

# One-Sample Bayesian Predictive Analyses for an Exponential Non-Homogeneous Poisson Process in Software Reliability

Albert Orwa Akuno, Luke Akong'o Orawo, Ali Salim Islam

Department of Mathematics, Egerton University, Egerton, Kenya  
Email: [orwaakuno@gmail.com](mailto:orwaakuno@gmail.com), [orawo2000@yahoo.com](mailto:orawo2000@yahoo.com), [asislam54@yahoo.com](mailto:asislam54@yahoo.com)

Received 21 May 2014; revised 28 June 2014; accepted 12 July 2014

Copyright © 2014 by authors and Scientific Research Publishing Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY).  
<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

The Goel-Okumoto software reliability model, also known as the Exponential Nonhomogeneous Poisson Process, is one of the earliest software reliability models to be proposed. From literature, it is evident that most of the study that has been done on the Goel-Okumoto software reliability model is parameter estimation using the MLE method and model fit. It is widely known that predictive analysis is very useful for modifying, debugging and determining when to terminate software development testing process. However, there is a conspicuous absence of literature on both the classical and Bayesian predictive analyses on the model. This paper presents some results about predictive analyses for the Goel-Okumoto software reliability model. Driven by the requirement of highly reliable software used in computers embedded in automotive, mechanical and safety control systems, industrial and quality process control, real-time sensor networks, aircrafts, nuclear reactors among others, we address four issues in single-sample prediction associated closely with software development process. We have adopted Bayesian methods based on non-informative priors to develop explicit solutions to these problems. An example with real data in the form of time between software failures will be used to illustrate the developed methodologies.

## Keywords

Nonhomogeneous Poisson Process, Non-Informative Priors, Software Reliability Models, Bayesian Approach

---

## 1. Introduction

Over the last decade of the 20<sup>th</sup> century and the first few years of the 21<sup>st</sup> century, the demand for complex soft-

ware systems has gone high as it is seen that today, computers are embedded in automotive mechanical and safety control systems, industrial and quality process control, real-time sensor networks, aircrafts, nuclear reactors, hospital healthcare and air traffic control systems among others; computer systems have become an indispensable component of our modern society today. Consequently, the reliability of software used in these systems has been a major concern and a requirement in the modern generation. Software reliability is defined as the probability of failure-free software operations for a specified period of time in a specified environment [1]. A single software defect can cause system failure and to avoid these failures, reliable software is required. Software reliability is achieved through testing during the software development testing stage [2]. The usual criteria of removing bugs in software are by running test cases in a manner that exercises the software similar to the way that users will operate in their particular environment. However, emulating end-user environment during the test interval is difficult and time-consuming especially when there are multiple types of end-users and also, business pressure to release a software system within a tight market window puts a constraint on the amount of time that can be spent testing the software. Software reliability modeling comes in handy to address this dilemma. As indicated by [3], software reliability modeling can provide the basis for planning reliability growth tests, monitoring progress and estimating current reliability and forecasting and predicting future reliability improvements. Forecasting and prediction are achieved through predictive analyses. In particular, predictive analyses are useful in determining when to terminate the development process of software or hardware. Often, a prediction interval is constructed to provide the time frame when the  $k$ th ( $k > 0$ ) future failure observation will occur with a pre-determined confidence level [4].

Many software reliability models have been developed by various authors and researchers in the past three decades. Amongst, an Exponential Nonhomogeneous Poisson Process with intensity function

$$\lambda(t) = \alpha\beta e^{-\beta t} \quad (1)$$

is the earliest software reliability model to be developed by Goel and Okumoto in 1979. In various literatures, this NHPP is called the Goel-Okumoto (1979) model.

As noted by [5], the Goel-Okumoto (1979) model has been applied to a number of software testing environments and its application on assessing and detecting software failures has been investigated by various authors. For instance, the Goel-Okumoto model has been used to develop a statistical control mechanism that could be used to detect whether a software process is statistically under control or not. ML estimation of the parameters of the Goel-Okumoto (1979) model has been conducted and in particular, it has been shown that the ML estimates of the parameters of the model are not consistent as the testing period extends to infinity. [6] presented an empirical method for selecting software reliability growth models for release decision-making where they applied iteratively various software reliability models namely Goel-Okumoto (1979), Delayed S-shaped, Gompertz and Yamada exponential software reliability growth models to weekly cumulative software failure data during system test to determine the number of remaining failures expected in software after release. [7] also performed parameter estimation of the Goel-Okumoto, Yamada S-shaped and Inflection S-shaped software reliability growth models where they also established a necessary and sufficient condition with respect to the software failure data, of which, if satisfied, will ensure that the MLE method returns a unique positive and finite estimation of the unknown parameters of the Goel-Okumoto and the Yamada S-shaped models. [8] presented software failure data which, after study, depicted that the failure rate, *i.e.* the number of failures per hour, seemed to be decreasing with time, an indication that a Nonhomogeneous Poisson Process with mean value function  $m(t) = \alpha(1 - e^{-\beta t})$ , a mean value function corresponding to that of the Goel-Okumoto software reliability model, was a reasonable model to describe the failure process. From the literature, it is evident that most of the study that has been done on the Goel-Okumoto software reliability model is parameter estimation using, especially, the MLE method and model fit. There is a conspicuous absence of literature on both the classical and Bayesian predictive analyses on the model.

