Module Partition Method of Embedded Multitasking Software Based on Fuzzy Set Theory

Embedded Multitasking Software

Yunpeng Gu[1]
Yangzhou Polytechnic Institute
Art Design College, Yangzhou, 225107, China

***Abstract*—In order to improve the reliability of embedded multitask software, a module partition method based on fuzzy set theory is proposed to solve the problem of large number of software failures and high frequency of failures. Firstly, the characteristics of embedded multitask software are analyzed, and the constraint parameter distribution model is constructed. Then the reliability parameter analysis model of embedded multitask software is constructed, and the multilevel fuzzy metric structure combined with quantitative recursive analysis method is used to complete the division of embedded multitask software modules. The simulation results show that the accuracy of the proposed method exceeds 99% after the number of iterations exceeds 50. The experimental results show that the method has high reliability, high precision and high practicability.**

***Keywords*—Multitasking; embedded; modular; fuzzy set theory**

I. Introduction

In the running of embedded software, the efficiency of information processing can be improved by dividing modules under multi-tasks [1, 2]. However, this method is easy to lead to software function conflicts, which will lead to software failures, and software reliability cannot be guaranteed, affecting the user experience. In the process of using embedded multitask software, the application direction of software can be combined to divide the modules, so as to improve the reliability of software application [3]. At present, some scholars have made relevant research on this. Some scholars proposed a software module division method based on the combination of genetic algorithm and particle swarm optimization, established a mathematical model of the division problem, used genetic algorithm to find a feasible solution, and used particle swarm optimization to find the optimal scheme [4]. Other scholars proposed to maintain the independence of modules by scheduling model partition [5]. However, this method does not consider the reliability of software module partition, so this paper proposes an embedded multi-task software module partition method based on fuzzy set theory. This method can improve the reliability of software module division on the basis of maintaining module independence, thereby reducing the probability of software failure and improving the customer experience. Embedded multitasking software reliability test is based on the reliability of embedded multi-tasking constraint parameter analysis software. The embedded multitasking software reliability parameters analysis model established the embedded multitasking software module which is divided by the intelligent analysis method and embedded multitasking software module partition method based on fuzzy set theory is put forward [6]. Firstly, the constraint parameters of embedded multi-task software module partition are analyzed, and then the software compatibility and human-computer interactions are tested by using multi-layer index parameter constraint control method. Combined with the software reliability constraint index, the software reliability automatic test is realized. Finally, the simulation results show that this method is effective in improving the partition ability of embedded multitask software modules. The proposed method can effectively divide embedded multitask software modules, improve the reliability of software, and provide new ideas and ways for software improvement in the current market.

II. Reliability Constraint Index and Feature Analysis of Embedded Multitasking Software

A. Reliability Constraint Index of Embedded Multitasking Software

In order to realize the partition of embedded multitask software modules, it is necessary to analyze the application characteristics of embedded multitask software modules. Based on the attribute, portability and importance of embedded multitask software, reliability constraint index analysis method is used to analyze the application of the software. The software reliability analysis mainly has four dimensions, namely, reliability measurement parameters, reliability parameter system, reliability measurement indicators and reliability indicator system. Among them, reliability measurement parameter refers to the mathematical attribute of quantitative description of reliability. Reliability parameter system is a set of software reliability parameters. Reliability metrics refer to the required value of a software reliability parameter. The reliability index system is the required value of all reliability parameters of a certain software. Analyze the characteristics of software module with product application. The product adaptability index distribution of embedded multitasking software is shown in Fig. 1.

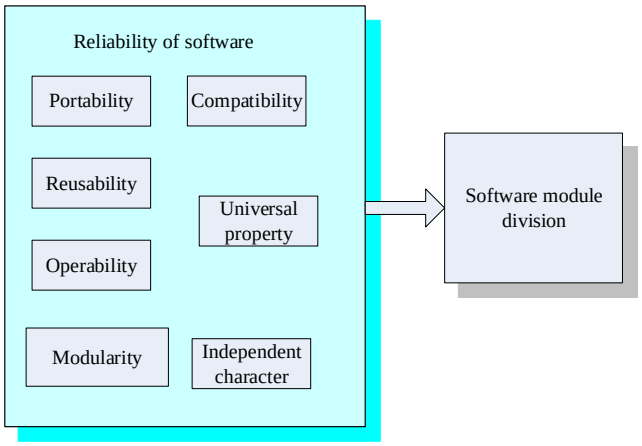


Fig. 1. Product Adaptability Index of Embedded Multitasking Software.

According to Fig. 1, the embedded multitask software modules are divided according to their quality. In combination with the operability of software applications, the adaptive feature detection model of embedded multitask software is established. In addition, the distributed algorithm [7] is used to establish the partition model of embedded multitask software modules, which improves the universality of embedded multitask software. A software quality measure is carried out by adopting a quantitative analysis method of software reliability. The reliability test of the software is mainly divided into the structural test and the object-oriented test. The complexity and the reliability distribution of the software are combined, and the embedded multi-task software module is divided into two levels. According to the above analysis, the parameter system of the secondary constraint index of the software reliability is established as shown in Fig. 2.

According to the reliability constraint parameter system of embedded multitask software shown in Fig. 2, the reliability of embedded multitask software is tested and evaluated by structure-oriented method.

The reliability of embedded multitask software refers to the probability that the software will not cause system failure within the specified time under the specified environment. Software reliability is one of the important indexes to measure software quality. It is not only related to software errors, but also related to system input and system use. Generally speaking, the more software failures occur or the shorter event interval, the lower software reliability.

Embedded multitask software reliability test is to ensure and verify the reliability of the software. It is a kind of random test. Its main feature is to test software according to the way users actually use software. Generally, embedded multitask software reliability test is divided into software reliability growth test and software reliability verification test. If there is no requirement of reliability index and the current reliability level needs to be evaluated, there are also embedded multitask software reliability bottom test. The reliability test of embedded multitask software is different from the general software test (such as unit test, integration test, function test, performance test, boundary test, etc.) in terms of test purpose, test method, test object, etc. The comparison between embedded multitask software reliability verification test and general software test is shown in Table I.

