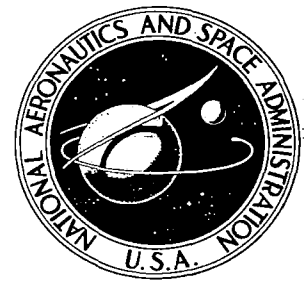
NASA CONTRACTOR REPORT



NASA CR-1758

FAULT ISOLATION COMPUTER METHODS

by Ricardo F. Garzia

Prepared by
COMPUTER SCIENCES CORPORATION
Huntsville, Ala.
for George C. Marshall Space Flight Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • FEBRUARY 1971

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
NASA CR-1758		

4. TITLE AND SUBTITLE	5. REPORT DATE
FAULT ISOLATION COMPUTER METHODS	February 1971
	6. PERFORMING ORGANIZATION CODE

7. AUTHOR(S)	8. PERFORMING ORGANIZATION REPORT #
Ricardo F. Garzia	Technical Report No. 1

9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. WORK UNIT NO.
Computer Sciences Corporation 8300 Whitesburg Dr. Huntsville, Alabama	11. CONTRACT OR GRANT NO. NAS 8-18405
	13. TYPE OF REPORT & PERIOD COVERED Contractor Report

12. SPONSORING AGENCY NAME AND ADDRESS	14. SPONSORING AGENCY CODE
NASA-George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812	

15. SUPPLEMENTARY NOTES

Work performed for Computation Laboratory, Science and Engineering Directorate

16. ABSTRACT

Presented in this report are ten computer methods for fault isolation, three of them employing two different techniques. In some cases computer applications are included, with some comments on the results of them. It is felt that some of the methods could be modified and improved to make them into practical tools. Others which assume ideal situations are impractical in a real environment.

17. KEY WORDS	18. DISTRIBUTION STATEMENT
Methods Application Requirements	Unclassified - Unlimited

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE *
U	U	158	$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

Acknowledgement

Recognition is given to Dr. Heinz Trauboth for his contribution, encouragement, suggestions, and advice during the work. This recognition also includes his contribution toward the text material, specifically the Introduction and Conclusions.

CONTENTS

CONTENTS (Continued)

CONTENTS (Concluded)

CHAPTER I

1.1 Introduction -

In the hearings of the Congressional Space Committee last year, NASA identified three major technology problem areas for the development of the Space Shuttle. These were

- New materials for ablative heat-shielding
- New propulsion systems
- New automatic checkout concepts

We who work in computer software technology feel that we can contribute significantly to the solution of this third technology problem area.

The Space Shuttle will have to withstand severe flight conditions not just for one flight but for many versatile missions, while the Space Station has to endure many years in space environment functioning reliability. Although the flight components and systems possess a high degree of reliability, malfunctions have to be expected. The time to refurbish and checkout the shuttle between missions will be very short compared to prelaunch checkout time of the present Saturn V flight system. Moreover, the costs for ground checkout equipment and operations have to be reduced considerably, for new space systems to be feasible, and the on-board checkout hardware has to be limited in weight and volume. These future space vehicles will be highly autonomous so that more checkout and maintenance functions have to be performed on-board the vehicle. The flight crew will be small and can devote only a small portion of their time to flight maintenance. A large portion of the time in checkout and maintenance operations is devoted to the detection and isolation of malfunctions.

Let us take a brief look at some cost relationship.* In Figure 1.1, qualitative curves of cost versus mean time between failure (MTBF) are shown for acquisition, maintenance and logistics. If one adds up these curves, the total cost versus MTBF curve has a minimum. Through employing fast and efficient fault isolation techniques together with less reliable components, the total cost could be shifted closer to the minimum of the total cost versus MTBF curve.

It is our opinion that this area of fault detection and isolation can be improved by greater utilization of computer technology.

---

*Ben S. Blanchard - Concepts of Cost Effectiveness, General Dynamics/ Electronics, Rochester, New York.

1.2

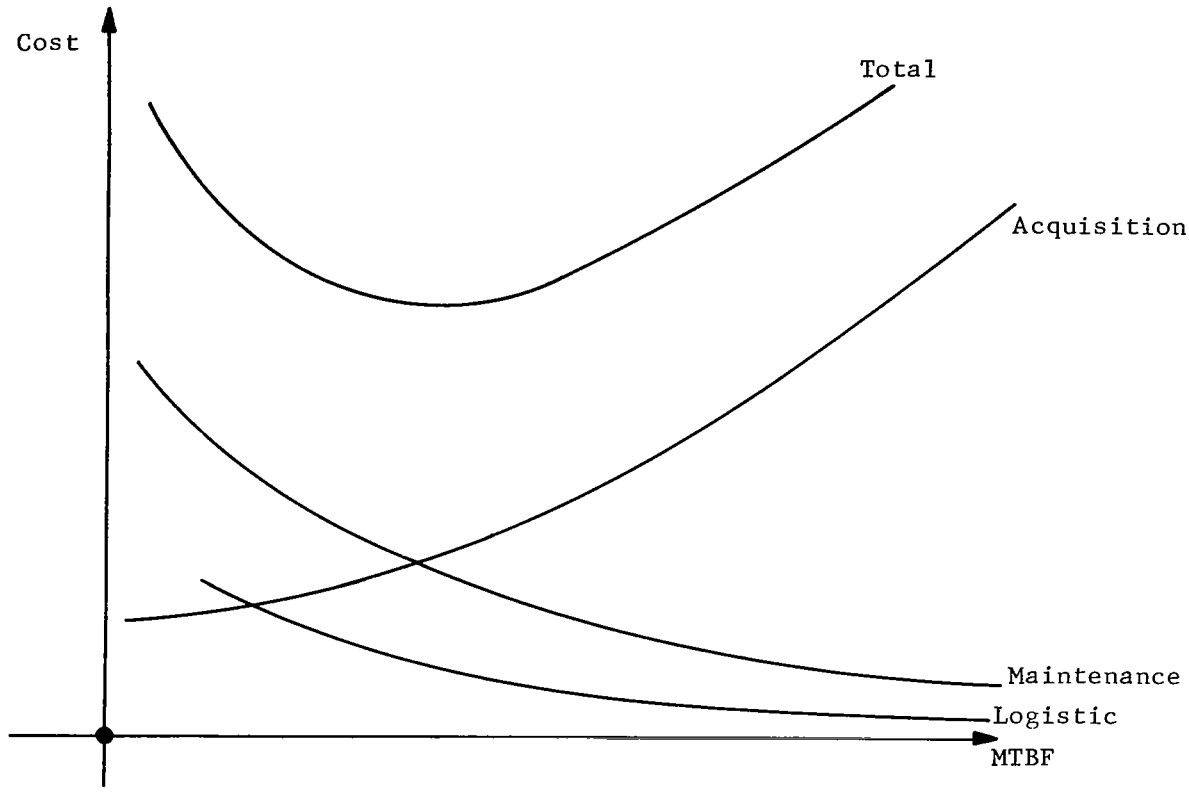


Figure 1.1

During the last five years, enormous progress in microelectronics has been made, which leads to more powerful computers and digital analyzers. Minicomputers with great processing speed, large memory capacities, and graphical input/output are now available for relative low cost. At the same time, great advances in new mathematical techniques in such disciplines as control systems engineering, information theory, and systems theory have been made. New techniques for parameter estimation, digital filtering, and adaptive control have been developed. It is expected that these modern developments could be put to work for improved automatic checkout. Presently in checkout operations, mainly static values of signals are used. By utilizing the information that is often available in dynamic and statistical signals at the outputs of a system, more information about the health of its units inside the system could be derived. For instance, instead of monitoring several components of a subsystem, the transfer function response could be measured and the coefficients of the transfer function which relate to component values could be computed. Thereby, the number of test points and the amount of instrumentation and data acquisition hardware might be reduced. This means that the computer

will be burdened with more complex analysis computations; in other words, by reducing the hardware requirements, the software requirements will be increased. Though these new techniques look very promising, more research in this area is needed to derive mathematical and computational techniques which are efficient.

The Computation Laboratory has developed for years systems and applications software for automatic checkout and testing in the Quality and Reliability Assurance Laboratory and the Astronautics Laboratory. It has developed under contract with General Dynamics the Automatic Malfunction Analysis Program (AMA) which isolates faults in discrete networks. We will now concentrate on the development of fault isolation methods for continuous systems which to a high degree requires investigations into applied mathematics and computational methods. It is not our intent to develop any hardware for checkout systems, rather we want to concentrate on the development of principal software techniques which are independent of a specific hardware design. Then, we will make recommendations to the hardware designers so they can adjust their design, if necessary, for better utilizing these software techniques.

In our research work in computer methods for fault isolation, we intend to develop a few software methods which are promising for practical applications in testing, ground checkout, and on-board checkout of systems and electronic circuits. We do not expect to develop one general method which can be applied throughout the checkout of a complete vehicle, rather we expect to develop different methods which can be applied for different types of systems.

A few years ago, the Quality and Reliability Assurance Laboratory developed a few techniques for dynamic testing. These techniques were programmed by Computation Laboratory and applied to the checkout of the stabilized platform of the Saturn V Instrumentation Unit.[23] The results showed that the approach seems to be feasible; however, it was not completely satisfactory and more research was needed. About a year ago, we began in-house a thorough literature study of various computerized fault isolation methods, in order to make use of work that has already been done. Out of about 70 papers, ten distinct methods based on different principles were identified and they form the subject matter of this report. They are now being evaluated with actual test cases and are being programmed for the computer. Most of these methods were developed under Government-funded contracts between 1958 and 1965; a great number of them were sponsored by Wright-Patterson Air Force Base. The majority of them were developed for fault isolation within electronic circuits. We think that several of them can also be applied for fault isolation within systems since the mathematical representation of circuits and systems is very similar. In the examples, although simple electrical circuits are used for illustrative purposes, this does not imply that

the method presented is necessarily restricted to these simple circuits. While these methods form a good base for our investigations, we think that improved fault isolation techniques could be developed utilizing latest analytical and numerical methods.

In our research, we plan to concentrate on a few methods which are promising for practical application in automatic checkout and testing. These methods will be tested by running them first against simulated systems and then against a hardware breadboard system. We will then make recommendations for their use in Space Shuttle and Space Station checkout and in checkout and testing operation at MSFC.

Thus, the report represents the first step of our endeavor in the development of computer methods for fault isolation. In preparing this report the first source of information has been the internal note "Preliminary Study of Some Fault Isolation Techniques" by Donald Nalley - IN-COMP-70-2-MSFC.

1.2 General Considerations -

In order to accomplish the objective of Automatic Diagnosis with full use of today's capabilities, the following steps have to be taken.

- Evaluation of the different mathematical methods which have been developed by Government and industry.
- Utilization of modern analysis techniques of linear and nonlinear systems in the diagnosis process.
- Utilization of the capabilities of today's computers more in the analysis of the process under test, rather than in handling big quantity of data. This is going to give us the possibility of performing the analysis of the unit under test prior to the checkout process, thereby decreasing the size of the on-line computer required during the checkout process.
- Correlation between the fault isolation computer methods and the instrumentation required.
- New measurement techniques.
- Better estimation of the measurements.

In evaluating the different mathematical tools that have been used in the solution of the Fault Isolation Problem, it is helpful to define clearly the terms "method" and "technique". See Figure 1.2.

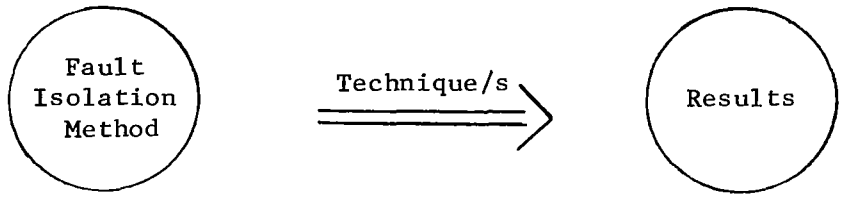


Figure 1.2

Method is the physical approach used in detecting the fault.

Technique is the effective mathematical tool used for the above purpose.

As a result of reviewing about 70 papers and applying the above definitions, we arrive at ten different methods, three of them employing two different techniques.

1.3 Some Comments on the Different Methods Studied -

Before beginning our study, it is convenient to make the following comments.

- All the papers that have been reviewed are indicated in the references.
- The papers that have been used for obtaining a given method, technique or some piece of information are indicated in the references with an asterisk before the name.
- At the beginning of each method, the reference number/s are indicated between brackets.
- For each example given, the program and computer that has been used are indicated in the Appendix.

CHAPTER II

The methods which will be discussed in this chapter are the following:

1. Classification Method

   a) Matrix Recognition Technique
   b) Sequential Recognition Technique

2. Key Element Search Method

   a) Nonlinear Technique
   b) Linear Technique

3. Iterative Method

4. Transfer Function Method

   a) Bode Diagram Technique
   b) Tracking Technique

5. Scalar Remnant Method (Dynamic System Testing)

6. Inverse Probability Method

7. Power Spectra Method

8. Parameter Identification Method

9. Max-Current Method

10. Detection Condition Method (Dynamic System Testing)

The order given in the presentation of the above methods is a good approach in the introduction of the different ideas involved in Fault Isolation Diagnosis. In some cases computer applications are included, with some comments on the results of them.

2.1 Classification Method [25]

2.1.1 General

The idea of this method is to choose a number of test points, from which it is possible to define, in the n-dimension space, regions of normal operation of the system, and regions of failure of the system. In this last case each region could make identification as to what kind of failure has occurred. It is understood that we may go to any level of failure classification that we need, changing the number of test points. This last statement suggests the name given to this method.

The number of test points depends on the

- Detail of failure analysis that is needed,
- Correct selection of the test points.

Suppose that our system looked like the one shown in Figure 2.1. The system under test has three test points.

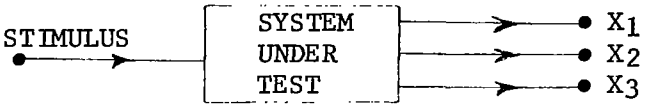


Figure 2.1

Now, we need to define the regions of normal operation, and the regions of failure. Let these regions be

Region 1 (Normal Operation)

$$X_1 < k_{12}$$
$$k_{21} < X_2 < k_{22}$$
$$k_{31} < X_3$$

Region 2 (Normal Operation)

$$X_1 < k_{12}$$
$$X_2 < k_{21}$$
$$X_3 < k_{31}$$

Region 3 (Failure 1)

$$X_1 < k_{12}$$
$$k_{21} < X_2 < k_{22}$$
$$X_3 < k_{31}$$

Region 4 (Failure 2)

$$k_{12} < X_1$$
$$k_{21} < X_2 < k_{22}$$
$$X_3 < k_{31}$$

Region 5 (Failure 3)

$$X_1 < k_{12}$$
$$X_2 < k_{21}$$
$$k_{31} < X_3$$

The above regions list is complete, in other words there is no possibility of getting values of X_1, X_2, and X_3 from our system, which will belong to a region that is not in the given list. This means that the other regions are not allowed for our system. Therefore, we can make the following classification

- Allowed Regions.
- Not Allowed Regions.

For a given system, we are interested in finding all the allowed regions. With this in mind, we take the measures of X_1, X_2, X_3 and look for that region which satisfies all conditions within the region. Thus, it is possible to know the operation region, and therefore, if the system under test is in normal operation, or in what type of failure mode.

In Figures 2.2, 2.3, and 2.4 are shown the normal operation regions, failure regions and not allowed regions, respectively, which belong to the example given. Not all the not allowed regions are shown in Figure 2.4; this is for drawing simplicity.

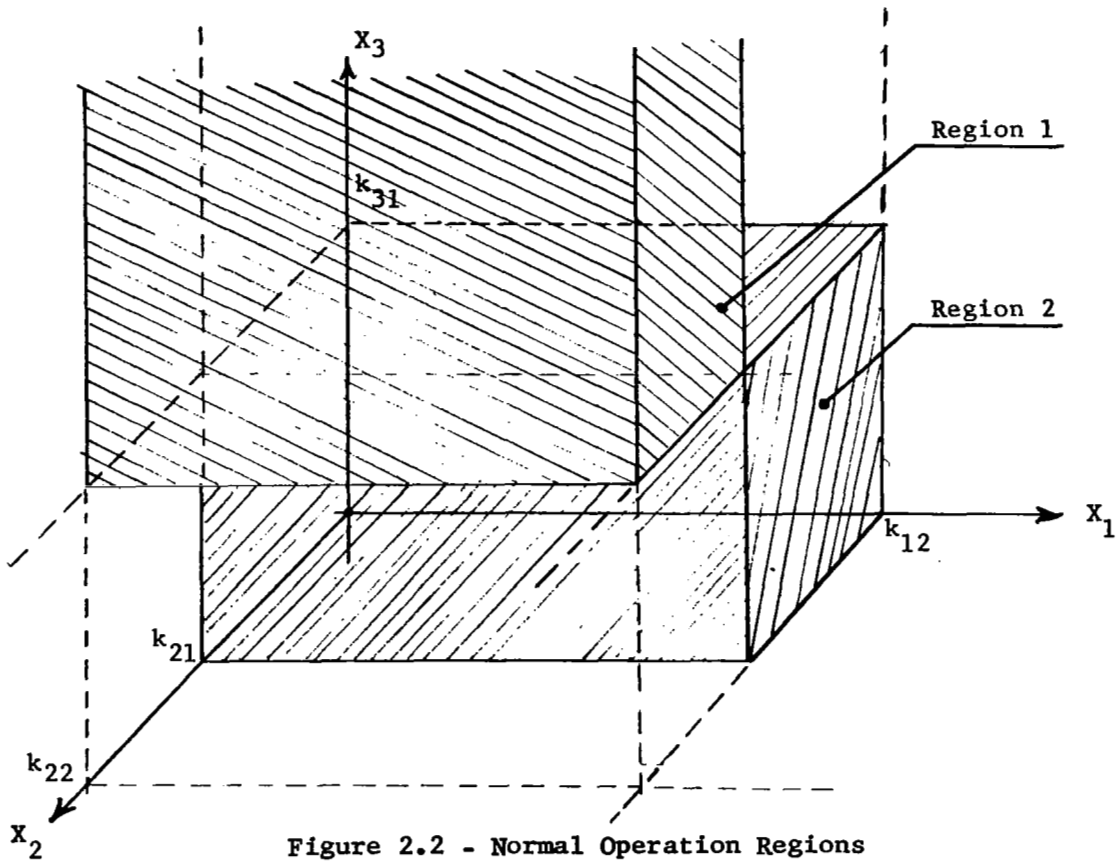


Figure 2.2 - Normal Operation Regions

Region 1	Region 2
$X_1 < k_{12}$	$X_1 < k_{12}$
$k_{21} < X_2 < k_{22}$	$X_2 < k_{21}$
$k_{31} < X_3$	$X_3 < k_{31}$

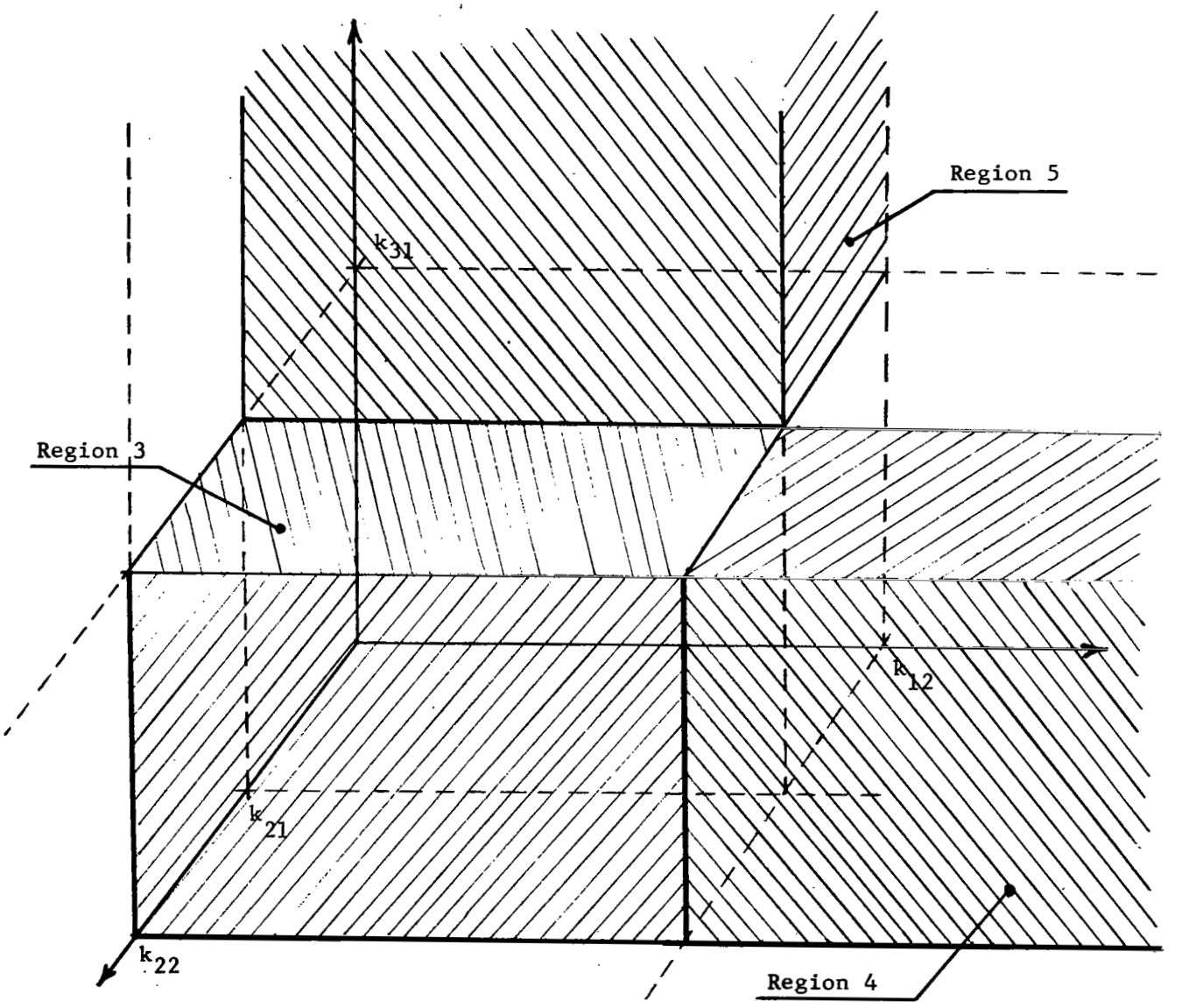


Figure 2.3 - Failure Regions

The technique to carry out the identification of the operation region could be by

a) Matrix Recognition
b) Sequential Recognition

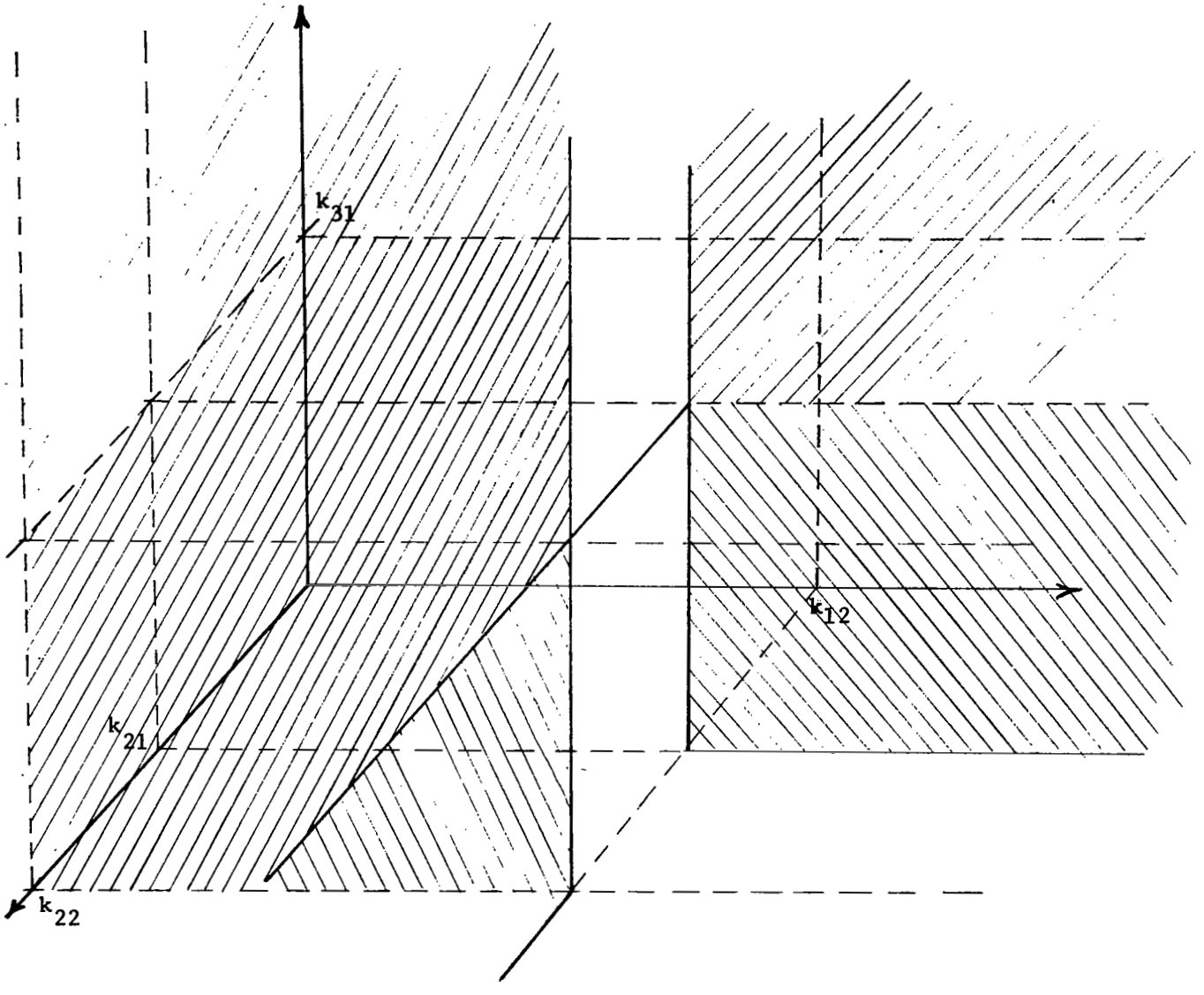


Figure 2.4 - Not Allowed Regions

2.1.2 Matrix Recognition Techniques

Given the matrix equation :

$$Y = XM \tag{2.1}$$

where,

Y = Output vector of the form $(Y_1, Y_2, \ldots Y_n)$

X = Input vector of the form $(X_1, X_2, \ldots . X_m, 1)$
M = Matrix of size (m+1) x n.
n = Number of regions
m = Number of measured variables.

We must form the matrix M in such a way that when the product of the vector X by the matrix M, gives the vector Y, the highest component of this last vector gives us the number of the region, i.e., if the highest component is the third, the region number three is in operation. Therefore, we are not looking for the value of the highest component instead, we are looking for the component in which this happens; being this the number of the region. The name given to the matrix M is the recognition matrix of the given circuit.

The way of getting the matrix M is through the product of the matrix H by the matrix K. Of course, this is not the only way, see for instance the method given in reference [25].

$$M = H.K \tag{2.2}$$

The matrix H is formed with the negative of the boundary values in the last row. First, all the different boundary values that belong to X_1, then all the different boundary values that belong to X_2, and so on. Elsewhere it has zeroes, except that it has 1's in the first row, but only in the columns that belong to X_1, 1's in the second row but only in the column that belongs to X_2, and so on. Therefore, this matrix looks like

$$H = \begin{bmatrix} 1 & 1 \ldots 1 & 0 & 0 \ldots 0 \ldots & 0 & 0 \ldots 0 \\ 0 & 0 \ldots 0 & 1 & 1 \ldots 1 \ldots & 0 & 0 \ldots 0 \\ 0 & 0 \ldots 0 & 0 & 0 \ldots 0 \ldots & 0 & 0 \ldots 0 \\ \cdot & \cdot \ldots \cdot & \cdot & \cdot \ldots \cdot \ldots & \cdot & \cdot \ldots \cdot \\ 0 & 0 \ldots 0 & 0 & 0 \ldots 0 \ldots & 1 & 1 \ldots 1 \\ -A_1 & -A_2 \ldots -A_{n_1} & -B_1 & -B_2 \ldots -B_{n_2} \ldots & -Z_1 & -Z_2 \ldots -Z_{n_m} \end{bmatrix} \tag{2.3}$$

The size of this matrix is given by [number of measured variables + 1] x [summation of (number of measured variables x different boundary values in each one)] .

The number of rows in the matrix K are the same as the number of columns in the matrix H. The number of columns in matrix K is equal to the number of regions that we have in our system (each column belongs to one of the regions). All the columns have the value 1 in each row. The sign associated with it obeys the following rule: "The components of the vector $(X\text{-}A_1, X\text{-}A_2, \ldots, X\text{-}A_{n_1}, X\text{-}B1, X\text{-}B_2, \ldots, X\text{-}B_{n_2}, \ldots \ldots \ldots,$

$X\text{-}Z_1$, $X\text{-}Z_2$,..., $X\text{-}Z_{n_m}$) are tested in order. If, for an operating point in the region, the component is positive, a '+' is entered. If it is negative, a '-' is entered". In order to illustrate the concept, let us take the following example

Region 1 (Normal Operation)	$10 < X_1 < 30$ $5 < X_2 < 40$ $X_3 < 10$
Region 2 (Failure Mode #1)	$10 < X_1 < 30$ $X_2 < 5$ $X_3 < 10$
Region 3 (Failure Mode #2)	$10 < X_1 < 30$ $40 < X_2$ $X_3 < 10$
Region 4 (Failure Mode #3)	$30 < X_1$ $X_2 < 5$ $10 < X_3$

The matrices H and K are given, respectively,

$$H = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -10 & -30 & -5 & -40 & -10 \end{bmatrix} \quad ; K = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix}$$

And the recognition matrix M=HK, are given by

$$M = \begin{bmatrix} 0 & 0 & 0 & 2 \\ 0 & -2 & 2 & -2 \\ -1 & -1 & -1 & 1 \\ 65 & 75 & -15 & -5 \end{bmatrix}$$

But, since Y=XM, and $X=(X_1,X_2,X_3,1)$, we get

$$Y = (65-X_3,\ 75-2X_2-X_3,\ 2X_2-X_3-15,\ 2X_1-2X_2+X_3-5)$$

If we assume that our measured values are given by

$X_1 = 20$
$X_2 = 50$
$X_3 = 5$

the output vector results

$Y = (60, -30, 80, -60)$

Therefore, our operation occurs in the region 3. It is very easy to see that this result is correct. The steps that are necessary to follow in order to get the output vector using a computer, are indicated in Figures 2.5, 2.6, and 2.7. The symbols used in the mentioned figures are indicated as follows:

N = number of regions
M = number of measured variables
JMAX = The sum of all the different bound used in each region
H(I,J) = Element of row I, and column J belonging to matrix H.
K(I,J) = Element of row I, and column J belonging to matrix K.
M(I,J) = Element of row I, and column J belonging to matrix M.
NUM(J) = Number of different bounds in region J
IVAL(J) =Bounds
X(J) = Element J of input vector
O(J) = Element J of output vector.

2.1.2.1 Computer Application

We are going to consider two different applications.

2.1.2.2 1st Application Let us define the following problem

Region 1

$10 < X_1 < 73$
$X_2 < 43$
$4 < X_3 < 18$
$35 < X_4 < 76$

Region 2

$10 < X_1 < 46$
$43 < X_2$
$4 < X_3 < 18$
$35 < X_4 < 76$

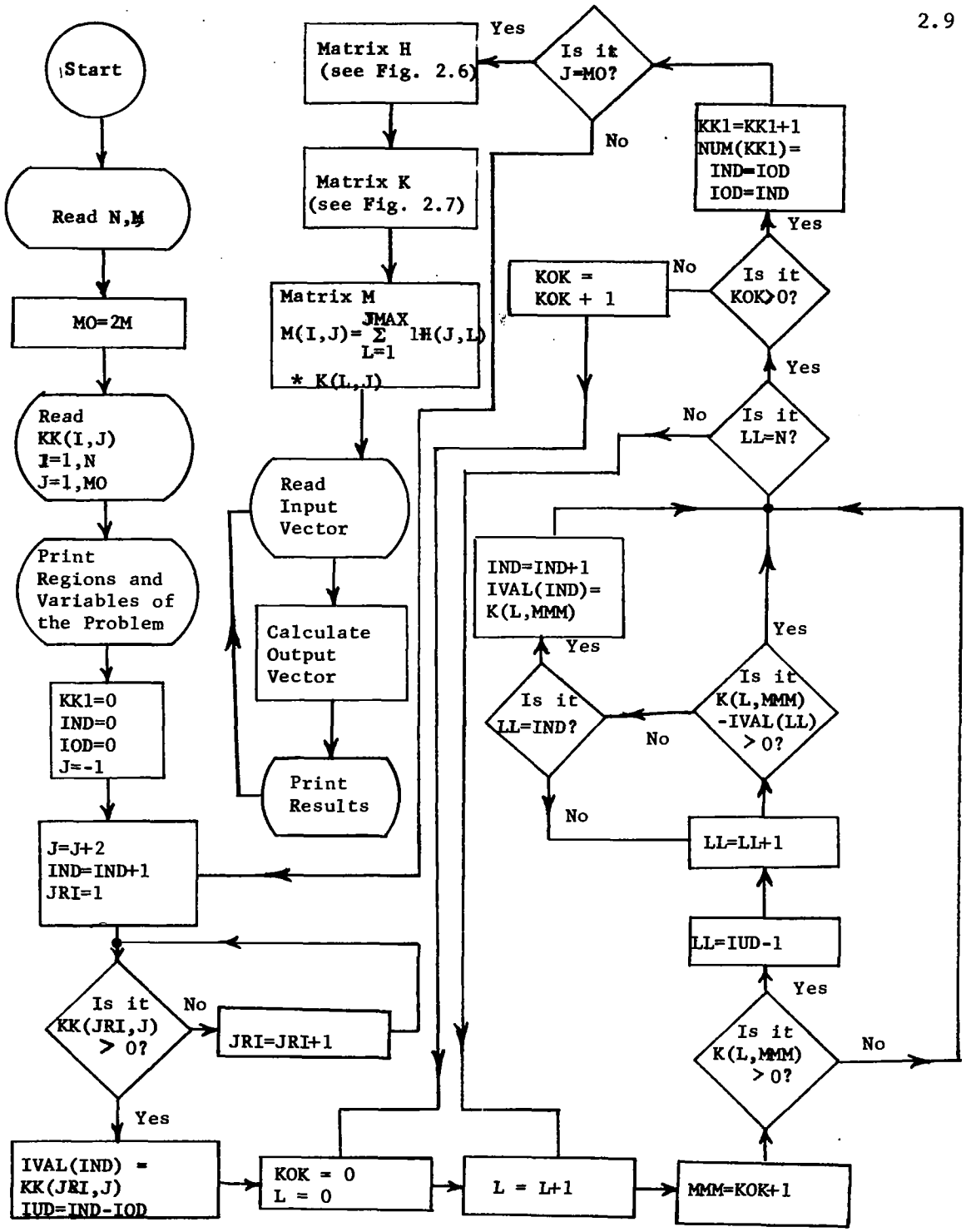


Figure 2.5 - Matrix Recognition Technique
Classification Method

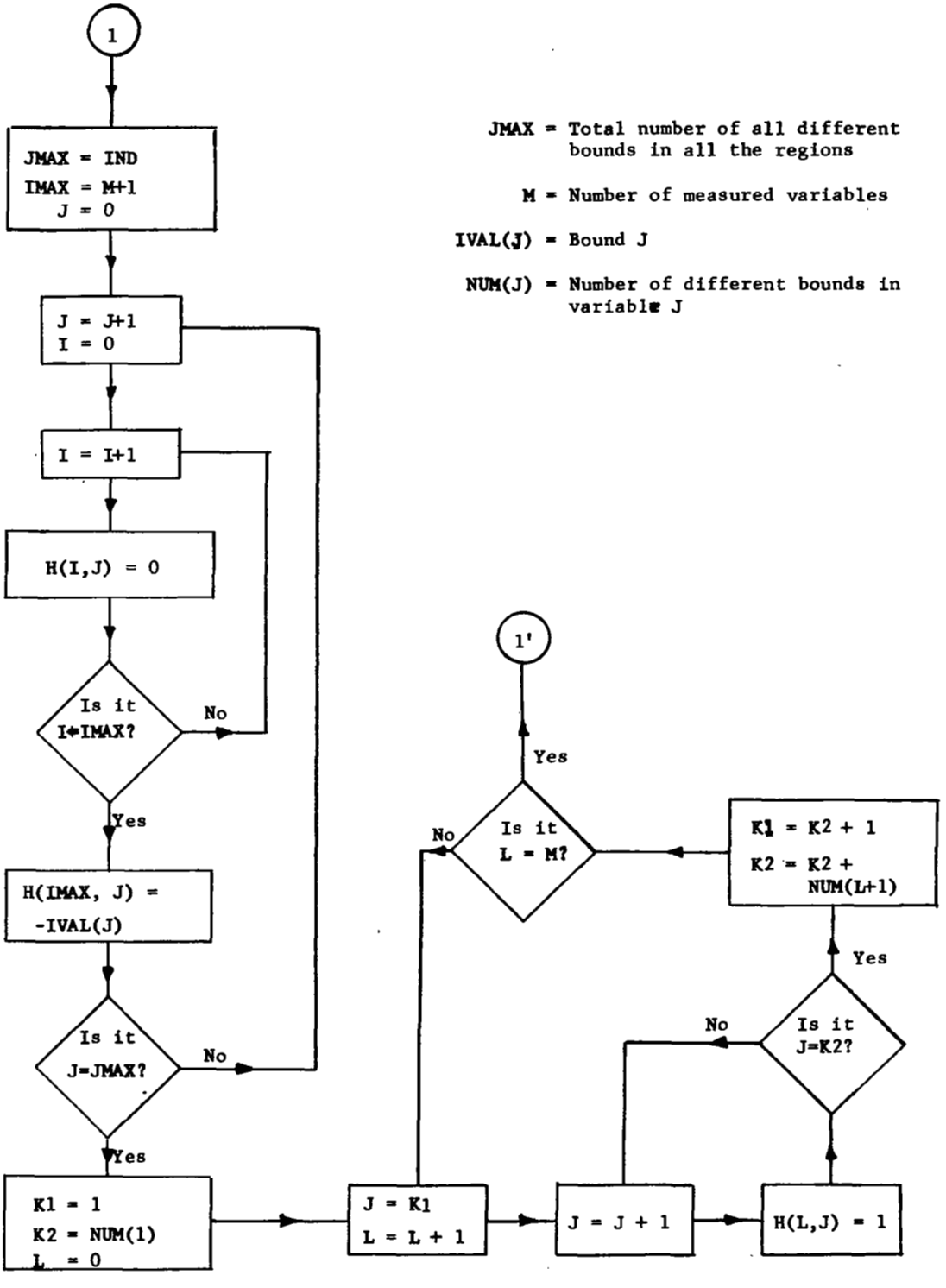


Figure 2.6 Matrix H - Matrix Recognition Technique Classification Method

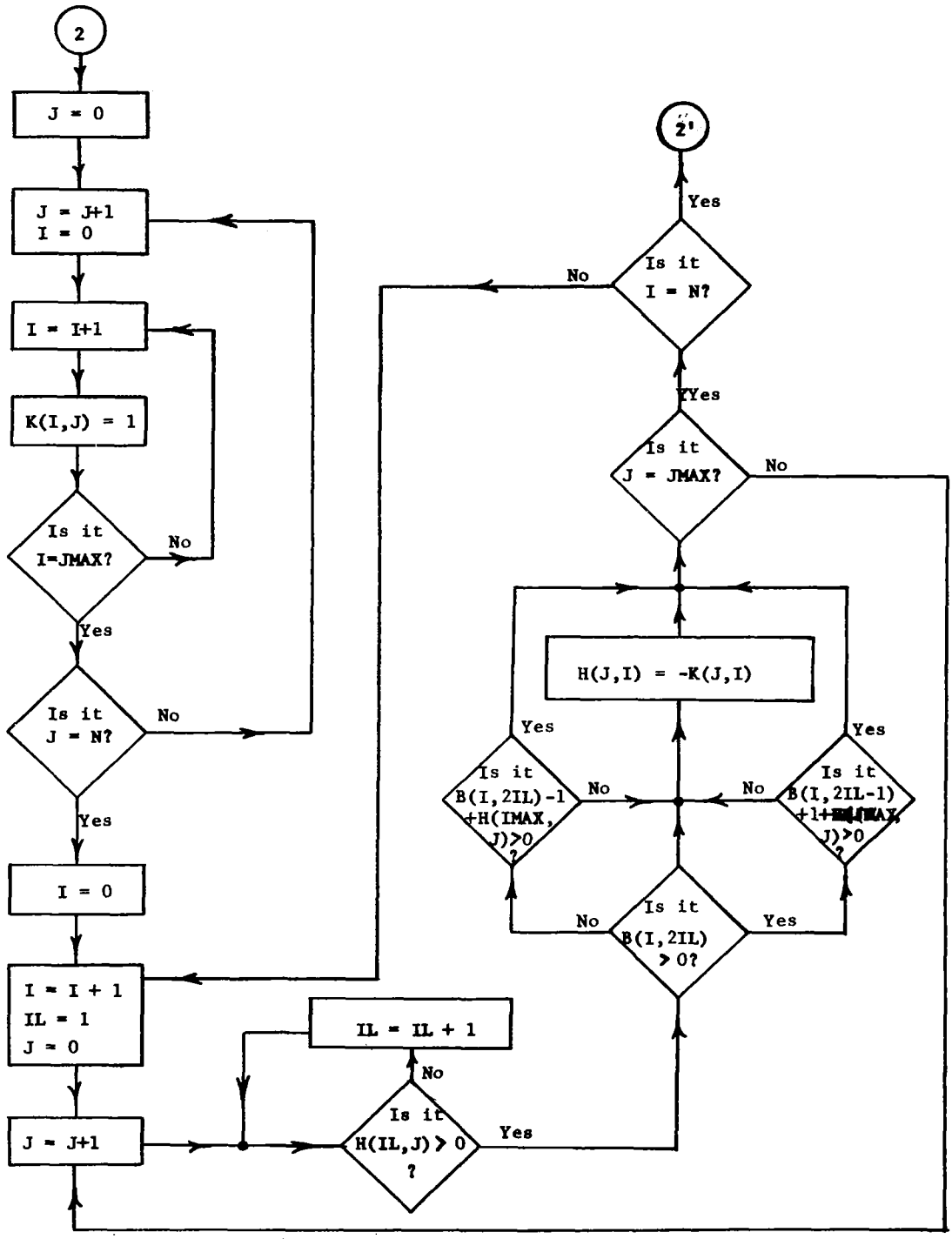


Figure 7.7 Matrix K - Matrix Recognition Technique Classification Method

Region 3

$$46 < X_1 < 73$$
$$43 < X_2$$
$$4 < X_3 < 18$$
$$35 < X_4 < 76$$

Region 4

$$10 < X_1 < 73$$
$$X_2 < 43$$
$$18 < X_3$$
$$35 < X_4 < 76$$

Region 5

$$10 < X_1 < 73$$
$$X_2 < 43$$
$$X_3 < 4$$
$$35 < X_4 < 76$$

Region 6

$$10 < X_1 < 73$$
$$X_2 < 43$$
$$4 < X_3 < 18$$
$$76 < X_4$$

Region 7

$$10 < X_1 < 73$$
$$X_2 < 43$$
$$4 < X_3 < 18$$
$$X_4 < 35$$

Region 8

$$X_1 < 10$$
$$X_2 < 43$$
$$4 < X_3 < 18$$
$$35 < X_4 < 76$$

The results obtained are as follows:

