IEICE TRANS. FUNDAMENTALS, VOL.E95–A, NO.5 MAY 2012

PAPER

NHPP-Based Software Reliability Models Using Equilibrium Distribution*

Xiao XIAO[†a)], ***Student Member***, **Hiroyuki OKAMURA**[†], ***and*** **Tadashi DOHI**[†], ***Members***

SUMMARY Non-homogeneous Poisson processes (NHPPs) have gained much popularity in actual software testing phases to estimate the software reliability, the number of remaining faults in software and the software release timing. In this paper, we propose a new modeling approach for the NHPP-based software reliability models (SRMs) to describe the stochastic behavior of software fault-detection processes. The fundamental idea is to apply the equilibrium distribution to the fault-detection time distribution in NHPP-based modeling. We also develop efficient parameter estimation procedures for the proposed NHPP-based SRMs. Through numerical experiments, it can be concluded that the proposed NHPP-based SRMs outperform the existing ones in many data sets from the perspective of goodness-of-fit and prediction performance.

key words: *software reliability, equilibrium distribution, EM algorithm, NHPP, real data analysis*

1. Introduction

During the last four decades, software reliability engineering has played a central role to provide several quantitation methods used in real software development processes. Since the assessment of software reliability is one of the main issues in this area, one needs several kinds of mathematical models to assess quantitatively the software reliability, which is defined as the probability that the software system does not fail during a specified time period under specified operational environment. In the software reliability research, a huge number of software reliability models (SRMs) have been proposed from various points of view [10], [11], [13], [16]. Among the SRMs, non-homogeneous Poisson process (NHPP) models have gained much popularity in actual software testing phases to estimate the software reliability, the number of remaining faults in software and the software release timing.

One class of NHPP-based SRMs is concerned with modeling the number of faults detected in testing phases, initiated by Goel and Okumoto [5]. This is an exponential software reliability growth model and a generalized Goel-Okumoto NHPP-based SRM [6] with S-shaped growth curve of the detected software faults was originally reported in 1982. Shortly afterwards, Yamada, Ohba and Osaki [17] proposed another S-shaped software reliability growth model. More specifically, the NHPP-based SRM mentioned above is characterized by the mean value function, which is proportional to the cumulative distribution function of fault-detection time distribution. If the fault-detection time distribution is assumed as an exponential distribution, the corresponding NHPP-based SRM is the Goel-Okumoto NHPP-based SRM [5], which draws a concave curve (exponentially saturated mean value function). If the fault-detection time distribution is given by the Weibull [6] or two-stage Erlang distribution [17], then the mean trends of the fault-detection processes exhibit S-shaped curves.

In general, the fault-detection time distribution of the SRM with concave curve has the decreasing fault-detection rate (DFDR) or equivalently decreasing failure rate (DFR) property, and the S-shaped curve is based on the increasing fault-detection rate (IFDR) or increasing failure rate (IFR) property of fault-detection time distribution. Since there are few statistical distributions having the DFR property, the SRMs with concave curve are less than those with S-shaped curve. In fact, although it is well known that Littlewood NHPP-based SRM [1] and the generalized Goel-Okumoto NHPP-based SRM [6] can also provide concave-like patterns of software fault-detection processes, there are not many NHPP-based SRMs that give concave curves in the past literature. However, a concave curve frequently appears as a pattern of software fault-detection process observed in real software development projects. This motivates us to consider an NHPP-based modeling framework that provides SRMs with concave curve.

In this paper, we propose a new modeling approach for the NHPP-based SRMs to describe the stochastic behavior of software fault-detection processes. The fundamental idea is to apply the equilibrium distribution to the fault-detection time distribution. This is similar but somewhat different from the approach in [4], where the equilibrium distribution was employed in the development of an infinite server queueing model, which unified the finite and infinite fault models. For an arbitrary probability distribution, the corresponding equilibrium distribution can be defined. Since the equilibrium distribution is the steady-state solution on the age and residual life of a renewal process, it can be regarded as the software fault-detection time distribution after the software test is executed for a long time. Moreover, the NHPP-based SRM with the equilibrium distribution ensures that the mean value function draws a concave curve. We study the effectiveness of equilibrium distribution in software reliability modeling and compare the proposed

Manuscript received September 27, 2011.
Manuscript revised January 13, 2012.
[†]The authors are with the Department of Information Engineering, Graduate School of Engineering, Hiroshima University, Higashihiroshima-shi, 739-8527 Japan.
[*]This paper is an extended version of the conference paper [15] presented at IEEE ARES 2009.
a) E-mail: xiaoxiao@rel.hiroshima-u.ac.jp
DOI: 10.1587/transfun.E95.A.894

NHPP-based SRMs with the existing ones. Additionally, we develop the EM (expectation-maximization) algorithms [3] for the proposed NHPP-based SRMs and compare the calculation accuracy of EM algorithm with the usual Newton's method.

The remainder of this paper is organized as follows. In Sect. 2, we introduce a basic modeling framework of NHPP-based SRMs. In Sect. 3, we introduce the equilibrium distribution and propose a new modeling approach based on the equilibrium distribution. We call the proposed NHPP-based SRMs with equilibrium distribution the ED-NHPP-based SRMs, which draw a concave curve as a mean value of the number of detected faults. In Sect. 4, we provide an efficient parameter estimation procedure for the ED-NHPP-based SRMs based on the EM algorithm. Section 5 is devoted to the real project data analysis, where we use 4 real project data [10] and investigate the goodness-of-fit performance of the ED-NHPP-based SRMs, where the maximum likelihood (ML) method is applied for estimating model parameters, and the Akaike information criterion (AIC) [2] and the Bayesian information criterion (BIC) [14] as well as the mean squares error (MSE) are used for model selection. We also conduct the prediction analysis and investigate the prediction ability of the ED-NHPP-based SRMs. The paper is concluded in Sect. 6 with some remarks.

2. NHPP-Based SRM

Let $N(t)$ denote the number of software faults detected by testing time t, and be a stochastic point process in continuous time. We make the following assumptions:

Assumption A: Software faults occur at independent and identically distributed (i.i.d.) random times having a cumulative distribution function (c.d.f.) $F(t)$ with a probability density function (p.d.f.) $f(t) = dF(t)/dt$.

Assumption B: The initial number of software faults, N, is nonnegative and finite.

