System-level fault diagnosis method based on test model

Wei Cong¹, Hongkun Yu² and Jing Liu³
¹School of Information Engineering, Xijing University, Xi’an 710023, China
²Department of Continuous Education, Xi’an Shiyou University, Xi’an 710065, China
³School of Information Engineering, Xijing University, Xi’an 710023, China

Corresponding author and e-mail: Wei Cong, congweicw@126.com

Abstract. Aiming at the problems of many test states and uncertain test results in system-level fault diagnosis, a system-level fault diagnosis method based on test model is proposed. This method first divides the system nodes into multiple regions, then hierarchically organizes the regions, and implements a top-down hierarchical structure through the control nodes. On this basis, the guide words are used to describe system-level failure modes, formulate centralized test rules, and use data flows to identify failure modes caused by fault propagation and fault mutation, which reduces the state of fault testing and improves the accuracy of fault diagnosis. The example analysis shows that the method in this paper can meet the needs of system-level fault diagnosis, and has practical application value for shortening the time of system fault determination and improving the accuracy of troubleshooting.

1. Introduction
At present, large-scale multiprocessor systems are becoming more and more popular. Once some processors in the system fail, it often affects the functioning of the system and even causes huge losses. Therefore, it is very important to seek an effective diagnosis technology to quickly and accurately locate the faulty processor in the system [1].

System-level fault diagnosis is a common method to ensure high reliability of multiprocessor systems. The PMC model is the earliest and most studied system-level fault model [1-5]. However, the model requires that the number of failed nodes in the system does not exceed half of all nodes [6]. When the faulty node tests other nodes, the test results are uncertain, which increases the difficulty of fault diagnosis. The related literature use equations, groups, and probabilistic matrix methods to break this restriction and find all compatible failure modes [5][7][8], but the mutual testing process has not been described.

This paper proposes a system-level fault diagnosis method. This method organises system nodes by region, using control nodes to implement layered fault diagnosis model. Further, centralised test rules and test data transmission specifications were formulated to reduce the pressure on test data transmission. The guide words are used to describe the system failure mode, and the node self-diagnosis and regional comprehensive diagnosis are used to realise the fault location. The example analysis shows that the method in this paper can meet the needs of system-level fault diagnosis, and has practical application value for improving the accuracy of fault location.

2. System-level fault diagnosis modelling
2.1. Distributed centralised fault diagnosis model

The model divides the system nodes into multiple regions, and each region is assigned a control node. Then the model organises these regions into a hierarchical structure according to the system control relationship. The control node in the upper region is responsible for the testing and fault location of the local and lower region [8]. Figure 1 describes the distributed centralised fault diagnosis model.

![Figure 1. Distributed centralised fault diagnosis model.](image)

All nodes in the system are divided into 3 layers, L-3 is the highest layer and L-1 is the lowest layer in Figure 1. The control node of the L-3 layer is the control node of the entire system. The direct objects of system-level fault diagnosis are the control nodes of the region R5 and R6 in the L-2 layer. The control nodes of the region R5 and R6 are again the control node of the region R1, R2, R3 and R4 in the L-1 layer.

The distributed centralised fault diagnosis model combines the advantages of distributed and centralised fault diagnosis structures. The region reduces the scope of system-level fault diagnosis. The introduction of control nodes implements centralised fault diagnosis of the system and improves the accuracy and speed of fault diagnosis.

2.2. System-level test rules

The test rules are organized hierarchically by region, as shown in Figure 2.

![Figure 2. The structure of test rules.](image)

In one region the control node sends a test task, and others nodes feedback the test results after receiving the test task, and the control node makes a fault judgment. In multiple regions the upper control node sends a test task and collects the test results of the lower control nodes to judge the fault. The test data is transmitted together with the normal data flow. Figure 3 is the test data format.

![Figure 3. Test data format.](image)

The test data consists of ID, test_type and test_result. The ID identifies the control node. The field test_type and test_result is a multiplex field. Test_type is issued by the control node. Test_result is issued by the non-control node and records the fault node and fault mode.
3. System-level fault diagnosis process

3.1. Failure mode description
Guided word is used for safety analysis of safety-critical computer systems [9]. A guided word represents a type of failure mode. This paper describes the system-level failure mode by using three types of guide words: value, time and provision. The value describes the failure modes due to incorrect values, including wrong and out of range. The time describes the failure modes caused by time, including early and late. The provision describes the failure mode caused by the service, including commission and omission.

The system-level failure mode set formed by the guide words is expressed as:

$$Fault\_Mode = \{W, R, E, L, C, O\}$$

Where $W$ means that the value is wrong, $R$ means that the value is out of range, $E$ means that the time is advanced, $L$ means that the time lags, $C$ means that the service is commission and $O$ means that the service is omission. Therefore, the state of each node is expressed as:

$$Status = Normal | Fault.\{c|eFault\_Mode\}$$

For example, $INS. ServiceA. \{L\}$ means that the $ServiceA$ provided by the $INS$ node has a time failure and the failure mode is late.

