Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 14, no. 12, December 2018, pp. 3087-3097
DOI: 10.23940/ijpe.18.12.p18.30873097

Software Reliability Demonstration Testing Scheme of Prior Dynamic Integration Bayesian Method based on the Idea of Decreasing Function

Zhenyu Ma[a,b], Wei Wu[c], Wei Zhang[a], Jianping Wang[a], Fusheng Liu[a,*], and Kun Han[c]

[a]*Department of Equipment Support and Remanufacturing, Army Academy of Armored Forces, Beijing, 100072, China*
[b]*Department of Information and Communication, Army Academy of Armored Forces, Beijing, 100072, China*
[c]*Troop No.63963 of PLA, Beijing, 100071, China*

Abstract

Nowadays, aiming at the problem of long testing duration of software reliability demonstration testing, this paper presents a method that can greatly shorten testing duration. First, the idea of a monotone decreasing function is integrated into the Bayesian theory of software reliability demonstration testing, and then a software reliability demonstration testing scheme of a Bayesian method based on monotone decreasing function is proposed. Secondly, according to the real-time prior information in the testing phase, priori information is dynamically integrated. Two kinds of priori dynamic integration methods suitable for two different conditions are proposed. Finally, through experimental analysis, it is proven that the testing scheme can significantly reduce testing duration under the same confidence condition, and the two kinds of prior dynamic integration methods are feasible.

Keywords: Bayesian theory; software reliability demonstration testing; monotone decreasing function; prior information; dynamic integration

(Submitted on September 5, 2018; Revised on October 9, 2018; Accepted on November 11, 2018)

1. Introduction

For software product acceptance, we test whether the software reliability index reaches the standard through software reliability demonstration testing [1]. In general, the higher the confidence coefficient of demonstration results, the better the reliability of software, but the workload of testers is also greatly increased. Especially for some software products with complex technology, the economic cost, labor cost, and time cost of acceptance are unacceptable, making software reliability demonstration testing difficult to complete. However, a Bayesian method can effectively utilize prior information and can significantly reduce the workload of reliability demonstration testing under the condition of ensuring a high confidence coefficient.

At present, experts and scholars have conducted a lot of research on software reliability demonstration testing of the Bayesian method. Tan applied a Bayesian method of constructing a conjugate distribution to verify software reliability [2]. Then, he continued to propose a reliability demonstration testing method based on prior dynamic integration [3]. Liu proposed a weighted Bayesian method based on multi-stage prior information [4]. By constructing a hybrid prior distribution, Zhang gave a reliability demonstration testing scheme under the condition of hybrid prior distribution. [5] Wang put forward a reliability demonstration testing scheme based on the prior distribution method with a decreasing function [6]. Liu proposed a multi-layer Bayesian software reliability demonstration testing method based on a decreasing function [7].

All of these methods are based on ideal and simple conditions [8-10]. For example, supposing the number of estimated parameters is reduced to one in the prior distribution function, the prior distribution function of a multi-layer Bayesian has uniform distribution. Additionally, prior dynamic integration considers the situation of zero-failure demonstration testing and does not consider the situation of non-zero-failure. Therefore, the results of previous studies exhibit one-sidedness and are also not suitable for universal software reliability demonstration testing. Therefore, this article makes further research on

* Corresponding author.
E-mail address: 625181316@qq.com

the basis of previous studies. Firstly, a prior distribution function of two parameters with a monotone decreasing function is constructed. Secondly, a software reliability demonstration testing scheme of a Bayesian method based on the monotone decreasing function is proposed. At the same time, a specific parameters estimation method is given. Thirdly, two kinds of prior dynamic integration methods are proposed to deal with the different failure times in the actual testing process. Finally, through experimental analysis, it is illustrated that the demonstration scheme can reduce the testing duration and the two kinds of prior dynamic integration methods are feasible.

2. Basic Principle of Bayesian Theory of Reliability Demonstration Testing

In the process of reliability demonstration testing [11-12], the failure rate λ is selected as the reliability index, and the conditional distribution density function of the failure times is $f(r|\lambda)$. At the same time, $h(\lambda)$ is the prior distribution density function of λ. Then, the joint distribution density function of the failure times and the failure rate is as follows:

$$g(r,\lambda) = f(r|\lambda)h(\lambda) \tag{1}$$

Using Formula (1), the edge distribution of the failure times is obtained:

$$p(r) = \int g(r,\lambda)\mathrm{d}\lambda = \int f(r|\lambda)h(\lambda)\mathrm{d}\lambda \tag{2}$$