Under the above assumptions, the probability mass function (p.m.f.) of the number of software faults detected by time t is given by the binomial p.m.f.:

$$\Pr\{N(t) = n \mid N\} = \binom{N}{n} F(t)^n \overline{F}(t)^{N-n}, \tag{1}$$

where $\overline{F}(\cdot) = 1 - F(\cdot)$. Figure 1 illustrates the configuration of software debugging theory under the above assumptions. When the software fault-detection time obeys an exponential distribution, then the stochastic process $\{N(t),\ t \geq 0\}$ is a pure birth process with absorbing boundary at $N(t) = N$ and is equivalent to the Jelinski and Moranda model [8].

If the initial number of faults N is unknown, it is appropriate to assume that N is a discrete (integer-valued) random variable. Langberg and Singpurwalla [9] proved that when the initial number of software faults N was a Poisson random variable with mean ω (> 0), the number of software faults detected before time t was given by the following NHPP:

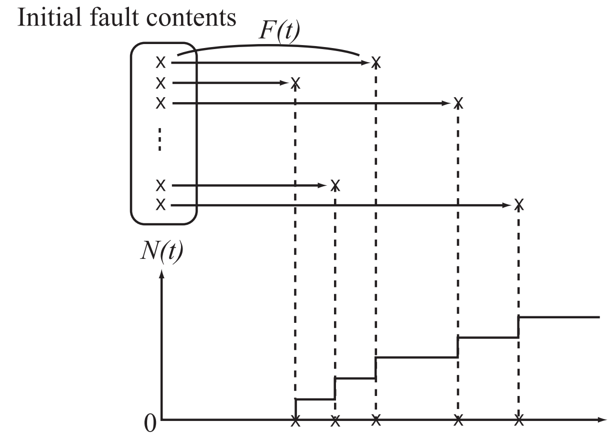

Fig. 1 Configuration of software debugging theory.

$$\begin{aligned}\Pr\{N(t) = n\} &= \sum_{x=n}^{\infty} \Pr\{N(t) = n \mid x\} \frac{\omega^x e^{-\omega}}{x!} \\ &= \frac{\{\omega F(t)\}^n}{n!} e^{-\omega F(t)}.\end{aligned} \tag{2}$$

Equation (2) is equivalent to the p.m.f. of the NHPP having a mean value function $\mathrm{E}[N(t)] = \Lambda(t) = \omega F(t)$. From this modeling framework, almost all NHPP-based SRMs can be derived by choosing the software fault-detection time distribution $F(t)$. If $F(t) = 1 - e^{-\beta t}$, then we can derive Goel-Okumoto NHPP-based SRM [5] with mean value function $\Lambda(t) = \omega\ (1 - e^{-\beta t})$. If the software fault-detection time distribution is given by the Weibull or two-stage Erlang distribution, then the resulting NHPP-based SRM becomes the Goel NHPP-based SRM [6] or the delayed S-shaped NHPP-based SRM [17].

3. NHPP-Based SRM Using Equilibrium Distribution

3.1 Model Description

Here we present an ED-NHPP-based SRM belonging to the general modeling framework described in Sect. 2. Before beginning the discussion of the proposal, we give a fundamental argument of renewal theory as a preliminary. Let $\{Y(t),\ t \geq 0\}$ be a renewal counting process with i.i.d. inter-failure time X_k ($k = 1, 2, \ldots$) having a continuous c.d.f. $G(t)$. Define the age and residual life of the renewal process by

$$\delta_t = t - S_{Y(t)}, \tag{3}$$
$$\gamma_t = S_{Y(t)+1} - t, \tag{4}$$

where $S_k = X_1 + X_2 + \ldots + X_k$. It is well known that $\delta_t + \gamma_t = S_{Y(t)+1} - S_{Y(t)} = X_{Y(t)+1}$ is called a *lifetime* of $X_{Y(t)+1}$ in the context of reliability engineering (see Fig. 2). From the renewal process argument, the age distribution and the residual life distribution satisfy the following renewal-type equations:

$$\Pr\{\delta_t \leq x\} = G(t) - \int_0^t \overline{G}(t-y)dM(y), \tag{5}$$

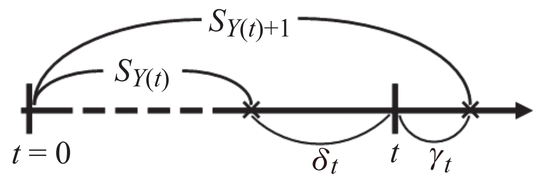

Fig. 2 A schematic illustration of age and residual life.

$$\Pr\{\gamma_t \le x\} = G(t+x) - \int_0^t \overline{G}(t+x-y)dM(y), \tag{6}$$

where $M(t) = \mathrm{E}[Y(t)] = G(t) + \int_0^t M(t-x)dG(x)$ is a renewal function of $Y(t)$. Taking account of a limit distribution for δ_t and γ_t, it can be seen that

$$\begin{aligned} G_e(x) &= \lim_{t\to\infty} \Pr\{\delta_t \le x\} = \lim_{t\to\infty} \Pr\{\gamma_t \le x\} \\ &= \frac{\int_0^x \overline{G}(y)dy}{\int_0^\infty \overline{G}(y)dy}, \end{aligned} \tag{7}$$

where $G_e(x)$ is called the *equilibrium distribution*.

Suppose that the software fault-detection time obeys the i.i.d. equilibrium distribution $F_e(t)$. From the similar discussion to Sect. 2, we have

$$\Pr\{N(t) = n\} = \frac{\{\omega F_e(t)\}^n}{n!} e^{-\omega F_e(t)}. \tag{8}$$

This paper calls the above model the ED-NHPP-based SRM with the mean value function $\Lambda_e(t) = \omega F_e(t)$.