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ -10 & -46 & -73 & -43 & -4 & -18 & -35 & -76 \end{bmatrix}$$

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 \\ -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 \end{bmatrix}$$

$$M = \begin{bmatrix} 1 & -1 & 1 & 1 & 1 & 1 & 1 & -3 \\ -1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 2 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & -2 & 0 \\ 115 & 121 & 29 & 79 & 123 & -37 & 185 & 227 \end{bmatrix}$$

Output

```
 68   26   17   85
157   79  123  155   131   175   57   -3
Region 6

 72   37   22   46
150   86  138  158   114    90   128   -26
Region 4

 11    6   17   37
120  116   46  118    94    42   116   188
Region 8
```

It is very easy to see that the first two results are correct, but the third one is not. The reason for this is the following.

Let us take only the variable X_1, in the first three regions

Region 1	$10 < X_1 < 73$; the other variables
Region 2	$10 < X_1 < 46$; the other variables
Region 3	$46 < X_1 < 73$; the other variables

This is not allowed with the matrix recognition technique. In this particular problem we must re-define the regions. We have to break region one into two regions, one that goes from 10 to 46 and the other from 46 to 73. Therefore we come out with four regions,

Region 1	$10 < X_1 < 46$; the other variables
Region 1'	$46 < X_1 < 73$; the other variables
Region 2	$10 < X_1 < 46$; the other variables
Region 3	$46 < X_1 < 73$; the other variables.

Taking this in consideration, the eight given regions are going to be transformed into thirteen, which are,

Region 1

$$10 < X_1 < 46$$
$$X_2 < 43$$
$$4 < X_3 < 18$$
$$35 < X_4 < 76$$

Region 2

$$46 < X_1 < 73$$
$$X_2 < 43$$
$$4 < X_3 < 18$$
$$35 < X_4 < 76$$

Region 3

$$10 < X_1 < 46$$
$$43 < X_2$$
$$4 < X_3 < 18$$
$$35 < X_4 < 76$$

Region 4

$$46 < X_1 < 73$$
$$43 < X_2$$
$$4 < X_3 < 18$$
$$35 < X_4 < 76$$

Region 5

$$X_1 < 46$$
$$X_2 < 43$$
$$18 < X_3$$
$$35 < X_4 < 76$$

Region 6

$$46 < X_1 < 73$$
$$X_2 < 43$$
$$18 < X_3$$
$$35 < X_4 < 76$$

Region 7

$$10 < X_1 < 46$$
$$X_2 < 43$$
$$X_3 < 4$$
$$35 < X_4 < 76$$

Region 8

$$46 < X_1 < 73$$
$$X_2 < 43$$
$$X_3 < 4$$
$$35 < X_4 < 76$$

Region 9

$$10 < X_1 < 46$$
$$X_2 < 43$$
$$4 < X_3 < 18$$
$$76 < X_4$$

Region 10

$$46 < X_1 < 73$$
$$X_2 < 43$$
$$4 < X_3 < 18$$
$$X_4 < 35$$

Region 11

$$10 < X_1 < 46$$
$$X_2 < 43$$
$$4 < X_3 < 18$$
$$X_4 < 35$$

Region 12

$$46 < X_1 < 73$$
$$X_2 < 43$$
$$4 < X_3 < 18$$
$$X_4 < 35$$

Region 13

$$X_1 < 10$$
$$X_2 < 43$$
$$4 < X_3 < 18$$
$$35 < X_4 < 76$$

The results obtained are as follows:

$$M = \begin{bmatrix} -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -3 \\ -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 2 & 2 & -2 & -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & -2 & -2 & 0 \\ 207 & 115 & 121 & 29 & 171 & 79 & 215 & 123 & 55 & -37 & 277 & 185 & 227 \end{bmatrix}$$

Output

```
 68    26    17    85
113   157    79   123   111   155    87   131   131   175    13    57    -3
Region 10

 72    37    22    46
 98   150    86   138   106   158    62   114    38    90    76   128    -26
Region 6

 11     6    17    38
190   120   116    46   188   118   164    94   112    42   186   116   188
Region 1

  6     6    17    39
195   115   121    41   193   113   169    89   121    41   187   107   203
Region 13
```

14	42	16	74									
151	87	149	85	147	83	127	63	147	83	73	9	143
Region 1												
48	42	16	74									
117	121	115	119	113	117	93	97	113	117	39	43	41
Region 2												
14	42	16	105									
151	87	147	85	147	83	127	63	209	145	11	-53	143
Region 9												
48	42	16	105									
117	121	115	119	113	117	93	97	175	179	-23	-19	41
Region 10												
46	42	16	74									
119	119	117	117	115	115	95	95	115	115	41	41	47
Region 1 - Region 2												
73	42	3	38									
92	146	90	144	62	116	94	148	16	70	86	140	-34
Region 8												
10	42	15	40									
155	83	153	81	149	77	133	61	83	11	145	73	155
Region 1 - Region 13												
22	42	15	40									
143	95	141	93	137	89	121	73	71	23	133	35	119
Region 1												
34	42	15	40									
131	107	129	105	125	101	109	85	59	35	121	97	83
Region 1												
46	42	15	40									
119	119	117	117	113	113	97	97	47	47	109	109	47
Region 1 - Region 2												

From this first application we arrive at the following conclusions:

- The real number of regions is obtained by means of the following bound's rule: "For a given variable, the bounds in one region must not be contained in the bounds of some other region, except when these bounds are the same."

- If the operation point is in between two regions (located on the bound), the maximum value appears twice in the output vector.

- If the operation point is moving inside of a region, the maximum value in the output vector changes. Therefore, looking for these changes as a function of time, it is possible to perform some predictions.

2.1.2.3 2nd Application As a second application we are going to consider the example given in page 2.7. The outputs obtained are as follows

```
20    50     5
60   -30    80   -60
Region 3

 4   500    30
35  -955   955  -967
Region 3

13    68     7
58   -68   114  -108
Region 3

17     3     7
58    62   -16    30
Region 2
```

It is very easy to see that the second result is incorrect. The reason of this is because, the input vector (4,500,30) does not belong to any given region.

2.1.3 Sequential Recognition Technique

This is a straight forward technique because we need only to check for each region if the boundaries are met or not by the measurements. The region that meets all the boundaries, gives the operate region.

For the example given in page 2.7, we have

REGION \ MEASURES	X_1	X_2	X_3
1	✓		✓
2	✓		✓
3	✓	✓	✓
4			

The principal advantage of this technique over the recognition matrix technique is that if because of a measurement error the components of the input vector do not belong to any given region, this situation is detected. On the other hand, with the recognition matrix technique, we arrive at a wrong result.

2.1.4 Conclusions

The essential features of the Classification Method could be summarized as follows:

- It possesses a high flexibility; this means that the diagnosis can be adjusted to any desirable level of identification.
- The possibility of making the diagnosis without interruption to the normal operation.
- The possibility to make a reduction in the number of the test points, without changing the diagnosis level.
- The possibility to perform trend analysis.
- The small amount of post-calculations that are necessary in order to get the operation diagnosis.

2.1.5 Applications

The application of this method covers linear and nonlinear circuits/systems. In relation to the level of identification, it seems to be the more important application at the modular level.

2.1.6 Computer Requirements

Only a few simple calculations are necessary to be performed, after the measured values are taken. Therefore, there are no strong requirements imposed over the computer system needed.

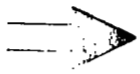
2.2 Key Element Search Method [13]

2.2.1. General

A major feature of this method involves the diagnosis of a fault component that can be done with an insufficient number of measurements to solve for network elements values. Of course, this is a most likelihood approach. We are going to consider two techniques

a) Nonlinear Technique
b) Linear Technique

In both techniques the idea of this method is to define an index that is function of the measurements and only one component. For each component in the network we are going to have an associated index value, therefore, we can prepare the following table as an example:



j	Sj
1	0.0135
2	0.0275
3	0.0036
4	0.0085
5	0.0123

The most likely faulty component, will be the component that has the minimum value of the index. In our case it will be the third component.

2.2.2 Nonlinear Technique

Consider a network with N components and M measurable nodes associated with it. The admittance* in each one of the measurable nodes will be given by

$$y_j = f_j\ (x_1, x_2, \ldots . x_N) \quad \text{for } j = 1, 2, \ldots . M \tag{2.4}$$

where,

x_i = value of component i
y_j = admittance of node j

Let us call X_{jo} (for j = 1, 2, ...N) the normal values of the components, and with g_j (for j = 1, 2, ...M) the measured admittance values. We are going to define the index Sj in the following way:

$$S_j = \sum_{i=1}^{M} \left[g_i - f_i\ (x_{10}, x_{20}, \ldots x_{(j-1)0}, x_j, x_{(j+1)0}, \ldots X_{NO}) \right]^2$$

$$= \sum_{i=1}^{M} \left[g_i - f_i(x_j) \right]^2 \tag{2.5}$$

* The reason to use the admittance instead of the impedance is because the former is more easy to get applying topological laws.

We are looking for the minimum value of S_j, that is function of the component x_j. Therefore, the derivate of (2.5) is given by

$$\frac{dS_j}{dx_j} = -2 \sum_{i=1}^{M} g_i \frac{df_i(x_j)}{dx_j} + 2 \sum_{i=1}^{M} f_i(x_j) \frac{df_i(x_j)}{dx_j} \qquad (2.6)$$

Setting this derivative equal to zero

$$\sum_{i=1}^{M} g_i \frac{df_i(x_j)}{dx_j} - \sum_{i=1}^{M} f_i(x_j) \frac{df_i(x_j)}{dx_j} = 0 \qquad (2.7)$$

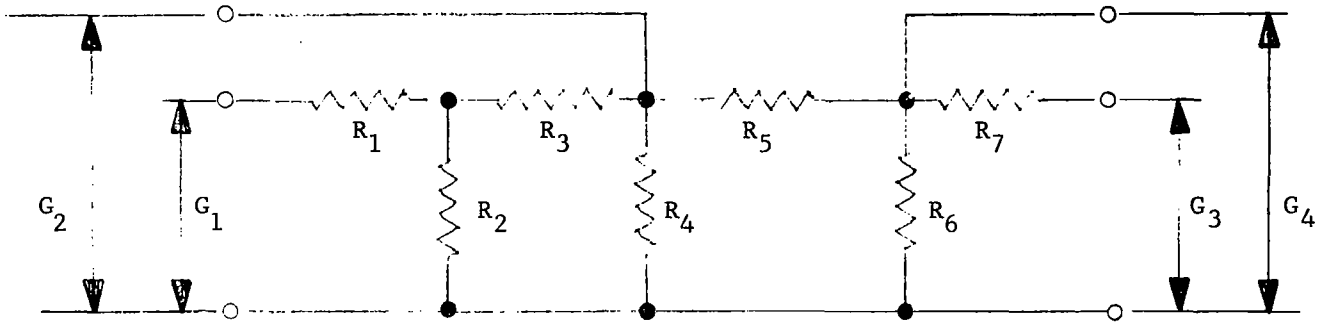
The numerical solution of equation (2.7) can be accomplished by some iterative technique such as Newton-Rapshon's.

The diagnosis procedure is as follows:

- Take the measurements at M pre-defined points. This number of points is less than the number of components.
- For each component, and by means of equation (2.7), we shall find the component value, that makes a minimum in equation (2.5).
- Substituting the value x_j as calculated in the previous step into the equation (2.5), we get S_j.
- With each component, we have associated a value of S_j. The smallest value of S_j gives us the most likely fault component.

With this method, the computer program depends on the circuit under test. Therefore, we are going to consider first the application, and then the program. Two applications will be studied.

2.2.2.1 1st Application Let us consider the circuit shown in Figure 2.8.



R_1= 63.0 Ω ; R_4= 46.0 Ω ; R_7 = 94.0 Ω ; G_3= 0.0068 mho
R_2= 43.0 Ω ; R_5= 100.0 Ω ; G_1 = 0.0106 mho ; G_4= 0.0195 mho
R_3= 78.0 Ω ; R_6= 83.0 Ω ; G_2 = 0.0354 mho ;

Figure 2.8

The admittance G1, G2, G3 and G4 are given by

$$\left\{\begin{array}{l} G_1 = \dfrac{(R_2+R_3)\,(R_4+R_5+R_6) + (R_5+R_6)R_4}{(R_4+R_5+R_6)\,(R_1R_2+R_2R_3+R_3R_1) + (R_5+R_6)\,(R_1R_4+R_4R_2)} \\ G_2 = \dfrac{(R_2+R_3)\,R_4 + (R_5+R_6)\,(R_2+R_3+R_4)}{(R_2+R_3)\,R_4\,(R_5+R_6)} \\ G_3 = \dfrac{(R_5+R_6)\,(R_2+R_3+R_4) + R_4(R_2+R_3)}{(R_2+R_3+R_4)\,(R_7R_6+R_6R_5+R_5R_7) + (R_2+R_3)\,(R_7R_4+R_4R_6)} \\ G_4 = \dfrac{(R_2+R_3)R_4+(R_5+R_6)\,(R_2+R_3+R_4)}{R_5R_6(R_2+R_3+R_4) + R_4R_6(R_2+R_3)} \end{array}\right. \qquad (2.8)$$

In order to be able to prepare a more simple program, we must write the admittance in each node as well as the first and second derivative of it, in a general form. Let us write as follows,

$$G_i = \frac{A_i R_j + B_i}{C_i R_j + D_i} \tag{2.9}$$

$$\frac{dG_i}{dR_j} = \frac{A_i(C_i R_j + D_i) - C_i (A_i R_j + B_i)}{(C_i R_j + D_i)^2}$$

$$= \frac{A_i D_i - B_i C_i}{(C_i R_j + D_i)^2} \tag{2.10}$$

and, similarly

$$\frac{d^2 G_i}{dR_j^2} = \frac{2C_i(B_i C_i - A_i D_i)}{(D_i + R_j C_i)^3} \tag{2.11}$$

for all j=1,2...7; and i=1,...4. In Table I the values A_i, B_i, C_i, and D_i are given for each i(i=1,...4) as function of R_j(j=1,2,...7).

The second derivative will be needed if the iterative technique of Newton-Raphson is used. In our application, we will use a simpler approach.

In order to be able to check the results obtained from the computer run, Table II gives the values of G_1, G_2, G_3, and G_4 for different combinations of R_1, R_2, R_3, R_4, R_5, R_6, and R_7.

The flowchart in Figure 2.9 is self-explanatory; however, we want to point out that the technique used to find the root of the equation 2.7, is the simple one of changing the size of the step any time that the value of the root is over. This is possible to apply because

- There is only one root
- The value of the root is real and greater than zero (for this reason we start from xj1=0)

The results obtained are as follow

Table 1

$Gx_i = \frac{AR_j + B}{CR_j + D}$			$\frac{dGx_i}{dR_j} = \frac{AD - BC}{(D + R_j C)^2}$		$\frac{d^2Gx_i}{dR_j^2} = \frac{2C(BC-AD)}{(D + R_j C)^3}$
R_j	i	A	B	C	D
R_1	1	0	$(R_2 + R_3)(R_4 + R_5 + R_6) + R_4(R_5 + R_6)$	$(R_2 + R_3)(R_4 + R_5 + R_6) + R_4(R_5 + R_6)$	$R_2R_3(R_4 + R_5 + R_6) R_2R_4(R_5 + R_6)$
	2	0	$(R_2 + R_3)(R_4 + R_5 + R_6) + R_4(R_5 + R_6)$	0	$(R_2 + R_3)(R_5 + R_6)R_4$
	3	0	$(R_2 + R_3)(R_4 + R_5 + R_6) + R_4(R_5 + R_6)$	0	$(R_2 + R_3 + R_4)(R_5R_7 + R_7R_6 + R_6R_5) + (R_2 + R_3) R_4(R_6 + R_7)$
	4	0	$(R_2 + R_3)(R_4 + R_5 + R_6) + R_4(R_5 + R_6)$	0	$R_6R_5(R_2 + R_3 + R_4) + R_4R_6(R_2 + R_3)$
R_2	1	$R_4 + R_5 + R_6$	$(R_3 + R_4)(R_5 + R_6) + R_3R_4$	$(R_1 + R_3)(R_4 + R_5 + R_6) + R_4(R_5 + R_6)$	$R_1R_3(R_4 + R_5 + R_6) + R_1R_4(R_5 + R_6)$
	2	$R_4 + R_5 + R_6$	$(R_3 + R_4)(R_5 + R_6) + R_3R_4$	$R_4(R_5 + R_6)$	$R_3R_4(R_5 + R_6)$
	3	$R_4 + R_5 + R_6$	$(R_3 + R_4)(R_5 + R_6) + R_3R_4$	$R_7(R_5 + R_6) + R_4(R_7 + R_6) + R_5R_6$	$R_7(R_3 + R_4)(R_5 + R_6) + R_3R_4 (R_7 + R_6) + R_5R_6(R_3 + R_4)$
	4	$R_4 + R_5 + R_6$	$(R_3 + R_4)(R_5 + R_6) + R_3R_4$	$R_6(R_4 + R_5)$	$(R_3 + R_4)R_5R_6 + R_3R_4R_6$
R_3	1	$R_4 + R_5 + R_6$	$(R_2 + R_4)(R_5 + R_6) + R_2R_4$	$(R_1 + R_2)(R_4 + R_5 + R_6)$	$R_1R_2(R_4 + R_5 + R_6) + (R_1 + R_2)(R_5 + R_6)R_4$
	2	$R_4 + R_5 + R_6$	$(R_2 + R_4)(R_5 + R_6) + R_2R_4$	$R_4(R_5 + R_6)$	$R_2R_4(R_5 + R_6)$
	3	$R_4 + R_5 + R_6$	$(R_2 + R_4)(R_5 + R_6) + R_2R_4$	$R_6(R_4 + R_5 + R_7) + R_4R_5 + R_5R_7$	$(R_2 + R_4)(R_5R_7 + R_7R_6 + R_6R_5) + R_2R_4(R_6 + R_7)$
	4	$R_4 + R_5 + R_6$	$(R_2 + R_4)(R_5 + R_6) + R_2R_4$	$R_6(R_4 + R_5)$	$(R_2 + R_4)R_5R_6 + R_2R_4R_6$
R_4	1	$R_2 + R_3 + R_5 + R_6$	$(R_2 + R_3)(R_5 + R_6)$	$R_1(R_2 + R_3 + R_5 + R_6) + R_2(R_3 + R_5 + R_6)$	$(R_5 + R_6)(R_1R_2 + R_2R_3 + R_3R_1)$
	2	$R_2 + R_3 + R_5 + R_6$	$(R_2 + R_3)(R_5 + R_6)$	$(R_2 + R_3)(R_5 + R_6)$	0
	3	$R_2 + R_3 + R_5 + R_6$	$(R_2 + R_3)(R_5 + R_6)$	$R_5R_7 + R_7R_6 + R_6R_5 + (R_6 + R_7)(R_2 + R_3)$	$(R_2 + R_3)(R_5R_7 + R_7R_6 + R_6R_5)$
	4	$R_2 + R_3 + R_5 + R_6$	$(R_2 + R_3)(R_5 + R_6)$	$R_6(R_2 + R_3 + R_5)$	$(R_2 + R_3)R_5R_6$
R_5	1	$R_2 + R_3 + R_4$	$(R_4 + R_6)(R_2 + R_3) + R_4R_6$	$R_1(R_2 + R_3) + R_4(R_1 + R_2) + R_2R_3$	$(R_4 + R_6)R_1(R_2 + R_3) + R_4R_6 (R_1 + R_2) + R_2R_3(R_4 + R_6)$
	2	$R_2 + R_3 + R_4$	$(R_4 + R_6)(R_2 + R_3) + R_4R_6$	$R_4(R_2 + R_3)$	$R_4R_6(R_2 + R_3)$
	3	$R_2 + R_3 + R_4$	$(R_4 + R_6)(R_2 + R_3) + R_4R_6$	$(R_6 + R_7)(R_2 + R_3 + R_4)$	$R_6R_7(R_2 + R_3 + R_4) + R_4R_7 (R_2 + R_3) + R_4R_6(R_2 + R_3)$
	4	$R_2 + R_3 + R_4$	$(R_4 + R_6)(R_2 + R_3) + R_4R_6$	$(R_2 + R_3 + R_4)R_6$	$R_4R_6(R_2 + R_3)$
R_6	1	$R_2 + R_3 + R_4$	$(R_4 + R_5)(R_2 + R_3) + R_4R_5$	$R_1(R_2 + R_3) + R_4(R_1 + R_2) + R_2R_3$	$(R_4 + R_5)R_1(R_2 + R_3) + R_4R_5 (R_1 + R_2) + R_2R_3(R_4 + R_5)$
	2	$R_2 + R_3 + R_4$	$(R_4 + R_5)(R_2 + R_3) + R_4R_5$	$R_4(R_2 + R_3)$	$R_4R_5(R_2 + R_3)$
	3	$R_2 + R_3 + R_4$	$(R_4 + R_5)(R_2 + R_3) + R_4R_5$	$(R_7 + R_5)(R_2 + R_3 + R_4) + R_4(R_2 + R_3)$	$R_5R_7(R_2 + R_3 + R_4) + R_4R_7 (R_2 + R_3)$
	4	$R_2 + R_3 + R_4$	$(R_4 + R_5)(R_2 + R_3) + R_4R_5$	$(R_2 + R_3 + R_4)R_5 + R_4(R_2 + R_3)$	0
R_7	1	0	$(R_2 + R_3)(R_4 + R_5 + R_6) + R_4(R_5 + R_6)$	0	$(R_4 + R_5 + R_6)(R_1R_2 + R_2R_3 + R_3R_1) + (R_5 + R_6)(R_1 + R_2)R_4$
	2	0	$(R_2 + R_3)(R_4 + R_5 + R_6) + R_4(R_5 + R_6)$	0	$(R_2 + R_3)(R_5 + R_6)R_4$
	3	0	$(R_2 + R_3)(R_4 + R_5 + R_6) + R_4(R_5 + R_6)$	$(R_5 + R_6)(R_2 + R_3 + R_4) + R_4(R_2 + R_3)$	$R_5R_6(R_2 + R_3 + R_4) + R_4R_6(R_2 + R_3)$
	4	0	$(R_2 + R_3)(R_4 + R_5 + R_6) + R_4(R_5 + R_6)$	0	$R_6R_5(R_2 + R_3 + R_4) + R_4R_6(R_2 + R_3)$

Table 2

R_1	R_2	R_3	R_4	R_5	R_6	R_7	G_1	G_2	G_3	G_4
63.00	43.00	78.00	46.00	100.00	83.00	94.00	0.0106	0.0354	0.0068	0.0195
63.00	43.00	97.50	46.00	100.00	83.00	94.00	0.0104	0.0343	0.0068	0.0194
63.00	43.00	136.50	46.00	100.00	83.00	94.00	0.0102	0.0327	0.0068	0.0193
63.00	43.00	78.00	46.00	100.00	103.75	94.00	0.0106	0.0349	0.0065	0.0171
63.00	43.00	78.00	46.00	100.00	145.25	94.00	0.0105	0.0340	0.0061	0.0143
63.00	53.75	78.00	46.00	100.00	83.00	94.00	0.0100	0.0347	0.0068	0.0195
63.00	72.25	78.00	46.00	100.00	83.00	94.00	0.0092	0.0337	0.0068	0.0194
63.00	43.00	78.00	46.00	100.00	83.00	117.50	0.0106	0.0354	0.0059	0.0195
63.00	43.00	78.00	46.00	100.00	83.00	164.50	0.0106	0.0354	0.0046	0.0195
63.00	43.00	78.00	46.00	125.00	83.00	94.00	0.0105	0.0348	0.0067	0.0183
63.00	43.00	78.00	46.00	175.00	83.00	94.00	0.0105	0.0338	0.0065	0.0168
78.75	53.75	117.00	57.50	125.00	103.75	117.50	0.0083	0.0276	0.0055	0.0155
78.75	53.75	156.00	57.50	125.00	103775	117.50	0.0082	0.0265	0.0054	0.0155
78.75	64.50	97.50	57.50	125.00	103.75	117.50	0.0081	0.0279	0.0055	0.0156
78.75	86.00	97.50	57.50	125.00	103.75	117.50	0.0075	0.0272	0.0055	0.0155
78.75	53.75	97.50	57.50	125.00	103.75	141.00	0.0084	0.0283	0.0048	0.0156
78.75	53.75	97.50	57.50	125.00	103.75	188.00	0.0084	0.0283	0.0039	0.0156
78.75	53.75	97.50	57.50	125.00	124.50	117.50	0.0084	0.0280	0.0052	0.0140
78.75	53.75	97.50	57.50	125.00	166.00	117.50	0.0084	0.0274	0.0049	0.0120
94.50	64.50	117.00	69.00	150.00	166.00	141.00	0.0070	0.0231	0.0043	0.0110

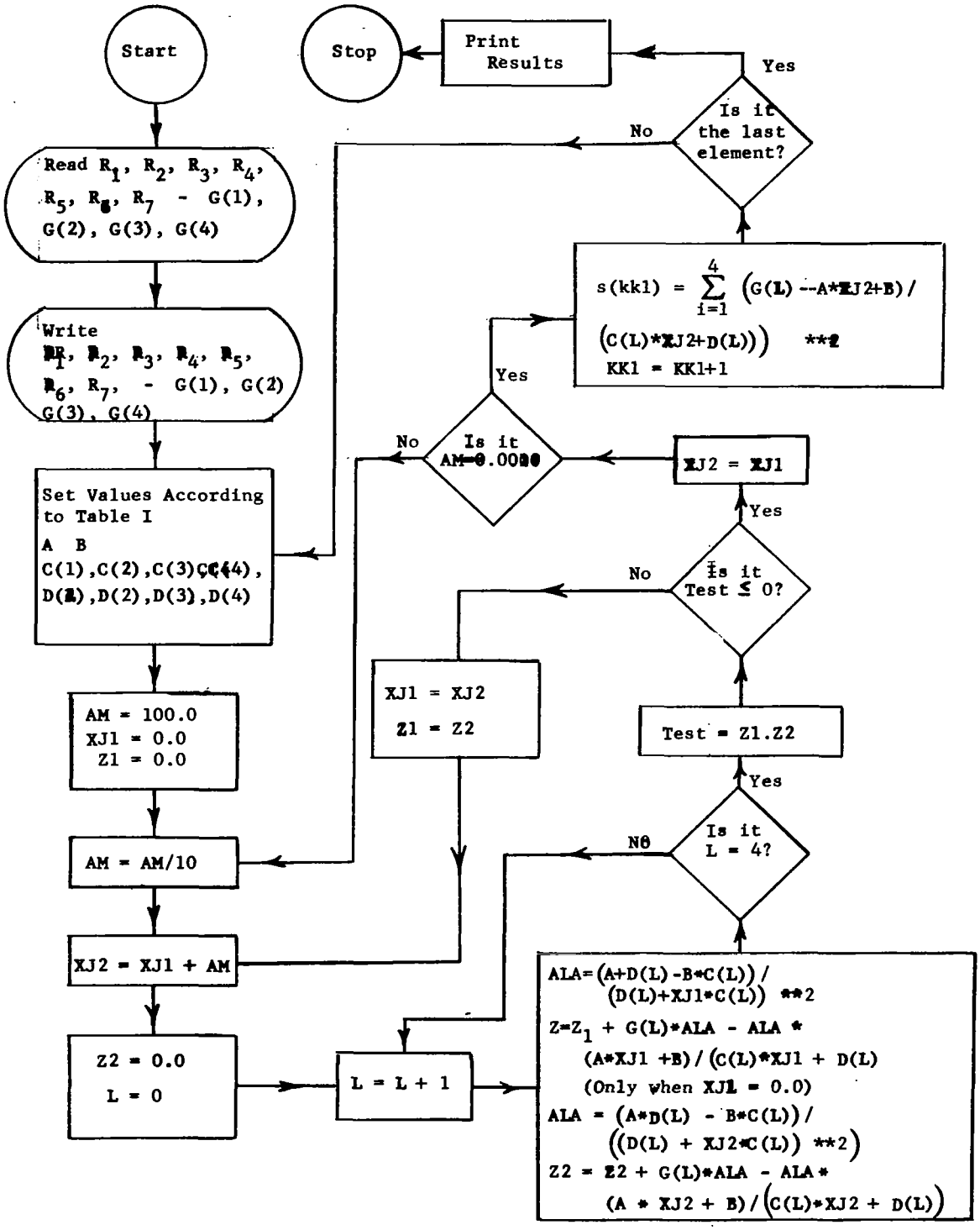


Figure 2.9 Nonlinear Technique
Key Element Search

Output

G1 = 0.0102 G2=0.0327 G3=0.0068 G4=0.0193

1	0.00000000
2	0.00000196
→ 3	0.00000001
4	0.00000014
5	0.00000543
6	0.00000721
7	0.00000000

G1=0.0106 G2=0.0354 G3=0.0046 G4=0.0195

1	0.00000000
2	0.00000524
3	0.00000523
4	0.00000523
5	0.00000514
6	0.00000513
→ 7	0.00000000

G1=0.0105 G2=0.0388 G3=0.0065 G4=0.0168

1	0.00000000
2	0.00000826
3	0.00000709
4	0.00000708
→ 5	0.00000000
6	0.00000098
7	0.00000000

G1=0.0105 G2=0.0340 G3=0.0061 G4=0.0143

1	0.00000000
2	0.00002818
3	0.00002704
4	0.00002704
5	0.00000245
→ 6	0.00000001
7	0.00000000

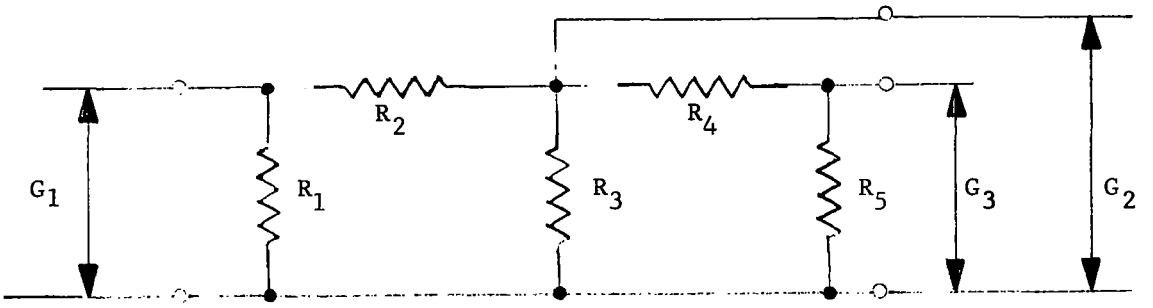
The arrow at the left side of the component number, in each one of the four given cases, indicates the faulty component (See Table II).

The following conclusions are obtained from this first application

- Always the first and seventh component have the index Sj=0. The reason of this is because G2, G3, and G4 are not a function of R_1; and G1, G2, and G4 are not a function of R_7. In order for this technique to work properly the admittance in each node must be a function of all the components (See 2nd Application).

- Disregarding the first and seventh component, this technique gives good results.

2.2.2.2 2nd Application Let us consider the circuit shown in Figure 2.10.



R_1=43.0 Ω
R_2=78.0 Ω
R_3=46.0 Ω
R_4=100.0 Ω
R_5= 83.0 Ω
G_1=0.0319 mho
G_2=0.0354 mho
G_3=0.0195 mho

Figure 2.10

The admittances G_1, G_2, and G_3 are given by

$$\left\{\begin{array}{l} G_1 = \dfrac{(R_3+R_4+R_5)\ (R_1+R_2) + R_3(R_4+R_5)}{(R_3+R_4+R_5)\ R_1R_2 + R_1R_3(R_4+R_5)} \\ G_2 = \dfrac{(R_1+R_2+R_4+R_5)\ R_3+(R_1+R_2)\ (R_4+R_5)}{(R_1+R_2)\ R_3(R_4+R_5)} \\ G_3 = \dfrac{(R_4+R_5)\ (R_1+R_2+R_3) + (R_1+R_2)\ R_3}{R_4R_5(R_1+R_2+R_3) + (R_1+R_2)\ R_3R_5} \end{array}\right.$$

The general form of the admittance in each node as well as the first and second derivatives are given by equations (2.9), (2.10) and (2.11). The results of this application could be checked by means of Table III. The flow-chart needed is similar to the one shown in Figure 2.9. The results obtained are as follow

Output

```
G1=0.0319     G2=0.0349     G3=0.0171

      1     0.00000629
      2     0.00000603
      3     0.00000579
      4     0.00000053
      5***  0.00000000

G1=0.0310     G2=0.0282     G3=0.0189

      1     0.00005046
      2     0.00002209
      3***  0.00000001
      4     0.00003459
      5     0.00004897
```

In both cases, the results are correct.

TABLE III

R_1	R_2	R_3	R_4	R_5	G_1	G_2	G_3
43.00	78.00	46.00	100.00	83.00	0.0319	0.0354	0.0195
53.75	78.00	46.00	100.00	83.00	0.0273	0.0347	0.0195
64.50	78.00	46.00	100.00	83.00	0.0242	0.0342	0.0194
43.00	97.50	46.00	100.00	83.00	0.0307	0.0343	0.0194
43.00	117.00	46.00	100.00	83.00	0.0297	0.0334	0.0194
43.00	78.00	57.50	100.00	83.00	0.0314	0.0311	0.0192
43.00	78.00	69.00	100.00	83.00	0.0310	0.0282	0.0189
43.00	78.00	46.00	125.00	83.00	0.0319	0.0348	0.0183
43.00	78.00	46.00	150.00	83.00	0.0318	0.0342	0.0175
43.00	78.00	46.00	100.00	103.75	0.0319	0.0349	0.0171
43.00	78.00	46.00	100.00	124.50	0.0318	0.0344	0.0155
53.75	97.50	57.50	125.00	124.50	0.0255	0.0280	0.0140
53.75	97.50	57.50	125.00	166.00	0.0254	0.0274	0.0120

2.2.3 Linear Technique

Let us take the difference between the measured value and the normal value of the admittance i

$$\Delta f_i = g_i - f_{io} \tag{2.12}$$

where f_{i_o} is the value of equation (2.4) when all the components are set equal to the normal values. We assume that the difference of equation (2.12) is due to the failure of an element x_j in the network.

