International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:9, No:5, 2015

A Comprehensive Method of Fault Detection and Isolation Based On Testability Modeling Data

Junyou Shi, Weiwei Cui

***Abstract*—**Testability modeling is a commonly used method in testability design and analysis of system. A dependency matrix will be obtained from testability modeling, and we will give a quantitative evaluation about fault detection and isolation.

Based on the dependency matrix, we can obtain the diagnosis tree. The tree provides the procedures of the fault detection and isolation. But the dependency matrix usually includes built-in test (BIT) and manual test in fact. BIT runs the test automatically and is not limited by the procedures. The method above cannot give a more efficient diagnosis and use the advantages of the BIT.

A Comprehensive method of fault detection and isolation is proposed. This method combines the advantages of the BIT and Manual test by splitting the matrix. The result of the case study shows that the method is effective.

***Keywords*—**BIT, fault detection, fault isolation, testability modeling.

I. Introduction

TESTABILITY modeling is commonly used method in the field of fault detection and isolation [1]-[5]. The key math element of the testability modeling is the dependency matrix. The matrix describes the relationship between the fault modes and tests. The diagnosis tree will be obtained based on the dependency matrix. The tree is a binary tree, which included the concrete diagnosis procedures [2]-{[]. A concrete diagnosis result will be given based on the tree.

In fact, the matrix usually contains the BIT and manual test. The BIT is a more advanced tool to detect and isolate the faults, which runs automatically. Therefore, the traditional method is not suitable when associated with BIT. That means the dependency matrix should be split into different kinds, and different diagnosis methods are given.

A comprehensive method is proposed. The key point of this new method is processing different dependency matrixes by splitting the original dependency matrix. The new method contains a series of steps, and every step contains very clear operations. A case study shows that this method can give a more effective and reasonable diagnosis result.

II. Principle of the Comprehensive Method

The comprehensive method aims to combine the advantages of the BIT and manual test, and give a more effective diagnosis result. The main operations of this method contain five steps, and they are shown as follows.

A. Constructing the Dependency Matrix of System

The dependency matrix of the system is constructed based on the fault and test data. The form of the matrix is shown as (1).

$$\begin{array}{c} \\ F_1 \\ \vdots \\ D = F_p \\ \vdots \\ F_m \end{array} \begin{array}{c} \begin{array}{ccc} BIT_1/M_1 & BIT_c/M_c & BIT_n/M_n \end{array} \\ \begin{bmatrix} d_{11} & \cdots & d_{1r} & \cdots & d_{1n} \\ \vdots & & & & \vdots \\ d_{p1} & & \ddots & & d_{pn} \\ \vdots & & & & \vdots \\ d_{m1} & \cdots & d_{mr} & \cdots & d_{mn} \end{bmatrix} \end{array} \quad (1)$$

In (1), D means dependency matrix, and this matrix is also named by the original dependency matrix; F={F_1,F_2,…F_m} means the fault set of the system; BIT_i means the BIT; M_j means the manual test; BIT_i /M_j means this kind of test is BIT or manual test; d_{ij} means the relation between the fault and test, and the concrete meaning of $d_{ij}$is shown in (2).

$$d_{ij} = \begin{cases} 1 & \textit{when } \mathrm{BIT}_i \text{ or } \mathrm{M}_i \text{ can detect the } \mathrm{F}_j \\ 0 & \textit{when } \mathrm{BIT}_i \text{ or } \mathrm{M}_i \text{ cannot detect the } \mathrm{F}_j \end{cases} \quad (2)$$

B. Extracting the Dependency Matrix about BIT, and Constructing the Corresponding the Fault Dictionary

The original dependency matrix contains two kinds of test. Extract the dependency matrix about BIT, and construct the corresponding fault dictionary. The fault dictionary can be regarded as a special and simplified dependency matrix. This step can be divided into two steps.

1) Extracting the Dependency Matrix about BIT

Extract the columns whose title contains BIT to construct a new matrix D_1. The remaining of the D is named by D_2. D_1 and D_2 are shown in (3) and (4).

$$\begin{array}{c} \\ F_1 \\ \vdots \\ D_1 = F_p \\ \vdots \\ F_m \end{array} \begin{array}{c} \begin{array}{ccc} BIT_1 & \cdots & BIT_s \end{array} \\ \begin{bmatrix} d_{11} & \cdots & d_{1s} \\ \vdots & & \vdots \\ d_{p1} & \ddots & d_{ps} \\ \vdots & & \vdots \\ d_{m1} & \cdots & d_{ms} \end{bmatrix} \end{array} \quad (3)$$

Junyou Shi is Associate Professor of the School of Reliability and System Engineering Beihang University, Beijing, 100191 China (corresponding author to provide phone: +86-010-82314704; e-mail: buaacvv@live.cn).

