Software Reliability Modeling based on Non-homogeneous Poisson Process for Error Occurrence in Each Fault with Periodic Debugging Schedule

TECHNICAL REPORT NO. ASU/2018/1

DATED: 23^{th} JULY, 2018

SUDIPTA DAS

Computer Science Department, Ramakrishna Mission Vivekananda Educational and Research Institute

Belur Math, Howrah 711202, India

e-mail: jusudipta@gmail.com

DAMITRI KUNDU

Indian Statistical Institute

203, B. T. Road, Kolkata 700 108, India

e-mail: damitri.1994@gmail.com

ANUP DEWANJI

Applied Statistics Unit, Indian Statistical Institute

203, B. T. Road, Kolkata 700 108, India

e-mail: dewanjia@isical.ac.in

Applied Statistics Unit

Indian Statistical Institute

Kolkata 700 108

STATISTICAL INDIAN INSTITUTE

भिन्नेष्वैक्यस्य दर्शनम्

UNITY IN DIVERSITY

Software Reliability Modeling based on Non-homogeneous Poisson Process for Error Occurrence in Each Fault with Periodic Debugging Schedule

SUMMARY. In this article, we discuss a continuous time software reliability model under the non-homogeneous Poisson process (NHPP) assumption for error occurrence in each fault to suit periodic debugging, in which errors are not corrected at the instants of their detection but at some pre-specified debugging times. We describe maximum likelihood estimation for the model parameters and provide a computational method to estimate those parameters. This in turn helps to estimate the reliability of the software. We also discuss some asymptotic properties of the estimated model parameters, specially the number of errors initially present in the software. Finally, we investigate the finite sample properties of the estimates under a specific family of NHPP models, specially that of the initial number of errors, through simulation.

KEY WORDS: Continuous time scale, Periodic debugging, NHPP model, failure intensity, Simulation

1 Introduction

Software reliability modeling in continuous time scale was introduced by Jelinski and Moranda [8]. Their model was built on the assumption that the software has a finite (unknown) number of errors ν, say, each with a common and constant failure (detection of an error) intensity. This simple assumption results in a step-wise decreasing failure rate for the software as a whole. Goel and Okumoto [7] and Musa and Okumoto [9] have considered smooth deterministic versions of such step-wise decreasing failure rate for the software, which is essentially a Non-homogeneous Poisson process (NHPP) modeling of the software failure. Goel and Okumoto have used an exponential form, whereas Musa and Okumoto have considered logarithmic functional form. Thereafter, many other software reliability models have been introduced (For example, Yamada [13], Dalal [2], Singpurwalla and Wilson [12] and others), however, keeping one of the major assumptions on fault removal unchanged. In all of these models, it was assumed that the faults are removed as and when they are detected. On the contrary, in many practical in-house testing, debugging of the faults takes place at some prefixed time points. Therefore, the faults are not removed at the instant of their detection; only a record of these detected faults is maintained and these are removed, assumed to be with certainty and introducing no other new fault, at the subsequent scheduled debugging time. This type of design for software debugging and data collection was named as 'periodic debugging schedule', introduced by Dewanji et al. [5]. In addition, this type of design is also essential when subsequent versions of the software are released at various times and testing continues with the most recent version [5]. Due to its importance, Das et. al. [3] demonstrated the analysis of data collected under this new debugging technique, however, based on simple Jelinski-Moranda model [8]. In this simple modeling approach, error occurrence from each of the ν faults is assumed to follow a homogeneous Poisson process (HPP) with a common rate parameter, in the sense that, if there was no debugging, the errors from this particular fault would occur according to a HPP. In this paper, we have studied the periodic debugging schedule, under a generalized model by assuming that the error occurrences caused by each of the ν individual faults follows independently a NHPP with a common time dependent failure intensity involving some unknown parameters, in contrast with the HPP assumption of [3]. We consider maximum likelihood estimation of ν, the parameter of primary interest, and the model parameters associated with the NHPP. Note that, unlike Goel and Okumoto and Musa and Okumoto, this NHPP modeling applies to each of the ν faults. Since detection and and subsequent error occurrences from a particular fault can be treated as a continuous time process, an NHPP model for each fault is a more reasonable assumption, as compared to the homogeneous Poisson process (HPP) model assumed by Jelinksi and Moranda [8], or the NHPP model for the software failure assumed by Goel and Okumoto [7] and Musa and Okumoto [9].

The article is organized as follows. In Section 2, we discuss the continuous time software reliability model under the above-mentioned NHPP assumption and derive the likelihood of the data obtained from the periodic debugging schedule. A computational method for obtaining $\hat{\nu}$, the MLE of ν, is provided in Section 3. In Section 4, we discuss the asymptotic properties of $\hat{\nu}$, as $\nu \to \infty$. Section 5 illustrates with a specific family of NHPP models. Section 6 reports results of a simulation study to investigate the properties of the estimators developed in Section 3. Section 7 ends with some concluding remarks.

2 Modeling and Likelihood

In our model, we assume that there are initially ν mutually independent faults in the software, each being identical with respect to the corresponding error occurrence, following a NHPP. In the periodic debugging schedule, there are k prefixed time points, $0 < t_1 < \cdots < t_k < \infty$, at which debugging is scheduled to take place. Suppose we observe the number of failures m_i (≥ 0) between t_{i-1} and t_i along with their times of occurrences as well as the identities of the corresponding faults, for $i = 1, \cdots, k$, with $t_0 = 0$. Let $\{t_{i_1}, \ldots, t_{i_{m_i}}\}$ be the times of occurrences of the m_i failures in the i^{th} interval $(t_{i-1}, t_i]$. Note that a particular fault may be responsible for more than one failure during the interval $(t_{i-1}, t_i]$; that is, the m_i failures may correspond to fewer than m_i distinct faults. Since we have the identities of the detected faults, let m_i^d (≥ 0) denote the number of distinct faults out of the m_i failures which take place in $(t_{i-1}, t_i]$. These m_i^d faults are removed or debugged at time t_i, with certainty, in a negligible amount of time and no new faults are introduced.