The combination of Formula (1) and Formula (2) is used to further deduce the posterior distribution density function of the failure rate, namely:

$$w(\lambda|r) = \frac{g(r,\lambda)}{p(r)} = \frac{f(r|\lambda)h(\lambda)}{\int f(r|\lambda)h(\lambda)\mathrm{d}\lambda} \tag{3}$$

According to the actual situation, the reliability index (λ_0, r, c) is set up in the reliability demonstration testing scheme. Among them: the failure times is *r*, the failure rate is λ_0, and the confidence coefficient is *c*. We solve the *t* of the minimum value, which is the shortest total testing time for the reliability demonstration testing scheme.

$$P(\lambda \le \lambda_0) = \int_0^{\lambda_0} w(\lambda|r,t)\mathrm{d}\lambda \ge c \tag{4}$$

3. Bayesian Software Reliability Testing Scheme based on the Idea of Monotone Decreasing Function

3.1. Determination of the Prior Distribution and the Posteriori Distribution

In order to construct a prior distribution function with a monotone decreasing function, the kernel of the function is explained. Suppose $f(x)$ is the distribution density function of a random variable. If $f(x) = A'g(x)$, A' is a polynomial that has nothing to do with random variables and $g(x)$ is a polynomial containing all the random variables. Then, $g(x)$ is the kernel of the $f(x)$ function, namely:

$$f(x) \propto g(x) \tag{5}$$

According to the idea of decreasing functions, we regard the failure rate λ as the random variable of a gamma distribution, and we construct the kernel of the distribution density function of a monotone decreasing function, that is:

$$h(\lambda) \propto \lambda^{a-1}e^{-b\lambda} \tag{6}$$

Among them: *a* and *b* are parameters to be estimated, $0 < a \le 1,\ b > 0$.

Using the concept of the kernel of the function, we can get the prior distribution density function of λ, that is:

$$h(\lambda) = A'\lambda^{a-1}e^{-b\lambda} \tag{7}$$

The Bayesian software reliability demonstration testing scheme selects the failure rate of the prior distribution density function, which is to satisfy the gamma distribution. For convenience of computing, $h(\lambda)$ is slightly changed, namely, $A'=Ab^{a-1}$. Then, the prior distribution density function $h(\lambda)$ based on the idea of decreasing functions is changed to:

$$h(\lambda) = A\cdot(b\lambda)^{a-1}e^{-b\lambda} \tag{8}$$

According to the property of the distribution density function, we can get the following:

$$\int_0^\infty h(\lambda)\mathrm{d}\lambda = \int_0^\infty A(b\lambda)^{a-1}e^{-b\lambda}\mathrm{d}\lambda = 1 \tag{9}$$

Then, we can get $A = b/\Gamma(a)$ and deduce the density function of λ prior distribution, that is:

$$h(\lambda) = \frac{b}{\Gamma(a)}(b\lambda)^{a-1}e^{-b\lambda} \tag{10}$$

Among them: $\Gamma(a)$ is a gamma function, and its expression is: $\Gamma(a) = \int_0^{+\infty} s^{a-1}e^{-s}\mathrm{d}s,\ a > 0$.

It is assumed that the probability of failure times ($X = r$) is the conditional probability of λ in the time interval $(0,t]$. Failure times obey the λt Poisson distribution, that is:

$$f(X = r|\lambda) = \frac{(\lambda t)^r}{r!}e^{-\lambda t} \tag{11}$$

Joining Formula (10) and Formula (11), we can get the joint distribution function of X, λ, that is:

$$g(X = r,\lambda) = f(X = r|\lambda)h(\lambda) = \frac{b^a t^r}{r!\Gamma(a)}\lambda^{a+r-1}e^{-\lambda(b+t)} \tag{12}$$

From the upper formula, we can deduce the edge distribution function of *X*, that is:

$$p(X = r) = \int_0^{+\infty} f(X = r,\lambda)h(\lambda)\mathrm{d}\lambda = \frac{ab^a t^r}{r!(b+t)^{a+r}} \tag{13}$$

Joining Formula (12) and Formula (13), the posterior distribution density function λ can be obtained, that is:

$$w(\lambda|r,t,a,b) = \frac{g(X = r,\lambda)}{p(X = r)} = \frac{(b+t)^{a+r}}{\Gamma(a+r)}\lambda^{a+r-1}e^{-\lambda(b+t)} = Gamma(a+r,b+t) \tag{14}$$

3.2. Solving Parameters' Values

In the prior distribution density function $h(\lambda)$, parameters a and b are estimated according to the obtained relevant information in the software reliability testing phase [13-14]. We can determine the estimated values of parameters using the software reliability testing total time *t* and the expectation and variance of failure times.