In Fig. 2, t1~t4 refer to reliability, failure probability, mean time before failure and mean time between failures respectively; wherein, reliability refers to the probability of success of software to complete specified functions under specified conditions and within specified time. Failure probability refers to the probability of losing normal operation and specified functions under specified conditions and within specified time. Average time before failure refers to the mean value from the current time to the next failure time. Mean time between failures refers to the mean time interval between two successive failures. The reliability test of embedded multitask software generally uses black box test, because it is a use-oriented test, it does not need to understand the structure of the program and how to realize it. The main flow of software reliability test is shown in Fig. 3.

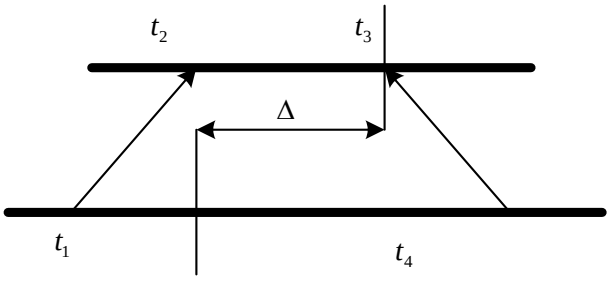


Fig. 2. Software Reliability Two-level Constraint Index Parameter System.

TABLE I. COMPARISON BETWEEN RELIABILITY VERIFICATION TEST AND GENERAL TEST OF EMBEDDED MULTITASK SOFTWARE

Item compared	Software reliability verification test	General software testing
Test purpose	Quantitative estimation of reliability level of software products	Discovery of software errors
Tester	The user usually takes part in the test	It is usually measured by the developer
Testing stage	Software acceptance phase	Trial software development stage
Test method	Random test based on software operation profile.	Testing based on requirements / structure
Test object	The final form of software products	Intermediate form of software products
Test feature	Do not eliminate software defects	Modify software errors and constantly improve software quality

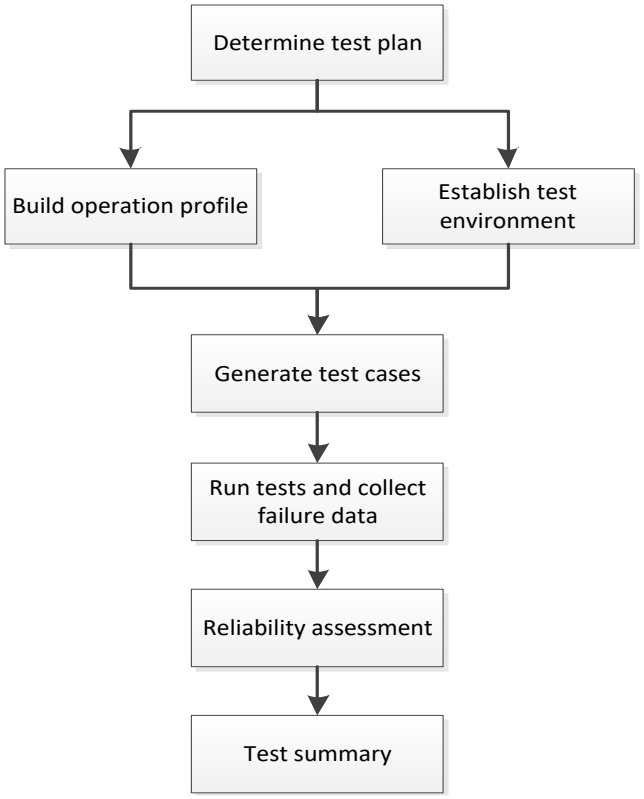


Fig. 3. Reliability Testing Process of Embedded Multitask Software.

The main idea of embedded multitask software reliability test is to test the software at random according to the statistical rules of users' actual use. At present, the use of software is usually modelled in the form of operation profile. The construction of operation section can be refined layer by layer from top to bottom. The development process of the operation profile is shown in Fig. 4.

1) The customer profile consists of independent customer type series.

Customer type is one or more customers in the group who use the system in a similar way. These customers are significantly different from other customers in the way of using the software.

The probability of occurrence needs to be determined for each customer type in the customer profile. In the case of single software customer, there is no need to establish customer profile.

2) User profile is a set of user groups and their probability of occurrence, which is usually based on the customer profile.

3) System mode profile is a set of system mode and its corresponding probability of occurrence. System patterns can be derived from software requirements or usage analysis.

4) The function profile is to decompose the functions required by the system in the system mode and determine the probability of each function.

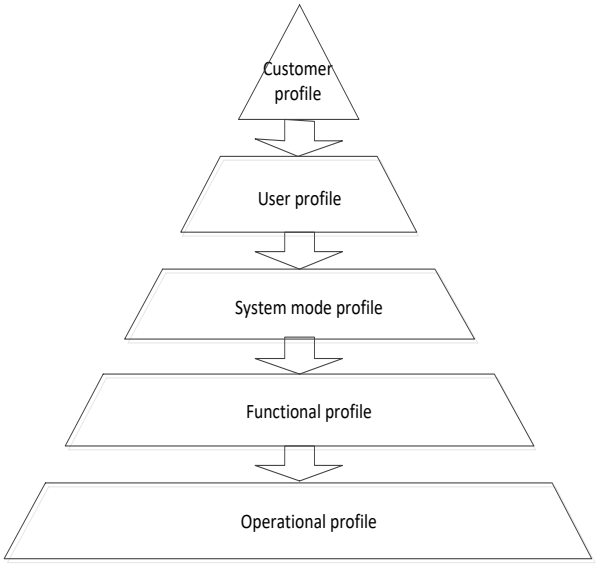


Fig. 4. Operation Profile Development Process.

The construction of function profile usually needs to get the function list according to the software development task or the software requirements document, and then combine with various situations of the operating environment, and assign the probability of occurrence for each function.

5) Operation profile is a set of operations and their probability of occurrence. The main step to determine operation profile is to list operations and determine the probability of occurrence of each operation. A function can be mapped to one or more operations, and a group of functions can also be re merged into a group of different operations, so the list of operations can be obtained according to the function profile.

