UDC 621.007.52

V.A. Mashkov, V.I. Lytvynenko

PROBLEM WITH DEVELOPING ALGORITHMS FOR SYSTEM LEVEL SELF-DIAGNOSIS

Abstract: *Article is devoted to sequence, structure and strategy of solving the problem of self diagnosis of homogeneous systems. Considered in detail the problem of the development of algorithms for system self test level. The conditions and requirements to obtain the correct diagnosis.*

Keywords: *self diagnosis, homogeneous systems, diagnosis algorithms, sequential diagnosis, excess diagnosis, intermittent faults, hybrid faulty situation*

1. Introduction

The paper tackles the system level self-diagnosis which was introduced by Preparata [1] and then has been broadly investigated in literature. Before providing self-diagnosis, it is necessary to perform several steps, such as to choose the diagnosis strategy (i.e., unique diagnosis, sequential diagnosis or excess diagnosis); to make decision about allowable faulty situations (i.e., only permanent faults, only intermittent faults or hybrid faulty situation); to make decision on how to interpret test results. Depending on the decision made at each step, different diagnosis algorithms can be developed. In the paper, some problems of developing of diagnosis algorithms for system level self-diagnosis are discussed for the case of uniquely diagnosable systems.

2. Algorithms for uniquely diagnosable systems

There have been developed many algorithms allowing to identify a fault set uniquely (when some assumptions are made about the faulty units). The main task while developing these algorithms is to reduce their complexity. Among the most efficient algorithms there could be named algorithm proposed by Dahbura and Masson [2] which has $O(N^{2.5})$ time complexity and $O(t^3 + |E|)$ algorithm suggested by Sullivan [3]. Both algorithms were developed for t-diagnosable systems under the symmetric invalidation model and when permanent fault are allowable only. There are many special classes of t-diagnosable systems that support more efficient diagnosis techniques than above mentioned ones, and this is reason to believe that an $O(|E|)$ diagnosis solution exists for all t-diagnosable systems. Preparata et al. defined the $D_{\delta,t}$ structure in which unit $u_i$tests u_j if and only if

© Mashkov V., Lytvynenko V., 2015

 ISSN 1562-9945

$$j - i = \delta m \pmod{n},$$

where $m=1,2,...,t$.

Meyer and Masson [4] gave $O(nt)$ solution to the case of $\delta=1$.

The efforts to develop the simplest algorithm are continue till now.

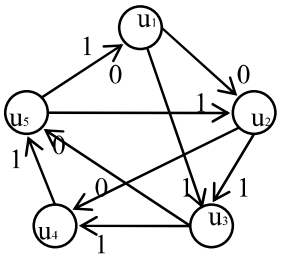
For developing the table algorithm there have been made the same assumptions as those made for the above mentioned algorithms. Underlying strategy before developing this algorithm is to structure the algorithm so that it can be split into several branches each of which is characterised by its own complexity and its own probability of their using while executing system diagnosis. It is expected that the simplest branch will be executed most of the time.

The algorithm deals with table of syndrome which is the matrix presentation of syndrome. Table of syndrome $M_R[r_{ij}]$ is square matrix of dimension NЧN, where N is the total number of units in the system.

If the result of the test which is performed by unit ui on unit uj is the element of syndrome, then the value of r_{ij}will be placed on the intersection of i-th row and j-th column of the table.

Since test result can take values either 0 or 1 only, a table of syndrome also contains only these values. If there is no test between two units (i.e., there is no such element in the syndrome) then dash will be placed on the corresponding intersection. To present the table of syndrome for computer modelling, it is possible to substitute the dash with the value of "-1".

As an example of table of syndrome, in the Figure 1, the testing graph with weights of edges and the table of syndrome corresponding to this graph are depicted.



$M_R =$

	1	2	3	4	5
1	-	0	1	-	-
2	-	-	1	0	-
3	-	-	-	1	0
4	0	-	-	-	1
5	1	1	-	-	-

Figure 1. Testing graph with weighted edges and table of syndrome M_R.

The goal of the diagnosis algorithm based on matrix $M_R$is to identify all faulty units (on the condition that the total number of faulty units

does not exceed the value t which is defined by the testing graph). Otherwise, the algorithm with a great probability should be able to find out that total number of faulty units has exceeded the value t.

As a preliminary step for developing the algorithm, the value t is determined by examining the testing graph.

Generally, it can be obtained arbitrary testing graph, but mostly tests have being performed until certain graph is formed. Usually, such graphs insure desired value of t which at the same time is equal to t_{max}.

