Ann Oper Res (2019) 277:83–93
https://doi.org/10.1007/s10479-017-2486-3

RELIABILITY AND QUALITY MANAGEMENT IN STOCHASTIC SYSTEMS

A generalized software reliability model with stochastic fault-detection rate

Triet Pham[1] · **Hoang Pham**[2]

Published online: 31 March 2017

Abstract We propose a theoretic model of software reliability where the fault detection rate is a stochastic process. This formulation provides the flexibility in modeling the random environment effects in testing software data. We examine two particular cases: additive and multiplicative noise and provide explicit representations for the expected number of software failures. Examples are included to demonstrate the formulas for specific choices of time dependent total number of faults and distribution of noise.

Keywords Non-homogeneous Poisson process · Software reliability model · Stochastic fault-detection rate

1 Introduction

Many existing non-homogeneous Poisson process (NHPP) software reliability model have been carried out through the fault intensity rate function $h(t)$ and the mean value function $m(t)$ within a controlled testing environment to estimate reliability metrics such as the number of residual faults, failure rate, and reliability of software. Specifically, the mean value function is modeled as

$$\frac{d}{dt}m(t) = h(t)(N(t) - m(t)), \tag{1.1}$$

where $m(t)$ is the number of software failures detected by time t, $N(t)$ is the expected number of faults that exist in the software before testing and $h(t)$ is the time dependent fault detection rate per unit of time. The goal of these models is to achieve an explicit formula for $m(t)$ which

✉ Triet Pham
tmp140@math.rutgers.edu

Hoang Pham
hopham@rci.rutgers.edu

[1] Department of Mathematics, Rutgers University, Pisctacaway, NJ, USA

[2] Department of Industrial Engineering, Rutgers University, Pisctacaway, NJ, USA

is applied to the software testing data and used to make predictions on the software failures and reliability in the field.

We observe, however, that model (1.1) is deterministic, while the operating environments in the field are random. Specifically, there are factors of the operating environments that will affect the software failure and software reliability in an unpredictable way. Many authors have worked in different directions to extend the model (1.1) to account for the random environment effects. Teng and Pham (2006) discussed a generalized model that captures the uncertainty of the environment and its effects upon the software failure rate. Other researchers (Chang et al. 2014; Goel and Okumoto 1979; Inoue et al. 2015, 2016; Kapur et al. 2011, 2012, 2014; Kumar et al. 2016; Lee et al. 2016; Liu et al. 2016; Minamino et al. 2016; Ohba 1984; Okamura and Dohi 2016; Ohba and Yamada 1984; Persona et al. 2010; Pham and Pham 2000; Pham 1993, 1996, 2006, 2007, 2013, 2014a, b, 2016; Pham and Deng 2003; Pham et al. 1999, 2014; Pham and Zhang 1997, 2003; Roy et al. 2014; Sgarbossa and Pham 2010; Sato and Yamada 2016; Teng and Pham 2006, 2004; Xiao and Dohi 2011; Yamada and Osaki 1985; Yamada et al. 1992; Zhang and Pham 2006; Zhu and Pham 2016; Zhu et al. 2015) have also developed reliability and cost models incorporating both testing phase and operating phase in the software development cycle for estimating the reliability of software systems in the field. citeP1 recently also developed a software reliability model with Vtub-shaped fault-detection rate subject to the uncertainty of operating environments.

One way we can capture the random environment effects abstractly is by letting $h(t, \omega)$ be a stochastic process adapted to a given filtration $\mathcal{F}(t)$, $0 \leq t < \infty$:

$$\frac{dm(t,\omega)}{dt} = h(t,\omega)(N(t) - m(t,\omega)). \tag{1.2}$$

Under the assumptions that $N(t) = N$ and $m(0) = 0$, one can derive the solution for (1.2) as

$$m(t,\omega) = N\left[1 - e^{-\int_0^t h(s,\omega)ds}\right]. \tag{1.3}$$

As $m(t, \omega)$ is a random variable, the interest now is to obtain an explicit representation of

$$\bar{m}(t) := E(m(t,\omega)). \tag{1.4}$$

The general formulation of (1.2) allows for flexibility in modelling the random environment effect via the term $h(t, \omega)$. For example, Pham (2014a) models

$$h(t,\omega) = \eta(\omega)\tilde{h}(t) \tag{1.5}$$

where $\eta(\omega)$ is a Gamma(α, β) random variable and $\tilde{h}(t)$ is a Vtub-shaped rate function. We will refer to this formulation as a (static) multiplicative noise model. Under these assumptions,

$$\bar{m}(t) = N\left[1 - \left(\frac{\beta}{\beta + \int_0^t \tilde{h}(s)ds}\right)^{\alpha}\right]. \tag{1.6}$$