This paper focuses on single-sample predictive inference for the Goel-Okumoto (1979) software reliability model using Bayesian approach. We first identify four issues in the single-sample prediction associated closely with the development testing process of software and proceed to develop and derive the corresponding predictive distributions in Section 2. The main results for single-sample prediction are presented in Section 3. A real example in the form of secondary software failure data in the form of execution times between successive software failures is used to illustrate the proposed and developed methodologies in Section 4. A discussion is given in Section 5 and thereafter, mathematical proofs are given in the Appendix.

## 2. Predictive Issues and Bayesian Method

During the development testing stage of a software, statisticians and engineers are overly interested in various predictive problems whose solutions are believed to be very important in modifying, debugging and determining when to terminate software development testing process. In this section, we present four issues associated closely with software development testing process and derive the predictive distributions using Bayesian approach. For the purposes of the four predictive issues, we assume that a reliability growth testing is performed on a software and the cumulative number of failures of the software in the time interval  $(0, t]$ , denoted by  $N(t)$  is observed. We further assume that  $[N(t), t > 0]$  follows the NHPP with intensity function given in Equation (1). Let  $0 < t_1 < t_2 < \dots$  be the observed failure times. Failure data is said to be failure-truncated when testing stops after a predetermined  $n$  number of failures occur. We denote the  $n$  failure times by  $Y_{obs}^f = [t_i]_{i=1}^n$  where  $0 < t_1 < t_2 < \dots < t_n$ . Failure data is said to be time truncated if testing stops at a predetermined time  $t$ . We denote the corresponding observed failure data by  $Y_{obs}^t = [n, t_1, \dots, t_n; t]$ , where  $0 < t_1 < t_2 < \dots < t_n \leq t$ .

Prediction interval is a confidence interval for a future observation or a function of some future observations. Specifically, a double-sided (bilateral) prediction interval for  $t_{n+k}$  with confidence level  $\gamma$  is defined by  $[T_{n+k,L(\gamma)}, T_{n+k,U(\gamma)}]$  such that  $\Pr[T_{n+k,L(\gamma)} \leq t_{n+k} \leq T_{n+k,U(\gamma)}] = \gamma$ . Similarly, a single-sided (unilateral) lower or upper prediction limit for  $t_{n+k}$  with level  $\gamma$  is defined by  $T_{n+k,L(\gamma)}$  (or  $T_{n+k,U(\gamma)}$ ) which satisfies  $\Pr[T_{n+k,L(\gamma)} \leq t_{n+k}] = \gamma$  (or  $\Pr[t_{n+k} \leq T_{n+k,U(\gamma)}] = \gamma$ ). Both  $T_{n+k,L(\gamma)}$  and  $T_{n+k,U(\gamma)}$  depends only on a single sample (or a single software) and are called single-sample prediction limits. Prediction limits involving two samples (or two softwares) can be defined similarly and are called two-sample prediction limits.

### 2.1. Issues in Single-Sample Software Reliability Prediction

Here, we consider one software and assume that its cumulative inter-failure times obey the Goel-Okumoto (1979) software reliability model with observed data being either  $Y_{obs}^f$  or  $Y_{obs}^t$ . Based on  $Y_{obs}^f$  or  $Y_{obs}^t$ , we are interested in the following problems:

*Issue A:* what is the probability that at most  $k$  software failures will occur in the future time period  $(T, \tau]$  with  $\tau > T$ ?

*Issue B:* suppose that the pre-determined target value  $\lambda_{tv}$  for the failure rate of the software undergoing development testing is not achieved at time  $T$ , what is the probability that the target value  $\lambda_{tv}$  will be achieved at time  $\tau, \tau > T$ ?

*Issue C:* suppose that the target value  $\lambda_{tv}$  for the software failure rate is not achieved at time  $T$ , how long will it take so that the software failure rate will be attained at  $\lambda_{tv}$ ?

*Issue D:* what is the upper prediction limit (UPL) of  $\lambda_\tau = \alpha\beta e^{\beta\tau}$  with level  $\gamma$ ,  $\tau$  being a predetermined value greater than  $T$ ?

### 2.2. Posterior and Predictive Distributions

Let  $Y_{obs}$  represent  $Y_{obs}^f$  or  $Y_{obs}^t$ . The joint density of  $Y_{obs}$  is therefore

$$f(Y_{obs} | \alpha, \beta) = \alpha^n \beta^n e^{-\beta \sum_{i=1}^n t_i} e^{-\alpha(1-e^{-\beta T})} \quad (2)$$

Case 1: When the shape parameter  $\beta$  is known, we adopt the following non-informative prior distribution of  $\alpha$ :

$$\pi(\alpha) \propto \frac{1}{\alpha}, \quad \alpha > 0. \quad (3)$$

The posterior distribution of  $\alpha$  is thus given by

$$h(\alpha | Y_{obs}) = [\Gamma(n)]^{-1} \alpha^{n-1} e^{-\alpha(1-e^{-\beta T})} (1-e^{-\beta T})^n \quad (4)$$

Let  $y^+$  be the random variable being predicted. Then the posterior predictive distribution of  $y^+$  is give as

$$f(y^+ | Y_{obs}) = \int_0^\infty f(y^+ | Y_{obs}, \alpha) h(\alpha | Y_{obs}) d\alpha \quad (5)$$