3.2. System-level fault diagnosis method
System-level fault diagnosis based on guide words includes node self-diagnosis and regional comprehensive diagnosis. The node self-diagnosis mainly aims at the faults generated by the node itself. The regional comprehensive diagnosis can identify multiple faulty nodes in one or more regions, mainly aiming at the failure propagation and failure mutation.

3.2.1. Node self-diagnosis. The node self-diagnosis is to compare the predefined data flow of each node with the actually received data flow, and determine the failure mode according to the comparison result. The following uses the time domain as an example to explain the method of node failure judgment. Let the data transmission time of the node be $T$, $\Delta T$ be the data transmission time threshold, and $T'$ is the actual data transmission time. The method for judging the time domain fault of node is described as follows:

$$\text{if } T-T' > \Delta T \text{ then } d_i\_status_t=fault.\{E\}$$
$$\text{elif } T-T' = \Delta T \text{ then } d_i\_status_t=normal$$
$$\text{if } T-T' < \Delta T \text{ then } d_i\_status_t=fault.\{L\}$$

The result of node self-diagnosis is the union of the range of values $d_i\_status_v = \{\cdots\}$, time $d_i\_status_t = \{\cdots\}$ and services $d_i\_status_d = \{\cdots\}$:

$$d_i\_status = d_i\_status_t \cup d_i\_status_v \cup d_i\_status_d$$

3.2.2. Regional comprehensive diagnosis. The regional comprehensive diagnosis is carried out by the control node of the region. The failure mode of the node must be defined accordingly the system function first. The fault location is realised according to the reverse reasoning of the data flow. The reasoning starts from the fault node to find the input node and fault mode. If the failure mode of the input node can cause a failure then the failure mode of input nodes is recorded in the sets $F\_node$ and $F\_mode$. And so on, to find all nodes that traverse the data flow. The node in the final set $F\_node$ is the node that caused the system failure. Figure 4 shows a system composed of 2 layers and 3 regions. Assuming that node D faults with wrong value, the fault reasoning process is as follows.

The data input nodes of node D are node C in region A and node B in region C. Node B works normally. The fault of wrong value of node D is caused by the time delay or fault of wrong value of node C. The failure of node C is related to the time delay failure of node A. Thus the faulty value of
node D is caused by both node A and node C. Since node A directly receives the system input; it has the greatest impact on node D.

\[ \text{node A} \rightarrow \text{A.C.F.}\{W\} \rightarrow \text{node D} \]

\[ \text{node C} \rightarrow \text{B.A.}\{L\} \rightarrow \text{node A} \]

\[ \text{node B} \rightarrow \text{C.B.normal} \rightarrow \text{node B} \]

\[ I_1 \rightarrow \text{region A} \]

\[ I_2 \rightarrow \text{region C} \]

**Figure 4.** Fault reasoning structure diagram.

4. **Case analysis**

The integrated avionics system is the mission system of modern military aircraft and a typical distributed system. When verifying the air-to-air interception attack function on the ground, the weapon attack data value displayed by the HUD is incorrect, and a numerical display failure occurs. In the above case, the numerical display error appears in the human-machine interface, which is a typical system-level fault. The following describes the specific process of using this method to locate the faulty node.

4.1. **Fault diagnosis model of integrated avionics system**

The integrated avionics system is composed of multiple regions, each of which includes multiple devices. Due to space limitations, Figure 5 only shows the system structure of only three regions is used as an example to illustrate.

**Figure 5.** Fault diagnosis structure of integrated avionics system.

In Figure 5 the display control region is composed of HUD, MFD, UFCP, DCMP and other equipment. DCMP is a control node that sends weapon attack data to the HUD for display. DCMP is also the control node of the entire system. The atmospheric data computer is the control node of atmospheric data region. The data management processor is the control node of EW region.

The encoding of the test type is “0001” means power-on test, “0010” means start-up test, “0011” means periodic test. The test_result records the faulty node and failure mode. Figure 6 is the test_result in this example using double-byte coding method.

**Figure 6.** The double-byte encoding of test results.
The bits D15 to D8 represent the device ID in the region. The bits D7 and D6 indicate the working status, “00” and “01” indicate normal and fault respectively. The bits D5 to D0 indicate the failure mode, where the wrong value code is “00001”, the out-of-range code is “00101”, the time early code is “00011”, the time late code is “00100”, and service commission code is “00101” and a service omission code is “00110”.

4.2. System data flow and failure modes

Figure 7 shows the data flow when the integrated avionics system works in interception attack mode.

Figure 7. Data flow under the interception attack mode.