3.2 Properties of ED-NHPP-Based SRMs

One of the most attractive properties of ED-NHPP-based SRMs is the age property of the fault-detection time distribution. Define the detection rate per fault and the mean residual testing time per fault as follows.

$$r(t) = \frac{f(t)}{\overline{F}(t)}, \tag{9}$$

and

$$\mu(t) = \frac{\int_t^\infty \overline{F}(s)ds}{\overline{F}(t)}, \tag{10}$$

where $f(t)$ and $\overline{F}(t)$ are p.d.f. and complement c.d.f. of fault-detection time. The detection rate and mean residual testing time are metrics for evaluating the testing effort of the software. If the detection rate function $r(t)$ increases or the mean residual testing time decreases as testing time elapses, it means that the software testing is successfully executed. The decreasing detection rate or the increasing residual testing time implies that the software tends to contain the faults difficult to detect, and that the testing might be inefficient. In general, the relationship among the properties of increasing fault-detection rate (IFDR), the decreasing fault-detection rate (DFDR), the increasing mean residual testing time (IMRT) and the decreasing mean residual testing time (DMRT) is given by[†]

$$\text{IFDR (DFDR)} \Rightarrow \text{DMRT (IMRT)}.$$

On the other hand, the detection rate per fault in the ED-NHPP-based SRM is written in the form:

$$r_e(t) = \frac{f_e(t)}{\overline{F}_e(t)} = \frac{\overline{F}(t)}{\int_t^\infty \overline{F}(s)ds} = \frac{1}{\mu(t)}. \tag{11}$$

That is, the detection rate per fault of the ED-NHPP-based SRM is equivalent to the reciprocal of mean residual testing time of the underlying NHPP-based SRM. This straightforwardly leads to the following result.

Property 1: If the fault-detection time has a DMRT (IMRT) property, then the fault-detection time in the corresponding ED-NHPP-based SRM has an IFDR (DFDR) property. □

Let T and T_e be the fault-detection times per fault in a NHPP-based SRM and its associated ED-NHPP-based SRM, respectively. Property 1 equivalently indicates the following result.

Property 2: If T is DMRT (IMRT), then T_e is stochastically less (greater) than T. □

The proofs of the above properties are given in Gupta [7]. Apart from the age property of fault-detection time, we discuss the behavior of the number of detected faults.

Property 3: The mean value function $\Lambda_e(t) = \mathrm{E}[N(t)]$ offers a concave function in t. □

Property 3 can be proved by the fact that $dF_e(x)/dx > 0$ and $d^2F_e(x)/dx^2 < 0$ in the ED-NHPP-based SRM. As a special case, if the underlying c.d.f. $F(t)$ is an exponential c.d.f. (EXP), then the resulting equilibrium distribution is also an identical exponential distribution. When the fault-detection phenomenon behaves with an exponential trend, the Goel-Okumoto NHPP-based SRM [5] was frequently and implicitly assumed without taking account of the distribution properties of fault-detection time. However, it can be found that our new modeling framework always provides several types of concave curves including an exponential trend curve.

Next, we give typical examples on the ED-NHPP-based SRMs. As mentioned before, the equilibrium distribution of an exponential distribution reduces the same distribution. When the fault-detection time distribution is the two-stage Erlang (TSE) distribution, i.e., $F(t) = 1 - (1 + \beta t)e^{-\beta t}$, then we have

$$\begin{aligned} \Lambda_e(t) &= \omega \frac{\int_0^t [(1+\beta t)e^{-\beta t}]dt}{\int_0^\infty [(1+\beta t)e^{-\beta t}]dt} \\ &= \omega \left[1 - \left(1 + \frac{1}{2}\beta t\right) e^{-\beta t}\right]. \end{aligned} \tag{12}$$

[†]IFDR, DFDR, IMRT and DMRT are equivalent to IFR, DFR, IMRL and DMRL properties of fault-detection time distribution, respectively.

Table 1 Typical NHPP-based SRMs.

Model	$\Lambda(t)$	$\Lambda_e(t)$
EXP	$\omega(1-e^{-\beta t})$	$\omega(1-e^{-\beta t})$
TSE	$\omega[1-(1+\beta t)e^{-\beta t}]$	$\omega[1-(1+\beta t/2)e^{-\beta t}]$
RAY	$\omega(1-e^{-\beta t^2})$	$\omega\mathrm{Erf}(\sqrt{\beta}t)$
WEB	$\omega(1-e^{-\beta t^c})$	$\frac{\omega\{\Gamma_1(1/c)-\Gamma_2(1/c,\beta t^c)\}}{c\Gamma_1(1+1/c)}$

On the other hand, in the case of the Rayleigh (RAY) distribution $F(t) = 1 - e^{-\beta t^2}$, it is seen that

$$\Lambda_e(t) = \omega\frac{\int_0^t e^{-\beta t^2}dt}{\int_0^\infty e^{-\beta t^2}dt} = \frac{\omega\sqrt{\pi}\mathrm{Erf}(\sqrt{\beta}t)/2\sqrt{\beta}}{\sqrt{\pi}/2\sqrt{\beta}} = \omega\mathrm{Erf}(\sqrt{\beta}t), \tag{13}$$

where $\mathrm{Erf}(\cdot)$ is the error function. This can be extended to the general case. For a constant c (> 0), we suppose the Weibull (WEB) fault-detection time, i.e., $F(t) = 1 - e^{-\beta t^c}$. Then, we get

$$\Lambda_e(t) = \omega\frac{\int_0^t e^{-\beta t^c}dt}{\int_0^\infty e^{-\beta t^c}dt} = \frac{\omega}{c}\frac{\int_0^{\beta t^c} t^{\frac{1}{c}-1}e^{-t}dt}{\int_0^\infty t^{\frac{1}{c}}e^{-t}dt} = \frac{\omega}{c}\frac{\Gamma_1(1/c)-\Gamma_2(1/c,\beta t^c)}{\Gamma_1(1+1/c)}, \tag{14}$$

where $\Gamma_1(\cdot)$ and $\Gamma_2(\cdot,\cdot)$ denote the complete and incomplete gamma functions, respectively. Table 1 summarizes the NHPP-based SRMs with different software fault-detection time distributions.