One can also generalize the model in Pham (2014a) by considering

$$h(t,\omega) = \eta(t,\omega)\tilde{h}(t), \tag{1.7}$$

where $\eta(t, \omega)$ is a well-known stochastic process adapted to a given filtration and $\tilde{h}(t)$ is a deterministic rate function. We refer to this formulation as a dynamic multiplicative noise model. A possible choice would be $\eta(t, \omega) = W(t, \omega)$ where $W(t, \omega)$ is a Brownian motion.

We do not pursue the dynamic multiplicative noise model and its implications in this paper. Instead, we propose a dynamic additive noise model:

$$h(t,\omega) = \tilde{h}(t) + \dot{M}(t,\omega). \tag{1.8}$$

The goal in this formulation is to choose $\dot{M}(t,\omega)$ such that $M(t,\omega)$, the (formal) anti-derivative of $\dot{M}(t,\omega)$ with respect to t, is a martingale. That is for all $s < t$

$$E(M(t,\omega)|\mathcal{F}(s)) = M(s,\omega), \tag{1.9}$$

where $\mathcal{F}(t), t \geq 0$ is a given filtration representing the flow of information in time. Equation (1.8) and the martingale property of $M(t,\omega)$ implies that for all $u < t$

$$E\left(\int_u^t h(s,\omega)ds\right) = \int_u^t \tilde{h}(s)ds. \tag{1.10}$$

This captures the intuition that on average the random effect of $h(s,\omega)$ cancels out to give us the equivalene of the original deterministic rate function $\tilde{h}(s)$. We note that relation (1.10) does not trivialize the model in the sense of reducing $\bar{m}(t)$ to the deterministic case since in general $E(e^{\int_u^t h(s,\omega)ds}) \neq e^{E(\int_u^t h(s,\omega)ds)}$.

Our main contribution of this paper is to provide a general theory for the software reliability model with stochastic fault-detection rate. The results when the rate follows a dynamic additive white noise model are presented in Sect. 2.1. To the best of our knowledge, the martingale framework, in particular the Brownian motion and white noise processes has not been utilized in the software reliability literature to model the random environment effect. We also extend the theory of the static multiplicative rate (1.5) in the model (1.2) by removing the assumption that $N(t)$ is constant. The results are presented in Sect. 2.2. Section 3 concludes the paper.

Notation

$m(t,\omega)$: number of software failures detected by time t
$\bar{m}(t)$: expected number of software failures detected by time t
$N(t)$: expected number of faults that exist in the software before testing
$h(t,\omega)$: time-dependent fault-detection rate per unit of time
$\phi_\eta(t)$: moment generating function of the random variable η evaluated at t

2 General fault-detection model with stochastic rate

This section presents the main contribution of this paper, which is a general software reliability model where the fault-detection rate $h(t,\omega)$ is a stochastic process. We present specific modelling details of $h(t,\omega)$ in subsequent sections.

Consider the initial value problem

$$\begin{aligned}\frac{d}{dt}m(t,\omega) &= h(t,\omega)(N(t) - m(t,\omega))\\ m(0) &= 0.\end{aligned} \tag{2.1}$$

This equation has the solution

$$m(t,\omega) = e^{-\int_0^t h(s,\omega)ds}\int_0^t e^{\int_0^u h(s,\omega)ds}h(u)N(u)du. \tag{2.2}$$

Assuming $N(t)$ is differentiable and using integration by parts, we have

$$m(t,\omega) = \left\{ N(t) - N(0)e^{-\int_0^t h(s,\omega)ds} \right\} - \int_0^t e^{-\int_u^t h(s,\omega)ds} N'(u)du. \tag{2.3}$$

