Analog Integr Circ Sig Process (2015) 82:435–448
DOI 10.1007/s10470-014-0469-8

A new test points selection method for analog fault dictionary techniques

Dongsheng Zhao · Yuzhu He

Received: 21 April 2014 / Accepted: 2 December 2014 / Published online: 12 December 2014

Abstract A new efficient method to select an optimum or near optimum set of test points for fault dictionary techniques in analog fault diagnosis is proposed. This is done by constructing a fault-isolated table firstly according to the integer-coded fault dictionary, and picking the special test points with particular fault isolate abilities out from the candidate test points set. This step can help us save the total cost of computation and even find the global minimum test-point set directly. After this step, the selected test points might isolate most of the defined faults together, and these isolated faults (rows) and the chosen test points (columns) are deleted from the fault dictionary, the dimension of which will reduce quite a lot. And then the inclusive and exclusive approach are combined together to choose other optimum test points from the candidate test points set to solve the problem. Three analog circuits' examples and the statistical experiments are given to demonstrate the feasibility and effectiveness of the proposed algorithm, the existing classical algorithms are also used to do the comparison. The results indicate that the proposed method has a better tradeoff between the solution accuracy and the computational cost, and it is superior to other methods in its computational efficiency and quality of final solution. Therefore, it is a good solution to minimize the size of the test-point set, and practical for medium and large scale systems.

Keywords Analog fault diagnosis · Integer-coded fault dictionary · Fault-isolated table · Test points selection

D. Zhao (✉) · Y. He
School of Instrumentation Science and Opto-electronics Engineering, Beihang University, Beijing 100191, China
e-mail: mailzds@126.com

1 Introduction

The analog circuit diagnosis methods are classified into two main categories: the simulation before test (SBT) and the simulation after test (SAT) approach [2–4, 7]. And the fault dictionary is a very important and practical method of SBT approach, especially in the diagnosing of catastrophic faults. A fault dictionary is a set of measurements of the circuit under test (CUT) simulated under potentially faulty conditions (including fault-free case) and organized before the test. The measurements could be at different test points, test frequencies, and sampling times [12, 15].

There are three important phases in the fault dictionary approach [9]. First of all, a network is simulated for each of the anticipated faults (including fault-free case) excited by the chosen stimuli (dc or ac), and the signatures of the responses are stored and organized in the dictionary for use. The second phase is the selection of test points. An optimum selection of test points is the main work of this stage. By doing this, we can achieve the desired degree of fault diagnosis with less test points and save the fault test and diagnosis time greatly. The last phase is fault isolation. At the time of testing, the CUT is excited by the same stimuli that are used in constructing the dictionary, and measurements are made at the preselected test points. They are compared with the values stored in the fault dictionary to identify the fault according to the preset criteria. This paper mainly focuses on the second phase.

Although the fault dictionary method has the advantage of minimum on-line computation time, a significant off-line computation is needed during the construction of the fault dictionary. To reduce the computation time and the dimension of the fault dictionary, the optimum selection of test points is especially important. It can also eliminate the redundant measurements and save the total cost of the analog circuit test greatly. But the global minimum set of test points can only be guaranteed by the exhaustive search method, which has been proven to be NP-hard [10, 12]. As pointed out in [1], the exhaustive search method is limited to small or near medium sized analog systems. For medium and large systems that can be found everywhere nowadays, whose fault dictionaries always with more than 40 faults and more than 40 test points, it is impractical to use the exhaustive search algorithm as its expensive computational cost, the tradeoff between the desired degree of fault diagnosis and the computational cost is to select a near minimum set (or a local minimum set) [15–17].

The test-point selection problem for the analog circuit fault dictionary has been studied extensively in many papers. Varghese [14] proposed a heuristic method based on given performance indexes to find the sets of test points. Hochwald and Bastian [6] proposed the concept of ambiguity sets and developed logical rules to select the test points. Lin and Elcherif [7] proposed two heuristic methods based on the two criteria proposed by Hochwald and Bastian. Stenbakken and Souders [13] proposed QR factorization for the circuit sensitivity matrix. Spaandonk and Kevenaar [11] proposed to select the test-point set by combining the decomposition method of the system's sensitivity matrix and an iterative algorithm. Prasad and Babu [9] proposed four algorithms based on three strategies of inclusive approach and three strategies of exclusive approach. Pinjala and Kim [8] proposed a method to find the test points set by computing the information content of all the candidate test points. Starzyk [12] proposed an entropy based approach to select the near minimum test-point set. Golonek and Rutkowsk [5] used a genetic algorithm based method to determine the optimal set of test points. Yang and Tian [15] used the graph node search method to find the near minimum test-point set.

The Integer-Coded fault dictionary technique was first proposed by Lin and Elcherif [7]. And this technique has been proven to be a very effective tool for the optimum test-point selection problem. The test-point selection algorithms in [1, 5, 8, 9, 12, 15–17] are all based on this technique. Each of these algorithms consists of a test-point selection strategy and a test-point selection criterion. The strategy of test-point selection method can be classified into the inclusive category and the exclusive category [9, 12]. And each test-point selection algorithm has its own criterion to select the optimum test points. In paper [12], an entropy-based measure is defined to choose the optimum test points. In paper [9], the number of ambiguity sets, the number of faults in a test point's biggest ambiguity set and the spread of faults in ambiguity sets of the test point are used to choose the optimum test points. In paper [1, 5, 8, 15], some special information contents are proposed as the criterion for test points selection. According to the existing methods, the different test points may be selected to construct the optimal test points set based on different criterion for the same CUT. A selected test-point set without redundant test points does not mean that this set is a minimum set. And seldom references are related to how to judge the effectiveness of the criterion until now. So the criterion is especially important for the test-point selection problem.

Since 90 % of all the analog faults found in practice are catastrophic faults [6, 15], and most of them are single faults that can be known from our experience, single catastrophic faults in analog circuits are considered in this paper.

A new method to select a near minimum test points set is proposed and compared with the other reported methods in this paper. In Sect. 2, the integer-coded fault dictionary approach and some classical test-point selection criteria are summarized. The new proposed algorithm is described in Sect. 3. Section 4 demonstrates the excellent performance in the solution efficiency and quality of the proposed algorithm by comparing it with other reported algorithms based on different kinds of experiments. Finally, brief conclusions are given in Sect. 5.

The nomenclatures of this paper are as follows:

n_j	The test point j
f_i	The fault i
N_T	Number of candidate test points
N_f	Number of all the faults (including fault-free case)
N_{Aj}	Number of ambiguity sets in test point n_j
NI_i	Number of test points that can isolate fault f_i
F_{ij}	Total number of faults contained in ambiguity set i of test point n_j
S_{opt}	Desired test-point set
S_c	Candidate test-point set
q_j	Number of faults that can be isolated by test point n_j
k_i	The number of test points that can isolate fault f_i.

2 Integer-coded fault dictionary and classical test-point selection criteria

2.1 Integer-coded fault dictionary

Ambiguity group is defined as that any two faulty cases fall into the same ambiguity set if the gap between the voltage values of their responses is less than 0.7 volts [6, 15–17]. The fault dictionary provides information about the ambiguity sets for each test point and the mutual information among them [8]. The purpose of test-point selection is to separate all the faults (including fault-free case) with a minimum number of test points, and reduce the dimension of the fault dictionary.

To construct a fault dictionary, all the potential faults are defined and listed first, and the stimuli are selected. The CUT is then simulated under the fault free and all the defined fault cases. The signatures of the responses are recorded and listed in the fault dictionary. In the fault dictionary, rows represent different faults (including fault-free case), and the columns show all the test points [15–17]. Since all the test points are independent from each other, the ambiguity groups of each test point can be numbered by the same integers without confusion. By this way, the integer-coded fault dictionary is generated. Assume that the voltages shown in Table 1 are the simulation results of a given analog circuit under different faulty conditions ((including the nominal case). Table 2 shows the ambiguity sets of Table 1, and Table 3 gives the integer-coded fault dictionary derived from Table 2. For more details about fault dictionary and ambiguity set can be found in [7] [12] [15] [9] [16] [17].

