www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 12 Dec. 2016, Page No. 19520-19526

Imperfect Debugging Software Reliability Growth Model for the Rayleigh failure time distribution

Balakrishna, K. Harishchandra
Department of Statistics
Vijaya Evening College, Bengalur,
Karnataka, Indian- 560 004
E-mail: sbk8060@rediffmail.com
Department of Statistics
Bangalore University, Bengalur,
Karnataka, India-560 056
E-mail: harish.jbc@gmail.com

Abstract: Non homogeneous Poisson process software reliability models assume perfect debugging. However in the testing process there is a chance of introducing new faults when detected faults are removed. A model with this concept is called imperfect debugging software reliability model. Imperfect debugging software reliability models proposed in the literature assume a constant or monotonically decreasing fault introduction function. In this article we propose a software reliability model that considers Rayleigh distribution fault content function. The model can capture increasing/decreasing nature of fault introduction rate per fault. The parameters of the model are estimated using maximum likelihood method. A real data are used to verify the model.

Keywords: *Software reliability, imperfect debugging, Rayleigh distribution, Non-homogeneous Poisson process, fault content function, failure occurrence rate per fault, Maximum likelihood estimation.*

1. Introduction

One of the main ingredients of a computer is software. Users may expect to purchase that software which is to run without fail. If there is any fault in the software then he may not recommend the other user to purchase the software. This may leads to huge loss for the companies releasing faulty software. Therefore studying software reliability is important. Software reliability is one of the significant attribute for development computer field. Software reliability is defined as *the probability of failure–free operation of a computer program in a specified environment for a specified period of time* (see Musa and Okumoto, (1982)). Over the last four decades many software reliability growth models have appeared in the literature. Many of these models describe the failure behavior in terms of failure occurrence rate per fault which is assumed to be constant, increasing and decreasing. In this article we propose a software reliability growth model in with imperfect debugging considering the Exponential distribution for mean value function.

During the past numerous software reliability models have been developed by the researchers to provide useful information about how to improve software reliability. For a detailed review of software reliability models see Lyu (1996) , Musa et. al. (1987). Goel and Okumoto (1979), Pham (2006) and Zhang and Pham (2000), have discussed some software reliability models with only fault detection processes with the assumption of perfect and immediate fault correction. Yamada, et.al. (1984) have studied an S-shaped software reliability growth model. Harishchandra and Manjunatha (2010, 2012) have discussed the maximum likelihood estimation of the parameters of a software reliability model including fault detection and correction processes. When a software error occurs but sometimes the experimenter fail to detect this error such software reliability models are known as models with imperfect debugging. Such models have been discussed by Yamada et. al.(1992) and Pham et. al. (1999) and several other researchers.

2. Assumption

The proposed model is based on the following assumptions:

a) The fault detection and removal phenomenon are modeled after the NHPP.
b) The software is subject to failure at any given time because of remaining faults in the software.
c) The fault content function is considers Rayleigh distribution. The fault introduction rate per fault follows an increasing and decreasing trend over time.
d) Each time a failure is observed the failure is removed immediately and a new fault can be introduced.

3. Model development

If a_0 denote the expected number of initial faults a_1 denote the expected number of initially introduced faults and $F(t)$ is fault introduction distribution function, then the fault content function can be written as

$$a(t) = a_0 + a_1 F(t) \tag{1}$$

The fault introduction intensity function is given by

$$\eta(t) = a_1 F'(t) \tag{2}$$

Equation (2) can also be expressed as

$$\eta(t) = a_1 F'(t) = [a_0 + a_1 - a(t)]\frac{F'(t)}{1 - F(t)} = [a_0 + a_1 - a(t)]h(t) \tag{3}$$

where $h(t)$ is the fault introduction rate function of the software. $[a_0 + a_1 - a(t)]$ is the expected number of introduced faults from the remaining faults in the software at the time t. It is monotonically non deceasing function of the time.

Here we fault content function is obtained using Rayleigh distribution and the fault introduction rate per fault is increases then decreases over time. If t is the time between two failures, then the probability density function of the Rayleigh distribution is given in (4)

$$f(t) = 2\theta t e^{-\theta t^2}, t \geq 0, \tag{4}$$

The corresponding distribution function, fault content function and fault introduction intensity function are given by (5)-(7).

