New NHPP SRM Based on Generalized S-shaped Fault-Detection Rate Function

Nguyen Hung-Cuong(✉) and Huynh Quyet-Thang

Department of Software Engineering, School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi, Vietnam
cuongnh86@gmail.com, thanghq@soict.hust.edu.vn

Abstract. Software reliability modelling (SRM) is a mathematics technique to estimate some measures of computer system that relate to software reliability. One group of existing models is using non-homogeneous Poisson process (NHPP) whose fault-number and failure-rate are constant or time-dependent functions. A few studies have been manipulated S-shaped curve to construct their models. However, those works remain some limitations. In this study, we introduce a new model that is based on a generalised S-shaped curve and evaluate it by real data set. After installing it in real code of Matlab and using MLE method to estimate parameter with a range of initial solution, we prove that our model converge to the most basic model of NHPP group, Goel-Okumoto model.

Keywords: Software Reliability Modelling · Non-Homogeneous Poisson Process · S-shaped curve

1 Introduction

Having a big number of applications in many areas of our life, computer and software technology are being developed day by day. Like other sciences, researchers and developers have to solve a sequence of entangles to improve contribution of their products to human society. One of the biggest problems is to ensure the working state of software system, which is called software reliability and is considered as one characteristic of software quality [3]. Many authors [4,7,12] focus on software reliability modelling (SRM) to model system mathematically, in that they can estimate some characteristics of system as a total number of errors, predicted time of next failure, etc.. NHPP is a stochastic process whose rate parameter is a time-dependent function and is used widely in SRM research with plenty models [7].

We can use a S-shaped curve to mathematically model many natural processes that go to a steady state after an early growth period. Many authors have used this curve to build their model in software reliability modelling research and practice [6,7,9,10]. However, they have to face with two limitations: firstly, function that describes total number of faults of system is unbounded, in other words it approaches to infinity when time approaches infinity; secondly, failure detection rate functions in their study have simple form.

P.C. Vinh et al. (Eds.): ICTCC 2014, LNICST 144, pp. 212–221, 2015.
DOI: 10.1007/978-3-319-15392-6_21

To improve reality of NHPP SRMs that use S-shaped curve, we try to generalize existing S-shaped curve failure-detection-rate. So when their reality is increased, the complexity of computation also increased. In this study, we introduce a new model whose a failure detection rate function is generalised S-shaped curve. After theoretical computations, we use three materials to install our model: firstly, T project data set of AT&T [1] to apply; secondly, MLE method to estimate parameters; and the last, Matlab to support mathematics computing.

Organization of our paper is: we start by introduction section, we will discuss about NHPP SRM clearly in section 2. Section 3 will show some basic computation about S-shaped curve and apply in NHPP SRM. At the end, section 4 shows experimental results to evaluate our idea and section 5 summarizes our work with extended opinions.

2 NHPP Software Reliability Models

In this section, we will discuss about characteristic of SRMs base on NHPP, Pham [7].

2.1 General NHPP Software Reliability Model Calculation

Let's use some function to describe characteristic of system in Table 1.

Table 1. Characteristic functions of software system

a(t)	Total number of faults
b(t)	Fault detection rate
m(t)	Expected number of fault detected by time t (mean value function)
$\lambda(t)$	Failure intensity

By time t, system have $a(t)$ faults and $m(t)$ faults have been detected so we have $a(t)-m(t)$ remaining faults. With detection rate is $b(t)$, we have relationship among number of faults detected in period Δt, total remaining faults of system and fault detection rate:

$$m(t+\Delta t) - m(t) = b(t)[a(t) - m(t)]\Delta t + o(\Delta t) \tag{1}$$

where $o(\Delta t)$ is infinitesimal value with Δt: $\lim_{\Delta t \to 0} \frac{o(\Delta t)}{\Delta t} = 0$. Let $\Delta t \to 0$, we have:

$$\frac{\partial}{\partial t} m(t) = b(t)[a(t) - m(t)] \tag{2}$$

If t_0 is the starting time of testing process, with initial conditions $m(t_0) = m_0$ and $\lim_{t\to\infty} m(t) = a(t)$, Pham shows that general solution of (2) is [7]:

$$m(t) = e^{-B(t)} \Big[m_0 + \int_{t_0}^{t} a(\tau) b(\tau) e^{B(\tau)} d\tau \Big] \tag{3}$$

where

$$B(t) = \int_{t_0}^{t} b(s)ds \tag{4}$$

2.2 Existing NHPP SRMs

We have some existing NHPP SRMs shown in Table 2, Goel [2], Pham [7][8][9][10], Ohba [5], Yamada [13][14].