Weiwei Cui is master degree candidate of school of reliability and system engineering with the Beihang University, Beijing, 100191 China (e-mail: sjy@buaa.edu.cn).

$$D_2 = \begin{matrix} & M_1 & \cdots & M_q \\ F_1 & d_{1t} & \cdots & d_{1n} \\ \vdots & \vdots & & \vdots \\ F_p & d_{pt} & \ddots & d_{pn} \\ \vdots & \vdots & & \vdots \\ F_m & d_{mt} & \cdots & d_{mn} \end{matrix} \quad (4)$$

2) Constructing the Fault Dictionary

If a special row whose element is zero all in D_1, this kind of row is deleted from D_1, and its corresponding fault mode forms a new set named $F_1^{(0)}$. Then merge the same row in D_1, and a new matrix is obtained named $D_1^{(1)}$. $D_1^{(1)}$ is the fault dictionary of BIT.

C. Constructing the Dependency Matrix about Manual Test of the Fuzzy Fault Set of BIT, and Generating the Diagnosis Tree

This step is divided into two steps.

1) Getting the Fuzzy Fault Set of BIT

Based on the $D_1^{(1)}$, we can get a fuzzy set named $F^{(2)}$. $F^{(2)}$ contains several fuzzy fault sets, and $F^{(2)} = \{F_1^{(2)}, F_2^{(2)}, \cdots, F_k^{(2)}\}$. Each $F_i^{(2)}$ contains several faults. The rows in D_2 whose titles are included in $F_i^{(2)}$ are extracted and named by $D_i^{(2)}$.

2) Generating the Diagnosis Tree

According to the method introduced in [2-3], we can generate the diagnosis tree of every $D_i^{(2)}$. This tree is based on the manual test to diagnosis the BIT fuzzy fault set. The form of the diagnosis tree is shown in Fig. 1.

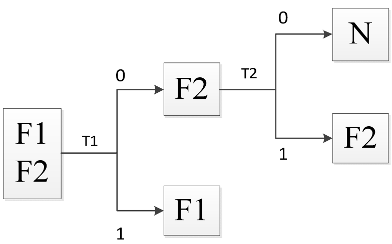

Fig. 1 Form of the diagnosis tree

In Fig. 1, N means the system has no fault. T_i means the test which could be BIT or manual test.

D. Constructing the Dependency Matrix of the Undetectable Faults of BIT, and Generating the Diagnosis Tree

This step is divided into two steps.

1) Constructing the Dependency Matrix of the Undetectable Faults

If the set $F_1^{(0)}$ is not empty, that means the BIT has the undetectable faults. Then we construct the dependency matrix about manual test of the undetectable faults of BIT. The rows in D_2 whose titles are included in $F_1^{(0)}$ are extracted and named by $D_2^{(1)}$.

2) Generating the Diagnosis Tree

According to the method introduced in [2], [3], we can generate the diagnosis tree of $D_2^{(1)}$. This tree is based on the manual test to diagnosis the BIT undetectable faults.

E. Synthesizing the Fault Dictionary, the Diagnosis Tree for Fuzzy Fault Set of BIT and the Diagnosis Tree for Undetectable Faults of BIT to Detect and Isolate Faults when System Checking and Maintaining

At first, we make sure whether the BIT gives the fault reports. If the BIT detects the fault, we can isolate the fault based on the fault dictionary. If the diagnosis result do not contain the fuzzy fault set, the fault detection and isolation is over. If the result contains the fuzzy fault set, we can isolate the fault based on the diagnosis tree of manual test for fuzzy fault of BIT, and the diagnosis may run for several times because of several diagnosis trees.

If the BIT does not give the fault report, that means system maybe has some faults which the BIT cannot detect. We can isolate the fault based on the diagnosis tree of manual test for undetectable fault of BIT.

III. Application

The graphic correlation model of a control system is shown in Fig. 2.

