Proceedings of the IASTED International Conference
Signal and Image Processing
October 27-31, 1998, Las Vegas, Nevada - USA

DICTIONARY OPTIMIZATION IN FAULT ANALYSIS APPLYING BINARY LOGICAL MANIPULATION ALGORITHM

Dražen Jurišić, Neven Mijat and Nino Stojković
Faculty of Electrical Engineering and Computing Zagreb
Unska 3, HR-10000 Zagreb, Croatia, Tel. ++385-1-6129911, Fax. ++385-1-6129652
E-mail: drazen.jurisic@fer.hr, neven.mijat@fer.hr, nino.stojkovic@fer.hr

Abstract

This paper presents fault dictionary optimization technique which uses binary logical manipulation algorithm. It uses brute force to find optimal testing conditions. Since the optimisation technique handles only numbers, it can be applied in all fault dictionary based techniques. Optimization method applied on fault dictionary, results in reducing its dimensions and in isolating uniquely defined faults. Remaining ununiquely defined faults are grouped into ambiguity sets. The universality of the technique is approved by an application to two different approaches: the approach in the frequency and in theDC-domains. In this paper the emphasis is on the frequency domain.

The performances of the method and the main steps are illustrated on an analog active filter example.

1. INTRODUCTION

If the component values in electrical circuits deviate from its nominal values and produce a circuit failure, the fault isolation procedure can be carried out to locate the faulty element. To test the circuit the isolation procedure uses the signatures previously defined for all possible faults cases and stored in the form of dictionary.

The method described in this paper is used to optimize the fault dictionary. It throws out redundant measurements, isolates unambiguously defined faults and ambiguous faults groups into ambiguity sets. If one of ambiguous faults occurs one should repair all faults in group.

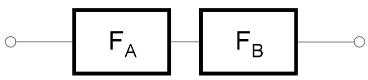
The fault isolation procedure developed by Hochwald and Bastian performs the testing in the DC domain. They measured DC nodal voltages in video amplifier circuit realized with discrete semiconductor elements (transistors, diodes, etc.) [4, 5]. The method forms a fault dictionary with voltages in every considered node and for every fault case. The objective of optimization of the fault dictionary is to reduce number of nodes and still retain ability to identify faulty element.

Jurisic, Mijat and Cosic developed method in the frequency domain for testing analog active filters for single hard faults of passive elements (short or open circuits) [1]. The filter transfer function magnitudes correspondent to particular single faults are calculated at a previously determined set of discrete test frequencies, forming a fault dictionary. To determine the minimal number of test frequencies sufficient to isolate all uniquely identified faults, the optimization method is applied.

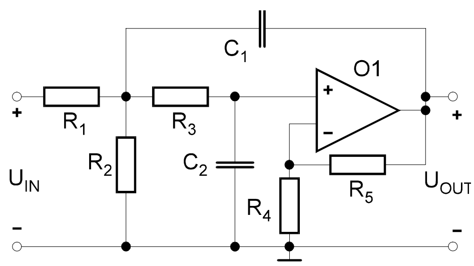
Both methods belong to simulation-before-test approach and construct fault dictionaries. The optimization procedure will be explained in details on the latter technique.

2. FAULT DICTIONARY

The fault location method in the frequency domain [1] is illustrated by an example of 4th order Butterworth LP filter. The filter is realized by cascading two 2nd order sections (Fig. 1).



(a)



(b)

Fig. 1. Active 4th order LP filter (a) cascaded sections, (b) biquadratic section.

It is assumed that all passive elements (14 ones) can produce either open or short circuit (n_{faul}=28), for example R_{1A}+ means R_1 in block A is an open circuit, and R_{3B}- means R_3 in block B is a short circuit. The faults are numbered with 0 to 27.

In our example the magnitudes in 100 discrete logarithmically spaced frequencies ranging from 10^1rad/s to 10^3rad/s, denoted by numbers 0,1,…,99 (n_{freq}=100), are calculated,.

