Refined Software Reliability Models Based on Non-Homogeneous Poisson Processes

Tadashi Dohi† (01307065) and Kishor S. Trivedi‡ (nomember)
†Hiroshima University, Hiroshima, Japan, ‡Duke University, NC, USA

1. INTRODUCTION

The non-homogeneous Poisson processes (NHPPs) with finite mean value function have played a central role for developing the software reliability models during the last two decades. In this article, we propose a modified modelling framework for the NHPP type of software reliability models. Our approach can describe consistently the debugging behavior from finite population based on a complete probabilistic argument, and can provide similar but somewhat different software reliability models from earlier ones. As one of the most powerful advantages for our approach, it is shown that the resulting software reliability models have non-defective inter-failure time distributions with finite MTBF.

2. FINITE FAILURE MODELS

Following Musa, Iannino and Okumoto [1], we make three assumptions to develop the software reliability models:

(i) Whenever a software failure occurs in the test, the fault that caused it will be removed instantaneously.

(ii) There are M inherent faults in the software program, where M is a non-negative integer valued random variable.

(iii) Each failure, caused by a fault, occurs independently and randomly in time, according to the identical continuous-probability distribution function $F(t)$ ($F(0) = 0$ and $F(\infty) = 1$).

Figure 1 depicts the configuration of debugging behavior from finite population. Let $\{N(t), t \geq 0\}$ be the cumulative number of faults detected during the time interval $[0, t)$. From the assumptions (i)–(iii), we obtain the following binomial distribution:

$$\Pr\{N(t) = x \mid M = m\} = \binom{m}{x} F(t)^x \overline{F}(t)^{m-x}, \quad x = 0, 1, \cdots, m, \quad (1)$$

where $\overline{F}(\cdot) = 1 - F(\cdot)$. If the total number of faults remaining in the program at $t = 0$ is a Poisson distributed random variable with mean a (> 0), then it is seen that

$$\begin{aligned}\Pr\{N(t) = x\} &= \sum_{m=0}^{\infty} \Pr\{N(t) = x \mid M = m\} \\ &\quad \times \Pr\{M = m\} \\ &= \sum_{m=0}^{\infty} \binom{m}{x} F(t)^x \overline{F}(t)^{m-x} \frac{a^m e^{-a}}{m!} \\ &= \frac{[aF(t)]^x e^{-aF(t)}}{x!}. \end{aligned} \quad (2)$$

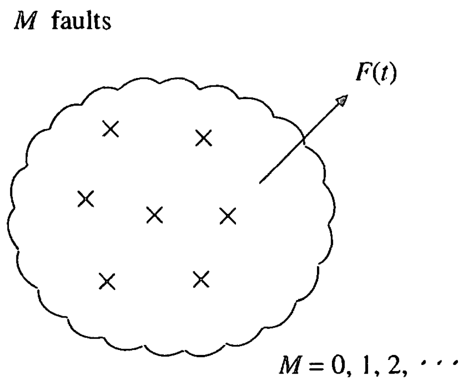

Figure 1: Configuration of debugging behavior from finite population.

This means that the total number of faults detected up to time t is reduced to the NHPP with mean value function $aF(t)$.

From Eq.(2), it is evident that the mean value function $aF(t)$ is finite, *i.e.* the function $aF(t)$ is non-decreasing and $\lim_{t\to\infty} aF(t) = a$. We call the NHPP given in Eq.(2) *the finite failure model* in this article. In fact, the finite failure model involves the almost classical software reliability models (see *e.g.* [2]).

It is well known that the NHPP in Eq.(2) is not a trivial stochastic process. Define the software reliability by $R(t) = \Pr\{N(t) = 0\} = \exp\{-aF(t)\}$. Then, it can be seen that the function $R(t)$ is monotonically decreasing inf t and converges to $\exp\{-a\}$ asymptotically. In other words, even if the software test is carried out for an infinite time span, there exists positive probability that no fault is found, *i.e.* that the program is "perfect". On the other hand, a technical problem will result from this perperty. Let T_i denote the ith ($i = 1, 2, \cdots$) software failure ocurrence time and $S_i = T_i - T_{i-1}$ ($i = 1, 2, \cdots, T_0 = 0$)

be the ith inter-failure time, having the distribution function $G_i(t)$. Since $G_1(t)$ has a mass part $\exp\{-a\}$ at infinity, the random variable S_1 is *defective*, that is, $\lim_{t\to\infty} G_1(t) = 1 - \exp\{-a\}$ and $\mathrm{E}[S_1]$ (and $\mathrm{E}[T_1]$) $\to \infty$. This leads to the fact that the inter-failure times S_i are all defective and that there does not exist a finite MTBF (mean time between failures). The relationship between the finite failure models and the deffective inter-failure times was investigated in [3]. However, note that the earlier literature [3] restricted itself to only the NHPPs with continuous probability mass function.

