BAYESIAN PREDICTIVE ANALYSES FOR AN EXPONENTIAL NON-HOMOGENEOUS POISSON PROCESS IN SOFTWARE RELIABILITY

ALBERT ORWA AKUNO

A Research Thesis Submitted to the Graduate School in Partial Fulfillment of the Requirements for the Award of Master of Science Degree in Statistics of Egerton University

EGERTON UNIVERSITY

MARCH, 2015

DECLARATION AND RECOMMENDATION

DECLARATION

This thesis is my original work and has not been submitted or presented in part or whole for examination in any institution.

Signature: ………………………….. **Date: ……………………….**

Mr. Akuno A. O.
SM12/3283/12

RECOMMENDATIONS

This thesis has been submitted with our approval as supervisors for examination according to Egerton University regulations.

Dr. Orawo L. A.
Department of Mathematics
Egerton University

Signature: ………………………….. **Date: ……………………….**

Prof. Islam A. S.
Department of Mathematics
Egerton University

Signature: ………………………….. **Date: ……………………….**

DEDICATION

This thesis is dedicated to my mother Miriam Achieng Akuno, my late father Jared Akuno Orwa and my siblings: Calvince Opiyo Akuno, Elver Awuor Akuno and Debrah Atieno Akuno, all to whom I attribute my success in life. They have been a pillar in my life and a constant source of encouragement and tireless support in my academic and social life.

ACKNOWLEDGEMENT

First, I would like to thank the Almighty God for having led me this far and having given me a good life, sound health and knowledge throughout this research. If it were not for God, I would have not gotten the opportunity to put this research work together and bring it to completion. Glory is to God for his grace is sufficient for all of us, Amen. I also want to express my sincere gratitude to the Egerton University council for having awarded me full masters scholarship to pursue this course. I will forever be grateful to my two supervisors, Dr. Luke Akong'o Orawo and Prof. Ali Salim Islam for their selfless support, positive criticism; both typographic and technical and guidance throughout the period when I was preparing this dissertation. Dr. Luke Akong'o Orawo was so resourceful and his programming techniques came in handy during the derivation of the methodologies in this dissertation. Thank you so much 'Daktari'.

In a special way, I want to thank my Mother for her constant prayers, financial support and undying understanding throughout my academic and social life. I will not fail to register my gratitude to my late father for his dedication and determination to see me reach to this level by working so hard to put me through secondary school.

I also want to register my appreciation to my classmate Veronica Kariuki with whom we were working on the same area of study. Her support and scholarly comments were so vital in the preparation of this document. I am equally grateful to my other classmates; Timothy Mutunga, Charles Wambugu, Jennifer Nthiwa, Walter Japar and Lawrence Areba for their support and positive comments when we were doing our classwork and also in my social life.

To the many people who are not mentioned but contributed to the fruition of this work, your support, care and understanding is hereby fully registered and recognized.

ABSTRACT

Software reliability is defined as the probability of failure free operation of a software in a specified environment during a specified period. Software reliability has been the focus of several researches over the last four decades. One of the earliest software reliability models is the exponential nonhomogeneous Poisson process developed by Goel and Okumoto in 1979. Most research works have considered fitting different software reliability models to different software reliability data where the estimates of the parameters of the models are obtained by maximum likelihood method. However, the problem of predictive analysis on the Goel – Okumoto software reliability model has not so far been explored despite the fact that predictive analysis is very useful for modifying, debugging and determining when to terminate software development testing process. This would lead to improved software reliability and efficient use of resources during software development testing. To assess and improve software reliability, software developers have to perform operational profile testing where they emulate the end-user environment during software testing. Operation profile testing is difficult and time consuming especially when there are multiple types of end-users and hence there is the need for software predictive analysis. The main objective of this study was to perform Bayesian predictive analyses on the Goel – Okumoto software reliability model. Informative and non-informative priors for one-sample case and non-informative prior for two-sample case has been used in the study. Brief literature on NHPP is given in chapter four. The various issues that are associated closely to software development testing process have been outlined in Chapter three as issues A1, B1, C1 and D1 for one-sample case and issues A2, B2 and C2 for the two-sample case. These issues have been addressed by various propositions given in chapter four as follows: Issues A1, B1, C1 and D1 have been addressed by propositions A1, B1, C1 and D1 respectively. The same issues A1, B1, C1 and D1 have been addressed by propositions A1.1, B1.1, C1.1 and D1.1. which have been developed using informative prior. Issues A2, B2 and C2 have been addressed by propositions A2, B2 and C2 respectively. The propositions for the single sample case have been illustrated by secondary software failure data while the propositions for the two-sample case have been illustrated by simulated software failure data. The summary, conclusions and recommendations for further research is given in chapter five. The R-language programs that were used in the analysis are provide in the appendix.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