The embedded multitask software reliability test environment refers to the software and hardware environment that provides test input and collects test output for the tested software. For the reliability test of real-time embedded software, we can use the test environment of full digital simulation technology, hardware in the loop simulation platform and system joint test. In order to get the real reliability test results, the reliability test should be carried out in the real environment. But in many cases, it is unrealistic to test the software reliability in real environment, so the software reliability test usually uses the semi physical simulation environment. Generally, the test environment of semi physical simulation (or full digital simulation) has the following requirements:

1) The operations involved in the operation profile can be performed in the test environment.

2) Meet the real-time requirements of the test excitation, that is, the excitation input meets the requirements of the input interface, data format and input timing of the real cross-linking environment, and the input logic of the real cross-linking environment is the same.

3) Be able to collect test result data for software reliability test analysis.

The failure data of embedded multitask software is the basis of software reliability test and evaluation. In the process of reliability test execution, it is necessary to record the actual results of each test case as required, and judge whether the use case passes according to the expected test results of each use case and the pass criteria of the use case. If not, record it. The problems found in the reliability test of embedded multitask software should be analyzed and dealt with according to the types of problems. The software problems caused by the defects of the tested embedded multitask software itself should be included in the failure of the tested software. The software problems caused by the improper design of the test environment, test methods and test cases other than the tested software should not be included in the failure of the software. In addition, the category and severity level of each embedded multitasking software problem should be described. The categories of software problems include design problems, document problems, program problems and other problems. The severity level of the problems can be described by level 4.

Reliability verification test is a kind of statistical test, which should be selected in the test planning stage. When choosing the statistical test plan, many factors should be considered, such as the type of verification index, the quality requirements of software, the maximum test time, the maximum failure number, the test cost, the trade-off between cost and time. For the verification test of product reliability using success rate, the verification test scheme of success rate is usually used. The specified success rate is the probability that a product will complete the required function or that the product will succeed in the test under the specified conditions. The observed success rate can be defined as the ratio of the number of products that have not failed to the total number of test products at the end of the test or the ratio of the number of successful tests to the total number of tests. The main parameters of the success rate verification test scheme are:

R : Success rate;

R_0 : Acceptable success rate;

R_1 : Unacceptable success rate;

d : Identification ratio of success rate $(1-R_1)/(1-R_0)$;

n : Receive the required number of fixed tests;

r : Accumulated failure number;

r_{RE} : Rejection failure number;

α : Manufacturer's risk, i.e. rejection probability when $R-R_0$;

β : User risk, i.e. the probability of reception when $R-R_1$.

The verification test scheme of success rate includes the truncated sequential statistical scheme and the fixed number test statistical scheme. The schematic diagram of the truncated sequential statistical scheme is shown in Fig. 5.

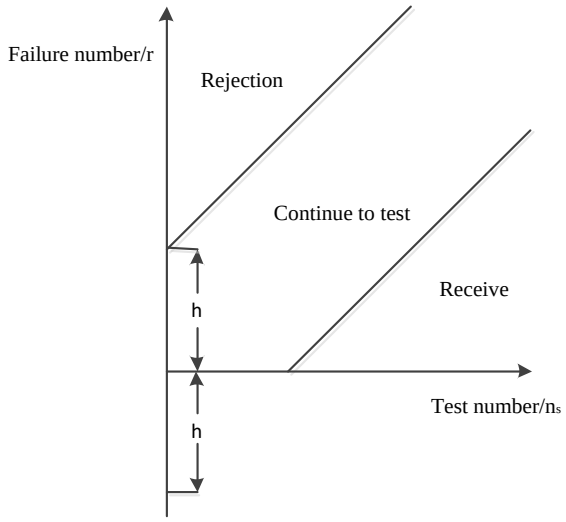


Fig. 5. Schematic Diagram of Truncated Sequential Statistical Scheme.

When $r \le s$. n_s-h, receive;

When $r \ge s$. $n_s + h$, rejection;

When $s.n_s - h \le r \le s.n_s + h$, continue the test.

Sequential test to cut off line, acceptance or rejection criteria.

When $n_s = n_t$, when $r<r_t$, receive;

When $r>r_t$, reject;

Among them, n_s is the number of tests accumulated in the sequential test scheme, s is the slope of the receiving and rejection lines, h is the intercept of the receiving and rejection lines on the vertical line, n_t is the number of truncated tests, and r_t is the number of truncated failures.

Given R_0, d, α, β, the test number n required for receiving the judgment and the failure number r_{RE} required for making the rejection judgment can be obtained by looking up the table. The comparison between the two schemes is shown in Table II.

TABLE II. COMPARISON OF TWO STATISTICAL SCHEMES

Comparison items	Advantage	shortcoming
Truncated sequential Statistical scheme	1. The minimum number of average tests and failures required for judgment; 2. It has a certain maximum value in cumulative test number and cumulative failure number.	1. The number of failures and the cost of tested products vary greatly; 2. The maximum number of accumulated tests and failures may exceed the equivalent number of accumulated tests and failures.
Fixed number test Statistical scheme	1. The change range of failure number and cost of tested products is small; 2. The maximum number of cumulative tests is less than that of similar truncated sequential tests.	1. The average failure number and test number are large; 2. In order to make a judgment, the products with good quality and bad quality shall experience the most cumulative tests or failures.

The embedded multitask software reliability testing technology is applied to a configuration item software of an airborne weapon product. Firstly, according to the requirement documents and the actual operation of the embedded multitask software, the operation profile covering the whole working process of the software is constructed. Extract specific system execution task scenarios, design test cases for each scenario respectively, and describe the test cases in detail. Object oriented reliability testing method. The calculation formula of software package constraint index is as follows:

$$MHF = \frac{\sum_{i=1}^{TC}\sum_{m=1}^{M_d(c_i)}\left(1-V\left(M_{mi}\right)\right)}{\sum_{i=1}^{TC}M_d\left(C_i\right)} \tag{1}$$

$$V\left(M_{mi}\right) = \frac{\sum_{j=1}^{TC} is\_visible\left(M_{mi}, C_j\right)}{TC-1} \tag{2}$$

$$is\_visible\left(M_{mi}, C_j\right) = \begin{cases} 1, & if \begin{cases} j \neq i, \\ C_j\, may call\, M_{mi} \end{cases} \\ 0, & otherwise \end{cases} \tag{3}$$

In the formula, TC is the total number of attributes distributed by embedded multitasking software, and $M_d\left(C_i\right)$ represents the number of categories in reliability attribute category C_i.