In order to express the equation (2.12) in a more suitable way, we define

$$Q_{ij} = \left.\frac{\partial f_i}{\partial x_j}\right|_{x=x_o} \tag{2.13}$$

Now, we can make the following linear approximation using the first two terms of the Taylor series expansion, then

$$f_i \cong f_{io} + \Delta x_1 Q_{i1} + \Delta x_2 Q_{i2} + \ldots + \Delta X_N Q_{iN} = \tag{2.14}$$

Now, if we assume that the measured values are given by an equation like (2.14), the equation (2.12) becomes

$$\Delta f_i = \Delta x_1 Q i_1 + \Delta x_2 Q i_2 + \ldots + \Delta x_N Q_{iN} = \sum_{j=1}^{N} \Delta x_j Q i_j \tag{2.15}$$

We have M equations like (2.15) (i=1,2,...M). Our index is defined as

$$S_j = \sum_{i=1}^{M} \left[\Delta f_i - \Delta x_j Q i_j\right]^2 \tag{2.16}$$

We must find for which component (xj) the equation (2.16) is a minimum. This component will be the likely fault component. Taking the derivative of (2.16), we have

$$\frac{dS_j}{dx_j} = -2 \sum_{i=1}^{M} \Delta f_i Q_{ij} + 2 \sum_{i=1}^{M} \Delta x_j Q_{ij}^2 \tag{2.17}$$

Setting this derivative equal to zero, and solving for Δx_j.

$$\Delta x_j = \frac{\sum_{i=1}^{M} \Delta f_i Q_{ij}}{\sum_{i=1}^{M} Q_{ij}^2} \tag{2.18}$$

Therefore, our S_{jmin} is given by

$$S_{jmin} = \sum_{i=1}^{M} \Delta f_i^2 - 2 \frac{\sum_{i=1}^{M} \Delta f_i Q_{ij}}{\sum_{i=1}^{M} Q_{ij}^2} \sum_{i=1}^{M} \Delta f_i Q_{ij}$$

$$+ \left[\frac{\sum_{i=1}^{M} f_i Q_{ij}}{\sum_{i=1}^{M} Q_{ij}^2} \right]^2 \sum_{i=1}^{M} Q_{ij}^2 \tag{2.19}$$

Then

$$S_{jmin} = \sum_{i=1}^{M} \Delta f_i^2 - \frac{\left[\sum_{i=1}^{M} \Delta f_i Q_{ij} \right]^2}{\sum_{i=1}^{M} Q_{ij}^2} \tag{2.20}$$

Let

$$M_j = \frac{\left[\sum_{i=1}^{M} \Delta f_i Q_{ij} \right]^2}{\sum_{i=1}^{M} Q_{ij}^2} \tag{2.21}$$

To minimize S_{jmin} is the same as to maximize M_j, because $\sum_{I=1}^{M} \Delta f_i^2$ and S_{jmin} are always positive quantities. Then, we are going to search for the component that has the maximum value of M_j. This component will be the likely fault component.

Therefore, for each component j there is an associated value M_j

j	M_j
1	0.36
2	0.48
3	1.73
4	1.72
5	1.68
6	0.48
7	1.77
8	0.85
9	1.25
10	1.38

In this case, the most likely fault component will be the number seven.

The diagnosis procedure is as follows:

- Take the measurements at M pre-defined points. This number of points is less than the number of components.
- For each component, and by means of equation (2.21) we shall find the value M_j.
- With each component, we have associated M_j. The highest value of M_j gives us the most likely fault component.

2.2.3.1 Application

Let us consider the circuit shown in Figure 2.8. The flow chart is indicated in Figure 2.11, the same is self-explanatory. The results obtained are as follows:

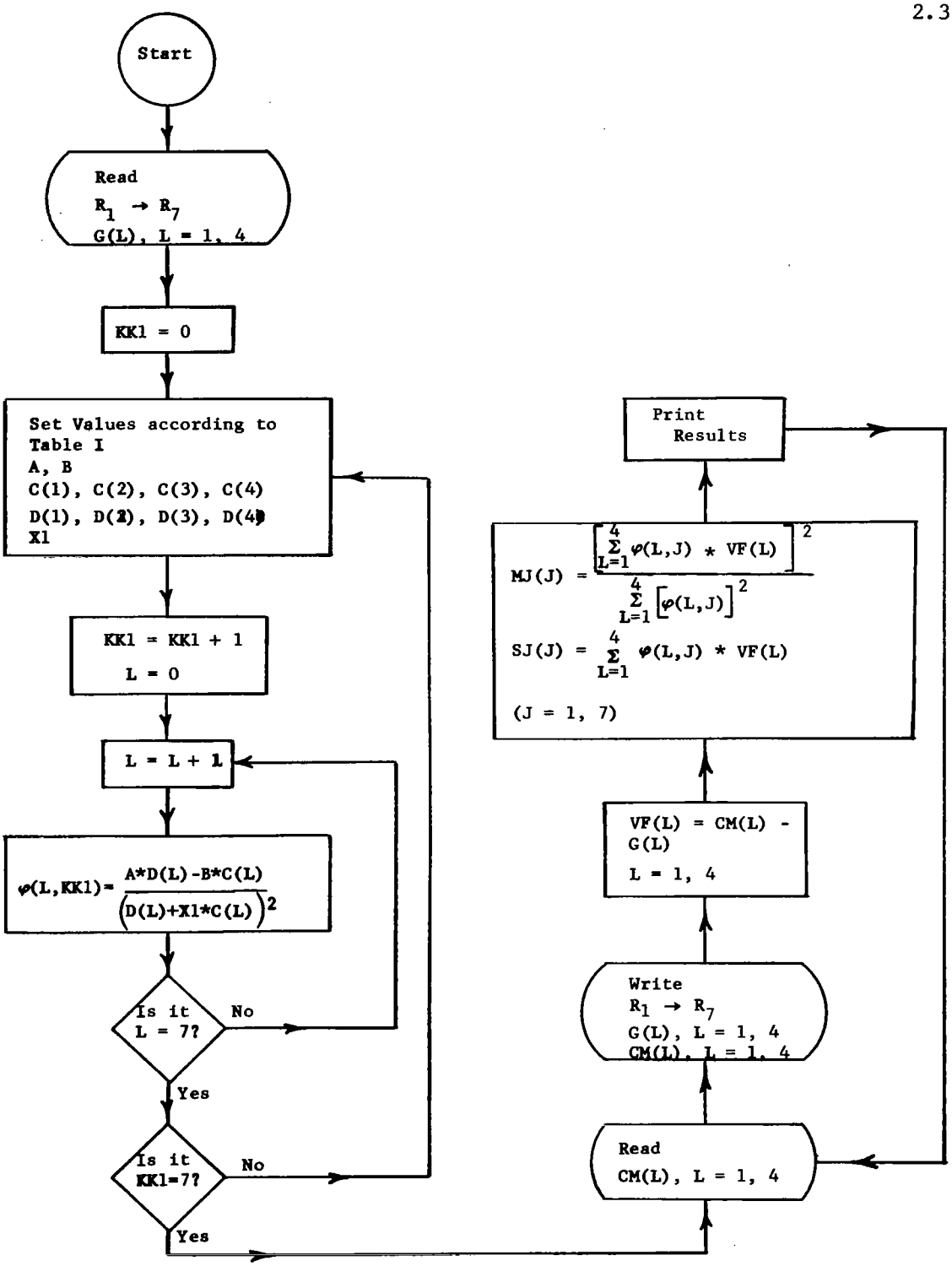


Figure 2.11 Linear Technique
Key Element Search

Output

G1=0.0102 G2=0.0327 G3=0.0068 G4=0.0193

J	MJ	SJ
1	0.00000015	0.00000733
2	0.00000531	0.00000217
3***	0.00000748	0.00000000
4	0.00000735	0.00000013
5	0.00000206	0.00000542
6	0.00000110	0.00000638
7	0.00000000	0.00000749

G1=0.0106 G2=0.0354 G3=0.0046 G4=0.0195

J	MJ	SJ
1	0.00000000	0.00000484
2	0.00000000	0.00000483
3	0.00000000	0.00000483
4	0.00000000	0.00000483
5	0.00000005	0.00000478
6	0.00000018	0.00000465
7***	0.00000484	0.00000000

In both techniques the results are correct (See Table II). The advantages of this technique over the nonlinear technique are the following

- There is no need for all the measured admittances to be a function of the components that belong to the circuit under test.
- There are less calculations that must be performed after the measurements are taken.

2.2.4 Conclusions

The two techniques given in the Key Element Search Method involve the use of insufficient data to isolate faulty components. Both techniques give a good answer if measurements can be made with sufficient accuracy, and the component tolerances are small in comparison with the derivation of the faulty component from its nominal value. From the points of view of the types of circuit that is possible to be handled, and post-calculation requirements, it is more convenient to use the linear technique. The future improvement that will be possible to do in this method is,

- The development of the relationship of diagnostic accuracy to the number of test terminals, the component tolerances and the measurement accuracy.

- The development of techniques which will facilitate the treatment of nonlinear circuit elements.

2.2.5 Applications The application of this method is for linear circuits at component level, and the extension to include linear systems is straight forward.

2.2.6 Computer Requirements

Only a few simple calculations are necessary after the measured values are taken. Therefore, there is no strong requirements imposed on the computer system needed.

2.3 Iterative Method [8], [9], [10]

2.3.1 General

The iterative method is based on the idea that the configuration of the circuit is known as well as the normal components values, but at some time after that, the components values are unknown. The procedure to get the component values is an interactive one using maximum likelihood estimation technique at each stage.

Let the circuit under test be the one that is shown in Figure 2.12.

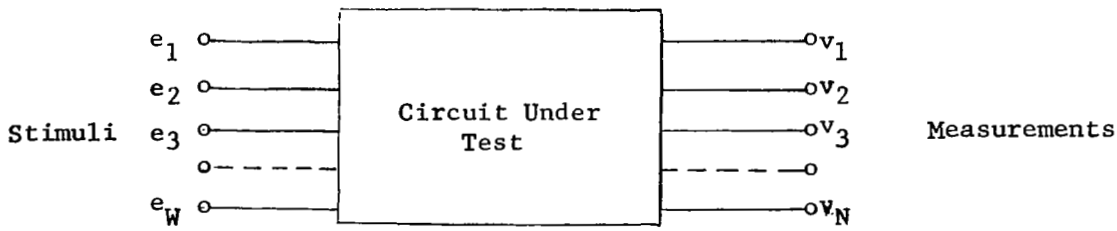


Figure 2.12

Let

M = Number of unknown circuit components.
N = Number of measurements made.
x_j = True values of unknown circuit components $(i \leq j \leq M)$
v_i = Observed values of measurements made in response to specified stimuli $(1 \leq i \leq N)$.
$f_i(x_1, x_2, \ldots x_M)$ = Theoretical expression for the ith measurement $(1 \leq i \leq N)$.

δ_i = Measurement error for ith measurement ($1 \leq i \leq N$)
σ_i = Standard deviation of ith measurement error.
Errors assumed independent, normally distributed with zero mean ($1 \leq i \leq N$).
x_{jo} = Initial values of circuit components ($1 \leq j \leq M$).

The following relation between M and N must hold

$$N \geq M \tag{2.22}$$

This does not give us an indication that we need more test points than circuit components. On the contrary, if M is the number of test points, the equation (2.22) tells us that we need to make the measurements at N/M frequencies in the given test points.

For the ith measurement, we can write

$$v_i = f_i (x_1, x_2, \ldots . x_M) + \delta_i \tag{2.23}$$

For maximum likelihood estimation the following sufficient statistic is used

$$S = \sum_{i=1}^{N} \frac{\left[v_i - f_i(x_1, x_2, \ldots x_N) \right]^2}{\sigma_i^2} \tag{2.24}$$

The maximum likelihood estimate of the components value is the set of x_i values which causes S to be a minimum. Therefore,

$$\frac{\partial S}{\partial x_j} = 0 \qquad 1 \leq j \leq M \tag{2.25}$$

If we designate with

$$f_{ijo} = \frac{\partial}{\partial x_j} f_i(x_1, x_2, \ldots x_M) \quad , \quad 1 \leq i \leq N; \; 1 \leq j \leq M \tag{2.26}$$

We can write as a first approximation of the Taylor series

$$f_i(x_1, x_2, \ldots x_M) = f_{io} + \sum_{j=1}^{M} \xi_j \, f_{ijo} \tag{2.27}$$

where $f_{io}=f_i(x10,x20,\ldots xM0)$, and $\xi_j=x_j-x_{jo}$. Now by means of (2.27), the equation (2.24) is written as follows

$$S = \sum_{i=1}^{N} \frac{\left[v_i - f_{io} - \sum_{j=1}^{M} \xi_j f_{ijo}\right]^2}{\sigma_i^2} \tag{2.28}$$

Let $\eta_i = v_i - f_{io}$, then equation (2.28) becomes

$$S = \sum_{i=1}^{N} \frac{\left[\eta_i - \sum_{j=1}^{M} \xi_j f_{ijo}\right]^2}{\sigma_i^2} \tag{2.29}$$

In order to carry out the minimization procedure, we write

$$\xi = \left[\xi_1, \xi_2, \ldots \xi_M\right] \tag{2.30}$$

$$\eta = \left[\eta_1, \eta_2, \ldots \eta_N\right] \tag{2.31}$$

$$F = \begin{bmatrix} f_{110} & f_{120} & \cdots & f_{1M0} \\ f_{210} & f_{220} & \cdots & f_{2M0} \\ \cdot & \cdot & \cdots & \cdot \\ f_{N10} & f_{N20} & \cdots & f_{NM0} \end{bmatrix} \tag{2.32}$$

$$P = \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma_2^2} & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & \cdots & \frac{1}{\sigma_N^2} \end{bmatrix} \tag{2.33}$$

In matrix form, (2.28) becomes

$$S = \eta P \eta_t - 2\eta A \xi_t + \xi F_t A \xi_t \qquad (2.34)$$

where the subscript t designates "transpose" and A is given by A=PF.

The derivative of equation (2.34) is as follows.

$$-2\eta A + 2\xi F_t A = 0 \qquad (2.35)$$

$$\xi = \eta A (F_t A)^{-1} \qquad (2.36)$$

The diagnosis procedure is as follows:

- Take the N measurements at the pre-defined points. If the number of points is less than N, we have to take the measurements at more than one frequency.
- Calculate the matrix P defined by means of (2.33).
- Set the components values to the initial values.
- Calculate the following matrices

  Matrix F by means of (2.32)

  Matrix η by means of (2.31)

  Matrix A by means of A=PF

  Matrix ξ by means of (2.36)

- Calculate S_1 by means of equation (2.24) and change the component values by means of $x_j = \xi_j + x_{jo}$ (j=1, ... M);

  calculate S_2 by means of equation (2.25) with the new set of values. If $\frac{S_1}{S_2} \cong 1$, the new set of values is the true one;

  otherwise, assuming that this new set of values is the initial one, go back to step 4 and repeat the process.

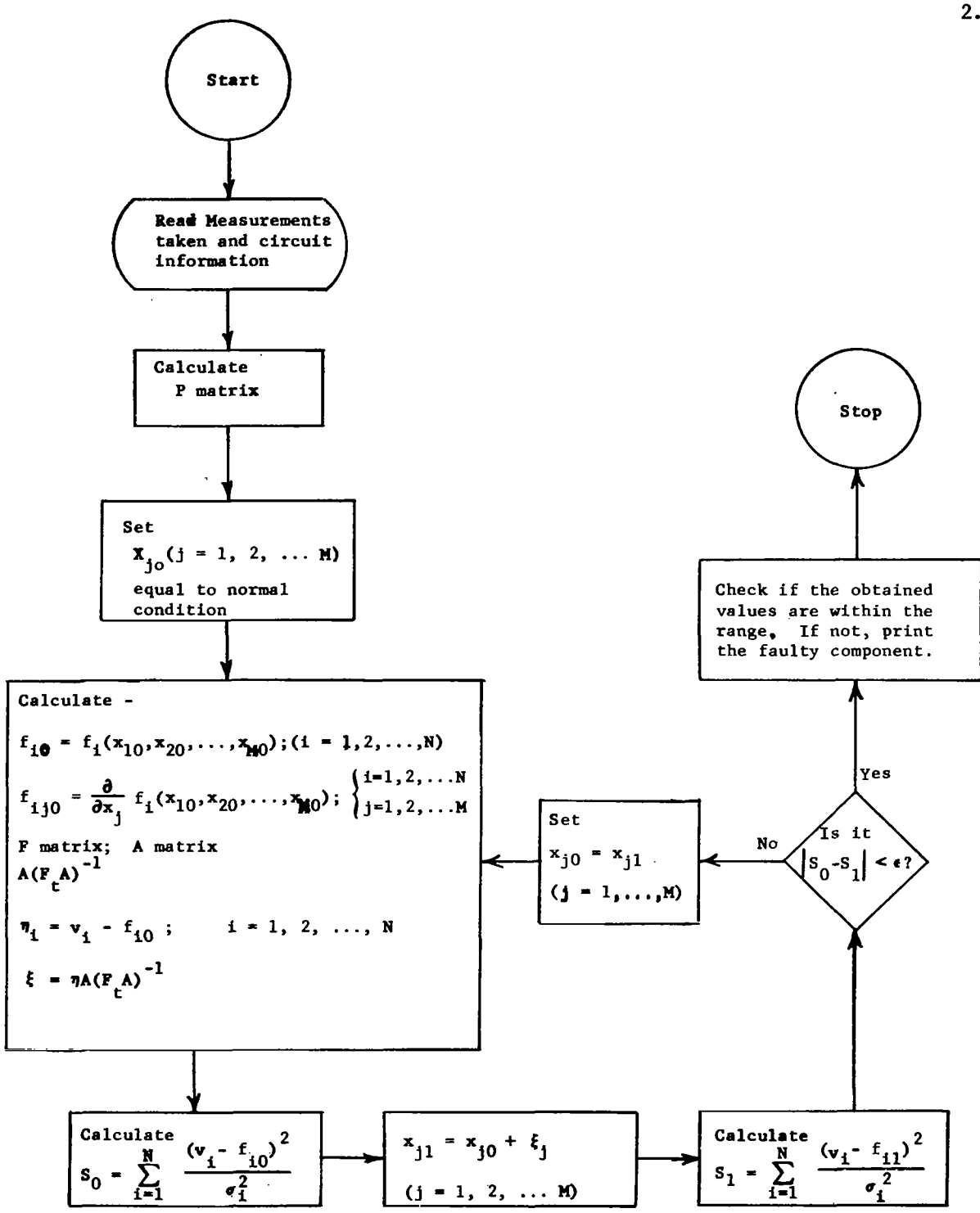


Figure 2.13 Iterative Method

The flow chart is indicated in figure 2.13.

2.3.1.1 Example

Let us consider the circuit shown 2.14.

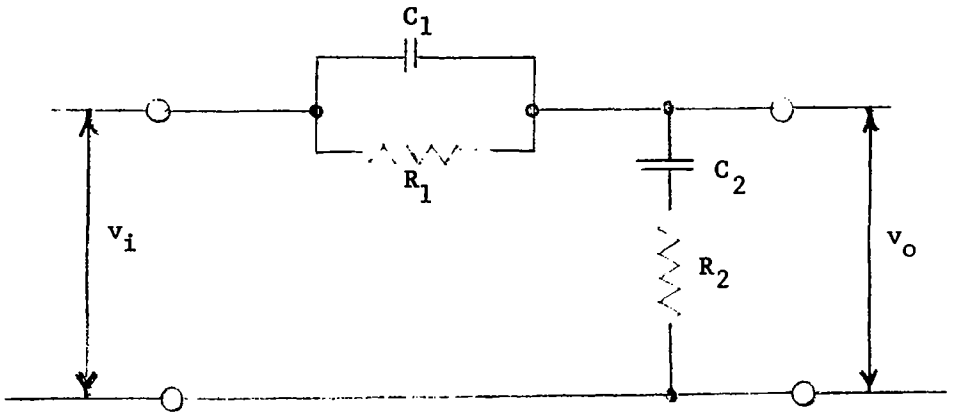


Figure 2.14 - Lead-Lag Network

The oupput/input relation is given by

$$\frac{v_o}{v_i} = \frac{\sqrt{R_2^2 + \left(\frac{1}{wC_2}\right)^2}}{\sqrt{\left[\frac{R_1}{1 + w^2C_1^2R_1^2} + R_2\right]^2 + \left[\frac{wC_1R_1^2}{1 + w^2C_1^2R_1^2} + \frac{1}{wC_2}\right]^2}} \qquad (2.37)$$

In order to ne able to check the result obtained with the computer run, Table IV gives the values of v_o/v_i for different combinations of R_1, C_1, R_2, C_2, and frequency.

Table IV

R_1	C_1	R_2	C_2	V_o/V_i at f_1 = 230	f_2 = 1,020	f_3 = 2,120	f_4 = 5,860	f_5 = 12,300
10,000	.0060	5,000	.020	0.90865	0.52429	0.44460	0.61463	0.82914
11,000	.0060	5,000	.020	0.89579	0.50005	0.43109	0.61571	0.83272
12,000	.0060	5,000	.020	0.88262	0.47867	0.41999	0.61709	0.83586
10,000	.0060	5,500	.020	0.90597	0.52840	0.46049	0.64141	0.84780
10,000	.0060	6,000	.020	0.90336	0.53313	0.47613	0.66545	0.86332
10,000	.0066	5,000	.020	0.90695	0.52503	0.45462	0.64084	0.84900
10,000	.0072	5,000	.020	0.90528	0.52619	0.46501	0.66486	0.86555
10,000	.0060	5,000	.022	0.89475	0.50348	0.43690	0.61639	0.83151
10,000	.0060	5,000	.024	0.88052	0.48577	0.43090	0.61807	0.83353
11,000	.0060	5,000	.024	0.86472	0.46246	0.41787	0.61942	0.83723
11,000	.0066	5,000	.024	0.86267	0.46443	0.42952	0.64684	0.85679
11,000	.0066	5,500	.024	0.85931	0.47153	0.44799	0.67398	0.87335
12,100	.0066	5,500	.024	0.84143	0.45018	0.43773	0.67670	0.87653
12,100	.0072	5,500	.024	0.83904	0.45388	0.45264	0.70385	0.89231

The results from the computer run are as follows

First Case -

Frequencies	Measurements	Standard Deviation
0.23	0.8826	0.43
1.02	0.4786	0.43
2.12	0.4199	0.43
5.86	0.6170	0.43
12.30	0.8358	0.43

Components	Values		
	Normal	Predicted	Real
R_1	.01	.0127	.012
C_1	.006	.00567	.006
R_2	.005	.00529	.005
C_2	.02	.0189	.02

Second Case -

Frequencies	Measurements	Standard Deviation
0.23	0.9052	0.43
1.02	0.5261	0.43
2.12	0.4650	0.43
5.86	0.6648	0.43
12.30	0.8655	0.43

Component	Values		
	Normal	Predicted	Real
R_1	.01	.0105	.01
C_1	.006	.00684	.0072
R_2	.005	.0052	.005
C_2	.02	.0191	.02

In both cases, the results obtained are satisfactory.

2.3.2 Conclusions -

This method gives a good approach in the solution of the fault isolation problem, but there are some difficulties in the matrix inversion process. In the second case of the example given above, the matrix $F_t A$ has the following values

$$\begin{bmatrix} .4396 \times 10^{-8} & -.1044 \times 10^{3} & -.1237 \times 10^{-8} & .2200 \times 10^{4} \\ -.1044 \times 10^{3} & .1520 \times 10^{17} & .2006 \times 10^{5} & .4611 \times 10^{14} \\ -.1237 \times 10^{-8} & .2006 \times 10^{5} & .2697 \times 10^{-7} & -.4371 \times 10^{3} \\ .2200 \times 10^{4} & .4611 \times 10^{14} & -.4371 \times 10^{3} & .1114 \times 10^{16} \end{bmatrix}$$

Therefore, using the inversion matrix subroutine that is included in the IBM/360 math-pack, we get the following inverse matrix

$$\begin{bmatrix} .5478 \times 10^{12} & -.2925 & .2267 \times 10^{12} & -.9806 \\ -.2925 & .1625 \times 10^{-12} & -.1258 & .5215 \times 10^{-12} \\ .2267 \times 10^{12} & -.1258 & .9745 \times 10^{11} & -.4043 \\ -.9806 & .5215 \times 10^{-12} & -.4043 & .1757 \times 10^{-11} \end{bmatrix}$$

The product of the two given matrices are the following

$$\begin{bmatrix} .1000 \times 10^{1} & .1387 \times 10^{-16} & .0000 & -.5551 \times 10^{-6} \\ .4456 \times 10^{8} & .9999 & .1520 \times 10^{8} & -.9536 \times 10^{-5} \\ .5722 \times 10^{-4} & -.3469 \times 10^{-17} & .1000 \times 10^{1} & -.6938 \times 10^{-16} \\ .1677 \times 10^{8} & -.7629 \times 10^{-5} & .4194 \times 10^{7} & .9999 \end{bmatrix}$$

Therefore, some improvement in the matrix inversion subroutine is needed.

2.3.3 Applications - The application of this method is for linear circuits at component level.

2.3.4 Computer Requirements - Since complex calculations are involved in the process of getting the predicted values of the circuit components, there are some requirements imposed over the computer system needed.

2.4 Transfer Function Method

2.4.1 Bode Diagram Technique [60]

The idea of this technique is to apply at the input of the circuit under test, signals of different frequencies (test frequencies), and for each one of them to measure the gain (output/input). These gains are checked against the normal gains. The difference between both sets of gains permits diagnosing which component is out of the normal range.

The principal advantages of this method could be summarized as follows:

- Use only available input and output terminals
- For diagnosis, it requires simple calculations
- Identification of the failure at component level.

In order to be able to determine the failure of the circuit at component level, it is necessary to prepare for each circuit, before the test is carried out, a "fault dictionary." This dictionary is a list of all the possible components variations (i.e., R_1 low, R_1 high, etc.), and the associated variations in the gain at some frequency (or frequencies). When this is done, we assign a "+" or "-" if the circuit gain was above or below the normal range, and zero if it was in the range. The fault dictionary looks like the one indicated in Table V (for a given circuit).

Table V Fault Dictionary

Gain at			Fault
f_1	f_2	f_3	
0	0	0	Normal
+	+	0	C_1 low
-	-	0	C_1 high
0	+	+	C_2 low
0	-	-	C_2 high
+	0	0	C_3 low
-	0	0	C_3 high
0	0	-	R_1 low
0	0	+	R_1 high

Now if we have this circuit under test, and we take the gains at frequencies f_1, f_2, f_3, we are going to come out with some sequence of "+", "-", or "0". By checking this sequence in the above table, we know which component is out of the normal range, i.e., if the sequence is +00, this means that the value of C_3 is low.

This method does not require post calculations, because after the measure is taken, we must look only for the row that matches with it. When the match is obtained, we know which is the fault component.

Our circuit under test has the configuration indicated in Figure 2.15.

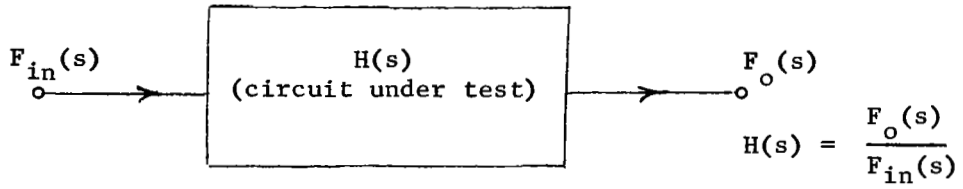


Figure 2.15

where

$F_{in}(s)$ = Laplace transform of the input function

$F_o(s)$ = Laplace transform of the output function

$H(s)$ = Transfer function

For lumped linear circuits, the transfer function H(s) may be expressed as a ratio of two polynomials, or

$$H(s) = \frac{A(s)}{B(s)} = K \frac{(s - s_1)(s - s_2) \ldots (s - s_n)}{s^m(s - s_a)(s - s_b) \ldots (s - s_p)} \qquad (2.38)$$

where

n = number of zeros of the transfer function

m+p = number of poles of the transfer function

K = gain constant

Now, if $s = jw$

$$H(jw) = K \frac{(jw - s_1)(jw - s_2) \ldots (jw - s_n)}{(jw)^m (jw - s_a)(jw - s_b)\ldots(jw - s_p)} \tag{2.39}$$

Let w_j designate the magnitude of s_j. At very low frequencies (for all $w < w_{min}$, where w_{min} is the lower value of w_j), the gain is given by

$$20 \log_{10} \left| H(jw) \right| = 20 \log_{10} K + 20 \sum_{i=1}^{n} \log_{10} w_i - 20 \sum_{i=1}^{p} \log_{10} w_i - 20 \log_{10} w^m \tag{2.40}$$

From the above equation we can see that any change in the components value is going to change the gain at low frequency, but there is no way to know, looking only at this very low frequency, which component is responsible for the change.

Now, if in equation (2.40) we have N parameters, we need to apply this equation for N different frequencies, and by means of conventional techniques to find the N parameters value. By an optimal selection of the test frequencies, it is possible to find the parameters value in a very simple way. In order to choose the optimum frequencies, we must remember the following:

- Each pole or zero (w_i) makes a breakpoint in the gain characteristic, the change in the slope is 20 db/decade (- if it a pole, and + if it is a zero). This value is obtained as follows:

  The contribution of each pole is given by

$$e_1 = \frac{1}{1 + jw_1T} \tag{2.41}$$

  Therefore,

$$\frac{e_1}{e_2} = \frac{1 + jw_2T}{1 + jw_1T} \tag{2.42}$$

if w_1T and w_2T are both > 1, we can make the following approximation

$$\frac{e_1}{e_2} = \frac{w_2}{w_1} \tag{2.43}$$

Then

$$20 \log_{10} \frac{e_1}{e_2} = 20 \log_{10} \frac{w_2}{w_1} \tag{2.44}$$

if in equation (2.44) $w_2 = 10w_1$, we have

$$20 \log_{10} \frac{e_1}{e_2} = 20 \text{ db/decade} \tag{2.45}$$

- Each pair of complex conjugated poles or zeros make a breakpoint in the gain characteristic. The change in the slope is 40 db/dec (- if it is a pole, and + if it is a zero). The gain in the breakpoint may vary from -6 db to + 20 db in function of the decrement factor ($1.0 \leq \xi \leq 0.05$).

Taking in consideration the above items, the test frequencies are chosen as follows:

- One test frequency must be below the lowest breakpoint.
- One test frequency must be above the highest breakpoint.
- One test frequency in between breakpoints.
- One test frequency for each breakpoint that belongs to the complex conjugated values. This must be on it or very close to it.

In order to apply this concept let us take the circuit shown in figure 2.16.

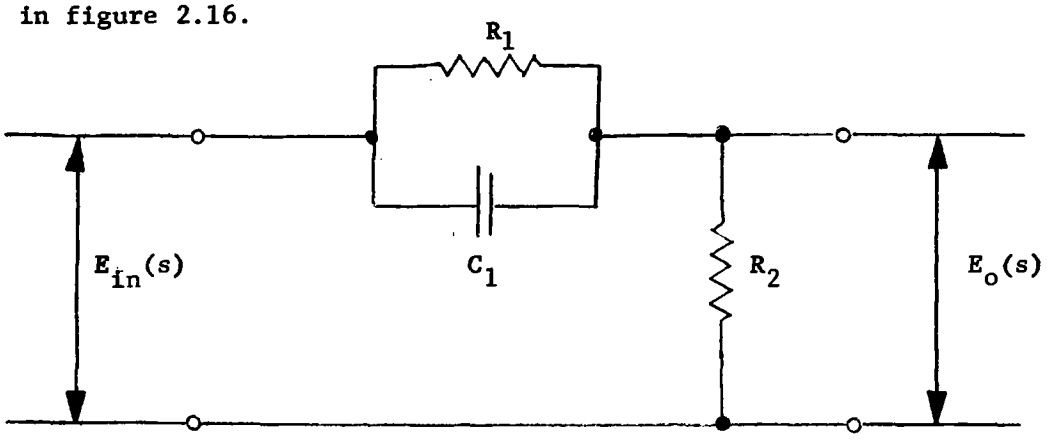


Figure 2.16 Lead Network

The transfer function of this circuit is given by

$$\frac{E_o(s)}{E_{in}(s)} = \frac{T_2}{T_1} \quad \frac{1 + sT_1}{1 + sT_2} \tag{2.46}$$

where

$$T_1 = C_1 R_1$$

$$T_2 = \frac{R_2}{R_1 + R_2} \, T_1$$

The Bode diagram (straight approximation) that belongs to this circuit is indicated in figure 2.17. Also, in the same figure we can see the three test frequencies that were taken following the given rules.

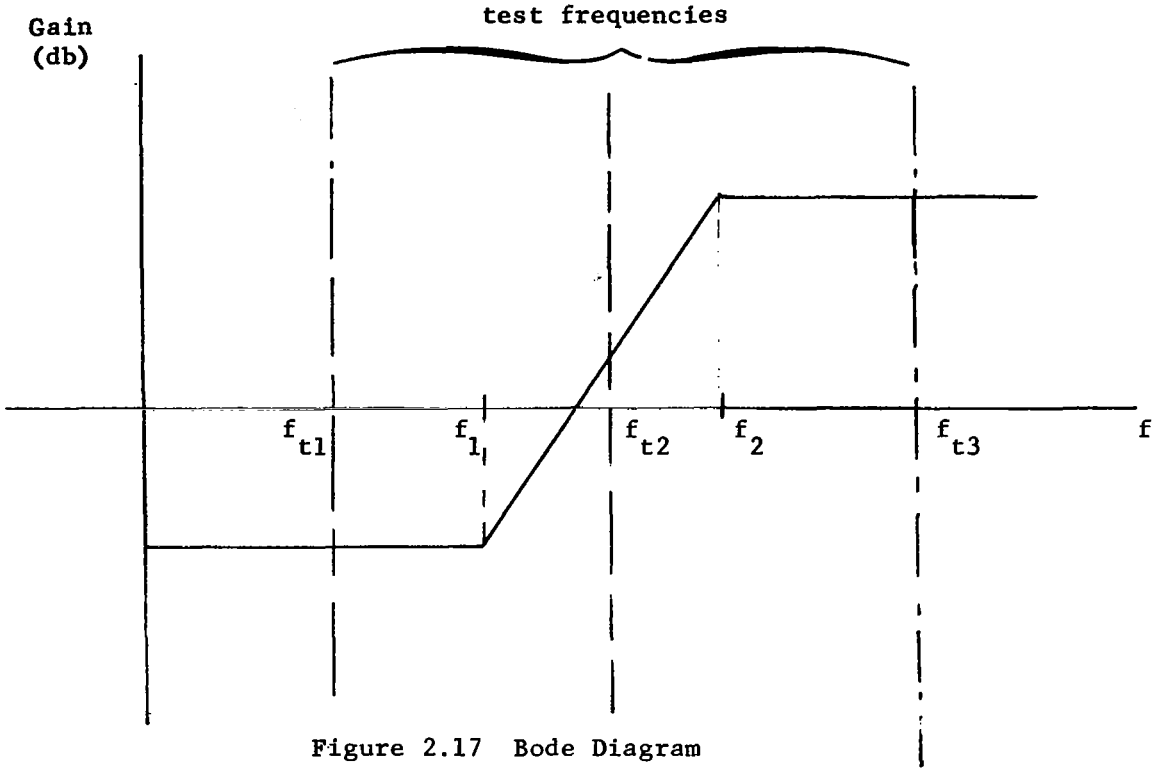


Figure 2.17 Bode Diagram

In figure 2.18 the influence on the Bode diagram by changes in the parameters value is indicated. Therefore, the fault dictionary given in Table VI is obtained.

Table VI Fault Dictionary

Gain at			Fault
f_{t1}	f_{t2}	f_{t3}	
0	0	0	Normal
+	0	+	R_1 low
-	0	-	R_1 high
-	-	0	R_2 low
+	+	0	R_2 high
0	+	0	C_1 low
0	-	0	C_1 high

Nonlinear circuits cannot be treated in a general way. For each type of nonlinear circuit, it is possible to find a general solution which can be linearly approximated. After this is done, we can apply the technique given for linear circuits. Of course, that same constraint will be applied to the linear approximation given as well as the general solution.

