

# Weighted Isotropic Laplacian Approach for Software Reliability Estimation for a Growth Model

Chandra Mouli Venkata Srinivas Akana<sup>1\*</sup>, C. Divakar<sup>2</sup> and Ch. Satyanarayana<sup>3</sup>

<sup>1</sup>Research Scholar, Dept. of CSE, JNTUK, Kakinada, Andhra Pradesh, India.

<sup>2</sup>Principal, Avanthi's Research and Technological Academy, Vijayanagaram, AP, India.

<sup>3</sup>Dept of CSE, JNTUK, Kakinada., AP., India.

\* Corresponding author e-mail: [mouliac@yahoo.co.in](mailto:mouliac@yahoo.co.in)

Received: 16 Jun. 2014, Revised: 3 Aug. 2014, Accepted: 18 Aug. 2014

Published online: 1 Jan. 2015

---

**Abstract:** *During the development of software applications many defects may be introduced and often leads to sudden breakdowns, so there is a great need to assess this development process in terms of quality and reliability. In this paper a test statistic comprised of both growth and non growth reliability model is tested. The performance is evaluated under a standard test dataset and proves to be more simpler and efficient in estimating the reliability of an growth model.*

**Keywords:** *Software reliability, statistics, growth models*

---

## 1. Introduction

A software reliability growth model (SRGM) can be observed to be a numerical expression which fits in to the experimental data. It may be acquired just by observing the overall tendency of reliability growth. However some of the models can be achieved analytically by making several assumptions about the software debugging and software testing process. Some of these assumptions are merely to remain the analysis biddable. Further are more essential in nature and comprise modeling of the testing and debugging process itself.

When the software applications are been developed many defects and bugs may introduced which often

leads to critical situations and sudden break downs of the systems [1] [2]. software reliability can be evaluated by the number of detected defects or the software failure occurrence time in the testing phase which is the last phase of development process and it can also be estimated using operational statistical approaches.

A software failure can be defined as an unacceptable departure of program operation caused by software fault remaining in the software system. [1] [2] [3]. In reality it is very difficult for the engineers to develop highly reliable software applications because of its complicated requirements. Software reliability models can provide qualitative measures of the reliability of software systems during the development process [4] [5]. The SRGMs have been proven to be successful in

estimating the software reliability and the number of errors remaining in the software and are very useful to assess the reliability for quality control and testing process control of software development [6].the main objective of this paper is to propose statistical analysis to determine whether the patterns of failures is significantly changing with time or not. For this purpose we consider both the HPP (Homogenous Poisson process) and NHPP (Non Homogenous Poisson process).

The rest of the paper is organized as follows, section II describes about the basic system model stressing on the HPP and NHPP ,section III discuss about the proposed Laplacian approach with weighted trend ending with the concerned results and conclusions.

## 2. System Model

Reliability analysis is needed to evaluate the progress of the development process. The hypothesis we consider here is

Let  $H_0$ : HPP and  $H_1$ : NHPP

Where  $H_0$  and  $H_1$  are the null and the alternative hypothesis respectively

So the Laplace trend is as follows

$$U = \frac{L(\theta_0)'}{E(-L(\theta_0))'} \quad (1)$$

Where  $\theta_0$  is a component of the vector  $\theta$  such that its value makes the intensity function  $\lambda(t; \theta)$  time independent

Assuming an error probability  $\alpha = P\{\text{reject } H_0 | H_0 \text{ is true}\}$  then the Laplace trend is as followed

$$U < Z_\alpha \quad \text{Reliability growth}$$

$$U > Z_\alpha \quad \text{Reliability deterioration}$$

$$-Z_\alpha < U < Z_\alpha \quad \text{Stable reliability}$$

Where  $Z_\alpha$  is the upper  $\alpha$  percentage of the standard normal distribution. So we can here by conclude that we can reject  $H_0$  if either  $U > Z_\alpha$  or  $U < -Z_\alpha$  and fail to reject  $H_0$  if  $-Z_\alpha \leq U \leq Z_\alpha$

The main limitation of the Laplacian trend is that it doesn't consider the activity or the presence of the system; in order to overcome this isotropic Laplacian trend factor is defined

$$A(k) = \begin{cases} g(U(k)) & \text{if no activity} \\ U(k) & \text{Otherwise} \end{cases} \quad (2)$$

Where 'g' is the growth reliability stopping function can be of the following things

(i) Green's Function [9]  $g(x) = \frac{\tanh(x)}{2x}$  (if  $x \neq 0$ )

(ii) Gaussian [10]  $g(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)$

(iii) Lorentzian [10]  $g(x) = \frac{1}{1+\frac{x^2}{\sigma^2}}$

The G- function is chosen to satisfy  $g(x) \rightarrow 0$  When  $x \rightarrow \text{inf}$  so that the reliability growth model stops when there is no activity .