2.2 Classical test-point selection criteria

As the test-point selection is so important and significant that plenty of research work has been done by the researchers all over the world. There are some classical test-point selection criteria need to be clarified. The criteria for inclusive category are listed as follows [16]:

Criterion 1 (C1): $\underset{j}{Max}(N_{Aj})$, which means to select the test point $\mathrm{n_j}$ that has the largest number of ambiguity sets among all the candidate test points [9].

Criterion 2 (C2): $\underset{j}{Min}(\underset{i}{Max}(F_{ij}))$, which means firstly to get the largest number of faults contained in all ambiguity sets of every candidate test points, and then choose the one that has the smallest value of all [9].

Criterion 3 (C3): $\underset{j}{Min}(s_j) = \underset{j}{Min}(\sum_{i=1}^{N_{Aj}}(F_{ij} - N_f/N_{Aj})^2/N_{Aj})$, which means to choose the test point $\mathrm{n_j}$ that has the smallest s_j [9].

Criterion 4 (C4): $\underset{j}{Max}(q_j)$, which means to select the test point $\mathrm{n_j}$ that has the largest ability to isolate faults [8].

Criterion 5 (C5): $\underset{j}{Min}(E(j)) = \underset{j}{Min}(\sum_{i=1}^{N_{Aj}} F_{ij}\ln F_{ij})$, which means to select the point $\mathrm{n_j}$ that has the smallest entropy index $E(j)$ [12].

Criterion 6 (C6): $\underset{j}{Max}(I(x_j)) = \underset{j}{Max}(\sum_i I(f_i)) = \underset{j}{Max}(\sum_i - \ln(k_i/N_T))$, where $I(f_i)$ is the information content of the fault f_i, and $I(x_j)$ is the sum of $I(f_i)$ of faults that can be isolated by test point $\mathrm{n_j}$. The test point that has the largest $I(x_j)$ will be selected [15].

Table 1 Fault dictionary

Faults	n_1 (V)	n_2 (V)	n_3 (V)	n_4 (V)
f_1	9.2	7.7	2.2	3.6
f_2	6.2	6.2	4.3	7.0
f_3	7.4	7.8	3.2	2.5
f_4	5.4	10.6	6.8	9.5
f_5	3.9	4.6	1.8	5.2
f_6	4.4	5.9	4.7	5.4
f_7	3.5	6.7	1.6	3.8
f_8	4.1	9.1	5.2	4.7
f_9	3.8	8.8	8.2	5.0

Table 3 Integer-coded fault dictionary of Table 2

Faults	n_1	n_2	n_3	n_4
f_1	4	2	0	1
f_2	2	1	2	3
f_3	3	2	1	0
f_4	1	4	3	4
f_5	0	0	0	2
f_6	0	1	2	2
f_7	0	1	0	1
f_8	0	3	2	2
f_9	0	3	4	2

Table 2 Ambiguity sets of Table 1

Ambiguity sets	n_1	n_2	n_3	n_4
0	(3.5–4.4 V) f_5, f_6, f_7, f_8, f_9	(4.6 V) f_5	(1.6–2.2 V) f_1, f_5, f_7	(2.5 V) f_3
1	(5.4 V) f_4	(5.9–6.7 V) f_2, f_6, f_7	(3.2 V) f_3	(3.6–3.8 V) f_1, f_7
2	(6.2 V) f_2	(7.7–7.8 V) f_1, f_3	(4.3–5.2 V) f_2, f_6, f_8	(4.7–5.4 V) f_5, f_6, f_8, f_9
3	(7.4 V) f_3	(8.8–9.1 V) f_8, f_9	(6.8 V) f_4	(7.0 V) f_2
4	(9.2 V) f_1	(10.6 V) f_4	(8.2 V) f_9	(9.5 V) f_4

The criteria for exclusive category are listed as follows [16]:

Criterion 7 (C7): $\underset{j}{Min}(N_{Aj})$

Criterion 8 (C8): $\underset{j}{Max}(\underset{i}{Max}(F_{ij}))$

Criterion 9 (C9): $\underset{j}{Max}(s_j) = \underset{j}{Max}(\sum_{i=1}^{N_{Aj}} (F_{ij} - N_f/\ N_{Aj})^2/ N_{Aj})$

Criterion 10 (C10): $\underset{j}{Min}(q_j)$

Criterion 11 (C11): $\underset{j}{Max}(E(j)) = \underset{j}{Max}(\sum_{i=1}^{N_{Aj}} F_{ij} \ln F_{ij})$

Criterion 12 (C12): $\underset{j}{Min}(I(x_j)) = \underset{j}{Min}(\sum_{i} I(f_i)) = \underset{j}{Min}(\sum_{i} -\ln(k_i/N_T))$

3 New algorithm for test-point selection

In many cases, some faults can only be detected and isolated by some special test points, whereas the others can be isolated by more than one test points [15]. It is much more difficult to isolate these faults by the criteria listed above as they do not take this important factor into consideration. If we could select these special test points firstly to isolate these particular faults, the problem of test-point selection would be solved easily and the total computational cost would reduce a lot.

In order to find these special test points, we propose to construct a fault-isolated table and extend it, and then combine it with the integer-coded fault dictionary to find the final solution. In this section, we will describe this new method in detail.

3.1 Construction of fault-isolated table

Since the integer-coded fault dictionary shows different ambiguity sets' information of different test points, we need to find the ambiguity sets that contain only one fault for each test point and mark them out. First of all, an all-zero table that has the same row and column as the integer-coded fault dictionary is constructed. For each fault in the fault dictionary, consider the integers along its row. Each integer in that row is compared with all the integers in its corresponding column. If the integer is not repeated at any row of the corresponding column, then the element "0" at the same place of the all-zero table is replaced by "1". After all the integers of the integer-coded fault dictionary have been dealt with by this way, the "0" elements of all the corresponding places of the all-zero table have been replaced by "1". The new table thus formed is the fault-isolated table. In the fault-isolated table, the element "1" means the fault of this row can be isolated by the test point of this column, and the element "0" means opposite. Table 4 gives the fault-isolated table based on the integer-coded fault dictionary that shows in Table 3.

In Table 4, we can see the fault isolation ability of every test point clearly. In order to use this fault-isolated table to select the special test points easily and quickly, we extended it to obtain the extended fault-isolated table.

3.2 Extension of the fault-isolated table

Since the faulty isolation ability of each test point is listed in the fault-isolated table, we need a search method to find the special test point more easily. We define a new column NI_i to calculate the total number of test points that can isolate fault f_i. As element "1" of the fault-isolated table shows the faulty isolation ability of the corresponding column, we sum all the elements "1" of the row f_i to calculate NI_i. Different NI_i values have different meanings. The NI_i value is one means that the fault f_i can be isolated by only one special test point, and the corresponding test point (column) of element "1" is the right special test point. The NI_i value is more than one means that there are more than one test points that can isolate fault f_i. The NI_i value is zero means that there is no test point that can isolate fault f_i alone.