4. EM Algorithms for ED-NHPP-Based SRMs

One of the practical issues in the software reliability assessment is how to estimate model parameters fitted to observed software fault data. The commonly used technique for parameter estimation is the ML method. Define the model parameter vector (ω, θ) in the mean value function $\Lambda(t) = \Lambda(t;\omega,\theta) = \omega F(t;\theta)$ or $\Lambda_e(t) = \Lambda_e(t;\omega,\theta) = \omega F_e(t;\theta)$. Suppose that n software fault (group) data $(t_1, x_1), \ldots, (t_n, x_n)$ are available, where t_k and x_k are the k-th testing date and the cumulative number of software faults detected by t_k, respectively. Then, for the ED-NHPP-based SRM, the log-likelihood function (LLF) is given by

$$\mathcal{L}_e(\omega,\theta) = \sum_{k=1}^{n}(x_k-x_{k-1})\ln[\Lambda_e(t_k;\omega,\theta)-\Lambda_e(t_{k-1};\omega,\theta)] - \Lambda_e(t_n;\omega,\theta) - \sum_{k=1}^{n}\ln[(x_k-x_{k-1})!]. \tag{15}$$

The ML estimate $(\hat{\omega},\hat{\theta})$ is formally defined as a solution of $\max_{\omega,\theta}\mathcal{L}_e(\omega,\theta)$. In most cases, the maximization problem has been solved by applying general-purpose and numerical optimization methods like Newton and quasi-Newton methods. However, it is empirically known that general-purpose optimization methods cause computational errors such as numerical exception due to their local convergence property when dealing with software fault data observed in real software projects. Thus the most practical issue to be addressed is to develop numerically stable algorithms finding the ML estimates from observed fault data. Okamura, Watanabe and Dohi [12] proposed stable ML estimation procedures for NHPP-based SRMs based on the EM (expectation-maximization) algorithms. This paper also provides ML estimation procedures for ED-NHPP-based SRMs by applying the fundamental idea discussed in [12].

The EM algorithm is an iterative procedure to compute ML estimates from incomplete data. Let D and Z be generally observable and unobservable data. When considering the ML estimation for a parameter vector θ, the EM algorithm can be formulated as the following equation [3]:

$$\theta = \underset{\theta}{\mathrm{argmax}} \int p(Z|D;\theta')\ln p(D,Z;\theta)dZ, \tag{16}$$

where $\ln p(D,Z;\theta)$ is an LLF for both observable and unobservable data, and $p(Z|D;\theta')$ is a posterior distribution of unobservable data, provided that the data D and a provisional parameter vector θ' are given. Equation (16) means the maximization problem of the expected LLF for the complete data (D, Z), although the usual ML estimation maximizes the LLF for D. In addition, the equation presents an update formula for the parameter vector θ. If the provisional parameter vector θ' equals the exact ML estimate of θ, the left-hand side of Eq. (16) also becomes the exact ML estimate. In other words, the iterative procedure based on Eq. (16) results the ML estimate of parameter θ. The advantage of EM algorithm over Newton and quasi-Newton methods is a global convergence property. Since the EM algorithm rarely leads to computational errors owing to the global convergence property, it allows us to reduce the effort of selecting initial (provisional) parameters. On the other hand, the concrete procedures of EM algorithm depend on model and data structures. Thus we should build a specific EM procedure for each model to be estimated.

Okamura, Watanabe and Dohi [12] developed a general framework of EM algorithms for NHPP-based SRMs for group data. In brief, the idea behind the EM algorithms for NHPP-based SRMs is to define the remaining software faults as unobservable data. Concretely, assuming all software fault-detection times $T_1 < \ldots < T_N$ are observed, the ML estimates for SRM parameters ω and θ are given by

$$\hat{\omega} = N, \tag{17}$$

and

$$\hat{\theta} = \underset{\theta}{\mathrm{argmax}} \sum_{i=1}^{N}\ln f(T_i;\theta). \tag{18}$$

That is, the ML estimation under all the software detection-time data is reduced to a parameter estimation for only fault-detection time distribution $f(\cdot)$. However, under the situation where the group data $(t_1, x_1), \ldots, (t_n, x_n)$ are observable, we know neither the total number of software faults nor the exact detection times, i.e., the unobservable data are defined as $Z = (T_1, \ldots, T_N, N)$.

Applying the fundamental formula (Eq. (16)), we get the following update formulas for NHPP-based SRMs:

$$\hat{\omega} = \mathrm{E}_Z \left[N | D; \omega', \theta' \right], \tag{19}$$

and

$$\hat{\theta} = \underset{\theta}{\mathrm{argmax}}\, \mathrm{E}_Z \left[\sum_{i=1}^{N} \ln f(T_i; \theta) \middle| D; \omega', \theta' \right], \tag{20}$$

where $\mathrm{E}_Z [\cdot | D; \omega', \theta']$ is the expected value of the posterior distribution $p(Z|D; \omega', \theta')$ using provisional parameters ω' and θ'. Although the computation of this kind of expected value is not so easy, a general formula was provided in [12]:

For an arbitrary function $h(\cdot)$ and group data $D = ((t_1, x_1), \ldots, (t_n, x_n))$, it holds

$$\mathrm{E}_Z \left[\sum_{i=1}^{N} h(T_i) \middle| D; \omega, \theta \right] = \sum_{i=1}^{n} \frac{(x_i - x_{i-1}) \int_{t_{i-1}}^{t_i} f(s;\theta) ds}{\int_{t_{i-1}}^{t_i} f(s;\theta) ds} + \omega \int_{t_n}^{\infty} f(s;\theta) ds. \tag{21}$$

Based on the above formulas, we develop the EM algorithms for ED-NHPP-based SRMs. The concrete parameter update formulas are obtained as follows.

ED-EXP:

$$\omega = \mathrm{E}_Z \left[N | D; \omega', \beta' \right], \tag{22}$$

$$\beta = \frac{\mathrm{E}_Z \left[N | D; \omega', \beta' \right]}{\mathrm{E}_Z \left[\sum_{i=1}^{N} T_i \middle| D; \omega', \beta' \right]}, \tag{23}$$

$$\mathrm{E}_Z \left[N | D; \omega, \beta \right] = x_n + \omega \xi_1(t_n; \beta), \tag{24}$$

$$\mathrm{E}_Z \left[\sum_{i=1}^{N} T_i \middle| D; \omega, \beta \right] = \sum_{i=1}^{n} \frac{(x_i - x_{i-1}) \left[\xi_2(t_{i-1}; \beta) - \xi_2(t_i; \beta) \right]}{\xi_1(t_{i-1}; \beta) - \xi_1(t_i; \beta)} + \omega \xi_2(t_n; \beta), \tag{25}$$

$$\xi_1(t; \beta) = \int_t^{\infty} \beta e^{-\beta s} ds, \tag{26}$$

$$\xi_2(t; \beta) = \int_t^{\infty} \beta s e^{-\beta s} ds. \tag{27}$$

ED-TSE:

$$\omega = \mathrm{E}_Z \left[N | D; \omega', \beta' \right], \tag{28}$$

$$\beta = \frac{\mathrm{E}_Z \left[N | D; \omega', \beta' \right]}{\mathrm{E}_Z \left[\sum_{i=1}^{N} \frac{\beta' T_i^2}{1 + \beta' T_i} \middle| D; \omega', \beta' \right]}, \tag{29}$$