Table 1 shows the failure mode of each node in Figure 7.

| device | fault mode | fault propagation mode | failure mutation mode |
|--------|------------|-------------------------|-----------------------|
| DCMP   | W,R,E,L,C,O- | -                       | -                     |
| MC     | W,R,E,L,C,O ADM.W→11DCMP.W | MC.R→DCMP.O MC.E, MC.L→DCMP.W |
| ADC    | W,R,E,L,C,O ADC.W→MC.W   | ADC.R→MC.O ADC.E, ADC.L→MC.W |
| INS    | W,R,E,L,C,O INS.W→MC.W  | INS.R→MC.O INS.E, INS.L→MC.W |
| PDR    | W,R,E,L,C,O PDR.W→MC.W   | PDR.R→MC.O PDR.E, PDR.L→MC.W |
| SMS    | W,R,E,L,C,O SMS.W→MC.W   | SMS.R→MC.O SMS.E, SMS.L→MC.W |

Table 2 describes the content and timing of the data flow.

| data flow         | data definition                                      | period(ms) |
|-------------------|-----------------------------------------------------|------------|
| flight data       | pitch angle, roll angle, true heading, speed, acceleration | 25±5%      |
| atmospheric data  | barometric pressure altitude, atmospheric density    | 25±5%      |
| missile correction command | position deviation                                   | 100±5%    |
| target data       | relative distance, relative bearing, approach speed  | 50±5%      |
| initialisation data | preset intercept point coordinates                  | 100±5%    |
| missile flight time | flight time                                         | 100±5%    |
| weapon attack data | control point, maximum launch distance, minimum launch distance | 100±5%    |

4.3. Analyse and locate the source of the fault

The failure mode of the Table 1 shows that the numerical display error is caused by the wrong value of the MC component, which is caused by the wrong value of the ADC, INS, PDR and SMS. To locate the source of the fault, it is necessary to further analyse the messages that are output by the four components to the MC component. During the analysis and test, the message time generated by the ADC was determined, and the conclusions obtained are shown in Table 3.

Table 3. Time analysis of determining atmospheric data messages.
| message time (ms) | calculation value of period (ms) | calculated value (air pressure) | conclusion |
|------------------|---------------------------------|--------------------------------|------------|
| 37230000         | -                               | 360                            | wrong      |
| 37230025         | 25                              | 360                            | wrong      |
| 37230049         | 24                              | 360                            | wrong      |
| 37230075         | 25                              | 360                            | wrong      |
| 37230100         | 25                              | 360                            | wrong      |
| 37230124         | 24                              | 360                            | wrong      |
| 37230149         | 25                              | 360                            | wrong      |

It can be concluded from Table 3 that the failure mode of the ADC component is determined to be the wrong value. By analogy, the analysis methods for determining the three components of INS, PDR, and SMS are the same. The analysis concludes that these three components are working properly. From this, it can be judged that the fault source of the numerical display error is ADC device.

5. Conclusions
First, a distributed centralised fault diagnosis model is proposed, regions and control nodes are introduced to reduce the test status of system nodes. Suppose that the system has 200 nodes, which are divided into 20 regions, and each region with 10 nodes. The 20 regions are divided into 4 layers, each layer with 5 regions, then the number of test states is $O(10^2 \times 5 \times 4)$ is far less than the number of test states before stratification is $O(200^2)$ [10].

Secondly, the centralised test rules are proposed, the control node in each area is used to send test tasks and receive the test results to improve the speed of fault diagnosis.

Finally, using guide words to describe failure modes, using data flow to establish failure propagation and failure mutation modes, using node self-diagnosis and system-level comprehensive diagnosis to achieve fault location, and improve the accuracy of fault location.

Application example shows that the method in this paper can meet the fault diagnosis needs of complex systems, and has practical application value for shortening the time of system fault determination and improving the accuracy of troubleshooting.

References
[1] GUI Weixia, Lu Qian and SU Meili 2020 *Journal of Electronics & Information Technology*. 42 1102-1109. (In Chinese)
[2] YUN Long, LIANG Jiarong and ZHOU Ning 2017 *Application Research of Computers*. 34 2638-2641+2650. (In Chinese)
[3] WANG Yuxi, LI Zhan, XU Minghui and Ma Hao 2016 *The 8th IEEE International Power Electronics and Motion Control Conference (IPEMC-ECCE Asia)*. 2493-2497
[4] GUI Weixia and LIU Cui 2017 *Computer Engineering and Applications*. 53 78-82+145. (In Chinese)
[5] FENG Hailin, LEI Hua and LIANG Lun 2017 *Journal of Nanjing University of Science and Technology*. 41 479-485. (In Chinese)
[6] XIE Min, YE Liangcheng and LIANG Jiarong 2018 *Concurrency and Computation: Practice and Experience*. 30 1-8
[7] XUAN Hengnong, ZHANG Dafang and ZHANG Ming 2003 *Journal of Electronic*. 31 694-697. (In Chinese)
[8] XUAN Hengnong, ZHANG Runchi, ZUO Miao and LIU Tiantian 2014 *Acta Electronica Sinica*. 42 2536-2542. (In Chinese)
[9] GAO Dong, Xiao Yao, ZHANG Beike, XU Xin and WU Chongguan 2020 *Chemical Industry and Engineering Progress*. 1-11[2020-05-27]. (In Chinese)
[10] ZHOU Hong and CHEN Zhixiong 2016 *Acta Aeronautica et Astronautica Sinica*. 37 3821-3831. (In Chinese)