Table 4 Fault-isolated table

Faults	n_1	n_2	n_3	n_4
f_1	1	0	0	0
f_2	1	0	0	1
f_3	1	0	1	1
f_4	1	1	1	1
f_5	0	1	0	0
f_6	0	0	0	0
f_7	0	0	0	0
f_8	0	0	0	0
f_9	0	0	1	0

Table 5 Extended fault-isolated table

Faults	n_1	n_2	n_3	n_4	NI_i
f_1	1	0	0	0	1
f_2	1	0	0	1	2
f_3	1	0	1	1	3
f_4	1	1	1	1	4
f_5	0	1	0	0	1
f_6	0	0	0	0	0
f_7	0	0	0	0	0
f_8	0	0	0	0	0
f_9	0	0	1	0	1

The extended fault-isolated table of Table 4 shows in Table 5. From the Table 5, we can find that f_1 can only be isolated by test point n_1, f_5 can only be isolated by test point n_2 and f_9 can only be isolated by test point n_3, f_2, f_3 and f_4 can be isolated by more than one test points, and f_6, f_7 and f_8 can not be isolated by any test points alone.

3.3 Algorithm for test-point selection

Our new algorithm mainly contains two parts, the inclusive part and the exclusive part. In the inclusive part, we use the new proposed fault-isolated table and its extended table to pick the special test points out from S_c to S_{opt}, and then judge whether the test points of S_{opt} can isolate all the listed faults or not. If they can, exit part 1, else, use one of the inclusive criteria to choose more optimum test points from S_c to S_{opt}. Repeat this until the test points of S_{opt} can isolate all the faults or the remainder faults can not be isolated by these existing test points. In the exclusive part, we use the advantage of exclusive technique to remove the redundant test points from S_{opt} in case of existing redundancy.

The proposed new algorithm is given as follows.

Part 1, the inclusive part:

Step (1) Initialize the desired test-point set S_{opt} as a null set, and let S_c consist of all the candidate test points. The integer-coded fault dictionary is constructed based on the voltage values of their responses. The rows of fault dictionary list all the N_f faults (including the nominal case), and the columns of it represent N_T test points. The fault-isolated table which is the same size as the fault dictionary is initialized with all zeros.

Step (2) Construct the fault-isolated table and extend it to obtain the extended fault-isolated table.

Step (3) Check the column NI_i of extended fault-isolated table, find all the corresponding test points (columns) of which the NI_i value equals to one, and add these test points to S_{opt}.

Step (4) Check the stop conditions. If the test points of S_{opt} can isolate all the faults or the remainder faults can not be isolated unless more test points are increased, stop the repeat. If the stop conditions don't fulfill, the algorithm goes to step 5.

Step (5) Rearrange the fault dictionary and eliminate the rows of the fault dictionary that can be isolated by test points in S_{opt} together. Remove the selected test points from S_c and eliminate the corresponding columns of the fault dictionary.

Step (6) Use one of the inclusive criteria (C1–C6) to select the optimum test point and add it to S_{opt}. In case of a tie, choose one among them. Then go to step 4.

Part 2, the exclusive part:

Step (7) Draw out a test point n_j from S_{opt} by using any one of the exclusive criteria (C7–C12).

Step (8) Check whether the residual test points in S_{opt} can isolate all the listed faults together. If yes, means that the test point n_j is redundant and should be removed from S_{opt}, else, add n_j back to S_{opt}, and then go to step 7.

Step (9) Repeat step 7 to step 8 until all the test points of S_{opt} are examined.

Remark 1 In step 3, we pick the special test points out and add them to S_{opt}, this step can help us find the optimum test points faster, and under some special conditions, we can find the final results directly. This will improve the efficiency of our algorithm greatly.

Remark 2 In step 6, the results' computation is related to the existing test points in S_{opt} and every candidate test point in S_c.

Now we use the data in Table 5 to illustrate the process of finding the special test points with particular fault isolate abilities in step 3. Search along the NI_i column, we can easily find that f_1, f_5 and f_9 have the NI_i value equal to one, and their corresponding test points (column) are n_1, n_2 and n_3 respectively. And then we add these special test points to S_{opt} and go to step 4. In this step, we find the test points in S_{opt} can isolate all the listed faults together, and the stop condition fulfills. So there is no need to repeat step 5 to step 6 to find any more optimum test points, we directly go to the exclusive part. After checking, there are no redundant test points in S_{opt} and we find the final solution $S_{opt} = \{n_1, n_2, n_3\}$, which is composed of all the special test points selected in step 3. Therefore, the proposed method could reduce the total computational cost and find the final solution with higher efficiency.

Theorem 1 The time complexity of the proposed algorithm is less than $O(N_f m N_T \log N_f)$.

Proof As discussed above, the fault-isolated table consists of N_f rows and N_T columns, and the extended fault-isolated table has one more column. The time complexity of step 2 is $O(N_f(N_T + 1))$.

In step 3, suppose m_1 test points are added to S_{opt}, and these points can isolate N_{f1} faults, and the time complexity is also $O(N_f(N_T + 1))$. In case of $N_{f1} = N_f$, the stop condition fulfills and we find the S_{opt} directly, and the time complexity will reduce quite a lot.

In step 6, suppose m_2 test points are added into S_{opt} in the next algorithm iterations, and the time complexity is $O((N_f - N_{f1})p' \log(N_f - N_{f1}))$, where $p' = (N_T - m_1) + (N_T - m_1 - 1) + \cdots + (N_T - m_1 - m_2 + 1)$.

In step 5, delete all the corresponding rows and columns in the fault dictionary has the time complexity of

$O(N_f) + O(m_1 + m_2) = O(N_f) + O(m)$. Where m is the total number of test points in Sopt ($m = |S_{opt}|$). Since the total number of test points $N_T > m$ in practice, $p' > \, > N_T$.

In the exclusive part, the time complexity is $O(N_f m \log N_f)$.

Thus the total time complexity is:

$$\begin{aligned}
&O(N_f(N_T+1)) + O((N_f - N_{f1})p' \log(N_f - N_{f1})) + O(N_f) \\
&+ O(m) + O(N_f m \log N_f) \\
&= O(N_f(N_T+1) + (N_f - N_{f1})p' \log(N_f - N_{f1}) + N_f \\
&+ m + N_f m \log N_f) \\
&\le O(N_f N_T + N_f p' \log N_f + N_f m \log N_f) \\
&= O(N_f p' \log N_f + N_f m \log N_f) \\
&\le O(N_f m_2 N_T \log N_f + N_f m \log N_f) \\
&= O(N_f m_2 N_T \log N_f + N_f (m_1 + m_2) \log N_f) \\
&= O(N_f m_2 N_T \log N_f + N_f m_1 \log N_f) \\
&\le O(N_f m N_T \log N_f)
\end{aligned}$$

4 Experiments

In order to demonstrate the feasibility and effectiveness of the new algorithm, we did different kinds of experiments. Since there are six different inclusive criteria (C1–C6) and six different exclusive criteria (C7–C12), and Yang has demonstrated in [16] that no one specific criterion is superior to the others in the accuracy and efficiency of finding the final solution, we chose C1, C4, C5 and C7 to compose our new algorithm 1 (C1 and C7), new algorithm 2 (C4 and C7) and new algorithm 3 (C5 and C7) to do the experiments.

4.1 Experiment on the circuits

4.1.1 Bandpass filter circuit example

The filter circuit with the nominal parameter values is shown in Fig. 1. This is the same example as in [8, 9, 12, 15–17]. The excitation signal is a 1-kHz, 4-V sinusoidal wave. Totally, there are nineteen potential faults f_1 to f_{19} (including the nominal case) and eleven test points n_1 to n_{11}. The responses of voltage values at all test points for different faulty conditions are obtained by PSPICE simulation and the integer-coded fault dictionary is constructed by procedures introduced in Sect. 2, and the results are shown in Table 6.