$$F(t) = 1 - e^{-\theta t^2} \tag{5}$$

$$a(t) = a_0 + a_1 F(t) = a_0 + a_1\left(1 - e^{-\theta t^2}\right) \tag{6}$$

$$\eta(t) = a_1 F'(t) = 2a_1 \theta t e^{-\theta t^2} \tag{7}$$

In the testing process, a failure phenomenon can be described as an NHPP. If $N(t)$ denote the cumulative number of faults detected by the time t, then $\{N(t), t > 0\}$ is the NHPP. Let $m(t)$ denote the expected cumulative number of faults detected by the time t. It is also called the mean value function. For NHPP the mean value $m(t)$ is given by

$$m(t) = \int_0^t \lambda(x) dx \tag{8}$$

For generalized imperfect debugging fault detection model the mean value function is the solution of the differential equation given in (9)

$$\frac{\delta m(t)}{\delta t} = b(t)\,[a(t) - m(t)] \tag{9}$$

Substituting (6) in (9) and by taking $b(t) = b$ we get

$$\frac{\delta m(t)}{\delta t} = b\left[a_0 + a_1\left(1 - e^{-\theta t^2}\right) - m(t)\right] \tag{10}$$

The solution of the differential equation given in (10) is

$$m(t) = (a_0 + a_1)(1 - e^{-bt}) - ba_1 \int_0^t e^{bs - \theta s^2} ds \tag{11}$$

The integral function given in (11) is evaluated as given in (12)

$$\int_0^t e^{bs-\theta s^2} ds = \frac{\sqrt{\pi} e^{\frac{b^2}{4\theta}} \left[erf\left(\frac{2\theta t - b}{2\sqrt{\theta}}\right) + erf\left(\frac{b}{2\sqrt{\theta}}\right)\right]}{2\sqrt{\theta}} \tag{12}$$

Here $erf(t)$ is called error function which is defined as

$$erf(t) = \int_0^t e^{-s^2} ds = \frac{2}{\sqrt{\pi}} \sum_{k=0}^{\infty} \frac{(-1)^k (t)^{1+2k}}{k!\,(1+2k)} \tag{13}$$

The error function satisfies some conditions which are give in (14)

$$\frac{d}{dx} erf\big(g(x)\big) = \frac{2e^{-[g(x)]^2} g'(x)}{\sqrt{\pi}}$$

$$\int \mathrm{erf}(t)\, dt = terf(t) + \frac{e^{-t^2}}{\sqrt{\pi}} \tag{14}$$

The $erf(t)$ given in (13) is expanded up to 2 terms to get the approximate expression for $m(t)$ which is given in (15)

$$erf(t) = \frac{2}{\sqrt{\pi}}\left[t - \frac{t^3}{3} + o(t)\right] \tag{15}$$

Using (15) in (12) we get

$$\int_0^t e^{bs-\theta s^2} ds = -e^{\frac{b^2}{4\theta}}\left[\frac{\theta}{3}t^3 - \frac{b}{2}t^2 + \left(\frac{b^2}{4\theta} - 1\right)t\right] \tag{16}$$

Now the approximate solution for (10) is obtained by substituting (16) in (11) which is given in (17)

$$m(t) = (a_0 + a_1)(1 - e^{-bt}) + ba_1 e^{\frac{b^2}{4\theta}}\left[\frac{\theta}{3}t^3 - \frac{b}{2}t^2 + \left(\frac{b^2}{4\theta} - 1\right)t\right] \tag{17}$$

The approximate expression for failure intensity function is given by

$$\lambda(t) = (a_0 + a_1)be^{-bt} + ba_1 e^{\frac{b^2}{4\theta}}\left[\theta t^2 - bt + \left(\frac{b^2}{4\theta} - 1\right)\right] \tag{18}$$

The Hazard function is given by

$$h(t) = \frac{\lambda(t)}{[a(t) - m(t)]}$$

$$= \frac{(a_0 + a_1)be^{-bt} + ba_1 e^{\frac{b^2}{4\theta}}\left[\theta t^2 - bt + \left(\frac{b^2}{4\theta} - 1\right)\right]}{(a_0 + a_1)e^{-bt} - a_1 e^{-\theta t^2} - ba_1 e^{\frac{b^2}{4\theta}}\left[\frac{\theta}{3}t^3 - \frac{b}{2}t^2 + \left(\frac{b^2}{4\theta} - 1\right)t\right]} \tag{19}$$

4. Existing models

Several imperfect debugging models have been developed in the literature such as Yamada imperfect debugging model (Yamada et.al 1992), the P-N-Z model (Pham et.al 1999), P-Z IFD model (Pham 2006) and many more. For model comparison purpose we choose the mean value function and failure intensity function of the P-Z imperfect debugging model which are given by

$$m(t) = \frac{(a_0 + a_1)[1 - e^{-bt}]}{1 + \theta e^{-bt}} \tag{20}$$

$$\lambda(t) = \frac{(a_0 + a_1)be^{-bt}[1 + \theta]}{(1 + \theta e^{-bt})^2} \tag{21}$$