The regression analysis of embedded multitasking software reliability measurement is carried out by using the method of multi-factor metrology [8]. Combined with the method of bottom design and top-level encapsulation, the product quality of embedded multitasking software is evaluated. Multi factor measurement is a method of measuring multiple evaluation units by using multiple indicators. Its basic idea is to convert multiple indicators into an indicator with comprehensive evaluation ability, so as to reduce calculation complexity and improve calculation efficiency and accuracy. Based on the above, the ambiguity constraint parameters of embedded multitasking software module division are obtained.

$$AHF = \frac{\sum_{i=1}^{TC}\sum_{m=1}^{A_d(c_i)}\left(1-V\left(A_{mi}\right)\right)}{\sum_{i=1}^{TC}A_d\left(C_i\right)} \tag{4}$$

$$V\left(A_{mi}\right) = \frac{\sum_{j=1}^{TC} is\_visible\left(A_{mi}, C_j\right)}{TC-1} \tag{5}$$

$$is\_visible\left(M_{mi}, C_j\right) = \begin{cases} 1, & if \begin{cases} j \neq i, \\ C_j\, may call\, A_{mi} \end{cases} \\ 0, & otherwise \end{cases} \tag{6}$$

In the formula, the $A_d\left(C_i\right)$ is the number of the attributes of the C_i, the constraint index parameter model divided by the embedded multi-task software module is constructed according to the analysis, and the indexes such as the correctness, the reliability, the operation efficiency, the integrity and the usability of the software are determined, and a constraint parameter index distribution model for dividing the embedded multi-task software modules is constructed [9, 10, 11].

B. Reliability Feature Analysis of Embedded Multitasking Software

Let the reliability measurement domain of embedded multitask software be U, and U can be expressed by exact numerical value. Linear programming technology is used to analyze the reliability characteristics of embedded multitasking software, and the next year planning analysis is carried out combined with the inheritance of software reliability. The reliability characteristic analysis model of embedded software is established, and the output: SC_i (Ex_i, En_i, He_i), I = 1, 2, 3, …, n. The formula for calculating the reliability feature distribution of embedded multitasking software is as follows:

$$MHF = \frac{\sum_{i=1}^{TC}M_i\left(C_i\right)}{\sum_{i=1}^{TC}M_a\left(C_i\right)} \tag{7}$$

$$M_a\left(C_i\right) = M_d\left(C_i\right) + M_i\left(C_i\right) \tag{8}$$

Wherein, $M_i\left(C_i\right)$ is the reliability test factor of $C_i\left(i=1,2,...,n\right)$ class, and the reliability design of embedded multitask software is carried out with the method of fuzzy clustering analysis. The calibration model characteristics of embedded multitasking software reliability measurement are expressed as $U \to [0,1], \forall x \in U, x \to SC(x)$. The reliability feature analysis of embedded software is carried out by using the method of multimodal data fusion feature analysis, and the comprehensive measurement model of software is described as follows:

$$AIF = \frac{\sum_{i=1}^{TC} A_i(C_i)}{\sum_{i=1}^{TC} A_a(C_i)} \tag{9}$$

$$A_a(C_i) = A_d(C_i) + A_i(C_i) \tag{10}$$

In the formula, the software reliability attribute class function is $A_i(C_i)$, and the quantitative characteristic formula of reliability test with property class $C_i(i=1,2,...,n)$ is the number of attributes whose $En_i = (Ex_i - Ex_{i-1})/3$ $A_d(C_i)$ is $C_i(i=1,2,...,n)$. Based on the analysis, the constraint parameter distribution model of embedded multitask software module partition is constructed, the ambiguity factor of embedded multitask scheduling and module segmentation is established, and the embedded multitasking software module is divided into embedded multitasking software modules combined with fuzzy dynamic testing method [12].

III. Optimal Design of Embedded Multitask Software Module Partition Method

A. Embedded Multi-task Scheduling and Module Segmentation and Ambiguity Analysis

On the basis of constructing the constraint parameter distribution model of embedded multitasking software module partition, the optimal design of embedded multitasking software module partition is carried out. In this paper, an embedded multitasking software module partition method based on fuzzy set theory is proposed [13]. Fuzzy set theory regards the object to be investigated and the fuzzy concept reflecting it as a certain fuzzy set, establishes an appropriate membership function, and analyzes the fuzzy object through the relevant operations and transformations of the fuzzy set. Based on fuzzy mathematics, fuzzy set theory studies the imprecise phenomena. The module partition of embedded multitask software is an imprecise problem, so the research is based on fuzzy set theory. Factor of embedded multi-task scheduling and module segmentation is established, and the compatibility and human-computer interactions of the software are measured by the method of multi-layer index parameter constraint control. The formula for calculating the coupling factor of embedded multi-task software module is as follows:

$$COF = \frac{\sum_{i=1}^{TC}\left(\sum_{j=1}^{TC} is\_client(C_i, C_j)\right)}{TC^2 - TC} \tag{11}$$

$$is\_client(C_c, C_s) = \begin{cases} 1 & iff\ C_c \Rightarrow C_s \wedge C_c \neq C_s \\ 0 & otherwise \end{cases} \tag{12}$$

In the formula, $TC^2 - TC$ is the maximum degree of coupling of the reliability distribution of embedded multi-task software. The multi-level fuzzy metric structure model of the embedded multi-task software module is established, and the method for combining the multi-state factor (POF) is adopted to obtain the multi-dimensional combined test index distribution set of the software to be as follows:

$$POF = \frac{\sum_{i=1}^{TC} M_o(C_i)}{\sum_{i=1}^{TC}\left[M_n(C_i) \times DC(C_i)\right]} \tag{13}$$

$$M_d(C_i) = M_n(C_i) + M_o(C_i) \tag{14}$$

The embedded multi-task software reliability measurement model is composed of five states, namely $\lambda = (X, O, A, B, \pi)$, where $X = \{x_i, i = 1,2,3\cdots, N\}$ is the fuzzy coupling state factor in the embedded multi-task software reliability measurement model, and the $O = \{o_j, j = 1,2,3\cdots, M\}$. O is an embedded multi-task software safety evaluation model observation state, $C_i(i=1,2,...,n)$ represents the reliability test set of embedded multi-task software, which is defined as follows:

$$\begin{aligned} &\max F(X) = (F_1(X), F_2(X), ..., F_n(X)) \\ &s.t. \quad g_j(X) \le 0 \quad (j = 1,2,..., p) \\ &\qquad h_k(X) = 0 \quad (k = 1,2,..., p) \end{aligned} \tag{15}$$

According to the above analysis, the multi-level fuzzy measurement structure model of embedded multitask software module partition is established, and the parameter indexes such as serviceability, compatibility and portability of embedded software are analyzed by combining reliability merging and comprehensive measurement method, so as to improve the ability of embedded multitask scheduling and module segmentation and reliability measurement [14].