The software reliability testing total time *t* is assumed to be long enough. We divide time interval $(0,t]$ into *m* time interval sequences $T_1, T_2, \cdots, T_m$. Then, failure times is $k_i = t/T_i$, $(i = 1,2,\cdots,m)$ in each time interval T_i.

The edge distribution function of $p(r)$ corresponding to the first moment:

$$\begin{aligned} E(r) &= \sum_{r=0}^{+\infty} r \cdot p(X=r) = \int_0^{+\infty} h(\lambda) \sum_{r=0}^{+\infty} r \cdot \frac{(\lambda t)^r}{r!} e^{-\lambda t} \mathrm{d}\lambda = \int_0^{+\infty} h(\lambda)\lambda t \mathrm{d}\lambda \\ &= \int_0^{+\infty} \frac{b}{\Gamma(a)} (b\lambda)^{a-1} e^{-b\lambda} \lambda t \mathrm{d}\lambda = \frac{t}{b\Gamma(a)} \int_0^{+\infty} (b\lambda)^a e^{-b\lambda} \mathrm{d}b\lambda = \frac{t}{b} \frac{\Gamma(a+1)}{\Gamma(a)} = \frac{at}{b} \end{aligned} \tag{15}$$

The edge distribution function of $p(r)$ corresponding to the second moment:

$$E(r^2) = \sum_{r=0}^{+\infty} r^2 \bullet p(X=r) = \int_0^{+\infty} h(\lambda) \sum_{r=0}^{+\infty} r^2 \bullet \frac{(\lambda t)^r}{r!} e^{-\lambda t} \mathrm{d}\lambda = \int_0^{+\infty} h(\lambda)(\lambda t + (\lambda t)^2) \mathrm{d}\lambda = \frac{at}{b} + \frac{(a^2+a)t^2}{b^2} \tag{16}$$

$E(r)$ and $E(r^2)$ are recorded as $w_1,\ w_2$. The estimated values of parameters *a*, *b* are:

$$\begin{aligned} a &= -(w_1^2 / w_1^2 + w_1 - w_2) \\ b &= -(w_1 t / w_1^2 + w_1 - w_2) \end{aligned} \tag{17}$$

Among them: $w_1 = \frac{1}{m} \sum_{i=1}^{m} k_i,\ w_2 = \frac{1}{m} \sum_{i=1}^{m} k_i^2$.

3.3. Software Reliability Demonstration Testing Scheme

According to the basic principle of the Bayesian reliability testing method [15], the software reliability demonstration testing scheme of the Bayesian method based on a monotone decreasing function is formulated, namely:

$$P(\lambda \le \lambda_0) = \int_0^{\lambda_0} w(\lambda | r,t,a,b) \mathrm{d}\lambda = \int_0^{\lambda_0} \frac{(b+t)^{a+r}}{\Gamma(a+r)} \lambda^{a+r-1} e^{-\lambda(b+t)} \mathrm{d}\lambda \ge c \tag{18}$$

4. Prior Dynamic Integration

If the software is tested in the specified reliability demonstration testing time [16], the actual total failure times is not greater than the maximum failure times. Then, we stop testing and receive this testing software; otherwise, the next software reliability demonstration testing is required to be continuous. Obviously, the experience of previous testing failure certainly provides historical information for the next reliability demonstration testing. Therefore, we should merge the result information of the last testing failure with prior information of the next software reliability demonstration testing.

4.1. Prior Dynamic Integration Method under the Condition of Error Correction

Generally speaking, for safety critical software, high reliability software, aerospace software, and other types of software, when a defect that causes failure is found during software reliability demonstration testing, it should be misarranged immediately; this is conducive to improve the reliability of the software itself. At the same time, for the high reliability software, the test conditions are more stringent, so we generally choose zero-failure testing. Aiming at the above two points, this paper puts forward a prior dynamic integration method under the condition of error correction.

If the software actual testing length t_1' does not exceed t_1 and $r_1 = r+1$, then the software reliability does not meet the reliability standard. At the moment, we should immediately stop the testing and correct the error, and then continue reliability demonstration testing. Obviously, when the software performs the first reliability demonstration testing, the failure times is greater than the maximum failure times at moment t_1'. The different moment of occurrence gives us different information. Therefore, when performing the next reliability demonstration testing, the result information of the last testing failure needs to be merged with prior information of the next software reliability demonstration testing. Then, software testing duration t_2 should satisfy the following formula *t*:

$$\int_0^{\lambda_0} Gamma(a + r_1 + r, b + t_1' + t)\mathrm{d}\lambda \ge c \tag{19}$$

It can be followed by analogy. When the (i-1)th reliability demonstration testing does not pass, then the corresponding testing length t_i of the ith demonstration testing should satisfy following formula t:

$$\int_0^{\lambda_0} Gamma(a + \sum_{i=1}^{i-1} r_i + r, b + \sum_{i=1}^{i-1} t_i' + t)\mathrm{d}\lambda \ge c \tag{20}$$

Through the above analysis, the following gives a prior dynamic integration method under the condition of error correction. The specific process is as follows:

The first step: According to formulating the software reliability demonstration testing scheme of a Bayesian method based on a monotone decreasing function and reliability index (λ_0, r, c), we use Formula (18) to calculate the testing duration t.