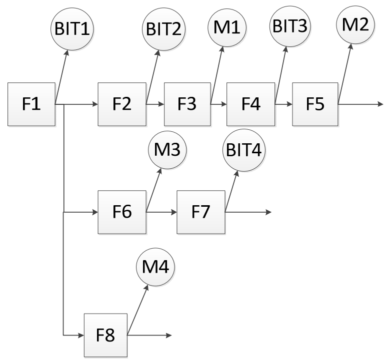

Fig. 2 Graphic correlation model of a control system

In Fig. 2, F_i means fault modes, and BIT/M means different tests. The concrete meanings of them are shown in Tables I and II.

Based on the graphic model, we can generate the corresponding dependency matrix (the original dependency matrix). If F_i can connect with T_j through the line, the d_{ij} in the dependency matrix is one. The dependency matrix is shown in Table III.

Based on the original dependency matrix, we can obtain D_1 and D_2. D_1 and D_2 are shown in Tables IV and V.

Delete the special row whose element is zero all in D_1. And the corresponding faults of the row form the set $F_1^{(0)}$.

$F_1^{(0)} = \{F_5, F_8\}$. Merge the same row and we get the $D_1^{(1)}$ (fault dictionary) as shown in Table VI.

TABLE I
THE CONCRETE MEANING OF F_i

Symbol	Meaning
F1	Fault with the signal acquisition unit
F2	Fault with intermediate frequency signal extraction unit
F3	Fault with calculating unit
F4	Fault with power amplifier unit
F5	Fault with actuator unit
F6	Fault with differential signal extraction unit
F7	Fault with characteristic signal output unit
F8	Fault with data acquisition recorder unit

TABLEII
THE CONCRETE MEANING OF TESTS

Symbol	Meaning
BIT1	Signal acquisition BIT
BIT2	Intermediate frequency signal extraction BIT
BIT3	Power amplifier BIT
BIT4	Characteristic signal BIT
M1	Calculating validity test
M2	Actuator range test
M3	Differential signal output manual test
M4	Data acquisition indicator light observation test

TABLEIII
THE ORIGINAL DEPENDENCY MATRIX

Test / Fault mode	BIT1	BIT2	M1	BIT3	M2	M3	BIT4	M4
F1	1	1	1	1	1	1	1	1
F2	0	1	1	1	1	0	0	0
F3	0	0	1	1	1	0	0	0
F4	0	0	0	1	1	0	0	0
F5	0	0	0	0	1	0	0	0
F6	0	0	0	0	0	1	1	0
F7	0	0	0	0	0	0	1	0
F8	0	0	0	0	0	0	0	1

TABLEIV
THE MATRIX D_1

	BIT1	BIT2	BIT3	BIT4
F1	1	1	1	1
F2	0	1	1	0
F3	0	0	1	0
F4	0	0	1	0
F5	0	0	0	0
F6	0	0	0	1
F7	0	0	0	1
F8	0	0	0	0

TABLEV
THE MATRIX D_2

	M1	M2	M3	M4
F1	1	1	1	1
F2	1	1	0	0
F3	1	1	0	0
F4	0	1	0	0
F5	0	1	0	0
F6	0	0	1	0
F7	0	0	0	0
F8	0	0	0	1

TABLEVI
THE MATRIX $D_1^{(1)}$

	BIT1	BIT2	BIT3	BIT4
F1	1	1	1	1
F2	0	1	1	0
F3/F4	0	0	1	0
F6/F7	0	0	0	1

We can get the fuzzy set $F^{(2)}$, and $F^{(2)}$ has two fuzzy fault sets. $F^{(2)} = \{F_1^{(2)}, F_2^{(2)}\}$. $F_1^{(2)} = \{F_3, F_4\}$, $F_2^{(2)} = \{F_6, F_7\}$. Construct two dependency matrix of manual test named by $D_1^{(2)}, D_2^{(2)}$ based on $F^{(2)}$. They are shown in Tables VII and VIII. According to the method introduced in [2], [3], we generate two diagnosis trees which are shown in Figs. 3 and 4.