In this expression, the randomness of $m(t,\omega)$ comes from $\int_0^t h(s,\omega)ds$ and $\int_u^t h(s,\omega)ds$. Thus the explicit representation of $\bar{m}(t)$ depends on the computation of

$$E\left(e^{-\int_u^t h(s,\omega)ds}\right), u \geq 0. \tag{2.4}$$

In Pham (2014a), the author facilitates this computation by imposing the multiplicative noise structure

$$h(t,\omega) = \eta(\omega)\tilde{h}(t). \tag{2.5}$$

It follows that

$$E\left(e^{-\int_u^t h(s,\omega)ds}\right) = \phi_\eta(-\int_u^t \tilde{h}(s)ds), \tag{2.6}$$

where $\phi_\eta(t)$ is the moment generating function of $\eta(\omega)$. On the other hand, the moment generating function $\phi_\eta(t)$ also poses difficulty in deriving the explicit representation of $\bar{m}(t)$ when we generalize the expected number of faults N to a non-constant function $N(t)$. Indeed as the examples in Sects. 2.2.1 and 2.2.2 show, we need to choose a suitable $N(t)$ to achieve such goal.

2.1 Fault-detection rate model with additive white noise

In this section, we utilize the martingale framework to model the random environment effect. For the convenience of the readers, we present the basic definitions of a martingale and a Brownian motion. For a thorough introduction we refer the readers to Protter (2013).

A stochastic process $M(t)$ is a martingale with respect to a filtration $\mathcal{F}(t)$, $0 \leq t \leq T$ if for any $s \leq t$

$$E(M(t)|\mathcal{F}(s)) = M(s). \tag{2.7}$$

We note that by convention we suppress the dependence of $M(t)$ (and later on, the Brownian motion $W(t)$) on ω. That is we write $M(t)$ instead of $M(t,\omega)$ (respectively $W(t)$ instead of $W(t,\omega)$) . For a continuous martingale $M(t)$, the quadratic variation process $[M](t)$ is such that $M^2(t) - [M](t)$ is also a $\mathcal{F}(t)$ martingale.

A particular and well-known example of a martingale is a Brownian motion. A stochastic process $W(t)$ is a Brownian motion with respect to a filtration $\mathcal{F}(t)$, $0 \leq t \leq T$ if

$$\begin{aligned} &W(0) = 0 \\ &W(t) - W(s) \text{ is independent of } \mathcal{F}(s), \text{ for any } s < t \\ &W(t) - W(s) \text{ has } N(0, t-s) \text{ distribution.} \end{aligned} \tag{2.8}$$

For our purpose, if $M(t)$ is a continuous martingale with suitable integrability condition then

$$E\left(e^{M(t)-M(u)}\right) = E\left(e^{\frac{1}{2}([M](t)-[M](u))}\right). \tag{2.9}$$

We can take advantage of this relation by using a martingale $M(t)$ where $[M](t)$ has a simple expression. For example, if $W(t)$ is a Brownian motion then $W(t)$ is also a martingale (with respect to its own filtration) and $[W](t) = t$.

To utilize the exponential martingale structure, we assume that

$$\int_0^t h(s,\omega)ds = M(t) + \int_0^t \tilde{h}(s)ds, \tag{2.10}$$

where $\tilde{h}(s)$ is a deterministic rate function. Equivalently

$$h(t,\omega) = \dot{M}(t) + \tilde{h}(t), \tag{2.11}$$

where $\dot{M}(t)$ formally denotes the derivative of M with respect to time. The technical difficulty here is that a non-trivial continuous martingale would almost surely have paths that are nowhere differentiable. Thus one has to be careful to interpret the process $h(t,\omega)$ in this formulation. For our starting point, we can choose $M(t) = W(t)$ to be a Brownian motion because the process $\dot{W}(t)$ is well-studied in the literature : it is referred to as the white noise process. We will investigate other possible choices for $M(t)$, including the martingale with jumps case in future works.

Thus we consider the model

$$\begin{aligned} \frac{d}{dt}m(t,\omega) &= h(t,\omega)(N(t) - m(t,\omega)) \\ m(0) &= 0. \end{aligned} \tag{2.12}$$

where

$$h(t,\omega) = \tilde{h}(t) + \sqrt{2}\dot{W}(t). \tag{2.13}$$

Here $\tilde{h}(t)$ is a usual deterministic rate process and $\dot{W}(t)$ is the white noise process. In other words, $\dot{W}(t)$ is a Gaussian process with covariance structure

$$E(\dot{W}(s)\dot{W}(t)) = \delta(t-s), 0 < s < t; \tag{2.14}$$

where $\delta(t)$ is the Dirac Delta measure. For more details about white noise processes, we refer the readers to Martin (2009). In our model, the structure (2.13) has the effect that