$$\mathrm{E}_Z \left[N | D; \omega, \beta \right] = x_n + \omega \xi_1(t_n; \beta), \tag{30}$$

$$\mathrm{E}_Z \left[\sum_{i=1}^{N} \frac{\beta T_i^2}{1 + \beta T_i} \middle| D; \omega, \beta \right] = \sum_{i=1}^{n} \frac{(x_i - x_{i-1}) \left[\xi_2(t_{i-1}; \beta) - \xi_2(t_i; \beta) \right]}{\xi_1(t_{i-1}; \beta) - \xi_1(t_i; \beta)} + \omega \xi_2(t_n; \beta), \tag{31}$$

$$\xi_1(t; \beta) = \int_t^{\infty} \frac{\beta}{2} (1 + \beta s) e^{-\beta s} ds, \tag{32}$$

$$\xi_2(t; \beta) = \int_t^{\infty} \frac{\beta^2 s^2}{2} e^{-\beta s} ds. \tag{33}$$

ED-WEB:

$$\omega = \mathrm{E}_Z \left[N | D; \omega', \beta' \right], \tag{34}$$

$$\beta = \frac{\mathrm{E}_Z \left[N | D; \omega', \beta' \right]}{c\mathrm{E}_Z \left[\sum_{i=1}^{N} T_i^c \middle| D; \omega', \beta' \right]}, \tag{35}$$

$$\mathrm{E}_Z \left[N | D; \omega, \beta \right] = x_n + \omega \xi_1(t_n; \beta), \tag{36}$$

$$\mathrm{E}_Z \left[\sum_{i=1}^{N} T_i^c \middle| D; \omega, \beta \right] = \sum_{i=1}^{n} \frac{(x_i - x_{i-1}) \left[\xi_2(t_{i-1}; \beta) - \xi_2(t_i; \beta) \right]}{\xi_1(t_{i-1}; \beta) - \xi_1(t_i; \beta)} + \omega \xi_2(t_n; \beta), \tag{37}$$

$$\xi_1(t; \beta) = \int_t^{\infty} \frac{\beta^{1/c} e^{-\beta s^c}}{\Gamma(1 + 1/c)} ds, \tag{38}$$

$$\xi_2(t; \beta) = \int_t^{\infty} \frac{s^c \beta^{1/c} e^{-\beta s^c}}{\Gamma(1 + 1/c)} ds. \tag{39}$$

Note that the shape parameter of ED-WEB is fixed for the simplification of the algorithm. Of course, by changing the shape parameter, one can estimate the mean value function effectively.

5. Project Data Analysis

5.1 Goodness-of-Fit Test

In the numerical examples, we use 4 real project data sets [10] and compare the ED-NHPP-based SRMs with their associated existing NHPP-based SRMs with 4 underlying distributions (EXP, TSE, RAY, WEB). The data sets used here are software fault count (group) data (the number of detected software faults on each testing date is recorded) and consist of 133, 351, 266 and 367 fault data, respectively. They are cited from reference [10], where named J1, J3, DATA14 and J5, respectively. We rename them as DS1 through DS4 here. We estimate model parameters included in the ED-NHPP-based SRMs and the existing NHPP-based SRMs by means of the ML estimation, and calculate the information criteria; AIC and BIC as well as MSE:

$$\mathrm{AIC} = -2\mathrm{MLLF} + 2\phi, \tag{40}$$

$$\mathrm{BIC} = -2\mathrm{MLLF} + \phi \ln(n), \tag{41}$$

$$\mathrm{MSE} = \frac{\sqrt{\sum_{i=1}^{n} (x_i - E[X(t_i)])^2}}{n}, \tag{42}$$

where MLLF denotes the maximum log likelihood, ϕ is the number of free parameters (model dimension) and n is the

Table 2 Goodness-of-fit test.

	$\Lambda(t)$	AIC	BIC	MSE	K-S test		$\Lambda_e(t)$	AIC	BIC	MSE	K-S test	
					5%	1%					5%	1%
DS1	EXP	288.142	292.396	0.666090	yes	yes	EXP	288.142	292.396	0.666090	yes	yes
(133)	TSE	312.193	316.447	0.917306	yes	yes	TSE	287.720	291.974	0.662027	yes	yes
	RAY	326.450	330.704	1.159128	yes	yes	RAY	287.380	291.634	0.669138	yes	yes
	WEB	290.028	296.409	0.678554	yes	yes	WEB	290.108	296.490	0.663190	yes	yes
	$\Lambda(t)$	AIC	BIC	MSE	K-S test		$\Lambda_e(t)$	AIC	BIC	MSE	K-S test	
					5%	1%					5%	1%
DS2	EXP	325.498	328.925	3.451813	yes	yes	EXP	325.498	328.925	3.451813	yes	yes
(351)	TSE	291.441	294.868	2.383778	yes	yes	TSE	309.219	312.646	3.052395	yes	yes
	RAY	314.617	318.045	2.767897	yes	yes	RAY	294.025	297.452	2.750434	yes	yes
	WEB	283.774	288.914	2.140826	yes	yes	WEB	278.248	283.389	1.916147	yes	yes
	$\Lambda(t)$	AIC	BIC	MSE	K-S test		$\Lambda_e(t)$	AIC	BIC	MSE	K-S test	
					5%	1%					5%	1%
DS3	EXP	491.258	494.915	1.961698	yes	yes	EXP	491.258	494.915	1.961698	yes	yes
(266)	TSE	512.326	515.983	2.811853	no	yes	TSE	489.625	493.283	1.966123	yes	yes
	RAY	564.395	568.052	3.670359	no	yes	RAY	487.299	490.956	1.963611	yes	yes
	WEB	489.153	494.639	2.046971	yes	yes	WEB	489.201	494.687	2.526868	yes	yes
	$\Lambda(t)$	AIC	BIC	MSE	K-S test		$\Lambda_e(t)$	AIC	BIC	MSE	K-S test	
					5%	1%					5%	1%
DS4	EXP	345.204	349.785	0.812538	yes	yes	EXP	345.204	349.785	0.812538	yes	yes
(367)	TSE	353.591	358.172	1.873637	yes	yes	TSE	344.553	349.134	0.802378	yes	yes
	RAY	396.936	401.516	2.930417	yes	yes	RAY	344.521	349.102	0.805137	yes	yes
	WEB	339.840	346.711	0.972155	yes	yes	WEB	346.658	353.529	0.804419	yes	yes

number of data set $(t_i,\ x_i) = (i = 1,\ 2, \ldots,\ n)$. In addition to the model selection based on the information criteria, we take place the Kolmogorov-Smirnov (K-S) test with two significance levels (5% and 1%). If the K-S test was accepted ('yes' in Table 2), it means that the SRM assumed fits to the underlying data. Hence, for the accepted data sets through the K-S test, we compare AIC, BIC and MSE, and select the best SRM based on them.