B. Embedded Multitasking Software Module Partition and Optimization Measurement

Based on the above contents, the multi-level fuzzy measurement structure model of embedded multitask software module is studied and constructed, and the number of embedded multitask software is set. However, the software module division based on this method has the problem of weak correlation between embedded multitask software. [15]. In order to solve this problem, a method is proposed to optimize the resource allocation of embedded multitask software by using association combination detection, and build an embedded multitask optimal scheduling and adaptive control model. The fuzzy membership function of embedded multi-task software is described as:

$$\begin{cases} L = \{L_1, L_2, ..., L_q\} \\ R = \{R_1, R_2, ..., R_p\} \end{cases} \tag{16}$$

In the above formula, V_k represents the nonlinear dynamic measurement function of software reliability. For the k software reliability attribute feature set, set up the security reliability measurement feature distribution set of embedded multitask software, map the reliability linear programming

feature set L_k of embedded multitask software to the target test set R_l by spatial programming method, and calculate the nonlinear feature quantity of software reliability measurement, which is described as follows:

$$G_{EV,q\times p}=\begin{vmatrix} G_{EV,11} & \cdots & G_{EV,1p} \\ \cdots & \cdots & \cdots \\ G_{EV,q1} & \cdots & G_{EV,qp} \end{vmatrix} \tag{17}$$

Based on the comprehensive measurement model, the quantitative feature distribution set of embedded multitasking software module partition is obtained, and the principal component feature quantity of embedded multitasking software module partition is represented by *SC* (*Ex*, *En*, *He*). According to the above analysis, the statistical analysis model of embedded multitasking software module partition is constructed, and the statistical feature distribution matrix is obtained as follows:

$$L=\begin{bmatrix} l_{11} & l_{12} & \cdots & l_{kp} \\ l_{21} & l_{22} & \cdots & l_{2p} \\ \cdots & & & \\ l_{q1} & l_{q2} & \cdots & l_{qp} \end{bmatrix} \tag{18}$$

The embedded multitasking software module is divided by quantitative recurrent analysis, and the optimized test set function is obtained as follows:

$$V=\begin{bmatrix} v_{11} & v_{12} & \cdots & v_{kp} \\ v_{21} & v_{22} & \cdots & v_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ v_{q1} & v_{q2} & \cdots & v_{qp} \end{bmatrix} \tag{19}$$

Based on the above contents, the division of embedded multi task software modules is realized on the basis of ensuring the independence and stability of software modules. [16, 17, 18, 19].

IV. Simulation Experiment and Result Analysis

In order to verify the application performance of the method in the implementation of the embedded multi-task software module division, the simulation test is carried out. The reliability test environment of embedded multitask software is built by the general system test platform Gete. Among them, the general embedded software simulation test platform consists of three test hosts, two test targets, related I / O and signal conversion devices. All devices in the platform are connected by Ethernet, and the two test target computers communicate with each other in real time through the memory reflective real-time optical fiber network [20, 21]. Position diagram is shown in Fig. 6.

Software reliability can be tested by using the following definitions: the probability that software will not cause system failure under specified conditions and within specified time. This probability is a function of system input and system use, as well as a function of defects in software. The system input will determine whether existing defects will be encountered. The probability that the software will not cause system failure within the specified time, that is, the success rate of the software. Therefore, the success rate is used to verify the fixed number test statistical scheme. After analysis, it is determined that α = 10%, β = 10%, R0 = 0.9701, R1 = 0.91.

Identification ratio D = (1-r1) / (1-r0) = 3.01

According to α = 10%, β = 10%, R0 = 0.9701, d = 3.01, the values of parameter n and RRE are obtained by looking up the table: n = 101, RRE = 6.

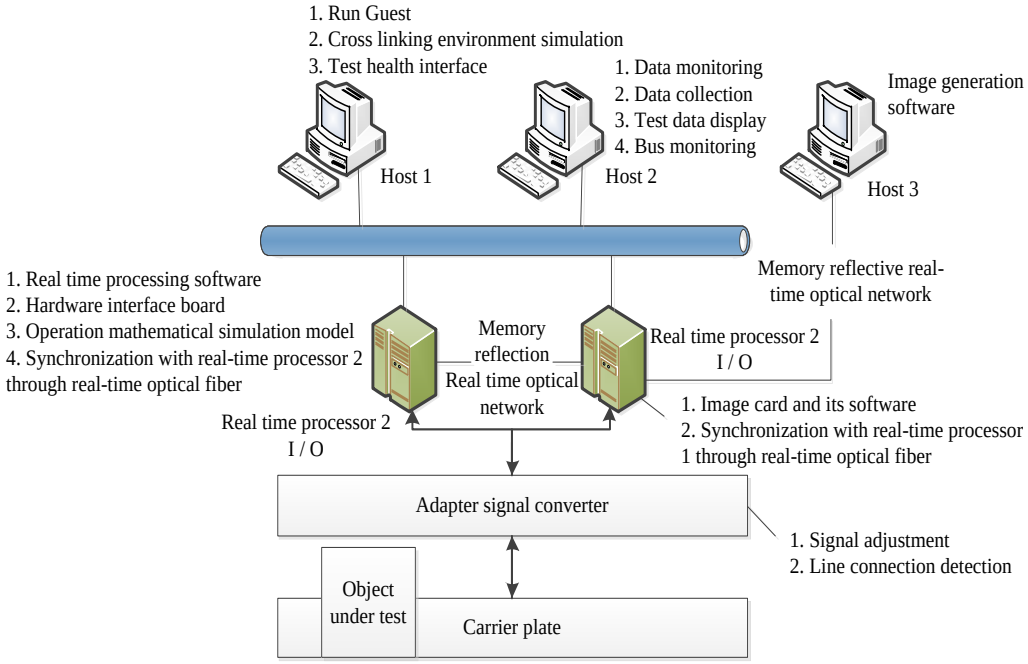


Fig. 6. General System Test Platform Geste Structure Diagram.