Hence the Bayesian UPL of  $y^+$  with level  $\gamma$  denoted as  $y_U^{(\beta)}$  must satisfy

$$\gamma = \int_{-\infty}^{y_U^{(\beta)}} f(y^+ | Y_{obs}) dy^+ . \tag{6}$$

Case 2: When the shape parameter  $\beta$  is unknown, we consider the following non-informative joint prior density for  $\alpha$  and  $\beta$  (we assume that  $\alpha$  and  $\beta$  are independent).

$$\pi(\alpha, \beta) \propto \frac{1}{\alpha\beta}, \quad \alpha, \beta > 0. \tag{7}$$

Hence the corresponding joint posterior density is given as

$$h(\alpha, \beta | Y_{obs}) = [k\Gamma(n)]^{-1} \alpha^{n-1} \beta^{n-1} e^{-\beta \sum_{i=1}^n t_i} e^{-\alpha(1-e^{-\beta T})} \tag{8}$$

where

$$k = \int_0^\infty \frac{\beta^{n-1} e^{-\beta \sum_{i=1}^n t_i}}{(1-e^{-\beta T})^n} d\beta . \tag{9}$$

Similar to Equation (5) and Equation (6), let  $yU$  denote the Bayesian UPL of  $y$  with level  $\gamma$ , then

$$f(y^+ | Y_{obs}) = \int_0^\infty \int_0^\infty f(y^+ | Y_{obs}, \alpha, \beta) h(\alpha, \beta | Y_{obs}) d\alpha d\beta \tag{10}$$

and

$$\gamma = \int_{-\infty}^{yU} f(y^+ | Y_{obs}) dy^+ \tag{11}$$

### 3. Main Results for the Prediction Problems

In this section, we address the four single-sample prediction issues raised in Section 2.1 using Bayesian approach. The following propositions are considered as the main results with proofs being given in the Appendix. In the subsequent results, we use  $\chi^2(n; \gamma)$  to represent the  $\gamma$  percentage point of the chi-square distribution with  $n$  degrees of freedom and we also assume the priors to be Equation (3) and Equation (7).

**Proposition 1 (for issue A):** The probability that at most  $k$  software failures will occur in the future time period  $(T, \tau]$  with  $\tau > T$  is

$$\gamma_k = \begin{cases} \left( \frac{1-e^{-\beta T}}{e^{-\beta T}-e^{-\beta \tau}} \right)^n \sum_{j=n}^{n+k} \binom{j-1}{n-1} \left( \frac{e^{-\beta T}-e^{-\beta \tau}}{1-e^{-\beta \tau}} \right)^j, & \text{when } \beta \text{ is known} \\ \sum_{j=n}^{n+k} \frac{\Gamma(n)}{d(j-n)! \Gamma(n)} \int_0^\infty \beta^{n-1} e^{-\beta \sum_{i=1}^n t_i} \frac{(e^{-\beta T}-e^{-\beta \tau})^{j-n}}{(1-e^{-\beta \tau})^j} d\beta, & \text{when } \beta \text{ is unknown} \end{cases} \tag{12}$$

**Proposition 2 (for issue B):** Suppose that the pre-determined target value  $\lambda_{tv}$  for the failure rate of the software undergoing development testing is not achieved at time  $T$ , the probability that the target value  $\lambda_{tv}$  will be achieved at time  $\tau, \tau > T$  is

$$\gamma = \begin{cases} 1 - \sum_{h=0}^{n-1} \frac{\left( \frac{1-e^{-\beta T}}{\beta e^{-\beta \tau}} \lambda_{tv} \right)^h}{h!} e^{-\frac{1-e^{-\beta T}}{\beta e^{-\beta \tau}} \lambda_{tv}} & \text{if } \beta \text{ is known} \\ 1 - \frac{1}{k} \sum_{h=0}^{n-1} \int_0^\infty \frac{\left( \frac{1-e^{-\beta T}}{\beta e^{-\beta \tau}} \lambda_{tv} \right)^h}{h!} e^{-\frac{(1-e^{-\beta T})}{\beta e^{-\beta \tau}} \lambda_{tv}} \beta^{n-1} e^{-\beta \sum_{i=1}^n t_i} \frac{1}{(1-e^{-\beta T})^n} d\beta & \text{if } \beta \text{ is unknown} \end{cases} \tag{13}$$

Remark 1: Let  $\{\beta^{(t)}\}_{t=1}^L$  be i.i.d. sample from  $\text{Gamma}\left(n, \sum_{i=1}^n t_i\right)$ , we can approximate the second part of (13) via MCMC method.