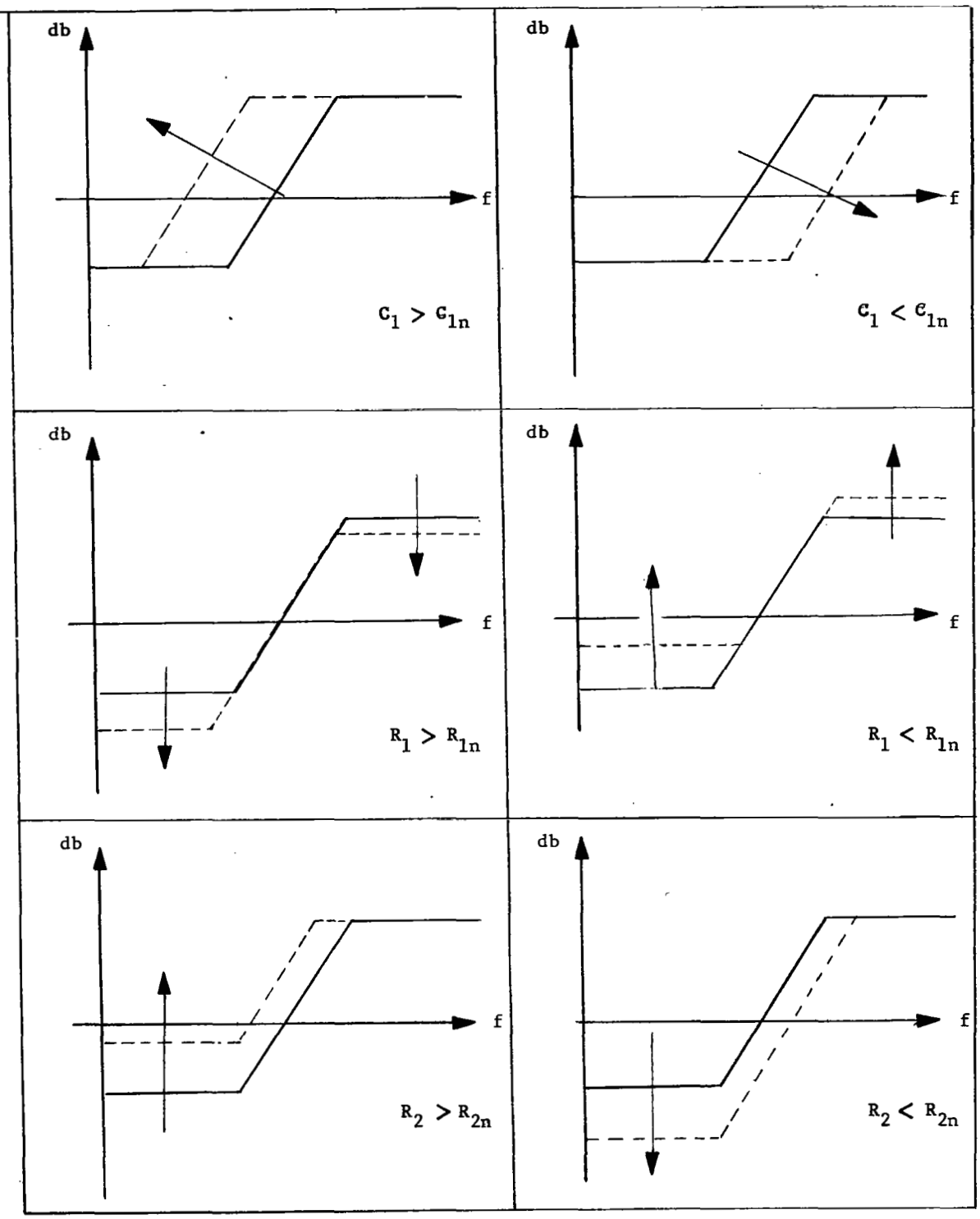


Figure 2.18 Influence of Changes in Parameters Value

Until now, we looked at this method from a theoretical approach, let us see what are the results in a practical application. To do this, let us take the following example.

2.4.1.1 Example - For the circuit shown in figure 2.14, we want to find the fault dictionary.

Let

R_1 = 0.01 Megohm

R_2 = 0.005 Megohm

C_1 = 0.006 Microfarad

C_2 = 0.02 Microfarad

Then

f_1 = 0.46 Kc/sec

f_2 = 1.59 Kc/sec

f_3 = 2.65 Kc/sec

f_4 = 9.08 Kc/sec

and the test frequencies are

f_{t1} = 0.23 Kc/sec

f_{t2} = 1.02 Kc/sec

f_{t3} = 2.12 Kc/sec

f_{t4} = 5.86 Kc/sec

f_{t5} = 12.30 Kc/sec

The flow chart that we need to perform the calculations is indicated in figure 2.19. The steps are self-explanatory, therefore, there is no need to make further comments about them. The results obtained are indicated in tables VII and VIII; from these we arrive at the following conclusions

- The sequence of gains which is associated with a faulty component are function not only of the component itself, but also of the variation of it and the zero detecting (the latter in our case was: normal condition $\pm$0.003). See Table VIII.

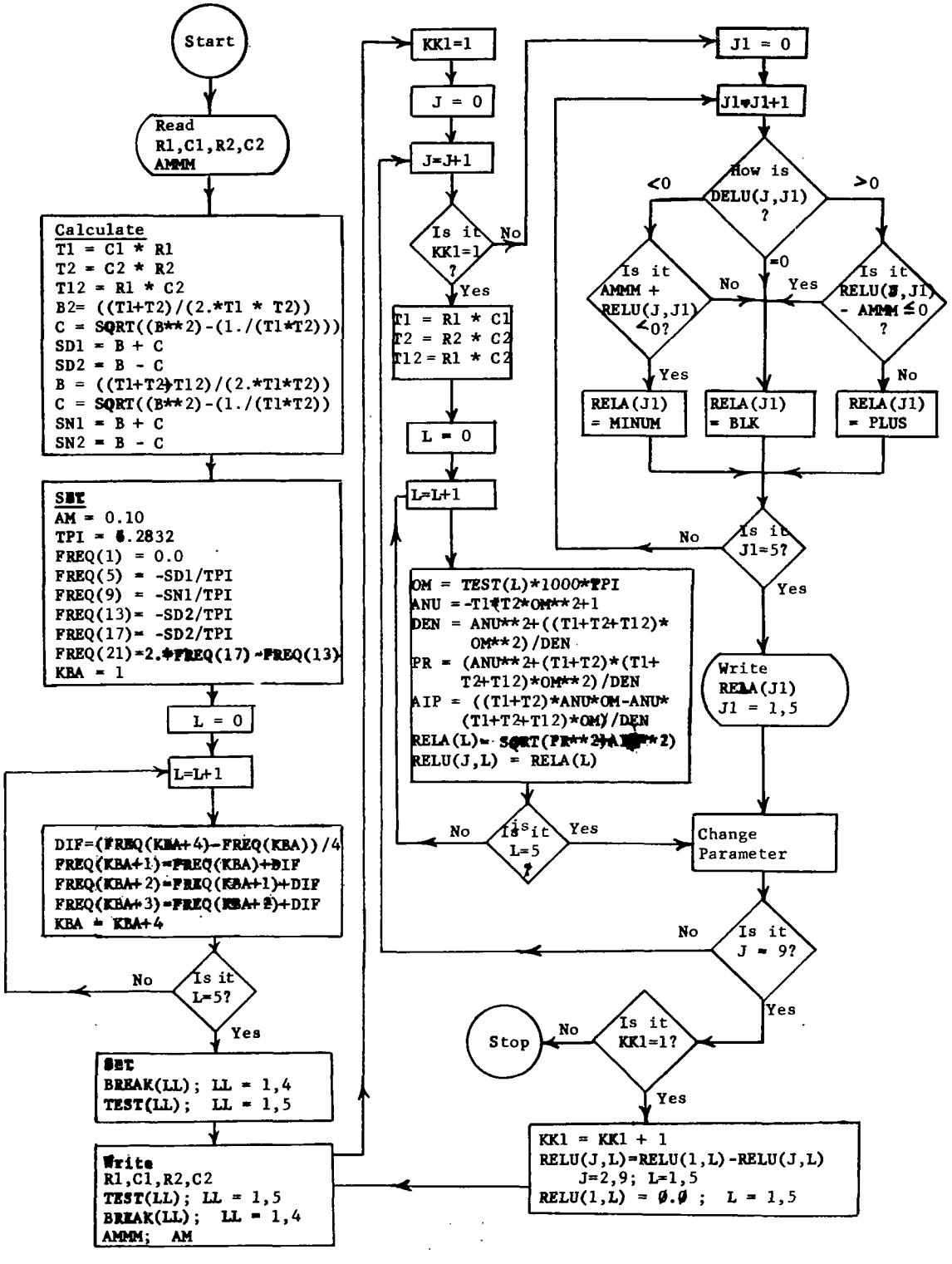


Figure 2.19 Bode Diagram Technique

TABLE VII

GAIN 1	GAIN 2	GAIN 3	GAIN 4	GAIN 5	CONDITION	ΔV
0.9070	0.5225	0.4446	0.6150	0.8291	NORMAL	
0.9197	0.5500	0.4611	0.6145	0.8250	R_1 LOW	
0.8940	0.4984	0.4311	0.6161	0.8327	R_1 HIGH	
0.9098	0.5190	0.4286	0.5852	0.8064	R_2 LOW	
0.9043	0.5267	0.4605	0.6418	0.8478	R_2 HIGH	± 10%
0.9088	0.5222	0.4350	0.5865	0.8051	C_1 LOW	
0.9036	0.5245	0.4650	0.6653	0.8655	C_1 HIGH	
0.9207	0.5469	0.4546	0.6132	0.8263	C_2 LOW	
0.8944	0.5038	0.4376	0.6166	0.8313	C_2 HIGH	
0.9070	0.5225	0.4446	0.6150	0.8291	NORMAL	
0.9318	0.5812	0.4815	0.6149	0.8203	R_1 LOW	
0.8806	0.4771	0.4200	0.6175	0.8358	R_1 HIGH	
0.9127	0.5164	0.4128	0.5519	0.7786	R_2 LOW	
0.9017	0.5316	0.4761	0.6658	0.8633	R_2 HIGH	± 20%
0.9105	0.5223	0.4262	0.5559	0.7759	C_1 LOW	
0.9036	0.5245	0.4650	0.6653	0.8655	C_1 HIGH	
0.9337	0.5758	0.4677	0.6115	0.8230	C_2 LOW	
0.8843	0.4910	0.4331	0.6178	0.8327	C_2 HIGH	
0.9070	0.5225	0.4446	0.6150	0.8291	NORMAL	
0.9434	0.6168	0.5068	0.6169	0.8148	R_1 LOW	
0.8671	0.4583	0.4108	0.6191	0.8386	R_1 HIGH	
0.9157	0.5149	0.3974	0.5150	0.7439	R_2 LOW	
0.8991	0.5369	0.4914	0.6874	0.8763	R_2 HIGH	± 30%
0.9123	0.5230	0.4182	0.5231	0.7401	C_1 LOW	
0.9036	0.5245	0.4650	0.6653	0.8655	C_1 HIGH	
0.9460	0.6100	0.4853	0.6101	0.8189	C_2 LOW	
0.8771	0.4827	0.4304	0.6186	0.8337	C_2 HIGH	

TABLE VIII

Fault Dictionary — Zero Detection — Normal Value ±0.003

GAIN 1	GAIN 2	GAIN 3	GAIN 4	GAIN 5	CONDITION	% FROM	% TO
0	0	0	0	0	NORMAL	N/A	N/A
+	+	+	0	-	R_1 LOW	0	30
+	+	+	+	-	R_1 LOW	30	40
-	-	-	0	+	R_1 HIGH	0	20
-	-	-	+	+	R_1 HIGH	20	40
0	-	-	-	-	R_2 LOW	0	10
+	-	-	-	-	R_2 LOW	10	40
0	+	+	+	+	R_2 HIGH	0	10
-	+	+	+	+	R_2 HIGH	10	40
0	0	-	-	-	C_1 LOW	0	10
+	0	-	-	-	C_1 LOW	10	40
-	0	+	+	+	C_1 HIGH	0	40
+	+	+	0	0	C_2 LOW	0	10
+	+	+	-	-	C_2 LOW	10	40
-	-	-	0	0	C_2 HIGH	0	10
-	-	-	0	+	C_2 HIGH	10	20
-	-	-	+	+	C_2 HIGH	20	40

- In some cases it is possible to detect the component; in others, we have the same sequence of gains for more than one component. See Table VIII: C_2 high 20 to 40%, and R_1 high 20 to 40%.

- All possible components variations must be simulated in order to arrive at a complete fault dictionary.

2.4.1.2 Diagnosis Discrimination -

In the assignment that we made like "+" or "-" in the gains, we get the faulty component and the range of the variation, i.e., if the sequence is

+, +, +, -, -

we know that the faulty component is C_2 and the value is between 10 to 40% low. But it is possible to get a little more information. A very easy way to do it is to make a scale for some frequency, like the one shown on figure 2.20. The frequency chosen must be the one that has the maximum variation, i.e., to detect variation in R_1, the second frequency will be convenient. See table VII.

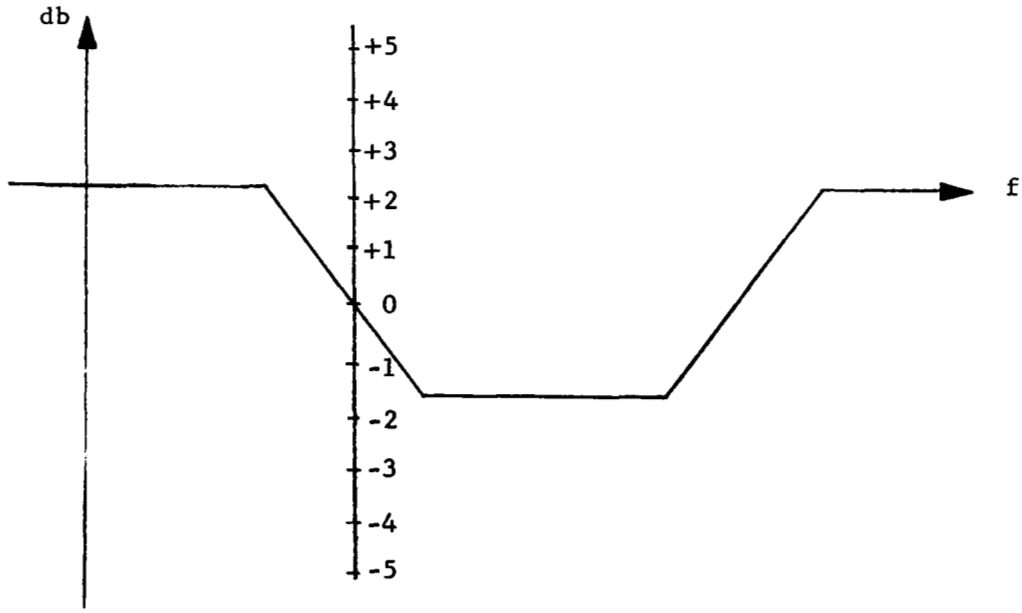


Figure 2.20

2.4.1.3 Conclusions -

The principal feature of this method is that this does not require post calculations. After the measurements are already taken, the only thing that remains is to look into the fault dictionary table where the results are located, and pick up the one that matches.

The fault dictionary could be generated by two methods: empirical and analytical. Improvement could be made in the determination of networks poles and zeros.

2.4.1.4 Applications -

The extension of this method to include systems instead of circuits is straight-forward and, in general, one can assume that a system is somewhat simpler than a circuit.

2.4.1.5 Computer Requirements -

Since there are no post-calculations involved in this method, no special computer requirements are needed.

2.4.2 Tracking Technique [58], [66], [67]

2.4.2.1 General

The basic idea of this method is to produce the tracking of the transfer function continuously, and without any kind of interruptions of the normal operation of the system under test. By tracking of the transfer function is understood the determination of the coefficients that belong to the dynamical equation of the system in a continuous way. Knowing these coefficients, it is possible to check if these are or are not within the normal operation range; therefore, for any time we know what is the status of the system under test. Since it is possible to know the values of the coefficients for any desirable t, it is very easy to perform trend analysis. If a diagnosis to component level is required, this will be carried out by means of the mathematical relationship between the coefficients of the dynamical equation and the values of the components.

A bounded exponential integral and a running leftside Laplace transform are defined, and a simple low pass filter is given in order to find them.

The Laplace Transform of a function f(t) is defined by

$$F(s) = \int_0^\infty f(t)\, e^{-st} dt \qquad (2.47)$$

In equation (2.47) we can see that the integration limits are zero and infinite, or in other words, that we are taking all the information that is possible to get from the function f(t). We assume that f(t) = 0, for $t < 0$.

We can also define the transform of the function f(t) only in the interval t_1 - t_2 as follows

$$F(s,\ t_1 \leq t \leq t_2) = \int_{t_1}^{t_2} f(t)e^{-st}dt \qquad (2.48)$$

The integral as defined in the above equation, provided that t_1 and t_2 are finite, is called the Bounded Exponential Integral.[66]

Now suppose that we want to know the bounded exponential integral of $\frac{df(t)}{dt}$, then

$$\int_{t_1}^{t_2} \frac{df(t)}{dt}\ e^{-st}dt = f(t)e^{-st}\Big|_{t_1}^{t_2} + s\int_{t_1}^{t_2} f(t)e^{-st}dt \qquad (2.49)$$

The above equation could be written in the following way

$$\int_{t_1}^{t_2} \frac{df(t)}{dt}\ e^{-st}dt = sF(s, t_1 \leq t \leq t_2) + f(t_2)e^{-st_2} - f(t_1)e^{-st_1} \qquad (2.50)$$

In equation (2.50) $f(t_2)e^{-st_2}$ is the terminal condition and $f_1(t)e^{-st_1}$ is the initial condition. We arrive at the important property that it is possible to operate on a function over the bounded region, provided that the initial conditions at t_1 and the terminal conditions at t_2 are known.

In general, for a function over a bound region in time, transformation of the function and its derivatives gives

$$f(t) \quad \Rightarrow F(s,\ t_1 \leq t \leq t_2)$$

$$\frac{df(t)}{dt} \Rightarrow s\,F(s,\ t_1 \leq t \leq t_2) + f(t_2)e^{-st_2} - f(t_1)e^{-st_1}$$

$$\frac{d^2f(t)}{dt^2} \Rightarrow s^2\,F(s,\ t_1 \leq t \leq t_2) + \left[sf(t_2) + \frac{df(t_2)}{dt}\right] e^{-st_2} - \left[sf(t_1) + \frac{df(t_1)}{dt}\right] e^{-st_1}$$

and in general

$$\frac{d^nf(t)}{dt^n} \Rightarrow s^n\,F(s,\ t_1 \leq t \leq t_2) + \left[\sum_{i=0}^{n-1} s^{n-1-i}\ \frac{d^if(t_2)}{dt^i}\right] e^{-st_2} - \left[\sum_{i=0}^{n-1} s^{n-1-i}\ \frac{d^if(t_1)}{dt^i}\right] e^{-st_1} \qquad (2.51)$$

If in the above equations, we place $t_1 = 0$ and $t_2 = \infty$, we get

$$f(t) \Rightarrow F(s)$$

$$\frac{df(t)}{dt} \Rightarrow sF(s) - f(0)$$

$$\frac{d^2f(t)}{dt^2} \Rightarrow s^2F(s) - sf(0) - \frac{df(0)}{dt}$$

and in general

$$\frac{d^nf(t)}{dt^n} \Rightarrow s^nF(s) - \sum_{i=0}^{n-1} s^{n-1-i}\ \frac{d^if(0)}{dt^i} \qquad (2.52)$$

that are the well-known expressions of the Laplace transforms of the derivatives.

Now, in order to make a more general Laplace transformation, let us go back to our definition given by (2.48). In this mentioned definition, we assume that the exponential kernel startsat t = 0 (see figure 2.21(a)); in the same we assume that s is real and

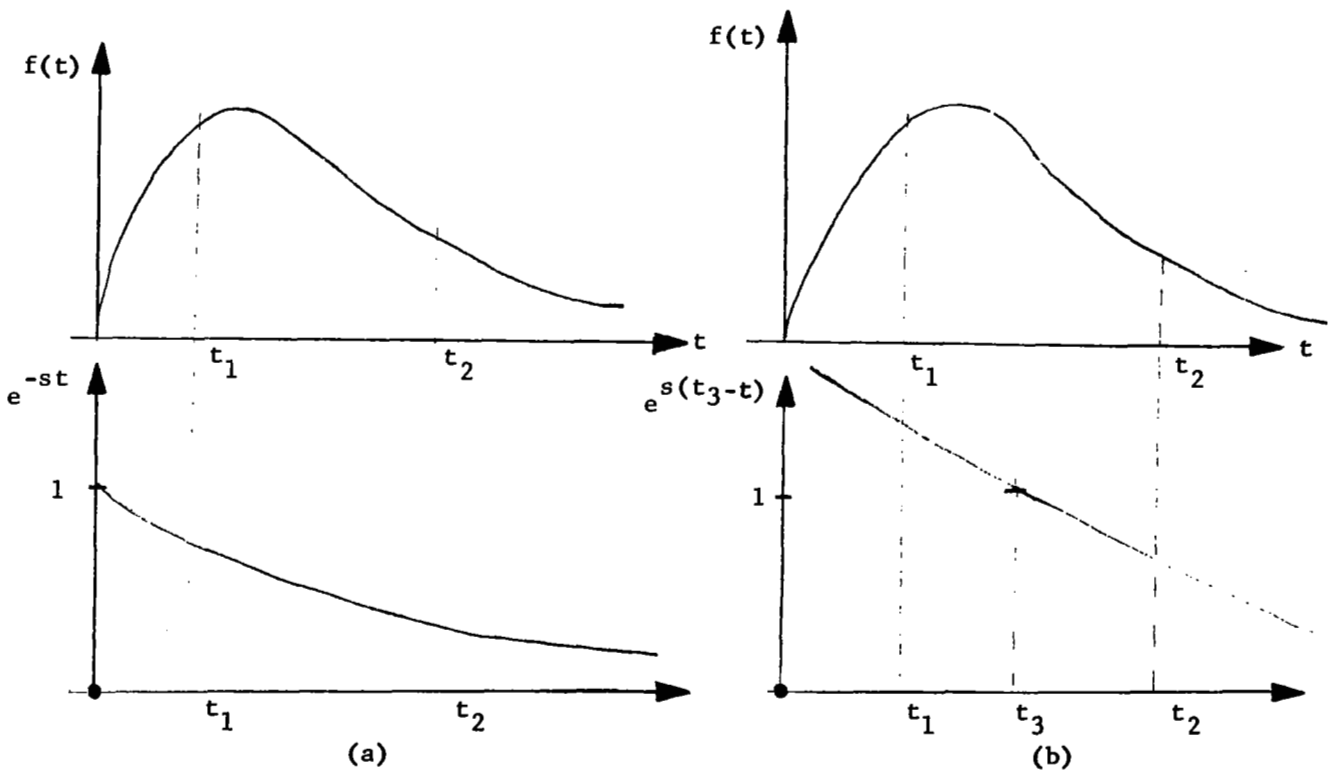


Figure 2.21

positive). Suppose now that our exponential kernel is equal to 1 for some t_3 not equal to zero (see figure 2.21(b)). With this modification the transform of the derivative is given by

$$\int_{t_1}^{t_2} \frac{df(t)}{dt} e^{s(t_3-t)} dt = f(t)\, e^{s(t_3-t)} \Big|_{t_1}^{t_2} + s \int_{t_1}^{t_2} f(t)\, e^{s(t_3-t)} dt$$

$$= s\, F(s,\ t_3,\ t_1 \leq t \leq t_2) + f(t_2)\, e^{s(t_3 - t_2)}$$

$$- f(t_1)\, e^{s(t_3-t_1)} \qquad (2.53)$$

and in general we can write

$$\frac{d^n f(t)}{dt^n} \Longrightarrow s^n\, F(s,\ t_3,\ t_1 \leq t \leq t_2) + \left[\sum_{i=0}^{n-1} s^{n-1-i}\ \frac{d^i f(t_2)}{dt^i}\right] e^{s(t_3-t_2)}$$

$$- \left[\sum_{i=0}^{n-1} s^{n-1-i}\ \frac{d^i f(t_1)}{dt^i}\right] e^{s(t_3-t_1)} \qquad (2.54)$$

That again, gives us the well-known expression of the Laplace transform of the derivatives if we make $t_3 = t_1 = 0$ and $t_2 = \infty$.

Now, let us take $t_2 = t_3 = t$ and $t_1 = t_s$, our equation (2.54) takes the following form

$$\frac{d^n f(t)}{dt^n} \Longrightarrow s^n\, F(s,t) + \sum_{i=0}^{n-1} s^{n-1-i}\ \frac{d^i f(t)}{dt^i}$$

$$- \left[\sum_{i=0}^{n-1} s^{n-1-i}\ \frac{d^i f(t_s)}{dt^i}\right] e^{s(t-t_s)} \qquad (2.55)$$

and our Laplace transformation is given by

$$F(s,t) = \int_{t_s}^{t} f(\nu)\, e^{s(t-\nu)}\, d\nu \qquad (2.56)$$

The name given to this transform is <u>running leftside Laplace transform</u>.

With the convolution theorem in mind, and looking at the equation (2.56), we arrive at the conclusion that F(s,t) will be the output of a system whose impulse response is given by e^{st} and has an input equal to f(t) (see figure 2.22).

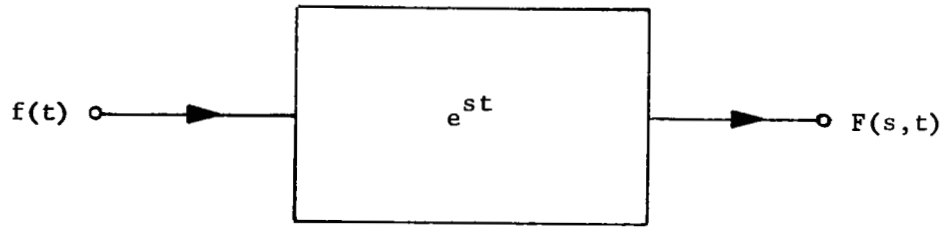


Figure 2.22

Now, let us take the low pass filter indicated in figure 2.23. The practical basis for the method is the discovery that a simple RC low-pass filter provides a running leftside Laplace transform.

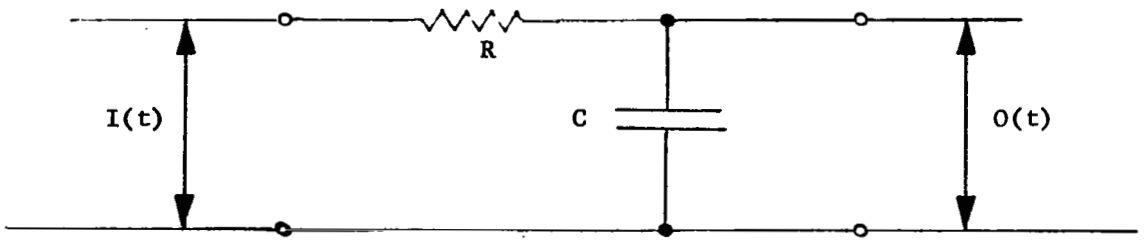


Figure 2.23

The impulse response of the filter is given by

$$h(t) = \frac{1}{RC} e^{-\frac{1}{RC}t} \tag{2.57}$$

Comparing the equation (2.57) with our hypothetical one (e^{st}), we see that $s = -\frac{1}{RC}$. Then

$$-s\,F(s,t) = -s \int_{t_s}^{t} f(\nu)\, e^{s(t-\nu)}\, d\nu \tag{2.58}$$

Our transformation filter gives us the outputs indicated in figure 2.24.

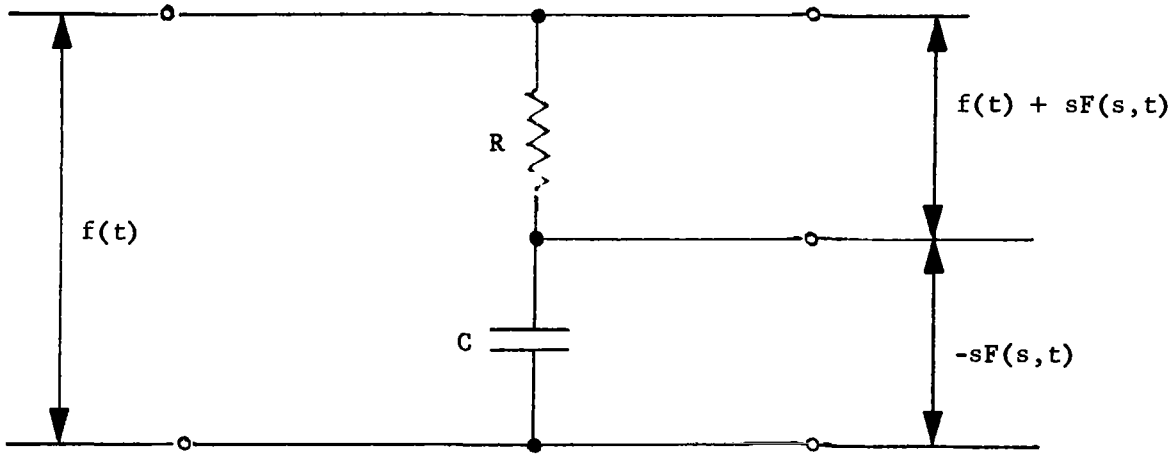


Figure 2.24

Now, we assume that our system is defined by means of the following linear differential equation.

$$\sum_{m=0}^{M} P_m \frac{d^m O(t)}{dt^m} = \sum_{n=0}^{N} Z_n \frac{d^n I(t)}{dt^n} \tag{2.59}$$

where

$O(t)$ = Output function

$I(t)$ = Input function

$\mathbb{P}_m$ = Coefficients of the part of the differential equation which determines the poles of the transfer function (we assume $\mathbb{P}_o = 1$)

$\mathbb{Z}_n$ = Coefficients which determine the zeros.

Since p_o is equal to 1, then z_o is the system gain at zero frequency.

A transform of equation (2.59) gives

$$\sum_{m=0}^{M} \mathbb{P}_m s^m \, O(s,t) + \sum_{m=1}^{M} \sum_{i=0}^{m-1} \mathbb{P}_m s^{m-1-i} \frac{d^i O(t)}{dt^i}$$

$$- \left[\sum_{m=1}^{M} \sum_{i=0}^{m-1} \mathbb{P}_m s^{m-1-i} \frac{d^i O(t_s)}{dt^i} \right] e^{s(t-t_s)} =$$

$$= \sum_{n=0}^{N} \mathbb{Z}_n s^n \, I(s,t) + \sum_{n=1}^{N} \sum_{i=0}^{n-1} \mathbb{Z}_n s^{n-1-i} \frac{d^i I(t)}{dt^i}$$

$$- \left[\sum_{n=1}^{N} \sum_{i=0}^{n-1} \mathbb{Z}_n s^{n-1-i} \frac{d^i I(t_s)}{dt^i} \right] e^{s(t-t_s)} \qquad (2.60)$$

The equation (2.60) is a new differential equation, the order of which is lowered by one due to the time functions in the terminal conditions. The order could be lowered to zero by transforming a sufficient number of times. As a result, no derivatives are required for the terminal conditions. Since s is a real negative quantity, the negative terms in both members of equation (2.60) becomes negligible after a sufficient waiting period from the beginning of the process. Therefore,

$$\sum_{m=0}^{M} P_m s^m O(s,t) + \sum_{m=1}^{M} \sum_{i=0}^{m-1} P_m s^{m-1-i} \frac{d^i O(t)}{dt^i} =$$

$$= \sum_{n=0}^{N} Z_n s^n I(s,t) + \sum_{m=1}^{N} \sum_{i=0}^{n-1} Z_n s^{n-1-i} \frac{d^i I(t)}{dt^i} \qquad (2.61)$$

In order to understand the application of this method, let us take the following

2.4.2.2 Example -

Let us assume that the dynamical equation of our system is given by

$$O(t) + P_1 \frac{dO(t)}{dt} + P_2 \frac{d^2 O(t)}{dt^2} = Z_0 I(t) + Z_1 \frac{dI(t)}{dt} \qquad (2.62)$$

Taking in consideration the equation (2.61), we can write the equation (2.62) as follows

$$O(s_{11},t)\left[1 + s_{11}P_1 + s_{11}^2 P_2\right] + O(t)\left[P_1 + s_{11}P_1\right] + \frac{dO(t)}{dt} P_2 =$$

$$= I(s_{11},t)\left[Z_0 + s_{11}Z_1\right] + I(t)\, Z_1 \qquad (2.63)$$

In order to produce the elimination of $\frac{dO(t)}{dt}$, we must make another transformation of equation (2.63). The result is as follows

$$O(s_{11}, s_{12}, t)\left[1 + s_{11}p_1 + s_{11}^2 p_2\right] + O(s_{12},t)\left[p_1 + s_{11}p_2 + s_{12}p_2\right] +$$

$$+ O(t)\, p_2 = I(s_{11}, s_{12}, t)\left[z_0 + s_{11}z_1\right] + I(s_{12}, t)\, z_1 \qquad (2.64)$$

Since we apply this technique to linear systems, the transformation process could be reversed. Therefore, we can transform for s_{12} first, and then s_{11}. We have seen that the values of s_{11} and s_{12} are given by the filter's constants $\left(s = -\frac{1}{RC}\right)$, therefore, in order to get the coefficients p_1, p_2, z_0 and z_1, four equations like (2.64) are needed. Our equations system takes the form

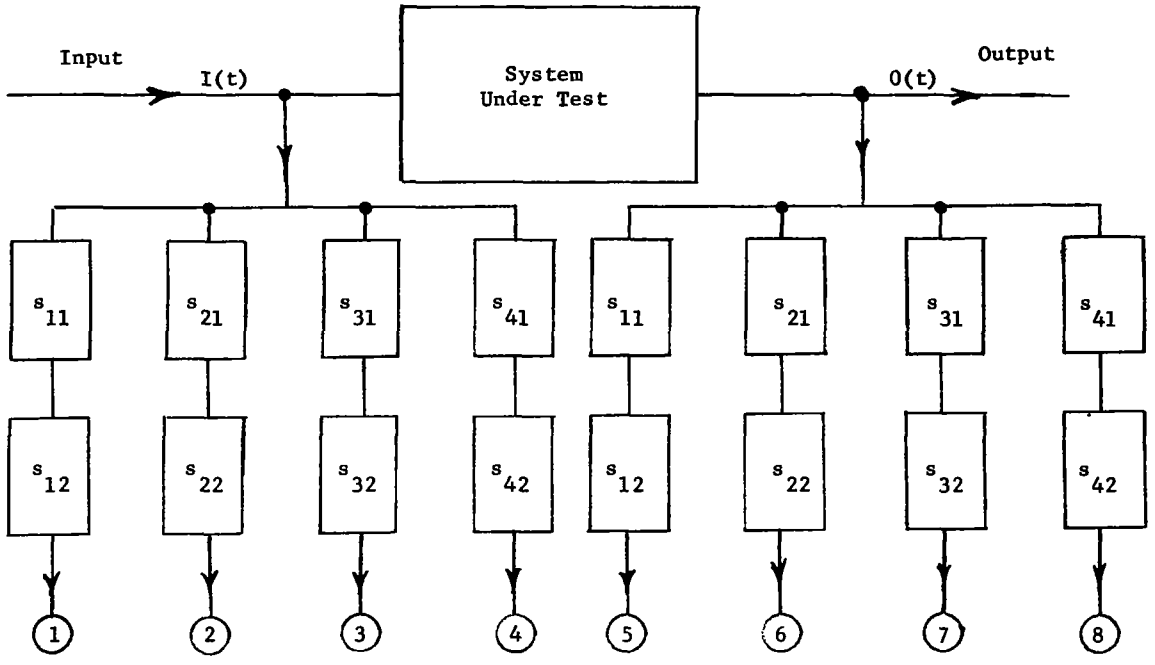
$$\begin{cases} O(s_{11},s_{12},t)\left[1 + s_{11}p_1 + s_{11}^2 p_2\right] + O(s_{12},t)\left[p_1 + s_{11}p_2 + s_{12}p_2\right] \\ \qquad + O(t)p_2 = I(s_{11},s_{12},t)\left[z_0 + s_{11}z_1\right] + I(s_{12},t)z_1 \\ O(s_{21},s_{22},t)\left[1 + s_{21}p_1 + s_{21}^2 p_2\right] + O(s_{22},t)\left[p_1 + s_{21}p_2 + s_{22}p_2\right] \\ \qquad + O(t)p_2 = I(s_{21},s_{22},t)\left[z_0 + s_{21}z_1\right] + I(s_{22},t)z_1 \\ O(s_{31},s_{32},t)\left[1 + s_{31}p_1 + s_{31}^2 p_2\right] + O(s_{32},t)\left[p_1 + s_{31}p_2 + s_{32}p_2\right] \\ \qquad + O(t)p_2 = I(s_{31},s_{32},t)\left[z_0 + s_{31}z_1\right] + I(s_{32},t)z_2 \\ O(s_{41},s_{42},t)\left[1 + s_{41}p_1 + s_{42}^2 p_2\right] + O(s_{42},t)\left[p_1 + s_{41}p_2 + s_{42}p_2\right] \\ \qquad + O(t)p_2 = I(s_{41},s_{42},t)\left[z_0 + s_{41}z_1\right] + I(s_{42},t)z_2 \end{cases} \qquad (2.65)$$

The values of p_1, p_2, z_0, and z_1 are given by

$$\begin{bmatrix} p_1 \\ p_2 \\ z_0 \\ z_1 \end{bmatrix} = \begin{bmatrix} -0(s_{11}, s_{12}, t) \\ -0(s_{21}, s_{22}, t) \\ -0(s_{31}, s_{32}, t) \\ -0(s_{41}, s_{42}, t) \end{bmatrix} \bullet A \qquad (2.66)$$

The matrix A is obtained through simple matrix manipulations.

The system configuration needed is shown in figure 2.25. We can see that we need sixteen transform filters in order to get the variables required.



#1 $s_{11}s_{12}\ I(s_{11},s_{12},t)$

#2 $s_{21}s_{22}\ I(s_{21},s_{22},t)$

#3 $s_{31}s_{32}\ I(s_{31},s_{32},t)$

#4 $s_{41}s_{42}\ I(s_{41},s_{42},t)$

#5 $s_{11}s_{12}\ O(s_{11},s_{12},t)$

#6 $s_{21}s_{22}\ O(s_{21},s_{22},t)$

#7 $s_{31}s_{32}\ O(s_{31},s_{32},t)$

#8 $s_{41}s_{42}\ O(s_{41},s_{42},t)$

Figure 2.25

By means of this method, we can track continuously the transfer function. We see from figure 2.25 that the values of the eight measurements are obtained without any interruption, and therefore by equation (2.66) we can get the values of $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{Z}_0$, $\mathbb{Z}_1$ at any moment that we want.

It is possible to make some reduction in the number of filters that are needed, see reference [67] .

2.4.2.3 Conclusions

The more important features of this method could be summarized as follows:

- It tracks transfer functions of systems continuously and without delay, while they are in operation.
- The possibility to perform trend analysis, without any further improvement of the system.
- There is no need of generation of special stimulus.
- Not only the mathematical manipulations of the method will be handled by the computer, also it is possible to perform the simulation of the filters.

2.4.2.4 Application

This method can be applied only in the linear case, circuits and systems.

2.4.2.5 Computer Requirements

The computer size required increases very fast with the complexity of the system under test. This increase is more strong and could place some limitations on its applicability, if also we need to make the simulation of the filters with the same computer.

2.5 Scalar Remnant Method (Dynamic System Testing)[5]

2.5.1 General

This method defines first a scalar performance indicator which will always be positive when the coefficients of the differential equation describing the system are within their prescribed tolerances; otherwise, it will be negative. This method does not indicate how many coefficients are out of their prescribed tolerances.

The scalar remnant function is formed from measurements of the input and output and their respective system derivatives as well as from the normal values of the system coefficients and their allowable deviations.

Therefore, in this method we have

- Indication that one or more coefficients of the differential equation describing the system are outside of the allowed range.
- Monitoring the scalar remnant with a zero-crossing detector provides an excellent system fault indicator.
- In function of frequency interruptions, it is possible to perform some trend analysis.
- The average value of the scalar remnant can indicate the degree to which tolerance limits are exceeded.