## 3. Proposed Method

The main objective of the software reliability test is to determine the pattern of failures which are significantly changing with respect to time. When the occurrence of the events is an NHPP process with a log linear failure intensity function  $\lambda(t) = \exp(\alpha + \beta t)$  then the null hypothesis the Laplace test statistics may be expressed as

$$U(k) = \frac{\sum_{i=1}^k (i-1)n_i - \frac{k-1}{2} \sum_{i=1}^k n_i}{\sqrt{\frac{k^2-1}{12} \sum_{i=1}^k n_i}} \quad (3)$$

a) *Weighted approach*

Let  $\omega \in [0,1]$  be a weight parameter denoting the proportions of software reliability growth during the period  $[t_b, t_l + t_w]$ . Then we can define a weighted failure intensity function as

$$\lambda_w(t) = \lambda(t) + w\lambda(t) + (1-w)\lambda(t_1) \quad (4)$$

Where  $\lambda(t)$  is the failure intensity function

When  $w=1$  the weighted failure intensity function reduces to the original intensity function and when  $w=0$  the function  $\lambda_w$  becomes a constant. More over for

$\omega \in [0,1]$  the weighted failure intensity function  $\lambda_w$  has a less heavier tail that  $\lambda(t)$  in the interval  $[t_b, t_l+t_w]$

Therefore we can define a weighted anisotropic Laplace test statistics as

$$A_w(k) = A(k) + wA(k) + (1 - w)A(t_l) \quad (5)$$

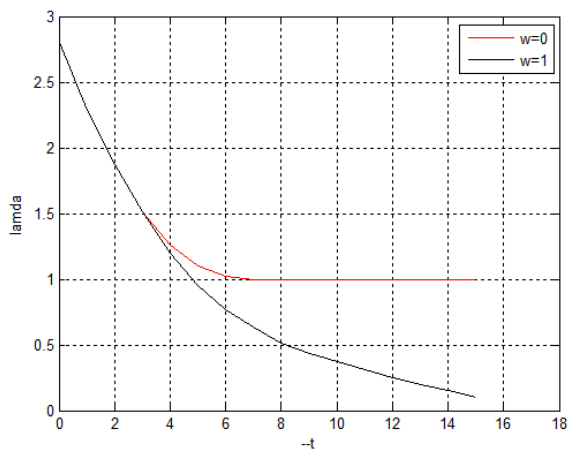


Figure 1: Weighted failure intensity function

#### 4. Experimental Analysis

The test was conducted on a real software failure data which was taken from an SAP development system. The data contains daily software failures that were recorded for a period of 170 days. More over there are no activities in the system during the test phase process on the days 121,122,128,142,143,144,145,147,148,149 and 150.

Below is the graph stating the CDF (cumulative distribution function)