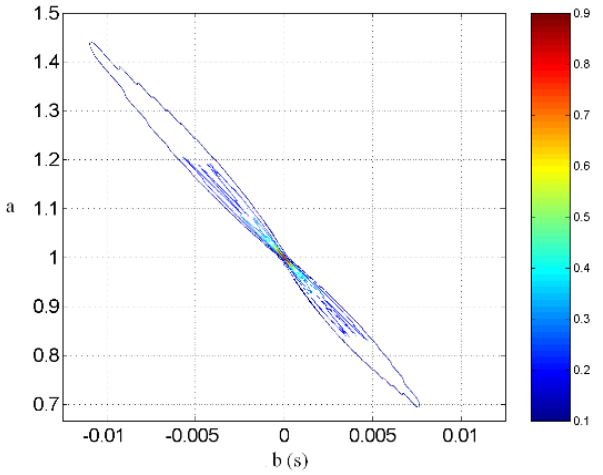
Therefore, 101 reliability test cases need to be designed for execution. If the number of software failures is less than or equal to six during the test, the software will be accepted, otherwise it will be rejected [22].

In the software reliability test, there are four failures, less than the maximum number of software failures, which meet the requirements of reliability test program. And the embedded multi-task software module is divided and analyzed in combination with the Matlab and the C ++, the distribution area of the parameter sampling of the embedded multi-task software reliability constraint index parameter is set to [0, 0.6] [0.6, 0.8] [0.8,1], and the distribution of test indicators at all levels is shown in Table III.

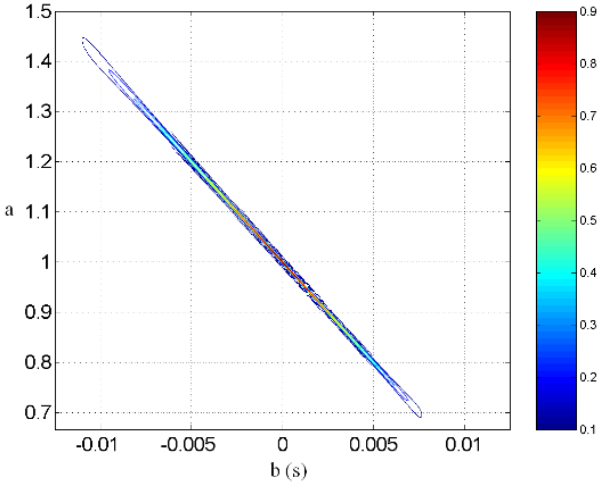
By comparing the reliability verification test of embedded multitask software with the general software test, the main process of software reliability test is designed. According to the embedded multitask software, the reliability software and hardware environment of test input and test output are provided [23]. According to the requirements document and the actual operation of the embedded multitask software, the operation section covering the whole working process of the software is constructed, and the specific system execution task scenarios are extracted. Test cases are designed for each scenario respectively, and the test cases are described in detail. The calculation formula of software package constraint index is listed, and the fuzzy constraint parameters of embedded multitask software module division are obtained [24, 25, 26, 27]. Then the fuzzy analysis of embedded multitask scheduling and module division is carried out, the embedded multitask software module is established, and the multi-level fuzzy measurement structure model is divided [28]. According to the test results of the above index parameters, the embedded multitasking software modules are divided, and the reliability test output of each software system is shown in Fig. 7.

TABLE III. DISTRIBUTION OF TEST INDICATORS AT ALL LEVELS OF EMBEDDED SOFTWARE

Primary test index	Weight	II. Test index	Test data
SC1	0.36	INDEX11	0.325
		INDEX12	0.156
		INDEX13	0.446
		INDEX14	0.332
		INDEX15	0.563
		INDEX16	0.532
		INDEX17	0.426
		INDEX18	0.675
SC2	0.67	INDEX21	0.532
		INDEX22	0.436
		INDEX23	0.547
SC3	0.32	INDEX31	0.437
		INDEX32	0.225
		INDEX33	0.853
		INDEX34	0.326
		INDEX35	0.485
		INDEX36	0.325
SC4	0.56	INDEX41	0.567



(a) Initial Status.



(b) Module Partition Output.

Fig. 7. Reliability Test Output of Software System.

Analysis of Fig. 3 shows that using this method can effectively achieve the embedded multi-task software module division, compare the reliability of the software, get the comparison results as shown in Fig. 8 [29].

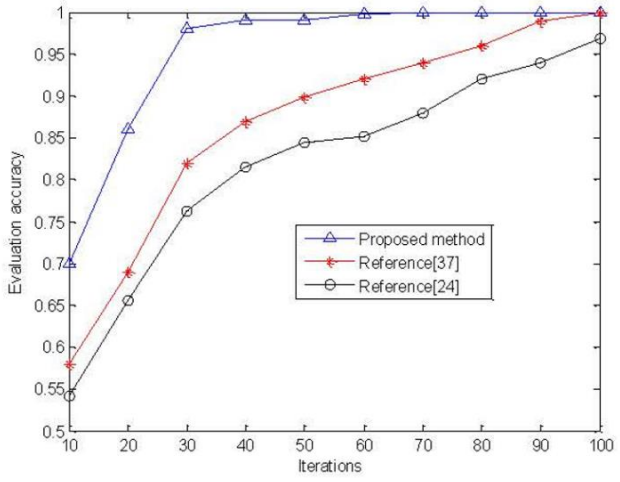


Fig. 8. Reliability Comparison Results for the Software.

In order to understand more clearly and accurately the comparison results of simulation experiments on the reliability of embedded multitask software in this paper, 10 iterations are taken and detailed data comparison is conducted according to Fig. 8 [30, 31, 32, 33]. The comparison data is shown in Table IV [34, 35, 36].