In step 1 of our algorithm, the initialized work is done. Since there are nineteen potential faults and eleven test points, the candidate test-point set S_c is initialized as $\{n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8, n_9, n_{10}, n_{11}\}$, and $N_T = 11$,

Table 6 Integer-coded fault dictionary for Fig. 1

Faults	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9	n_{10}	n_{11}
f_1(NOM)	3	2	2	3	3	3	4	4	1	1	1
f_2(R1 open)	0	0	0	0	0	0	0	0	1	1	7
f_3(R1 short)	3	2	2	3	4	3	4	4	1	1	0
f_4(R2 open)	1	0	0	0	0	0	0	0	1	1	7
f_5(R2 short)	2	3	3	4	6	5	6	6	1	1	2
f_6(R3 open)	1	1	1	1	1	1	1	1	1	1	5
f_7(R4 open)	0	0	0	2	3	2	3	3	1	1	8
f_8(R5 open)	3	2	2	3	5	4	5	5	1	1	4
f_9(R5 short)	3	2	2	3	0	0	0	0	1	1	7
f_{10}(R6 open)	3	2	2	3	6	6	7	7	1	1	6
f_{11}(R6 short)	3	2	2	3	2	0	0	0	1	1	7
f_{12}(R7 open)	3	2	2	3	3	3	4	2	1	1	4
f_{13}(R7 short)	3	2	2	3	3	2	0	8	5	1	8
f_{14}(R8 open)	3	2	2	3	3	2	2	8	4	1	8
f_{15}(R9 open)	3	2	2	3	3	3	4	4	1	1	0
f_{16}(R9 short)	3	2	2	3	3	3	4	4	2	1	8
f_{17}(R10 open)	3	2	2	3	3	3	4	4	3	1	8
f_{18}(R11 open)	3	2	2	3	3	3	4	4	2	2	3
f_{19}(R12 open)	3	2	2	3	3	3	4	4	0	0	1

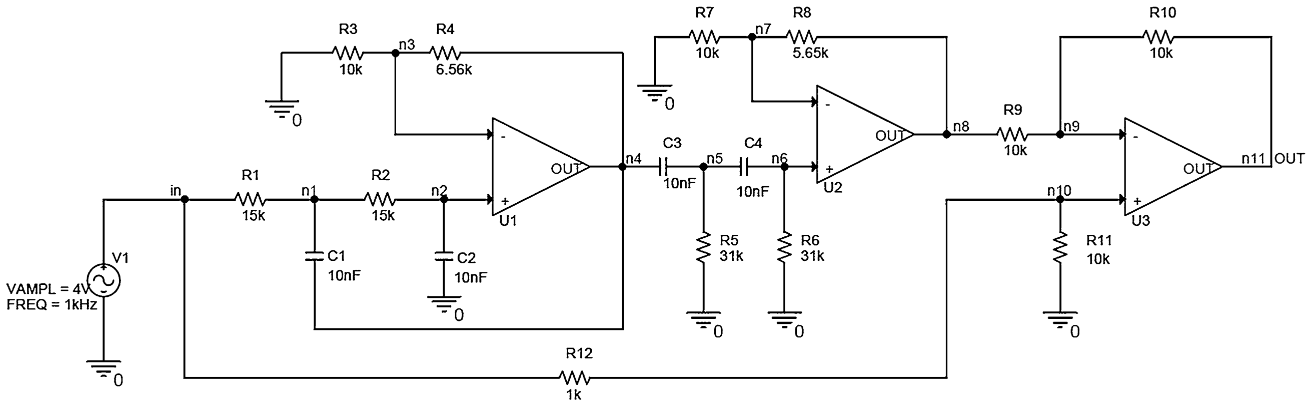

Fig. 1 Bandpass filter circuit

Table 7 Extended fault-isolated table for Fig. 1

Faults	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9	n_{10}	n_{11}	NI_i
f_1(NOM)	0	0	0	0	0	0	0	0	0	0	0	0
f_2(R1 open)	0	0	0	0	0	0	0	0	0	0	0	0
f_3(R1 short)	0	0	0	0	1	0	0	0	0	0	0	1
f_4(R2 open)	0	0	0	0	0	0	0	0	0	0	0	0
f_5(R2 short)	1	1	1	1	0	1	1	1	0	0	1	8
f_6(R3 open)	0	1	1	1	1	1	1	1	0	0	1	8
f_7(R4 open)	0	0	0	1	0	0	1	1	0	0	0	3
f_8(R5 open)	0	0	0	0	1	1	1	1	0	0	0	4
f_9(R5 short)	0	0	0	0	0	0	0	0	0	0	0	0
f_{10}(R6 open)	0	0	0	0	0	1	1	1	0	0	1	4
f_{11}(R6 short)	0	0	0	0	1	0	0	0	0	0	0	1
f_{12}(R7 open)	0	0	0	0	0	0	0	1	0	0	0	1
f_{13}(R7 short)	0	0	0	0	0	0	0	0	1	0	0	1
f_{14}(R8 open)	0	0	0	0	0	0	1	0	1	0	0	2
f_{15}(R9 open)	0	0	0	0	0	0	0	0	0	0	0	0
f_{16}(R9 short)	0	0	0	0	0	0	0	0	0	0	0	0
f_{17}(R10 open)	0	0	0	0	0	0	0	0	1	0	0	1
f_{18}(R11 open)	0	0	0	0	0	0	0	0	0	1	1	2
f_{19}(R12 open)	0	0	0	0	0	0	0	0	1	1	0	2

Table 8 Simplified integer-coded fault dictionary for Fig. 1

Faults	n_1	n_2	n_3	n_4	n_6	n_7	n_{10}	n_{11}
f_1(NOM)	3	2	2	3	3	4	1	1
f_2(R1 open)	0	0	0	0	0	0	1	7
f_4(R2 open)	1	0	0	0	0	0	1	7
f_9(R5 short)	3	2	2	3	0	0	1	7
f_{15}(R9 open)	3	2	2	3	3	4	1	0
f_{16}(R9 short)	3	2	2	3	3	4	1	8
f_{18}(R11 open)	3	2	2	3	3	4	2	3

$N_f = 19$. In step 2, the fault-isolated table and its extended table are constructed by the procedures introduced in Sect. 3. Table 7 shows the extended fault-isolated table. In this table, we can easily find the special test points that have particular fault isolate abilities. They are n_5, n_8 and n_9 that can isolate faults f_3 and f_{11}, f_{12}, f_{13} and f_{17} respectively. Therefore, test points n_5, n_8 and n_9 are selected first and added to S_{opt} in step 3. After checking the stop condition in step 4, test points n_5, n_8 and n_9 can not isolate all the nineteen potential faults together, so the algorithm goes to step 5. In this step, the integer-coded fault dictionary is rearranged and the corresponding rows of faults f_3, f_5, f_6, f_7, f_8, f_{10}, f_{11}, f_{12}, f_{13}, f_{14}, f_{17} and f_{19}, which can be isolated by n_5, n_8 and n_9 together are eliminated, and the corresponding columns of test points n_5, n_8 and n_9 are also removed from the integer-coded fault dictionary. Now, $S_c = \{n_1, n_2, n_3, n_4, n_6, n_7, n_{10}, n_{11}\}$, $N_T = 8$, and $N_f = 7$. Thus we obtain the new simplified integer-coded fault dictionary that shows in Table 8. From this table, we can see that the size of the integer-coded fault dictionary is reduced greatly, and this will surely save the total cost of computation tremendously in case of there are more faults and test points in the CUT. In step 6, test points n_1 and n_{11} are selected according the inclusive criterion (we choose C1, C4 and C5 respectively). Until now, the test points of S_{opt}, n_1, n_5, n_8, n_9 and n_{11} can isolate all the defined faults and the stop condition fulfills. In order to check and remove the redundant test points from S_{opt}, the algorithm goes to the exclusive part. In this part, the test point n_8 is judged to be the redundant test point and deleted from S_{opt}. Finally, we obtain the optimum test-point set $S_{opt} = \{n_1, n_5, n_9, n_{11}\}$, which is the same results as that obtained in [8, 9, 12, 15–17].