**Proposition 3 (for issue C):** For given level  $\gamma$ , the time  $\tau^*$  required to attain  $\lambda_{tv}$  is

$$(i) \quad \tau^* = \frac{1}{\beta} \ln \left[ \frac{\beta \chi^2(2n; \gamma)}{2\lambda_{tv}(1 - e^{-\beta T})} \right] - T \quad \text{if } \beta \text{ is known} \tag{14}$$

$$(ii) \quad \tau^* = \tau - T \quad \text{if } \beta \text{ is unknown} \tag{15}$$

where  $\tau$  is the solution to the following equation:

$$\gamma = 1 - \frac{1}{k} \sum_{h=0}^{n-1} \int_0^\infty \frac{\left( \frac{1 - e^{-\beta T}}{\beta e^{-\beta \tau}} \lambda_{tv} \right)^h}{h!} e^{-\frac{(1 - e^{-\beta T})}{\beta e^{-\beta \tau}} \lambda_{tv}} \frac{\beta^{n-1} e^{-\beta \sum_{i=1}^n t_i}}{(1 - e^{-\beta T})^n} d\beta \tag{16}$$

**Proposition 4 (for issue D):** The Bayesian UPL of  $\lambda_\tau = \alpha \beta e^{\beta \tau}$  with level  $\gamma$  is

$$(i) \quad \lambda_{tv}^{(\beta)} = \frac{\chi^2(2n; \gamma) \beta e^{-\beta \tau}}{2(1 - e^{-\beta T})} \quad \text{if } \beta \text{ is known} \tag{17}$$

$$(ii) \quad (\beta \text{ unknown}) \quad \lambda_{tv} \text{ such that} \tag{18}$$

$$\gamma = 1 - \frac{1}{k} \sum_{h=0}^{n-1} \int_0^\infty \frac{\left( \frac{1 - e^{-\beta T}}{\beta e^{-\beta \tau}} \lambda_{tv} \right)^h}{h!} e^{-\frac{(1 - e^{-\beta T})}{\beta e^{-\beta \tau}} \lambda_{tv}} \frac{\beta^{n-1} e^{-\beta \sum_{i=1}^n t_i}}{(1 - e^{-\beta T})^n} d\beta \tag{19}$$

### 4. Example

In this section, a real example from the time between failure data given by [9] is used to illustrate the developed methodologies for the single-sample Bayesian predictive analysis. The **Table 1** gives the Time Between Failure.

The study has used the cumulative time between failures as failure times  $0 < t_1 < t_2 < \dots < t_{30}$  where  $n = 30$ . These data obey the Goel-Okumoto (1979) software reliability model [10]. The MLEs of the parameters of the software reliability model based on the data are  $\hat{\alpha} = 31.698171$  and  $\hat{\beta} = 0.003962$ . In the illustration of the developed methodologies, the study has used these MLEs.

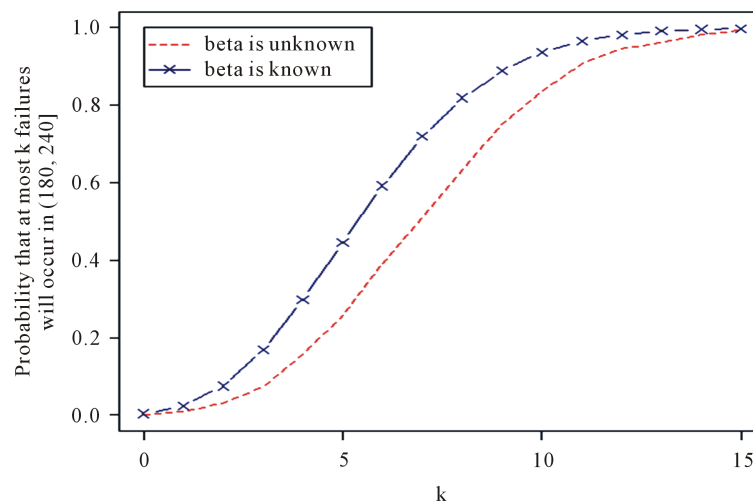
1) Suppose that we are interested in the probability  $\gamma_k$  that at most  $k$  failures will occur in the future time period  $(\tau, T] = (180, 250]$ . a) When  $\beta$  is known (say  $\beta = 0.003969$ ), using the first formula in Equation (12), we have  $\gamma_0 = 0.0039$ ,  $\gamma_1 = 0.0235$ ,  $\gamma_2 = 0.0750$ ,  $\gamma_3 = 0.1677$ ,  $\gamma_4 = 0.2970$ ,  $\gamma_5 = 0.4456$ ,  $\gamma_6 = 0.5920$ ,  $\gamma_7 = 0.7193$ ,  $\gamma_8 = 0.8188$ ,  $\gamma_9 = 0.8898$ ,  $\gamma_{10} = 0.9366$ ,  $\gamma_{11} = 0.9653$ ,  $\gamma_{12} = 0.9819$ ,  $\gamma_{13} = 0.9910$ ,  $\gamma_{14} = 0.9957$ , and  $\gamma_{15} = 0.9980$ . b) When  $\beta$  is unknown, from the second formula in Equation (12) we obtain,  $\gamma_0 = 0.0010$ ,  $\gamma_1 = 0.0087$ ,  $\gamma_2 = 0.0292$ ,  $\gamma_3 = 0.0721$ ,  $\gamma_4 = 0.1575$ ,  $\gamma_5 = 0.2566$ ,  $\gamma_6 = 0.3920$ ,  $\gamma_7 = 0.5099$ ,  $\gamma_8 = 0.6344$ ,  $\gamma_9 = 0.7534$ ,  $\gamma_{10} = 0.8350$ ,  $\gamma_{11} = 0.9062$ ,  $\gamma_{12} = 0.9462$ ,  $\gamma_{13} = 0.9621$ ,  $\gamma_{14} = 0.9831$ ,  $\gamma_{15} = 0.9944$ .

**Figure 1** shows the graph of the desired probabilities for the case when  $\beta$  is known and when  $\beta$  is unknown.