Having the value of t and matrix M_R, the following steps should be performed:

Step 1. For each row and column of matrix M_R the total number of "1" is counted. The total number of "1" in the row $\boldsymbol{x}_i$ is equal to

$$S_{x_i} = \sum_{j=1}^{n} r_{ij}$$

The total number of "1" in the column $\boldsymbol{y}_j$is equal to

$$S_{y_j} = \sum_{i=1}^{n} r_{ij}$$

Step 2. For each $i=1,2,..,n$ the following sum is calculated

$$S_i = S_{xi} + S_{yi}$$

As a result, there will be obtained the tuple $(S_1, S_2,..,S_n)$.

Step 3. Successively each value S_i , $i=1,2,..,n$ is compared with the value of t. As a result of comparing there are possible three situations:

A) $S_i > t$

B) $S_i = t$

C) $S_i < t$

The next step depends on the arising situation. Thus, after performing of *Step 3* there possible three branches: A, B and C.

Branch A.In this case, it is easy to show that unit $u_i$is faulty. For this, let's assume that unit $u_i$is fault-free. Then, all of the units that evaluate the unit $u_i$as faulty must be faulty since they have tested unit $u_i$incorrectly. Consequently, the total number of faulty units in the system will be greater than t, which contradicts the main assumption about *t-diagnosability* of the system. Thus, our assumption about the state of unit u_i was wrong, and unit $u_i$must be identified as faulty.

 ISSN 1562-9945

Branch B.In this case, given value S_i, it is not possible to conclude directly whether unit u_i is faulty or not. Firstly, it is assumed that unit u_i is fault-free. Given such assumption, all of the units which either evaluate unit u_i as faulty or those that are evaluated as faulty by unit u_i must be identified as faulty.

Thus, unit u_j is marked as faulty if

$$(r_{ij} = 1) \vee (r_{ji} = 1).$$

On the contrary, units that are evaluated by unit u_i as fault-free (i.e., $r_{ij}=0$) must be identified and marked as fault-free. The set of such units is denoted as Z_i.

Next, the units from the set $Z_i$are successively selected, and the whole procedure described above for unit u_i is repeated, but this time for the selected unit.

This way there can be selected further units, and the procedure will continue until either there is found a contradiction in marking of a unit, or all of the units which are assumed as fault-free have been inspected.

In the former case, when a contradiction is found, unit u_i should be identified as faulty. In the latter case, the initial assumption about the state of unit $u_i$is correct.

Branch C.If this situation arises, the next operations on matrix $M_R$depend considerably on the number of system units and on the system testing assignment.

For $N<13$ it is sufficient to find in matrix M_R the column which contains only zero elements. This column corresponds to the unit which should be identified as fault-free. Using tests results related to this unit, it is possible to make decision about the states of all the remaining system units. This can be done by using the procedure which is similar to the one described for branch *B*.

For $N\geq13$ it is possible to prove that the testing graph contains the simple directed cycle of length $t+1$ each edge of which has the weight of "0". Such cycle can be determined directly from the matrix M_R. All of the units which are represented in the testing graph by the vertices of this cycle can be identified as fault-free.

It should be noted that the probabilities of the event that the situations *A*, *B* or C will arise depend considerably on the number of

system units, on the structure of testing graph, and on the interpretation of test result (see Table 1).

Table 1.

Test results and their probabilities

Test resultandits probability		Testing unit u_i	
		fault-free	faulty
Testedunit u_j	fault-free	$r_{ij} = 0$ (P_C)	$r_{ij} = 0$ $(1 - P_S)$
		$r_{ij} = 1$ $(1 - P_C)$	$r_{ij} = 1$ (P_S)
	faulty	$r_{ij} = 0$ $(1 - P_{AT})$	$r_{ij} = 0$ $(1 - P_F)$
		$r_{ij} = 1$ (P_{AT})	$r_{ij} = 1$ (P_F)

So, for example, for the testing graph depicted in Fig. 1 there are possible only situations *A* and *B*. For this example with $t=2$ and testing graph $D_{1,2}$of type $D_{\delta,t}$, it is easy to prove that S_i cannot be lesser than t.

Proof.Since $t=2$, there are possible only two cases. In the first case, only one unit is faulty, and in the second case, two units are faulty.

Case 1. In the system with $D_{1,2}$graph, each unit is tested by two other system units. For this case, both units which test the faulty unit are fault-free and will produce the test result equal to 1. Thus, the total number of 1 will be equal to two (i.e., not lesser than t).