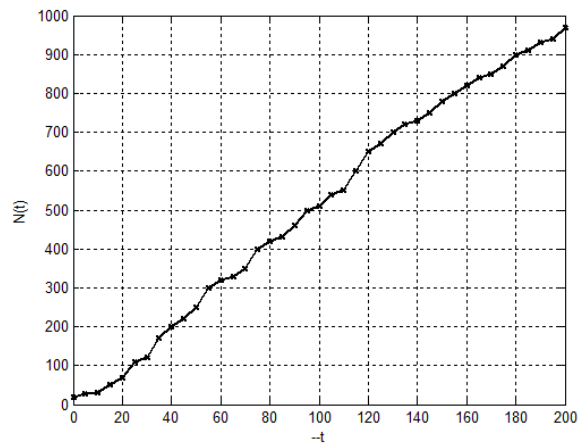


Figure 2: CDF of no. of failures Vs failure time

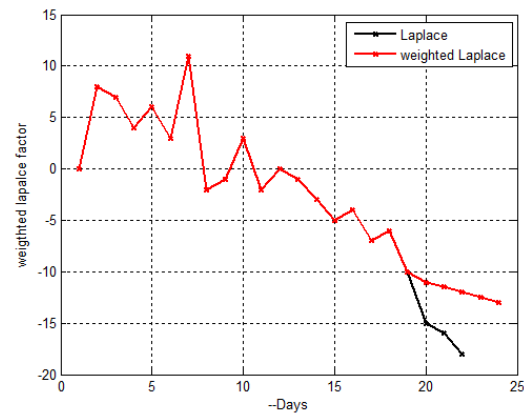


Figure 3: Weighted Laplace trend using Green's [9] function with w=0.5

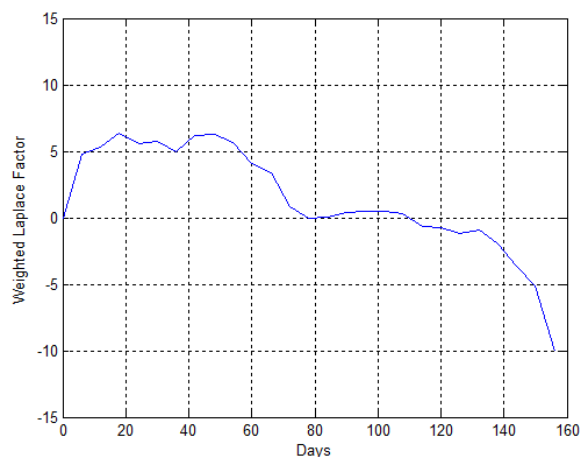


Figure 4: Weighted Laplace trend using Gaussian's [10] function with w=0.5

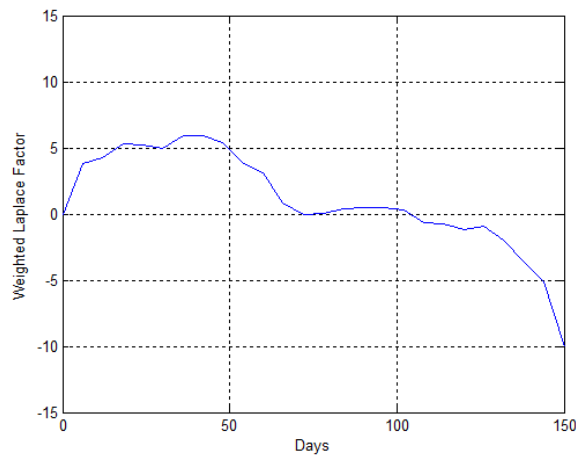


Figure 5: Weighted Laplace trend using Lorentzian [10] function with  $w=0.5$

From the above figures it is clear that the proposed weighted approach under different stopping functions and provides better reliability assessment when compared against the conventional Laplacian approach.

## 5. Conclusion

The proposed model not only takes into account the activity in the system but also the proportion of reliability growth within the provided model. This approach may be termed as the generalized weighted Laplacian approach for NHPP models. The experimental results clearly show that this statically approaches is sufficient for the reliability assessment.

## References

- [1] J.D.Musa. " A theory of software reliability and its applications ", IEEE transactions on software engineering, Vol1, no1, pp: 312-327,1975
- [2]J.D.Musa, A. Iannino and K.okumoto "Software reliability : measurement, prediction application, Mc-Graw hill book company ,1987
- [3]A.L.goel and K.okumoto," Time dependent error detection rate models for software reliability and other performances measures," IEEE transactions on reliability, Vol 28, no 3,pp: 206-211 ,1979

[4]M.R.Bastos Martini,k.Kanoun and J Moreira de souza ," software reliability evaluation of TROPICO-R switching sytem", IEEE transactions on reliability ,Vol39, no3 ,pp: 369-379,1990

[5]K.kanoun and J.C Lapire ," Software reliability trend and analysis from theoretical to practical considerations," IEEE transactions on software engineering ,vol41,no:4mpp:525-532,1992

[6] X.Ahanf ,H.Pham ," software field failure rate prediction before software development ", the journal of systems and software ,vol 79 , pp: 291-300 ,2006

[7] D.R.Cox and P.A.Lewis ," The statistical analysis of a series of events , chapman and Hall London,1978

[8] ) O.Gauodin ," Optimal properties of the laplace trends test for software reliability models ,IEEE transactions on reliability Vol 20 ,no9 , pp: 740-747 ,1992

[9] P.J.Green ," Bayesian reconstruction from emission tomography data using modified EM algorithm ", IEEE transactions on medical imaging ,vol9,no1, pp:84-93 ,1990

[10] P.Perona and J Mallik ," scale space and edge detection using anisotropic diffusion ", IEEE transaction on pattern analysis and machine intelligence ,Vol 12, no 7 pp:629-639 ,1990