2) Suppose that the target value is given by  $\lambda_{tv} = 0.03$ . At time  $T = 182.21$ , the MLE of the achieved failure rate for this software is  $\hat{\lambda}(182.21) = \hat{\alpha} \hat{\beta} e^{-182.21 \hat{\beta}} = 0.061$  which is greater than  $\lambda_{tv}$  i.e. it cannot be achieved at time  $T = 182.21$ . Thus the development testing will continue. Suppose we want to predict the probability that the target value  $\lambda_{tv}$  will be achieved at time  $\tau = 277.83$ . a) When  $\beta$  is known, say  $\beta = 0.003962$ , from the first formula in Equation (13) we obtain  $\gamma = 0$ . Thus we can conclude that the target value (failure rate) will not be achieved. b) When  $\beta$  is unknown, from the second formula in Equation (13) and Remark 1, we obtain  $\gamma = 0.0576$  where the Monte Carlo sample size is  $L = 1000$ .

**Table 1.** Time between failures data.

Failure No.	Time between failures	Cumulative time between failures	Failure No.	Time between failures	Cumulative time between failures
1	30.02	30.02	16	15.53	151.78
2	1.44	31.46	17	25.72	177.50
3	22.47	53.93	18	2.79	180.29
4	1.36	55.29	19	1.92	182.21
5	3.43	58.72	20	4.13	186.34
6	13.2	71.92	21	70.47	256.81
7	5.15	77.07	22	17.07	273.88
8	3.83	80.90	23	3.99	277.83
9	21	101.90	24	176.06	453.93
10	12.97	114.87	25	81.07	535.00
11	0.47	115.34	26	2.27	537.27
12	6.23	121.57	27	15.63	552.90
13	3.39	124.96	28	120.78	673.68
14	9.11	134.07	29	30.81	704.49
15	2.18	136.25	30	34.19	738.68



**Figure 1.** Comparison of the probabilities  $\gamma_k$  that at most  $k$  failures will occur in the time interval  $(180, 240]$  for the cases of known and unknown  $\beta$ .

3) Since the target value  $\lambda_n$  was not achieved at time  $T = 182.21$ , we want to know how long it will require in order to attain  $\lambda_n$ . a) When  $\beta$  is known (i.e.  $\beta = 0.003962$ ), let  $\gamma = 0.90$ , from Equation (14) we obtain  $\tau^* = 268.6116h$ . In other words, it will take another 268.6116h in order to achieve the desired failure rate. b) When  $\beta$  is unknown, from Equation (15) and Equation (16), we obtain  $\tau^* = 770.79$  h. In other words, it will take another 770.79 h in order to achieve the desired failure rate when  $\beta$  is unknown.

4) Given  $\tau = 900$  h, a) when  $\beta$  is known, from Equation (17), the Bayesian Upper Prediction Limit of  $\lambda_\tau = \alpha\beta e^{-\beta\tau}$  with level 0.90 is given by  $\lambda_u^{(\beta)}(\tau) = 0.0051$ . b) When  $\beta$  is unknown, from Equation (18) and Equation (19), the Bayesian UPL of  $\lambda_\tau = \alpha\beta e^{-\beta\tau}$  with level 0.90 is given by  $\lambda_u^{(\beta)}(\tau) = 0.131952$ .

## 5. Discussion

Several prediction problems arise during the development of any software especially when the Goel-Okumoto (1979) software reliability model is used to model the failure process. We have used Bayesian approach with non-informative priors to address some of the prediction problems that may arise during software development testing stage. We have obtained explicit solutions to these problems, which may prove useful for the modification, debugging and for the decision to terminate the development testing process of the software.

The adoption of Bayesian approach for the derivation of the solutions is advantageous in that the approach is available for cases of small sample sizes [11] [12]. Another advantage of the Bayesian approach is that it allows the input of prior information about the reliability growth process and provides full posterior and predictive distributions.

In this paper, we have used non-informative priors to derive the methodologies to address the said prediction problems. However, informative priors can similarly be used in place of non-informative priors. The same procedures presented in this paper can also be applied to other NHPPs such as the delayed S-shaped process and the Cox-Lewis process.