HPP	Homogeneous Poisson Process
MCMC	Monte Carlo Markov Chain
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimate
MTBF	Mean Time Between Failures
NHPP	Nonhomogeneous Poisson Process
PLP	Power Law Process
UPL	Upper Prediction Limit

LIST OF SYMBOLS

$\boldsymbol{M(t)}$	Mean value function, mean number of failures up to time t
$\boldsymbol{N(t)}$	Cumulative number of software failures observed by time t
n	Total number of software failures
$\boldsymbol{p(\theta\|y)}$	Bayesian posterior density
$\boldsymbol{p(y^{+}\|y)}$	Bayesian posterior predictive distribution of y^{+}
$\boldsymbol{t_i}$	Time up to the i^{th} failure
$\boldsymbol{\lambda(t)}$	Failure intensity function at time t
$\boldsymbol{\alpha}$	Scale parameter for the Goel – Okumoto software reliability model
$\boldsymbol{\beta}$	Shape parameter for the Goel – Okumoto software reliability model

CHAPTER ONE
INTRODUCTION

1.1 Background Information

Over the last decade of the 20th century and the first few years of the 21st century, the demand for complex software systems has gone up as it is seen that currently, computer systems have become an indispensable component of our modern society. Today, computers are embedded in automotive mechanical and safety control systems, industrial and quality control processes, real-time sensor networks, aircrafts, nuclear reactors, hospital healthcare and air traffic control systems among others. 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 (Nuria, 2011).

A single software defect can cause system failure and to avoid these failures, reliable software is required (Satya *et al.* 2011). Software reliability is achieved through testing during the software development stage (Daniel and Hoang, 2001). Running test cases in a manner that exercises the software similar to the way users will operate the software in their particular environment is the usual criteria for removing bugs in the software that may cause its failure (Daniel and Hoang, 2001). However, emulating end-user environment during the test interval is difficult and time consuming especially when there are multiple types of end-users. Besides, 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 (Daniel and Hoang, 2001). Software reliability modeling comes in handy to address this dilemma.

Software reliability modeling can provide the basis for planning reliability growth tests, monitoring progress, estimating current reliability, forecasting and predicting future reliability improvements (Meth, 1992). This implies that a software reliability growth model is a powerful tool for forecasting and predicting the next failure time of software given initial failures and the software user-environment. Forecasting and prediction is achieved through predictive analyses. In particular, predictive analyses are useful in determining when to terminate the development process of software or hardware (Jun-Wu *et al.* 2007). Often, a prediction interval is constructed to provide the time frame when the $kth\ (k > 0)$ future failure observation will occur with a pre-determined confidence level.

Many software reliability growth models that can be used for predictive analyses have been developed by various authors in the past three decades. The Goel – Okumoto software reliability model is among the pioneer exponential non-homogeneous Poisson process software model having been proposed by Goel and Okumoto in 1979. The model is based on the assumptions that failures are observed during execution caused by remaining faults in the software; whenever a failure is observed, an instantaneous effort is made to find what caused the failure and the faults are removed prior to future tests; all faults in the software are mutually independent and that there is a perfect debugging process, i.e. there is no new fault that is introduced into the software during the debugging process (Kapur *et al.* 2011). These assumptions lead to a mean value function that depicts a decreasing failure rate. Meaning that the number of failures per given time is decreasing as test time grows large. When predictive analysis is done based on this model, the predicted future failures should portray this phenomenon of decreasing failure rate.

There are two main aspects of a good reliability model. First, the model must remain stable during the entire testing period for any particular testing environment. Secondly, a reasonably accurate prediction of reliability must be provided by the model (Kapur *et al*. 2011). The Goel – Okumoto (1979) model has been used in various testing environment and in many instances, it provides good estimation and prediction of software reliability.

Bayesian reliability modeling is anchored on the development of reliability posterior distribution from which predictive inference is made. The reliability posterior distribution is often constructed using prior distribution which encapsulates prior information about the parameters of the software reliability model. The advantage of using Bayesian statistics is that it allows prior information such as engineering judgments and test results to be combined with more recent information like test or field data. This is important since it helps software developers to arrive at a prediction of reliability based upon a combination of all available information (Allan, 2012). Furthermore, it is important to note that in software reliability modeling, early test results do not tell the whole results. A reliability assessment comes not only from testing the product itself but also from the information which is available prior to the start of the test. This information may include; the environment under which the software will work, previous tests on the software and even intuition based upon experience (Allan, 2012). The fundamental question therefore is ‘why should this useful prior information not be used to supplement and achieve more robust final reliability results?’