Choosing a smaller subset of n_{freq}=12 test frequencies one can form initial fault dictionary. The initial test frequency set is {47, 50, 52, 53, 54, 57, 58, 60, 62, 66, 67, 71}. There are $n_{freq}\times n_{faul}$ records in the dictionary. The signatures in the fault dictionary are quantified error characteristic using scale of marks from 0 to 8 (n_{mark}=9). Error characteristic is difference between nominal characteristic and faulty characteristic when element value changes ±50% from its nominals.

The initial fault dictionary is given in Table I. Columns correspond to test frequencies and rows correspond to faults. Note that some of the faults (R_{1B}+, R_{3B}+), (R_{1B}-, R_{3B}-), (R_{4B}+, R_{5B}-) and (R_{4B}-, R_{5B}+) have the same signatures. Other faults have unique signatures.

TABLE I Fault Dictionary (Shaded columns are the result of the optimization)

Test Frequency		47	50	52	53	54	57	58	60	62	66	67	71
Nr.	Fault	Signature											
0	R_{1A}+	1	1	1	1	1	1	1	2	2	2	2	2
1	R_{1A}-	5	5	5	5	5	5	5	6	6	7	7	7
2	R_{2A}+	6	6	6	6	6	6	5	5	5	0	0	0
3	R_{2A}-	3	3	3	3	3	3	3	3	2	2	1	1
4	R_{3A}+	1	1	2	2	2	3	3	3	3	3	3	3
5	R_{3A}-	1	0	0	0	0	5	6	7	8	8	8	8
6	R_{4A}+	2	3	3	3	3	3	3	3	3	3	2	2
7	R_{4A}-	7	7	7	7	7	8	8	8	7	7	7	7
8	R_{5A}+	6	7	7	8	8	8	8	8	8	7	7	6
9	R_{5A}-	3	3	3	4	4	4	4	4	4	3	3	3
10	R_{1B}+	1	1	1	1	1	2	2	2	2	2	2	2
11	R_{1B}-	5	5	5	5	5	6	6	6	6	6	6	7
12	R_{2B}+	0	0	0	0	0	0	0	0	0	0	0	0
13	R_{2B}-	1	1	1	1	1	1	1	1	1	0	0	0
14	R_{3B}+	1	1	1	1	1	2	2	2	2	2	2	2
15	R_{3B}-	5	5	5	5	5	6	6	6	6	6	6	7
16	R_{4B}+	1	1	1	1	1	1	1	1	1	1	1	1
17	R_{4B}-	5	5	5	5	5	5	5	5	5	5	5	5
18	R_{5B}+	5	5	5	5	5	5	5	5	5	5	5	5
19	R_{5B}-	1	1	1	1	1	1	1	1	1	1	1	1
20	C_{1A}+	7	8	8	8	8	7	6	0	1	2	2	2
21	C_{1A}-	2	3	3	3	3	3	3	3	3	2	1	0
22	C_{2A}+	2	3	3	3	3	4	4	4	4	3	3	3
23	C_{2A}-	0	0	0	0	0	5	6	7	8	8	8	8
24	C_{1B}+	5	5	5	5	5	0	0	0	1	1	1	1
25	C_{1B}-	1	1	1	1	1	1	1	0	0	0	0	5
26	C_{2B}+	2	2	2	2	2	2	2	2	3	3	3	3
27	C_{2B}-	6	6	6	6	7	7	7	7	7	8	8	8

3. FAULT DICTIONARY OPTIMISATION

As can be seen, it is possible to throw out some columns from Table I (i.e. to eliminate some test frequencies), keeping the signatures different from each other. The problem is to find minimal number of test frequencies sufficient to isolate all uniquely identified faults. Uniquely identified fault has code which is different from any other code in the dictionary. When minimal number of test frequencies is established second problem is to find which faults do not have unique codes and group them into pairs, sets of three, four, etc. faults.

The optimization method described in this paper is based on the binary logical manipulation algorithm. Although it is the brute force algorithm, it has substantial improvements saving more time in comparison with the real brute force algorithm.

3.1 Ambiguity sets

In the first step for each test frequency the faults are grouped concerning to the marks into ambiguity sets shown in the Table II.

Ambiguity set contains faults which produce amplitude deviation at the frequency inside interval for that set and thus have the same mark.