$$\int_u^t h(t,\omega)ds = \int_u^t \tilde{h}(s)ds + \sqrt{2}(W(t) - W(u)), \tag{2.15}$$

where $W(t)$ is a Brownian motion. Plugging this expression in (2.3), we have the computation of $\bar{m}(t)$ as followed

$$\begin{aligned} \bar{m}(t) &= \left\{N(t) - N(0)e^{-\int_0^t \tilde{h}(s)ds}E(e^{-\sqrt{2}W(t)})\right\} \\ &\quad - \int_0^t e^{-\int_u^t \tilde{h}(s)ds}E(e^{-\sqrt{2}(W(t)-W(u))})N'(u)du \\ &= \left\{N(t) - N(0)e^{-\int_0^t \tilde{h}(s)ds}e^t\right\} - \int_0^t e^{-\int_u^t \tilde{h}(s)ds}e^{t-u}N'(u)du \\ &= \left\{N(t) - N(0)e^{-\int_0^t (\tilde{h}(s)-1)ds}\right\} - \int_0^t e^{-\int_u^t (\tilde{h}(s)-1)ds}N'(u)du. \end{aligned} \tag{2.16}$$

The expression (2.16) is suitable for the choice of $\tilde{h}(t)$ such that $\int \tilde{h}(t)dt$ is of the log type. We demonstrate this by several examples below.

Example 1 Let

$$\tilde{h}(t) = \frac{b}{1+\gamma e^{-bt}}. \tag{2.17}$$

This is the S-shape detection rate used in Pham et al. (1999). Then

$$\bar{m}(t) = \left\{ N(t) - N(0)\frac{\gamma + 1}{\gamma + e^{bt}} e^{t} \right\} - \frac{e^{t}}{\gamma + e^{bt}} \int_0^t \left(\gamma e^{-u} + e^{(b-1)u} \right) N'(u) du. \quad (2.18)$$

Let

$$N(t) = ke^{t}$$

then

$$\bar{m}(t) = k\left\{ e^{t} - \frac{\gamma + 1}{\gamma + e^{bt}} e^{t} \right\} - \frac{ke^{t}}{\gamma + e^{bt}} \left(\gamma t + \frac{e^{bt} - 1}{b} \right). \quad (2.19)$$

Example 2 Let

$$\tilde{h}(t) = \frac{bt}{1 + bt}, b > 0. \quad (2.20)$$

Letting $\tilde{b} = \frac{1}{b}$, we have

$$\tilde{h}(t) = \frac{t}{t + \tilde{b}}. \quad (2.21)$$

Then

$$\bar{m}(t) = \left\{ N(t) - N(0) \left(1 + \frac{t}{\tilde{b}} \right)^{\tilde{b}} \right\} - (t + \tilde{b})^{\tilde{b}} \int_0^t \frac{N'(u)}{(u + \tilde{b})^{\tilde{b}}} du. \quad (2.22)$$

Let

$$N(u) = \log(u + \tilde{b})$$

or

$$N'(u) = \frac{1}{u + \tilde{b}}$$

then

$$\bar{m}(t) = \log(t + \tilde{b}) - \left(\log(\tilde{b}) + \frac{1}{\tilde{b}} \right) \left(1 + \frac{t}{\tilde{b}} \right)^{\tilde{b}} + \frac{1}{\tilde{b}}.$$

2.2 Fault-detection rate model with static multiplicative noise

We consider the model

$$\begin{aligned} \frac{d}{dt} m(t, \omega) &= h(t, \omega)(N(t) - m(t, \omega)) \\ m(0) &= 0. \end{aligned} \quad (2.23)$$

where

$$h(t, \omega) = \eta(\omega)\tilde{h}(t). \quad (2.24)$$

In this section, we present the explicit representations of $\bar{m}(t)$ under the choice of $\tilde{h}(t)$ as the S-shaped rate function

$$\tilde{h}(t) = \frac{b}{1 + \gamma e^{-bt}}, \quad (2.25)$$

assuming certain form of $N(t)$. We have

$$m(t,\omega) = N(t) - N(0)e^{-\eta\int_0^t h(s)ds} - \int_0^t e^{-\eta\int_u^t h(s)ds} N'(u)du \tag{2.26}$$