1.2 Statement of the problem

The use of computer systems and modern communication technologies has become an integrated part of human activities in the world. Computer systems are enormously utilized in several areas including but not limited to those with safety – critical functions and thus production of reliable software through statistical predictive analyses is of great interest. The Goel – Okumoto software reliability model, an exponential non-homogeneous Poisson process, has been used in various software testing environment. In many instances the model has provided good fit to software reliability data and hence, can be considered as a useful reliability model. Most research works have considered fitting the model to different software reliability data where the parameters of the model are obtained by maximum likelihood method. However, both frequentist and Bayesian predictive analyses on the model has not so far been explored. Predictive analysis is very useful for modifying, debugging and determining when to terminate software development testing process, leading to improved software reliability. Bayesian predictive procedure is advantageous over frequentist approach in that it allows the input of prior information about the reliability growth process. This research has therefore performed one-sample and two-sample Bayesian predictive analyses on the Goel – Okumoto software reliability model using informative and non-informative priors.

1.3 Objectives

1.3.1 General objective

To conduct Bayesian predictive analyses on the Goel – Okumoto software reliability model.

1.3.2 Specific objectives

1. To perform one-sample Bayesian predictive analysis on the model using non-informative and informative priors.
2. To perform two-sample Bayesian predictive analyses on the model using non-informative priors.

3. To generate synthetic software failure data from the Goel – Okumto software reliability model.

1.4 Assumptions

1. There must have been initial operational profile tests of the software.
2. If the software failure times are $0 < t_1 < t_2 < \cdots < t_n < T$ where T is time truncated, we assume that $t_n = T$

1.5 Justification

The reliability of any software is of great interest to the software developers before a decision is made to release the software into the market. Software developers need correct and concise information about how reliable software is before they decide to release the software into the market. It has been widely observed that software warranties, even of a primitive kind are hard to come by. It has also been observed that software producers often attempt to avoid any responsibility of software failure once it has been released into the market. Therefore, the producers will prefer any information that will lead to low cost but high quality software. Software reliability is achieved through testing during the software development stage. However, there exists a conspicuous trade-off between spending too much time testing software, which obviously delays the release, and too little time testing software and eventually exposing the users to poor quality software. In order to improve software reliability, most software developers will go for that procedure that minimizes the cost of software development and at the same time, guarantees the reliability of the software. This can be achieved through software predictive analysis. This study has therefore developed Bayesian predictive analyses procedures on the Goel – Okumoto software reliability model, an exponential non-homogeneous Poisson process. The procedures will enable software developers to achieve the desire of early release of high quality and reliable software into the market.

CHAPTER TWO

LITERATURE REVIEW

2.1 Counting processes

A counting process is simply the count of the number of events that occur in any time interval. An indexed collection of random variables is called a stochastic process and when the focus is on counts, the process is called a counting process and is denoted by $N(t),\ t \geq 0$. Models employing a counting process have played a major role in the analysis of systems composed of randomly occurring events. For instance, suppose the interest is in observing repeatable events occurring over a period of time. A simple example is the arrival of customers at a station e.g. a bank for service. Other examples include the occurrence of earthquakes of a certain magnitude at a particular location over time and the times of software failures. What is of importance in this study is the point in time when software failure is experienced.

A software system receives different types of input each with its own different path through the software thus the creation of a capability of bringing different errors into light (Jelinski and Moranda, 1972). The different input types are viewed as arriving randomly to the software leading to detection of errors in a random way. The end result is that there is an underlying random process that governs the software failures. This justifies the use of stochastic methods to model software failures (Singpurwala and Simon, 1994). There are some probabilistic models describing the counting process. These are homogeneous and non-homogeneous Poisson processes. The following definitions are given in terms of software failure as that is the focus of the study.

2.1.1 Poisson processes

A counting process $\{N(t),\ t > 0\}$ is said to be a Poisson process if

i. $N(0) = 0$

ii. For any time points $t_0 = 0 < t_1 < t_2 < ... < t_n$ the random

variables $N(t_0, t_1],\ N(t_1, t_2]$,..., $N(t_{n-1}, t_n]$ are independent random variables. This is called the independent increment property.

iii. There is a function λ such that $\lambda(t) = \lim_{\Delta t \to 0} \frac{\Pr[N(t+\Delta t) - N(t) \geq 1]}{\Delta t}$

iv. $\lambda(t) = \lim_{\Delta t \to 0} \frac{\Pr[N(t+\Delta t) - N(t) \geq 2]}{\Delta t} = 0$. This property precludes the possibility of simultaneous failures.