Each row corresponds to one test frequency, and columns correspond to marks. Every test frequency has as many ambiguity sets as the number of marks. In every set there can be any number of faults, just one fault or the set can be empty. In one row particular fault can exist in only one ambiguity set. In all ambiguity sets in one row all n_{faul} faults are enumerated.

Table II Ambiguity sets generated from fault dictionary on Table I

FREQ.	SET 0	SET 1	SET 2	SET 3		SET 8
47	12,23	0,4,5,10,13, 14,16,19,25	6,21,22, 26	3,9		EMPTY
50	5,12,2 3	0,4,10,13,14 ,16, 19,25	26	3,6,9,21 , 22		20
52	5,12,2 3	0,10,13,14,1 6,19, 25	4,26	3,6,9,21 , 22		20
53	5,12,2 3	0,10,13,14,1 6,19, 25	4,26	3,6,21,2 2		8,20
54	5,12,2 3	0,10,13,14,1 6,19, 25	4,26	3,6,21,2 2		8,20
57	12,24	0,13,16,19, 25	10,14,2 6	3,4,6,21		7,8
58	12,24	0,13,16,19, 25	10,14,2 6	3,4,6,21		7,8
60	12,20, 24, 25	13,16,19	0,10,14, 26	3,4,6,21		7,8
62	12,25	13,16,19,20, 24	0,3,10, 14	4,6,21,2 6		5,8,23
66	2,12,1 3,25	16,19,24	0,3,10,1 4, 20,21	4,6,9,22 , 26		5,23,27
67	2,12, 13,25	3,16,19,21,2 4	0,6,10,1 4,20	4,9,22,2 6		5,23,27
71	2,12, 13,21	3,16,19,24	0,6,10,1 4,20	4,9,22,2 6		5,23,27

The ambiguity sets uniquely identify the faults applying the following rules:

RULE-1 Any ambiguity set consisting of a single fault uniquely identifies that fault (Fig. 2).

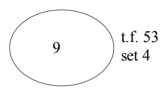


Fig. 2. Single fault in ambiguity set

RULE-2A Two ambiguity sets associated to two different test frequencies, whose intersection results in a single fault, uniquely identify that fault (Fig. 3).

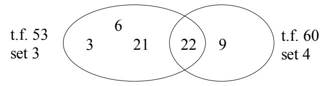
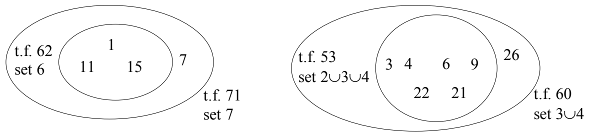


Fig. 3. Single fault in intersection of two ambiguity sets

RULE-2B Two ambiguity sets whose symmetric difference (the fault is the only different element and is contained in only one set) results in a single fault, also uniquely define that fault (Fig. 4(a)). The RULE-2B can be extended on combinations of unions of ambiguity sets (Fig. 4(b)).



(a) (b)

Fig. 4. Single fault in symmetrical difference of two ambiguity sets (a) two sets, (b) two unions of sets.

If an ambiguity set isolates a single fault one concludes that test frequency correspondent to the set isolates the fault.

3.2 Searching for uniquely identified faults

Optimization procedure handles ambiguity sets in the Table II applying rules in the following way: first RULE-1 is applied on rows relevant to single test frequency. Algorithm writes down single faults in all ambiguity sets.

Furthermore, the RULE-2A and RULE-2B are applied to each combination of test frequency pairs in Table II. There are

$$\binom{n_{freq}}{2} = \frac{n_{freq}(n_{freq}-1)}{1\cdot 2} \qquad (1)$$

such combinations. Algorithm writes down single faults in intersections of all ambiguity sets. For particular pair of test frequencies intersections are constructed between ambiguity sets from one test frequency to every ambiguity set on another frequency.