A dynamical equation describing a linear system of n^{th} order, has the following form

$$a_n \frac{d^n x}{dt^n} + a_{n-1} \frac{d^{n-1} x}{dt^{n-1}} + \ldots + a_1 x + a_0 = b_0 + b_1 u + \ldots + b_m \frac{d^m u}{dt^m} \qquad (2.67)$$

or

$$\sum_{i=0}^{n} a_i \frac{d^i x}{dt^i} = \sum_{i=0}^{m} b_i \frac{d^i u}{dt^i} \qquad (2.68)$$

If after some time, the coefficients of our system have been changed, we have

$$\sum_{i=0}^{n} a_i' \frac{d^i x}{dt^i} = \sum_{i=0}^{m} b_i' \frac{d^i u}{dt^i} \tag{2.69}$$

or, in function of the same variables x and u, the equation (2.68) becomes

$$\sum_{i=0}^{n} a_i \frac{d^i x'}{dt^i} = \sum_{i=0}^{m} b_i \frac{d^i u'}{dt^i} - \epsilon(t) \tag{2.70}$$

Subtracting equation (2.70) from (2.68) and denoting with x and u the variables x' and u' respectively, we have

$$\sum_{i=0}^{n} (a_i - a_i') \frac{d^i x}{dt^i} = \sum_{i=0}^{m} (b_i - b_i') \frac{d^i u}{dt^i} + \epsilon(t) \tag{2.71}$$

By introducing the notation $\Delta a_i = a_i - a_i'$; and $\Delta b_i = b_i - b_i'$ for the respective deviations of the coefficients, the equation (2.71) can be written

$$\sum_{i=0}^{n} \Delta a_i \frac{d^i x}{dt^i} - \sum_{i=0}^{m} \Delta b_i \frac{d^i u}{dt^i} = \epsilon(t) \tag{2.72}$$

The maximum error is given by

$$\text{Max } \epsilon(t) = \sum_{i=0}^{n} \text{Max} \left[\Delta a_i \frac{d^i x}{dt^i} \right] + \sum_{i=0}^{m} \text{Max} \left[- \Delta b_i \frac{d^i u}{dt^i} \right] \tag{2.73}$$

and in the same way, the minimum error is given by

$$\text{Min } \epsilon(t) = -\sum_{i=0}^{n} \text{Min}\left[-\Delta a_i \frac{d^i x}{dt^i}\right] - \sum_{i=0}^{m} \text{Min}\left[\Delta b_i \frac{d^i u}{dt^i}\right] \tag{2.74}$$

Now it must be determined at what point the magnitude of one or more of the deviations $|\Delta a_i|$ or $|\Delta b_i|$, exceeds its respective limit, δa_i or δb_i.

$$\begin{cases} + \delta a_i \dfrac{d^i x}{dt^i} ; \dfrac{d^i x}{dt^i} > 0 \\ - \delta a_i \dfrac{d^i x}{dt^i} ; \dfrac{d^i x}{dt^i} < 0 \end{cases} \tag{2.75}$$

$$\begin{cases} + \delta b_i \dfrac{d^i u}{dt^i} ; \dfrac{d^i u}{dt^i} > 0 \\ - \delta b_i \dfrac{d^i u}{dt^i} ; \dfrac{d^i u}{dt^i} < 0 \end{cases} \tag{2.76}$$

Therefore, the equation (2.73) becomes

$$\text{Max } \epsilon(t) = \sum_{i=0}^{n} \delta a_i \left|\frac{d^i x}{dt^i}\right| + \sum_{i=0}^{m} \delta b_i \left|\frac{d^i u}{dt^i}\right| \tag{2.77}$$

and, in the same way

$$\text{Min } \epsilon(t) = -\sum_{i=0}^{n} \delta a_i \left|\frac{d^i x}{dt^i}\right| - \sum_{i=0}^{m} \delta b_i \left|\frac{d^i u}{dt^i}\right| \tag{2.78}$$

but Max $\epsilon(t)$ = - Min $\epsilon(t)$, therefore when the coefficients are within tolerance

$$\left|\epsilon(t)\right| < \text{Max } \epsilon(t) \tag{2.79}$$

Then, our scalar remnant function is given by

$$\delta(t) = \text{Max } \epsilon(t) - \left|\epsilon(t)\right| \tag{2.80}$$

The condition $\delta(t) > 0$ is a necessary but not sufficient condition for all coefficients of the system to be within their tolerance limits. However, $\delta < 0$ is a sufficient condition for one or more coefficients to be out of tolerance.

Combining equations (2.70) and (2.77) with (2.80), we arrive at the following expression for $\delta(t)$

$$\delta(t) = \sum_{i=0}^{n} \delta a_i \left| \frac{d^i x}{dt^i} \right| + \sum_{i=0}^{m} \delta b_i \left| \frac{d^i u}{dt^i} \right|$$

$$- \left| \sum_{i=0}^{n} a_i \frac{d^i x}{dt^i} - \sum_{i=0}^{m} b_i \frac{d^i u}{dt^i} \right| \tag{2.81}$$

In the above equation δa_i and δb_i are the allowed tolerances for the coefficients a_i and b_i, respectively; and x and u are the measurements of the input and output, respectively.

Elimination of the derivatives involved in the equation (2.81) could be done following the technique given in the mentioned reference at the beginning of this method.

2.5.2 Example

In order to apply this method let us take the circuit shown in Figure 2.26.

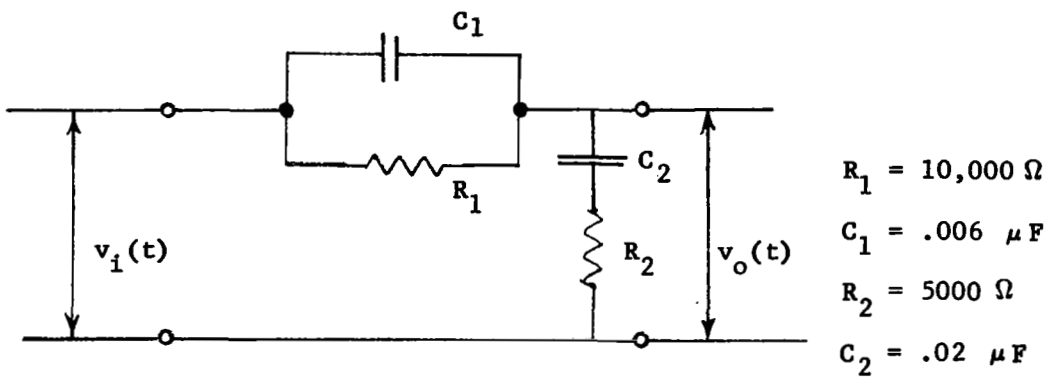


Figure 2.26 Lead-Lag Network

If we assume that the input are given by

$$v_i(t) = \text{Sin } \beta t \tag{2.82}$$

The first and second derivatives of equation (2.82) are given by

$$v_i'(t) = \beta \cos \beta t$$

$$v_i''(t) = -\beta^2 \sin \beta t$$

The transfer function of the Lead-Lag network is given by

$$\frac{V_o}{V_i}(s) = \frac{T_1T_2\, s^2 + (T_1 + T_2)\, s + 1}{T_1T_2\, s^2 + (T_1 + T_2 + T_{12})\, s + 1} \tag{2.83}$$

where

$$T_1 = R_1C_1$$

$$T_2 = R_2C_2$$

$$T_{12} = R_1C_2$$

Now, since the input function is given by equation (2.82), the equation (2.83) could be written as follows

$$V_o(s) = \frac{\beta}{s^2 + \beta^2} \quad \frac{T_1T_2\, s^2 + (T_1 + T_2)\, s + 1}{T_1T_2\, s^2 + (T_1 + T_2 + T_{12})\, s + 1}$$

$$= \frac{K_\beta}{s + j\beta} + \frac{K - \beta}{s - j\beta} + \frac{K_1}{s - \alpha_1} + \frac{K_2}{s - \alpha_2}$$

Therefore

$$\begin{cases} v_o(t) = 2\,|K_\beta| \cos(\beta t + \varphi) + K_1 e^{\alpha_1 t} + K_2 e^{\alpha_2 t} \\ v_o'(t) = -2\beta\,|K_\beta| \sin(\beta t + \varphi) + K_1\alpha_1 e^{\alpha_1 t} + K_2\alpha_2 e^{\alpha_2 t} \\ v_o''(t) = -2\beta^2\,|K_\beta| \cos(\beta t + \varphi) + K_1\alpha_1^2 e^{\alpha_1 t} + K_2\alpha_2^2 e^{\alpha_2 t} \end{cases}$$

where

$$K_1 = \left| \frac{\beta}{s^2 + \beta^2} \quad \frac{s^2 + \frac{T_1 + T_2}{T_2T_2}\, s + \frac{1}{T_1T_2}}{s - \alpha_2} \right|_{s = \alpha_1}$$

$$K_2 = \left| \frac{\beta}{s^2 + \beta^2} \quad \frac{s^2 + \frac{T_1 + T_2}{T_1 T_2} s + \frac{1}{T_1 T_2}}{s - \alpha_1} \right|_{s = \alpha_2}$$

$$K_\beta = \left| \frac{\beta}{s - j\beta} \quad \frac{T_1 T_2 s^2 + (T_1 + T_2)s + 1}{T_1 T_2 s^2 + (T_1 + T_2 + T_{12})s + 1} \right|_{s = -j\beta} = A + j\beta$$

$$\varphi = \text{tang}^{-1} \frac{B}{A}$$

The values of the coefficients as well as the allowed ranges are indicated in Table IX. These are obtained allowing a range of variation in each component of $\pm$25%.

TABLE IX

Coeff.	Normal Value	Lower Bound	Higher Bound	Percentage (%)
a_2	$.600 \times 10^{-8}$	$.450 \times 10^{-8}$	$.750 \times 10^{-8}$	25.0
a_1	$.360 \times 10^{-3}$	$.285 \times 10^{-3}$	$.435 \times 10^{-3}$	20.8
a_0	$.100 \times 10^{1}$	$.100 \times 10^{1}$	$.100 \times 10^{1}$	0.0
b_2	$.600 \times 10^{-8}$	$.450 \times 10^{-8}$	$.750 \times 10^{-8}$	25.0
b_1	$.160 \times 10^{-3}$	$.135 \times 10^{-3}$	$.185 \times 10^{-3}$	15.6
b_0	$.100 \times 10^{1}$	$.100 \times 10^{1}$	$.100 \times 10^{1}$	0.0

The output of the program, for different cases, looks like the following

Case	Scalar Remnant	Param. Status	Coefficients Within Normal Range
1	$.119 \times 10^{-5}$	+25% R_1	Yes
2	$-.370 \times 10^{-4}$	+150% R_1	No
3	$-.544 \times 10^{-4}$	+136% C_2	No
4	$.532 \times 10^{-4}$	-10% R_2	Yes
5	$.712 \times 10^{-5}$	+24.3% R_1	Yes
6	$-.108 \times 10^{-3}$	+200.0% C_2	No
7	$.725 \times 10^{-4}$	+100.0% C_1	No

2.5.3 Conclusions -

The results of the seven cases are satisfactory, except that of the seventh case. The reason for this complies with what was stated earlier, namely, that the condition $\delta(t) > 0$ is necessary but not a sufficient condition for all coefficients of the system to be within their tolerance limits. However, $\delta < 0$ is a sufficient condition for one or more coefficients to be out of tolerance.

2.5.4 Application -

The field of applicability of this method covers only the linear case of circuits and systems.

2.5.5 Computer Requirements -

Since there are only simple post-calculations involved in this method, no special computer requirements are needed.

2.6 Inverse Probability Method

2.6.1 General

The philosophy involved in this method is as followes: "Given m measurements that could be taken in m test points or less, find the difference between these values and the normal values at the same points. By means of the inverse probability, the probability that each component will be a faulty component is found by this difference. The component that has the highest probability associated with it will be the most probable faulty component."

If we assume that we have a circuit with 10 components, and after having taken the measurements and found the probability for each component, the following table is obtained.

TABLE X

COMPONENT	PROBABILITY
1	0.09875
2	0.11032
3	0.09923
4	0.12543
5	0.11342
6	0.09987
7	0.09996
8	0.12135
9	0.09345
10	0.03831

The most probable faulty component will be the 4th component because it has the highest probability associated to it.

The circuit under test has the configuration shown in figure 2.27.

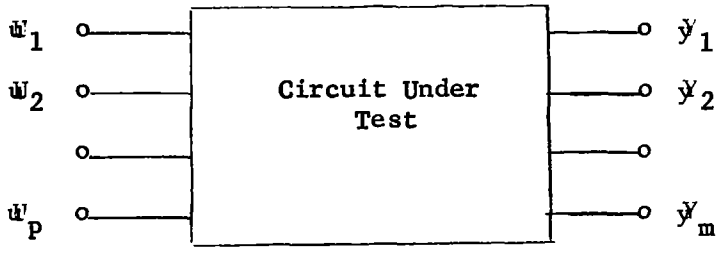


Figure 2.27

The circuit under test has n components that can fail. The number of components is greater than the number of measurements taken.

Let

$y_j = f_j(x_1, x_2, \ldots, x_n)$ = Theoretical expression for the j^{th} measurement.

x_i = actual value of the i^{th} component within the network.

X_i = nominal value of each component

n = number of components

m = number of measurements taken

By means of the Taylor series, it is possible to write

$$f_j(x_1,x_2,\ldots x_n) = f_j(X_1,X_2,\ldots X_n) + \left[(x_1-X_1)\frac{\partial}{\partial x_1}+\ldots+(x_n-X_n)\frac{\partial}{\partial x_n}\right] f_j(x_1,x_2,\ldots x_n)\Big|_{X_1,X_2,\ldots X_n} + \frac{1}{2!}\left[(x_1-X_1)\frac{\partial}{\partial x_1} + \ldots + (x_n-X_n)\frac{\partial}{\partial x_n}\right]^2 f_j(x_1,x_2,\ldots x_n)\Big|_{X_1,X_2,\ldots X_n} + \ldots \quad (2.84)$$

If we designate with $\Delta Y_j = f_j(x_1,x_2,\ldots x_n) - f_j(X_1,X_2,\ldots X_n)$ and with $\underline{x}$ and $\underline{X}$ the vectors $(x_1,x_2,\ldots x_n)$, and $(X_1,X_2,\ldots X_n)$ respectively, the equation (2.84) could be written as follows

$$\Delta Y_j \cong f_j(\underline{x}) - f_j(\underline{X}) = \sum_{i=1}^{n} x_i \frac{\partial f_j(\underline{X})}{\partial x_i} + \frac{1}{2!}\sum_{i,k=1}^{n} \Delta x_i \, \Delta x_k \frac{\partial^2 f_j(\underline{X})}{\partial x_i \, \partial x_k} + \ldots \quad (2.85)$$

where Δx_i stands for $x_i - X_i$. If one of the increments Δx_L is greater than the others, our equation (2.85) can be written neglecting some terms, in the following way

$$\Delta Y_j \cong \sum_{i=1}^{n} \Delta x_i \frac{\partial f_j(\underline{X})}{\partial x_i} + \frac{1}{2!}\left[\Delta x_L^2 \frac{\partial^2 f_j(\underline{X})}{\partial x_L^2} + 2\sum_{\substack{k=1 \\ (k\neq L)}}^{n} \Delta X_L \Delta x_k \frac{\partial^2 f_j(\underline{X})}{\partial x_L \partial x_k}\right] \tag{2.86}$$

if $\Delta x_L \gg \Delta x_i$ for all $i \neq L$

or rewriting in terms of functions of Δx_L

$$\Delta Y_j \cong g_{jL}(\Delta x_L)\Delta x_1 + g_{jL}(\Delta x_L)\Delta x_2 + \ldots + g_{j(L-1)}(\Delta x_L)\Delta x_{L-1} + g_{jL}(\Delta x_L) + g_{j(L+1)}(\Delta x_L)\Delta x_{L+1} + \ldots + g_{jn}(\Delta x_L)\Delta x_n \tag{2.87}$$

Then, for this component, we can form the following system

$$\begin{cases} \Delta Y_1 = g_{1L}(\Delta x_L)\Delta x_1 + \ldots + g_{1L}(\Delta x_L) + \ldots + g_{1n}(\Delta x_L)\Delta x_n \\ \quad\bullet \qquad\qquad \bullet \qquad\qquad \bullet \qquad\qquad \bullet \\ \Delta Y_j = g_{jL}(\Delta x_L)\Delta x_1 + \ldots + g_{jL}(\Delta x_L) + \ldots + g_{jn}(\Delta x_L)\Delta x_n \\ \quad\bullet \qquad\qquad \bullet \qquad\qquad \bullet \qquad\qquad \bullet \\ \Delta Y_m = g_{mL}(\Delta x_L)\Delta x_1 + \ldots + g_{mL}(\Delta x_L) + \ldots + g_{mn}(\Delta x_L)\Delta x_n \end{cases} \tag{2.88}$$

A given test result is comprised of contributions of random deviations of components which are within tolerance, as well as from the faulty component. We see in the above system that only one column is a function of one increment, all the others are function of two increments. Therefore, if in our circuit all the components remain at the normal value, except the faulty component (X_L), the system (6) becomes

$$\begin{cases} \Delta Y_1 = g_{1L}(\Delta x_L) \\ \quad\bullet \qquad\bullet \\ \Delta Y_j = g_{jL}(\Delta x_L) \\ \quad\bullet \qquad\bullet \\ \Delta Y_m = g_{mL}(\Delta x_L) \end{cases} \tag{2.89}$$

The above is the contribution in each measurement value of the faulty component. Therefore, all the other terms in the equation (2.88) are the contribution of each measurement value, made by the remaining components that are within the allowed range. For all this reasoning, our problem is similar to the classical communication problem of detecting a signal in the presence of noise.

Our system (2.88) could be written as follows

$$\begin{cases} \Delta Y_1 = S_{1L}(\Delta x_L) + N_{1L}(\Delta x_1, \Delta x_2, \ldots \Delta x_n) \\ \quad \bullet \qquad\qquad \bullet \qquad\qquad\qquad \bullet \\ \Delta Y_j = S_{jL}(\Delta x_L) + N_{jL}(\Delta x_1, \Delta x_2, \ldots \Delta x_n) \\ \quad \bullet \qquad\qquad \bullet \qquad\qquad\qquad \bullet \\ \Delta Y_m = S_{mL}(\Delta x_L) + N_{mL}(\Delta x_1, \Delta x_2, \ldots \Delta x_n) \end{cases} \tag{2.90}$$

It is interesting to note that the "noise" characteristics are function of what was called "signal" (Δx_L).

$$N_{jL} = \Delta Y_j - S_{jL} \tag{2.91}$$

When N_{jL} is composed of a large number of independent Gaussian-like random variables $(\Delta x_1, \Delta x_2, \ldots, \Delta x_n)$, the joint distribution approaches a Gaussian distribution.

Let

$$\begin{aligned} m_{jkL} \triangleq \overline{N_{jL}N_{kL}} = &\left[g_{jL}(\Delta x_L)g_{kL}(\Delta x_L)\right] \sigma^2(\Delta x_1) + \ldots + \\ &\left[g_{j(L-1)}(\Delta x_L)g_{k(L-1)}(\Delta x_L)\right] \sigma^2(\Delta x_{L-1}) \\ &+ \left[g_{j(L+1)}(\Delta x_L)g_{k(L+1)}(\Delta x_L)\right] \sigma^2(\Delta x_{L+1}) + \ldots \end{aligned} \tag{2.92}$$

where $\sigma^2(\Delta x_j)$ is the variance of an in-tolerance component x_j. The moment matrix is given by

$$M_{\ell} = \begin{bmatrix} m_{11L} & m_{12L} & \cdots & m_{1mL} \\ m_{21L} & m_{22L} & \cdots & m_{2mL} \\ . & . & \cdots & . \\ m_{m1L} & m_{m2L} & \cdots & m_{mmL} \end{bmatrix} \tag{2.93}$$

The joint probability of the m normally distributed random variables N_1, N_2, ... N_m are given by

$$P_L(N_{1L}, N_{2L}, \ldots N_{mL}) = \frac{1}{(2\pi)^{\frac{m}{2}}\sqrt{|M_L|}} \exp{-\frac{1}{2}}\left(\sum_{i=1}^{m}\sum_{k=1}^{m} m^{-1}_{ikL} N_{iL} N_{kL}\right) \tag{2.94}$$

where m^{-1}_{ikL} stands for the element that appears in the i^{th} row and k^{th} column of the inverse of the M_L matrix. (For a deduction of the equation (2.94) see Laning & Battin - Random Processes in Automatic Control - McGraw Hill - page 73.) Given a set of measurements values $(\Delta Y_1, \Delta Y_2, \ldots \Delta Y_m)$, the probability that a given component is responsible for it is as follows

$$P(X_L/\Delta Y_1, \Delta Y_2, \ldots \Delta Y_m) = K\int P(X_L, x_L) P(\Delta Y_1, \Delta Y_2, \ldots \Delta Y_m / X_L, x_L) dx_L \tag{2.95}$$

where $P(X_L, x_L)$ stands for the density distribution of component L, that with nominal value X_L, had the value x_L.

The value of K is equation (2.95) is found knowing that

$$\sum_{L=1}^{n} P(X_L/\Delta Y_1, \Delta Y_2, \ldots \Delta Y_m) = 1 \tag{2.96}$$

In order to be able to use equation (2.95), we must know the value $P(\Delta Y_1, \Delta Y_2, \ldots \Delta Y_m / X_L, x_L)$; to do this, we use the following equation

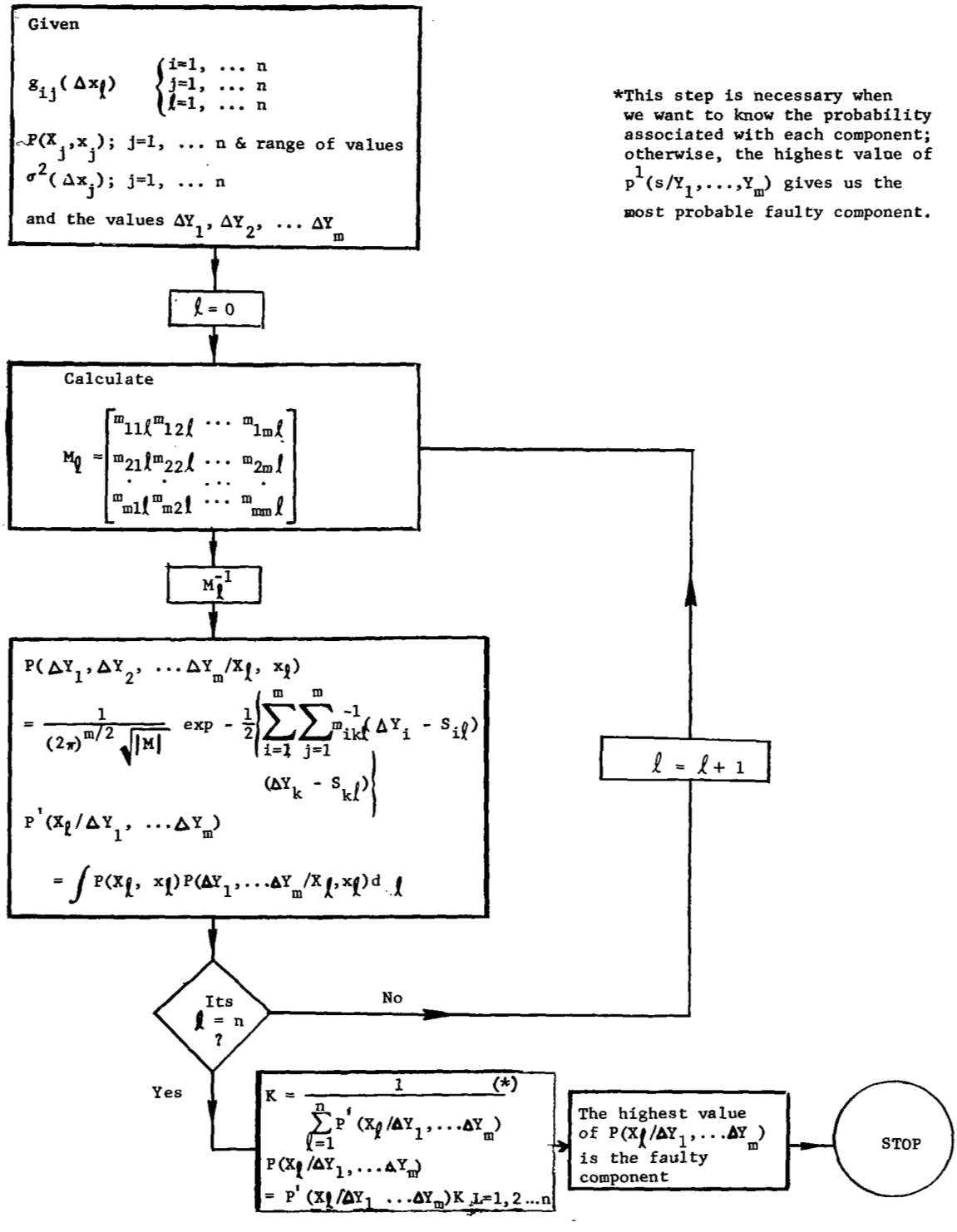
$$P(\Delta Y_1,\Delta Y_2, \ldots \Delta Y_m / X_L, x_L) = \frac{1}{(2\pi)^{\frac{m}{2}}\sqrt{|M_L|}} \exp -\tfrac{1}{2}\left\{\sum_{i=1}^{m}\sum_{j=1}^{m} m_{ikL}^{-1}(\Delta Y_i - S_{iL})\right.$$

$$\left.(\Delta Y_k - S_{kL})\right\} \qquad (2.97)$$

Knowing the circuit, and given a set of measurement values $(\Delta Y_1,\Delta Y_2, \ldots \Delta Y_m)$, for each component, we can get by equation (2.97) which is the probability that this will be the faulty component. Therefore, we can prepare the following list

$$\begin{aligned} &P(x_1/\Delta Y_1,\Delta Y_2, \ldots \Delta Y_m) \\ &P(x_2/\Delta Y_1,\Delta Y_2, \ldots \Delta Y_m) \\ &\quad\vdots \\ &P(x_n/\Delta Y_1,\Delta Y_2, \ldots \Delta Y_m) \end{aligned} \qquad (2.98)$$

The most probable faulty component will be the component that has the highest probability associated with it. As a clarification about the steps that are involved in this process, a flow graph is indicated in Figure 2.28.



*This step is necessary when we want to know the probability associated with each component; otherwise, the highest value of $p^1(s/Y_1, \ldots, Y_m)$ gives us the most probable faulty component.

Figure 2.28

2.6.2 Conclusions -

The most important features of this method could be summarized as follows

- This method makes a good utilization of all the information that is possible to get from the circuit under test, like for instance the density distribution values for each component.
- The number of measurements are less than the number of components.

2.6.3 Applications -

The application of this method is only for linear circuits/systems. The extension of this method to cover non-linear circuits/systems appears not to be feasible.

2.6.4 Computer Requirements -

The complex post-calculations involved in this method impossed strong requirements on the computer required. These requirements increase rapidly with the increase at the size of the circuit under test.

For any set of measurements taken it is possible to see in the flow-chart shown in figure 2.28 all the mathematical operations involved in the process. With some improvements on this method it would be possible to make some reduction in the post-calculations involved.

2.7 Power Spectra Method -

2.7.1 General

This method consists of obtaining the faulty component through the knowledge of the transfer function, which is found by power spectra analysis. There are two big steps involved in the process

- Transfer function determination
- Faulty component diagnosis

2.7.1.1 Transfer Function Determination - Let the configuration of our system under test be given by Figure 2.29. The input to the system is formed by two parts: the normal signal $f_s(t)$, and the random signal $\eta(t)$. The input to the digital computer will be the input and the output of the system under test. With these two signals we may form the autocorrelation function of the input and the cross correlation function between the input and the output. The Fourier transforms of both functions give us the power spectra and the cross power spectra; the relation between them permits the determination of the transfer function.

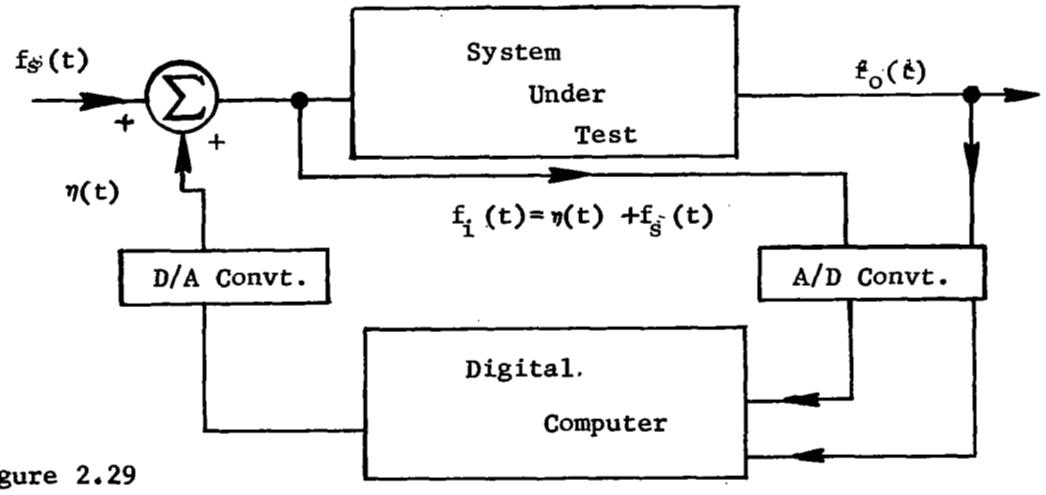


Figure 2.29

Figure 2.30 gives us a clear idea of the relations involved.

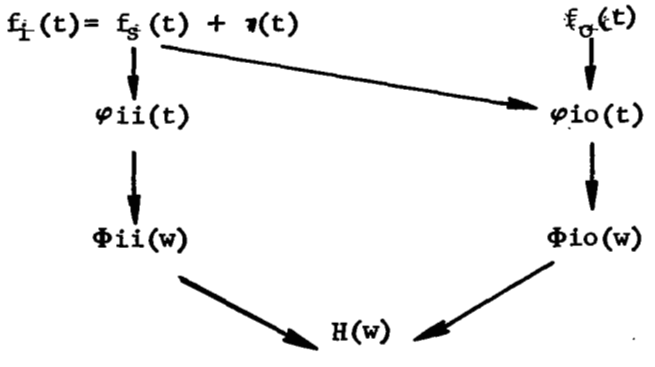


Figure 2.30

In order to develop the equations which we need to carry out the calculations, let us take into consideration the system shown in Figure 2.31.

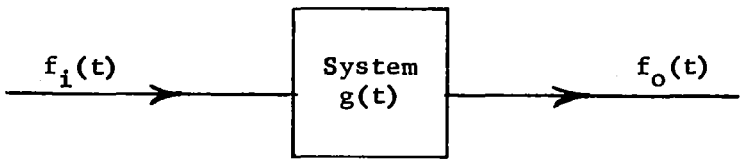


Figure 2.31

The autocorrelation function of the input signal is given by

$$\varphi_{ii}(\tau) = \operatorname{Lim}_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} f_i(t)\ f_i(t+\tau)dt \qquad (2.99)$$

and the cross correlation function between the input and output signals by

$$\varphi_{io}(\tau) = \operatorname{Lim}_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} f_i(t)\ f_o(t+\tau)dt \qquad (2.100)$$

The Fourier transform of $\varphi_{ii}(\tau)$ is the power spectra of the input signal and is designated by $\Phi_{ii}(w)$. In the same way, $\Phi_{io}(w)$ is the cross power spectra between the input and output signals. By means of the convolution function, it is possible to write the output signal in function of the input signal and the impulse response of the system.

$$f_o(t) = \int_{-\infty}^{\infty} g(x)\ f_i(t-x)dx \qquad (2.101)$$

and then

$$f_o(t+\tau) = \int_{-\infty}^{\infty} g(x)\ f_i(t+\tau-x)dx \qquad (2.102)$$

From equations (2.100) and (2.102), it is possible to write

$$\varphi_{io}(\tau) = \operatorname{Lim}_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} f_i(t) \int_{-\infty}^{\infty} g(x)\ f_i(t+\tau-x)dx\ dt \qquad (2.103)$$

If the order of integration is interchanged, equation (2.103) may be written as

$$\varphi_{io}(\tau) = \int_{-\infty}^{\infty} g(x) \left[\operatorname{Lim}_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} f_i(t)\ f_i(t+\tau-x)dt \right] dx \qquad (2.104)$$

It is easy to recognize that the bracketed portion is the auto-correlation function of the input. The convolution between the impulse response of the system and the auto-correlation function, results in the cross-correlation function; or in other words, if we apply in the input of the system the function $\varphi_{ii}(\tau)$, we are going to get in the output $\varphi_{io}(\tau)$. See Figure 2.32.

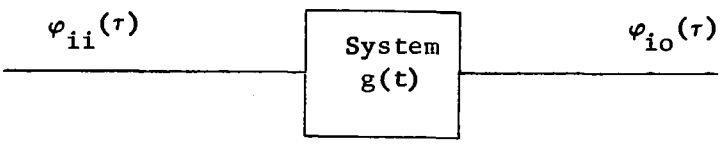


Figure 2.32

In the transform domain the equation (2.104) becomes

$$\Phi_{io}(w) = G(w) \ \Phi_{ii}(w) \tag{2.105}$$

and then

$$G(w) = \frac{\Phi_{io}(w)}{\Phi_{ii}(w)} \tag{2.106}$$

If the input to our system is white noise, the power spectra is a constant, therefore

$$G(w) = K \ \Phi_{io}(w) \tag{2.107}$$

Depending on the case, the transfer function is found by use of equation (2.106) or (2.107).

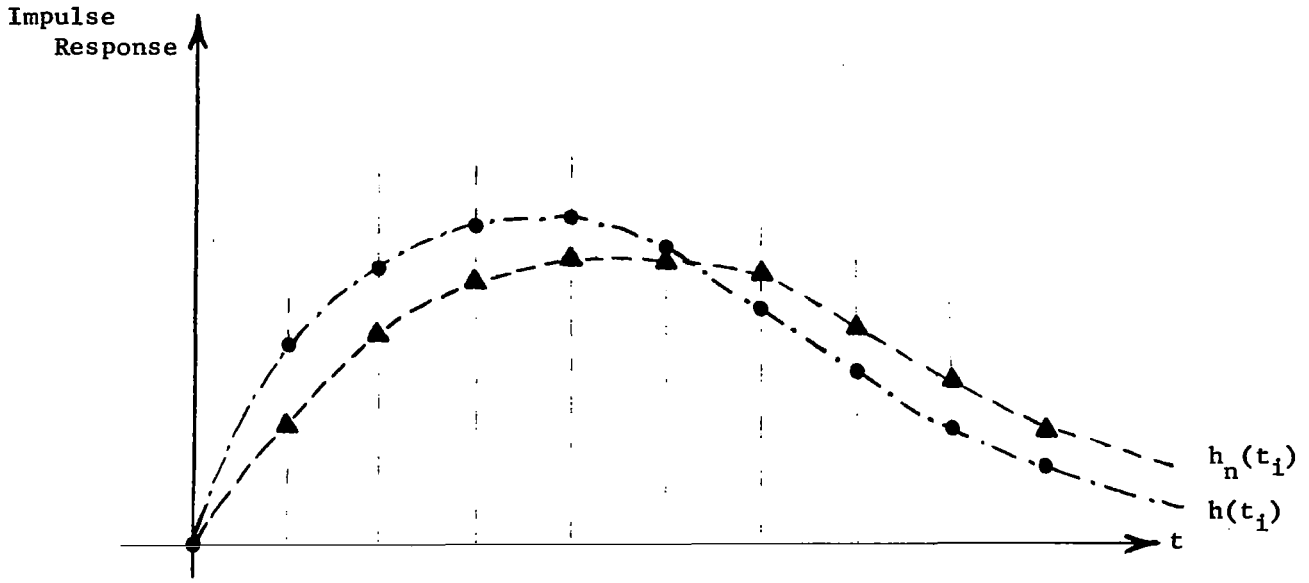
2.7.1.2 Faulty Component Diagnosis [17]

After the transfer function is obtained, the impulse response is calculated by means of numerical methods.

A sample of this data must be compared with a nominal sample that is found knowing the impulse response that describes the system under test and the nominal values of the components. Now, to complete the diagnosis we have two steps in our process

- Check if both results represent the same impulse response.
- If both results represent two different impulse responses, find the component/s responsible for this cause.

In the first step we have two sets of values $h(t_i)$ and $h_n(t_i)$, both for i=1,2,...,n. We can apply the chi-square goodness of fit criteria. In order to apply this criteria, the interval chosen for sampling the functions must be such that the points obtained will be independent.



$h_n(t_i)$ = Nominal values

$h(t_i)$ = Experimental values

Figure 2.33

The chi-square value is given by

$$\chi^2 = \sum_{i=1}^{n} \frac{\left[h(t_i) - h_n(t_i)\right]^2}{h_n(t_i)} \qquad (2.108)$$

The degree of freedom k is one less than the total number of independent points in one curve. Given the probability that both data sets belong to the same curve, and knowing the degree of freedom, from Table V (page 368 - John E. Freund - "Mathematical Statistics") we get the value of χ^2. For instance, for a probability of 0.975, and a degree of freedom 26, we get $\chi^2 = 13.844$. This means that our chi-square value calculated by means of equation (11) will be less or equal to 13.844 ($\chi^2_{i=1} \leq 13.844$), in order to say that with a probability of 0.975 both data sets belong to the same curve.

If the result of this step is positive it will mean that our system is working in optimum performance. Now, on the other hand, if the result of this step is negative, it will mean that we must go to the second step, which is to find the components responsible for this result and check if their values are within the allowed range. If this is the case, our system is operating in a normal condition; if not, we have to diagnose the faulty component/s.