And thus

$$\bar{m}(t) = N(t) - N(0)\phi_\eta(-\int_0^t h(s)ds) - \int_0^t \phi_\eta(-\int_u^t h(s)ds)N'(u)du, \tag{2.27}$$

where $\phi_\eta(t)$ is the moment generating function of the random variable $\eta(\omega)$. For convenience of notation, we introduce the function

$$H(t) := \int_0^t h(s)ds = \log\left(1 + \frac{e^{bt}}{\gamma}\right) - \log\left(1 + \frac{1}{\gamma}\right). \tag{2.28}$$

Thus $\bar{m}(t)$ can be expressed as

$$\bar{m}(t) = N(t) - N(0)\phi_\eta(-H(t)) - \int_0^t \phi_\eta(H(u) - H(t))N'(u)du. \tag{2.29}$$

2.2.1 $N(t) = H(t) + N(0)$

Under the assumption that $N(t) = H(t) + N(0)$, we have $N'(t) = h(t)$. This assumption still ensures that $N(t)$ is increasing since $N'(t) = h(t) > 0$. Equation (2.27) becomes

$$\bar{m}(t) = \left\{N(t) - N(0)\phi_\eta(-H(t))\right\} - \int_0^{H(t)} \phi_\eta(-u)du. \tag{2.30}$$

Example 3 Let $\eta(\omega)$ follows a Gamma(α, β) distribution. That is the probability density function of $\eta(\omega)$ is given by

$$g(x) = \frac{\beta^\alpha x^{\alpha-1}e^{-\beta x}}{\Gamma(\alpha)}; quad \alpha, \beta > 0, x > 0. \tag{2.31}$$

We have

$$\phi_\eta(t) = \left(1 - \frac{t}{\beta}\right)^{-\alpha}, \tag{2.32}$$

and

$$\int_0^t \phi_\eta(-u)du = \beta\frac{\left(1 + \frac{t}{\beta}\right)^{1-\alpha} - 1}{1-\alpha}. \tag{2.33}$$

Thus

$$\bar{m}(t) = \left\{N(t) - N(0)(1 + \frac{H(t)}{\beta})^{-\alpha}\right\} - \beta\frac{\left(1 + \frac{H(t)}{\beta}\right)^{1-\alpha} - 1}{1-\alpha}, \tag{2.34}$$

where $N(t) = H(t) + N(0)$ and $H(t)$ is given by (2.28).

Remark 1 There are 5 parameters in example (3): b, γ from $h(t)$, α, β from the distribution of η and $N(0)$. One can choose these parameters to fit a given existing data by standard methods such as maximum likelihood or least square estimations.

2.2.2 $N(t) = G(H(t))$

Under the assumption that $N(t) = G(H(t))$ and $G(x)$ is differentiable, we have

$$N'(t) = G'(H(t))h(t).$$

In particular, when $G(x) = x + N(0)$ this reduces to the assumption we have in section (2.2.1). To ensure that $N(t)$ is increasing, we also require that

$$g(x) := G'(x) > 0.$$

Now equation (2.27) becomes

$$\begin{aligned}\bar{m}(t) &= \left\{N(t) - N(0)\phi_\eta(-H(t))\right\} - \int_0^{H(t)} \phi_\eta[-(H(t)-u)]g(u)du \\ &= \left\{N(t) - N(0)\phi_\eta(-H(t))\right\} - (\hat{\phi}_\eta * g)(H(t)),\end{aligned}$$

where $\hat{\phi}_\eta(u) = \phi_\eta(-u)$ and $*$ denotes the convolution operator:

$$f * g(t) := \int_0^t f(t-s)g(s)ds. \tag{2.35}$$

Remark 2 The challenge in finding the explicit representation for (2.35) is to compute the convolution $\hat{\phi}_\eta * g(t)$. In the following examples, we choose certain combinations of η and g that allow for such computation.

Example 4 Let η follows a Gamma(α, β) in Example (3) with α an integer. Let $g(t) = t$. Then

$$\hat{\phi}_\eta * g(t) = \beta^2 \frac{\left(1 + \frac{t}{\beta}\right)^{2-\alpha} - (1 + \frac{1}{\beta})^{2-\alpha}}{(1-\alpha)(2-\alpha)} - \frac{\beta\left(1 + \frac{1}{\beta}\right)^{1-\alpha}}{1-\alpha}. \tag{2.36}$$