4.1.2 Experiment on a negative feedback circuit

The proposed algorithm can reduce the size of the fault dictionary and help us choose the optimum test points. Similarly, changing the test stimuli can do the same assist. This example is used to clarify how to construct the fault-isolated table and use the proposed algorithm to obtain the optimum test points set under multi-measurements condition.

The negative feedback circuit is shown in Fig. 2. The input signal is a 1-kHz, 7 mV sinusoidal wave. Totally, there are 35 potential faults f_1 to f_{35} (including the nominal case) and ten test points n_1 to n_{10}. The responses of all the test points under different faulty conditions are obtained by PSPICE simulation. Yang [15] has illustrated that there are many faults can not be isolated by dc responses as no more test points can be selected, combining the dc and ac responses of the test points can solve the problem. In order to illustrate the effectiveness of the proposed algorithm and compare with other algorithms, we choose the same gap and method as Yang [15] to construct the combined integer-coded fault dictionary which is shown in Table 9. In

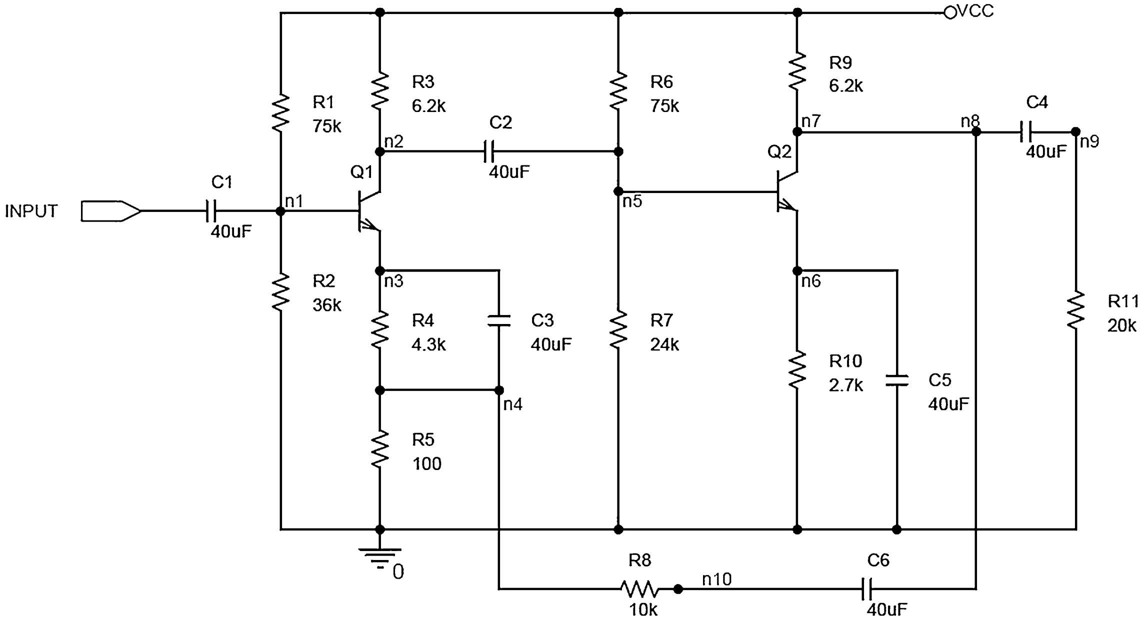

Fig. 2 Negative feedback circuit

the table, "D" represents the integer code of dc voltage responses, and "A" represents the integer code of ac voltage responses. The extended fault-isolated table is shown in Table 10. The special tests t_1, t_4, t_9, t_{19}, t_{20} are selected first and added to S_{opt} according to the proposed algorithm. Finally, our proposed algorithm 1, algorithm 2 and algorithm 3 obtain the same $S_{opt} = \{t_1, t_3, t_4, t_5, t_9, t_{13}, t_{18}, t_{19}, t_{20}\}$. The other existing classical algorithms are used to select the optimum test points, and the results are listed in Table 11. All the algorithms are programmed by MATLAB. In Table 11, we can obviously see that Starzyk's and Pinjala's algorithms have not found the global minimum set of test points, but Yang's algorithm and our proposed algorithms have found the minimum set whose size is the same as the exhaustive search algorithm. That the new proposed algorithms have found a different S_{opt} from the Yang's and the exhaustive search method illustrates that a CUT could have more than one different minimum test point sets. The elapsed time column shows different algorithms' computation cost. From this column we can clearly see that the exhaustive search algorithm cost too much more than other algorithms and our new proposed algorithms spend the shortest time of all. And there are not great differences among our new proposed three algorithms (algorithm 1, algorithm 2 and algorithm 3) in finding the final solution. This just reflects the high efficiency of our new method. Assume the CUT has a larger scale and there are more test points and defined faults, our new method will greatly save the computation time.

In fact, no matter what kinds of measurements are used to simulate the CUT, if only the integer-coded fault dictionary and the fault-isolated table can be constructed, our proposed algorithm can be used to choose the near minimum set of test points.

4.1.3 Leapfrog filter circuit example

Since there are more and more application specific integrated circuit (ASIC) and very large scale integration (VLSI) in practice nowadays, isolating faults to the module level is enough. After listing all the potential module failure modes and choosing the right stimuli, we can simulate the CUT by PSPICE and then construct the integer-coded fault dictionary and the fault-isolated table to select the optimum test points. This example will show the use of proposed algorithm to the module level fault isolation of the CUT.

The leapfrog filter circuit is shown in Fig. 3. We can find six operational amplifiers in the circuit and these amplifiers can be treated as modules in practice. We assume the amplifiers have low-input-impedance failure besides the resistance and capacitance's catastrophic faults. The input signal is a 1-kHz, 4 V sinusoidal wave. Totally, there are 23 potential faults f_1 to f_{23} (including the nominal case) and 12 test points n_1 to n_{12}. The responses of all the test points under different faulty conditions are obtained by PSPICE simulation. According to the response voltages of the test points under different faulty mode, we construct the integer-coded fault dictionary and the extended fault-isolated table which is shown in Table 12 and Table 13 respectively. The special test points n_1, n_2, n_3, n_5, n_7, n_{11} are selected first and added to S_{opt} according to the