3. MODIFIED MODELING APPROACH

In order to overcome the problems mentioned in Section 2, we propose a modified modelling framework to explain the NHPP type of software reliability models. Suppose that M is a non-zero Poisson distributed random variable, *i.e.* M obeys the following truncated Poisson distribution:

$$\Pr\{M = m\} = \frac{a^m e^{-a}}{(1-e^{-a})m!}, \quad m = 1, 2, \cdots, \tag{3}$$

with mean $\mathrm{E}[M] = \exp\{-a\}/(1 - \exp\{-a\})$ and positive constant a. This assumption indicates that the software program is "not perfect" and posesses at least one fault. In a fashion similar to the previous discussion, we obtain

$$\begin{aligned}
\Pr\{N(t) = x\} &= \sum_{m=1}^{\infty} F(t)^x \overline{F}(t)^{m-x} \Pr\{M = m\} \\
&= \begin{cases} \sum_{m=1}^{\infty} \binom{m}{0} \overline{F}(t)^m \frac{a^m e^{-a}}{(1-e^{-a})m!} & (x = 0) \\ \sum_{m=x}^{\infty} \binom{m}{x} F(t)^x \overline{F}(t)^{m-x} \frac{a^m e^{-a}}{(1-e^{-a})m!} & (x > 0) \end{cases} \\
&= \begin{cases} \frac{e^{-aF(t)} - e^{-a}}{1-e^{-a}} & (x = 0) \\ \frac{[aF(t)]^x e^{-aF(t)}}{(1-e^{-a})x!} & (x > 0). \end{cases}
\end{aligned} \tag{4}$$

Hence, the probability mass function of $N(t)$ has a special mass part at $x = 0$ and is discontinuous with respect to x. It is evident that $\mathrm{E}[N(t)] = aF(t)/(1 - \exp\{-a\})$ and $\sum_{x=0}^{\infty} \Pr\{N(t) = x\} = 1$. From these results, we find the following properties:

(i) The NHPP in Eq.(4) has finite mean value function $\mathrm{E}[N(t)] = aF(t)/(1 - \exp\{-a\})$ and $\lim_{t\to\infty} \mathrm{E}[N(t)] = a/(1 - \exp\{-a\})$.

(ii) The corresponding reliability function $R(t)$ asymptotically converges to 0.

(iii) The first inter-failure time distribution $G_1(t) = \Pr\{X_1 \le t\} = (1 - \exp\{-aF(t)/(1 - \exp\{-a\})$ is non-defective. Hence, all inter-failure time distributions are also non-defective.

(iv) The fault detection rate

$$d(t) = \frac{d\mathrm{E}[N(t)]/dt}{\mathrm{E}[N(\infty)] - \mathrm{E}[N(t)]} \tag{5}$$

in our modified models is equivalent to that in the classical software reliability models with the same distribution function $F(t)$.

Assuming some theoretical distributions, we develop somewhat new software reliability models as follows.

(a) modified exponential: $F(t) = 1 - e^{-bt}$ $(b > 0)$

(b) modified hyper-exponential:

$$F(t) = \frac{1 - \sum_{i=1}^{n} p_i b_i e^{-b_i t}}{\sum_{i=1}^{n} p_i b_i} \quad (n > 1, b_i > 0, p_i \in [0,1], \sum_{i=1}^{n} p_i = 1)$$

(c) modified S-shape: $F(t) = 1 - (1 + bt)e^{-bt}$ $(b > 0)$

(d) modified gamma:

$$F(t) = 1 - \int_t^{\infty} \frac{b^\alpha y^{\alpha-1} e^{-by}}{\Gamma(\alpha)} dy \ (b > 0, \alpha > 1)$$

(e) modified Rayleigh: $F(t) = 1 - e^{-bt^2}$ $(b > 0)$

(f) modified Weibull: $F(t) = 1 - e^{-bt^\alpha}$ $(b > 0, \alpha > 1)$

(g) modified pareto:

$$F(t) = 1 - b^\alpha/(t + b)^\alpha \ (b > 0, \alpha > 0)$$

(h) modified extreme:

$$F(t) = 1 - \exp\{-[e^{\alpha+b} - e^\alpha]/b\} \quad (b > 0, \alpha > 0)$$

The above modeling framework is quite straightforward but overcomes serious problems in the classical finite failure models. Applying our modified approach, we can estimate the MTBF as an important software dependability measure and can treat the failure occurrence time sequence as well as the number of failures observed in the test.

REFERENCES

[1] Musa, J. D., Iannino, A. and Okumoto, K., *Software Reliability, Measurement, Prediction, Application*, McGraw-Hill, NY (1987).

[2] Langberg, N. and Singpurwalla, N. D., Unification of some software reliability models, *SIAM Journal on Scientific and Statistical Computing* **6**, 781–790 (1985).

[3] Koshimae, H., Tanaka, H. and Osaki, S., Some remarks on MTBF's for non-homogeneous Poisson process, *IEICE Trans. Fundamentals* **E77-A**, 144–149 (1994).