Let us write M_i to denote the cumulative number of distinct faults removed on or before the ith debugging time t_i, for $i = 0, 1, \cdots, k$, with $M_0 = 0$; that is, $M_i = \sum_{l=1}^{i} m_l^d$.

Note that M_k is the number of detected faults by the last debugging time t_k. We assume that each of the ν distinct faults have identical time dependent failure rate $\lambda(t)$ at time t. Therefore, the overall failure rate of the software at a time t during the ith time interval of debugging [i.e., $t_{i-1} < t \leq t_i$] is given by $\lambda_i(t) = (\nu - M_{i-1})\lambda(t)$, for $i = 1, \cdots, k$. Let, H_i be the history of the testing process up to the ith debugging time t_i, for $i = 1, \cdots, k$, so that $H_i = \{(m_l, m_l^d, \{t_{l_1}, \ldots, t_{l_{m_l}}\}),\ l = 1, \cdots, i\}$ and H_0 is empty.

Clearly, since no debugging takes place between 0 and t_1, the occurrence of failures in $(0, t_1]$ follows a Non-homogeneous Poisson process with rate $\nu\lambda(t), 0 < t \leq t_1$. Therefore, the distribution of m_1 is Poisson with expectation $\nu\Lambda(t_1)$, where $\Lambda(t) = \int_0^t \lambda(u)du$. In general, given the history H_{i-1} up to time t_{i-1} including the number M_{i-1} of distinct faults debugged so far, the conditional distribution of m_i is Poisson with expectation $(\nu - M_{i-1})(\Lambda(t_i) - \Lambda(t_{i-1}))$, for $i = 1, \cdots, k$. Therefore,

$$P[m_i|H_{i-1}] \propto e^{-(\nu - M_{i-1})(\Lambda(t_i) - \Lambda(t_{i-1}))} \left[(\nu - M_{i-1})(\Lambda(t_i) - \Lambda(t_{i-1}))\right]^{m_i}. \qquad (1)$$

Using the property of NHPP, the conditional joint distribution of $\{t_{i_1}, \ldots t_{i_{m_i}}\}$, given m_i and H_{i-1}, can be found as

$$P[t_{i_1}, \ldots t_{i_{m_i}}|m_i, H_{i-1}] = \frac{\Pi_{r=1}^{m_i} \lambda(t_{i_r})}{(\Lambda(t_i) - \Lambda(t_{i-1}))^{m_i}}. \qquad (2)$$

It is to be noted that, since the failure rates are identical for all the faults, the conditional probability of m_i^d, given m_i is independent of $\{t_{i_1}, \ldots, t_{i_{m_i}}\}$. To find the conditional probability of m_i^d, given m_i and H_{i-1}, we use Equation (2.4) of Page 102 of [6], regarding derivation of the probability of finding exactly a number of specified cells empty while a fixed number of balls are randomly distributed in a given number of cells. The specified number of cells in that equation is the number of faults remaining after the $(i-1)^{th}$ debugging $(= \nu - M_{i-1})$ in our set up. The fixed number of balls and the given number of empty cells in the equation are the total number of failures observed in the i^{th} debugging interval $(= m_i)$ and the number of faults remaining after the i^{th} debugging $(= \nu - M_i)$, respectively, in our set up. Therefore, the conditional probability of m_i^d, given m_i and H_{i-1}, is given by

$$
\begin{aligned}
P[m_i^d|m_i, H_{i-1}] &= \frac{(\nu - M_{i-1})!}{(\nu - M_i)!m_i^d!}\Sigma_{k=0}^{m_i^d}(-1)^k \binom{m_i^d}{k}\left(\frac{m_i^d - k}{\nu - M_{i-1}}\right)^{m_i} \\
&\propto \frac{(\nu - M_{i-1})!}{(\nu - M_i)!(\nu - M_{i-1})^{m_i}}. \qquad (3)
\end{aligned}
$$

Using (1), (2) and (3), the likelihood function can be written as

$$
\begin{aligned}
L(\nu, \underline{\theta}) &= \prod_{i=1}^{k} P[m_i, \{t_{i_1}, \ldots, t_{i_{m_i}}\}, m_i^d|H_{i-1}] \\
&= \prod_{i=1}^{k} P[m_i|H_{i-1}]P[t_{i_1}, \ldots t_{i_{m_i}}|m_i, H_{i-1}]P[m_i^d|m_i, H_{i-1}] \\
&\propto \prod_{i=1}^{k} \Big[e^{-(\nu - M_{i-1})(\Lambda(\underline{\theta}, t_i) - \Lambda(\underline{\theta}, t_{i-1}))}[(\nu - M_{i-1})(\Lambda(\underline{\theta}, t_i) - \Lambda(\underline{\theta}, t_{i-1}))]^{m_i} \\
&\quad \times \frac{\Pi_{r=1}^{m_i}\lambda(\underline{\theta}, t_{i_r})}{(\Lambda(\underline{\theta}, t_i) - \Lambda(\underline{\theta}, t_{i-1}))^{m_i}} \times \frac{(\nu - M_{i-1})!}{(\nu - M_i)!(\nu - M_{i-1})^{m_i}}\Big] \\
&= \prod_{i=1}^{k} \left[e^{-(\nu - M_{i-1})(\Lambda(\underline{\theta}, t_i) - \Lambda(\underline{\theta}, t_{i-1}))} \times \Pi_{r=1}^{m_i}\lambda(\underline{\theta}, t_{i_r}) \times \frac{(\nu - M_{i-1})!}{(\nu - M_i)!}\right] \\
&= \frac{\nu!}{(\nu - M_k)!}\prod_{i=1}^{k} \left[e^{-(\nu - M_{i-1})(\Lambda(\underline{\theta}, t_i) - \Lambda(\underline{\theta}, t_{i-1}))} \times \Pi_{r=1}^{m_i}\lambda(\underline{\theta}, t_{i_r})\right], \qquad (4)
\end{aligned}
$$