## References

- [1] Nuria, T.R. (2011) Stochastic Comparisons and Bayesian Inference in Software Reliability. Ph.D. Thesis, Universidad Carlos III de Madrid, Madrid.
- [2] Daniel, R.J. and Hoang, P. (2001) On the Maximum Likelihood Estimates for the Goel-Okumoto Software Reliability Model. *The American Statistician*, **55**, 219-222. <http://dx.doi.org/10.1198/000313001317098211>
- [3] Meth, M. (1992) Reliability Growth Myths and Methodologies: A Critical View. *Proceedings of the Annual Reliability and Maintainability Symposium*, New York, 21-23 January 1992, 230-238.
- [4] Yu, J.-W., Tian, G.-L. and Tang, M.-L. (2007) Predictive Analyses for Nonhomogeneous Poisson Processes with Power Law Using Bayesian Approach. *Computational Statistics & Data Analysis*, **51**, 4254-4268. <http://dx.doi.org/10.1016/j.csda.2006.05.010>
- [5] Kapur, P.K., Pham, H., Gupta, A. and Jha, P.C. (2011) Software Reliability Assessment with OR Applications. Springer Series in Reliability Engineering, Springer-Verlag London Limited, London. <http://dx.doi.org/10.1007/978-0-85729-204-9>
- [6] Stringfellow, C. and Amschler, A.A. (2002) An Empirical Method for Selecting Software Reliability Growth Models. *Empirical Software Engineering*, **7**, 319-343. <http://dx.doi.org/10.1023/A:1020515105175>
- [7] Meyfroyt, P.H.A. (2012) Parameter Estimation for Software Reliability Models. Ph.D. Thesis, Universidad Carlos III de Madrid, Madrid.
- [8] Razeef, M. and Mohsin, N. (2012) Software Reliability Growth Models: Overview and Applications. *Journal of Emerging Trends in Computing and Information Sciences*, **3**, 1309-1320.
- [9] Xie, M., Goh, T.N. and Ranjan, P. (2002) Some Effective Control Chart Procedures for Reliability Monitoring. *Reliability Engineering and System Safety*, **77**, 143-150. [http://dx.doi.org/10.1016/S0951-8320\(02\)00041-8](http://dx.doi.org/10.1016/S0951-8320(02)00041-8)
- [10] Satya, P., Bandla, S.R. and Kantham, R.R.L. (2011) Assessing Software Reliability Using Inter Failures Time Data. *International Journal of Computer Applications*, **18**, 975-978.
- [11] Phillips, M.J. (2000) Bootstrap Confidence Regions for the Expected ROCOF of a Repairable System. *IEEE Transactions on Reliability*, **49**, 204-208. <http://dx.doi.org/10.1109/24.877339>
- [12] Quigley, J. and Walls, L. (2003) Confidence Intervals for Reliability-Growth Models with Small Sample-Sizes. *IEEE Transactions on Reliability*, **52**, 257-262. <http://dx.doi.org/10.1109/TR.2003.811865>



## Appendix (Proofs of Propositions 1 - 4)

In order to prove the propositions, we first give an identity without proof. The identity is

$$\int_{D(m;a,b)} dF(t_1) \cdots dF(t_m) = [F(b) - F(a)]^m / m! \quad (\text{A.1})$$

where  $m$  is any positive integer,  $a$  and  $b$  are two real numbers  $a < b$ ,  $F(t)$  is an increasing and differentiable function, and  $D(m;a,b) = [(t_1, \dots, t_m) : a < t_1 < \dots < t_m < b]$ .

**Proof of Proposition 1:** The probability that at most  $k$  failures will occur in the interval  $(T, \tau)$  is  $\gamma_k = \Pr[N(\tau) \leq n+k | Y_{obs}]$ . When  $\beta$  is known, we have

$$\gamma_k = \int_0^\infty \Pr[N(\tau) \leq n+k | Y_{obs}, \alpha] h(\alpha | Y_{obs}) d\alpha \quad (\text{A.2})$$

where  $h(\alpha | Y_{obs})$  is given by Equation (4) and

$$\Pr[N(\tau) \leq n+k | Y_{obs}, \alpha] = \sum_{j=n}^{n+k} f(Y_{obs}, N(\tau) = j | \alpha) / f(Y_{obs} | \alpha) \quad (\text{A.3})$$

From Equation (2) we have  $f(Y_{obs} | \alpha) = \alpha^n \beta^n e^{-\beta \sum_{i=1}^n t_i} e^{-\alpha(1-e^{-\beta T})}$  and

$$\begin{aligned} f(Y_{obs}, N(\tau) = j | \alpha) &= \int_{D(j-n:T,\tau)} f(Y_{obs}, x_{n+1}, \dots, x_j, N(\tau) = j) \prod_{\ell=n+1}^j dt_\ell \\ f(Y_{obs}, N(\tau) = j | \alpha) &= \int_{D(j-n:T,\tau)} \alpha^j \beta^j e^{-\beta \sum_{i=1}^j t_i} e^{-\alpha(1-e^{-\beta \tau})} \prod_{\ell=n+1}^j dt_\ell \\ &= \alpha^j \beta^j e^{-\beta \sum_{i=1}^n t_i} e^{-\alpha(1-e^{-\beta \tau})} \int_{D(j-n:T,\tau)} e^{-\beta \sum_{\ell=n+1}^j t_\ell} \prod_{\ell=n+1}^j dt_\ell \\ &\stackrel{\text{A.1}}{=} \alpha^j \beta^j e^{-\beta \sum_{i=1}^n t_i} e^{-\alpha(1-e^{-\beta \tau})} \frac{1}{\beta^{j-n}} \frac{(e^{-\beta T} - e^{-\beta \tau})^{j-n}}{(j-n)!} \end{aligned} \quad (\text{A.4})$$

Hence (A.3) becomes

$$\Pr[N(\tau) \leq n+k | Y_{obs}, \alpha] = \sum_{j=n}^{n+k} \alpha^{j-n} e^{-\alpha(e^{-\beta T} - e^{-\beta \tau})} \frac{(e^{-\beta T} - e^{-\beta \tau})^{j-n}}{(j-n)!} \quad (\text{A.5})$$