Table 9 Integer-coded fault dictionary for Fig. 2

Faults	n_1		n_2		n_3		n_4		n_5		n_6		n_7		n_8		n_9		n_{10}	
	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A
	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}	t_{20}
f_1(NOM)	2	0	2	0	1	0	0	0	2	0	1	0	4	7	4	7	0	7	0	7
f_2(R1 open)	0	0	3	0	0	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
f_3(R2 open)	3	0	1	0	3	0	0	0	2	0	1	0	4	6	4	6	0	6	0	6
f_4(R3 open)	1	0	0	0	0	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
f_5(R4 open)	2	0	3	0	2	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
f_6(R5 open)	2	0	3	0	2	0	1	0	2	0	1	0	4	0	4	0	0	0	1	0
f_7(R6 open)	2	0	2	6	1	0	0	0	0	6	0	0	5	0	5	0	0	0	0	0
f_8(R7 open)	2	0	2	0	1	0	0	0	3	0	3	0	1	1	1	1	0	1	0	1
f_9(R8 open)	2	0	2	1	1	0	0	0	2	1	1	0	4	10	4	10	0	10	0	10
f_{10}(R9 open)	2	0	2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
f_{11}(R10 open)	2	0	2	4	1	0	0	0	2	4	2	0	5	2	5	2	0	2	0	2
f_{12}(R1 short)	4	0	3	0	4	0	0	0	2	0	1	0	4	3	4	3	0	3	0	3
f_{13}(R3 short)	2	0	3	0	1	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
f_{14}(R4 short)	1	0	0	0	0	0	0	0	2	0	1	0	4	5	4	5	0	5	0	5
f_{15}(R5 short)	2	0	2	8	1	0	0	0	2	7	1	0	3	9	3	9	0	9	0	9
f_{16}(R6 short)	2	0	2	0	1	0	0	0	5	0	4	0	5	0	5	0	0	0	0	0
f_{17}(R7 short)	2	0	2	0	1	0	0	0	0	0	0	0	5	0	5	0	0	0	0	0
f_{18}(R8 short)	2	0	2	0	1	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
f_{19}(R9 short)	2	0	2	2	1	0	0	0	2	2	1	0	5	0	5	0	0	0	0	0
f_{20}(R11 short)	2	0	2	2	1	0	0	0	2	2	1	0	4	0	4	0	0	0	0	0
f_{21}(C2 open)	2	0	2	7	1	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
f_{22}(C3 open)	2	0	2	0	1	0	0	0	2	0	1	0	4	3	4	3	0	3	0	3
f_{23}(C4 open)	2	0	2	0	1	0	0	0	2	0	1	0	4	7	4	7	0	0	0	7
f_{24}(C5 open)	2	0	2	3	1	0	0	0	2	3	1	1	4	4	4	4	0	4	0	4
f_{25}(C6 open)	2	0	2	1	1	0	0	0	2	1	1	0	4	10	4	10	0	10	0	0
f_{26}(C2 short)	2	0	1	0	1	0	0	0	4	0	3	0	1	0	1	0	0	0	0	0
f_{27}(C4 short)	2	0	2	0	1	0	0	0	2	0	1	0	2	7	2	7	1	7	0	7
f_{28}(C6 short)	2	0	2	0	1	0	0	0	2	0	1	0	1	7	1	7	0	7	2	7
f_{29}(Q1B open)	2	0	3	0	0	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
f_{30}(Q1C open)	1	0	3	0	0	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
f_{31}(Q1BC short)	3	0	1	0	3	0	0	0	2	0	1	0	4	8	4	8	0	8	0	8
f_{32}(Q1CE short)	2	0	1	0	3	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
f_{33}(Q2B open)	2	0	2	5	1	0	0	0	2	5	0	0	5	0	5	0	0	0	0	0
f_{34}(Q2BC short)	2	0	2	0	1	0	0	0	3	0	3	0	1	0	1	0	0	0	0	0
f_{35}(Q2CE short)	2	0	2	5	1	0	0	0	2	5	3	0	1	0	1	0	0	0	0	0

proposed method. Finally, our proposed algorithm 1, algorithm 2 and algorithm 3 obtain the same optimum test-point set $S_{opt} = \{n_1, n_2, n_3, n_4, n_7, n_8, n_{11}\}$. The other existing classical algorithms are also used to select the optimum test points, and the results are listed in Table 14. We can see clearly from the table that all the algorithms have found the same S_{opt} except for the different cost of computation time. We can draw the same conclusion with the negative feedback circuit example that our new proposed algorithms have higher efficiency, and there are not great differences among our new proposed three algorithms (algorithm 1, algorithm 2 and algorithm 3) in finding the final solution.

In fact, no matter what kinds of faults (such as component faults, module faults, etc.) need to be diagnosed, if only the integer-coded fault dictionary and the fault-isolated table can be constructed, our proposed method can be used to choose the near minimum set of test points.

Table 10 Extended fault-isolated table for Fig. 2

Faults	n_1		n_2		n_3		n_4		n_5		n_6		n_7		n_8		n_9		n_{10}		NI_i
	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	
	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}	t_{20}	
f_1(NOM)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_2(R1 open)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
f_3(R2 open)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	4
f_4(R3 open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_5(R4 open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_6(R5 open)	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2
f_7(R6 open)	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2
f_8(R7 open)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	4
f_9(R8 open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
f_{10}(R9 open)	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	3
f_{11}(R10 open)	0	0	0	1	0	0	0	0	0	1	1	0	0	1	0	1	0	1	0	1	7
f_{12}(R1 short)	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
f_{13}(R3 short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{14}(R4 short)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	4
f_{15}(R5 short)	0	0	0	1	0	0	0	0	0	1	0	0	1	1	1	1	0	1	0	1	8
f_{16}(R6 short)	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	2
f_{17}(R7 short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{18}(R8 short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{19}(R9 short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{20}(R11 short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{21}(C2 open)	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
f_{22}(C3 open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{23}(C4 open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{24}(C5 open)	0	0	0	1	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	7
f_{25}(C6 open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{26}(C2 short)	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
f_{27}(C4 short)	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	3
f_{28}(C6 short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
f_{29}(Q1B open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{30}(Q1C open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{31}(Q1BC short)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	4
f_{32}(Q1CE short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{33}(Q2B open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{34}(Q2BC short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{35}(Q2CE short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 11 Results of different algorithms for Fig. 2

Algorithms	S_{opt}	Number of optimum test points	Elapsed time(s)
Starzyk's [12]	t_1, t_3, t_4, t_5, t_7, t_9, t_{14}, t_{15}, t_{18}, t_{20}	10	0.138
Yang's [15]	t_1, t_3, t_4, t_5, t_7, t_9, t_{13}, t_{18}, t_{20}	9	0.133
Pinjala's [8]	t_1, t_3, t_4, t_5, t_7, t_9, t_{11}, t_{13}, t_{18}, t_{20}	10	0.106
Exhaustive	t_1, t_3, t_4, t_5, t_7, t_9, t_{13}, t_{18}, t_{20}	9	74.287
New algorithm 1	t_1, t_3, t_4, t_5, t_9, t_{13}, t_{18}, t_{19}, t_{20}	9	0.085
New algorithm 2	t_1, t_3, t_4, t_5, t_9, t_{13}, t_{18}, t_{19}, t_{20}	9	0.084
New algorithm 3	t_1, t_3, t_4, t_5, t_9, t_{13}, t_{18}, t_{19}, t_{20}	9	0.090