In the second step we assume that the impulse response of a system $h(t, p_1, p_2, \ldots, p_n)$ is differentiable at $p_1, p_2, \ldots, p_n$, then it can be represented by a multiple Taylor series.

$$h(t,p_1,p_2,\ldots,p_n) = h(t,p_{10},p_{20},\ldots,p_{n0}) + \sum_{j=1}^{n} \frac{\partial h(t,p_{10},p_{20},\ldots,p_{n0})}{\partial p_j} \Delta p_j$$

$$+ \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 h(t,p_{10},p_{20},\ldots,p_{n0})}{\partial p_i \, \partial p_j} \Delta p_i \Delta p_j + \ldots \qquad (2.109)$$

where p_{j0} (for j=1,2,...,n) indicates the nominal parameter values. For small variations in the parameters, an approximation to the impulse response can be obtained, if we consider the terms up to the first order only. Then

$$h(t,p_1,p_2,\ldots,p_n) \cong h(t,p_{10},p_{20},\ldots,p_{n0}) + \sum_{j=1}^{n} \frac{\partial h(t,p_{10},p_{20},\ldots p_{n0})}{\partial p_j} \Delta p_j \qquad (2.110)$$

The difference between the impulse response at the operating parameter values and the impulse response at the nominal parameter values is given by

$$e(t) = h(t,p_1,p_2,\ldots,p_n) - h(t, p_{10},p_{20},\ldots,p_{n0})$$

$$= \sum_{j=1}^{n} \frac{\partial h(t,p_{10},\ldots,p_{n0})}{\partial p_j} \Delta p_j \qquad (2.111)$$

Since equation (2.111) is a function of the time(t), we can apply it for n different values. Therefore, we get the following equation system

$$\begin{cases} e(t_1) = a_{11} \Delta p_1 + a_{12} \Delta p_2 + \ldots + a_{1n} \Delta p_n \\ e(t_2) = a_{21} \Delta p_1 + a_{22} \Delta p_2 + \ldots + a_{2n} \Delta p_n \\ \quad \bullet \qquad \bullet \qquad \bullet \qquad \bullet \\ e(t_n) = a_{n1} \Delta p_1 + a_{n2} \Delta p_2 + \ldots + a_{nn} \Delta p_n \end{cases} \qquad (2.112)$$

where

$$a_{ij} = \frac{\partial h(t_1,p_{10},p_{20},\ldots,p_{n0})}{\partial p_j}$$

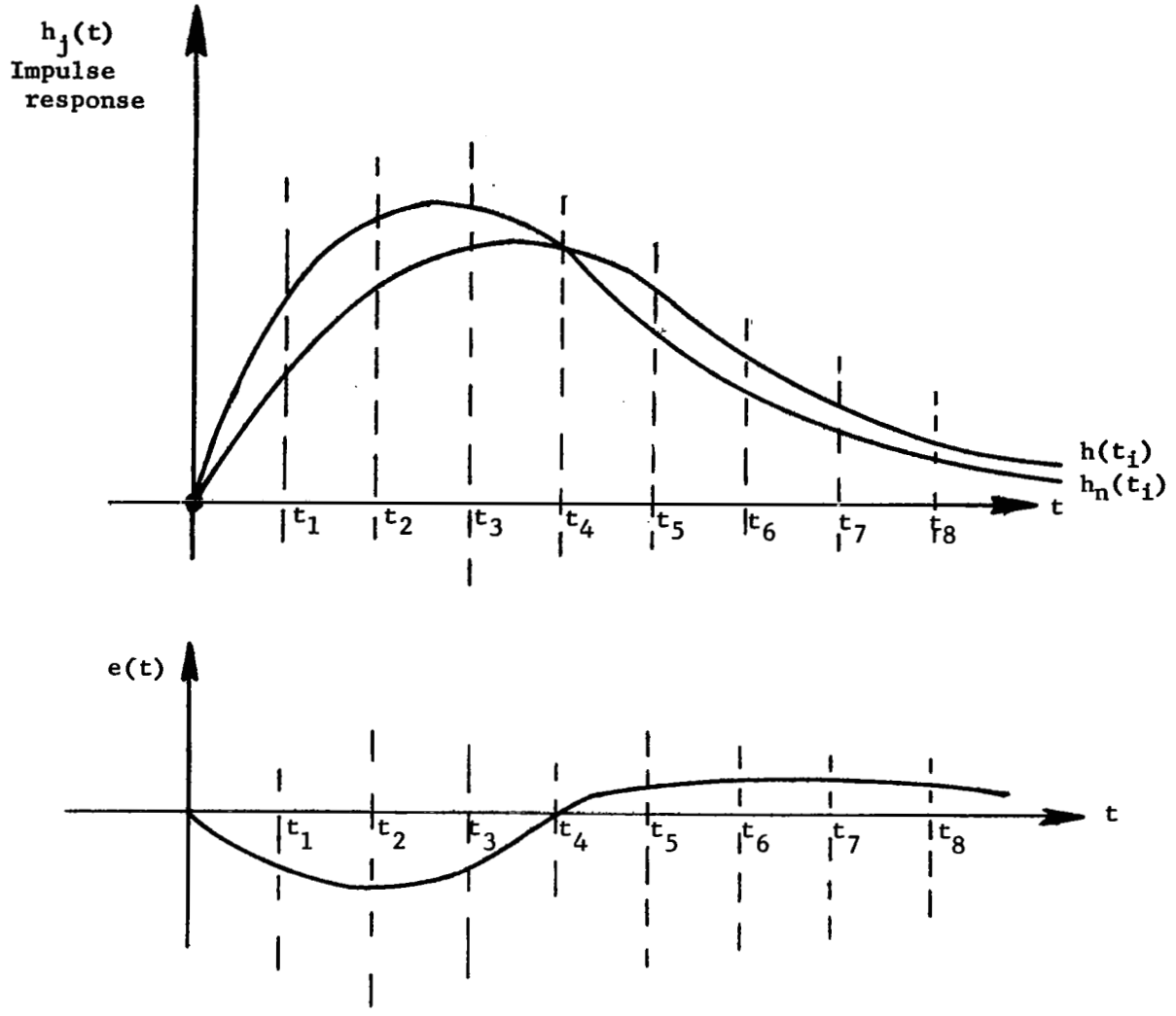


Figure 2.34

The equation (2.112) can be put in the following matrix form

$$\begin{bmatrix} e(t_1) \\ e(t_2) \\ \vdots \\ e(t_n) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdot & \cdot & a_{nn} \end{bmatrix} \begin{bmatrix} \Delta P_1 \\ \Delta P_2 \\ \vdots \\ \Delta P_n \end{bmatrix} \qquad (2.113)$$

and then

$$\begin{bmatrix} \Delta p_1 \\ \Delta p_2 \\ \cdot \\ \cdot \\ \Delta p_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdot & \cdot & a_{nn} \end{bmatrix}^{-1} \begin{bmatrix} e(t_1) \\ e(t_2) \\ \cdot \\ \cdot \\ e(t_n) \end{bmatrix} \qquad (2.114)$$

or

$$\Delta p = A^{-1} E \qquad (2.115)$$

The matrix A is determinated a priori, and the same happens with A^{-1}. Therefore, knowing the error at n different times, it is possible to find the variation (Δp_j) in each parameter, and therefore, to see if the same is in the allowed tolerance range or not.

2.7.2 Conclusions -

The only test points needed in this method are the input and the output of the circuit/system under test; therefore, it is possible to apply it to any circuit/system that is already in use.

The level of noise introduced for test purposes must be kept low in order not to produce any effects in the normal operation of the circuit/system under test.

2.7.3 Applications -

The principal application of this method is to perform the checkout of a linear circuit/system while it is in normal operation.

The extension of this method to include nonlinear circuits/systems appears not to be feasible.

2.7.4 Computer Requirements -

The computer requirements for handling all the post-calculations involved in the faulty component/s determination are extensive, but the advantages of performing the checkout without interruption of the operation makes this method very attractive.

The diagnosis calculations will be performed only when the nominal impulse response and the experimental impulse response do not represent the same impulse response (chi-square goodness of fit criteria).

2.8 Parameter Identification Method [4]

2.8.1 General

This method identifies the coefficients of the transfer function of the unit under test. The output of the unit is cross-correlated with the outputs of orthogonal filters that are excited by the same stimulus (white noise); from this cross-correlation, the coefficients of the transfer function are obtained.

The response of a system to white noise is sufficient to completely describe both linear and nonlinear systems.

Let us consider the linear system indicated in Figure 2.35.

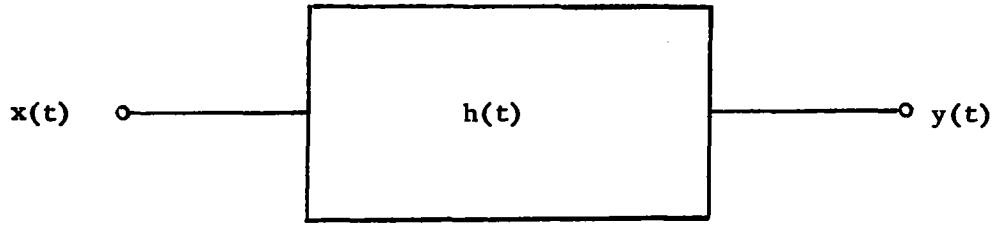


Figure 2.35

where

h(t) = impulse response of the system

y(t) = output of the system

x(t) = input of the system

The output of the system is given by

$$y(t) = \int_0^{\infty} x(t - \tau)\, h(\tau)\, d\tau \qquad (2.116)$$

Similarly, the output of the ℓth filter will be given by

$$y_\ell(t) = \int_0^\infty x(t - \tau)\, \ell_n(\tau)\, d\tau \tag{2.117}$$

where

$\ell_n(t)$ = impulse response of the nth filter

$y_\ell(t)$ = output of the nth filter.

Let us define the average of the product of $y(t)$ and $Y_\ell(t)$ in the following way

$$\overline{y(t) \cdot y_\ell(t)} = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} y(t) \cdot y_\ell(t)\, dt \tag{2.118}$$

Then

$$\overline{y(t) \cdot y_\ell(t)} = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} dt \int_0^\infty x(t - \tau_1)\, h(\tau_1)\, d\tau_1 \int_0^\infty x(t - \tau_2)\ell_n(\tau_2) d\tau_2 \tag{2.119}$$

By interchanging the integration order, we get

$$\overline{y(t) \cdot y_\ell(t)} = \int_0^\infty d\tau_1 h(\tau_1) \int_0^\infty d\tau_2\, \ell_n(\tau_2) \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} x(t - \tau_1)x(t - \tau_2)dt \tag{2.120}$$

But $\lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} x(t - \tau_1)x(t - \tau_2)dt = \varphi_{xx}(\tau_1 - \tau_2)$, then

$$\overline{y(t) \cdot y_\ell(t)} = \int_0^\infty d\tau_1 h(\tau_1) \int_0^\infty d\tau_2\, \ell_n(\tau_2)\, \varphi_{xx}(\tau_1 - \tau_2) \tag{2.121}$$

The value $\overline{y(t) \cdot y_\ell(t)}$ represents voltage level output and will be designated with A_n.

$$A_n = \int_0^\infty d\tau_1 \, h(\tau_1) \int_0^\infty d\tau_2 \, \ell_n(\tau_2) \, \varphi_{xx}(\tau_1 - \tau_2) \tag{2.122}$$

If x(t) is white noise, the power spectra is theoretically the same value K^2 for all frequencies, or $\Phi_{xx}(w) = K^2$, and therefore the auto-correlation of the noise is an impulse of area $2\pi K^2$, or

$$\varphi_{xx}(\tau_1 - \tau_2) = 2\pi K^2 \, \delta(\tau_1 - \tau_2) = N^2 \delta(\tau_1 - \tau_2) \tag{2.123}$$

Considering equation (2.123), equation (2.122) becomes

$$A_n = \int_0^\infty d\tau_1 h(\tau_1) \int_0^\infty d\tau_2 \, \ell n(\tau_2) \, N^2 \delta(\tau_1 - \tau_2)$$

$$= N^2 \int_0^\infty d\tau \, h(\tau) \, \ell_n(\tau) \tag{2.124}$$

By means of the Parseval's theorem, equation (2.124) becomes

$$A_n = \frac{N^2}{2\pi j} \int_{-j\infty}^{+j\infty} \ell_n(s) \quad h(-s) ds \tag{2.125}$$

This general expression may be expanded in terms of the residues of the orthogonal filter transfer function; in this way, a relationship for the coefficients of Y(s) is obtained. If p is the number of coefficients in the transfer function, we need to have n from 1 to p. Therefore, our p equations will be given by

$$A_n = \frac{N^2}{2\pi j} \int_{-j\infty}^{+j\infty} \ell_n(s) \, h(-s) \, ds \quad \text{for} \quad n = 1, 2, \ldots, p$$

The configuration needed is shown in Figure 2.36.

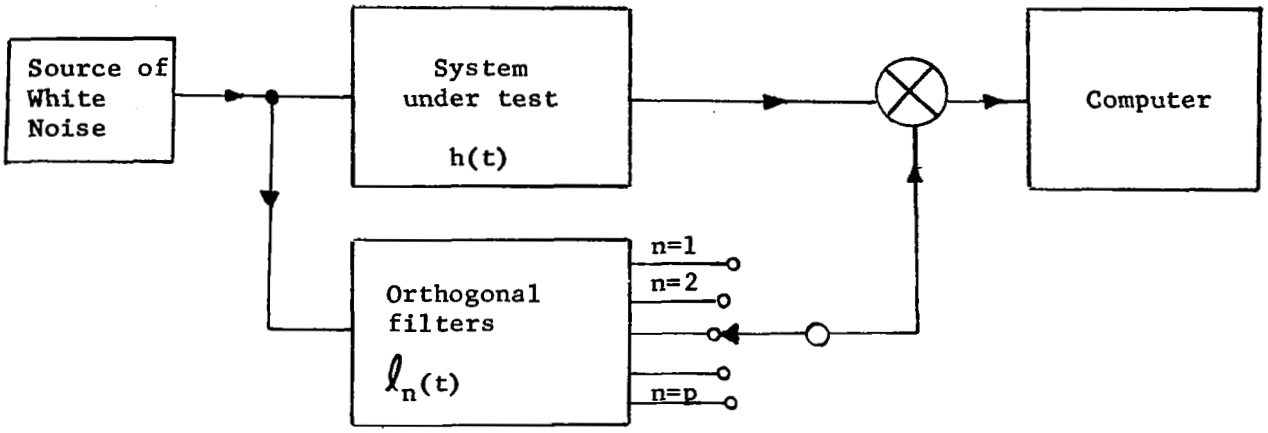


Figure 2.36

If the impulse response of the nth filter could be written as

$$\ell_n(s) = \frac{A_n(s)}{B_n(s)} \tag{2.126}$$

where $B_n(s)$ is a polynomial of nth order and within different roots, the value A_n could be written as

$$A_n = N^2 \sum_{m=1}^{n} \left| (s + s_m)\, \ell_n(s)\, h(-s) \right|_{s = -s_m} \tag{2.127}$$

Therefore, our system will be given by

$$\begin{cases} A_1 = N^2 \left| (s + s_1)\, \ell_1(s)\, h(-s) \right|_{s = -s_1} \\ A_2 = N^2 \left\{ \left| (s + s_1)\, \ell_2(s)\, h(-s) \right|_{s=-s_1} + \left| (s + s_2)\, \ell_2(s) h(-s) \right|_{s=-s_2} \right\} \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ A_p = N^2 \sum_{m=1}^{p} \left| (s + s_m)\, \ell_n(s)\, h(-s) \right|_{s=-s_m} \end{cases} \tag{2.128}$$

From this system, the p coefficients of $Y_h(s)$ are found.

The corresponding analysis for non-linear systems is more complex and makes use of Wiener's nonlinear theory. See for more details the reference given at the beginning of this method.

2.82 Conclusions -

The only test points needed in this method are the input and the output of the circuit/system under test; therefore, it is possible to apply it to any circuit/system that is already in use, no hardware requirements are needed.

2.8.3 Applications -

The application of this method covers linear and nonlinear circuits/systems, but the main disadvantages is that the circuit/system must be out of operation.

2.8.4 Computer Requirements -

The complex post-calculations involved in this method impose strong requirements on the computer. These requirements increase if the circuit/system under test is nonlinear. For this reason, the applicability of this method is only justified when there is no possibility of applying any other method of diagnosis.

2.9 Max-Current Method [3]

2.9.1 General

The idea of this method is as follows: "If between 2 nodes there are more than one branch, a variation in one or more of the components that belong to the branch which carries a predominant amount of current can be detected at the given nodes. On the other hand, any variation in one or more of the components that belong to the branch that only carries a small amount of current are masked by the normal intolerance variations of all the components. Therefore, the only path in which failure produce a unique effect is in the path that carries a predominant amount of current. This is called the majority current path (MCP)."

The calculations proceed from the test point or node (say node k) at which the positive test voltage is applied. See figure 2.37.

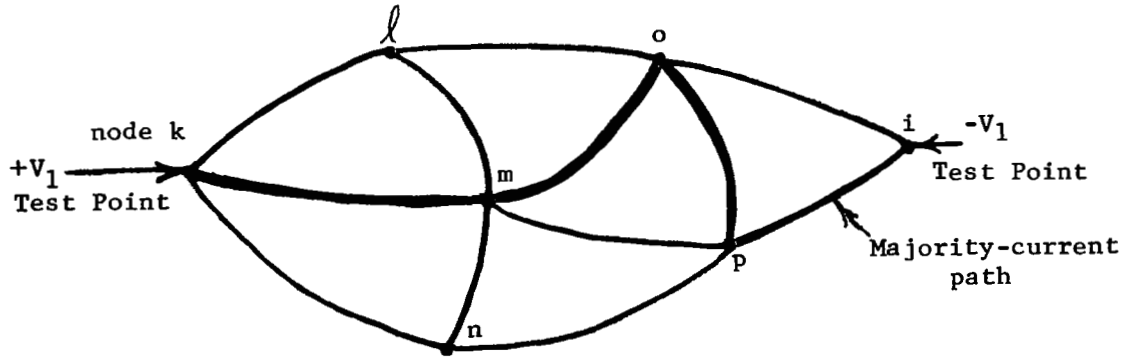


Figure 2.37

The remaining nodes that are connected to node k are examined and we choose the one that carries the highest current, say n. This new node now replaces node k and the process is continued until the return node of the test point pair is encountered, node i. All the nodes and branches that were selected belong to the MCP.

Now, from measurements that will be done between the test points (k-i), the only components that are going to have a unique effect in its variations will be the components that belong to the MCP; therefore, the fault isolation from these two test points will be only for the components belonging to the MCP. It is obvious, that we need more test points to make the isolation in all the components that belong to the circuit given in figure 2.37. In order to select the adequate number and placement of the test points, the following criteria are adequate:

1. There should be a test point at the output of each stage or at the input to the following stage.

2. There should be a minimum of one test point (including power, one input, and one output) for every seven passive components (resistor/capacitor/inductor) and semi-conductor elements (diode/transistor) in the circuit.

3. There should be a test point placed at each node to which three or more transistors are connected.

4. There should be one additional test point on each bridge type configuration (see Note below).

5. The added test points should not be connected at a node where the connection will adversely affect the circuit configuration, especially the input to a stage of isolation or amplification.

Note - Since the bridge configuration is the most difficult to analyze , the following procedure is suggested in the mentioned reference:

1. Determine the lowest numbered node to which three or more elements are connected.

2. Find the next lowest numbered node to which three or more elements are connected.

3. Is there an element from a node in step (1) to a node in step (2)? If yes, a bridge has been found, proceed to step (4). If not, return to step (2) to process the next node (i.e. to find the next higher numbered node with three elements connected to it).

4. Assume that a bridge has been found and the center nodes of the bridge are numbered i and j where i is lower in number than j. Determine if some element goes from node i or node j to ground. If yes, proceed to step (5); if not, that bridge configuration does not require an additional test point; return to step (2) to process the next node.

5. Determine if the base or the emitter of a transistor is connected to node i.

   a) If not, proceed to step (6)

   b) If yes, determine if the emitter has a test point or if it is connected to ground. If not, place a test point on the emitter; if yes, place a test point on the collector if one does not already exist there. Proceed to step (8).

6. Determine if the collector of a transistor is connected to node i.

   a) If not, repeat steps (5) and (6) for node j; then proceed to step (7).

   b) If yes, place a test point at the collector, proceed to step (8).

7. If a test point has not yet been assigned, node i is assigned as a test point.

8. Update node i as the next three element node and continue the procedure. (If node i is assigned as a test point, this node is never considered again for a potential bridge circuit. If a node k on the transistor connected to node i is assigned as a test point, node k is never considered again but node i is still under consideration.)

All the above steps are indicated in the flowchart shown in figure 2.38.

2.9.2 Conclusions -

The number of test points required, increase rapidly with the increase of the size of the circuit under test. Therefore, this places a limitation on the size that is possible to be handled.

Dynamic and static measurements will be performed in order to detect the faulty component.

2.9.3 Applications -

The application of this method is for transistorized circuits, and not for systems.

2.9.4 Computer Requirements -

Only simple post calculations are involved, therefore there are not special requirements about the computer that will be needed.

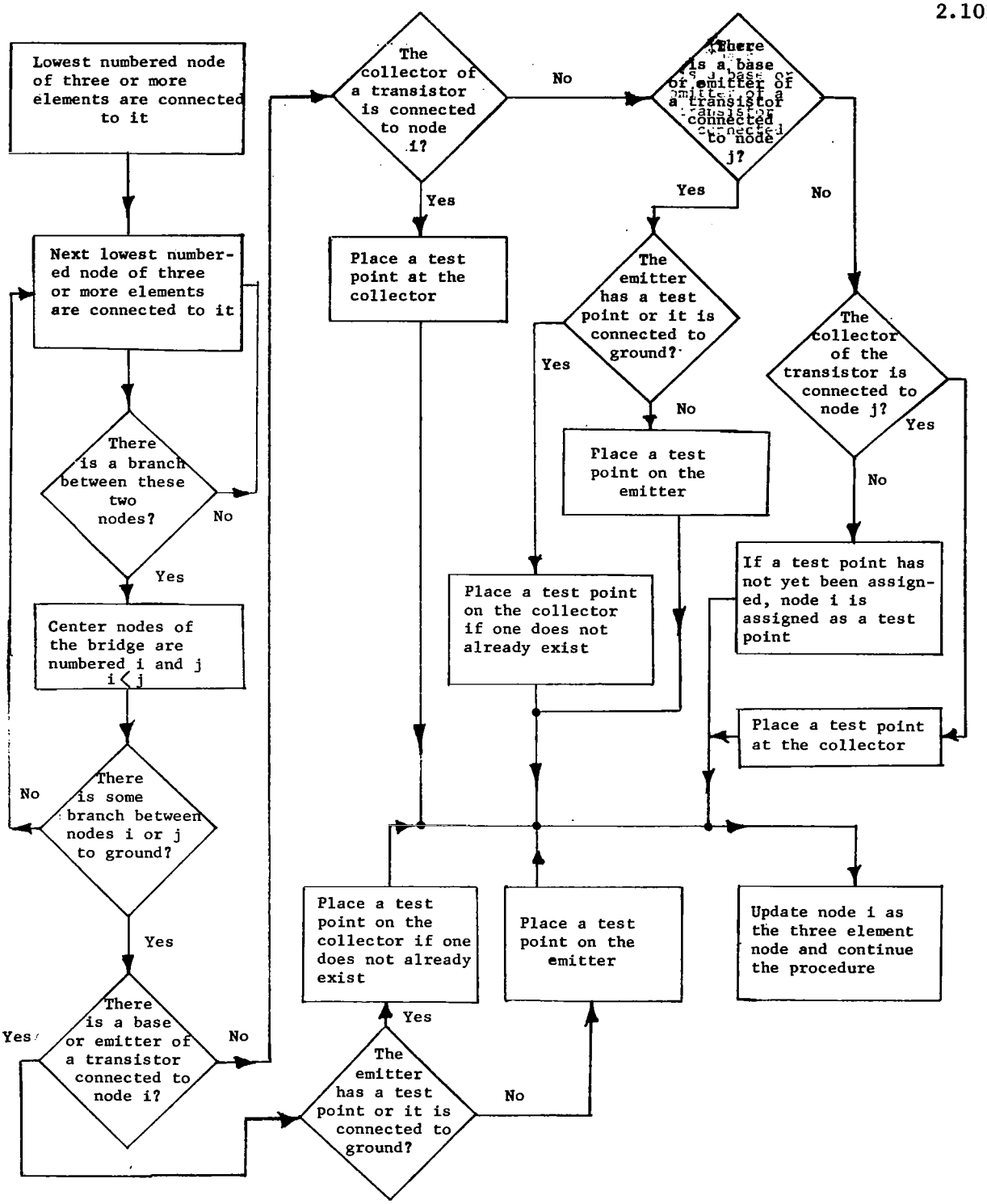


Figure 2.38 - Procedure developed to analyze bridge configuration.

2.10 Detection Condition Method (Dynamic System Testing) -

2.10.1 General

This method consists of taking measurements from a transient response of a system. The following assumptions are made regarding the unit that will be checked.

- The configuration of the unit is known, but the actual values of its components are unknown.
- The design values of the components that belong to the unit are known; therefore, the theoretical coefficients of the difference equation are known.
- In order for the performance of the unit to be satisfactory, the coefficients of the actual difference equation that describe the system must be within known tolerance bounds.

With all the above assumptions, a mathematical condition is obtained between the theoretical coefficients, the measurements taken and the bounds of each coefficient. If this condition is satisfied, the performance of the unit is acceptable; otherwise, it is faulty.* If the performance of the unit is nonsatisfactory, the data obtained could be used to make the diagnosis.[23]

Therefore, the two approaches outlined above are

- First, if our system is working with all the components within the normal range, and
- Second, if this is not the case, what are the component/s responsible for this change.

In order to accomplish the first step, we assume that the curve $h_n(t)$ shown in Figure 2.39 represents the theoretical impulse response of the system under test. Since the components have some range of tolerance, for any specific combination of them, we are going to have another impulse response, as the one that is shown in the mentioned figure $(h(t))$. Also the difference between $h_n(t)$ and $h(t)$ is indicated in the same figure.

The samples obtained from both curves must satisfy the following difference equations

*Garzia, R. F. - A Fast Technique for Dynamic Fault Detection - 1970 IEEE Automatic Support Systems for Advanced Maintainability - St. Louis, Missouri - October 1970.

$$\begin{cases} \sum_{i=0}^{m} a_i \, h_n(n + iT) = 0 \\ \sum_{i=0}^{m} a'_i \, h(n + iT) = 0 \end{cases} \qquad (2.129)$$

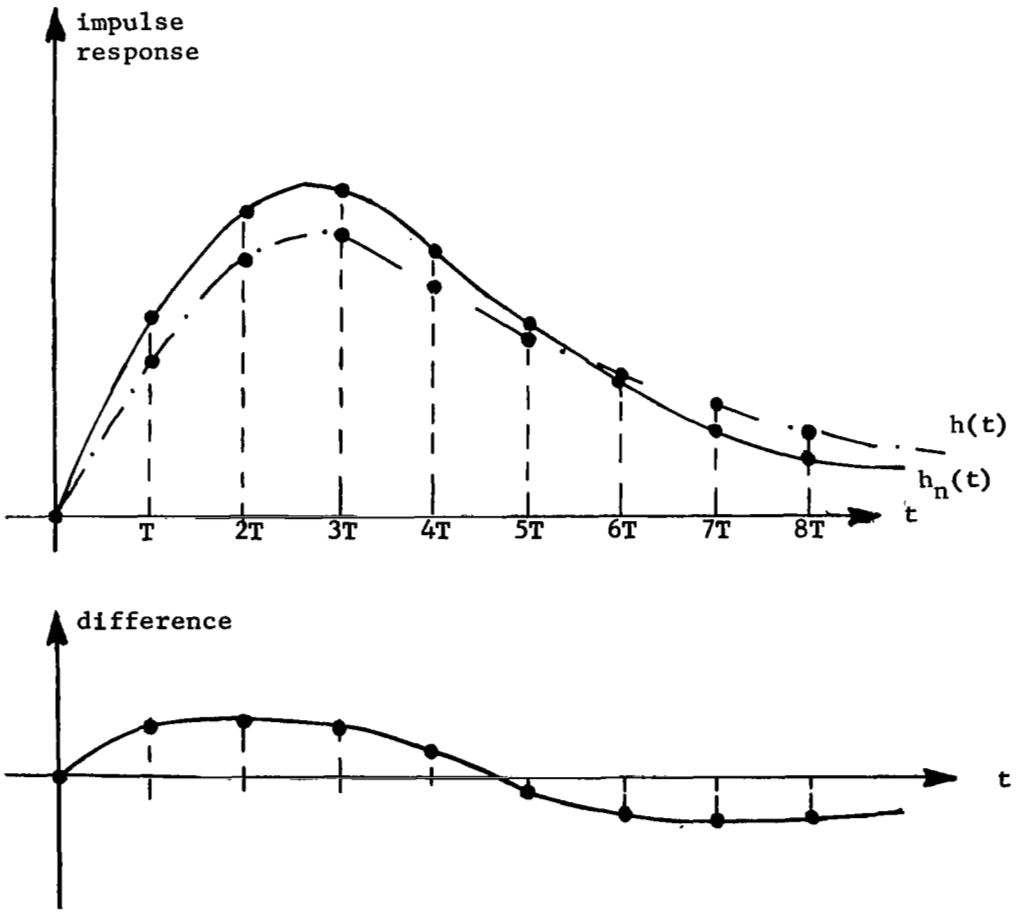


Figure 2.39

Now, in order for the system to be within the normal range, each coefficient must be subject to the following constraint.

$$a_i - \epsilon_{imin} \leq a_i' \leq a_i + \epsilon_{imax} \tag{2.130}$$

In general, we can write

$$a_i' = a_i + \epsilon_i \tag{2.131}$$

Therefore, the second equation of (2.129) becomes

$$\sum_{i=0}^{m} a_i\, h(n + iT) + \sum_{i=0}^{m} \epsilon_i\, h(n + iT) = 0 \tag{2.132}$$

Now, if in equation (2.132) we write ϵ_{imax} instead of ϵ_i, we get the following relationship

$$\sum_{i=0}^{m} a_i\, h(n + iT) + \sum_{i=0}^{m} \epsilon_{imax}\, h(n + iT) \geq 0 \tag{2.133}$$

Making the same thing with the lower bound, we arrive at the following equation

$$\sum_{i=0}^{m} a_i\, h(n + iT) - \sum_{i=0}^{m} \epsilon_{imin}\, h(n + iT) \leq 0 \,. \tag{2.134}$$

If $\epsilon_{imin} = \epsilon_{imax} = \epsilon_{im}$

$$\begin{cases} \displaystyle\sum_{i=0}^{m} a_i\, h(n + iT) + \sum_{i=0}^{m} \epsilon_{im}\, h(n + iT) \geq 0 \\ \displaystyle\sum_{i=0}^{m} a_i\, h(n + iT) - \sum_{i=0}^{m} \epsilon_{im}\, h(n + iT) \leq 0 \end{cases} \tag{2.135}$$

Therefore, we can state the following condition

$$\left| \sum_{i=0}^{m} a_i \, h(n + iT) \right| \leq \left| \sum_{i=0}^{m} \epsilon_{im} \, h(n + iT) \right| \qquad (2.136)$$

The above condition is a necessary but not sufficient one for all coefficients of the system to be within their tolerance limits. However, if $\left| \sum_{i=0}^{m} a_i \, h(n + iT) \right| > \left| \sum_{i=0}^{m} \epsilon_{im} \, h(n + iT) \right|$ is a sufficient condition for one or more coefficients to be out of tolerance.

In order to accomplish the second step, we are going to use the Prony's method that is outlined below and makes use of a number of measurements given by two times the degree of the differential equation that describes the system. [23] If we have indicated with m the degree of our differential equation, the transfer function of our system is given by

$$H(s) = \frac{K(s - s_I)(s - s_{II}) \ldots (s - s_d)}{(s - s_1)(s - s_2) \ldots (s - s_m)} \qquad (2.137)$$

Now, making the fraction partial expansion of equation (2.137)

$$H(s) = \frac{K_1}{s - s_1} + \frac{K_2}{s - s_2} + \ldots + \frac{K_m}{s - s_m} \qquad (2.138)$$

Taking the anti-Laplace transform of equation (2.138) we get

$$h(t) = K_1 e^{s_1 t} + K_2 e^{s_2 t} + \ldots + K_m e^{s_m t}$$

In the derivation of this equation, we assume that all the roots are different.

If instead of having t as a continuous variable, we assume that the values are taken at equal interval, called T, it is possible to write the equation (2.139) as follows

$$h(nT) = K_1 e^{s_1 nT} + K_2 e^{s_2 nT} + \ldots + K_m e^{s_m nT} \qquad (2.140)$$

Of course, both expressions identify the same system. Now, applying the equation (2.140) to different values of n, we can form the following system

$$\begin{cases} h(0) = K_1 + K_2 + \ldots + K_m \\ h(T) = K_1 e^{s_1 T} + K_2 e^{s_2 T} + \ldots + K_m e^{s_m T} \\ \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \\ h(mT) = K_1 e^{s_1 mT} + K_2 e^{s_2 mT} + \ldots + K_m e^{s_m mT} \end{cases} \qquad (2.141)$$

In order to get the solution of equations (2.141), we take a_1 times the second equation, a_2 the third equation, and so forth. Now making the sum with the system formed in that way, we get

$$\begin{aligned} a_m h(mT) + \ldots + a_1 h(T) + h(0) = {} & K_1 \left[a_m e^{s_1 mT} + \ldots + e^{s_1 T} a_1 + 1 \right] \\ & + K_2 \left[a_m e^{s_2 mT} + \ldots + e^{s_2 T} a_1 + 1 \right] \\ & + \ldots \\ & + K_m \left[a_m e^{s_m mT} + \ldots + e^{s_m T} a_1 + 1 \right] \end{aligned} \qquad (2.142)$$

Calling $e^{s_i T} = Z_i$, the above equation could be written as follows

$$\begin{aligned} a_m h(mT) + \ldots + a_1 h(T) + h(0) = {} & K_1 \left[a_m Z_1^m + \ldots + a_1 Z_1 + 1 \right] \\ & + K_2 \left[a_m Z_2^m + \ldots + a_1 Z_2 + 1 \right] \\ & + \ldots \\ & + K_m \left[a_m Z_m^m + \ldots + a_1 Z_m + 1 \right] \end{aligned} \qquad (2.143)$$

If $Z_1, Z_2, \ldots Z_m$ are the roots of the equation $a_m Z^m + a_{m-1} Z^{m-1} + \ldots + a_1 Z + 1 = 0$, the equation (2.143) becomes

$$a_m h(mT) + a_{m-1} h\Big((m-1)T\Big) + \ldots + a_1 h(T) + h(0) = 0 \qquad (2.144)$$

And, since this is true for any n, we can write the equation (2.144) as follows

$$a_m h(n+mT) + a_{m-1} h\Big(n + (m-1)T\Big) + \ldots + a_1 h(n+T) + h(n) = 0 \qquad (2.145)$$

Applying m times the equation (2.145) for different values of n, we form the following equation system

$$\begin{cases} a_m h(mT) + a_{m-1} h\Big((m-1)T\Big) + \ldots + a_1 h(T) = -h(0) \\ a_m h\Big((m+1)T\Big) + a_{m-1} h(mT) + \ldots + a_1 h(2T) = -h(T) \\ \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ a_m h(2mT) + a_{m-1} h\Big((2m-1)T\Big) + \ldots + a_1 h(mT) = -h\Big((m-1)T\Big) \end{cases} \qquad (2.146)$$

Therefore, the procedure for applying the Prony's method is as follows

- Make 2m measurements of the impulse response at equal intervals.
- With the measurements of the above item, find the values of the coefficients $a_1, a_2, \ldots a_m$ with the equations system (2.146).
- Find the roots of the equation $a_m Z^m + a_{m-1} Z^{m-1} + \ldots + a_1 Z + 1 = 0$
- With the equations system (2.141) find the values $K_1, K_2, \ldots K_m$.
- The poles of the system are given by

$$S_i = \frac{1}{T} \ln Z_i \qquad (2.147)$$

2.10.2 Conclusions -

This method provides a good approach to the solution of the fault isolation problem. An optimum sampling interval exists when Prony's technique is used, the same is a function of the highest frequency desired to be retrieved.

The detection condition is a very fast procedure for checking if further analysis with the data will be necessary to be performed.

2.10.3 Applications -

The application of this method covers only linear circuits/ systems, but the main disadvantages is that the system must be out of operation.

2.10.4 Computer Requirements -

The complex post-calculations involved in this method imposed some requirements on the computer required. These requirements increase if the circuit/system under test becomes more complex.

CHAPTER III

3.1 Comments on the Methods Described

Ten methods have been presented, three of them with two different techniques. In some cases, computer applications have been included.

The programs that have been used are given in the Appendix. A list of them is as follows:

Program No. 1 - Classification Method
Matrix Recognition Technique

Program No. 2 - Key Element Search Method
Nonlinear Technique (First Application)

Program No. 3 - Key Element Search Method
Nonlinear Technique (Second Application)

Program No. 4 - Key Element Search
Linear Technique

Program No. 5 - Iterative Method

Program No. 6 - Transfer Function Method
Bode Diagram Technique

Program No. 7 - Scalar Remnant Method

In order to have some idea about each program and also to know some of the computer requirements, a brief description of each one of them is given below.

3.1.1 Program No. 1

This program is the implementation of the classification method using the matrix recognition technique. The flowcharts that describe it are given in Figures 2.5, 2.6 and 2.7 (pages 2.9, 2.10 and 2.11). Up to 14 variables can be handled, with not more than 28 bounds associated with them. Therefore, if each variable has more than two bounds, the number of variables that will be possible to handle are given by the following inequality

$$\sum_{i=1}^{N} \leq 28 \qquad (5.1)$$

where bi designates the number of different bounds that belong to the variable i, and N gives the number of variables. The number of regions can go up to 31. If for some applications, we need to run programs not subject to these constraints, changes in the DIMENSION and FORMATs statements could be easily done.

The input to the program is the number of variables, number of regions, the bounds in each region and the input vectors.

The output is

- number of variables
- number of regions
- description of each region
- matrix H
- matrix K
- Matrix M
- for each input vector, its values, the output vector and the region/s of operation.

The computer used has been the UNIVAC 1108, and the memory requirements are as follows.

IBANK	(Decimal)	4149 words
DBANK	(Decimal)	4763 words
Total	(Decimal)	8906 words

3.1.2 Program No. 2

Since the program for the implementation of the key element search method - nonlinear technique depends on the circuit under test, the mentioned program has been developed for the circuit given in Figure 2.8 (page 2.21). The flowchart is indicated in Figure 2.9 (page 2.25).

The input to the program is the design values of the seven resistors and the four measurements values.

The output is

- design values of the seven resistors
- measurement values
- the values of S_j and the root for each component

The computer used has been the IBM 1130, and the memory requirements are as follows

VARIABLES	(Decimal)	192	bytes
PROGRAM	(Decimal)	1892	bytes
Total	(Decimal)	2084	bytes

3.1.3 Program No. 3

The description is exactly the same as the former program, except that the circuit under test is the one shown in Figure 2.10 (page 2.27).