TABLE IV. RELIABILITY COMPARISON DATA OF EMBEDDED MULTITASK SOFTWARE

iterations	Proposed method Evaluation accuracy	Reference [37] Evaluation accuracy	Reference [24] Evaluation accuracy
10	0.7	0.58	0.54
20	0.86	0.82	0.65
30	0.98	0.86	0.76
40	0.99	0.88	0.81
50	0.99	0.90	0.83
60	1	0.92	0.84
70	1	0.93	0.86
80	1	0.95	0.92
90	1	0.97	0.93
100	1	1	0.96

According to Table IV and Fig. 8, when the number of iterations is 10, the accuracy of Proposed method is 0.12 higher than Reference [37], and 0.16 higher than Reference [24]; When the number of iterations is 20, the accuracy of Proposed method is 0.04 higher than Reference [37], and 0.21 higher than Reference [24]; When the number of iterations is 30, the accuracy of Proposed method is 0.12 higher than Reference [37], and 0.22 higher than Reference [24]; When the number of iterations is 40, the accuracy of Proposed method is 0.11 higher than Reference [37], and 0.18 higher than Reference [24]; When the number of iterations is 50, the accuracy of Proposed method is 0.09 higher than Reference [37], and 0.16 higher than Reference [24]; When the number of iterations is 60, the accuracy of Proposed method is 0.08 higher than Reference [37], and 0.16 higher than Reference [24]; When the number of iterations is 70, the accuracy of Proposed method is 0.07 higher than Reference [37], and 0.14 higher than Reference [24]; When the number of iterations is 80, the accuracy of Proposed method is 0.05 higher than Reference [37], and 0.08 higher than Reference [24]; When the number of iterations is 90, the accuracy of Proposed method is 0.03 higher than Reference [37], and 0.07 higher than Reference [24]; When the number of iterations is 100, the accuracy of Proposed method is the same as Reference [37], 0.04 higher than Reference [24]; It can be seen that this method improves the reliability of the software through the division of embedded multitask software modules [37, 38, 39].

V. CONCLUSIONS

The reliability parameter analysis model of embedded multitask software is established, and the embedded multitasking software module is divided by intelligent analysis method. The embedded multitasking software module partition method based on fuzzy set theory is proposed to construct the constraint parameter distribution model of embedded multitasking software module partition according to the correctness, reliability, running efficiency, integrity and availability of the software. The ambiguity factor of embedded multi-task scheduling and module segmentation is established, the compatibility and human-computer interactions of software are measured by multi-layer index parameter constraint control, the multi-level fuzzy metric structure model of embedded multi-task software module partition is established, and the embedded multi-task software module is divided by quantitative recurrent analysis. It is found that the embedded multitasking software module can be divided effectively by using this method, and the reliability of the software can be improved.

VI. DISCUSSION

In order to solve the problem that the current software module partition method of embedded multitask software is easy to cause software failures and affect user experience, a module partition method of embedded multitask software based on fuzzy set theory is proposed. The basic process of the method is as follows: firstly, the characteristics of embedded multitask software are analyzed, the constraint parameter distribution model is constructed, and the interaction measurement is carried out by combining multitask scheduling and ambiguity segmentation. Then the reliability parameter analysis model of embedded multitask software is constructed, and the multilevel fuzzy metric structure combined with quantitative recursive analysis method is used to complete the division of embedded multitask software modules. In order to verify the application performance of this method in the implementation of embedded multitask software module partition, simulation tests are carried out. The results show that the accuracy of the proposed method is significantly higher than that of the other two methods when the number of iterations is the same. This is because the research builds the optimal scheduling and adaptive control model of embedded multitask software based on fuzzy membership function, which makes embedded multitask software have strong correlation, thus optimizing the reasonable allocation of resources of embedded multitask software. It can be seen that this method improves the reliability of the software through the division of embedded multi task software modules

REFERENCES

[1] Ma, X.T., Luo, J.Q. & Wu, S.L. 2015. Joint sorting and location method using TDOA and multi-parameter of multi-station. *Journal of National University of Defense Technology* 37(6): 78-83.

[2] Rodriguez, A. & Laio, A. 2014. Clustering by fast search and find of density peaks. *Science* 344(6191): 1492.

[3] Zhou, S.B. & Xu, W.X. 2018. A novel clustering algorithm based on relative density and decision graph. *Control and Decision* 33(11): 1921-1930.

[4] Ma, R.F., Yang, L. & Lei, L.F. 2017. Study on cross-platform modularizing embedded multi-task software framework. *Fire Control Radar Technology* 46(04): 46-50.

[5] Liu, Y.C., Zhang, H.X. & Huang, H. 2017. Embedded software framework based on Module-MSG. Computer Engineering and Design 38(7): 1798-1802.

[6] Yan, D.L., Zheng, Y., Wang, W.Y. & Chen, Q.J. 2021. Modeling and dynamic analyses of the bulb turbine blade with crack fault. *Applied Mathematical Modelling* 89: 731-751.

[7] Li, B.F., Tang, Y.D. & Han, Z. 2017. A geometric structure preserving

nonnegative matrix factorization for data representation. *Information and Control* 46(1): 53-59+64.

[8] Wu, Y., Shen, B. & Ling, H. 2014. Visual tracking via online nonnegative matrix factorization. *IEEE Transactions on Circuits and Systems for Video Technology* 24(3): 374-383.

[9] Hu, L.R., Wu, J.G. & Wang, L. 2013. Application and method for linear projective non-negative matrix factorization. *Computer Science* 40(10): 269-273.

[10] Huang, X.J., You, R.Y. & Zhou, C.J. 2013. Study on optical properties of equivalent film constructed of metal nanoparticle arrays. *Journal of Optoelectronics· Laser* 24(7): 1434-1438.

[11] Ye, M., Qian, Y. & Zhou, J. 2015. Multitask sparse nonnegative matrix factorization for joint spectral-spatil hyperspectral imagery denoising. *IEEE Transactions on Geoscience and Remote Sensing* 53(5): 2621-2639.

[12] Zheng, Y.Z. & You, R.Y. 2013. Study on segmented correlation in EEG based on principal component analysis. *Chinese Journal of Biomedical Engineering* 22(3): 93-97.

[13] Sahu, P.K., Manna, K., Shah, N. & Chattopadhyay, S. 2014. Extending Kernighan-Lin partitioning heuristic for application mapping onto Network-on-Chip. *Journal of Systems Architecture* 60(7): 562-578.