TABLEVII
THE MATRIX $D_1^{(2)}$

	M1	M2	M3	M4
F3	1	1	0	0
F4	0	1	0	0

TABLEVIII
THE MATRIX $D_2^{(2)}$

	M1	M2	M3	M4
F3	0	0	1	0
F4	0	0	0	0

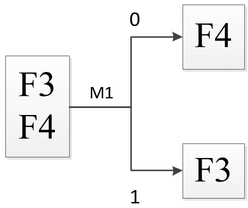

Fig. 3 The diagnosis tree based on $D_1^{(2)}$

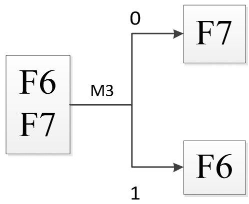

Fig. 4 The diagnosis tree based on $D_2^{(2)}$

In Fig. 3, we can use the M1 test to isolate the fault. If the test passes, we can isolate the F4. On the contrary, the isolation result is F3. Fig. 4 is the same with Fig. 3, and we can isolate the F6 or F7.

The undetectable set is $F_1^{(0)}$, and $F_1^{(0)}=\{F_5,F_8\}$. Construct the dependency matrix of manual test for the undetectable faults of BIT. Extract the corresponding row in D_2, and we get a new matrix $D_2^{(1)}$ shown in Table IX. We can generate a diagnosis tree based on Table IX. The tree is shown in Fig. 5.

TABLE IX
THE MATRIX $D_2^{(1)}$

	M1	M2	M3	M4
F5	0	1	0	0
F8	0	0	0	1

In Fig. 5, we can diagnosis and isolate the fault which the BIT cannot detect. This tree has two branches and tests. If the M2 does not pass, the result is F5; or we execute the next test M4, if M4 does not pass, the result is F8. If M4 passes, that means system does not have fault at present.

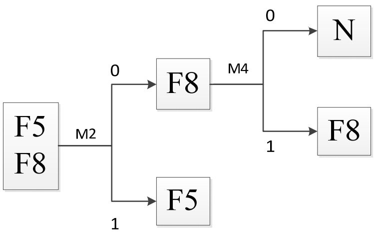

Fig. 5 The diagnosis tree based on $D_2^{(1)}$

The fault dictionary of BIT, the diagnosis tree for fuzzy fault set and undetectable faults have been obtained. We can give the fault detection and isolation. It has two typical applications.

Case I: When system is checking and maintaining, the BIT3 detects faults and other BITs are normal. We define the fault detection of BIT as the logical “1”, and normality as logical “0”. We have 4 BITs, so we get a logical vector [0,0,1,0]. Based on the $D_1^{(1)}$, we compare every row of the matrix with the vector, and make sure that the fuzzy fault set {F3, F4} is the result.

Then we diagnosis the fault based on the Fig. 3. The manual test M1 does not pass, we make sure that the final diagnosis result is F3.

Case II: When system is checking and maintaining, the BIT does not give any fault report. The undetectable fault set $F_1^{(0)}$ should be checked. According to Fig. 5, the manual test M2 does not pass, the diagnosis result is F5.

IV. CONCLUSION

This method combines the advantages of BIT and manual test, and gives a more reasonable diagnosis process. The method can directly use the test result of the BIT, and cover the shortages of BIT by using the manual test. At the same time, the method do not need other diagnosis knowledge, and only use the testability modeling data, which improve the feasibility of this method.

The case study shows that the method can detect and isolate the fault correctly, and give a more effective result.

ACKNOWLEDGMENT

The authors wish to express sincere gratitude to the anonymous reviewers and the Editor-in-Chief for their valuable comments and helpful suggestions, which have improved the quality of this paper greatly.

This work is supported by 973 Program (No.61316705), Basic Research Program (No. Z132014B002) and Pre-Research Foundation (No.51319040301).

REFERENCES

[1] Shi J. Testability Design Analysis and Verification(M). 1. Beijing: National Defense Industry Press, 2011.

[2] Zheng, Wei, Feng, Fuqiang, PI, Wenjing, CHEN, Xiaohui. Study on Fault Diagnosis Tree Generation Technology and Real Equipment Validation (J). Journal of the Academy of Equipment Command & Technology, 2013, 23(1): 119-122.

[3] Shi, Junyou, Gong, Jingjing, Xu, Qingbo. Improvement method for testability modeling with multiple faults (J). Journal of Beijing University of Aeronautics and Astronautics, 2010, 36(3): 270-273.

[4] Yang, Zhiyong, Xu, Aiqiang, Niu, Shuangcheng. Model and analysis of system testability based on multi-signal model (J). Journal of Engineering Design, 2007, 14(5): 364-368.

[5] Shi, Junyou, Zhang, Xin, Zou, Tiangang. Application of multi-signal modeling and diagnosis strategy design technology (J). Systems Engineering and Electronics, 2011, 33(4): 811-815.