The above properties (i) to (iv) of the Poisson process imply that

$$\Pr[N(t) = n] = \frac{1}{n!}\left(\int_0^t \lambda(x)dx\right)^n \exp\left(-\int_0^t \lambda(x)dx\right) \tag{1}$$

2.1.2 Homogeneous Poisson Processes

A counting process $(N(t),\ t > 0)$ is said to be a homogeneous Poisson process (HPP) if the intensity function $\lambda(t)$ is constant (Zhao, 2004), i.e. $\lambda(t) = \lambda,\ \lambda > 0$ and

i. The failure at time zero, $N(0) = 0$.

ii. The process has independent increment and stationary increment. A point process has stationary increments if for all k, $\left[\Pr(n, n+s) = k\right]$ is independent of t

iii. The number of events occurring in any interval of length $t = t_2 - t_1$ has a Poisson distribution with mean λt , that is

$$\Pr[N(t_2) - N(t_1) = n] = \frac{e^{-\lambda t}(\lambda t)^n}{n!}, \qquad 0 \leq t_1 \leq t_2, n = 0,1,\ldots \tag{2}$$

A homogeneous Poisson process has the following properties (Rigdon and Basu, 2000)

i. A process is a Homogeneous Poisson process with constant intensity function λ if and only if the times between events are independent and identically distributed exponential random variables with mean $1/\lambda$

ii. If $0 < T_1 < T_2 < \ldots < T_n$ are the failure times from a HPP, then the joint probability distribution function of $T_1, T_2, \ldots T_n$ is

$$f(t_1, t_2, \ldots, t_n) = \lambda^n e^{-\lambda t_n}, 0 < t_1 < t_2 < \ldots < t_n \tag{3}$$

iii. The time to the nth failure from a system modeled by a HPP has a gamma distribution with parameter $\alpha = n,\ \beta = 1/\lambda$.

iv. For a HPP, conditional on $N(t) = n$, the failure times $0 < T_1 < T_2 < \ldots < T_n$ are distributed as order statistics from uniform distribution in the interval $(0, t)$.

v. The probability of a system failure after time t is $R(t) = \Pr[T > t] = \Pr[N(t) = 0] = e^{-\lambda t}$

2.1.3 Nonhomogeneous Poisson Processes

A Nonhomogeneous Poisson Process (NHPP) is a Poisson process whose intensity function is not a constant (Zhao, 2004). A counting process $(N(t) > 0,\ t > 0)$ has a nonhomogeneous Poisson process if

i. $N(0) = 0$

ii. The process has independent increment

iii. The number of failures in any interval $(t_1, t_2]$ has a Poisson distribution with mean $\int_{t_1}^{t_2} \lambda(t)dt$, that is

$$\Pr[(N(t_2) - N(t_1)) = k] = \frac{1}{k!}\left(\exp\left(-\int_{t_1}^{t_2} \lambda(t)dt \right)\left(-\int_{t_1}^{t_2} \lambda(t)dt \right)^k \right) \tag{4}$$

The following are the properties of non-homogeneous Poisson process (Rigdon and Basu, 2000).

i. The joint pdf of the failure times $T_1, T_2, \ldots T_n$ from a nonhomogeneous Poisson process with intensity function $\lambda(t)$ is given by

$$f(t_1, t_2, \ldots, t_n) = \left(\prod_{i=1}^{n} \lambda(t_i) \right) \exp\left(-\int_0^T \lambda(t)dt \right) \tag{5}$$

where T is the stopping time: $T = t_n$ for the failure truncated case, $T = n$ for the time truncated case. This is also known as the likelihood function.

ii. If $0 < t_1 < t_2 < \ldots$ are the epochs at which the failure times occur, then the between occurrence intervals $T_k = t_k - t_{k-1}$ $(k = 1, 2, \ldots)$ are independent random variables and with densities

$$f_{t_k}(t_k) = \lambda(t_k)\exp\left(-\int_{t_{k-1}}^{t_k} \lambda(t)dt\right) \tag{6}$$

Again, $(t_1, t_2, \ldots)$ is a Markov sequence with transition density given as

$$\Pr(t_k \mid t_{k-1}) = \lambda(t_k)\exp\left(-\int_{t_{k-1}}^{t_k} \lambda(t)dt\right) \tag{7}$$

iii. Conditional on $N(t) = n$, the n failure times $0 < T_1 < T_2 < \ldots < T_n$ have the same distribution as n order statistics corresponding to a random sample of n observation from the density

$$f(t) = \lambda(t) / \int_0^{T_0} \lambda(t)dt, \ \ 0 \le t \le T_0, \tag{8}$$

which reduces to the uniform distribution over $[0, T_0]$ when $\lambda(t) = \lambda$.