The algorithm has to apply RULE-2B on symmetrical difference of all ambiguity set unions. If there is n_{mark} ambiguity sets in one row, algorithm produces

$$\binom{n_{mark}}{m} = \frac{n_{mark}!}{m!(n_{mark}-m)!} \qquad (2)$$

combinations of unions for m=1,…,n_{mark}-1. Since m=0 and m=n_{mark} are excluded total number of unions is

$$n_{unions} = 2^{n_{mark}} - 2\,. \qquad (3)$$

In type of ambiguity sets as in Table II where all possible faults are enumerated in each row one does not need to apply RULE 2-B. In example given by Hochwald and Bastian table of ambiguity sets does not contain all faults in each row. In that case RULE 2-B has to be applied.

The procedure produces table with all uniquely defined faults. The table is triangular. The diagonal contains faults isolated at the test frequency by itself, while the other cells contain faults isolated by test frequency pairs correspondent to the cell.

Table III shows n_{faul}=20 uniquely defined faults forming a set {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 20, 21, 22, 23, 24, 25, 26, 27}. It is desirable that all existing faults are enumerated in this table. This is an optimal case and would mean that all faults in the dictionary have unique code.

3.3 Searching for minimal number of test frequencies

The procedure searches throughout the Table III to find minimal number of test frequencies which still isolate all unique faults.

Search algorithm forms permutations of nfreq test frequencies. There are n_{freq}! such permutations and they can be presented by tree. Nodes are sorted in n=n_{freq} columns and every node is correspondent to one test frequency. From every node in first column there is a path from the first to the nth column correspondent to one permutation. There are n! such paths.

TABLE III Uniquely identified faults vs. test frequencies (shaded cells represent path 47-50-52-71-57)

FREQ.	47	50	52	53		67	71
47	EMPTY	5,7,8,20,26	4,5,7,8,20,26	3,4,5,7,8,9,20,26		1,2,3,4,5,6,7,8,9,12,20,21,23,24,27	2,3,4,5,6,7,8,9 12,13,20,21,23,24,25,27
50		20,26	4,20,26	4,7,8,9,20,26		1,2,4,6,12,20,24,26,27	2,3,4,6,7,8,12 13,20,21,24,25,26,27
52			20	7,8,9,20		1,2,6,12,20,24,27	2,3,6,7,8,12,13,20,21,24,25,27
53				7,9		1,2,6,7,8,9 12,20,22,24 27	2,3,6,7,8,9,12 13,20,21,22,24,25,27
54						1,2,6,7,8,9,12,20,22,24 27	2,3,6,7,8,9,12 13,20,21,22,24,25,27
57						0,1,2,4,6,12 20,24,26,27	0,1,2,3,4,6,7,8 12,13,20,21,24,25,26,27
67						EMPTY	8,21,25
71							8

The problem is to find a path (permutation) with minimal number k leading nodes which still isolate all uniquely defined faults. The path is getting shorter, while other n-k trailing nodes are redundant and are thrown off. In general, there exist one or more permutations with minimal k.

The test frequency set is Ω={47, 50, 52, 53, 54, 58, 60, 62, 66, 67}. The number of permutations n_{freq}= 12 is 12!, but the algorithm does not take all of them into consideration. In the example in the Fig. 5 it searches the minimal sufficient set of test frequencies in Table III. The number of steps needed to construct the tree of permutations, is equal to the number of test frequencies.

For every new step algorithm chooses next test frequency from Ω as the first node. Let us describe one step on the Fig. 5.

The first node is a test frequency from Ω in the first column (LEVEL 0). Algorithm chooses the next node in next column correspondent to test frequency different from the one before. It branches to the new node forming a path of permutation. Each time it branches to new level the union of faults, isolated by using all nodes in the path, is formed. Algorithm checks if new faults are added to union.