Table 2. Existing NHPP SRMs

Model	$a(t)$	$b(t)$	$m(t)$
Goel Okumoto	a (const)	b (const)	$a(1-e^{-bt})$
Inflection S-shaped (Ohba)	a (const)	$\frac{b}{1+\beta e^{-bt}}$	$a \times \frac{e^{bt}-1}{e^{bt}+\beta}$
Delayed S-shaped (Yamada)	a (const)	$\frac{b^2 t}{bt+1}$	$a[1-(1+bt)e^{-bt}]$
Yamada 1	ae^{at}	b (const)	$\frac{ab}{b+a} \times (e^{at} - e^{-bt})$
Yamada 2	$a(1+\alpha t)$	b (const)	$a(1-e^{-bt})(1-\frac{a}{\beta}) + a\alpha t$
PNZ	$a(1+\alpha t)$	$\frac{b}{1+\beta e^{-bt}}$	$\frac{a}{1+\beta e^{-bt}}[(1-e^{-bt})(1-\frac{a}{\beta}) + at]$
Pham exponential	$ae^{\beta t}$	$\frac{b}{1+ce^{-bt}}$	$\frac{ab}{b+\beta} \times \frac{e^{(\beta+b)t}-1}{e^{bt}+c}$
Pham-Zhang	$c + a(1-e^{-at})$	$\frac{b}{1+\beta e^{-bt}}$	$\frac{1}{1+\beta e^{-bt}}[(c+a)(1-e^{-bt}) - \frac{ab}{b-a}(e^{-at} - e^{-bt})]$
Pham fault detection dependent parameter	$a(1+bt)^2$	$\frac{b^2 t}{bt+1}$	$a(bt+1)(bt+e^{-bt}-1)$

2.3 Parameter Estimation Using MLE Method

We work with the second data type that records the individual times at which failure occurred. So given data is a set of t_i, or occurrence time of N observed failures. Our model can have some parameter, called θ generally. Using MLE method, we have the following equation that related to each parameter θ is [7]

$$\sum_{i=1}^{N} \frac{\frac{\partial}{\partial\theta}\lambda(t_i)}{\lambda(t_i)} - \frac{\partial}{\partial\theta} m(t_N) = 0 \tag{5}$$

Assump that our model have n parameter $\theta_1, \theta_2, \cdots, \theta_n$, we will have system of n equations with n variables. Solve it, we will get estimated parameter of our model.

2.4 Application of SRM

SRM is a stochastic technique to model a set of occurrence time of failure. After collect those set, known as data set, practitioner will apply one of SRMs to get his MLE system of equations. Solution of this system of equations is a estimator of the set of parameter of system, then we have numeric model. From this model, we can estimate some characteristic measures of software system as a total number of errors, predicted time of next failure, etc.. There are two problem with any SRMs. Firstly, different SRMs have own advantages and limitations, then practitioner have to decide what model will be chosen. Secondly, complex assumptions will make a better functions of model, so estimated measures of system will be better. But we have to face with complex computation when build it, for example system of MLE equations can not be solved manually.

3 Generalised S-shaped Fault-Detection-Rate Function

We will introduce S-shaped function and its computation when applying it into NHPP SRM.

3.1 Generalised S-shaped Function

Consider S-shaped function whose equation is:

$$f(t) = b \times \frac{1 + m \times e^{-\frac{t}{\tau}}}{1 + n \times e^{-\frac{t}{\tau}}} \tag{6}$$

where $m < n$. This function can be described as follows:

1. In early time, its rapid increasing depends on value of m, n and τ.
2. After this period, it goes to constant b: $\lim_{t\to\infty} f(t) = b$.

This function has been widely applied in many science areas, where it describes state of some processes in real life. We have a simple S-shaped function called *sigmoid* in equation (7) where $b = 1$, $m = 0$, $n = 1$, $\tau = 1$ and that is shown in Figure 1.

$$f(t) = \frac{1}{1 + e^{-t}} \tag{7}$$

3.2 Proposed Fault-Detection-Rate Function Base on Generalised S-shaped

As shown before, some existing SRMs used specific type of S-shaped function: Infection S-shaped of Ohba [6]; Pham exponential imperfect [11], PNZ [9] and Pham-Zhang [10] of Pham et al. In those studies, authors use:

$$b(t) = b \times \frac{1}{1 + \beta \times e^{-bt}} \tag{8}$$

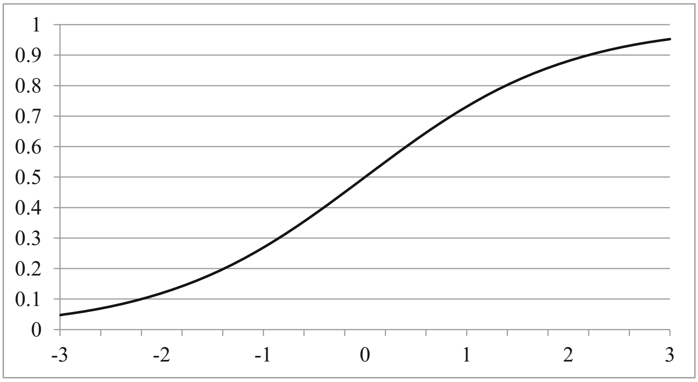

Fig. 1. Sigmoid function

We suggest using generalised S-shaped function as follows:

$$b(t) = b \times \frac{1 + m \times e^{-bt}}{1 + n \times e^{-bt}} \tag{9}$$

where $0 < m < n < +\infty$, $b > 0$. Obviously, $b(t)$ in (8) is a specific case of equation in (9) when $m = 0$. To evaluate the appearance of m, let $b = 1$ and $n = 1$, consider functions with $m = 0.05$ (with dashed line) and $m = 0.7$ (with normal line) that are presented in Figure 2. From this, we realize that m affect to the initial value and the increment of S-shaped function.

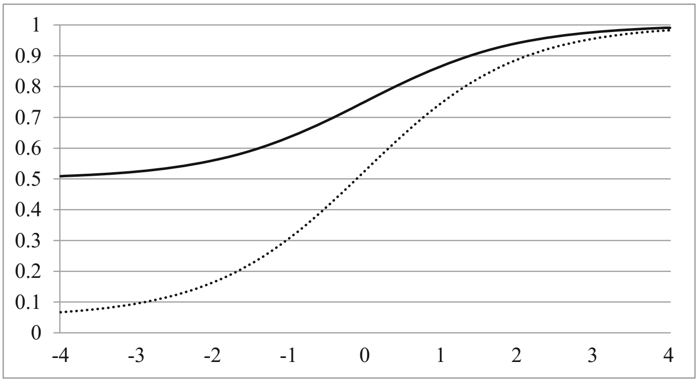

Fig. 2. Effect of parameter m to generalised S-shaped function

Let $k = \frac{m}{n} \Leftrightarrow m = k \times n$, note that $0 < k < 1$, we have:

$$b(t) = b \times \frac{e^{bt} + k \times n}{e^{bt} + n} \tag{10}$$

So:

$$B(t) = bkt + (1-k) \times \ln \frac{e^{bt}+n}{1+n} \tag{11}$$

And:

$$e^{B(t)} = e^{bkt} \times \left(\frac{e^{bt}+n}{1+n}\right)^{1-k} \tag{12}$$

3.3 Calculation of Mean Value Function of NHPP SRMs Base on General S-shaped Fault-Detection-Rate Function

From general solution in (3), substitute (10) and (12) into it, we have:

$$\int_0^t a(\tau)b(\tau)e^{B(\tau)}d\tau = \frac{b}{(1+n)^{1-k}} \times \int_0^t a(\tau) \times e^{bk\tau} \times \frac{e^{b\tau}+kn}{(e^{b\tau}+n)^k}d\tau \tag{13}$$

So:

$$m(t) = e^{-bkt} \times (e^{bt}+n)^{k-1} \times b \times \int_0^t a(\tau) \times e^{bk\tau} \times \frac{e^{b\tau}+kn}{(e^{b\tau}+n)^k}d\tau \tag{14}$$

Equation (15) shows the relationship between mean value function $m(t)$ and fault-number function $a(t)$. Being a basic case of fault-number function, we have:

$$a(t) = a \tag{15}$$

Substituting (16) into (15) we have:

$$m(t) = a - a \times e^{-bkt} \times \left(\frac{e^{bt}+n}{1+n}\right)^{k-1} \tag{16}$$

And

$$\lambda(t) = \frac{ab}{(1+n)^{k-1}} \times e^{-bkt}(e^{bt}+n)^{k-2}(e^{bt}+kn) \tag{17}$$

4 Experimental Results

4.1 Data Set

We use the failure data set of Ehrlich [1] to analysis our new SRM. This data set has been widely used in analysing and assessment SRMs. The data is testing data of project T that is developed in AT&T [1]. This system is a network management center that works as a connector between data collectors and operators [1]. Occurrence time of failures after testing period is given as in Table 3 [1]. The *Failure time* column provides exactly when each failure occur. The *Inter-failure time* column provides the length of period between two consecutive errors.