And (A.2) becomes

$$\begin{aligned} \gamma_k &= \int_0^\infty \sum_{j=n}^{n+k} \alpha^{j-n} e^{-\alpha(e^{-\beta T} - e^{-\beta \tau})} \frac{(e^{-\beta T} - e^{-\beta \tau})^{j-n}}{(j-n)! \Gamma(n)} \alpha^{n-1} e^{-\alpha(1-e^{-\beta T})} (1-e^{-\beta T})^n d\alpha \\ &= \sum_{j=n}^{n+k} \frac{(e^{-\beta T} - e^{-\beta \tau})^{j-n} (1-e^{-\beta T})^n}{(j-n)! \Gamma(n)} \int_0^\infty \alpha^{j-1} e^{-\alpha(1-e^{-\beta \tau})} d\alpha \\ &= \left( \frac{1-e^{-\beta T}}{e^{-\beta T} - e^{-\beta \tau}} \right)^n \sum_{j=n}^{n+k} \binom{j-1}{n-1} \left( \frac{e^{-\beta T} - e^{-\beta \tau}}{1-e^{-\beta \tau}} \right)^j \end{aligned} \quad (\text{A.6})$$

Equation (A.6) implies the first formula of Equation (12).

When  $\beta$  is unknown, noting that  $\Pr[N(\tau) \leq n+k | Y_{obs}, \alpha, \beta]$  and  $h(\alpha, \beta | Y_{obs})$  are given by Equation (A.3) and Equation (8) respectively, we have



$$\begin{aligned}
 \gamma_k &= \int_0^\infty \int_0^\infty \Pr[N(\tau) \leq n+k | Y_{obs}, \alpha, \beta] h(\alpha, \beta | Y_{obs}) d\alpha d\beta \\
 &= \sum_{j=n}^{n+k} \frac{1}{q(j-n)! \Gamma(n)} \int_0^\infty \int_0^\infty \alpha^{j-n} e^{-\alpha(e^{-\beta T} - e^{-\beta \tau})} (e^{-\beta T} - e^{-\beta \tau})^{j-n} \alpha^{n-1} \beta^{n-1} e^{-\beta \sum_{i=1}^n t_i} e^{-\alpha(1-e^{-\beta T})} d\alpha d\beta \quad (A.7) \\
 &= \sum_{j=n}^{n+k} \frac{\Gamma(n)}{d(j-n)! \Gamma(n)} \int_0^\infty \beta^{n-1} e^{-\beta \sum_{i=1}^n t_i} \frac{(e^{-\beta T} - e^{-\beta \tau})^{j-n}}{(1-e^{-\beta T})^j} d\beta
 \end{aligned}$$

Equation (A.7) implies the second formula of Equation (12).

**Proof of Proposition 2:** Let  $f(\lambda_\tau | Y_{obs})$  denote the posterior density of  $\lambda_\tau = \alpha\beta e^{-\beta\tau}$ . Hence the probability that the target value  $\lambda_{tv}$  will be achieved at time  $\tau$  ( $\tau > T$ ) is given by

$$\gamma = \Pr(\lambda_\tau \leq \lambda_{tv} | Y_{obs}) = \int_0^{\lambda_{tv}} f(\lambda_\tau | Y_{obs}) d\lambda_\tau. \quad (A.8)$$

When  $\beta$  is known, making the transformation  $\lambda_\tau = \alpha\beta e^{-\beta\tau}$ , we have  $\alpha = \frac{\lambda_\tau}{\beta e^{-\beta\tau}}$  and  $\frac{d\alpha}{d\lambda_\tau} = \frac{1}{\beta e^{-\beta\tau}}$ .

Consequently, the posterior density of  $\lambda_\tau$  is  $f(\lambda_\tau | Y_{obs}) = h(\alpha | Y_{obs}) \left| \frac{d\alpha}{d\lambda_\tau} \right|$ . This implies that

$$f(\lambda_\tau | Y_{obs}) = \left( \frac{\lambda_\tau}{\beta e^{-\beta\tau}} \right)^{n-1} \frac{1}{\Gamma(n)} e^{-\frac{\lambda_\tau}{\beta e^{-\beta\tau}}(1-e^{-\beta T})} (1-e^{-\beta T})^n \cdot \frac{1}{\beta e^{-\beta\tau}} \text{ which after simplification reduces to}$$

$$f(\lambda_\tau | Y_{obs}) = \frac{\left( \frac{1-e^{-\beta T}}{\beta e^{-\beta\tau}} \right)^n}{\Gamma(n)} \lambda_\tau^{n-1} e^{-\lambda_\tau \left( \frac{1-e^{-\beta T}}{\beta e^{-\beta\tau}} \right)} \quad (A.9)$$

We note that  $\lambda_\tau$  from Equation (A.9) follows a gamma distribution with parameters  $n$  and  $\frac{1-e^{-\beta T}}{\beta e^{-\beta\tau}}$ . Noting the relationship between gamma and Poisson distributions as

$$\frac{\beta^\alpha}{\Gamma(\alpha)} \int_0^\lambda x^{\alpha-1} e^{-\beta x} dx = 1 - \sum_{h=0}^{\alpha-1} \frac{(\beta\lambda)^h}{h!} e^{-\beta\lambda} \quad (A.10)$$

and from Equations (A.8), (A.9) and (A.10), we obtain the first formula of Equation (13).

