB2	-1.19	At rest	-50.9 uA	At rest
	IC2	74.22	IB,IC $\simeq$ 0	-244.3uA	IB,IC $\simeq$ 0
	VBE2	-6.10	VBE <0	-6.1 v	VBE <0
	VCE2	9.76		9.76 v	
Q3	IB3	-5.28	Bad	-5.34 mA	Bad
	IC3	5.28	IB= -IC	5.34 mA	IB=-IC
	VBE3	18.11	VBE>>.6v	18.20 v	VBE>>.6v
	VCE3	20.86		20.86 v	
Q4	IB4	-8.66	Bad	-8.66 mA	Bad
	IC4	8.66	same as	8.66 mA	same as
	VBE4	19.25	Q3	19.30 v	Q3
	VCE4	26.20		26.20 v	

CHAPTER VII

CONCLUSION

The dual-mode fault-diagnosis technique for non-linear systems has been formulated in the preceding sections via the component-connection equations. Mode 2 or non-linear D.C. analysis is performed at the device or element level rather than at the internal parameter level of the corresponding non-linear devices, which is not only compatible to previous successful fault-diagnosis techniques in digital systems, but also very practical in today's increasingly complex electronic systems.

Computation-wise, this technique is much more advantageous than the other existing techniques in analog non-linear systems because of the use of linear matrix manipulations rather than solving complex non-linear equations. However, this technique requires a relatively larger number of test points compared to the method of V. Visvanathan and A. Sangiovani-Vincentelli [2].

The experimental results indicate that the D.C. or non-linear fault-diagnosis techniques cannot be used to diagnose the tolerance or soft-fault problems due to the significant errors introduced by the use of the manufacturer's claimed component values. However, these errors can be eliminated by the dual-mode technique since the measured values of the linear components can be computed in Mode 1 or the multi-frequency test. Furthermore, for the Mode 2 analysis, the bad value of a faulty linear component has to be calculated so that the more accurate results and fault-diagnosis can be obtained.

REFERENCES

[1] N. Sen and R. Saeks, "Fault Diagnosis for Linear Systems via Multi-Frequency Measurements," IEEE Transaction on Circuits and Systems, Vol. Cas-26, No. 7, pp. 457-465, July 1979.

[2] V. Visvanathan and A. Sangiovanni-Vincentelli, "Fault-Diagnosis of Non-Linear Memoryless Systems," unpublished note, University of California at Berkeley.

[3] N. David and A. N. Wilson, Jr., "A Theory and an Algorithm for Analog Circuit Fault-Diagnosis," IEEE Transaction on Circuits and Systems, Vol. Cas-26, No. 7, pp. 440-456, 1979.

[4] M. N. Ransom and R. Saeks, "The Connection Function--Theory and Application," Int. J. Circuit Theory and Its Applications, Vol. 3, pp. 5-21, 1975.

[5] F. Wang and H. H. Schreiber, "A Pragmatic Approach to Automatic Test Generation and Failure Isolation of Analog Systems," IEEE Transactions on Circuits and Systems, Vol. Cas-26, No. 7, pp. 584-585, July 1979.

[6] P. Duhamel and J. C. Rault, "Automatic Test Generation Techniques for Analog Circuits and Systems: A Review," IEEE Transaction on Circuits and Systems, Vol. Cas-26, No. 7, pp. 411-439, July 1979.

[7] W. J. Dejka, "Measure of Testability in Device and System Design," Proceedings 20th Midwest Symposium on Circuits and Systems, Lubbock, Texas.

[8] W. J. Dejka, "A Review of Measurements of Testability for Analog Systems," Proc. 1977 QUTOTESTCON, Hyannis, Massachusetts, pp. 279-284, 1977.

[9] N. Sen, M. S. Thesis, Texas Tech University, Lubbock, Texas

[10] R. Saeks, R. DeCarlo, and S. Sangani, "Fault Isolation via Affinization," Fault Analysis in Electronic Circuits and System II, Texas Tech University Publication, pp. 109-114, January 1978.

[11] Chua L., "A Glimpse at Some Physical Models," in Computer Circuit Models of Electronic Devices and Components, pp. 76-77, 1977.