where $\underline{\theta}$, denotes the set of parameters associated with the cumulative failure intensity $\Lambda(\cdot)$. We write $\Lambda(t)$ as $\Lambda(\underline{\theta}, t)$ to indicate the dependence of $\Lambda(t)$ on $\underline{\theta}$ explicit. Note that this likelihood reduces to that of Das et. al. [3] (See Equation 3 for the special case of HPP model, that is $\lambda(t) = \lambda$ a constant).

3 Maximum Likelihood Estimation

One can obtain the maximum likelihood estimate (MLE) of the model parameters using the likelihood (4). Recall that ν is the primary parameter of interest while $\underline{\theta}$ may be treated as nuisance. We present a profile likelihood method to obtain the MLE of ν.

For a fixed $\underline{\theta}$, using (4), let us consider

$$
\frac{L(\nu + 1, \underline{\theta})}{L(\nu, \underline{\theta})} = \left(1 + \frac{M_k}{\nu + 1 - M_k}\right) e^{-\Lambda(\underline{\theta}, t_k)}. \qquad (5)
$$

It can be seen that, as ν goes from M_k to ∞, the ratio in (5) is greater than one, when ν is in $\left[M_k, \frac{M_k}{1-e^{-\Lambda(\underline{\theta},t_k)}} - 1\right)$, and it is less than one and goes to zero, monotonically, afterwards. Therefore, for a fixed $\underline{\theta}$, the likelihood $L(\nu, \underline{\theta})$ is maximized at $\hat{\nu}(\underline{\theta}) = \lceil \frac{M_k}{1-e^{-\Lambda(\underline{\theta},t_k)}} \rceil$, where $\lceil x \rceil$, is the smallest integer greater than x. Now, put $\nu = \hat{\nu}(\underline{\theta})$ in the log-likelihood function of the likelihood function in (4), i.e.

$$\begin{aligned} l(\hat{\nu}(\underline{\theta}), \underline{\theta}) &= \log\left(\frac{\hat{\nu}(\underline{\theta})!}{(\hat{\nu}(\underline{\theta}) - M_k)!}\right) + \sum_{i=1}^{k} [-(\hat{\nu}(\underline{\theta}) - M_{i-1})(\Lambda(\underline{\theta}, t_i) - \Lambda(\underline{\theta}, t_{i-1}))] \\ &\quad + \sum_{i=1}^{k} \sum_{r=1}^{m_i} \log \lambda(\underline{\theta}, t_{i_r}). \end{aligned} \tag{6}$$

Thereafter, one can maximize (6) over $\underline{\theta}$ to obtain the MLE $\hat{\underline{\theta}}$ and then $\hat{\nu}$ as $\lceil M_k/(1 - e^{-\Lambda(\hat{\underline{\theta}},t_k)}) \rceil$.

4 Asymptotic Results

In order to derive the asymptotic properties of the MLEs, $\hat{\nu}$ and $\hat{\underline{\theta}}$ of ν and $\underline{\theta}$, respectively, obtained by the method of Section 3, following Dewanji et al. [4], let the ν faults be labeled as $1, \cdots, \nu$ and let X_j denote the (unknown) observation from the jth fault. If the jth fault is not detected, let us write $X_j = 0$; otherwise, X_j consists of the debugging time t_{i_j} (say) of the jth fault and the number m_j^* (say) of times it appears in $[t_{i_j-1}, t_{i_j})$ along with the times of appearances $\{t_{i_{j_1}} < t_{i_{j_2}} < \ldots < t_{i_{j_{m_j^*}}}\}$ (say). The X_j's are clearly hypothetical since the labeling $1, \cdots, \nu$ is not observed; however, the X_j's are independent and identically distributed with the probability distribution given by

$$p_X(x_j; \underline{\theta}) = \begin{cases} e^{-\Lambda(\underline{\theta},t_k)}, & \text{if } x_j = 0 \\ e^{-\Lambda(\underline{\theta},t_{i_j})} \prod_{s=1}^{m_j^*} \lambda(\underline{\theta}, t_{i_{j_s}}), & \text{otherwise.} \end{cases}$$

The joint distribution of $(X_1, \cdots, X_\nu)$ is, therefore, $L^*(\nu, \underline{\theta}) = \prod_{j=1}^{\nu} p_X(x_j; \underline{\theta})$. Note that the observed data is a function of the unobserved X_j's. Hence, the observed likelihood (4) can be obtained from the joint distribution of the X_j's and is given by

$$\frac{\nu!}{(\nu - M_k)!} \prod_{j=1}^{\nu} p_X(x_j; \underline{\theta}).$$

For a finite dimensional $\underline{\theta} = [\theta_1, \ldots, \theta_n]^T$, similar to the technique used in Dewanji et al.[4], we define $\underline{V_j} = [V_{1j}, V_{2j}, \ldots, V_{n+1,j}]^T$ where

$$V_{1j} = \begin{cases} 1, & \text{if } x_j \neq 0 \\ -\frac{1-e^{-\Lambda(\underline{\theta},t_k)}}{e^{-\Lambda(\underline{\theta},t_k)}}, & \text{otherwise,} \end{cases} ;$$

and

$$V_{h+1,j} = \frac{d}{d\theta_h} \log\ p_X(x_j; \underline{\theta})$$

for $h = 1, \ldots, n$ and $j = 1, \ldots, \nu$.