Case 2. In this case, there are possible two situations. The first situation arises when faulty unit is not tested by another faulty unit. The second situation arises when faulty unit is tested by another faulty unit. The first situation is identical to the Case 1 and leads to the same result. In the second situation, there can be use the property of structure $D_{\delta,t}$ such that there are no units which test each other. It means that if one faulty unit tests another faulty unit, then this faulty testing unit is not tested by faulty unit. Thus, this faulty testing unit is tested by two fault-free units with the test result equal to 1. It means that the total number of 1 will be equal to two (i.e., not lesser than t).

Now, let's take into consideration the fact that faulty testing unit can produce the test result equal to 1 with some probability (i.e., probabilities P_F and P_S are not equal to zero), and estimate the probability of the event that the situation *A* will arise, P_a. For the system with $D_{1,2}$ testing graph the probability $P_a$can be estimated as follows.

 ISSN 1562-9945

It is evident that the more faulty units in the system, the greater is probability P_a. That is why we consider only the "worst" case when only one faulty unit is in the system. This faulty unit will perform two tests on other system units. If we assume that $P_F = P_S = 0.5$ then $P_a = 0.75$. The greater the values of P_F and P_S the greater the probability P_a. As one of the solutions of how to increase the probability P_a, there can be suggested to repeat the tests two or more times. So, for example, if the tests will be repeated twice the probability P_a will increase to 0,9375.

4. Algorithm based on the table of potential syndromes

Vedeshenkov in [5] suggested an original algorithm based on the table of potential syndromes. Similarly to the table algorithm, the execution of this algorithm depends considerably on the obtained syndrome and may be very simple for most cases. The table of potential syndromes is formed before the testing procedure begins.

The table contains all possible syndromes that can be obtained when the system has different combinations of faulty units. The total number of faulty units is restricted by value of t. Consequently, only combinations that contain not more than t faulty units are presented in the table.

The number of combinations of faulty units, Q, is equal to

$$Q = \sum_{i=1}^{t} \binom{n}{i}$$

For example, for the system with testing graph shown in Fig.1 there should be considered the following fifteen combinations:

S_1: unit u_1 is faulty
S_2: unit u_2 is faulty
S_3: unit u_3 is faulty
S_4: unit $u_4$is faulty
S_5: unit u_5 is faulty
S_6: units u_1 and u_2 are faulty
S_7: units u_1 and u_3 are faulty
S_8: units u_1 and u_4 are faulty
S_9: units u_1 and $u_5$are faulty
S_{10}: units u_2 and u_3 are faulty
S_{11}: units $u_2$and $u_4$are faulty
S_{12}: units u_2 and u_5 are faulty
S_{13}: units u_3 and u_4 are faulty
S_{14}: units u_3 and u_5 are faulty
S_{15}: units u_4 and $u_5$are faulty

If actual state of the system is so that it has more than *t* faulty units, then result of diagnosis based on the table of potential syndromes may be either incorrect (false negative) or zero. Zero result means that it is not possible to make any solutions about the faulty situation in the system.

A table of potential syndromes has the following columns:

- denotations of the situations (S_i)
- the set of faulty units (e.g., $\{u_1, u_5\}$)
- denotations of the potential syndromes (R_p^i) that correspond to the situations S_i, $i=1,2,...Q$
- separate elements (r_{ij}) of the syndrome (R_p^i). These elements construct l columns of the table, where l is the number of edges (resp. tests) in testing graph.

The table of potential syndromes contains as many rows as there are situations. The values of separate elements of potential syndrome (r_{ij}) are set according to the model of test results suggested by Preparata [1]. It means that a test result can be represented in the table as 0 or 1 when testing unit is fault-free or as X when testing unit is faulty. Actual syndrome as distinct to the potential syndrome contains only values of 0 or 1.

For the example under consideration the table of potential syndrome looks like as follows

Table 2

Table of potential syndromes

S_i	*faulty units*	*potential syndrome*	*test results*									
			r_{12}	r_{12}	r_{12}	r_{12}	r_{12}	r_{12}	r_{12}	r_{12}	r_{12}	r_{12}
S_1	u_1	R_p^1	X	X	0	0	0	0	0	1	1	0
S_2	u_1	R_p^2	1	0	X	X	0	0	0	0	0	1
S_3	u_1	R_p^3	0	1	1	0	X	X	0	0	0	0
S_4	u_1	R_p^4	0	0	0	1	1	0	X	X	0	0
S_5	u_1	R_p^5	0	0	0	0	0	1	1	0	X	X
S_6	u_1 , u_2	R_p^6	X	X	X	X	0	0	0	1	1	1
S_7	u_1 , u_3	R_p^7	X	X	1	0	X	X	0	1	1	0
S_8	u_1 , u_4	R_p^8	X	X	0	1	1	0	X	X	1	0
S_9	u_1 , u_5	R_p^9	X	X	0	0	0	1	1	1	X	X
S_{10}	u_2 , u_3	R_p^{10}	1	1	X	X	X	X	0	0	0	1
S_{11}	u_2 , u_4	R_p^{11}	1	0	X	X	1	0	X	X	0	1
S_{12}	u_2 , u_5	R_p^{12}	1	0	X	X	0	1	1	0	X	X
S_{13}	u_3 , u_4	R_p^{13}	0	1	1	1	X	X	X	X	0	0
S_{14}	u_3 , u_5	R_p^{14}	0	1	1	0	X	X	1	0	X	X
S_{15}	u_4 , u_5	R_p^{15}	0	0	0	1	1	1	X	X	X	X