Since the 1970s, software reliability has been an important research topic. Models based on non-homogeneous Poisson processes (NHPP) have played an important role in such studies since NHPPs are key in describing the fault detection process of software (Zhao and Xie, 1996). The NHPP models are also used when modeling and analyzing the failure process of repairable systems. A good example of a NHPP is the Weibull process with intensity function $\lambda(t) = \left(\frac{\beta}{\alpha}\right)\left(\frac{t}{\alpha}\right)^{\beta-1}$. The Weibull process can in some cases be used to model software failures but it is mostly used to model and analyze the failure process of repairable systems. For instance, Jun-Wu *et al*. (2007) performed a predictive analysis for nonhomogeneous Poisson processes with power law using Bayesian approach. The analysis was done to predict failure times of radar system development and an electronic system development. In the literature, this NHPP possess different names such as power law process (Preeti and Nidhi, 2011; Jun-Wu *et al*., 2007; Muralidharan *et al*., 2008; Zhao, 2004) and the Duane (1964) model.

The power law process has been widely used in modeling failure times of repairable systems and detecting software failures. For instance, frequentist estimation of unknown parameters of this model has been studied (Crow, 1982; Bain, 1978; Bain and Engelhardt, 1980). Classical estimation of the future reliability based on a predictive distribution has also been done (Muralidharan *et al*., 2008). They also obtained point and interval estimates of the parameters as well as reliability analysis based on full likelihood and predictive likelihood. Empirical Bayes Analysis on the Power Law Process with Natural Conjugate Priors was done by Zhao (2010). Bayesian predictive analysis on the power law process using non – informative priors has been done by (Zhao, 2004; Guida *et al*., 1989; Shaul *et al*., 1992; Jun – Wu *et al*., 2007). The power law process is widely applied in monitoring reliability growth during the development test phase. However, the Power Law Process has some drawbacks in the reliability growth context. This is because the intensity function of the PLP model brings along with it two unrealistic situations; it tends to infinity as t tends to zero and tends to zero as t tends to infinity (Preeti and Nidhi, 2011). Modified PLP model has been proposed to overcome the latter drawback and Doubly Bounded PLP model to overcome both the drawbacks (Preeti and Nidhi, 2011). System reliability measures of intensity function and Mean Time Between Failures (MTBF) has also been used for reliability prediction of the PLP model.

Other non-homogeneous Poisson processes include Goel – Okumoto (1979) software reliability model with intensity function $\lambda(t) = \alpha\beta e^{-\beta t}$, Musa – Okumoto (1984) model with $\lambda(t) = \alpha / (t + \beta)$, the delayed S – shaped model, and Yamada *et al*., (1983) model with $\lambda(t) = \alpha\beta^2 te^{-\beta t}$.

2.2 Bayesian Methods

The Bayes rule in statistical inference was first introduced by the 18th century clergyman and mathematician Thomas Bayes. Andrew *et al*. (1995) outlines the Bayes procedure. Bayesian procedures are widely known and may be found in many Bayesian books and statistical journals that apply Bayes methods in their study.

2.2.1 Prior distributions

The main goal of a typical Bayesian statistical analysis is to obtain the posterior distribution of model parameters. The posterior distribution is best understood as a weighted average between the knowledge about the parameters before data is observed, which is represented by the prior distribution and the information contained in the data about the unknown parameters which is represented by the likelihood function (Glickman and Van, 2007).

Before a Bayesian analysis is conducted, the statistician needs to observe the data at hand and formulate or choose a probability model for the data. Once the data model is formulated, a Bayesian analysis requires the assertion of a prior distribution for the unknown parameters of the model. The prior distribution can be viewed as representing the current state of knowledge or current description of uncertainty, about the model parameters prior to data being observed (Glickman and Van, 2007). Prior distributions are divided into two categories namely, informative and non-informative priors.