It can be checked that $E[\underline{V_j}] = \underline{0}$ and their variances are as follows

$$Var[V_{1j}] = e^{\Lambda(\underline{\theta},t_k)} - 1$$

and

$$Var[V_{h+1,j}] = E\left[-\frac{d^2}{d\theta_h^2}\ \log\ p_X(x_j; \underline{\theta})\right] = I_{h+1,h+1}(\underline{\theta}),\ \text{say,}$$

for $h = 1, \ldots, n$ and $j = 1, \ldots, \nu$. It can also be checked that

$$Cov[V_{1j}, V_{h+1,j}] = \frac{d}{d\theta_h}\Lambda(\underline{\theta}, t_k) = I_{1,h+1}(\underline{\theta}),\ \text{say,}$$

and

$$Cov[V_{g+1,j}, V_{h+1,j}] = E\left[-\frac{d^2}{d\theta_g d\theta_h} \log\ p_X(x_j; \underline{\theta})\right] = I_{g+1,h+1}(\underline{\theta}),\ \text{say,}$$

for $g, h = 1, \ldots, n$ and $j = 1, \ldots, \nu$. Therefore, writing

$$u_{h,\nu} = \nu^{-1/2} \sum_{j=1}^{\nu} V_{h,j},\ \text{for } h = 1, \ldots, n+1$$

we have, by the central limit theorem,

$$\underline{u}_\nu = \begin{pmatrix} u_{1\nu} \\ \vdots \\ u_{n+1,\nu} \end{pmatrix} = \nu^{-1/2} \sum_{j=1}^{\nu} \begin{pmatrix} V_{1j} \\ \ldots \\ V_{n+1,j} \end{pmatrix} \xrightarrow{L} N(\underline{0}, \Sigma^{-1}) \text{ as } \nu \to \infty,$$

where

$$\Sigma = \begin{pmatrix} e^{\Lambda(\underline{\theta},t_k)} - 1 & I_{12}(\underline{\theta}) & \dots & I_{1,n+1}(\underline{\theta}) \\ I_{12}(\underline{\theta}) & I_{22}(\underline{\theta}) & \dots & I_{2,n+1}(\underline{\theta}) \\ \vdots & \vdots & \ddots & \vdots \\ I_{1,n+1}(\underline{\theta}) & I_{2,n+1}(\underline{\theta}) & \dots & I_{n+1,n+1}(\underline{\theta}) \end{pmatrix}^{-1}.$$

For bounded $(a_1, \underline{a})$, where $\underline{a} = [a_2, \dots, a_{n+1}]^T$, following the technique of Dewanji et al. [4], we consider

$$l(\nu + \nu^{1/2}a_1, \underline{\theta} + \nu^{-1/2}\underline{a}) - l(\nu, \underline{\theta}),$$

which can be reduced to

$$\sum_{h=1}^{n+1} a_h u_{h,\nu} - \sum_{h=2}^{n+1}\sum_{g=1}^{h-1} a_g a_h I_{h,g}(\underline{\theta}) - \frac{a_1^2}{2}\left(e^{\Lambda(\underline{\theta},t_k)} - 1\right) - \sum_{h=2}^{n+1} \frac{a_h^2}{2} I_{h,h}(\underline{\theta}) + o_p(1).$$

Then, using the argument of Sen and Singer ([11], p207), as in Dewanji et al. [4], we have the following result.

Result 1: As $\nu \to \infty$,

$$[\nu^{-1/2}(\hat{\nu} - \nu), \nu^{1/2}(\hat{\underline{\theta}} - \underline{\theta})] \xrightarrow{L} N(0, \Sigma).$$

The covariance matrix Σ can be consistently estimated by

$$\hat{\Sigma} = \begin{pmatrix} e^{\Lambda(\hat{\underline{\theta}},t_k)} - 1 & \frac{d}{d\underline{\theta}^T}\Lambda(\hat{\underline{\theta}}, t_k) \\ \frac{d}{d\underline{\theta}}\Lambda(\hat{\underline{\theta}}, t_k) & -\hat{\nu}^{-1}\frac{d^2}{d\underline{\theta}d\underline{\theta}^T}\log L(\hat{\nu}, \hat{\underline{\theta}}) \end{pmatrix}^{-1}.$$

In particular, the asymptotic variance of $\hat{\nu}$ can be consistently estimated by

$$\left[e^{\Lambda(\hat{\underline{\theta}},t_k)} - 1 - \hat{\nu}\frac{d}{d\underline{\theta}^T}\Lambda(\hat{\underline{\theta}}, t_k)\left(\frac{d^2}{d\underline{\theta}d\underline{\theta}^T}\log L(\hat{\nu}, \hat{\underline{\theta}})\right)^{-1}\frac{d}{d\underline{\theta}}\Lambda(\hat{\underline{\theta}}, t_k)\right]^{-1}. \quad (7)$$