 ISSN 1562-9945

From the table it is easy to infer that any two potential syndromes differ at least in one element. This difference plays the key role for diagnosis based on the table of potential syndromes.

Having performed all tests and obtained actual syndrome, one can apply diagnosis algorithm. Diagnosis algorithm consists in comparing an actual syndrome with the potential syndromes presented in the table. The goal of diagnosis algorithm is to find in the table the potential syndrome that coincides with actual syndrome. Comparing is performed so that each element of actual syndrome is compared with the corresponding element of the potential syndrome. Element of potential syndrome denoted as X is always complying with element of actual syndrome.

Comparing procedure can be executed according to different strategies. Basic and straightforward strategy consists in successive comparing the elements of actual syndrome with elements of separate potential syndromes depicted in separate rows of the table. Comparing starts from the first row and goes on until there is coincidence.

It is worth noting, that such strategy may be not efficient because the comparing procedure may require a large number of single comparing (up to $QЧl$ single comparing). Application of this basic strategy to the system under consideration gives the result that actual syndrome coincides with the potential syndrome R_p^{14}. It means that units $u_3$and $u_5$are identified as faulty. In this case, the actual syndrome was compared with 14 potential syndromes (i.e., there were 14Ч10 = 140 single comparing).

One can easily notice that total number of single comparing can be reduce if comparing procedure is organised so that only part of the syndrome is chosen in each step of comparing procedure.

So, for example, if it is chosen only first element of the syndrome for comparing, then only 11 potential syndromes remain for further consideration.

The algorithm developed according to this strategy consists in the following:

At the first step, only rows that contain the first element that coincides with the first element of actual syndrome are chosen. For the considered example these are the rows 1,3,4,5,6,7,8,9,13,14,15.

At the second step, the second element of just chosen rows is compared with the second element of actual syndrome. Only rows which

contain the second element that coincides with the second element of actual syndrome remain for further consideration. In the given case, these are the rows 1,3,6,7,8,9,13,14.

This procedure continues for the next elements and ends at the seven element of actual syndrome when only one potential syndrome remains, particularly R_p^{14}.

It is worth noting, that there can be suggested other strategies for organising the comparing of actual syndrome with potential syndromes which could allow to reduce the total number of single comparing.

Both considered above algorithms use the table as the starting point for identifying the set of faulty units. These algorithms have advantages and drawbacks with regards to their development and application.

Advantages:

- their development requires only basic information about the system (such as, total number of system units, system testing assignment and the results of testing)
- the tables could be very easy developed. They are illustrative (telling) which may reduce mistakes while their processing.

Drawback:

- these algorithms don't take into account the reliability of system units which reduces the credibility of result of diagnosis.

5. Conclusions

Unique diagnosis considered in the paper is based on the statement that the made assumptions about allowable faulty sets will be satisfied. Thus, the diagnosis algorithms developed for the unique diagnosis will provide correct diagnosis only in the case when the made assumptions are correct. In practice, we can find much evidence of the fact that making such assumptions is reasonable. It is also worth noting that the algorithms considered in the paper are applicable only for homogeneous systems.

REFERENCES

1. Preparata, T.; Metze, G.; Chien, R. On the connection assignment problem of diagnosable system. IEEE Transactions on Electronic Computers. Vol.EC-16, No.12, 1967. pp. 848-854.
2. Dahbura, A.; Masson, G. An O(n2.5) fault identification algorithm for diagnosable systems. IEEE Trans. Comput., Vol.C-33, 1984. pp.486-492.

3. Sullivan, G. An O(t3 + |E) fault identification algorithm for diagnosable systems. IEEE Trans. Comput., Vol.C-37, 1988. pp.388-397.
4. Meyer, G.; Masson, G. An efficient fault diagnosis algorithm for symmetric multiple processor architectures. IEEE Trans. Comput., Vol.C-27, 1978. pp.1059-1063.
5. Vedeshenkov, V. On organization of self-diagnosable digital systems. Automation and Computer Engineering. Vol.7, 1983. pp.133-137.