Table 2 presents the goodness-of-fit results for all the data sets. The ED-NHPP-based SRMs could fit to the software fault data in all data sets. On the other hand, when the usual NHPP-based SRMs are assumed, the K-S test could not accept all models in DS3. On the information criteria; AIC and BIC, the ED-NHPP-based SRMs with TSE and RAY gave better results than that of the NHPP-based SRMs in 3 data sets, but the ED-NHPP-based SRM with WEB provided the smaller AICs/BICs only in DS2. It is also observed that WEB best fitted to the data in NHPP-based SRMs, while in the case of ED-NHPP-based SRMs, RAY performanced the best. On the other hand, if one is interested in MSE, the ED-NHPP-based SRMs could minimize it in almost all cases. Throughout the comparative study performed here, it can be concluded that the ED-NHPP-based SRMs can provide the satisfactory goodness-of-fit performance to the real software fault data. Figure 3 illustrates the behavior of both the observed and the estimated number of accumulated software faults by each testing date.

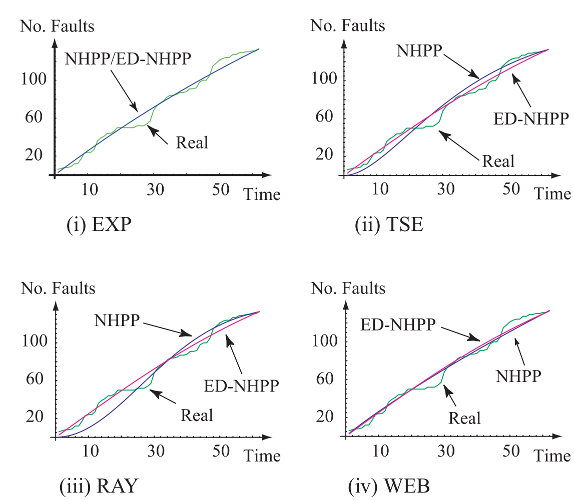

Fig. 3 Behavior of cumulative number of detected software faults (DS1).

We evaluate the quantitative software reliability, which

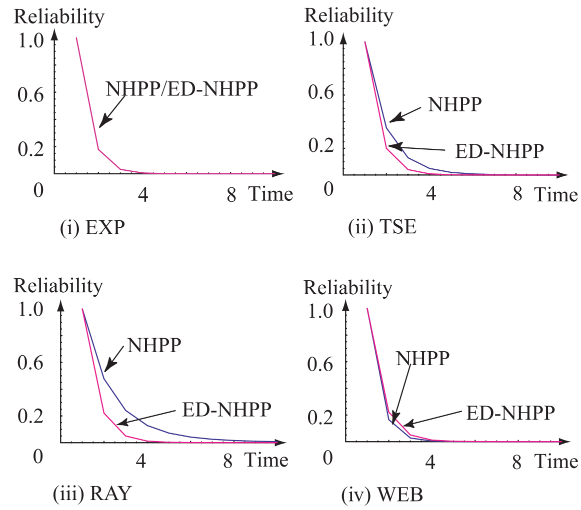

Fig. 4 Plot of software reliability function (DS1).

is the probability that the software system does not fail during a specified time interval after release. Let t_n be the time to detect the n-th fault. Suppose that the software test terminates at time t_n and the product is released at the same time to the user or market. Then, the software reliability for the operational period $[t_n,\ t_n + x)$ is defined by

$$R(x \mid t_n) = \exp\{-[\Lambda(t_n + x) - \Lambda(t_n)]\}, \tag{43}$$

where the reliability for ED-NHPP-based SRMs is derived by replacing $\Lambda(\cdot)$ with $\Lambda_e(\cdot)$. Figure 4 shows the behavior of software reliability function with DS1. In TSE and RAY, the ED-NHPP-based SRMs tend to under-estimate the usual NHPP-based SRMs. In other words, our ED-NHPP-based SRMs provide pessimistic prediction in assessing the software reliability. Actually this property would be acceptable for practitioners, because the software reliability should be estimated smaller from the safety point of view in practice.

5.2 Effectiveness of EM Algorithm

We investigate the effectiveness of the EM algorithm for ED-NHPP-based SRMs and compare it with the Newton's method which is employed by the ML estimation in goodness-of-fit test. In using of the Newton's method, the accuracy of the estimates strongly depends on the choice of initial values. In our experiment, the initial values for model parameter ω and β are set by uniform random numbers that are generated in the range of (1, 10), (10, 100), (100, 1000) and (0.1, 1), (0.01, 0.1), (0.001, 0.01), (0.0001, 0.001), (0.00001, 0.0001), respectively. That is, we executed the Newton's method for 18 times with different initial parameter values to estimate the model parameters. On the other hand, since EM algorithm has the global convergence property, there is almost no need to adjust carefully the initial values. But for purpose of comparison, we set the initial parameter values in the same way as metioned above to keep the comparison fairly. In use of the Newton's method, several estimations ended up in numerical exception such as

Table 3 Rate of successful estimations of EM algorithm and Newton's method in ED-NHPP-based SRMs.

		DS1	DS2	DS3	DS4
Newton's method	EXP	0.83	0.67	0.44	0.83
	TSE	0.83	1.00	1.00	0.83
	RAY	0.50	0.67	0.50	0.83
EM algorithm	EXP	1.00	1.00	1.00	1.00
	TSE	1.00	1.00	1.00	1.00
	RAY	0.83	0.83	0.83	0.83

division by zero. These failures arised from the setting of the initial values but this kind of problem rarely occured in case of EM algorithm. So we take in the rate of successful estimations (ROS) of each method where ROS = (the number of successful estimations)/18 $\times 100\%$ is a criterion for evaluation.