[12] R. Saeks and N. Sen, "Formulation of the Faulty-Diagnosis Equations," Fault-Analysis in Electronic System II, Texas Tech University Publication, pp. 1-23, January 1978.

[13] H. S. M. Chen and R. Saeks, "A Search Algorithm for the Solution of the Multi-Frequency Fault-Diagnosis Equations," IEEE Transaction on Circuits and Systems, Vol. Cas-26, No. 7, pp. 589-594, July 1979.

APPENDIX

TEST POINT SELECTIONS

Analysis of the Singularity of the Matrix

$$F = \left(M_{21}^{L}(Z_L^{-1}-M_{11}^{LL})^{-1}M_{11}^{LN} + M_{21}^{N} \right)$$

This analysis is intended to expose the readers to the problem of the test point selection in the D.C. analysis of the dual-mode fault-diagnosis technique. The matrix F can be arranged in the following form:

$$F = \left(M_{21}^{L}(Z_L^{-1}-M_{11}^{LL})^{-1}M_{11}^{LN} + M_{21}^{N} \right)$$

$$= \left[M_{21}^{L} \vdots M_{21}^{N} \right] \begin{bmatrix} (Z_L^{-1} - M_{11}^{LL})^{-1} M_{11}^{LN} \\ \hline I \end{bmatrix}$$

$$= M_{21} \begin{bmatrix} K \\ \hline I \end{bmatrix}$$

$$= M_{21} \cdot K^{1}$$

where: $K = (Z_L^{-1} - M_{11}^{LL})^{-1} M_{11}^{LN}$

I = identity matrix

The matrix M_{21} depends totally on output selections; therefore, the singularity of matrix F is also determined by the test point or output selections.

Referring to Example 2 of Section III, the matrix K^1 is:

$$
\begin{bmatrix}
\dfrac{R_2+R_3}{R} & \dfrac{R_3}{R} & 0 & 0 \\[2ex]
\dfrac{R_E}{R_E+R_C} & \dfrac{-R_E}{R_E+R_C} & \dfrac{1}{R_E+R_C} & \dfrac{1}{R_E+R_C} \\[2ex]
\dfrac{-R_1R_2}{R} & \dfrac{R_2R_3}{R} & 0 & 0 \\[2ex]
\dfrac{-R_1R_3}{R} & \dfrac{R_3(R_1+R_2)}{R} & 0 & 0 \\[2ex]
\dfrac{R_ER_C}{R_E+R_C} & \dfrac{R_ER_C}{R_E+R_C} & \dfrac{-R_E}{R_E+R_C} & \dfrac{-R_E}{R_E+R_C} \\
\hline
1 & & & \\
 & 1 & & \\
 & & 1 & \\
 & & & 1
\end{bmatrix}
$$

If V_M^1, IR1, IRC, VR3 are chosen as test measurements, where $V_M^1 = V_M + VCE\,2$ the matrix is:

$$M_{21} = \left[\begin{array}{ccccc:cccc} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

The resulting matrix F is singular because row 1 and 3 are linear dependent.

$$F = \begin{bmatrix} \frac{R_E R_C}{R_E + R_C} & \frac{R_E R_C}{R_E + R_C} & \frac{R_C}{R_E + R_G} & \frac{R_C}{R_E + R_C} \\ \frac{R_2 + R_3}{R} & \frac{R_3}{R} & 0 & 0 \\ \frac{-R_E}{R_C + R_E} & \frac{-R_E}{R_C + R_E} & \frac{-1}{R_C + R_E} & \frac{-1}{R_C + R_E} \\ \frac{-R_1 R_3}{R} & \frac{-R_3(R_1 + R_2)}{R} & 0 & 0 \end{bmatrix}$$

However, with a minor change in output selection, V_M, IR1, IRC, VR3 are chosen, the first row of M_{21} becomes:

$$\left[\begin{array}{cccc|cccc} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right] ,$$

and the first row of matrix F becomes:

$$\left[\begin{array}{cccc} \dfrac{R_E R_C}{R_E + R_C} & \dfrac{R_E R_C}{R_E + R_C} & \dfrac{-R_E}{R_E + R_C} & \dfrac{R_C}{R_E + R_C} \end{array}\right]$$

Therefore, F is non-singular.