5 Illustration with a specific family

For the purpose of illustrating the method, we consider a family of NHPP models in which the cumulative failure intensity, $\Lambda(\underline{\theta}, t)$, is of the form $\Lambda((\alpha, \beta), t) = \alpha w(\beta, t)$, with $\underline{\theta} = (\alpha, \beta)$, where $\alpha > 0$ and β are unknown parameters and $w(\beta, t)$ is a differentiable function of time t involving β. This family is quite general including, for example, the HPP model and also the Goel and Okumoto; and Musa and Okumoto models (See Section 1) for failures due to a single fault. For this family, the likelihood function (4) reduces to

$$L(\nu, \alpha, \beta) = \frac{\nu!}{(\nu - M_k)!} \prod_{i=1}^{k} \left[e^{-(\nu - M_{i-1})(\alpha w(\beta, t_i) - \alpha w(\beta, t_{i-1}))} \times \Pi_{r=1}^{m_i} \alpha w_t(\beta, t_{i_r}) \right], \tag{8}$$

where $w_t(\beta, t)$ is the first derivative of $w(\beta, t)$ with respect to t. Therefore, the MLE of ν becomes $\hat{\nu} = \lceil \frac{M_k}{1 - e^{-\hat{\alpha} w(\hat{\beta}, t_k)}} \rceil$, where the MLE $(\hat{\alpha}, \hat{\beta})$ can be obtained by maximizing (See (6))

$$\begin{aligned} l(\alpha, \beta) &= \log\left(\frac{\hat{\nu}(\alpha, \beta)!}{(\hat{\nu}(\alpha, \beta) - M_k)!}\right) + \sum_{i=1}^{k} [-\alpha(\hat{\nu}(\alpha, \beta) - M_{i-1})(w(\beta, t_i) - w(\beta, t_{i-1}))] \\ &\quad + m \log \alpha + \sum_{i=1}^{k} \sum_{r=1}^{m_i} \log w_t(\beta, t_{i_r}), \end{aligned} \tag{9}$$

over (α, β), where $\hat{\nu}(\alpha, \beta) = \lceil M_k / (1 - e^{-\alpha w(\beta, t_k)}) \rceil$ (See Section 3).

It is to be noted that, for this family of models, ν can also be estimated by the following alternative way. We consider $M_k \geq 1$, otherwise the MLE of ν is zero. The log-likelihood function corresponding to (8) is given by

$$\begin{aligned} l(\nu, \alpha, \beta) &= \log\left(\frac{\nu!}{(\nu - M_k)!}\right) + \sum_{i=1}^{k} [-\alpha(\nu - M_{i-1})(w(\beta, t_i) - w(\beta, t_{i-1}))] \\ &\quad + m \log \alpha + \sum_{i=1}^{k} \sum_{r=1}^{m_i} \log w_t(\beta, t_{i_r}), \end{aligned} \tag{10}$$

which gives the MLE of α, for a given ν and β, as

$$\hat{\alpha}(\nu, \beta) = \frac{m}{\sum_{i=1}^{k} (\nu - M_{i-1})(w(\beta, t_i) - w(\beta, t_{i-1}))}. \tag{11}$$

Therefore, substituting α by $\hat{\alpha}(\nu, \beta)$ in (10), we get the profile likelihood

$$\begin{aligned} l(\nu, \beta) &= \log\left(\frac{\nu!}{(\nu - M_k)!}\right) - m \log\left[\sum_{i=1}^{k} (\nu - M_{i-1})(w(\beta, t_i) - w(\beta, t_{i-1}))\right] \\ &\quad - m + m \log m + \sum_{i=1}^{k} \sum_{r=1}^{m_i} \log w_t(\beta, t_{i_r}), \end{aligned} \tag{12}$$

which is maximized over ν and β in the following way. First we fix $\nu = M_k$ and find β which maximizes $l(M_k, \beta)$. Denote it by β_ν. Then we move to $\nu = M_k + 1$ and repeat the above procedure to find the corresponding β maximizing $l(M_k + 1, \beta)$. We repeat this step up to a large value of ν and note all the pairs of (ν, β_ν) with the corresponding values of $l(\nu, \beta_\nu)$. Finally, by comparing the values of $l(\nu, \beta_\nu)$ at each pair (ν, β_ν), we choose the pair $(\hat{\nu}, \hat{\beta})$, say, which gives the largest value of $l(\nu, \beta_\nu)$, over the choices of ν, denoted by $l(\hat{\nu}, \hat{\beta})$.

The asymptotic variance of $\hat{\nu}$ can be derived by using Result 1 of Section 4. In this case, the matrix, Σ is estimated as

$$\hat{\Sigma} = \begin{pmatrix} e^{\hat{\alpha} w(\hat{\beta}, t_k)} - 1 & w(\hat{\beta}, t_k) & \hat{\alpha} w_\beta(\hat{\beta}, t_k) \\ w(\hat{\beta}, t_k) & \frac{m}{\hat{\nu}\hat{\alpha}^2} & I_{23}(\hat{\alpha}, \hat{\beta}) \\ \hat{\alpha} w_\beta(\hat{\beta}, t_k) & I_{23}(\hat{\alpha}, \hat{\beta}) & I_{33}(\hat{\alpha}, \hat{\beta}) \end{pmatrix}^{-1},$$

where

$$\begin{aligned} I_{33}(\hat{\alpha}, \hat{\beta}) &= (1 - \frac{M_k}{\hat{\nu}}) \hat{\alpha} w_{\beta^2}(\hat{\beta}, t_k) + \frac{\hat{\alpha}}{\hat{\nu}} \sum_{i=1}^{k} m_i^d w_{\beta^2}(\hat{\beta}, t_i) \\ &\quad + \frac{1}{\hat{\nu}} \sum_{i=1}^{k} \sum_{j=1}^{m_i} \frac{w_{\beta^2,t}(\hat{\beta}, t_{i_j}) w_t(\hat{\beta}, t_{i_j}) - (w_{\beta,t}(\hat{\beta}, t_{i_j}))^2}{(w_t(\hat{\beta}, t_{i_j}))^2}, \end{aligned}$$

$$I_{23}(\hat{\alpha}, \hat{\beta}) = (1 - \frac{M_k}{\hat{\nu}}) w_\beta(\hat{\beta}, t_k) + \frac{1}{\hat{\nu}} \sum_{i=1}^{k} m_i^d w_\beta(\hat{\beta}, t_i)$$

and $w_\beta(\beta, t), w_{\beta^2}(\beta, t), w_{\beta,t}(\beta, t)$ and $w_{\beta^2,t}(\beta, t)$ are the derivatives of $w(\beta, t)$ of the respective order and with respect to β and/or t, as applicable. In particular, the variance of $\hat{\nu}$ can be consistently estimated by (See (7))

$$\hat{\nu} \left[\frac{m}{\hat{\nu}\hat{\alpha}^2} I_{33}(\hat{\alpha}, \hat{\beta}) - I_{23}^2(\hat{\alpha}, \hat{\beta}) \right] |\hat{\Sigma}|^{-1}. \tag{13}$$

6 A Simulation Study

Here, we report results of some simulation study assessing the performance of the MLEs of ν, α and β under the specific family of NHPP models, as discussed in Section 5.