The second step: In the testing duration t, if $r_i \le r$ and the demonstration testing is passed, then the software is received and testing is over; otherwise, proceed to the third step.

The third step: The longest testing duration that the tester can accept is $t_{\max}$, the maximum number of iterations is I, and the number of initial iterations is one, namely $i = 1$.

The fourth step: Let the actual total testing duration time $t_{all} = \sum_{i=1}^{i-1} t_i' + t_i$ and $i = i + 1$. According to Formula (20), we calculate the testing duration t_i under the condition of reliability indicators (λ_0, r, c). If $r_i \le r$, the demonstration testing passes, the software is received, and testing ends; otherwise, judge whether the demonstration testing is over and calculate total actual testing duration t_{all} at the moment. If $t_{all} > t_{\max}$ or $i > I$, the testing ends, the acceptance fails, and the software is rejected. Otherwise, we find the moment t_i' of the actual failure times $r_{ac} = r + 1$ in the testing process and then continue to repeat the fourth step.

4.2. Prior Dynamic Integration Method Without Correction Error

Analogous to the error correction prior dynamic integration method, a reliability demonstration testing that does not consider zero-failure testing is proposed. The method is aimed at software that fails at least once in the actual testing process, that is, $r \ne 0$. The following introduces a prior dynamic integration method without correction error.

The first step: According to formulating a software reliability demonstration testing scheme of a Bayesian method based on a monotone decreasing function and reliability index (λ_0, r, c), we use Formula (18) to calculate the testing duration t.

The second step: In the testing duration t, if $r_i \le r$ and demonstration testing is passed, then the software is received and testing is over; otherwise, proceed to the third step.

The third step: The longest testing duration that the tester can accept is $t_{\max}$, the maximum number of iterations is I, and the number of initial iterations is one, namely $i = 1$.

The fourth step: Let $r^* = (\left\lceil \frac{r_i - r}{r} \right\rceil + 1)r$ (as shown in the formula, r must not be 0, which means that this method is not suitable for zero-failure software reliability demonstration testing). Substituting failure times r^* into Formula (18) during the actual testing process, the total testing duration t^* is calculated under the reliability index (a, b, c). If $t^* > t_{\max}$ or $i > I$, the testing is over, the acceptance is not passed, and the software is rejected; otherwise, we proceed to the fifth step.

The fifth step: We continue software reliability demonstration testing during ($t^* - t$) hours. If failure times is not greater than ($r^* - r_i$), the validation passes, the software is received, and the testing is finished; otherwise, we proceed to the sixth step.

The sixth step: Supposing there are k failures in the (t^*)-t testing duration, then let $r_{i+1} = r_i + k,\ t = t^*$ and $i = i+1$, and then we repeat the fourth step.

4.3. Prior Dynamic Integration Method under Two Conditions

According to the above two kinds of situations, through the combination of the two methods, a prior dynamic integration method under two different conditions is proposed. The flowchart is shown in Figure 1, and detailed steps are as follows:

The first step: According to the software reliability demonstration testing scheme of a Bayesian method based on a monotone decreasing function, we use Formula (18) to calculate the testing duration. In the process of testing, if $r_{ac} \le r$, then software passes reliability demonstration testing; otherwise, we take the second step.

The second step: The longest testing duration that the tester can accept is $t_{\max}$, the maximum number of iterations is I, and the number of initial iterations is one, namely $i = 1$.

The third step: According to the actual requirements of testing, we determine whether the reliability demonstration testing is zero-failure demonstration testing. If there is zero-failure demonstration testing, then the prior dynamic integration method under the condition of error correction is chosen. The specific process refers to Section 4.1; otherwise, proceed to the fourth step.

The fourth step: In the case of non zero-failure times, we can choose either of the two methods. If we choose the prior dynamic integration method without correction error, then the specific steps refer to Section 4.2. Otherwise, refer to Section 4.1.