5. Parameter estimation for interval domain data

The time domain data are available in terms of time between $(i-1)^{th} and\ i^{th}$ failures t_i or in terms of time of occurrence of i^{th} failure $s_i = \sum_{j=0}^{i} t_j$, $t_0 = 0$. For time domain data the likelihood function and log likelihood function are given by

$$L = \prod_{i=1}^{n} \lambda(s_i) e^{-\int_{s_{i-1}}^{s_i} \lambda(x)} \tag{22}$$

$$lnL = -m(s_n) + \sum_{i=1}^{n} ln[\lambda(s_i)] \tag{23}$$

Substituting the mean value function and the failure intensity function given in (17) and (18) in the log likelihood function given in (23) we get the log likelihood function as

$$lnL = -\left((a_0 + a_1)(1 - e^{-bs_n}) + ba_1 e^{\frac{b^2}{4\theta}}\left[\frac{\theta}{3}s_n^3 - \frac{b}{2}s_n^2 + \left(\frac{b^2}{4\theta} - 1\right)s_n\right]\right) + nlnb + \sum_{i=1}^{n} ln\left[(a_0 + a_1)e^{-bs_i} + a_1 e^{\frac{b^2}{4\theta}}\left[\theta s_i^2 - bs_i + \left(\frac{b^2}{4\theta} - 1\right)\right]\right] \quad (24)$$

Differentiating (24) with respect to $a_0, a_1, b, and\ \theta$ and equating to zero, we get the following system of equations

$$(1 - e^{-bs_n}) = \sum_{i=1}^{n} \frac{e^{-bs_i}}{\left[(a_0 + a_1)e^{-bs_i} + a_1 e^{\frac{b^2}{4\theta}}\left[\theta s_i^2 - bs_i + \left(\frac{b^2}{4\theta} - 1\right)\right]\right]}$$

$$(1 - e^{-bs_n}) + be^{\frac{b^2}{4\theta}}\left[\frac{\theta}{3}s_n^3 - \frac{b}{2}s_n^2 + \left(\frac{b^2}{4\theta} - 1\right)s_n\right] = \sum_{i=1}^{n} \frac{e^{-bs_i} + e^{\frac{b^2}{4\theta}}\left[\theta s_i^2 - bs_i + \left(\frac{b^2}{4\theta} - 1\right)\right]}{\left[(a_0 + a_1)e^{-bs_i} + a_1 e^{\frac{b^2}{4\theta}}\left[\theta s_i^2 - bs_i + \left(\frac{b^2}{4\theta} - 1\right)\right]\right]}$$

$$(a_0 + a_1)s_n e^{-bs_n} + a_1 e^{\frac{b^2}{4\theta}}\left[\frac{\theta s_n^3}{3}\left(\frac{b^2}{2\theta} + 1\right) - \frac{bs_n^2}{2}\left(\frac{b^2}{2\theta} + 2\right) + \frac{b^2 s_n}{4\theta}\left(\frac{b^2}{2\theta} + 3\right)\right] = \frac{1}{b} + \sum_{i=1}^{n} \frac{(a_0 + a_1)s_i e^{-bs_i} + a_1 e^{\frac{b^2}{4\theta}}\left[\frac{b}{2}s_i^2 - \left(\frac{b^2}{2\theta} + 1\right)s_i + \left(\frac{b^3}{8\theta^3}\right)\right]}{\left[(a_0 + a_1)e^{-bs_i} + a_1 e^{\frac{b^2}{4\theta}}\left[\theta s_i^2 - bs_i + \left(\frac{b^2}{4\theta} - 1\right)\right]\right]}$$

$$a_1 be^{\frac{b^2}{4\theta}}\left[\frac{s_n^3}{3}\left(1 - \frac{b^2}{4\theta}\right) + \frac{s_n^2}{8}\left(\frac{b^3}{\theta^2}\right) - \frac{s_n}{16}\left(\frac{b^4}{\theta^3}\right)\right] = \sum_{i=1}^{n} \frac{-\frac{a_1 b^2 e^{\frac{b^2}{4\theta}}}{8\theta^2}\left[bs_i^2 - \left(\frac{b^2}{\theta} + 2\right)s_i + \frac{b(b^2 + 12\theta)}{4\theta^3}\right]}{\left[(a_0 + a_1)e^{-bs_i} + a_1 e^{\frac{b^2}{4\theta}}\left[\theta s_i^2 - bs_i + \left(\frac{b^2}{4\theta} - 1\right)\right]\right]} \quad (25)$$

Here we do not get closed form expressions for MLEs of the parameter. Therefore we use numerical method to obtain MLEs of the parameters.