For the case of informative priors, the statistician uses his intuitive knowledge about the substantive problem at hand, perhaps based on past data along with expert opinion to formulate a prior distribution that properly reflects his (and experts') beliefs about the unknown parameters of the model. This approach has always been criticized as it seems at first to be overly subjective and unscientific. However, it can be argued that if prior knowledge or information about the model parameters exists prior to observing data, then it would be unscientific not to include such knowledge or information into data analysis.

The second main approach to choosing a prior distribution is by using non-informative prior. This approach represents ignorance about the model parameters. This approach is also called objective, vague, diffuse and sometimes, reference prior distribution. Choosing a non-informative prior distribution is an attempt towards objectivity as it involves acting as though no prior knowledge about the parameters exists before data is observed. This is achieved through assigning equal probabilities to all values of the parameters. The beauty of this approach is that it directly addresses the criticism of informative prior distributions as being subjectively chosen.

2.2.2 Bayes rule

If θ is a parameter and y is a random variable, the probability statement about θ given y can be made when we first consider a model providing a joint probability distribution for θ and y (Andrew *et al*., 1995). The joint probability mass or density function are written as a product of two densities that are often referred to as the prior distribution $\Pr(\theta)$ and the sampling distribution $\Pr(y \mid \theta)$ respectively, that is

$$\Pr(\theta, y) = \Pr(\theta)\Pr(y \mid \theta)$$

Conditioning on the known value of y and using the basic conditioning property known as the Bayes' rule, we obtain the posterior density as

$$\Pr(\theta \mid y) = \frac{\Pr(\theta, y)}{\Pr(y)}$$

$$= \frac{\Pr(\theta)\Pr(y \mid \theta)}{\Pr(y)} \qquad (9)$$

where $\Pr(y) = \sum_{\theta}\Pr(\theta)\Pr(y \mid \theta)$, and the sum is over all possible values of θ and for the case of continuous θ, $\Pr(y) = \int \Pr(\theta)\Pr(y \mid \theta)d\theta$. An equivalent form of the posterior distribution above omits the factor $\Pr(y)$ that is independent of θ and with fixed y which is considered as a constant of proportionality yielding the unnormalized posterior density which is the right side of the equation $\Pr(\theta \mid y) \propto \Pr(\theta)\Pr(y \mid \theta)$.

This expression encloses the technical core of Bayesian inference. The primary task of any specific application is to develop the model $\Pr(\theta, y)$ and perform the necessary computation to summarize $\Pr(\theta \mid y)$ in appropriate ways (Andrew *et al*., 1995).

2.2.3 Bayesian Predictive inference

The posterior distribution, $p(\theta|y)$, is used as a means of making inference about the parameter θ. In order to make inference about an unknown independent future observation, Andrew *et al*.

(1995) indicate that before the data y are considered, the distribution of the unknown but observable y is

$$\Pr(y) = \int \Pr(y, \theta) d\theta$$

$$= \int \Pr(\theta) \Pr(y \mid \theta) d\theta$$

This is often called the prior predictive distribution. The predictive distribution $\Pr(y)$, is called prior because it is not conditional on a previous observation of the process and predictive because it is the distribution of a quantity that is observable. After the data y has been observed, the future but unknown observation y^+ from the same process can be predicted. The distribution of y^+ is called the posterior predictive distribution since it is conditional on the observed y. The posterior predictive distribution of y^+ is given as

$$\Pr(y^+ \mid y) = \int \Pr(y^+, \theta \mid y) d\theta$$

$$= \int \Pr(y^+ \mid \theta, y) \Pr(\theta \mid y) d\theta$$

$$= \int \Pr(y^+ \mid \theta) \Pr(\theta \mid y) d\theta \tag{10}$$

Bayesian methods have been widely used in the study of non-homogeneous Poisson processes, specifically, in software reliability models. Gibbs sampling technique with data augmentation and with the metropolis algorithm has been used to compute the Bayes credible sets estimates, mean time between failures and present system reliability (Lynn and Tae, 1996). This was a unified framework that incorporated four software reliability growth models namely; Duane (1964), Jelinski and Moranda (1972), Goel – Okumoto (1979) and Musa – Okumoto (1984) software reliability models.

Bayesian methods has been used as a basis for computation for the superposition of non-homogeneous Poisson processes (Tae and Lynn, 1999). Bayesian and empirical Bayes approaches have also been applied on reliability growth models based on the power law process and also in the study of microarrays (Zhao, 2004). Bayesian predictive analysis on the power law process using non – informative priors have also been conducted (Jun – Wu *et al*., 2007).