[14] Komai, Y., Nguyen, D.H., Hara, T. & Nishio, S. 2014. KNN search utilizing index of the minimum road travel time in time-dependent road networks. *In: Proceedings of the 2014 IEEE 33rd International Symposium on Reliable Distributed Systems Workshops. Piscataway, NJ: IEEE* 1: 131-137.

[15] Ke, S.N., Gong, J. Li, S.N., Zhu, Q., Liu, X.T. & Zhang, Y.T. 2014. A hybrid spatio-temporal data indexing method for trajectory databases. *Sensors* 14(7): 12990-13005.

[16] Ana Carolina, S.M.C., Bertrand, M. & Adiel Teixeira, D.A. 2015. Fuzzy flow sort: An integration of the flow sort method and fuzzy set theory for decision making on the basis of inaccurate quantitative data. *Information Sciences* 293(15): 115-124.

[17] Fan, X.M., Song, L.X. & Ji, D.B. 2017. Research on mechanism of algal blooms based on the critical depth theory. *Environmental Science & Technology* 35(7): 77-81.

[18] Ma, C.L., Shan, H. & Ma, T. 2016. Improved density peaks based clustering algorithm with strategy choosing cluster center automatically. *Computer Science* 43(7): 255-258.

[19] Pei, D. & Li, L. 2019. Simulation research on security defense of electronic document information transmission. *Computer Simulation* 36(2): 146-149+175.

[20] Bhavani, R.R. & Jiji, G.W. 2018. Image registration for varicose ulcer classification using KNN classifier. *International Journal of Computers and Applications* 40(2): 88-97.

[21] Hifi, A.M. 2018. A hybrid reactive search for solving the max-min knapsack problem with multi-scenarios. *International Journal of Computers and Applications* 40(1): 1-13.

[22] Yang, Z.F., Xu, P., Wei, W.F., Gao, G.Q., Zhou, N. & Wu, G.N. 2020. Influence of the crosswind on the pantograph arcing dynamics. IEEE Transactions on Plasma Science 48(8): 2822-2830.

[23] Wang, B., Zou, F.C., Cheng, J. & Zhong, S.M. 2017. Fault detection filter design for continuous-time nonlinear Markovian jump systems with mode-dependent delay and time-varying transition probabilities. Advances in Difference Equations 2017(1): 1-23.

[24] Citil, H.G. 2019. Important notes for a fuzzy boundary value problem. *Applied Mathematics and Nonlinear Sciences* 4(2): 305-314.

[25] Wu, J.H., Zhang, L., Yin, S.F., Wang, H.D., Wang, G.L. & Yuan, J.X. 2018. Differential diagnosis model of hypocellular myelodysplastic syndrome and aplastic anemia based on the medical big data platform. *Complexity* (16): 1-12.

[26] Xiong, Z.G.,Wu, Y., Ye, C.H., Zhang, X.M. & Xu, F. 2019. Color image chaos encryption algorithm combining CRC and nine palace Map. *Multimedia Tools and Applications* 22(78): 31035-55.

[27] Yang, A.M., Li, Y.F., Kong, F.B., Wang, G.Q. & Chen, E.H. 2018. Security control redundancy allocation technology and security keys based on Internet of things. *IEEE ACCESS* 6: 50187-96.

[28] Sarac, Y. & Sener, S.S. 2019. Identification of the initial temperature from the given temperature data at the left end of a rod. *Applied Mathematics and Nonlinear Sciences* 4(2): 469-474.

[29] Shi, K.B., Tang, Y.Y., Liu, X.Z. & Zhong, S.M. 2017. Secondary delay-partition approach on robust performance analysis for uncertain time-varying Lurie nonlinear control system. *Optimal Control Applications and Methods* 38(6): 1208-1226.

[30] Song, J.C.; Zhong, Q.; Wang, W.Z. & Su, C.H. 2020. FPDP: Flexible privacy-preserving data publishing scheme for smart agriculture. *IEEE Sensors Journal* (99): 1-1.

[31] Shi, K.B., Tang, YY., Liu, X.Z. & Zhong, S.M. 2017. Non-fragile sampled-data robust synchronization of uncertain delayed chaotic Lurie systems with randomly occurring controller gain fluctuation. *ISA Transactions* 66: 185-199.

[32] Wang, P.; Chen, C.M.; Kumari, S., Shojafar, M. & Liu, Y. 2020. HDMA: Hybrid D2D message authentication scheme for 5G-enabled VANETs. *IEEE transactions on intelligent transportation systems* 1-10.

[33] Wang, X., Liu, Y. & Choo, K.R. 2020. Fault tolerant multi-subset aggregation scheme for smart grid. *IEEE Transactions on Industrial Informatics* 1-1.

[34] Chen, H., Fang, L., Fan, D.L. Huang, W.J., Huang, J.M.,Cao, C.H., Yang, L., He, Y.B. & Zeng, L. 2019. Particle swarm optimization algorithm with mutation operator for particle filter noise reduction in mechanical fault diagnosis. International Journal of Pattern Recognition and Artificial Intelligence 34(3): 2058012.

[35] Li, X., Zhu, Y. & Wang, J. 2019. Highly efficient privacy preserving location-based services with enhanced one-round blind filter. *IEEE Transactions on Emerging Topics in Computing* (99): 1-1.

[36] Xue, Q., Zhu, Y. & Wang, J. 2019. Joint distribution estimation and naïve bayes classification under local differential privacy. *IEEE Transactions on Emerging Topics in Computing* (99): 1-1.

[37] Chen, H.X., Huang, L., Yang, L.,Chen, Y.T. & Huang, J.M. 2020. Model-based method with nonlinear ultrasonic system identification for mechanical structural health assessment. *Trans Emerging Tel Tech* 31(5): e3955.

[38] Qu, S., Zhao, L. & Xiong, Z. 2020. Cross-layer congestion control of wireless sensor networks based on fuzzy sliding mode control. *Neural Computing and Applications* 32(7): 13505-13520.

[39] Zhu, Q. 2020. Research on road traffic situation awareness system based on image big data. *IEEE Intelligent Systems* 35(1): 18-26.