6. Numerical Analysis

We now present a numerical example for finite failure Software Reliability Growth models based on the actual testing-data. The testing data is time domain data. The time domain data was extracted from information about failures in the development of software for real-time multi computer complex of the US

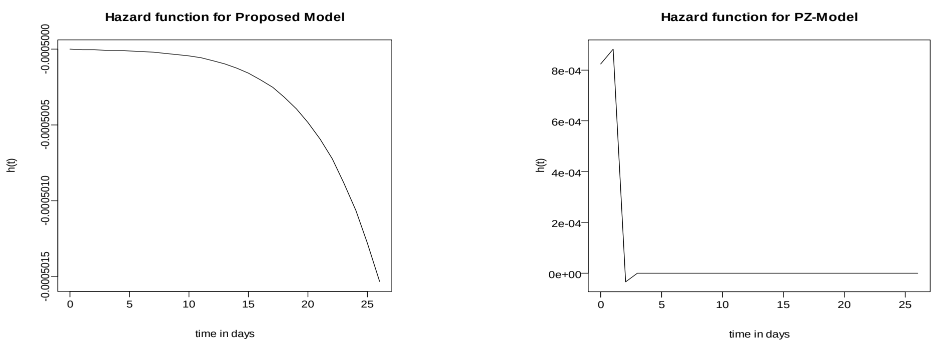
Naval Fleet Computer Center of the US Naval Tactical Data System (NTDS) (Goel and Okumoto [3]). The data consists of 26 software failure occurrence time $\{ s_i(days); i = 1,2,\dots,26\}$ out of 250 days. Here we obtain the estimation results and AIC value for PZ model and our proposed models. The output are summarized in the table 6.1.

Table. 6.1

Model	Estimates of parameters				AIC
	a_0	a_1	b	θ	
Proposed	200	0.8	-0.0005	-0.0015	-159.60
P-Z	200	0.04	-0.004	-5.86	-157.99

Using the above estimates of the parameters, the graph of hazard function of the existing and our proposed model based on log logistic distribution are plotted against the time in days. The graphs are shown below in Figure 6.1.

Figure 6.1



7. Conclusions

In the NHPP it is assumed that the failure rate assumed to follow exponential distribution. In this paper, we propose an imperfect debugging SRGM that considers Rayleigh distribution as fault content function. By using real data set we estimated the parameters of the model by the method of maximum likelihood. We also used AIC to compare our proposed model with P-Z model. The proposed model can better fit software failure data compared with P-Z model. We also plotted the graph of the hazard function for time domain data for the proposed model and P-Z model.

References

DOI: 10.18535/ijecs/v5i12.46

1. Akaike H., "*A new look at statistical model identification*", IEEE Trans. Autom control, 19: 716-723. 1974.
2. Goel, A. L., "*Software Reliability Models: Assumptions, Limitations and Applicability",* IEEE Trans. Software Engineering; SE-11, pp. 1411-1423, 1985.
3. Lyu. M. R., '*Handbook of Software Reliability Engineering*'. McGraw Hill, 1996.
4. Goel, A. L., and Okumoto. K,. "*Time-dependent error detection rate model for software reliability and other performance, Measure"*, IEEE Trans. Reliability R-28(3), pp 206-211, 1979.
5. Harishchandra K, Manjunatha K M. "*Statistical inference on software reliability assuming exponential fault correction time". Economic Quality Control ,* **25**; 269-279. 2010.
6. Harishchandra K and Manjunath K. M ,: '*Modeling and statistical Inference on Generalized Inverse Exponential Software Reliability Growth Model*', Far East Journal of Theoretical Statistics 39(1): 67-77, 2012.
7. Musa J. D, Iannino A, and Okumoto K., '*Software Reliability: Measurement, Prediction, Application'*, McGraw-Hill, Inc., USA, 1987.
8. Pham H , '*System software Reliability*' Springer London, 2006.
9. Pham H, Nordmann L and Zhang K., "*A general imperfect software debugging model with s-shaped fault detection rate'*, IEEE Trans. Reliab. R-48(2):169-175, 1999.
10. Zhang X, Pham H., *'Comparisons of non homogeneous Poisson process software reliability models and its applications',* International Journal of Systems Science, Vol. 31, No. 9, 1115- 1123, 2000.
11. Yamada. S, Ohba. M and Osoki. S., '*S shaped reliability growth modeling for software error detection',* IEEE Trans. Reliability R-32(5), pp. 475-484, 1983.