The computer used has been the IBM 1130, and the memory requirements are as follows

VARIABLES	(Decimal)	160	bytes
PROGRAM	(Decimal)	1230	bytes
Total	(Decimal)	1390	bytes

For the type of circuit given, we can say that the memory requirements in function of the number of components will be a curve that looks like to one shown in Figure 3.1 - nonlinear case.

3.1.4 Program No. 4

Since the program for the implementation of the key element search method - linear technique depends on the circuit under test, the mentioned program has been developed for the circuit given in Figure 2.8 (page 2.21). The flowchart is indicated in Figure 2.11 (page 2.33).

The input to the program is the design values of the seven resistors and the four measurements values.

The output is

- design values of the seven resistors
- measurements values
- the values of S_j and M_j for each component
- the most likely fault component (This is indicated by printing three stars at the right of the component number.)

The computer used has been the IBM 1130, and the memory requirements are as follows

VARIABLES	(Decimal)	278 bytes
PROGRAM	(Decimal)	2100 bytes
Total	(Decimal)	2378 bytes

For the type of circuit under study, the curve that shows the memory required in function of the number of components is given by the one shown in Figure 3.1 (linear case).

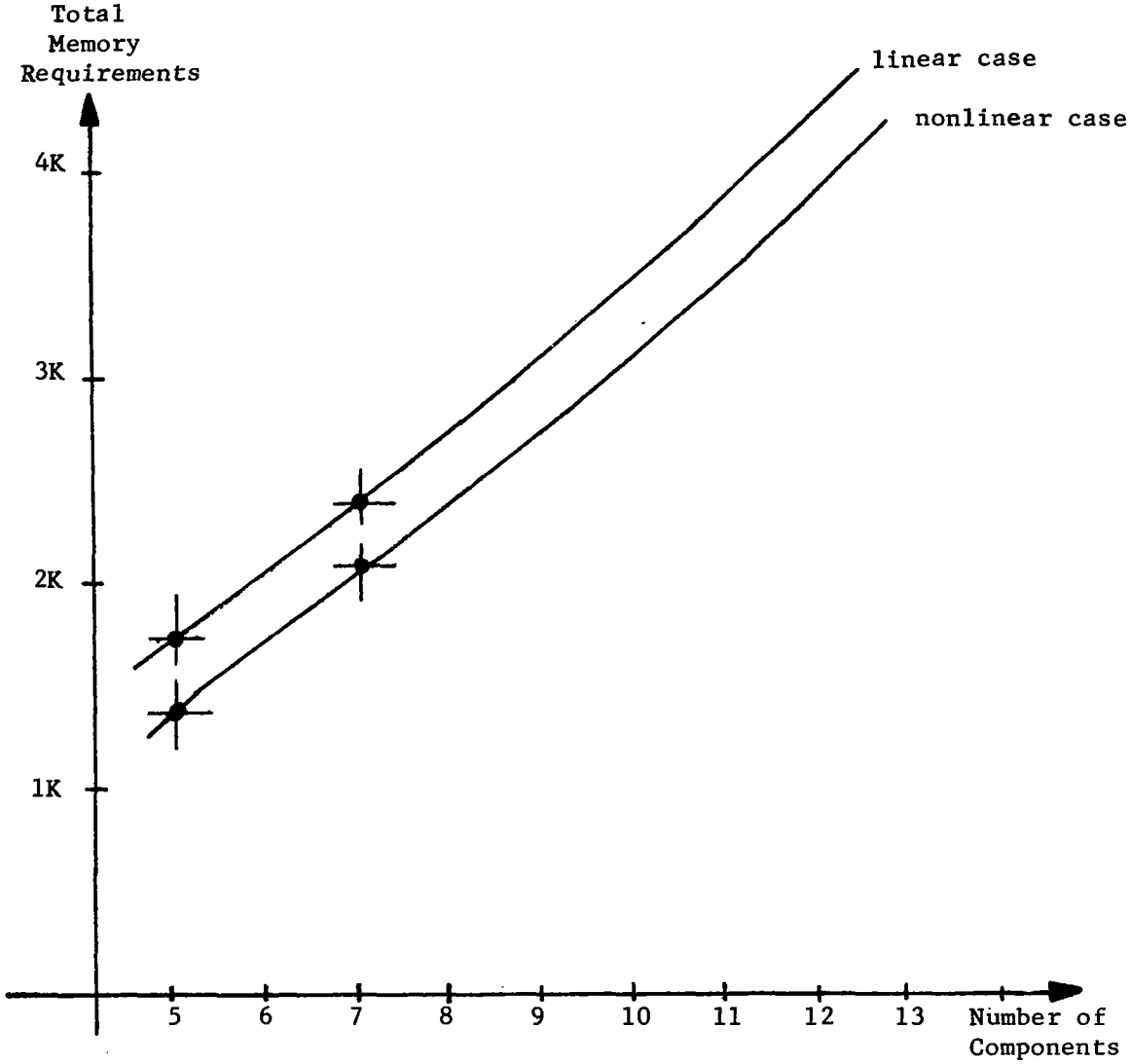


Figure 3.1 - Computer Requirements

3.1.5 Program No. 5

The steps needed in order to implement this program are given in the flowchart shown in Figure 2.13 (page 2.39). The circuit under study is indicated in Figure 2.14 (page 2.40). Up to ten measurements of O/I gain could be used in the program.

The input to the program is the number of frequencies, the values of them, the output/input gains, and the standard deviations.

The output is

- normal components
- frequencies
- measurements
- standard deviations
- normal O/I relations
- Matrix F_tA
- Matrix $(F_tA)^{-1}$
- product matrix $(F_tA)(F_tA)^{-1}$
- components values
- S index

The computer used has been the UNIVAC 1108, and the memory requirements are as follows

IBANK	(Decimal)	4680 words
DBANK	(Decimal)	2710 words
Total	(Decimal)	7390 words

In the above requirements, it is also included the subroutine MINV for the matrix inversion.

3.1.6 Program No. 6

This program has been developed for the circuit given in Figure 2.14 (page 2.40). The flow chart is indicated in Figure 2.19 (page 2.52).

The input to the program is the design values of the components (R_1, C_1, R_2 and C_2), and the detection condition for the zero (in our case was $\pm$ 0.003).

The output is

- the four break frequencies
- the five test frequencies
- the gain at each test frequency for different ranges of variation of each component
- a dictionary table

The computer used has been the UNIVAC 1108, and the memory requirements are as follows

IBANK	(Decimal)	4168 words
DBANK	(Decimal)	2312 words
Total	(Decimal)	6480 words

3.1.7 Program No. 7

Since the program for the implementation of the Scalar Remnant Method depends on the circuit under test, the mentioned program has been developed for the circuit given in Figure 2.26 (page 2.72).

The input to the program is the design values of the components (R_1, C_1, R_2 and C_2), the value of β and the changes in the components.

The output is

- the design values of the components
- the coefficients of the fraction expansion
- the values of the input and output, and the first and second derivatives of the same, in function of changes in the values of the components ($\pm$25%)
- allowed coefficients variation
- scalar remnant indexes

The computer used has been the UNIVAC 1108, and the memory requirements are as follows

IBANK	(Decimal)	5050 words
DBANK	(Decimal)	2547 words
Total	(Decimal)	7597 words

3.2 Conclusions

Since only recently we have started to evaluate the methods presented in this report, we cannot make a final judgement about these methods but simply state a few observations.

Nearly all methods require a mathematical model of the circuit or system under test which is very close to the actual system. The model equations are being used for calculating the nominal response which is compared with the measured response. The difference between these two responses, i.e., the error response, is then messaged to estimate the parameter deviations from their nominal values. Many methods assume that the physical system can be represented by linear models which is good for certain circuits and systems or within certain operating ranges; but in the majority of cases linear models are probably not sufficient. Hence, a prerequisite for accurate computation of these deviations is that accurate mathematical models of the physical system under test can be found. This points to the importance of modeling and the verification of models by simulation.

Many methods have been developed for circuits; however, some of them can also be applied to systems since the mathematical representation of electrical circuits and certain systems such as feedback control systems is very similar.

Some of the methods require large amounts of computations which might lead to large numerical errors so that they can be applied to small circuits or systems only. If the large amount of computations has to be done once for a particular hardware systems prior to the test and if the amount of on-line processing during the test itself is small, the extensive computational load may be tolerated, since it is required only once and not for each test.

None of the methods presented took into account any tolerance bounds of the components. In all methods, it is assumed that the nominal value of a component is a fixed number and would not change between allowable tolerance margins. The determination of the allowable tolerance margins at the outputs of a system for given tolerance margins of the parameters of subsystems and components within the system is a difficult task by itself. For this reason, we have started to work on sensitivity analysis methods to find out how parameter changes of components affect the output signal of a subsystem or system/circuit. This study can also lead to determining the optimum location of the test points. A computer program like MARSYAS* which can be expanded to include sensitivity analysis capability for any systems configuration would be helpful.

*Trauboth, H. and Prasad, N. - MARSYAS - A Software System for the Digital Simulation of Physical Systems - Proc. of SJCC, Atlantic City, May 1970.

Many methods assume that the measurements of the responses are not distorted by noise.

However, in practice some noise and instrumentation errors superimpose the measurement signal. For this reason, we are looking into filtering techniques such as Kalman filtering for better estimation of the measurements and for suppressing signal noise.

Since eventually the computer software methods have to interface with actual hardware, it is important to know what the constraints from the instrumentation and data acquisition system are.

We are just starting with an out-of-house contract to define these constraints such as sensor accuracy, sampling rate, noise level, location of test points, performance variable dynamics, etc., based on typical subsystems designs for Space Shuttle.

We feel that some of the methods could be modified and improved to make them into practical tools. Others which assume ideal situations are impractical in a real environment. However, we think that the application of modern techniques of control systems theory, information theory, and network theory could lead to better fault isolation methods which are practical and efficient. After thoroughly evaluating the methods presented in the report, we will concentrate on the development of a few methods which are promising for practical applications in automatic testing and automatic ground/on-board checkout.

3.3 Future Research Needed in this Area

The future research needed in this area would be

- Instrumentation requirements for fault isolation computer methods.
- Sensitivity analysis of fault isolation computer methods
- Better estimation of measurements values
- Functional Simulation of the SSV and SS
- Additional fault isolation computer methods
- New measurements techniques
- Mathematical models of the SSV and SS
- Logistic/operational studies
- Fault isolation method/s for the SSV and SS

A flowchart on these steps is shown in Figure 3.2.

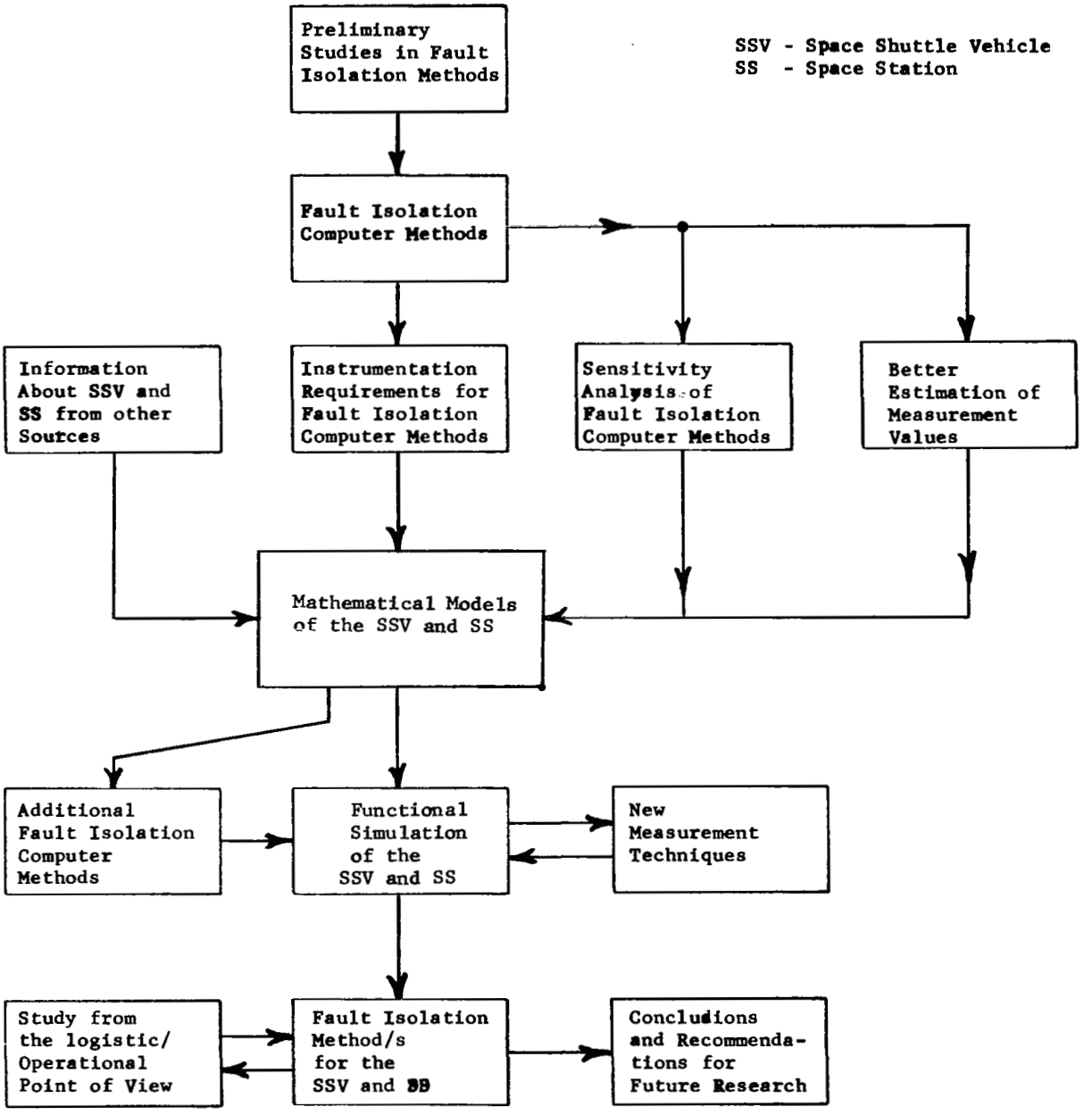

Preliminary Studies in Fault Isolation Methods
SSV - Space Shuttle Vehicle
SS - Space Station
Fault Isolation Computer Methods
Information About SSV and SS from other Sources
Instrumentation Requirements for Fault Isolation Computer Methods
Sensitivity Analysis of Fault Isolation Computer Methods
Better Estimation of Measurement Values
Mathematical Models of the SSV and SS
Additional Fault Isolation Computer Methods
Functional Simulation of the SSV and SS
New Measurement Techniques
Study from the logistic/ Operational Point of View
Fault Isolation Method/s for the SSV and SS
Concludions and Recommenda- tions for Future Research

Figure 3.2

4.1 References

1. Allen Robert J. - Failure Prediction Employing Continuous Monitoring Techniques - IEEE Transactions on Aerospace Support Conference Proc. 1963

2. *Barry, R.F., Fisher, R.S., and Mattison,R.R. - Programmed Algorithm For Test Point Selection and Fault Isolation - Vol.1 - RCA August 1966 AD-487520

3. Barry, R.F., Fisher, R.S., and Mattison, R.R. - Programmed Algorithm For Test Point Selection and Fault Isolation - Vol. 2 - RCA August 1966 AD-487521

4. *Barry, R.F. - Fault Isolation by Parameter Identification - Listed In Automatic Support Systems Symposium - RCA - Aerospace Systems Division October 1968

5. *Bayly, E.J., And Leradi, V.S. - A Method of Dynamic System Testing - Proc. Eighth Conf. on Military Electronics - September 1964

6. Beckman, C., Adler, J., Bedrosian, S.D., Berkowitz, R.S., and Chen, T.C. - Study of Piece Part Isolation of Computer Logic Vol 5 - Penna Univ. June 1964 AD-615337

7. *Beckman, C., Adler, J., Bedrosian, S.D., Berkowitz, R.S., and Chen, T.E. - Study of Piece Part Isolation of Computer Logic Vol 4 - Penna Univ. June 1964 AD-136615

8. *Berkowitz, R.S., and Wexelbrat, R.L. - Statistical Considerations in Element Value Solutions - IRE Transactions on Military Electronics 1962

9. *Berkowitz, R.S. - Conditions For Network Element Value Solvability - IRE Transactions on Circuit Theory. March 1962

10. *Berkowitz R.S., and Krishnaswamy - Computer Techniques for Solving Electric Circuits for Fault Isolation - IEEE Transactions ASCP

11. Boor, P.M. and Grimmer, R.L. - A Computerized Data Acquisition System for Real Time Fault Isolation - Lockheed Aircraft Service Company November 1968

12. *Brown, F.D., McAllister, N.F., and Perry, R.P. - An Application of Inverse Probability to Fault Isolation - IRE Transactions on Military E.

13. *Buschsbaum, L., Dunning, M., Hannom, T.J.B., and Math, L. - Investigation of Fault Diagnosis by Computational Methods - Remington Rand Univac
May 1964 AD-601204

14. Cabra, D - On-Board Checkout Systems Concept and Philosophy - NASA Manned Spacecraft Center - Automatic Support Systems for AD. Maint.
November 1968

15. Chesler, L., and Turn, R. - The Monitoring Task in Automated Checkout of Space Vehicles - The Rand Corporation - Memorandum RM-4678 NASA
September 1965

16. Chesler, L.G. and Turn, R. - Some Aspects of Man-Computer Communication in Active Monitoring of Automated Checkout-The Rand Corporation
March 1967 AD-648553

17. *Chorzel, J.P., Thompson, J.R., and Myers, R.G. - System Parameter Mearurement Using Transient Response Sampling

18. Costanza, J.L. and Osborne, R.L. - Fault Detection and Diagnosis, An Energy Point of View, Proceedings of the Automatic Support Systems Symposium for Advanced Maintainability, June 1965.

19. Critique of Sogomonyan's Method of Monitoring Operability and Isolating Faults

20. Curtis, G.C., and Mash, J.T. - An Airborne Manual/Automatic Malfunction Detection System - Lockheed Georgia Company - Marietta, Georgia

21. Dorrough, D.C. - A Methodical Approach to Analyzing and Synthesizing A Self Repariing Computer - IEEE Transaction on Computer - Volume C-18
January 1969

22. Drezner, S.M. and Gatt, O.T. - Computer - Assisted Countdown - The Rand Corporation for NASA - Memorandum RM-4565-NASA
May 1965

23. *Dynamic Testing - A Technique For Automatic Checkout of Closed - Loop Systems - Contract NASW410-General Electric - Apollo Systems Department
April 1968

24. Even, S., and Lempel, A. - On A Problem of Diagnosis - Transactions On Circuti Theory
September 1967

25. *Finkel, S.I. - Nilson, R.N., and Clair, E.S.T. - A Mathematical Automatic Fault Isolation In A Complex System - Vitro Laboratories May 1963 AD-413305

26. Firstman, Sidney I, and Gluss, Brian - Optimum Search Routines for Automatic Fault Location - 15th Meeting ORSA - Washington DC May 1959

27. Firstman, Sidney I - Current Concepts and Issues of In-Space Support Rand Corporation - Santa Monica, California August 1964 AD-606580

28. *Fisch, S.M. and Brigida, G.R. - Computer Algorithm for Fault Isolation And Test Point Selection - RCA - Burlington, Mass. December 1964 AD-613960

29. Furst, U.R. - Automatic Built-In Test of Advanced Avionics Systems - Part II - Design Techniques for Automatic Built In Test - TAES July 1966

30. Gluss, Brian - An Optimum Policy For Detecting A Faulty In A Complex System - Presented at the 15th Metting ORSA - Washington, DC May 1959

31. Goodman, David M. - Nondestructive Testing of Electronic Circuits With Fiber Optic Scientillators Vidicon Data Sampling and Pat.Reg. 1966

32. Goodman, David M. - A Review of the State of the Art in Automatic Electronic Test Equipment - Reprint Paper No. 389 - International Conv. 1966

33. Hennebuckle, G.D., and Spann, R.M. - Diagnosis of Single Gate Failure Combinational Circuits - IEEE Transactions on Computer - Volume C-18 March 1969

34. Jaegly, Robert L. - Test Procedure Validation by Computer Simulation AIAA Second Flight Test Simulation and Support Conference - LA-Cal. 1968

35. Johnson, J.L., McKenzie, H.R., and Moore, D.H. - On-Board In Flight Checkout Evaluation - IBM Federal Systems Division June 1968 AD-883685

36. Johnson, R.A., Kletsky, E., and Brule, J. - Diangosis of Equipment Failures - Syracuse University Research Institute April 1959 AD-213876

37. Johnson, James L. - On-Board In Flight Checkout Part I - IBM Federal Systems Division
December 1966 AD-807534

38. Kirkman, Robert - The Relative Effectiveness of Internally Programmed and Sequential Programmed Machine for Automatic Checkout - IEEE T.ASCP

39. *Kraabel, Paul B. - Power Spectra Analysis as a Means of On-line Checkout

40. *Kranton, J., and Libenson, A. - A Pattern Recognition Approach to Fault Isolation - IEEE Transactions on Aerospace Support Conference Proc.

41. Levadi, Victor S. and Turner, Lawrence D. - Fault Diagnosis by White Noise Techniques - Honeywell, Inc. - Minneapolis, Minnesota May 1966

42. Levaldi, V. S. - Pattern Recognition Applied to Fault Detection - IEEE Conference on Military Electronics, 1965.

43. *Levenstein, H. - Use Difference Equations to Calculate Frequency Response from Transient Response-Control Engineering, April 1957.

44. Lohse, Robert and Lauler, Louis - VADE - A System for Real-Time Space Vehicle Checkout and Launch Monitoring - IEEE Transactions on ASCP, 1969.

45. Lux, Paul A. - An Adaptive Machine for Fault Detection or Process Control - National Electronics Conference - Sandia Corporation, October 1966.

46. Lux, P. A., and Drake, K. W. - Fault Detection with a Simple Adaptive - IEEE Transactions on Industrial Electronics and Control Instrument, December 1967.

47. Mace, A. E., Pesat, R. N., Minkoff, R. T., Wertz, J. B., and Lipis, A. H. - Formulation of System Status Control Techniques, September 1963, AD-421743.

48. Mash, J. T. - An Airborne Manual/Automatic Malfunction Detection System - Lockheed Georgia Company, Marietta, Georgia, 1968.

49. Mast, L. T. - Prelaunch Checkout in the 1970's - The Rand Corporation, Santa Monica, California, April 1965, AD-614410.

50. Mast, Larry T. - Growth of Automation in Prelaunch Checkout for Space Vehicles - The Rand Corporation, Santa Monica, California, April 1965, AD-614410.

51. Mast, Lawrence T. - Automatic Checkout of Space Systems - Rand Corporation, Santa Monica, California, May 1967, AD-652458.

52. *Math, L., Buchsbaum, L., and Hannom, T. J. B. - Investigation of Fault Diagnosis by Computer Methods for Microcircuits - Univac, Blue Bell, Pennsylvania, November 1965, AD-623951.

53. McSweeney, K. F. - Malfunction Detection System for Advanced Spacecraft - IEEE Transactions on Aerospace and Electronics Systems, January 1966.

54. Moon, Warren D. - Periodic Checkout and Associated Errors - IEEE Transactions on Aerospace, Volume 2, RCA, Burlington, Mass., April 1964.

55. Newcomb, Robert W. - Linear Multiport Synthesis - McGraw-Hill Book Company, 1966.

56. *Osborne, Robert L. - The Detection and Diagnosis of Malfunction in Energy Manipulating Systems - University of California, Berkley, December 1967.

57. Programming, Integration and Checkout of the Advanced Radar Traffic Control System - Univac Division of Sperry Rand Corporation, April 1965, AD-622865.

58. *Puri, N. N., and Weygandt, C. N. - Transfer Function Tracking of Linear Time Varying by Means of Auxiliary Simple Lag Networks - Joint ACC, 1963.

59. Stahl, Walter J. - Dynamic Fault Diagnosis Techniques - Scully International, Inc., Downers, Grove, Illinois, September 1968.

60. *Stahl, Walter J., and Maenfaa, John H. - Development of Advanced Dynamic Fault Diagnosis Techniques - Scully International, Inc., May 1967, AD-814457.

61. Swain, A. D., and Whol, J. G. - Factors Affecting Degree of Automation in Test and Checkout Equipment - The Rand Corporation, March 1961, AD-257887.

62. Taylor, J. W. - Automatic Checkout Systems for Combat Vehicles - Frankford Arsenal, Philadelphia, Pennsylvania, April 1964, AD-823067.

63. *Teasdale, A. R., and Reynolds, J. B. - Two Ways to get Frequency Response from Transient Data - Control Engineering, October 1955.

64. Townes, J. R., Dwyer, S. J., McLaren, R. W., and Zobrist, G. W. - Linear Techniques for Binary Fault Isolation, - University of Missouri.

65. Ulrich, Earnst G. - The Evaluation of Digital Diagnosis Programs Through Digital Simulation - IEEE Conference, Publication No. 307, 1967.

66. *Valstar, J. E. - In Flight Dynamic Checkout - IEEE Transactions on Aerospace Support Conference, August 1963.

67. *Valstar, J. E. - Fault Isolation of Electronic Circuits Using Transfer Function Methods - AF Aero Propulsion Lab, Wright Patterson AF, Ohio, October 1964, AD-608176.

APPENDIX

5.1 Program No. 1*

```
C       CLASSIFICATION METHOD
        INTEGER H(15,28),X(15)
        DIMENSION K(31,28),NUM(31),IVAL(28),KK(28,31),MM(15,31)
   10 FORMAT(2I2,3I4,2A2)
   11 FORMAT(20I4/)
   12 FORMAT(45X,2(I2,'/'),I4/5X,'NUMBER OF VARIABLES =',I4/5X,'NUMBER O
     2F REGIONS  =',I4/)
   13 FORMAT(5X,'REGION',I3,24X,'X',I2)
   14 FORMAT('+',29X,I4,1X,A2)
   15 FORMAT('+',4IX,A2,I3)
   16 FORMAT(38X,'X',I2)
   17 FORMAT(/)
   18 FORMAT('1',44X,2(I2,'/'),I4/)
   19 FORMAT(2X,28I4)
   20 FORMAT(27I3)
   21 FORMAT(5X,'REGION',I3)
   22 FORMAT(2X,'MATRIX H'//)
   23 FORMAT(2X,'MATRIX K'//)
   24 FORMAT(2X,'MATRIX M'//)
   25 FORMAT('+',14X,'- REGION',I3)
      MM1=5
      MM2=6
      READ(MM1,10) M1,M2,M3,N,M,GA,CH
      MO=2*M
      DO 100 I=1,N
  100 READ(MM1,11) (K(I,J),J=1,MC)
C       STATEMENT OF THE PROBLEM
      WRITE(MM2,12) M1,M2,M3,M,N
      IL=1
      DO 101 I=1,N
      WRITE(MM2,13) I,IL
      IF (K(I,1)) 102,103,102
  102 WRITE(MM2,14) K(I,1),CH
  103 IF (K(I,2)) 104,105,104
  104 WRITE(MM2,15) GA,K(I,2)
  105 DO 106 J=2,M
      WRITE(MM2,16) J
      IF (K(I,2*J-1)) 107,108,107
  107 WRITE(MM2,14) K(I,2*J-1),CH
  108 IF (K(I,2*J)) 109,106,109
  109 WRITE(MM2,15) GA,K(I,2*J)
  106 CONTINUE
  101 WRITE(MM2,17)
```

*UNIVAC 1108

```
C       BOUNDS OF EACH VARIABLE
        KK1=0
        IND=0
        IOD=0
        DO 110 J=1,MO,2
        IND=IND+1
        JRI=1
  303 IF (K(JRI,J)) 301,301,302
  301 JRI=JRI+1
        GO TO 303
  302 IVAL (IND)=K(JRI,J)
        IUD=IND-IOD
        KOK=0
  115 DO 111 L=1,N
        MMM=KOK+J
        IF(K(L,MMM)) 111,111,300
  300 DO 191 LL=IUD, IND
        IF(K(L,MMM)-IVAL(LL)) 191,111,191
  191 CONTINUE
        IND=IND+1
        IVAL (IND)=K(L,MMM)
  111 CONTINUE
        IF(KOK) 113,113,114
  113 KOK=KOK+1
        GO TO 115
  114 KK1=KK1+1
        NUM(KK1)=IND-IOD
        IOD=IND
  110 CONTINUE
C       MATRIX H
        JMAX=IND
        IMAX=M+1
        DO 116 J=1,JMAX
        DO 117 I=1,IMAX
  117 H(I,J)=0
  116 H(IMAX,J)=-IVAL (J)
        K1=1
        K2=NUM(1)
        DO 118 L=1,KK1
        DO 119 J=K1,K2
  119 H(L,J)=1
        K1=K2+1
  118 K2=NUM(L+1)+K2
        WRITE(MM2,18) M1,M2,M3
        WRITE(MM2,22)
        DO 120 I=1,IMAX
  120 WRITE(MM2,19) (H(I,J),J=1,JMAX)
C       MATRIX K
        DO 140 J=1,N
        DO 140 I=1,JMAX
```

```
    140 KK(I,J)=1
C       SIGN CORRECTION
        DO 130 I=1,N
        IL=1
        DO 130 J=1,JMAX
    135 IF (H(IL,J)) 131,131,132
    132 IF (K(I,2*IL)) 799,799,800
    799 IF(K(I,2*IL-1)+1+H(IMAX,J)) 133,130,130
    800 IF(K(I,2*IL)-1+H(IMAX,J)) 133,130,130
    133 KK(J,I)=-KK(J,I)
        GO TO 130
    131 IL=IL+1
        GO TO 135
    130 CONTINUE
        WRITE(MM2,18) M1,M2,M3
        WRITE(MM2,23)
        DO 141 I=1,JMAX
    141 WRITE(MM2,19) (KK(I,J),J=1,N)
C       MATRIX M
        DO 142 I=1,IMAX
        DO 142 J=1,N
        MM(I,J)=0
        DO 142 L=1,JMAX
    142 MM(I,J)=MM(I,J)+H(I,L)*KK(L,J)
        WRITE(MM2,18) M1,M2,M3
        WRITE(MM2,24)
        DO 143 I=1,IMAX
    143 WRITE(MM2,19) (MM(I,J),J=1,N)
C       OUTPUT VECTOR
        WRITE(MM2,18) M1,M2,M3
    150 READ(MM1,20) (X(I),I=1,M)
        X(M+1)=1
        DO 144 I=1,N
        NUM(I)=0
        DO 144 J=1,IMAX
    144 NUM(I)=NUM(I)+X(J)*MM(J,I)
        WRITE(MM2,19) (X(J),J=1,M)
        WRITE(MM2,19) (NUM(J),J=1,N)
C       REGION DETERMINATION
        MAX=0
        DO 145 J=1,N
        IF (NUM(J)-MAX) 145,145,146
    146 MAX=NUM(J)
        LL=J
    145 CONTINUE
        WRITE(MM2,21) LL
        DO 701 J=1,N
        IF(LL-J) 702,701,702
```

```
702 IF(NUM(J)-MAX) 701,703,701
703 WRITE(MM2,25) J
701 CONTINUE
    WRITE(MM2,17)
    GO TO 150
    END
```

5.2 Program No. 2*

```
C       KEY ELEMENT SEARCH
C       NONLINEAR TECHNIQUE

        DIMENSION G(4),C(4),D(4),S(7),XX(7)
   10   FORMAT (2I2,I4)
   11   FORMAT (7F10.4/4F10.4)
   12   FORMAT (10X,I3,2F25.8)
   13   FORMAT (60X,2(I2,'/'),I4/5X,'R1=',F9.4,3X,'G1=',F9.4/5X,'R2=',F9.4,
       23X,'G2=',F9.4/5X,'R3=',F9.4,3X,'G3=',F9.4/5X,'R4=',F9.4,3X,'G4=',
       2F9.4/5X,'R5=',F9.4/5X,'R6=',F9.4/5X,'R7=',F9.4)
   14   FORMAT (/////)
   15   FORMAT ('1')
        MM1=2
        MM2=3
        READ(MM1,10) M1,M2,M3
 1000   READ(MM1,11) R1,R2,R3,R4,R5,R6,R7,(G(L),L=1,4)
        KK1=0
        WRITE(MM2,13) M1,M2,M3,R1,G(1),R2,G(2),R3,G(3),R4,G(4),R5,R6,R7
        KKO=1
        KKK=1
        A=0.0
        B=(R2+R3)*(R4+R5+R6)+R4*(R5+R6)
        C(1)=B
        D(1)=R2*R3*(R4+R5+R6)+R2*R4*(R5+R6)
  111   KK1=KK1+1
        ANUM=0.0
        DEN1=0.0
        DEN2=0.0
C       CHOOSING THE ROOT
        KLS=0
        AMMS=10.0
        X2=0.0
        Z1=0.0
        DO 707 L=KKO,KKK
        ALA=(A*D(L)-B*C(L))/((D(L)+X2*C(L))**2)
  707   Z1=G(L)*ALA-((A*X2+B)/(C(L)*X2+D(1)))*ALA+Z1
  616   KLS=KLS+1
  612   X2P=X2+AMMS
        Z2=0.0
        DO 610 L=KKO,KKK
        OLA=(A*D(L)-B*C(L))/((D(L)+X2*C(L))**2)
  610   Z2=G (L)*OLA-((A*X2P+B)/(C(L)*X2P+D(L)))*OLA+Z2
        TEST=Z1*Z2
        X2=X2P
        AZ1=Z1
        Z1=Z2
        IF (TEST) 611,611,612
```

*IBM 1130

```
611  X2=X2-AMMS
     Z1=AZ1
     GO TO (613,614,815,816,110),KLS
613  AMMS=1.0
     GO TO 616
614  AMMS=0.1
     GO TO 616
815  AMMS=0.01
     GO TO 616
816  AMMS=0.001
     GO TO 616
110  XX(KK1)=X2
     S(KK1)=0.0
     DO 130 L=KKO,KKK
130  S(KK1)=S+(G(L)-(A*X2+B)/(C(L)*X2+D(L)))**2
     GO TO (101,102,103,104,105,106,107),KK1
101  KKK=4
     A=R4+R5+R6
     B=(R3+R4)*(R5+R6)+R3*R4
     C(1)=(R1+R3)*(R4+R5+R6)+R4*(R5+R6)
     C(2)=R4*(R5+R6)
     C(3)=R7*(R5+R6)+R4*(R7+R6)+R5*R6
     C(4)=R6*(R4+R5)
     D(1)=R1*R3*(R4+R5+R6)+R1*R4*(R5+R6)
     D(2)=R3*R4*(R5+R6)
     D(3)=R7*(R3+R4)*(R5+R6)+R3*R4*(R7+R6)+R5*R6*(R3+R4)
     D(4)=(R3+R4)*R5*R6+R3*R4*R6
     GO TO 111
102  B=(R2+R4)*(R5+R6)+R2*R4
     C(1)=(R1+R2)*(R4+R5+R6)
     C(2)=R4*(R5+R6)
     C(3)=R6*(R4+R5+R7)+R7*R4+R5*R7
     C(4)=R6*(R4+R5)
     D(1)=R1*R2*(R4+R5+R6)+(R1+R2)*(R5+R6)*R4
     D(2)=R2*R4*(R5+R6)
     D(3)=(R2+R4)*(R5*R7+R7*R6+R6*R5)+R2*R4*(R6+R7)
     D(4)=(R2+R4)*R5*R6+R2*R4*R6
     GO TO 111
103  A=R2+R3+R5+R6
     B=(R2+R3)*(R5+R6)
     C(1)=R1*(R2+R3+R5+R6)+R2*(R3+R5+R6)
     C(2)=(R2+R3)*(R5+R6)
     C(3)=R5*R7+R7*R6+R6*R5+(R6+R7)*(R2+R3)
     C(4)=R6*(R2+R3+R5)
     D(1)=(R5+R6)*(R1*R2+R2*R3+R3*R1)
     D(2)=0.0
     D(3)=(R2+R3)*(R5*R7+R7*R6+R6*R5)
     D(4)=(R2+R3)*R5*R6
     GO TO 111
```

```
104   A=R2+R3+R4
      B=(R4+R6)*(R2+R3)+R4*R6
      C(1)=R1*(R2+R3)+R4*(R1+R2)+R2*R3
      C(2)=R4*(R2+R3)
      C(3)=(R6+R7)*(R2+R3+R4)
      C(4)=(R2+R3+R4)*R6
      D(1)=(R4+R6)*R1*(R2+R3)+R4*R6*(R1+R2)+R2*R3*(R4+R6)
      D(2)=R4*R6*(R2+R3)
      D(3)=R6*R7*(R2+R3+R4)+R4*R7*(R2+R3)+R4*R6*(R2+R3)
      D(4)=R4*R6*(R2+R3)
      GO TO 111
105   B=(R4+R5)*(R2+R3)+R4*R5
      C(1)=R1*(R2+R3)+R4*(R1+R2)+R2*R3
      C(2)=R4*(R2+R3)
      C(3)=(R5+R7)*(R2+R3+R4)+R4*(R2+R3)
      C(4)=(R2+R3+R4)*R5+R4*(R2+R3)
      D(1)=(R4+R5)*R1*(R2+R3)+R4*R5*(R1+R2)+R2*R3*(R4+R5)
      D(2)=R4*R5*(R2+R3)
      D(3)=R5*R7*(R2+R3+R4)+R4*R7*(R2+R3)
      D(4)=0.0
      GO TO 111
106   KKO=3
      KKK=3
      A=0.0
      B=(R2+R3)*(R4+R5+R6)+R4*(R5+R6)
      C(3)=(R5+R6)*(R2+R3+R4)+R4*(R2+R3)
      D(3)=R5*R6*(R2+R3+R4)+R4*R6*(R2+R3)
      GO TO 111
107   WRITE(MM2,14)
      DO 307 L=1,7
307   WRITE(MM2,12)L,S(L),XX(L)
      WRITE(MM2,15)
      GO TO 1000
      END
```