Table 3. AT&T system T project failures data set

Index	Failure time	Inter-failure time
1	5.50	5.50
2	7.33	1.83
3	10.08	2.75
4	80.97	70.89
5	84.91	3.94
6	99.89	14.98
7	103.36	3.47
8	113.32	9.96
9	124.71	11.39
10	144.59	19.88
11	152.40	7.81
12	166.99	14.60
13	178.41	11.41
14	197.35	18.94
15	262.65	65.30
16	262.69	0.04
17	388.36	125.67
18	471.05	82.69
19	471.50	0.46
20	503.11	31.61
21	632.42	129.31
22	680.02	47.60

4.2 Installing Environment

Our calculation is deployed in Thinkpad personal machine with technical information as follows:

- Processor Intel(R) Core(TM) i5-2410M CPU @2.30 GHz, 4.00 GB memory.
- Window 7 Professional Service Pack 1 operating system.
- Matlab R2012a stand alone version.

4.3 Parameter Estimation

Our new SRM has 4 parameter a, b, k and n. From equation (5), with derivation computations of $m(t)$ and $\lambda(t)$ in Appendix, we have system of 4 MLE equations. Those parameter can be estimated based on value of t_i of data set T that described in the first sub-section. Because this system of equations can not be solved primary, we need to use the support of Matlab tool.

We estimate parameters using `fsolve()`function of Matlab to solve MLE system of equations. Because there can have more than 1 solution, we use this Matlab statement with an array of initial solutions as:

- Initial solution vector begins from
  `[a,b,k,n]=[24,0.01,0.1,1]`

- Next initial solution vector of `[a,b,k,n]` is `[a+1,b+0.01,k+0.1,n+1]`
- To initial solution vector `[a,b,k,n]=[50,0.99,0.9,30]`

Furthermore, `optimset()` options is set to:

- `'MaxFunEvals'=1000000`
- `'MaxIter'=1000`

After 721710 loop instances, we have only one solution:

`[a*,b*,k*,n*]=[23.7451,0.00341519,1.0,0.0].`

From this calculation, we confirm that using MLE method to estimate parameters, our model will converge on Goel-Okumoto SRM with mean value function:

$$m(t) = a(1 - e^{-bt}) \tag{18}$$

From existing result, Pham [7] indicates that G-O SRM have good prediction value when comparing with other SRMs.

5 Conclusions and Future Works

In this paper, we have proposed new SRM by generalising existing S-shaped curve. Authors applied S-shaped functions in their and have many result. Based on the advantages of this curve, we expand their work by using generalised S-shaped function. However, the increment of reality of this generalising will make the computation more complex.

When apply our new model in real failure data set about project T of AT&T, with the computation support of Matlab, the result shows that this new SRM will converge to the most basic SRM, Goel-Okumoto.

For the further works, like any SRMs, let consider non-zero initial debugging time to get a better estimators. In addition, fault-number function of our model is constant, so some better time-dependent fault-number functions should be considered. The last extended idea is using other parameter estimations to get the set of parameters.

Appendix: Derivation Computations of $m(t)$ and $\lambda(t)$

We have some mathematics computations as follows:

- Derivation of $m(t)$ in each variables a, b, k and n:

$$\frac{\partial}{\partial a} m(t) = 1 - e^{-bkt} \times \left(\frac{e^{bt} + n}{1 + n}\right)^{k-1} \tag{19}$$

$$\frac{\partial}{\partial b}m(t) = \frac{b \times e^{-bkt}}{(1+n)^{k-1}} \times (e^{bt}+n)^{k-2} \times (e^{bt}+kn) \tag{20}$$

$$\frac{\partial}{\partial k}m(t) = -a \times e^{-bkt} \times \left(\frac{e^{bt}+n}{1+n}\right)^{k-1} \times \left[\ln\left(\frac{e^{bt}+n}{1+n}\right) - bt\right] \tag{21}$$

$$\frac{\partial}{\partial n}m(t) = e^{-bkt} \times (k-1) \times (e^{bt}+n)^{k-2} \times \frac{e^{bt}-1}{(n+1)^k} \tag{22}$$