Thus

$$\begin{aligned}\bar{m}(t) = &\left\{N(t) - N(0)(1 + \frac{H(t)}{\beta})^{-\alpha}\right\} - \beta^2 \frac{\left(1 + \frac{H(t)}{\beta}\right)^{2-\alpha} - (1 + \frac{1}{\beta})^{2-\alpha}}{(1-\alpha)(2-\alpha)} \\ &+ \frac{\beta(1 + \frac{1}{\beta})^{1-\alpha}}{1-\alpha} \\ H(t) &= \log\left(1 + \frac{e^{bt}}{\gamma}\right) - \log\left(1 + \frac{1}{\gamma}\right).\end{aligned} \tag{2.37}$$

Example 5 Let η follows a Normal(μ, σ^2) distribution. Then

$$\phi_\eta(t) = e^{\mu t + \frac{\sigma^2 t^2}{2}}. \tag{2.38}$$

Let $g(t) = e^{\alpha t - \frac{\sigma^2 t^2}{2}}$. Then

$$\hat{\phi}_\eta * g(t) = \frac{e^{\alpha t - \frac{\sigma^2 t^2}{2}} - e^{-\mu t + \frac{\sigma^2 t^2}{2}}}{\mu + \alpha - \sigma^2 t}. \tag{2.39}$$

Thus

$$\bar{m}(t) = \left\{ N(t) - N(0)e^{-\mu H(t) + \frac{\sigma^2 H^2(t)}{2}} \right\} - \frac{e^{\alpha H(t) - \frac{\sigma^2 H(t)^2}{2}} - e^{-\mu H(t) + \frac{\sigma^2 H(t)^2}{2}}}{\mu + \alpha - \sigma^2 H(t)} \tag{2.40}$$

$$H(t) = \log\left(1 + \frac{e^{bt}}{\gamma}\right) - \log\left(1 + \frac{1}{\gamma}\right). \tag{2.41}$$

In particular, if $\mu = \alpha = 0$ and $\sigma = \sqrt{2}$

$$\hat{\phi}_\eta * g(t) = \frac{\sinh(t^2)}{2t}, \tag{2.42}$$

and

$$\bar{m}(t) = \left\{ N(t) - N(0)e^{-H^2(t)} \right\} - \frac{\sinh(H^2(t))}{2H(t)}. \tag{2.43}$$

Example 6 Let η follows a Poisson(λ) distribution. Then

$$\phi_\eta(t) = e^{\lambda(e^t - 1)}. \tag{2.44}$$

Let $g(t) = e^t$. Then

$$\hat{\phi}_\eta * g(t) = \frac{e^{t-\lambda}}{\lambda}\left(e^\lambda - e^{\lambda e^{-t}}\right). \tag{2.45}$$

Thus

$$\bar{m}(t) = N(t) - N(0)e^{\lambda(e^{-H(t)} - 1)} - \frac{e^{H(t)-\lambda}}{\lambda}\left(e^\lambda - e^{\lambda e^{-H(t)}}\right) \tag{2.46}$$

$$H(t) = \log\left(1 + \frac{e^{bt}}{\gamma}\right) - \log\left(1 + \frac{1}{\gamma}\right). \tag{2.47}$$

Remark 3 The advantage of the assumption $N(t) = G(H(t))$ compared with $N(t) = H(t) + N(0)$ is that it allows us to take advantage of the choice of the function $g(x) = G'(x)$ in the calculation. In assuming $N(t) = H(t) + N(0)$ we need to compute $\int_0^t \phi_\eta(-u)du$ and there are not many choices of the distribution of η where we know this integral explicitly. In Sect. 2.2 we have seen one example where that is possible: the Gamma distribution. Examples (5) and (6) show instances where this computation is not known explicitly because it involves the anti-derivative of $\phi_\eta(t)$. On the other hand, by assuming a suitable $g(t)$ we can still carry out the calculation of $\hat{\phi}_\eta * g(t)$ and achieve a representation of $\bar{m}(t)$.

3 Conclusion and future works

We have presented the explicit representations for $\bar{m}(t)$ for two different models, one with dynamic additive noise in Sect. (2.1) and one with static multiplicative noise in section (2.2). In both cases, $N(t)$ is allowed to be a function of t. We observe that the model with additive noise is most suitable when the anti-derivative of the rate function $h(t)$ is of a log type. On the other hand, the multiplicative model is suitable when we have a nice moment generating function ϕ_η to work with.