2.3 Prediction interval

A prediction interval is a confidence interval for a future observation or a function of some future observations (Jun – Wu *et al.*, 2007). Specifically, a double-sided (bilateral) prediction interval for a future failure time t_{n+k} with confidence level γ is defined by $\left[T_{n+k,L(\gamma)}, T_{n+k,U(\gamma)}\right]$ such that $\Pr\left[T_{n+k,L(\gamma)} \leq t_{n+k} \leq T_{n+k,U(\gamma)}\right] = \gamma$. Similarly, a single-sided (unilateral) lower or upper prediction limit for t_{n+k} with level γ is defined by $T_{n+k,L(\gamma)}$ (or $T_{n+k,U(\gamma)}$) which satisfies $\Pr\left[T_{n+k,L(\gamma)} \leq t_{n+k}\right] = \gamma$ (or $\Pr\left[t_{n+k} \leq T_{n+k,U(\gamma)}\right] = \gamma$). Both $T_{n+k,L(\gamma)}$ and $T_{n+k,U(\gamma)}$ depend 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.

2.4 The Goel – Okumoto (1979) software reliability Model

Many software reliability models have been developed by various authors in the past three decades. These include the Goel – Okumoto (1979), Duane (1964), Jelinsiki and Moranda (1972), and the Musa – Okumoto (1984). The models are mainly based on the history of failure of software and they can be categorized depending on the nature of the failure process studied. For instance, the Goel – Okumoto (1979), Duane (1964), and the Musa – Okumoto (1984) software reliability growth models are categorized as NHPP (Razeef and Mohsin, 2012; Amrit, 1985; Jun-Wu *et al*., 2007; Kapur *et al*., 2011; Lynn and Tae, 1996).

The Goel – Okumoto (1979) model is among the earliest exponential NHPP software reliability model to be developed. The model was proposed by Goel and Okumoto in 1979. The model has been applied to a variety of software testing environments and therefore, it can be considered as a useful reliability model (Kapur *et al*., 2011). The model is based on the following assumptions:

i. The number of failures experienced by time t follows a Poisson distribution with mean value function $m(t)$
ii. The number of software failures that occur in $(t, t + \Delta t]$ with $\Delta t \rightarrow 0$ is proportional to the number of undetected faults, $N - m(t)$ with constant of proportionality being ϕ

iii. For any finite collection of times $t_1 < t_2 < ... < t_n$, the number of failures occurring in each of the disjoint intervals $(0, t_1), (t_1, t_2), ..., (t_{n-1}, t_n)$ are independent

iv. Whenever a failure has occurred, the fault that caused it is removed instantaneously and without introducing any new fault in the software.

In this model, it is assumed that a software system is subject to failures at random time caused by faults present in the system (Goel and Okumoto, 1979). If $N(t)$, from the model, is the cumulative number of failures observed by time t, then $N(t)$ can be modeled as a non-homogeneous Poisson process. The above assumptions are best summarized by the following equations

$$\Pr[N(t) = y] = \frac{[m(t)]^y e^{-m(t)}}{y!}, \quad y = 1, 2, \ldots \tag{11}$$

$$m(t) = \alpha(1 - e^{-\beta t}) \tag{12}$$

$$\lambda(t) = m'(t) = \alpha\beta e^{-\beta t} \tag{13}$$

where $m(t)$ is the expected number of failures observed by time t and $\lambda(t)$ is the failure rate, also known as the intensity function. In this model, α is the expected number of failures to be observed eventually and β is the fault detection rate per fault. In this model, the number of faults to be detected is a random variable whose observed value is dependent on the test and other environmental factor.

The Goel – Okumoto (1979) model has been applied to a number of software testing environment (Kapur *et al*., 2011). For instance the Goel – Okumoto (1979) 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 (Satya *et al*., 2011). This has been done using the ML estimates of the parameters of the model to calculate $m(t)$ of the model and getting the respective control limits. Maximum Likelihood estimation of the parameters of the Goel – Okumoto (1979) model has been performed. 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 (Daniel and Hoang, 2001). However, the ML estimate of the failure rate, a function of the ML estimates of the Goel – Okumoto (1979) model parameters is consistent as the testing period grows large (Daniel and Hoang, 2001). An empirical method for selecting software reliability growth models for release-decision making has been given (Stringfellow and Amschler, 2002). This was achieved through applying 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. The iterative application of the mentioned models to weekly cumulative software failure data aided in determining the number of remaining failures expected in software after release.