When  $\beta$  is unknown, making the transformation  $\lambda_\tau = \alpha\beta e^{-\beta\tau}$  and  $\beta = \beta$ , we obtain  $\alpha = \frac{\lambda_\tau}{\beta e^{-\beta\tau}}$  and  $\beta = \beta$ . Note that the Jacobian is  $\frac{d(\alpha, \beta)}{d(\lambda_\tau, \beta)} = \frac{1}{\beta e^{-\beta\tau}}$ . From Equation (8), the joint posterior density of  $(\lambda_\tau, \beta)$  is given as

$$f(\lambda_\tau, \beta | Y_{obs}) = h(\alpha, \beta | Y_{obs}) \left| \frac{d(\alpha, \beta)}{d(\lambda_\tau, \beta)} \right| = \frac{\beta^{n-1} e^{-\beta \sum_{i=1}^n t_i} \left( \frac{1-e^{-\beta T}}{\beta e^{-\beta\tau}} \right)^n}{k(1-e^{-\beta T})^n \Gamma(n)} \lambda_\tau^{n-1} e^{-\lambda_\tau \left( \frac{1-e^{-\beta T}}{\beta e^{-\beta\tau}} \right)} \quad (A.11)$$

From Equation (A.8), Equation (A.10) and Equation (A.11) we obtain

$$\begin{aligned}
\gamma &= \frac{1}{k} \int_0^\infty \frac{\beta^{n-1} e^{-\beta \sum_{i=1}^n t_i}}{(1-e^{-\beta T})^n} \left\{ 1 - \sum_{h=0}^{n-1} \frac{\left( \frac{1-e^{-\beta T}}{\beta e^{-\beta \tau}} \lambda_{tv} \right)^h}{h!} e^{-\frac{(1-e^{-\beta T})}{\beta e^{-\beta \tau}} \lambda_{tv}} \right\} d\beta \\
&= \frac{1}{k} \int_0^\infty \frac{\beta^{n-1} e^{-\beta \sum_{i=1}^n t_i}}{(1-e^{-\beta T})^n} d\beta - \frac{1}{k} \sum_{h=0}^{n-1} \int_0^\infty \frac{\left( \frac{1-e^{-\beta T}}{\beta e^{-\beta \tau}} \lambda_{tv} \right)^h}{h!} e^{-\frac{(1-e^{-\beta T})}{\beta e^{-\beta \tau}} \lambda_{tv}} \frac{\beta^{n-1} e^{-\beta \sum_{i=1}^n t_i}}{(1-e^{-\beta T})^n} d\beta \\
&= 1 - \frac{1}{k} \sum_{h=0}^{n-1} \int_0^\infty \frac{\left( \frac{1-e^{-\beta T}}{\beta e^{-\beta \tau}} \lambda_{tv} \right)^h}{h!} e^{-\frac{(1-e^{-\beta T})}{\beta e^{-\beta \tau}} \lambda_{tv}} \frac{\beta^{n-1} e^{-\beta \sum_{i=1}^n t_i}}{(1-e^{-\beta T})^n} d\beta
\end{aligned} \tag{A.12}$$

Equation (A.12) implies the second formula of Equation (13).

**Proof of Proposition 3:** For given level  $\gamma$ , the time required to attain the target value  $\lambda_{tv}$  is  $\tau^* = \tau - T$  where  $\tau$  satisfies Equation (A.8). When  $\beta$  is known, from Equation (A.9), it can easily be seen that

$2 \left( \frac{1-e^{-\beta T}}{\beta e^{-\beta \tau}} \right) \lambda_{tv}$  follows a chi-square distribution with  $2n$  degrees of freedom. Therefore, we have

$$2 \left( \frac{1-e^{-\beta T}}{\beta e^{-\beta \tau}} \right) \lambda_{tv} = \chi^2(2n; \gamma) \tag{A.13}$$

and Equation (14) follows immediately. We can obtain (ii) by following similar arguments given in the proof for the second part of Proposition 2.

**Proof of Proposition 4:** For a pre-determined  $\tau$  ( $\tau > T$ ), the Bayesian Upper Prediction Limit (UPL) for  $\lambda_{\tau}$  with level  $\gamma$  is  $\lambda_U^{(\beta)}(\tau)$  satisfying  $\gamma = \Pr(\lambda_{\tau} \leq \lambda_U^{(\beta)}(\tau) | Y_{obs})$ . From Equation (A.8) and Equation (A.13)

we have  $\gamma = \int_0^{\lambda_U^{(\beta)}(\tau)} \frac{\left( \frac{1-e^{-\beta T}}{\beta e^{-\beta \tau}} \right)^n}{\Gamma(n)} \lambda_{\tau}^{n-1} e^{-\lambda_{\tau} \frac{(1-e^{-\beta T})}{\beta e^{-\beta \tau}}} d\lambda_{\tau}$ . This implies that

$$2 \left( \frac{1-e^{-\beta T}}{\beta e^{-\beta \tau}} \right) \lambda_U^{(\beta)}(\tau) = \chi^2(2n; \gamma) \tag{A.14}$$

Making  $\lambda_U^{(\beta)}(\tau)$  the subject from Equation (A.14) we arrive at

$$\lambda_U^{(\beta)}(\tau) = \frac{\chi^2(2n; \gamma) \beta e^{-\beta \tau}}{2(1-e^{-\beta T})} \tag{A.15}$$

Equation (A.15) is the exact formula in Equation (17).

The formula in Equation (18) can be obtained by similar arguments.

Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either [submit@scirp.org](mailto:submit@scirp.org) or [Online Submission Portal](#).