For our future works, we will apply the models we have derived in this paper to an existing data and compare them under various fitting criteria. Because we allow $N(t)$ to be a function of t we hope to achieve a better fit in cases where N was traditionally assumed to be a constant. We will also investigate and generalize the additive and multiplicative noise structure mentioned in Sects. (2.2) and (2.1). One direction is to investigate the dynamic multiplicative noise model. For the additive model, we plan to consider other possible choices of $M(t)$, including the non-continuous case such as a compensated Poisson process.

So far in this paper we have limited the source of randomness to the rate function $h(t, \omega)$. One can also consider other places where randomness comes in, such as in the total number of faults $N(t)$. While $h(t, \omega)$ models the endogenous random aspects of the software testing process, $N(t, \omega)$ models the exogeneous aspects. Thus another direction we can take for our future work is to investigate the model

$$\frac{dm(t, \omega)}{dt} = h(t, \omega)(N(t, \omega) - m(t, \omega)), \tag{3.1}$$

were $h(t, \omega)$ and $N(t, \omega)$ follows some correlation structure, given by a function $\rho(t)$.

References

Chang, I. H., Pham, H., Lee, S. W., & Song, K. Y. (2014). A testing-coverage software reliability model with the uncertainty of operating environments. *International Journal of Systems Science*, *1*(4), 220–227.

Goel, A. L., & Okumoto, K. (1979). Time-dependent fault-detection rate model for software and other performance measures. *IEEE Transactions on Reliability*, *28*, 206–211.

Inoue, S., Ikeda, J., & Yamada, S. (2016). Bivariate change-point modeling for software reliability assessment with uncertainty of testing-environment factor. *Annals of Operations Research*, *244*(1), 209–220.

Inoue, S., Taniguchi, S., & Yamada, S. (2015). An all-stage truncated multiple change-point model for software reliability assessment. *International Journal of Reliability, Quality and Safety Engineering*, *22*(4), 1550017.

Kapur, P. K., Pham, H., Aggarwal, Anu G., Kaur, Gurjeet, & Kaur, Gurjeet. (2012). Two dimensional multi-release software reliability modeling and optimal release planning. *IEEE Transaction on Reliability*, *61*(3), 758–768.

Kapur, P. K., Pham, Hoang, Anand, Sameer, & Yadav, Kalpana. (2011). A unified approach for developing software reliability growth models in the presence of imperfect debugging and error generation. *IEEE Transaction on Reliability*, *60*(1), 331–340.

Kapur, P. K., Pham, H., Singh, J. N. P., & Sachdeva, N. (2014). When to stop testing multi-upgradations of software based on cost criteria. *International Journal of Systems Science*, *1*(2), 84–93.

Kumar, V., Mathur, P., Sahni, R., & Anand, M. (2016). Two-dimensional multi-release software reliability modeling for fault detection and fault correction processes. *International Journal of Reliability, Quality and Safety Engineering*, *23*(3), 1640002.

Lee, S. W., Chang, I. H., Pham, H., & Song, K. Y. (2016). A three-parameter fault-detection software reliability model with the uncertainty of operating environment. *Journal of Systems Science and Systems Engineering*,. doi:10.1007/s11518-016-5322-4.

Liu, Y., Li, D., Wang, L., & Hu, Q. (2016). A general modeling and analysis framework for software fault detection and correction process. *Software Testing, Verification and Reliability*, *26*(5), 351–365.

Martin, H. (2009). An introduction to stochastic PDEs. arXiv preprint arXiv:0907.4178.

Minamino, Y., Inoue, S., & Yamada, S. (2016). NHPP-based change-point modeling for software reliability assessment and its application to software development management. *Annals of Operations Research*, *244*(1), 85–101.

Ohba, M., & Yamada, S. (1984). S-shaped software reliability growth models. In *Proceedings of 4th international conference reliability and maintainability* (pp. 430–436).

Ohba, M. (1984). Inflexion S-shaped software reliability growth models. In S. Osaki & Y. Hatoyama (Eds.), *Stochastic models in reliability theory* (pp. 144–162). Berlin: Springer.

Okamura, H., & Dohi, T. (2016). Phase-type software reliability model: Parameter estimation algorithms with grouped data. *Annals of Operations Research*, *244*(1), 177–208.

Persona, A., Pham, Hoang, & Sgarbossa, F. (2010). Age replacement policy in random environment using systemability. *International Journal of Systems Science*, *41*(11), 1383–1397.