5.3 Program No. 3*

```
C       KEY ELEMENT SEARCH
C       NONLINEAR TECHNIQUE

        DIMENSION G(3),C(3),D(3),S(5),XX(5)
   10   FORMAT (2I2,I4,A1)
   11   FORMAT (8F10.4)
   12   FORMAT (10X,I3,2F25.8)
   13   FORMAT (60X,2(I2,'/'),I4/5X,'R1=',F9.4,3X,'G1=',F9.4/5X,'R2=',F9.4,
       23X,'G2=',F9.4/5X,'R3=',F9.4,3X,'G3=',F9.4/5X,'R4=',F9.4/4X,'R5=',
       3F9.4)
   14   FORMAT (/////)
   15   FORMAT ('1')
   16   FORMAT ('+',12X,3A1)
        MM1=2
        MM2=3
        READ(MM1,10) M1,M2,M3,STAR
 1000   READ(MM1,11) R2,R3,R4,R5,R6,(G(L),L=1,3)
        KK1=0
        WRITE(MM2,13) M1,M2,M3,R2,G(1),R3,G(2),R4,G(3),R5,R6
        A=R4+R5+R6
        B=(R3+R4)*(R5+R6)+R3*R4
        C(1)=R3*(R4+R5+R6)+R4*(R5+R6)
        C(2)=R4*(R5+R6)
        C(3)=R4*R6+R5*R6
        D(1)=0.0
        D(2)=R3*R4*(R5+R6)
        D(3)=R3*R4*R6+R5*R6*(R3+R4)
   111  KK1=KK1+1
        ANUM=0.0
        DEN1=0.0
        DEN2=0.0
C       CHOOSING THE ROOT
        KLS=0
        AMMS=10.0
        X2=0.0
        Z1=0.0
        DO 707 L=1,3
        ALA=(A*D(L)-B*C(L))/((D(L)+X2*C(L))**2)
   707  Z1=G(L)*ALA-((A*X2+B)/(C(L)*X2+D(L)))*ALA+Z1
   616  KLS=KLS+1
   612  X2P=X2+AMMS
        Z2=0.0
        DO 610 L=1,3
        OLA=(A*D(L)-B*C(L))/((D(L)+X2P*C(L))**2)
   610  Z2=G(L)*OLA-((A*X2P+B)/(C(L)*X2P+D(L)))*OLA+Z2
        TEST=Z1*Z2
        X2=X2P
        AZ1=Z1
        Z1=Z2
        IF (TEST) 611,611,612
```

*IBM 1130

```
611   X2=X2-AMMS
      Z1=AZ1
      GO TO (613,614,815,816,110),KLS
613   AMMS=1.0
      GO TO 616
614   AMMS=0.1
      GO TO 616
815   AMMS=0.01
      GO TO 616
816   AMMS=0.001
      GO TO 616
110   XX(KK1)=X2
      S(KK1)=0.0
DO 130 L=1,3
130   S(KK1)=S(KK1)+(G(L)-(A*X2+B)/(C(L)*X2+D(L)))**2
      GO TO (101, 102,103,104,105),KK1
101   B=(R2+R4)*(R5+R6)+R2*R4
      C(1)=R2*(R4+R5+R6)
      C(2)=R4*(R5+R6)
      C(3)=R5*(R4+R5)
      D(1)=R2*R4*(R5+R6)
      D(2)=R2*R4*(R5+R6)
      D(3)=(R2+R4)*R5*R6+R2*R4*R6
      GO TO 111
102   A=R2+R3+R5+R6
      B=(R2+R3)*(R5+R6)
      C(1)=R2*(R3+R5+R6)
      C(2)=(R2+R3)*(R5+R6)
      C(3)=R6*(R2+R3+R5)
      D(1)=(R5+R6)*R2*R3
      D(2)=0.0
      D(3)=(R2+R3)*R5*R6
      GO TO 111
103   A=R2+R3+R4
      B=(R4+R6)*(R2+R3)+R4*R6
      C(1)=R4*R2+R2*R3
      C(2)=R4*(R2+R3)
      C(3)=(R2+R3+R4)*R6
      D(1)=R4*R6*R2+R2*R3*(R4+R6)
      D(2)=R4*R6*(R2+R3)
      D(3)=R4*R6*(R2+R3)
      GO TO 111
104   B=(R4+R5)*(R2+R3)+R4*R5
      C(1)=R4*R2+R2*R3
      C(2)=R4*(R2+R3)
      C(3)=(R2+R3+R4)*R5+R4*(R2+R3)
      D(1)=R2*R4*R5+R2*R3*(R4+R5)
      D(2)=R4*R5*(R2+R3)
      D(3)=0.0
      GO TO 111
```

```
105  WRITE(MM2,14)
     AMAX=100.0
     JL=0
     DO 333 J=1,5
     IF(S(J)-AMAX) 334,333,333
334  AMAX=S(J)
     JL=J
333  CONTINUE
     DO 307 L=1,5
     WRITE(MM2,12) L,S(L),XX(L)
     IF(JL-L) 307,388,307
388  WRITE(MM2,16) STAR,STAR,STAR
307  CONTINUE
     WRITE(MM2,15)
     GO TO 1000
     END
```

5.4 Program No. 4*

```
C       KEY ELEMENT SEARCH
C       LINEAL TECHNIQUE

        DIMENSION G(4),Q(4,7),C(4),D(4),VF(4),CM(4),RES(7,2)
   10   FORMAT(2I2,I4,A1)
   11   FORMAT(7F10.4/4F10.4)
   12   FORMAT(4F10.4)
   13   FORMAT(60X,2(I2,'/'),I4/5X,'R1=',F9.4,3X,'G1=',F9.4,3X'GM1=',F9.4/
       25X,'R2=',F9.4,3X,'G2=',F9.4,3X,'GM2=',F9.4/5X,'R3=',F9.4,3X,'G3=',
       3F9.4,3X,'GM3=',F9.4/5X,'R4=',F9.4,3X,'G4=',F9.4,3X,'GM4=',F9.4/5X,
       3'R5=',F9.4/5X,'R6=',F9.4/5X,'R7=',F9.4///9X,'J',21X,'MJ',26X,'SJ'/
       2)
   14   FORMAT(5X,I5,2F28.8)
   15   FORMAT('1')
   16   FORMAT('+',10X,3A1)
        MM1=2
        MM2=3
        READ(MM1,10) M1,M2,M3,STAR
        READ(MM1,11) R1,R2,R3,R4,R5,R6,R7,(G(L),L=1,4)
        KK1=0
        A=0.0
        B=(R3+R2)*(R4+R5+R6)+R4*(R5+R6)
        C(1)=B
        D(1)=R2*R3*(R4+R5+R6)+R2*R4*(R5+R6)
        DO 200 L=2,4
   200  C(L)=0.0
        D(2)=(R2+R3)*(R5+R6)*R4
        D(3)=(R2+R3+R4)*(R5*R7+R7*R6+R6*R5)+(R2+R3)*(R6*R7)*R4
        D(4)=R6*R5*(R2+R3+R4)+R4*R6*(R2+R3)
        X1=R1
   111  KK1=KK1+1
        DO 201 L=1,4
   201  Q(L,KK1)=(A*D(L)-B*C(L))/((D(L)+X1*C(L))**2)
        GO TO (101,102,103,104,105,106,107),KK1
   101  A=R4+R5+R6
        B=(R3+R4)*(R5+R6)+R3*R4
        C(1)=(R1+R3)*(R4+R5+R6)+R4*(R5+R6)
        C(2)=R4*(R5+R6)
        C(3)=R7*(R5+R6)+R4*(R7+R6)+R5*R6
        C(4)=R6*(R4+R5)
        D(1)=R1*R3*(R4+R5+R6)+R1*(R5+R6)
        D(2)=R3*R4*(R5+R6)
        D(3)=R7*(R3+R4)*(R5+R6)+R3*R4*(R7+R6)+R5*R6*(R3+R4)
        D(4)=(R3+R4)*R5*R6+R3*R4*R6
        X1=R2
        GO TO 111
```

*IBM 1130

5.12

```
102  B=(R2+R4)*(R5+R6)+R2*R4
     C(1)=(R1+R2)*(R4+R5+R6)
     C(2)=R4*(R5+R6)
     C(3)=R6*(R4+R5+R7)+R2*R4+R5*R7
     C(4)=R6*(R4+R5)
     D(1)=R1*R2*(R4+R5+R6)+(R1+R2)*(R5+R6)*R4
     D(2)=R2*R4*(R5+R6)
     D(3)=(R2+R4)*(R5+R7+R7*R6+R6*R5)+R2*R4*(R6+R7)
     D(4)=(R2+R4)*R5*R6+R2*R4*R6
     X1=R3
     GO TO 111
103  A=R2+R3+R5+R6
     B=(R2+R3)*(R5+R6)
     C(1)=R1*(R2+R3+R5+R6)+R2*(R3+R5+R6)
     C(2)=(R2+R3)*(R5+R6)
     C(3)=R5*R7+R7*R6+R6*R5+(R6+R7)*(R2+R3)
     C(4)=R6*(R2+R3+R5)
     D(1)=(R5+R6)*(R1*R2+R2*R3+R3*R1)
     D(2)=0.0
     D(3)=(R2+R3)*(R5*R7+R7*R6+R6*R5)
     D(4)=(R2+R3)*R5*R6
     X1=R4
     GO TO 111
104  A=R2+R3+R4
     B=(R4+R6)*(R2+R3)+R4*R6
     C(1)=R1*(R2+R3)+R4*(R1+R2)+R2*R3
     C(2)=R4*(R2+R3)
     C(3)=(R6+R7)*(R2+R3+R4)
     C(4)=(R2+R3+R4)*R6
     D(1)=(R4+R6)*R1*(R2+R3)+R4*R6*(R1+R2)+R2*R3*(R4+R6)
     D(2)=R4*R6*(R2+R3)
     D(3)=R6*R7*(R2+R3+R4)+R4*R7*(R2+R3)+R4*R6*(R2+R3)
     D(4)=R4*R6*(R2+R3)
     X1=R5
     GO TO 111
105  B=(R4+R5)*(R2+R3)+R4*R5
     C(1)=R1*(R2+R3)+R4*(R1+R2)+R2*R3
     C(2)=P4*(R2+R3)
     C(3)=(R5+R7)*(R2+R3+R4)+R4*(R2+R3)
     C(4)=(R2+R3+R4)*R5
     D(1)=(R4+R5)*R1*(R2+R3)+R4*R5*(R1+R2)+R2*R3*(R4+R5)
     D(2)=R4*R5*(R2+R3)
     D(3)=R5*R7*(R2+R3+R4)+R4*R7*(R2+R3)
     D(4)=R4*R6*(R2+R3)
     X1=R6
     GO TO 111
```

```
106  A=0.0
     B=(R2+R3)*(R4+R5+R6)+R4*(R5+R6)
     C(1)=0.0
     C(2)=0.0
     C(3)=(R5+R6)*(R2+R3+R4)+R4*(R2+R3)
     C(4)=0.0
     D(1)=(R4+R5+R6)*(R1*R2+R2*R3+R3*R1)+(R1+R2)*(R5+R6)*R4
     D(2)=(R2+R3)*(R5+R6)*R4
     D(3)=R5*R6*(R2+R3+R4)+R4*R6*(R2+R3)
     D(4)=D(3)
     X1=R7
     GO TO 111
107  READ(MM1,12) (CM(L),L=1,4)
     WRITE(MM2,13) M1,M2,M3,R1,G(1),CM(1),R2,G(2),CM(2),R3,G(3),CM(3),
    2R4,G(4),CM(4),R5,R6,R7
     VF2=0.0
     DO 208 L=1,4
     VF(1)=CM(L)-G(L)
208  VF2=VF2+VF(L)**2
     DO 220 J=1,7
     ANUM=0.0
     DENO=0.0
     DO 221 L=1,4
     ANUM=ANUM+VF(L)*Q(L,J)
221  DENO=DENO+Q(L,J)**2
     ANUM=ANUM**2/DENO
     DENO=VF2-ANUM
     RES(J,1)=ANUM
220  RES(J,2)=DENO
     AMAX=0.0
     JL=8
     DO 400 J=1,7
     IF(RES(J,1)-AMAX) 400,400,401
401  AMAX=RES(J,1)
     JL=J
400  CONTINUE
     DO 404 J=1,7
     WRITE(MM2,14) J,RES(J,1),RES(J,2)
     IF(JL-J) 404,405,404
405  WRITE(MM2,16) STAR,STAR,STAR
404  CONTINUE
     WRITE(MM2,15)
     GO TO 107
     END
```

5.5 Program No. 5*

```
C       ITERATIVE METHOD-LEAD-LAG NETWORK APPLICATION
        DIMENSION FREQ(10),P(10,10),X(4),VO(10),SD(10),F(10,4),ETA(10),
       2B(4,4),BB(10,4),PSI(4),          L1(16),MMM1(16),A(10,4),B2B(4,4),
       3C2C(4,4)
        DOUBLE PRECISION CC(100),DET
   10 FORMAT(212,214,6F10.5/4F10.5)
   11 FORMAT(4F10.5)
   12 FORMAT(8F10.4/8F10.4/4F10.4)
   13 FORMAT(5X,4E9.3,F24.6)
   14 FORMAT('1',60X,2(I2,'/'),I4///)
        DEFINE V(X1,X2,X3,X4,FR)=(     SQRT(X3**2+1./((6.2832*FR*X4)**2)))/
       2(SQRT((X1/(1.+(6.2832*FR*X2*X1)**2)+X3)**2+((6.2832*FR*X2*X1*X1)/
       3(1.+(6.2832*FR*X2*X1)**2)+1./(6.2832*FR*X4))**2))
   15 FORMAT(13X,'NORMAL COMPONENTS',7X,'FREQUENCIES',5X,'MEASUREMENTS',
       25X,'STANDARD DEVIATION'/10X,'R1= ',F8.5,' MEGOHM',10X,F8.4,8X,F8.4
       3,13X,F6.3/10X,'C1= ',F8.5,' MICROFARAD',6X,F8.4,8X,F8.4,13X,F6.3/
       410X,'R2= ',F8.5,' MEGOHM',10X,F8.4,8X,F8.4,13X,F6.3/10K,'C2= ',F8.
       55,' MICROFARAD',6X,F8.4,8X,F8.4,13X,F6.3)
   16 FORMAT(39X,F8.4,8X,F8.4,13X,F6.3)
   17 FORMAT(5X,5(E18.8,2X))
   18 FORMAT(///)
   19 FORMAT(//13X,'NORMAL VALUES'/13X,'FREQUENCIES',14X,'I/O RELATION'/
       `2)
   20 FORMAT(14X,F8.4,15X,F8.4)
        MM1=5
        MM2=6
        AMMAA=0.000001
        READ(MM1,10) M1,M2,M3,MA,(FREQ(L),L=1,MA)
  203 WRITE(MM2,14) M1,M2,M3
        KK1=1
        KKKK1=0
        DO 101 J=1,MA
        DO 101 I=1,MA
  101 P(I,J)=0.0
        READ(MM1,11) (X(I),I=1,4)
        X1=X(1)*1.E+6
        X2=X(2)*1.E-6
        X3=X(3)*1.E+6
        X4=X(4)*1.E-6
        READ(MM1,12) (VO(I),SD(I),1=1,MA)
        DO 102 J=1,MA
  102 P(J,J)=1./(SD(J)**2)
        WRITE(MM2,15) (X(L),FREQ(L),VO(L),SD(L),L=1,4)
        IF (MA-4) 1000,1000,150
  150 WRITE(MM2,16) (FREQ(L),VO(L),SD(L),L=5,MA)
        WRITE(MM2,19)
```

*UNIVAC 1108

```
      DO 181 L=1,MA
      FR=FREQ(L)*1000.
      ACCU=V(X1,X2,X3,X4,FR)
  181 WRITE(MM2,20) FREQ(L),ACCU
      WRITE(MM2,14) M1,M2,M3
C     MATRICES F AND ETA
 1000 AIN1=AMMAA*X1
      AIN2=AMMAA*X2
      AIN3=AMMAA*X3
      AIN4=AMMAA*X4
      DO 103 L=1,MA
      FR=FREQ(L)*1000.
      ANA=V(X1,X2,X3,X4,FR)
      F(L,1)=(V(X1+AIN1,X2,X3,X4,FR)-ANA)/AIN1
      F(L,2)=(V(X1,X2+AIN2,X3,X4,FR)-ANA)/AIN2
      F(L,3)=(V(X1,X2,X3+AIN3,X4,FR)-ANA)/AIN3
      F(L,4)=(V(X1,X2,X3,X4+AIN4,FR)-ANA)/AIN4
  103 ETA(L)=VO(L)-ANA
C     MATRIX A
      DO 104 I=1,MA
      DO 104 J=1,4
      A(I,J)=0.0
      DO 104 L=1,MA
  104 A(I,J)=A(I,J)+P(I,L)*F(L,J)
C     MATRIX PSI
      DO 105 I=1,4
      DO 105 J=1,4
      B(I,J)=0.0
      DO 105 L=1,MA
  105 B(I,J)=B(I,J)+F(L,I)*A(L,J)
      KL=1
      DO 301 J=1,4
      DO 301 I=1,4
      CC(KL)=B(I,J)
      B2B(I,J)=B(I,J)
  301 KL=KL+1
      DO 152 J=1,4
  152 WRITE(MM2,17) (B(J,I),I=1,4 )
      MJ=4
      CALL MINV(CC,MU,DET,L1,MMM1)
      WRITE(MM2,18)
      KL=1
      DO 302 J=1,4
      DO 302 I=1,4
      B(I,J)=CC(KL)
  302 KL=KL+1
      DO 153 J=1,4
  153 WRITE(MM2,17) (B(J,I),I=1,4)
      WRITE(MM2,18)
      DO 1002 J=1,4
```

```
      DO 1002 I=1,4
      C2C(I,J)=0.0
      DO 1002 L=1,4
 1002 C2C(I,J)=C2C(I,J)+B2B(I,L)*B(L,J)
      DO 1003 J=1,4
 1003 WRITE(MM2,17) (C2C(J,I),I=1,4)
      DO 115 I=1,MA
      DO 115 J=1,4
      BB(I,J)=0.0
      DO 115 L=1,4
  115 BB(I,J)=BB(I,J)+A(I,L)*B(L,J)
      DO 116 I=1,4
      PSI(I)=0.0
      DO 116 L=1,MA
  116 PSI(I)=PSI(I)+ETA(L)*BB(L,I)
      X1=X1+PSI(1)
      X2=X2+PSI(2)
      X3=X3+PSI(3)
      X4=X4+PSI(4)
      S2=0.0
      DO 117 L=1,MA
      FR=FREQ(L)*1000.
  117 S2=S2+((VO(L)-V(X1,X2,X3,X4,FR))**2)/(SD(L)**2)
      XX1=X1*1.E-6
      XX2=X2*1.E+6
      XX3=X3*1.E-6
      XX4=X4*1.E+6
      WRITE(MM2,13) XX1,XX2,XX3,XX4,S2
      WRITE(MM2,18)
      GO TO (201,202),KK1
  201 KK1=KK1+1
  208 S1=S2
      WRITE(MM2,14) M1,M2,M3
      GO TO 1000
  202 IF(ABS(S1-S2)-0.000001)204,203,203
  204 KKKK1=KKKK1+1
      IF(KKKK1-20) 208,208,203
      END
```

NOTE

The Subroutine MINV that is call in this program belongs to the math-pack of IBM/360

5.6 Program No. 6*

```
C       BODE DIAGRAM TECHNIQUE
C       LEAD-LAG NETWORK
        REAL MINUS
        DATA PLUS/'+'/MINUS/'-'/ BLK/'0'/
     10 FORMAT(2I2,I4,F10.4)
     11 FORMAT(4F12.6)
     12 FORMAT(70X,2(I2,'/'),I4///5X,'TEST FREQUENCY 1=',F6.2,' KC   BREAK
       2 FREQUENCY 1=',F6.2,' KC   R1=',F8.4,' MEGOHM'/5X,'TEST FREQUENCY
       22=',F6.2,' KC   BREAK FREQUENCY 2=',F6.2,' KC   R2=',F8.4,' MEGOHM
       4'/5X,'TEST FREQUENCY 3=',F6.2,' KC   BREAK FREQUENCY 3=',F6.2,' KC
       5   C1=',F8.4,' MICROFARAD')
     14 FORMAT(                5X,'TEST FREQUENCY 4=',F6.2,' KC   BREAK FR
       SEQUENCY 4=',F6.2,' KC   C2=',F8.4,' MICROFARAD'/5X,'TEST FREQUENCY
       7 5=',F6.2,' KC',3X,'VARIATION        =',F6.2,' PC',3X,'AM=',F8.4)
     13 FORMAT(9X,5('*',2X,F6.4,2X),'*',10X,'*')
     20 FORMAT('+',65X,'NORMAL')
     21 FORMAT('+',65X,'R1 LOW')
     22 FORMAT('+',65X,'R1 HIGH')
     23 FORMAT('+',65X,'R2 LOW')
     24 FORMAT('+',65X,'R2 HIGH')
     25 FORMAT('+',65X,'C1 LOW')
     26 FORMAT('+',65X,'C1 HIGH')
     27 FORMAT('+',65X,'C2 LOW')
     28 FORMAT('+',65X,'C2 HIGH')
     29 FORMAT(9X,67('*'))
     17 FORMAT(14X,'*',F7.3,'  *',F10.2,'  *',3(F7.2,'  *'))
     18 FORMAT('1',60X,2(I2,'/'),I4//14X,54('*')/14X,'*    KC   *      W
       2   *    PR   *    IP   *   REL   *'/14X,54('*')/14X,'*',9X,'*',12X
       3,'*',3(9X,'*'))
     30 FORMAT(14X,54('*'))
     31 FORMAT(  //9X,67('*')/9X,'*  GAIN 1   * GAIN 2  *  GAIN 3  *  GAIN
       8 4  *  GAIN 5  * CONDITION*'/9X,67('*')/9X,6('*',10X),'*')
     32 FORMAT(9X,'*',5(5X,AI,4X,'*'),10X,'*')
     33 FORMAT('1')
        MM1=5
        MM2=6
        TPI=6.2832
    100 READ(MM1,10) M1,M2,M3,AMMM
        READ(MM1,11) R1,C1,R2,C2
        AM=0.10
        KK2=1
        T1=C1*R1
        T2=C2*R2
        T12=R1*C2
        B=-((T1+T2)/(2.*T1*T2))
        C=SQRT((B**2)-(1./(T1*T2)))
        SN1=B+C
```

*UNIVAC 1108

```
      SN2=B-C
      B=-((T1+T2+T12)/(2.*T1*T2))
      C=SQRT((B**2)-(1./(T1*T2)))
      SD1=B+C
      SD2=B-C
      FREQ(1)=0.0
      FREQ(5)=-SD1/TPI
      FREQ(9)=-SN1/TPI
      FREQ(13)=-SN2/TPI
      FREQ(17)=-SD2/TPI
      FREQ(21)=2.*FREQ(17)-FREQ(13)
      KBA=1
      DO 180 L=1,5
      DIF=(FREQ(KBA+4)-FREQ(KBA))/4.
      FREQ(KBA+1)=FREQ(KBA)+DIF
      FREQ(KBA+2)=FREQ(KBA+1)+DIF
      FREQ(KBA+3)=FREQ(KBA+2)+DIF
  180 KBA=KBA+4
      LL=1
      DO 102 L=5,17,4
      BREAK(LL)=FREQ(L)/1000.
  102 LL=LL+1
      LL=1
      DO 103 L=3,21,4
      TEST(LL)=FREQ(L)/1000.
  103 LL=LL+1
      WRITE(MM2,33)
  305 WRITE(MM2,12) M1,M2,M3,TEST(1),BREAK(1),R1,TEST(2),BREAK(2),R2,
     2TEST(3),BREAK(3),C1
      WRITE(MM2,14) TEST(4),BREAK(4),C2,TEST(5),AM,AMMM
      KK1=1
  406 WRITE(MM2,31)
      DO 106 J=1,9
      GO TO (401,402),KK1
  401 T1=C1*R1
      T2=C2*R2
      T12=R1*C2
      DO 105 L=1,5
      CM=TEST(L)*1000.*TPI
      ANU=-T1*T2*OM**2+1.
      DEN=ANU**2+((T1+T2+T12)**2)*OM**2
      PR=(ANU**2+(T1+T2)*(T1+T2+T12)*OM**2)/DEN
      AIP=((T1+T2)*ANU*OM-ANU*(T1+T2+T12)*OM)/DEN
      RELA(L)=SQRT(PR**2+AIP**2)
  105 RELU(J,L)=RELA(L)
      WRITE(MM2,13) (RELA(JL),JL=1,5)
      GO TO 403
  402 DO 408 J1=1,5
      IF (RELU(J,J1)) 411,412,413
```

```
411 IF (AMMM+RELU(J,J1)) 443,412,412
443 RELA(J1)=MINUS
    GO TO 408
412 RELA(J1)=BLK
    GO TO 408
413 IF (RELU(J,J1)-AMMM) 412,412,444
444 RELA(J1)=PLUS
408 CONTINUE
    WRITE(MM2,32) (RELA(J1),J1=1,5)
403 GO TO (107,108,109,110,111,112,113,114,115),J
107 WRITE(MM2,20)
    RES=R1
    R1=R1-AM*R1
    GO TO 106
108 WRITE(MM2,21)
    R1=RES+AM*RES
    GO TO 106
109 WRITE(MM2,22)
    R1=RES
    RES=R2
    R2=R2-AM*R2
    GO TO 106
110 WRITE(MM2,23)
    R2=RES+AM*RES
    GO TO 106
111 WRITE(MM2,24)
    R2=RES
    RES=C1
    C1=C1-AM*C1
    GO TO 106
112 WRITE(MM2,25)
    C1=RES+   AM*RES
    GO TO 106
113 WRITE(MM2,26)
    C1=RES
    RES=C2
    C2=C2-AM*C2
    GO TO 106
114 WRITE(MM2,27)
    C2=RES+AM*C2
    GO TO 106
115 WRITE(MM2,28)
    C2=RES
106 CONTINUE
    WRITE(MM2,29)
    GO TO (404,405),KK1
404 KK1=KK1+1
    DO 409 J=2,9
```

```
      DO 409 L=1,5
  409 RELU(J,L)=RELU(J,L)-RELU(1,L)
      DO 1000 L=1,5
 1000 RELU(1,L)=0.0
      GO TO 406
  405 GO TO (301,301,301,500),KK2
  301 KK2=KK2+1
      WRITE(MM2,33)
      AM=AM+0.10
      GO TO 305
  500 WRITE(MM2,18) M1,M2,M3
      T1=C1*R1
      T2=C2*R2
      T12=R1*C2
      DO 181 L=1,21
      OM=TPI*FREQ(L)
      ANU=-T1*T2*OM**2+1.
      DEN=ANU**2+((T1+T2+T12)**2)*OM**2
      PR=(ANU**2+(T1+T2)*(T1+T2+T12)*OM**2)/DEN
      AIP=((T1+T2)*ANU*OM-ANU*(T1+T2+T12)*OM)/DEN
      REL=SQRT(PR**2+AIP**2)
      FRU=FREQ(L)/1000.
  181 WRITE(MM2,17) FRU,OM,PR,AIP,REL
      WRITE(MM2,30)
      GO TO 100
      END
```

5.7 Program No. 7*

```
C       LEAD-LAG NETWORK (REMNANT METHOD)
        REAL K1,K2,KBB
        COMPLEX AA,KB
        DIMENSION COEF (6,9),DIFX(6)
   10   FORMAT(2I2,I4)
   11   FORMAT(5F10.6)
   12   FORMAT('1',T90,2(I2,'/'),I4)
   13   FORMAT(2X,'*',7(2X,F11.7,2X,'*'))
   14   FORMAT(///10X,'R1 =',F9.6,3X,'C1   =',F9.6,3X,'BETHA=',F9.4/10X,
       2'R2 =',F9.6,3X,'C2   =',F9.6/10X,'K1 =',F9.6,3X,'K2   =',F9.6/
       210X,'S1 =',E9.3,3X,'S2   =',E9.3                          /10X,
       3'KKB=',F9.6,3X,'PHASE=',F9.6/10X,'RP =',F9.6,3X,'IP   =',F9.6)
   15   FORMAT(//10X,'PROOF 2.*KKB*COS(-PHASE)+K1+K2=',F10.6)
   16   FORMAT(///2X,113('*')/2X,'*',21X,'INPUT',21X,'*',21X,'OUTPUT',20X,
       2'*',15X,'*'/2X,97('*'),6X,'SUM',6X,'*'/2X,'*',6X,2HVI,7X,'*',6X,3H
       3VI',6X,'*',6X,4HVI'',5X,'*',6X,'VO',7X,'*',6X,3HVO',6X,'*'6X,4HVO'
       4',5X,'*',15X,'*'/2X,113('*')/2X,7('*',15X),'*')
   17   FORMAT(2X,113('*'))
   18   FORMAT(10X,'A2 =',E13.6,2X,'A1 =',E13.6,2X,'AD =',E13.6  /10X,'B2
       2 =',E13.6,2X,'B1 =',E13.6,2X,'BC =',E13.6/10X,'VRI=',F12.6,2X,'VO1
       3=',F12.6,2X,'VR2=',F12.6,2X,'VO2=',F12.6)
   19   FORMAT('1',70X,2(I2,'/'),I4//15X,'ALLOWED COEFFICIENTS VARIATION'/
       215X,72('*')/15X,'*COEFF.*  LOWER BOUND   *   HIGHER BOUND   *   NORMAL
       3VALUE  *   PERCENTAGE   *'/15X,72('*')/15X,'*',6X,'*',4(15X,'*'))
   20   FORMAT(15X,'*',6X,'*',3(1X,E13.6,'*'),1X,F13.3,' *')
   21   FORMAT('+',17X,'A2')
   22   FORMAT('+',17X,'A1')
   23   FORMAT('+',17X,'A0')
   24   FORMAT('+',17X,'B2')
   25   FORMAT('+',17X,'B1')
   26   FORMAT('+',17X,'B0')
   27   FORMAT(15X,72('*'))
   30   FORMAT(/)
   31   FORMAT('1',T90,2(I2,'/'),I4/10X,'REMNANT ANALYSIS'//10X,'A2=',E10.
       23,3X,'VA2=',E10.3,3X,'B2=',E10.3,3X,'VB2=',E10.3/10X,'A1=',E10.3,3
       3X,'VA1=',E10.3,3X,'B1=',E10.3,3X,'VB1=',E10.3/10X,'A0=',E10.3,3X,'
       4VA0=',E10.3,3X,'B0=',E10.3,3X,'VB0=',E10.3,//10X,107('*')/10X,'*',
       56X,'*',14X,'INPUT',13X,'*',13X,'OUTPUT',13X,'*  SCALAR  *  PARAM.
       6 *   COEF.   *'/10X,'* CASE ',67('*'),3(10X,'*')/10X,'*',6X,'*',4X,
       7'VI',4X,'*',4X,3HVI',3X,'*',3X,4HVI'',3X,'*',4X,'VO',4X,'*',4X,3HV
       80',3X,'*',3X,4HVO'',3X,'*  REMNANT '  STATUS  8  N.R.   *'/10X,10
       97('*'1/10X,'*',6X,'*',9(10X,'*'))
   32   FORMAT(10X,107('*'))
   33   FORMAT(10X,'*',I4,'  *',6(1X,F8.5,' *'),E9.3   ,' * ',F5.1,1X,A2
       2' *            *')
   35   FORMAT(F8.2,A2,I4)
   40   FORMAT('+',109X,'YES')
   41   FORMAT('+',109X,'NO ')
        KK1=0
```

*UNIVAC 1108

```
      KK2=1
      JJ1=1
      ALL=0.25
      INDICE=0
      MAXE=30
      TO=1.8
      MM1=5
      MM2=6
      READ(MM1,10) M1,M2,M3
      READ(MM1,11) R1,C1,R2,C2,BETHA
      RRR1=R1
      RRR2=R2
      CCC1=C1
      CCC2=C2
      VAR1=0.0
      VAR2=0.0
      VAR3=0.0
      VAR4=0.0
1000  T1=R1*C1
      T2=R2*C2
      T12=R1*C2
      GO TO (501,502),KK2
 501  A=-(T1+T2+T12)/(2.*T1*T2)
      B=SQRT(A**2-(1./(T1*T2)))
      A1=A+B
      A2=A-B
      K1=(BETHA/(BETHA**2+A1**2))*(A1**2+((T1+T2)*A1)/(T1*T2)+1./(T1*T2)
     2)/(A1-A2)
      K2=(BETHA/(BETHA**2+A2**2))*(A2**2+((T1+T2)*A2)/(T1*T2)+1./(T1*T2)
     2)/(A2-A1)
      AA=CMPLX(0.0,-BETHA)
      KB=CMPLX(0.0,-2.*BETHA)
      KB=(BETHA/KB              )*(T1*T2*AA**2+(T1+T2)*AA+1.)/(T1*T2*AA**
     22+(T1+T2+T12)*AA+1.)
      A11=REA-(KB)
      A22=AIMAG(KB)
      AA11=A11
      AA22=A22
      KBB=SQRT(A11**2+A22**2)
      CALL PHA(A11,A22,PHASE)
      T=TO+0.2
      GO TO (1001,1002,3001),JJ1
1001  WRITE(MM2,12) M1,M2,M3
      WRITE(MM2,14) R1,C1,BETHA,R2,C2,K1,K2,A1,A2,KBB,PHASE,AA11,AA22
      SUM=2.*(BB*COS(-PHASE)+K1+K2
      WRITE(MM2,15)SUM
      JJ1=2
1002  WRITE(MM2,12) M1,M2,M3
```

```
 502  ZZ1=T1*T2
      ZZ2=T1+T2
      ZZ3=T1+T2+T12
      ZZ4=1.0
      GO TO (203,203,6007),JJ1
 203  WRITE(MM2,18) ZZ1,ZZ3,ZZ4,ZZ1,ZZ2,ZZ4,VAR1,VAR2,VAR3,VAR4
      WRITE(MM2,30)
      GO TO (503,504),KK2
 503  WRITE(MM2,16)
      INDICE=INDICE+1
      COEF(1,INDICE)=ZZ1
      COEF(2,INDICE)=ZZ3
      COEF(3,INDICE)=ZZ4
      COEF(4,INDICE)=ZZ1
      COEF(5,INDICE)=ZZ2
      COEF(6,INDICE)=ZZ4
3001  DO 108 J=1,MAXE
      BET=BETHA*T
      VI=SIN(BET)
      VI1P=BETHA*COS(BET)
      VI2P=-BETHA**2*VI
      VO=2.*KBB*COS(BET-PHASE)+K1*EXP(A1*T)+K2*EXP(A2*T)
      VO1P=-2.*BETHA*KBB*SIN(BET-PHASE)+K1*A1*EXP(A1*T)+K2*A2*EXP(A2*T)
      VO2P=-2.*BETHA**2*KBB*COS(BET-PHASE)+K1*A1**2*EXP(A1*T)+K2*A2**2*
     2EXP(A2*T)
      SUM=T1*T2*VO2P+(T1+T2+T12)*VO1P+VO-T1*T2*VI2P-(T1+T2)*VI1P-VI
      GO TO (8700,8700,108),JJ1
8700  WRITE(MM2,13) VI,VI1P,VI2P,VO,VO1P,VO2P,SUM
 108  T=T+0.2
      GO TO (6006,6006, 502),JJ1
6006  WRITE(MM2,17)
      KK1=KK1+1
      GO TO (150,151,152,153,154,155,156,157,158),KK1
 150  VAR1=R1*ALL
      R1=R1+VAR1
      GO TO 1000
 151  R1=R1-2.*VAR1
      VAR1=-VAR1
      GO TO 1000
 152  R1=R1-VAR1
      VAR1=0.0
      VAR2=C1*ALL
      C1=C1+VAR2
      GO TO 1000
 153  C1=C1-2.*VAR2
      VAR2=-VAR2
      GO TO 1000
 154  C1=C1-VAR2
      VAR2=0.0
```

```
      VAR3=R2*ALL
      R2=R2+VAR3
      GO TO 1000
  155 R2=R2-2,*VAR3
      VAR3=-VAR3
      GO TO 1000
  156 R2=R2-VAR3
      VAR3=0.0
      VAR4=C2*ALL
      C2=C2+VAR4
      GO TO 1000
  157 C2=C2-2.*VAR4
      VAR4=-VAR4
      GO TO 1000
  158 C2=C2-VAR4
      VAR4=0.0
      VARI=R1*ALL
      R1=R1+VAR1
      KK1=0
      KK2=KK2+1
      WRITE(MM2,12) M1,M2,M3
      GO TO 1000
  504 KK1=KK1+1
      GO TO (601,602,603,604,605,606,607,608),KK1
  601 R1=R1-2.*VAR1
      VAR1=-VAR1
      GO TO 1000
  602 R1=R1-VAR1
      VAR1=0.0
      VAR2=C1*ALL
      C1=C1+VAR2
      GO TO 1000
  603 C1=C1-2.*VAR2
      VAR2=-VAR2
      GO TO 1000
  604 C1=C1-VAR2
      VAR2=0.0
      VAR3=R2*ALL
      R2=R2+VAR3
      GO TO 1000
  605 R2=R2-2.*VAR3
      VAR3=-VAR3
      GO TO 1000
  606 R2=R2-VAR3
      VAR3=0.0
      VAR4=C2*ALL
      C2=C2+VAR4
      GO TO 1000
```

```
 607  C2=C2-2.*VAR4
      VAR4=-VAR4
      GO TO 1000
 608  WRITE(MM2,19) M1,M2,M3
      DO 707 J=1,6
      AMIN=COEF(J,1)
      AMAX=COEF(J,1)
      DO 708 L=2,9
      IF(COEF(J,L)-AMIN)709,710,710
 709  AMIN=COEF(J,L)
      GO TO 708
 710  IF(COEF(J,L)-AMAX) 708,708,711
 711  AMAX=COEF(J,L)
 708  CONTINUE
      POR=((COEF(J,1)-AMIN)/COEF(J,1))*100.
      WRITE(MM2,20) AMIN,AMAX,COEF(J,1),POR
      DIFX(J)=AMAX-COEF(J,1)
      GO TO (720,721,722,723,725,726),J
 720  WRITE(MM2,21)
      GO TO 707
 721  WRITE(MM2,22)
5005  IF(VVBO) 5000,5006,5006
5000  JKL=2
5006  GO TO (5007,5008),JKL
5007  WRITE(MM2,40)
      GO TO 205
5008  WRITE(MM2,41)
      GO TO 205
      END
      SUBROUTINE PHA(R,I,PHASE)
      REAL I
      IF(I) 100,101,102
 102  IF(R) 103,104,105
 103  R=-R
      PHASE=3.14159265-ATAN(I/R)
      RETURN
 104  PHASE=1.57079632
      RETURN
 105  PHASE=ATAN(I/R)
      RETURN
 101  IF(R) 107,109,109
 107  PHASE=3.14159265
      RETURN
 109  PHASE=0.0
      RETURN
 100  IF(R) 150,151,152
 150  I=-I
      R=-R
      PHASE=-3.14159265+ATAN(I/R)
      RETURN
```

```
151   PHASE=-1.57079632
      RETURN
152   I=-I
      PHASE=-ATAN(I/R)
      RETURN
      END
```