The general simulation setup is as follows. We consider three values of ν, namely, $\nu = 100, 500$ and 1000, with $t_k = 10$, while time between consecutive debugging is 1, so that the number of debugging time points k is equal to 10. We consider three types of NHPP processes, namely, the power law process, Musa and Okomkuto process and Goel and Okumoto process having a common cumulative functional form as $\Lambda(t) = \alpha w(\beta, t)$. For the power law process, having $\Lambda(t) = \alpha t^{\beta}$, we take three choices of the pair (α, β) as $(0.1, 0.2303), (0.1, 0.1609)$ and $(0.1, 0.0916)$, resulting in $\bar{F}(t_k, \alpha, \beta) = 0.1$, 0.2 and 0.3, respectively, reflecting different extent of non-detection (or, missingness). As a result, we have got $3 \times 3 = 9$ different parameter configurations. For each of the 9 configurations, we generate 5000 data sets and obtain the MLEs $\hat{\nu}, \hat{\alpha}$ and $\hat{\beta}$ along with the corresponding standard errors (See Sections 4 and 5) for each data set. As ν is the parameter of primary interest, we report the simulation results regarding the performance of $\hat{\nu}$ in Table 1A.

We take the averages of $\hat{\nu}$, and the corresponding standard error $s(\hat{\nu})$ over the 5000 simulations and denote them by $\bar{\hat{\nu}}$ and $\overline{s(\hat{\nu})}$, respectively . In addition, we also obtain the sample standard error of $\hat{\nu}$, denoted by $sse(\hat{\nu})$, defined as the sample standard deviation of 5000 estimates of ν. The estimated coverage probability, denoted by CP, is computed as the proportion of times (out of 5000 simulations) the asymptotic 95% confidence interval, obtained through the normal approximation of $\hat{\nu}$ (see Result 1) and using (13), contains the true ν. For the purpose of comparison, we also provide relative bias and relative standard error defined as $(\bar{\hat{\nu}} - \nu)/\nu$ and $\overline{s(\hat{\nu})}/\nu$, respectively.

Table 1A: Simulation results of Power Law process on the ML estimator of ν with corresponding standard errors and estimated coverage probability for $t_k = 10$.

ν	$\bar{F}(t_k, \alpha, \beta)$	$\bar{\hat{\nu}}$	$\frac{(\bar{\hat{\nu}}-\nu)}{\nu}$	$\overline{s(\hat{\nu})}$	$sse(\hat{\nu})$	$\frac{\overline{s(\hat{\nu})}}{\nu}$	CP
100	0.1	99.79	-0.0021	6.29	6.81	.06	.86
	0.2	100.88	0.0088	13.01	11.54	.13	.86
	0.3	98.93	-0.0107	22.26	18.02	.22	.85
500	0.1	498.90	-0.0022	14.13	13.70	.03	.94
	0.2	498.33	-0.0033	28.07	27.23	.06	.92
	0.3	501.72	0.0034	46.70	47.68	.09	.91
1000	0.1	999.20	-0.0008	20.21	20.76	.02	.95
	0.2	997.54	-0.0025	39.71	39.47	.04	.94
	0.3	1002.72	0.0027	65.33	65.29	.07	.93

Table 1A shows that the sample standard error under $sse(\hat{\nu})$ and the average standard error under $\overline{s(\hat{\nu})}$ are similar specially for small $\bar{F}(t_k, \alpha, \beta)$ and large ν. This testifies the convergence to the asymptotic variance of $\hat{\nu}$ in (13). The estimator $\hat{\nu}$ appears to perform better with respect to relative bias, relative standard error and CP with increasing ν and decreasing $\bar{F}(t_k, \alpha, \beta)$. The ML estimates are almost unbiased in all cases. Also, the CP values are close to 0.95, particularly for large ν, indicating convergence to normality as given in Result 1. The results on the MLEs $\hat{\alpha}$ and $\hat{\beta}$ are qualitatively similar (not reported here).

We have also investigated the performances of the estimators, under Musa and Okumoto and Goel and Okumoto processes. In both of these cases, the simulation setup remains same as the one used under the power law process, except the choices of parameters (α, β). For Musa and Okumoto process, with $\Lambda(t) = \alpha \log(1 + \beta t)$, the pair (α, β) are varied as (1,0.9), (1,0.4) and (1,0.23) to keep $\bar{F}(t_k, \alpha) = 0.1$, 0.2 and 0.3, respectively, as before. Similarly, for Goel and Okumoto process, with $\Lambda(t) = \alpha(1 - e^{-\beta t})$, values of the pair (α, β) are varied as (10,0.026), (10,0.018) and (10,0.013). The results of the simulation study under Musa and Okumoto and Goel and Okumoto processes are reported in Table 1B and Table 1C, respectively. From these tables we observe that the estimator $\hat{\nu}$ seems to perform better with respect to relative bias, relative standard error and CP with increasing ν and decreasing $\bar{F}(t_k, \alpha, \beta)$, as in Table 1A. The sample standard error and the average standard error turn out to be close, as before.