Pham, H. (1993). Software reliability assessment: Imperfect debugging and multiple failure types in software development. EG&G-RAAM-10737; Idaho National Engineering Laboratory.

Pham, H. (1996). A software cost model with imperfect debugging, random life cycle and penalty cost. *International Journal of Systems Science*, *27*(5), 455–463.

Pham, H. (2006). *System software reliability*. Berlin: Springer.

Pham, H. (2007). An imperfect-debugging fault-detection dependent-parameter software. *International Journal of Automation and Computing*, *4*(4), 325–328.

Pham, H. (August 2013). A software reliability model with Vtub-shaped fault-detection rate subject to operating environments. In *Proceedings of the19th ISSAT international conference on reliability and quality in design, Hawaii*

Pham, H. (2014a). A new software reliability model with Vtub-shaped fault-detection rate and the uncertainty of operating environments. *Optimization*, *63*(10), 1481–1490.

Pham, H. (2014b). Loglog fault-detection rate and testing coverage software reliability models subject to random environments. *Vietnam Journal of Computer Science*, *1*(1), 39–45.

Pham, H. (2016). A generalized fault-detection software reliability model subject to random operating environments. *Vietnam Journal of Computer Science*, *3*(3), 145–150.

Pham, H., & Deng, C. (August 2003). Predictive-ratio risk criterion for selecting software reliability models. In *Proceedings of the ninth international conference on reliability and quality in design*.

Pham, H., Nordmann, L., & Zhang, Z. (1999). A general imperfect-software-debugging model with S-shaped fault-detection rate. *IEEE Transactions on Reliability*, *48*(2), 169–175.

Pham, L., & Pham, H. (2000). Software reliability models with time-dependent hazard function based on Bayesian approach. *IEEE Transactions on Systems, Man, and Cybernetics: Part A*, *30*(1), 25–35.

Pham, H., Pham, D. H., & Pham, H, Jr. (2014). A new mathematical logistic model and its applications. *International Journal of Information and Management Sciences*, *25*, 79–99.

Pham, H., & Zhang, X. (1997). An NHPP software reliability model and its comparison. *International Journal of Reliability, Quality and Safety Engineering*, *4*(3), 269–282.

Pham, H., & Zhang, X. (2003). NHPP software reliability and cost models with testing coverage. *European Journal of Operational Research*, *145*, 443–454.

Protter, P. E. (2013). *Stochastic integration and differential equations* (Vol. 21). Springer.

Roy, P., Mahapatra, G. S., & Dey, K. N. (2014). An NHPP software reliability growth model with imperfect debugging and error generation. *International Journal of Reliability, Quality and Safety Engineering*, *21*(2), 1450008.

Sato, T., & Yamada, S. (2016). Analysis of process factors affecting software quality based on design review record and product metrics. *International Journal of Reliability, Quality and Safety Engineering*, *23*(4), 1650011.

Sgarbossa, F., & Pham, H. (2010). A cost analysis of systems subject to random field environments and reliability. *IEEE Transactions on Systems, Man, and Cybernetics Part C*, *40*(4), 429–437.

Teng, X., & Pham, H. (2004). Software cost model for quantifying the gain with considerations of random field environments. *IEEE Transactions on Computers*, *53*(3), 380–384.

Teng, X., & Pham, H. (2006). A new methodology for predicting software reliability in the random field environments. *IEEE Transactions on Reliability*, *55*(3), 458–468.

Xiao, X., & Dohi, T. (2011). Wavelet shrinkage estimation for non-homogeneous Poisson process based software reliability models. *IEEE Transactions on Reliability*, *60*(1), 211–225.

Yamada, S., Tokuno, K., & Osaki, S. (1992). Imperfect debugging models with fault introduction rate for software reliability assessment. *International Journal of Systems Science*, *23*(12), 2241–2252.

Yamada, S., & Osaki, S. (1985). Software reliability growth modeling: Models and applications. *IEEE Transactions on Software Engineering*, *11*, 1431–1437.

Zhang, X., & Pham, H. (2006). Software field failure rate prediction before software deployment. *Journal of Systems and Software*, *79*, 291–300.

Zhu, M., & Pham, H. (2016). A software reliability model with time-dependent fault detection and fault removal. *Vietnam Journal of Computer Science*, *3*(2), 71–79.

Zhu, M., Zhang, X., & Pham, H. (2015). A comparison analysis of environmental factors affecting software reliability. *Journal of Systems and Software*, *109*, 150–160.