- Derivation of $\lambda(t)$ in each variables a, b, k and n:

$$\frac{\partial}{\partial a}\lambda(t) = \frac{b \times e^{-bkt}}{(1+n)^{k-1}} \times (e^{bt}+n)^{k-2} \times (e^{bt}+kn) \tag{23}$$

$$\begin{aligned}\frac{\partial}{\partial b}\lambda(t) = \frac{-a \times e^{-bkt}}{(1+n)^{k-1}} \times (e^{bt}+n)^{k-3} \times \Big[kn^2(bkt-1) \\ +ne^{bt}(3bkt-k-bt-1) + e^{2bt}(bt-1)\Big]\end{aligned} \tag{24}$$

$$\begin{aligned}\frac{\partial}{\partial k}\lambda(t) = \frac{ab \times e^{-bkt}}{(1+n)^{k-1}} \times (e^{bt}+n)^{k-2} \\ \times\left[n - bte^{bt} - bknt + (e^{bt}+kn) \times \ln\left(\frac{e^{bt}+n}{1+n}\right)\right]\end{aligned} \tag{25}$$

$$\begin{aligned}\frac{\partial}{\partial n}\lambda(t) = \frac{-ab \times e^{-bkt}}{(1+n)^{k}} \times (k-1) \times (e^{bt}+n)^{k-3} \\ \times\Big[e^{bt}(kn-n-2) + e^{2bt} - kn\Big]\end{aligned} \tag{26}$$

So:

$$\frac{\frac{\partial}{\partial a}\lambda(t)}{\lambda(t)} = \frac{1}{a} \tag{27}$$

$$\frac{\frac{\partial}{\partial b}\lambda(t)}{\lambda(t)} = \frac{kn^2(bkt-1) + e^{2bt}(bt-1)}{-b(e^{bt}+n)(e^{bt}+kn)} + \frac{ne^{bt}(3bkt-k-bt-1)}{-b(e^{bt}+n)(e^{bt}+kn)} \tag{28}$$

$$\frac{\frac{\partial}{\partial k}\lambda(t)}{\lambda(t)} = \frac{(e^{bt}+kn) \times \ln\left(\frac{e^{bt}+n}{1+n}\right) - bte^{bt} + n - bknt}{e^{bt}+kn} \tag{29}$$

$$\frac{\frac{\partial}{\partial n}\lambda(t)}{\lambda(t)} = \frac{-(k-1)\Big[e^{bt}(kn-n-2) + e^{2bt} - kn\Big]}{(1+n)(e^{bt}+n)(e^{bt}+kn)} \tag{30}$$

Acknowledgments. This research was supported partly by The National Foundation for Science and Technology Development (NAFOSTED) under Grant 102.03-2013.39: Automated verification and error localization methods for component-based software.

The authors would like to thank Dr. Nguyen Thanh-Hung in Hanoi University of Science and Technology about his helpful comment; Le Yen, Hung-Cuong's colleague, about mathematical checking in early time of our work.

References

1. Ehrlich, W., Prasanna, B., Stampfel, J.: Determining the cost of a stop-test decision (software reliability). IEEE Software **10**(2), 33–42 (1993)
2. Goel, A.L., Okumoto, K.: Time-dependent error-detection rate model for software reliability and other performance measures. IEEE Transactions on Reliability **28**(3), 206–211 (1979)
3. Jung, H.-W., Kim, S.-G., Chung, C.-S.: Measuring software product quality: A survey of iso/iec 9126. IEEE Software **21**(5), 88–92 (2004)
4. Lyu, M.R., et al.: Handbook of software reliability engineering, vol. 3. IEEE Computer Society Press, CA (1996)
5. Ohba, M, Yamada, S.: S-shaped software reliability growth models. In: International Colloquium on Reliability and Maintainability, 4 th, Tregastel, France, pp. 430–436 (1984)
6. Ohba, M.: Software reliability analysis models. IBM Journal of Research and Development **28**(4), 428–443 (1984)
7. Pham, H.: System software reliability. Springer (2006)
8. Pham, H.: An imperfect-debugging fault-detection dependent-parameter software. International Journal of Automation and Computing **4**(4), 325–328 (2007)
9. Pham, H., Nordmann, L., Zhang, Z.: A general imperfect-software-debugging model with s-shaped fault-detection rate. IEEE Transactions on Reliability **48**(2), 169–175 (1999)
10. Pham, H., Zhang, X.: An nhpp software reliability model and its comparison. International Journal of Reliability, Quality and Safety Engineering **4**(03), 269–282 (1997)
11. Pham, L., Pham, H.: Software reliability models with time-dependent hazard function based on bayesian approach. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans **30**(1), 25–35 (2000)
12. Xie, M., Dai, Y.-S., Poh, K.-L.: Computing system reliability: models and analysis. Springer (2004)
13. Yamada, S., Ohba, M., Osaki, S.: S-shaped software reliability growth models and their applications. IEEE Transactions on Reliability **33**(4), 289–292 (1984)
14. Yamada, S., Osaki, S.: Software reliability growth modeling: Models and applications. IEEE Transactions on Software Engineering **12**, 1431–1437 (1985)