Table 1B: Simulation results of Musa and Okumoto process on the ML estimator of ν with corresponding standard errors and estimated coverage probability for $t_k = 10$.

ν	$\bar{F}(t_k, \alpha, \beta)$	$\bar{\hat{\nu}}$	$\frac{(\bar{\hat{\nu}}-\nu)}{\nu}$	$\overline{s(\hat{\nu})}$	$sse(\hat{\nu})$	$\frac{\overline{s(\hat{\nu})}}{\nu}$	CP
100	0.1	99.68	-0.0032	6.68	7.24	0.07	0.88
	0.2	100.56	0.0056	15.76	14.13	0.16	0.87
	0.3	100.85	0.0085	27.33	19.35	0.27	0.86
500	0.1	499.11	-0.0018	15.01	14.77	0.03	0.93
	0.2	501.78	0.0036	33.49	35.33	0.07	0.93
	0.3	503.67	0.0073	56.76	56.40	0.11	0.92
1000	0.1	998.98	-0.0010	21.31	21.15	0.02	0.95
	0.2	998.01	-0.0020	46.82	47.74	0.05	0.94
	0.3	1003.30	0.0033	78.28	80.44	0.08	0.94

Table 1C: Simulation results of Goel and Okumoto process on the ML estimator of ν with corresponding standard errors and estimated coverage probability for $t_k = 10$.

ν	$\bar{F}(t_k, \alpha, \beta)$	$\bar{\hat{\nu}}$	$\frac{(\bar{\hat{\nu}}-\nu)}{\nu}$	$\overline{s(\hat{\nu})}$	$sse(\hat{\nu})$	$\frac{\overline{s(\hat{\nu})}}{\nu}$	CP
100	0.1	98.28	-0.0172	8.89	11.61	0.09	0.93
	0.2	107.20	0.0720	15.48	13.67	0.15	0.84
	0.3	109.18	0.0918	15.72	12.25	0.16	0.80
500	0.1	507.70	0.0154	13.16	14.32	0.02	0.94
	0.2	486.22	-0.0276	27.60	30.83	0.06	0.87
	0.3	521.52	0.0430	40.66	44.44	0.08	0.85
1000	0.1	992.46	-0.0075	17.48	18.33	0.02	0.96
	0.2	1014.88	0.0149	35.72	37.09	0.04	0.94
	0.3	1028.99	0.0290	57.54	65.04	0.06	0.93

To study the effect of the number and size of debugging intervals, we have carried out another simulation study with fixed $t_k = 10$, while the time Δ between successive debugging is varied as 1, 5 and 10 with the corresponding number k of debugging being 10, 2 and 1. The probability of non-detection $\bar{F}(t_k, \alpha, \beta)$ is kept fixed at 0.2. The same simulation exercise as before is carried out with 5000 repetitions. The results of power law process, Musa and Okumoto process and Goel and Okumoto process are presented in Table 2A, Table 2B and Table 2C, respectively.

Table 2A: Simulation results of Power Law Process with varying number of debugging and $\bar{F}(t_k, \alpha, \beta) = 0.2$ for $t_k = 10$.

ν	k	$\bar{\hat{\nu}}$	$(\bar{\hat{\nu}} - \nu)/\nu$	$\overline{s(\hat{\nu})}$	$\overline{s(\hat{\nu})}/\nu$
100	1	99.51	0.0049	6.71	0.0671
	2	100.04	-0.0004	8.09	0.0809
	10	100.21	-0.0021	12.81	0.1281
500	1	499.22	0.0016	14.68	0.0294
	2	499.90	0.0002	17.30	0.0346
	10	500.11	-0.0002	28.66	0.0573
1000	1	998.93	0.0011	19.78	0.0198
	2	1000.01	0.0000	24.61	0.0246
	10	998.43	0.0016	39.00	0.0390

Table 2B: Simulation results of Musa and Okumoto process with varying number of debugging and $\bar{F}(t_k, \alpha, \beta) = 0.2$ for $t_k = 10$.

ν	k	$\bar{\hat{\nu}}$	$(\bar{\hat{\nu}} - \nu)/\nu$	$\overline{s(\hat{\nu})}$	$\overline{s(\hat{\nu})}/\nu$
100	1	99.86	-0.0014	6.55	0.0655
	2	99.34	-0.0066	7.52	0.0752
	10	99.76	-0.0024	14.32	0.1432
500	1	500.04	0.0001	15.28	0.0306
	2	499.64	-0.0007	17.02	0.0340
	10	499.98	0.0001	34.18	0.0684
1000	1	1001.30	0.0013	20.92	0.0209
	2	999.35	-0.0006	23.73	0.0237
	10	1001.77	0.0018	46.80	0.0468

Table 2C: Simulation results of Goel and Okumoto process with varying number of debugging and $\bar{F}(t_k, \alpha, \beta) = 0.2$ for $t_k = 10$.

ν	k	$\bar{\hat{\nu}}$	$(\bar{\hat{\nu}} - \nu)/\nu$	$\overline{s(\hat{\nu})}$	$\overline{s(\hat{\nu})}/\nu$
100	1	99.38	-0.0062	6.71	0.0671
	2	102.31	0.0231	8.76	0.0876
	10	109.30	0.0930	13.73	0.1373
500	1	499.03	0.0019	14.58	0.0292
	2	502.40	0.0048	18.41	0.0368
	10	519.86	0.0397	37.85	0.0757
1000	1	998.95	-0.0010	20.39	0.0204
	2	1001.87	0.0019	25.77	0.0258
	10	1025.24	0.0252	54.45	0.0544

In all the cases, the average standard error and the sample standard error turn out to be very close, as before, and so only the average standard errors ($\overline{s(\hat{\nu})}$'s) and the relative standard errors ($\overline{s(\hat{\nu})}/\nu$'s) are reported. The performance of $\hat{\nu}$ in terms of average as well as relative standard errors seems to be better with decreasing number k of debugging intervals, for all ν. Although this may sound counter-intuitive, it is to be noted that, with decreasing k, there are more replications of failure time from a fault resulting in more information and, hence, better efficiency. Therefore, a single debugging schedule at the end of testing at time t_k, as in [3], seems to be the most efficient design. However, as remarked in Section 1, a schedule of more than one debugging intervals may be necessary due to the market demand for software release.