Table 3 presents the ROS of the Newton's method and EM algorithm. We executed EM algorithm for 1000 times which was considered as a sufficient number of iterations because the estimator converged to a stable value. Note that we examined the special case of ED-WEB (ED-RAY) where the shape parameter c is set to be 2. From Table 3, it is clear that ROS of EM algorithm for ED-EXP and ED-TSE show 100% in all data sets. On the other hand, the Newton's method gives lower ROS than that of EM algorithm in most cases, and especially shows ROS with only 44% in DS3. For ED-RAY, EM algorithm also failed in a few cases but still provides high ROS (83%). However, ROS of the usual Newton's method decreases to 50% in 3 data sets. From these ovservations, it can be concluded that the Newton's method does not often function well, and EM algorithm is superior to the usual method.

Note that the global convergence property of the EM algorithm is guaranteed theoretically [3], but it does not always guarantees a global optimizer. As the evaluation executed above, compared with the Newton's method, the EM algorithm makes it less necessary to carefully adjust the initial values. However, in order to avoid the local optimizer, it in some cases may be needed to adjust the initial values for the EM algorithm. In the case of multi-modal, the EM algorithm guarantees a global optimizer under certain regularity conditions. In our experiments, the estimator of EM algorithm provided the same value when it converged.

5.3 Prediction Analysis

Finally, we examine the prediction performance of the ED-NHPP-based SRMs, where two prediction measures are used: predictive log likelihood (PLL) and predictive least square error (PLS). Regard the data from an observation point to the end of the observation as future data. The PLL is defined as the logarithm of likelihood function with future data, and the PLS is the residual sum of errors between the mean value function and the future data. Table 4 presents the

Table 4 Prediction performance based on predictive log likelihood.

DS1	50%		75%		90%	
	$\Lambda(t)$	$\Lambda_e(t)$	$\Lambda(t)$	$\Lambda_e(t)$	$\Lambda(t)$	$\Lambda_e(t)$
EXP	−138.071	−138.071	−136.067	−136.067	−139.528	−139.528
TSE	−154.378	−138.072	−141.994	−135.919	−136.143	−139.529
RAY	−191.241	−138.071	−151.081	−135.808	−135.585	−139.527
WEB	−137.095	−138.065	−136.319	−136.143	−139.050	−139.525
DS2	50%		75%		90%	
	$\Lambda(t)$	$\Lambda_e(t)$	$\Lambda(t)$	$\Lambda_e(t)$	$\Lambda(t)$	$\Lambda_e(t)$
EXP	−419.714	−419.714	−391.755	−391.755	−367.970	−367.970
TSE	−891.195	−419.718	−361.419	−380.925	−357.690	−363.850
RAY	−319.908	−419.691	−358.796	−370.700	−353.871	−359.339
WEB	−428.378	−419.805	−365.780	−357.947	−357.948	−354.522
DS3	50%		75%		90%	
	$\Lambda(t)$	$\Lambda_e(t)$	$\Lambda(t)$	$\Lambda_e(t)$	$\Lambda(t)$	$\Lambda_e(t)$
EXP	−265.067	−265.067	−296.335	−296.335	−267.243	−267.243
TSE	−549.440	−262.246	−274.752	−297.429	−265.849	−266.431
RAY	−382.043	−264.081	−273.005	−299.211	−268.347	−265.710
WEB	−262.041	−267.790	−296.747	−298.149	−266.264	−266.022
DS4	50%		75%		90%	
	$\Lambda(t)$	$\Lambda_e(t)$	$\Lambda(t)$	$\Lambda_e(t)$	$\Lambda(t)$	$\Lambda_e(t)$
EXP	−272.217	−272.217	−322.470	−322.470	−346.961	−346.961
TSE	−403.711	−284.189	−334.324	−321.895	−350.246	−346.856
RAY	−799.785	−306.209	−363.309	−321.881	−356.112	−346.844
WEB	−516.521	−663.779	−323.093	−322.305	−347.244	−346.850

Table 5 Prediction performance based on predictive least squares error.

DS1	50%		75%		90%	
	$\Lambda(t)$	$\Lambda_e(t)$	$\Lambda(t)$	$\Lambda_e(t)$	$\Lambda(t)$	$\Lambda_e(t)$
EXP	1.050	1.050	2.492	2.492	2.550	2.550
TSE	3.807	1.050	4.330	2.645	1.053	2.550
RAY	4.721	1.050	5.090	2.741	0.630	2.549
WEB	0.762	1.047	2.342	2.477	2.366	2.549
DS2	50%		75%		90%	
	$\Lambda(t)$	$\Lambda_e(t)$	$\Lambda(t)$	$\Lambda_e(t)$	$\Lambda(t)$	$\Lambda_e(t)$
EXP	20.393	20.393	11.054	11.054	6.206	6.206
TSE	84.668	20.393	3.116	8.608	2.654	4.849
RAY	5.919	20.389	1.193	6.132	0.715	3.280
WEB	22.471	20.410	4.661	0.837	2.755	1.140
DS3	50%		75%		90%	
	$\Lambda(t)$	$\Lambda_e(t)$	$\Lambda(t)$	$\Lambda_e(t)$	$\Lambda(t)$	$\Lambda_e(t)$
EXP	1.988	1.988	8.224	8.224	1.333	1.333
TSE	46.197	2.085	2.081	8.542	1.698	0.954
RAY	11.352	2.017	0.841	8.986	2.733	0.898
WEB	2.745	2.171	8.316	8.602	0.898	2.072
DS4	50%		75%		90%	
	$\Lambda(t)$	$\Lambda_e(t)$	$\Lambda(t)$	$\Lambda_e(t)$	$\Lambda(t)$	$\Lambda_e(t)$
EXP	1.598	1.598	1.331	1.331	0.370	0.370
TSE	12.169	5.308	5.081	0.754	3.162	0.336
RAY	15.810	7.495	7.432	0.761	4.489	0.426
WEB	13.837	13.536	2.154	1.183	1.427	0.383

prediction result at each observation point, 50%, 75% and 90% of a whole data and calculate the PLL for both NHPP-based SRMs with the same underlying fault-detection time distribution. In the 50% observation, it is checked that the ED-NHPP-based SRMs with TSE (RAY) provide the larger PLL than the usual NHPP-based SRMs with all 4 (3) data sets. Similar to this case, in 75% (90%) points, the ED-NHPP-based SRMs with TSE and RAY outperform more than the usual NHPP-based SRMs in 2 and 2 (1 and 2) data sets, and the ED-NHPP-based SRMs with WEB do in 3 (3) data sets. Especially, our model with the equilibrium distributions provides the best prediction performance in DS4 in spite of their observation points. In Table 5 we present the prediction performance with the PLS. The prediction results with PLS are almost similar to ones with PLL, so that the ED-NHPP-based SRMs outperform in terms of the minimization of predictive least square error more than the usual NHPP-based SRMs with the same fault-detection time distributions in many cases.