5. Experimental Analysis

5.1. Experimental Results

In this paper, the time-controlled software that is developed by our laboratory is selected. The prior information of the experiment comes from the special vehicle software evaluation centre. During the process of software reliability growth, the failure interval data of the last 10 sets is obtained and constitutes historical information. Selecting t = 100000 hours, we obtain the empirical sample values t/T_i of failure times at this stage, which constitute prior information data, as shown in Table 1.

Table 1. Prior information data

Failure interval time	Failure times
1136.36	88
649.35	154
7142.86	14
396.83	252
943.40	106
8333.33	12
335.57	298
2173.91	46
16666.67	6
5555.56	18

Through prior information data, combining with Formula (17), we can get the estimated value of the prior distribution parameter of failure rate.

The values of $a = 1,\ b = 1008$ are consistent with the range of parameter values of the prior distribution function based on a monotone decreasing function, then $h(\lambda) = 1008e^{-1008\lambda}$. In the demonstration testing scheme, the confidence coefficient c is 0.99, failure rate λ_0 is 0.001, and failure times r are from 0 to 10. Using Formula (18), we get the total time length of software reliability demonstration testing under the condition of no prior information and prior information with a monotone decreasing function. The results are shown in Table 2.

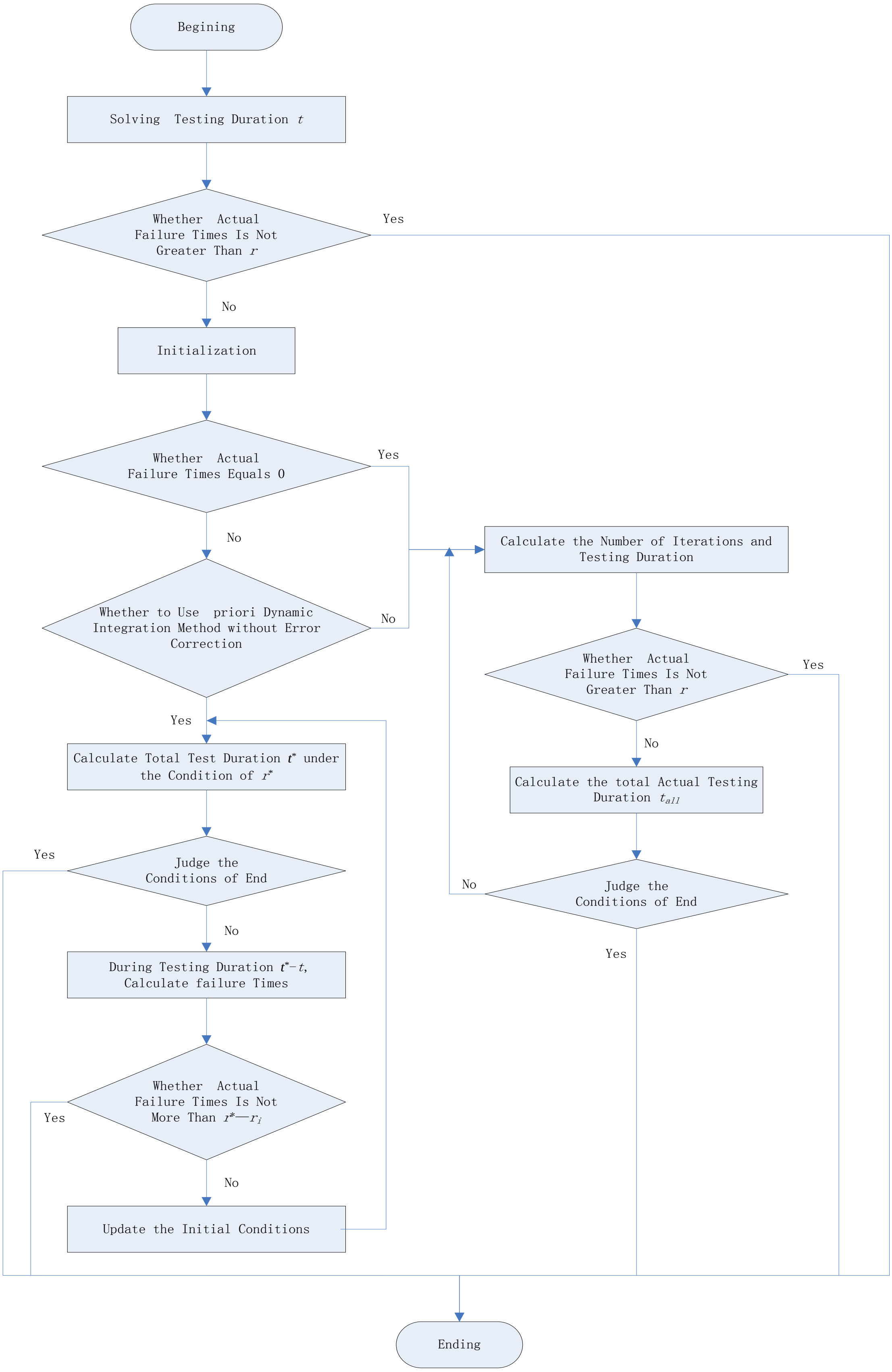

Figure 1. The specific flow chart of prior dynamic integration method

Table 2. Total testing duration based on different prior information

Failure times	No prior information	Prior information with monotone decreasing function
0	4601	3593
1	6633	5625
2	8400	7392
3	10039	9031
4	11598	10590
5	13100	12092
6	14563	13555
7	15992	14984
8	17395	16387
9	18775	17767

According to the prior dynamic integration method under the conditions of no prior information and prior information with a monotone decreasing function and with the conditions of failure times respectively from 0 to 3, the relationship between the actual failure times and the total demonstration testing duration is shown in Tables 3-6 during the software reliability demonstration testing.

Table 3. r = 0, the result diagram of total testing duration

Actual failure times	The total testing duration under the condition of error correction
0	3593
1	5625
2	7392
3	9031
4	10590
5	12092
6	13555
7	14984

Table 4. r = 1, the result diagram of total testing duration

<table>
<tr><th>Actual failure times</th><th>The total testing duration under the condition of error correction</th><th>The total testing duration without correction error</th></tr>
<tr><td>0</td><td colspan="2" rowspan="2">5625</td></tr>
<tr><td>1</td></tr>
<tr><td>2</td><td rowspan="2">9031</td><td>7392</td></tr>
<tr><td>3</td><td>9031</td></tr>
<tr><td>4</td><td rowspan="2">12092</td><td>10590</td></tr>
<tr><td>5</td><td>12092</td></tr>
<tr><td>6</td><td rowspan="2">14984</td><td>13555</td></tr>
<tr><td>7</td><td>14984</td></tr>