It is of interest to investigate the robustness of the estimator ν, against misspecification of the NHPP model. For this purpose, while the true failure rate corresponds to Musa and Okumoto process, we have obtained the estimate $\hat{\nu}$ assuming the Goel and Okumoto process, and vice-versa. As before, we consider 10 debugging intervals with one unit width of debugging. and the estimates are averaged over 5000 repetitions. The results are reported in Tables 3. From the results of this limited simulation study, the estimator $\hat{\nu}$ seems to be mildly sensitive to the misspecification of the NHPP model. As expected, the relative standard error increases with the extent of missingness.

Table 3: Sensitivity of $\hat{\nu}$ in terms of relative bias $(\hat{\nu}-\nu)/\nu$ and relative standard error $\overline{s(\hat{\nu})}/\nu$ for misspecification of NHPP model, when $t_k = 10$.

		True Model: G-O* Assumed Model: M-O*		True Model: M-O* Assumed Model: G-O*	
ν	$\bar{F}(t_k,\alpha,\beta)$	$(\bar{\hat{\nu}}-\nu)/\nu$	$\overline{s(\hat{\nu})}/\nu$	$(\bar{\hat{\nu}}-\nu)/\nu$	$\overline{s(\hat{\nu})}/\nu$
100	0.1	0.0622	0.1064	0.0051	0.0794
	0.2	0.1098	0.1392	-0.0197	0.1247
	0.3	0.1220	0.1711	-0.0812	0.1300
500	0.1	0.0117	0.0365	0.0108	0.0323
	0.2	0.0384	0.0743	0.0050	0.0665
	0.3	0.0678	0.1087	-0.0114	0.0867
1000	0.1	0.0056	0.0267	0.0108	0.0235
	0.2	0.0217	0.0519	0.0070	0.0505
	0.3	0.0419	0.0768	-0.0002	0.0715

G-O*, Goel and Okumoto process; M-O*, Musa and Okumoto process

7 Concluding Remarks

In this article, we have studied the design of periodic debugging schedule, introduced in Dewanji et.al. [5] and apparently practiced in many real applications. Extending the work of Das et.al. [3], we have used a model based on the assumption that the failure occurrences caused by each individual fault follow independent and identically distributed Non-homogeneous Poisson processes with a common but unknown failure rate. We have considered a general functional form of the cumulative hazard function, which includes the power law, Goel and Okumoto and Musa and Okumoto process, as special cases. We have provided a method for estimating the unknown number of bugs in the software and studied the asymptotic properties of the estimators. Among the competing models, discussed in Section 6, one can choose a model by comparing the AIC values [1] of different models (lower the AIC, better the model). Finally, at the end of debugging, one can estimate the software reliability, defined as the probability of failure free operation for next s unit of time, by the the following expression

$$\hat{R}(s) = e^{-(\hat{\nu}-M_k)(\Lambda(\underline{\hat{\theta}},t_k+s)-\Lambda(\underline{\hat{\theta}},t_k))}.$$

References

[1] Hirotogu Akaike. Information theory as an exten- sion of the maximum likelihood principle. In New York Springer, editor, *Selected papers of hirotugu akaike*, pages 199–213, 1988.

[2] S. R. Dalal. *Software reliability models: A selective survey and new directions.* Handbook of Reliability Engineering, 2003.

[3] Sudipta Das, Anup Dewanji, and Ashis Kumar Chakraborty. Software reliability modeling with periodic debugging. *IEEE Transactions On Reliability*, 65(3):1449–14556, September 2016.

[4] Anup Dewanji, Tapan K. Nayak, and Pranab K. Sen. Estimating the number of components of a system of superimposed renewal processes. *Sankhyā: The Indian Journal of Statistics Series A*, pages 486–499, 1995.

[5] Anup Dewanji, Debasis Sengupta, and Ashis Kumar Chakraborty. A discrete time model for software reliability with application to a flight control software. *Applied Stochastic Models in Business and Industry*, 27:723–731, 2011.

[6] W. Feller. *An Introduction to Probability Theory and Its Application*, volume 1. Wiley Eastern Limited, New Delhi, 1968.

[7] Amrit L. Goel and Kazuhira Okumoto. When to stop testing and start using software? In *Proceedings of the 1981 ACM workshop/symposium on Measurement and evaluation of software quality*, pages 131–138, 1981.

[8] Z. Jelinski and P. Moranda. Software reliability research. *Statistical Computer Performance Evaluation, Academic Press*, pages 465–484, 1972.

[9] John D. Musa and Kazuhira Okumoto. A logarithmic poisson execution time model for software reliability measurement. In *Proceedings of the 7th international conference on Software engineering*. IEEE Press, 1984.

[10] Tapan K. Nayak. A note on estimating the number of errors in a system by recapture sampling. *Statistics & Probability Letters*, 7.3:191–194, 1988.

[11] P. K. Sen and J. M. Singer. *Large sample methods in statistics an introduction with application.* Chapman and Hall, London, 1993.

[12] Nozer D. Singpurwalla and Simon P. Wilson. Software reliability modeling. *International Statistical Review*, 62(3):289–317, December 1994.

[13] Shigeru Yamada. *Software reliability modeling: fundamentals and applications.*, volume 5. Springer, 2014.