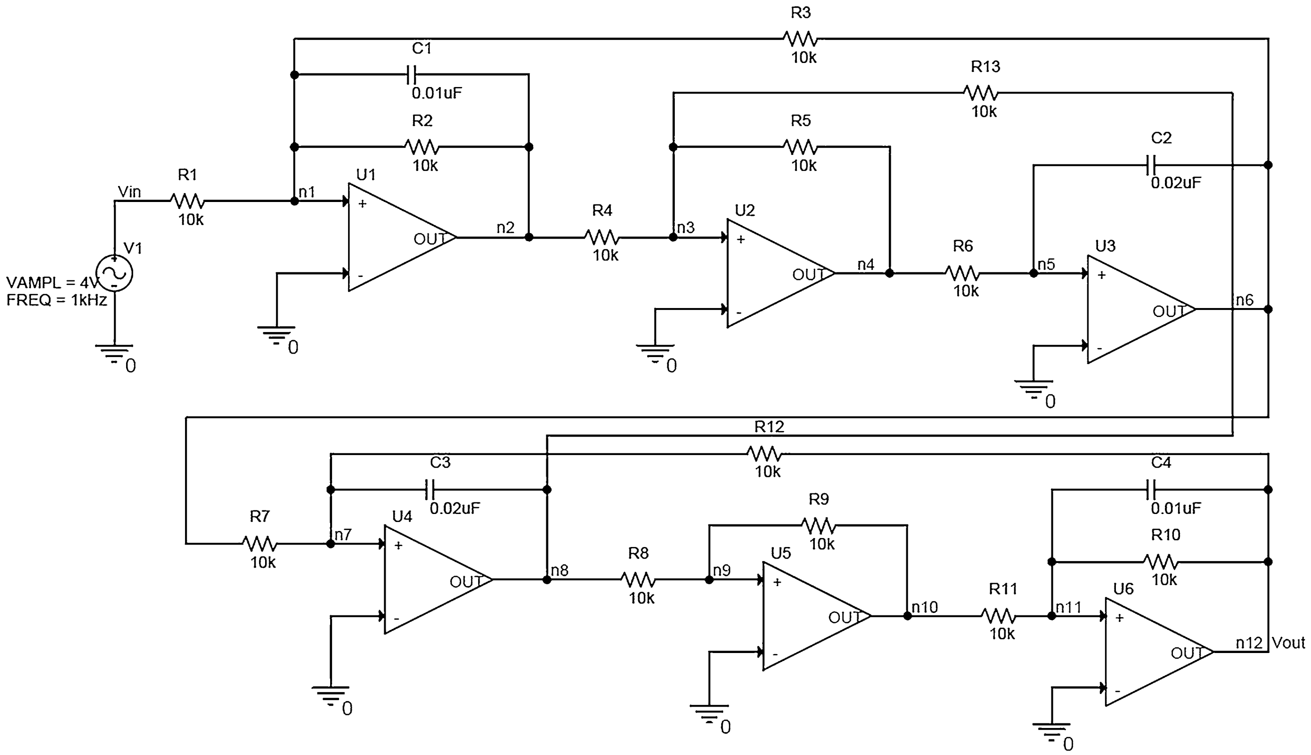

Fig. 3 Leapfrog filter

Table 12 Integer-coded fault dictionary for Fig. 3

Faults	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9	n_{10}	n_{11}	n_{12}
f_1(NOM)	4	2	2	3	1	5	3	3	2	3	3	3
f_2(C2 open)	8	7	6	7	4	9	7	9	4	6	4	7
f_3(C2 short)	4	3	2	3	1	3	3	2	2	2	3	2
f_4(C3 open)	1	0	0	0	0	1	1	1	1	1	1	1
f_5(R1 open)	9	8	6	7	4	10	7	9	4	6	4	7
f_6(R1 short)	5	5	4	6	4	8	7	8	4	6	4	7
f_7(R2 open)	6	7	6	6	4	9	7	8	4	6	4	7
f_8(R2 short)	4	1	2	1	1	3	3	2	2	2	3	2
f_9(R3 open)	2	0	0	0	0	0	0	0	0	0	0	0
f_{10}(R3 short)	0	0	0	0	0	1	1	0	0	0	0	0
f_{11}(R4 open)	4	3	2	1	1	3	3	2	2	2	3	2
f_{12}(R7 short)	1	0	0	0	0	1	1	0	0	0	0	0
f_{13}(R8 open)	8	7	5	6	4	9	6	6	4	6	4	6
f_{14}(R10 open)	4	4	2	2	1	4	3	3	2	3	3	5
f_{15}(R11 open)	7	6	3	5	3	7	4	7	4	6	4	7
f_{16}(R12 short)	1	0	0	0	0	1	0	0	0	0	0	0
f_{17}(R13 open)	4	4	2	4	1	6	3	4	2	4	3	4
f_{18}(U1 fail)	3	0	0	0	0	1	1	0	0	0	0	0
f_{19}(U2 fail)	1	0	1	0	0	1	1	1	1	1	1	1
f_{20}(U3 fail)	8	7	6	6	2	9	7	8	4	6	4	7
f_{21}(U4 fail)	1	0	0	0	0	1	2	1	1	1	1	1
f_{22}(U5 fail)	8	7	6	7	4	9	7	7	3	5	4	7
f_{23}(U6 fail)	1	0	0	0	0	1	1	1	1	1	2	1

Table 13 Extended fault-isolated table For Fig. 3

Faults	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9	n_{10}	n_{11}	n_{12}	NI_i
f_1(NOM)	0	1	0	0	0	1	0	0	0	0	0	1	3
f_2(C2 open)	0	0	0	0	0	0	0	0	0	0	0	0	0
f_3(C2 short)	0	0	0	0	0	0	0	0	0	0	0	0	0
f_4(C3 open)	0	0	0	0	0	0	0	0	0	0	0	0	0
f_5(R1 open)	1	1	0	0	0	1	0	0	0	0	0	0	3
f_6(R1 short)	1	1	1	0	0	1	0	0	0	0	0	0	4
f_7(R2 open)	1	0	0	0	0	0	0	0	0	0	0	0	1
f_8(R2 short)	0	1	0	0	0	0	0	0	0	0	0	0	1
f_9(R3 open)	1	0	0	0	0	1	0	0	0	0	0	0	2
f_{10}(R3 short)	1	0	0	0	0	0	0	0	0	0	0	0	1
f_{11}(R4 open)	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{12}(R7 short)	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{13}(R8 open)	0	0	1	0	0	0	1	1	0	0	0	1	4
f_{14}(R10 open)	0	0	0	1	0	1	0	0	0	0	0	1	3
f_{15}(R11 open)	1	1	1	1	1	1	1	0	0	0	0	0	7
f_{16}(R12 short)	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{17}(R13 open)	0	0	0	1	0	1	0	1	0	1	0	1	5
f_{18}(U1 fail)	1	0	0	0	0	0	0	0	0	0	0	0	1
f_{19}(U2 fail)	0	0	1	0	0	0	0	0	0	0	0	0	1
f_{20}(U3 fail)	0	0	0	0	1	0	0	0	0	0	0	0	1
f_{21}(U4 fail)	0	0	0	0	0	0	1	0	0	0	0	0	1
f_{22}(U5 fail)	0	0	0	0	0	0	0	0	1	1	0	0	2
f_{23}(U6 fail)	0	0	0	0	0	0	0	0	0	0	1	0	1

Table 14 Results of different algorithms for Fig. 3

Algorithms	S_{opt}	Number of optimum test points	Elapsed time(s)
Starzyk's [12]	$n_1, n_2, n_3, n_4, n_7, n_8, n_{11}$	7	0.093
Yang's [15]	$n_1, n_2, n_3, n_4, n_7, n_8, n_{11}$	7	0.083
Pinjala's [8]	$n_1, n_2, n_3, n_4, n_7, n_8, n_{11}$	7	0.084
Exhaustive	$n_1, n_2, n_3, n_4, n_7, n_8, n_{11}$	7	0.649
New algorithm 1	$n_1, n_2, n_3, n_4, n_7, n_8, n_{11}$	7	0.070
New algorithm 2	$n_1, n_2, n_3, n_4, n_7, n_8, n_{11}$	7	0.074
New algorithm 3	$n_1, n_2, n_3, n_4, n_7, n_8, n_{11}$	7	0.073

4.2 Statistical experiments

Although the above three experiments have shown the great advantage of the proposed algorithm to find the near minimum test point set, there still no theoretical proof can be offered to demonstrate a specific non-exhaustive algorithm's optimality [8, 12, 15], the new proposed algorithm must statistically be tested on larger number of fault dictionaries to demonstrate its efficiency and qualities of generating optimum test point sets.

Such statistical experiments were carried out on the randomly computer-generated integer-coded fault dictionaries by using the proposed algorithm, Starzyk's algorithm [12], Yang's algorithm [15], and Pinjala's algorithm [8] respectively. For these simulation dictionaries, the optimal solutions via exhaustive search method are infeasible. Therefore, to check the degree of suboptimality is meaningless. However, the comparison between different algorithms can also show the advantages and disadvantages of the methods. All the algorithms are programmed by MATLAB and tested on an Intel 2.2 GHz processor computer. Totally, there are 100 randomly computer-generated integer-coded fault dictionaries, and every dictionary includes 1,000 simulated faults, 40 test

Table 15 Statistical results of the solution accuracy and computation time per dictionary

Size of min. set	Starzyk's [12]	Yang's [15]	Pinjala's [8]	New algorithm 1	New algorithm 2	New algorithm 3
7	51	42	0	52	54	56
8	49	58	19	48	46	44
9	0	0	75	0	0	0
10	0	0	6	0	0	0
Computation time per dictionary (s)	2.094	1.637	0.173	0.120	0.280	0.381

points and randomly five to seven ambiguity sets per test point. The obtained statistical results concerning the solution accuracy and the average computation time per dictionary are shown in Table 15.