In general, the goodness-of-fit performance to the past observations does not always link to the prediction performance. However, it should be noted that the ED-NHPP-based SRMs can possess nice prediction abilities as well as the goodness-of-fit abilities. Another advantage of the ED-NHPP-based SRMs over the usual ones is that they provide considerably better prediction performance in the 50% observation point. This implies that our models are much helpful in the ealier period of the test phase. In the long history of software reliability engineering it has been known that there was no uniquely best SRM which could be fitted to all the software failure data. In other words, the software reliability research suggests that the SRM used for analysis strongly depends on the kind of software fault data. In that sense, we can recommend that the ED-NHPP-based SRMs with the representative fault-detection time distributions should be applied at the same time when the usual NHPP-based SRMs are tried.

6. Concluding Remarks

In this paper we have proposed a new modeling framework for the NHPP-based SRMs by using the equilibrium distribution of fault-detection time. Moreover, we have provided stable procedures for estimating the parameters of the proposed models based on the EM algorithm. In the numerical examples with 4 real software fault data we compared our ED-NHPP-based SRMs with the existing ones. The numerical results have shown that the goodness-of-fit performance for the proposed SRMs were rather stable and could sometimes outperform the existing SRMs. As we mentioned in Sect. 1, in the past literature as well as the software reliability assessment practice, the exponential SRM [1], [6] were implicitly assumed without careful verification of its probabilistic characteristics, if the underlying software fault data behaves like a concave curve. This is obviously an inappropriate approach since it may be possible to find a better concave curve from the ED-NHPP-based SRMs by assuming different fault-detection time distribution.

It is worth mentioning that our ED-NHPP-based SRMs proposed in this paper have the same number of model parameters as the corresponding existing NHPP-based SRMs. In the past literature, considerable attentions have been paid to select the fault-detection time distribution in the mean value function. Although we have just treated 4 types of ED-NHPP-based SRMs in this paper, of course, the other types of distribution can be applied to build the ED-NHPP-based SRMs. In future, we will study the other types of ED-NHPP-based SRM with more flexible distributions like Hyper-Erlang distribution, and investigate their applicability to the actual software reliability evaluation.

References

[1] A.A. Abdel-Ghaly, P.Y. Chan, and B. Littlewood, "Evaluation of competing software reliability predictions," IEEE Trans. Softw. Eng., vol.SE-12, no.9, pp.950–967, 1986.

[2] H. Akaike, "A new look at the statistical model identification," IEEE Trans. Autom. Control, vol.AC-19, pp.716–723, 1974.

[3] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," J. Royal Statistical Society, Series B, vol.B-39, pp.1–38, 1977.

[4] T. Dohi, S. Osaki, and K.S. Trivedi, "An infinite server Queueing approach for describing software reliability growth — Unified modeling and estimation framework," Proc. 11th Asia-Pacific Software Engineering Conference, pp.110–119, 2004.

[5] A.L. Goel and K. Okumoto, "Time-dependent error-detection rate model for software reliability and other performance measures," IEEE Trans. Reliab., vol.R-28, no.3, pp.206–211, 1979.

[6] A.L. Goel, "Software reliability models: Assumptions, limitations and applicability," IEEE Trans. Softw. Eng., vol.SE-11, no.12, pp.1411–1423, 1985.

[7] R.C. Gupta, "Role of equilibrium distribution in reliability studies," Probability in the Engineering and Informational Sciences, vol.21, pp.315–334, 2007.

[8] Z. Jelinski and P.B. Moranda, "Software reliability research," in Statistical Computer Performance Evaluation, ed. W. Freiberger, pp.465–484, Academic Press, New York, 1972.

[9] N. Langberg and N.D. Singpurwalla, "Unification of some software reliability models," SIAM J. Sci. Comput., vol.6, pp.781–790, 1985.

[10] M.R. Lyu, Handbook of Software Reliability Engineering, McGraw-Hill, New York, 1996.

[11] J.D. Musa, A. Iannino, and K. Okumoto, Software Reliability, Measurement, Prediction, Application, McGraw-Hill, New York, 1987.

[12] H. Okamura, Y. Watanabe, and T. Dohi, "An iterative scheme for maximum likelihood estimation in software reliability modeling," Proc. 14th International Symposium on Software Reliability Engineering, pp.246–256, 2003.

[13] H. Pham, Software Reliability, Springer, Singapore, 2000.

[14] G. Schwarz, "Estimating the dimension of a model," Ann. Statist., vol.6, pp.461–464, 1978.

[15] X. Xiao and T. Dohi, "On equilibrium distribution properties in software reliability modeling," Proc. 4th International Conference on Availability, Reliability and Security, pp.158–165, 2009.

[16] M. Xie, Software Reliability Modelling, World Scientific, Singapore, 1991.

[17] S. Yamada, M. Ohba, and S. Osaki, "S-shaped reliability growth modeling for software error detection," IEEE Trans. Reliab., vol.R-32, no.5, pp.475–484, 1983.

Xiao Xiao received the B.Sc. (Engineering) and M.Sc. (Engineering) from Hiroshima University, Japan, in 2008 and 2010, respectively. She is now working as a Ph.D. candidate in Graduate School of Engineering, Hiroshima University. She is a Student Member of ORSJ.

Hiroyuki Okamura received the B.Sc. (Engineering), M.Sc. (Engineering) and Ph.D. (Engineering) from Hiroshima University, Japan, in 1995, 1997 and 2001, respectively. Since 2003, he has been working as an Associate Professor in the Department of Information Engineering, Graduate School of Engineering, Hiroshima University.

Tadashi Dohi received the B.Sc. (Engineering), M.Sc. (Engineering) and Ph.D. (Engineering) from Hiroshima University, Japan, in 1989, 1991 and 1995, respectively. Since 2002, he has been working as a Full Professor in the Department of Information Engineering, Graduate School of Engineering, Hiroshima University. He is a Regular Member of ORSJ, JSIAM, REAJ and IEEE.