Referring to the Example of Section III, the matrix K^1 is:

$$\left[\begin{array}{cccc} 1/R1 & 0 & 0 & 0 \\ 1/R2 & 0 & 0 & 0 \\ 0 & R3 & R3 & 0 \\ 0 & 0 & 0 & R4 \\ 0 & R5 & 0 & R5 \\ \hline 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{array}\right]$$

If V_M, I+, VB1, VB2 are chosen as test points, the matrices M_{21} and F are:

$$M_{21} = \left[\begin{array}{ccccc|cccc} 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$F = \begin{bmatrix} 0 & 0 & 0 & -R_4 \\ -1/R1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & -R3 & -R3 & 0 \end{bmatrix}$$

The F matrix is singular because the columns 2 and 3 are linearly dependent.

However, if V_M, I+, VR5, VB2 are chosen, the matrix F is non-singular:

$$M_{21} = \left[\begin{array}{ccccc|cccc} 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$F = \begin{bmatrix} -1/R1 & 0 & 0 & 0 \\ -1/R1 & 1 & 1 & 1 \\ 0 & R5 & 0 & R5 \\ 0 & -R3 & -R3 & 0 \end{bmatrix}$$

Considering the second and third column of matrix K^1, the differences between these two columns are the elements $K^1_{52} = R_5$, $K^1_{53} = 0$ and $K^1_{72} = 1$, $K^1_{83} = 1$. A subset of M_{21}, which consists of two outputs, has to be chosen, in such a way that its product with the second and third column of K^1 is non-singular. This condition is satisfied when one of the outputs contains VR5, as in the third row of M_{21} of the preceding discussion, which results a non-zero element at row 3 and column 2 of matrix F.

Referring to Example 4 of Section III, the matrix K^1 is:

$$
\left[\begin{array}{cccccccc}
R1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
R3 & R3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & R2 & R2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & R4 & R4 & -R4 & 0 & -R4 & 0 \\
0 & 0 & 0 & 0 & -R5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -R6 & 0 \\
0 & 0 & 0 & 0 & 0 & R7 & 0 & 0 \\
0 & 0 & 0 & 0 & R8 & R8 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & R9 & R9
\end{array}\right]
$$

$$
\left[\begin{array}{cccccccc}
1 & & & & & & & \\
 & 1 & & & & \oslash & & \\
 & & 1 & & & & & \\
 & & & 1 & & & & \\
 & \oslash & & & 1 & & & \\
 & & & & & 1 & & \\
 & & & & & & 1 & \\
 & & & & & & & 1
\end{array}\right]
$$

Suppose VR2, VR3, VR7, VR9, VR8 and VR4 are chosen to be the test points, the matrices M_{21} and F are:

$$F = M_{12}K^{1} = \begin{bmatrix} 0 & R2 & R2 & 0 & 0 & 0 & 0 & 0 \\ R3 & R3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & R7 & 0 & 0 \\ R1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & R9 & R9 \\ 0 & 0 & 0 & 0 & R8 & R8 & 0 & 0 \\ 0 & 0 & R4 & R4 & -R4 & 0 & -R4 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

The determinant of F is:

$$\det(F) = -R_1R_3R_2R_7R_8 . \det \begin{bmatrix} 0 & R_9 & R_9 \\ R_4 & -R_4 & 0 \\ 1 & 0 & 1 \end{bmatrix} = 0$$

because

$$\det \begin{bmatrix} 0 & R9 & R9 \\ R4 & -R4 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \det \begin{bmatrix} 0 & R9 \\ R4 & -R4 \end{bmatrix} + \det \begin{bmatrix} R9 & R9 \\ -R4 & 0 \end{bmatrix} =$$

$$- R_4R_9 + R_4R_9 = 0$$

In order to make $\det \begin{bmatrix} 0 & R9 & R9 \\ R4 & -R4 & 0 \\ 1 & 0 & 1 \end{bmatrix} \neq 0$, one of the elements of

the third row of this matrix has to be changed to zero as follows:

$$\begin{bmatrix} 0 & R9 & R9 \\ R4 & -R4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 0 & R9 & R9 \\ R4 & -R4 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Physically, this means one has to select the last output measurement such that it does not contain both IC4 and IC3.

END

DATE
FILMED

7-81

DTIC