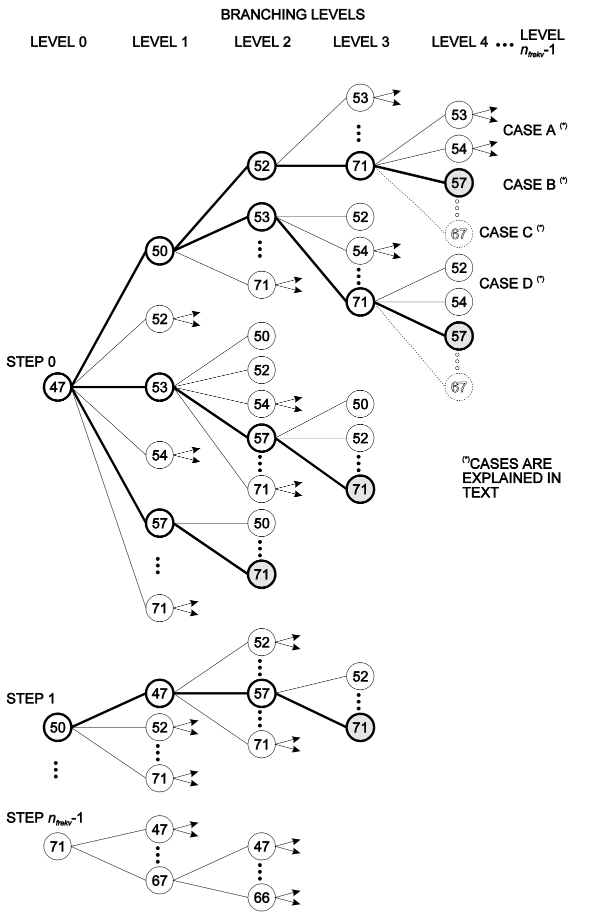


Fig. 5. Searching algorithm for minimal test frequency number applied on the Table III

If new faults had been isolated but still not all faults in Table II, algorithm branches to new level (CASE A). If no new faults had been isolated the algorithm should go one level back (CASE D). If algorithm isolated new faults and those are all unique faults it will write down the path 47-50-52-71-57 (CASE B) and will go two levels back jumping over remaining nodes from Ω (CASE C). The path 47-50-52-71-57 is marked in the Table III with gray cells. One has to notice that all faults existing in that table are contained in those cells.

Efficient algorithm for searching minimal test frequencies number is defined and it does not need to handle the majority of permutations what saves time. It is convenient to be programmed by computer languages using nested functions.

The minimal permutations are of interest. In our example there are two such a minimal paths: 47-57-71 and 47-58-71. The former will be chosen to construct optimized fault dictionary which is presented by shaded columns in the Table I.

3.4 Grouping ununiquely defined faults

Finally, the optimization procedure applies the same rules on the ununiquely defined faults. The step for finding minimal number of test frequencies is not needed any more. The procedure groups the remaining ununiquely defined faults together. If the number of ambiguously defined faults is *m* the procedure forms groups of *n* faults where *n*=2,…, *m*. If all *m* faults have the same signature there exists only one group of *m* ambiguous faults. These are the faults which have the identical codes.

4. CONCLUSION

This paper presents very efficient fault dictionary optimization technique using binary logical manipulation algorithm. Since it handles only numbers, can be applied in all fault dictionary based techniques. This technique is using brute force and is more likely to find optimal solution than similar techniques. The algorithm saves time expectualy in the minimal test frequencies exploring stage.

It has been applied on ambiguity sets formed by the method in the frequency domain and the results has been presented. The universality of the technique has been illustrated by an application on two different approaches. Applying this technique on ambiguity sets formed by Hochwald and Bastian resulted in the same minimal number of test nodes [5].

REFERENCES

[1] D. Jurišić, N. Mijat and V. Čosić, Frequency domain approach to Fault Diagnosis of Analog Filters, *Proc. Third IEEE International Conference on Electronics Circuits and systems ICECS '96* (Rodos, Greece), 2, 1996, 1135-1138.

[2] D. Jurišić, *Fault analysis of analog networks using fault dictionary*, (Ms Thesis, Zagreb, Croatia, 1995, in Croatian)

[3] D. Jurišić, V. Čosić and N. Mijat, Fault analysis of analog networks using fault dictionary, *Proc. 40th anniversary conference KoREMA* (Zagreb, Croatia), 1995, 474-477 (in Croatian).

[4] J. W. Bandler and A. E. Salama, Fault diagnosis of analog circuits, *Proc. IEEE,* 73(8), 1985, 1279-1325.

[5] W. Hochwald and J. D. Bastian, A dc approach for analog fault dictionary determination, *IEEE Trans. Circuits Syst.*, CAS-26, 1979, 523-529.