</table>

Table 5. r = 2, the result diagram of total testing duration

<table>
<tr><th>Actual failure times</th><th>The total testing duration under the condition of error correction</th><th>The total testing duration without correction error</th></tr>
<tr><td>0</td><td colspan="2" rowspan="3">7392</td></tr>
<tr><td>1</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td><td rowspan="3">12092</td><td rowspan="2">10590</td></tr>
<tr><td>4</td></tr>
<tr><td>5</td><td rowspan="2">13555</td></tr>
<tr><td>6</td><td rowspan="2">16387</td></tr>
<tr><td>7</td><td>16387</td></tr>
</table>

Table 6. r = 3, the result diagram of total testing duration

<table>
<tr><th>Actual failure times</th><th>The total testing duration under the condition of error correction</th><th>The total testing duration without correction error</th></tr>
<tr><td>0</td><td colspan="2" rowspan="4">9031</td></tr>
<tr><td>1</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td></tr>
<tr><td>4</td><td rowspan="4">14984</td><td rowspan="3">13555</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td><td>17767</td></tr>
</table>

5.2. Results Analysis

According to Table 2, the following three related conclusions can be obtained:

1) First of all, with an increase in failure times, the testing duration increases regardless of whether there is prior information or not. Secondly, it can be found that with an increase in failure times, the growth rate of testing duration decreases. For example, in the demonstration testing scheme without prior information, when failure times increase from 0 to 1, the growth rate of testing duration is 44.2%. When failure times increase from 9 to 10, the growth rate of testing duration is 5.3%. In the Bayesian demonstration testing scheme based on the monotone decreasing function, the growth rate of testing duration is 56.6% when failure times increase from 0 to 1. The growth rate of testing duration is 7.7% when failure times increase from 9 to 10.

2) Obviously, in the Bayesian demonstration scheme based on the monotone decreasing function, the testing duration is significantly less than the testing duration without prior information testing scheme. When failure times is 0, the testing duration of the proposed testing scheme is shortened by 28.1% compared with the testing duration without prior information. When the failure number is 10, the testing duration of the proposed testing scheme is shortened by 5.3% compared with the testing duration without prior information.

3) As failure times increases, the testing duration becomes closer and closer between the two schemes. This shows that if software reliability is not high, the software will be very prone to failure in software reliability demonstration testing. If the condition of receiving software corresponding to failure times is too large, no matter which testing scheme is used, this will bring tremendous workload to testers and expend a large amount of time, possibly even making it impossible to complete the software reliability demonstration testing.