The conclusion from Table 15 is that the proposed method can find the final solution more quickly and effectively. Obviously, three new algorithms find all the optimum test-point sets whose size are 7 or 8, and this result is nearly the same as Starzyk's and Yang's algorithm. Meanwhile, our new algorithms cost less computation time than Starzyk's and Yang's algorithm. For the Pinjala's algorithm, although it takes shorter time to finish one run on each dictionary, the accuracy is the worst of all the algorithms, since it finds 0 set of test points whose size is 7 and 75 % of its results have a larger size by 1 than the other algorithms' solution. Although new algorithm 1 costs less computation time than algorithm 3, algorithm 3 finds 4 more minimum test point sets whose size is 7. Overall, there are not dramatic differences among the three new algorithms. In short, our new proposed method has a better tradeoff between the solution accuracy and the computational cost, and it is superior to other methods in its computational efficiency and quality of final solution. Therefore, the significance of the new proposed algorithm is that it offers an effective test-point selection method for medium and large scale systems within a reasonable computational cost.

5 Conclusion

As the fast development of modern electronic industry, more and more medium and large scale circuits with hundreds of components and modules are widely used in engineering, and this brings great difficulties and challenges to the fault diagnosis. How to find a minimum set of test points efficiently to isolate all the faults to a desired degree, therefore, becomes the key point. The global minimum solution is only guaranteed by exhaustive search method, which is proved to be NP-hard and impractical for medium or large circuit systems as its expensive computation time cost. Hence the near minimum set of test points is the best choice to meet the requirements of efficiency and practice. Based on the integer-coded fault dictionary, an efficient method to find a near minimum test-point set is proposed in this paper. This algorithm contains two main parts, the inclusive part and exclusive part. First of all, a new fault-isolated table is constructed and extended to select the special test points with particular fault isolate abilities. And then, judge the fault isolation ability of the set composed of all the special test points. If these special test points can isolate all the listed faults, exit this inclusive part and directly go to the exclusive part, else, use one of the inclusive criteria to choose more test points until they can isolate all the listed faults together. In the exclusive part, we use the advantage of exclusive technique to delete the redundant test points in case of existing redundancy. Under the help of fault-isolated table, we can find the special test points quickly and some times even find the final solution directly. This is why our proposed algorithm can always find the optimum test-point set at a shorter computation time. The time complexity of the proposed algorithm is proved to be less than $O(N_f m N_T \log N_f)$. Carried out on the same trademark analog circuits, it shows better computational efficiency and quality of finding the final results than the other methods. Since no theoretical proof can be given to the proposed algorithm, statistical experiments are utilized for its evaluation. The results demonstrate that our proposed method has a better tradeoff between the solution accuracy and the computational cost, and it is superior to other methods in its computational efficiency and quality of final solution. Therefore, it is a good solution to minimize the size of the test-point set, and practical for medium and large scale systems.

References

1. Augusto, J. S. & Almeida, C. (2006). A tool for test point selection and single fault diagnosis in linear analog circuits. In *Proceeding of the XXI International Conference on Design of Systems and Integrated Systems*, Barcelona, Spain.
2. Bandler, J. W., & Salama, A. E. (1981). Fault diagnosis of analog circuits. *Proceedings of the IEEE, 73*, 1279–1325.
3. Bandler, J. W., & Salama, A. E. (1985). Fault diagnosis of analog circuits. *Proceedings of the IEEE, 73*(8), 1279–1325.

4. Duhamal, P., & Rault, J. C. (1979). Automatic tests generation techniques for analog circuits and systems: A review. *IEEE Transactions on Circuits and Systems I, CAS-26*, 440–441.
5. Golonek, T., & Rutkowski, J. (2007). Genetic-algorithm-based method for optimal analog test points selection. *IEEE Transactions on Circuits and Systems II: Express Briefs, 54*(2), 117–121.
6. Hochwald, W., & Bastian, J. D. (1979). A dc approach for analog fault dictionary determination. *IEEE Transactions on Circuits and Systems, CAS-26*, 523–529.
7. Lin, P. M., & Elcherif, Y. S. (1985). Analogue circuits fault dictionary—New approaches and implementation. *The International Journal of Circuit Theory and Applications, 13*(2), 149–172.
8. Pinjala, K. K. & Bruce, C. K. (2003). An approach for selection of test points for analog fault diagnosis. In *Proceedings of the 18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems* (pp. 287–294).
9. Prasad, V. C., & Babu, N. S. C. (2000). Selection of test nodes for analog fault diagnosis in dictionary approach. *The IEEE Transactions on Instrumentation and Measurement, 49*(6), 1289–1297.
10. Skowron, A., & Stepaniuk, J. (1991). Toward an approximation theory of discrete problems: Part I. *Fundamenta Informaticae, 15*(2), 187–208.
11. Spaandonk, J. & Kevenaar, T. (1996). Iterative test point selection for analog circuits. In *Proceedings of 14th on VLSI Test Symposium* (pp. 66–71). Princeton, USA.
12. Starzyk, Janusz A., Liu, Dong, Liu, Zhi-Hong, Nelson, Dale E., & Rutkowski, Jerzy O. (2004). Entropy-based optimum test nodes selection for analog fault dictionary techniques. *The IEEE Transactions on Instrumentation and Measurement, 53*, 754–761.
13. Stenbakken, G. N., & Souders, T. M. (1987). Test point selection and testability measure via QR factorization of linear models. *The IEEE Transactions on Instrumentation and Measurement, IM-36*(6), 406–410.
14. Varghese, X., Williams, J. H., & Towill, D. R. (1978). Computer aided feature selection for enhanced analogue system fault location. *Pattern Recognition, 10*(4), 265–280.
15. Yang, C. L., Tian, S. L., & Long, B. (2009). Application of heuristic graph search to test point selection for analog fault dictionary techniques. *The IEEE Transactions on Instrumentation and Measurement, 58*(7), 2145–2158.
16. Yang, C. L., Tian, S. L., & Long, B. (2009). Test points selection for analog fault dictionary techniques. *The Journal of Electronic Testing, 25*(2–3), 157–168.
17. Yang, C. L., Tian, S. L., & Long, B. (2010). A test points selection method for analog fault dictionary techniques. *Analog Integrated Circuits and Signal Processing, 63*(2), 349–357.

Dongsheng Zhao was born in 1981. He received the B.S. degree in electronic science and technology from Beijing Jiaotong University, Beijing, China, in 2006 and the M.S. degree in precision instruments and machinery from Xi'an Shiyou University, Xi'an, China, in 2012. He is currently working toward the Ph.D. degree at the School of Instrumentation Science and Opto-electronics Engineering, Beihang University, Beijing, China. His research interests include analog circuit test and measurement, electronic testing and fault diagnosis of complicated electronic systems, signal processing, artificial intelligence and pattern recognition.

Yuzhu He was born in 1955. He received the B.S. degree from Beihang University, Beijing, China, in 1994 and the Ph.D. degree from City University of Hong Kong, Hong Kong, China, in 2004. He is currently a professor and Ph.D. supervisor in School of Instrumentation Science and Opto-electronics Engineering, Beihang University. His teaching and researching interests include modern electronic system testing and testing technology, universal test system integration, electronic system fault diagnosis and location technology.