According to Tables 3-6, the following three related conclusions can be obtained:

1) Aiming at different failure times, when the actual cumulative failure times is in single digits during the demonstration testing, the testing duration of the prior dynamic integration method without error correction is no longer than the prior dynamic integration method with error correction. For example, when $r=1$ and the actual cumulative failure times is 2, the testing duration of the prior dynamic integration method without error correction is reduced by 1639 hours compared with that of the prior dynamic integration method with error correction. However, when $r=1$ and the actual cumulative failure times is 5, the prior dynamic integration method without error correction has the same testing duration as the prior dynamic integration method with error correction.

When the actual cumulative failure times is too large in the process of demonstration testing, the testing duration of the prior dynamic integration method with error correction is shorter than that of the prior dynamic integration method without error correction. With the continuous increase in actual cumulative failure times, the testing duration of the prior dynamic integration method with error correction is obviously smaller than that of the prior dynamic integration method without error correction. For example, when $r=3$ and the actual cumulative failure times is 7, the testing duration of the prior dynamic integration method with error correction is shortened by 2784 hours compared with that of the prior dynamic integration method without error correction. This shows that when software cumulative failure times frequently increase, if the defect is not corrected, then the prior dynamic integration method without error correction will waste more testing time. The testing duration will even exceed the controllable range of testers, meaning that software reliability demonstration testing cannot be completed. Therefore, in order to avoid such events, failures will be corrected in real time during the testing process, and then the next software reliability demonstration testing will be continued.

2) Analysis of the prior dynamic integration method under the condition of error correction: Because all the demonstration testing results are similar, select the number of failure times as 1 to analyse in the testing scheme. When the software performs the first reliability demonstration testing, the testing duration is 5625 hours. During this period, if software failure times is no more than 1, then the software is accepted, the reliability indicator is up to standard, and the software is received. If the value of failure times is two at $t_1'=3000$ hours ($t_1'<t_1$), then the software acceptance fails and the reliability does not reach the standard. In this case, we need to stop the demonstration testing, immediately correct the software default, and then continue with the second demonstration testing. The second testing duration is $t_2=5625-t_1'=2625$ hours. Similarly, in the first demonstration testing process, if the value of failure times is two at $t_1'=4200$ hours, then the software acceptance fails and the reliability indicators do not meet the standard. Namely, the second testing duration is $t_2=5625-t_1'=1425$ hours. It is obvious that the time point that is not passed through the first

demonstration testing has a direct impact on the duration of the second demonstration testing. If we proceed with the second demonstration testing and the value of failure times is two at the moment t_2', then the software acceptance is not passed and the reliability is not up to the standard. We need to perform the third reliability demonstration testing for software, and the third testing duration is $t_3 = 9031 - t_2' - t_1'$ hours. It can be seen that the third testing duration is related to the previous two time points that are not passed through the demonstration testing, which indicates that the previous two demonstration testing results have an impact on the third testing duration. Therefore, in the course of the next software reliability demonstration testing, all of the preceding demonstration testing results (testing duration) need to be integrated and analysed. The analysis of the following testing results is analogous in turn.

3) Analysis of the prior dynamic integration method without error correction: Because all the demonstration testing results are similar, select the number of failure times as 2 to analyse in the testing scheme. When the software performs the first reliability demonstration testing, the testing duration is 7392 hours. During this period, if software failure times is no more than 2, then the software is accepted, the reliability indicator is up to standard, and the software is received. If $r_1 = 4$, ($r_1 > r$) at the end of the testing, then the software acceptance fails and the reliability is not up to standard. At the moment, we do not need to troubleshoot, but instead follow the second demonstration testing. The second demonstration testing duration is $t = t^* - t = 10590 - 7932 = 3198$ hours. Similarly, if $r_1 = 5$ in the process of the first testing, then the reliability indicator did not meet the standard. Namely, the second testing duration is $t = t^* - t = 13555 - 7392 = 6163$. After the first demonstration testing, it can also be obtained that the actual cumulative failure times have a direct impact on the duration of the second demonstration testing. If we proceed with the second demonstration testing and $r_2 > r$, then the software acceptance is not passed and the reliability is not up to standard. We need to perform the third reliability demonstration testing for the software, and the third testing duration equals this testing duration minus the last testing duration. Therefore, over the course of the next software reliability demonstration testing, all of the preceding demonstration testing results (cumulative failure times) need to be integrated and analysed. The analysis of the following testing results is analogous in turn.

6. Conclusions

In this paper, a reliability demonstration testing scheme based on a Bayesian method is introduced. Then, the prior distribution function with a monotone decreasing function is proposed, and the prior distribution based on a monotone decreasing function and the corresponding posterior distribution are established. Then, the estimation method of parameters is given in prior distribution. On the basis of this, a software reliability demonstration testing scheme of a Bayesian method based on a monotone decreasing function is proposed. Then, prior dynamic integration methods under two different conditions are proposed. One of them is a prior dynamic integration method under the condition of error correction, which mainly integrates prior information of testing duration; the other is a prior dynamic integration method without error correction, which mainly integrates prior information of failure times.

Through experimental analysis, compared with the demonstration testing scheme without prior information, this scheme can significantly reduce demonstration testing duration of software reliability and ensure software reliability demonstration testing results under the same confidence condition. At the same time, the prior dynamic integration method makes comprehensive consideration of the real-time dynamic prior information of testing software during the process of software reliability demonstration testing, which effectively reduces the testing duration of reliability demonstration.

Of course, the software reliability demonstration testing scheme of a Bayesian method based on a monotone decreasing function still needs to be invested in many demonstration testing experiments, so as to continuously improve this method. Then, it will be more efficient to apply this method to software reliability demonstration testing.

References

1. G. Kirbiš, D. Selčan, and I. Kramberger, "Software Reliability Validation and Verification Using Fault Injection Techniques on a Fault Tolerant Processor," *Ifac Papers Online*, Vol. 48, No. 10, pp. 252-257, 2015
2. H. F. Li, C. Liu, and J. Zheng, "Research on Reliability Demonstration Testing of Safety Critical Software," *Aeronautic Standardization & Quality*, Vol. 3, pp. 46-50, 2013
3. Q. Y. Li and H. N. Tong, "Software Reliability Demonstration Testing Plan based on Bayesian Theory," *Advanced Materials Research*, Vol. 658, pp. 513-517, 2013
4. J. F. Liu, S. F. Liu, and Z. G. Fang, "Weighted Bayesian Method of Reliability Evaluation for Binomial Product," *China Mechanical Engineering*, Vol. 24, No. 24, pp. 3371-3374, 2013
5. G. Liu, B. Q. Huang, and C. Liu, "Multiple-Layer Bayesian Discrete Software Reliability Demonstration Testing Scheme with Decreasing Function," *Application Research of Computers*, Vol. 33, No. 3, pp. 761-764, 2017

6. Z. Y. Ma, W. Wu, and W. Zhang, "Analysis of Correlation Coefficient Influence on Software Reliability Demonstration Testing," *Microelectronics & Computer*, 2018
7. Z. D. Qin, H. Lei, and N. Sang, "Reliability Demonstration Testing Method for Continuous Execution Software," *Computer Science*, Vol. 32, No. 6, pp. 202-205, 2005
8. Z. D. Qin, H. Lei, and N. Sang, "Study on the Reliability Demonstration Testing Method for Safety-Critical Software," *Acta Aeronautica ET Astronautica Sinica*, Vol. 26, No. 3, pp. 334-339, 2005
9. W. Wang, "Planning Reliability Demonstration Test with Performance Requirements," in P*roceedings of 2017 Annual Reliability and Maintainability Symposium*, pp. 1-5, 2017
10. X. C. Wang, M. Y. Lu, and F. H. Li, "Continues Software Reliability Demonstration Testing Scheme with Decreasing Function," *Journal of Chongqing University*, Vol. 35, No. 10, pp. 136-143, 2012
11. Z. D. Wu and L. Deng, "Method for Testability Demonstration Test Design based on Entropy of Prior Test Information," *Computer Measurement & Control*, Vol. 24, No. 6, pp. 286-288, 2016
12. Y. M. Wu, H. Li, and Y. Yu, "Design of Safety-Critical Software Reliability Demonstration Test based on Bayesian Theory," in *Proceedings of International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering*, pp. 996-1005, 2013
13. Y. M. Wu, R. Yang, and H. L. Lu, "Bayesian Theory based Software Reliability Demonstration Test Method for Safety Critical Software," *Mathematical Structures in Computer Science*, Vol. 24, No. 5, pp. e240508, 2014
14. L. Yang, "Analysis on Software Reliability Testing Technology," Wireless Internet Technology, 2017
15. L. Yu, W. S. Xu, and S. F. Hou, "A New Hierarchical Testing Model in Software Reliability Testing," in *Proceedings of International Conference on Computer Science and Artificial Intelligence*, pp. 331-338, 2017
16. W. J. Zhang, H. B. Yang, and S. F. Zhang, "Reliability Assessment for Device with Only Safe-or-Failure Pattern based on Bayesian Hybrid Prior Approach," *Acta Armamentarii*, Vol. 37, No. 3, pp. 505-511, 2016