Parameter estimation of the Goel – Okumoto (1979), Yamada S-shaped and Inflection S-shaped software reliability growth models has also been considered (Meyfroyt, 2012), where a necessary and sufficient condition with respect to the software failure data was established. The condition established ensures that the MLE method returns a unique positive and finite estimates of the unknown parameters of the models. Razeef and Mohsin (2012) 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. This indicated that a non-homogeneous Poisson process with mean value function, $m(t) = \alpha(1 - e^{-\beta t})$, corresponding to that of the Goel – Okumoto (1979) software reliability model, was a reasonable model to describe the failure process.

It is necessary to check if the data to be analyzed obey the Goel – Okumoto (1979) model. This can be done using, among other tests, the Kolmorgorov-Sminorv goodness- of-fit test for checking the adequacy of the model (Razeef and Mohsin, 2012).

2.5 Summary

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 means that predictive analyses on the model have not so far been explored. This study therefore intends to perform Bayesian predictive analyses on the Goel – Okumoto (1979) software reliability model.

CHAPTER THREE
MATERIALS AND METHODS

3.1 Research Design

The study has been confined within the limits of developing procedures that has been used to address four issues in single sample and three issues in two-sample prediction associated closely with software development testing program.

The issues addressed in one-sample software development program are:

A1: Suppose that the pre-determined target value λ_{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 λ_{tv} will be achieved at time τ, $\tau > \mathrm{T}$?

B1: Suppose that the target value λ_{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 λ_{tv} ?

C1: What is the upper prediction limit (UPL) of $\lambda_{\tau} = \alpha\beta e^{-\beta\tau}$ with level γ , τ being a predetermined value greater than T ?

D1: What is the probability that at most k software failures will occur in the future time period $(T,\tau]$, $\tau > \mathrm{T}$?

For the two-sample case, the issues addressed are:

A2: How to predict the rth $(r \geq 1)$ failure time y_r of the second software;

B2: How to predict the number of failures that will occur in the time interval $(0,t_2]$ for the Second software system.

B3: How to predict the rth, $(1 \leq \mathrm{r} \leq \mathrm{m})$ failure time y_r of the second system supposing that the number of failures in the interval $(0,t_2]$ for the second system is m but the exact occurrence times are unavailable.

Bayesian approaches based on informative and non-informative priors for the single sample case and non-informative priors for the two-sample case have been adopted to develop predictive distributions and derive explicit solutions to the above mentioned issues.

The study has used uniform non-informative prior distributions for both the parameters α and β. For the informative prior distribution, the study has assumed that the parameters α and β both follow a gamma distribution with parameters a, b and c, d respectively where a, b, c and d are known, i.e. $\alpha \sim Gamma\ (a,b)$ and $\beta \sim Gamma\ (c,d)$.

3.2 Source of data

The developed methodologies for both the single and two-sample cases have been illustrated by secondary software failure data. Data simulated from the Goel – Okumoto (1979) model has also been used to illustrate the derived methodologies. For the single sample case, the study has used secondary software failure data in the form of execution times between successive failures from one software system (Xie *et al.,* 2002). The study has assumed that the failure times follow the NHPP with intensity function given in Equation (13). For the two-sample case, the study has used two data sets simulated from the Goel – Okumoto (1979) software reliability model. The study has also assumed that the simulated failure times follow the same NHPP with intensity function given in Equation (13). Further, for the two-sample case, the study has assumed that the failure times for the first software have been observed.

3.3 Data analysis

The study has used simulated and secondary software failure data. In some cases, the closed form for the posterior distributions and predictive inferences for the one-sample and two-sample cases using both informative and non-informative priors are not available. Hence, the study has employed the MCMC technique to obtain the predictive estimates. Programs for obtaining the predictive estimates using the MCMC technique for the secondary software failure data and simulated data have been developed. Analysis for both real and simulated data has been carried out using a statistical package called R-language version 3.0.1.

CHAPTER FOUR

RESULTS AND DISCUSSIONS

4.1 Introduction

In this chapter, deviation of the methodologies that addresses the issues outlined in chapter three are given. The methodologies are given in terms of propositions. Proofs to these propositions are also given. An illustration of the derived methodologies using real data for the one-sample prediction and simulated data for the two-sample prediction is also performed.

4.2 Some results used in the derivation of the methodologies

In this section, formulas that will help to develop methodologies that will address the four issues A1, B1, C1 and D1 in single-sample prediction and three issues A2, B2 and C2 in two-sample prediction associated closely with software development testing program are formulated. The issues addressed in this chapter are as outlined in Chapter Three of this thesis. Propositions that address the issues and proof of the propositions are given. For these purposes, it is assumed that reliability growth testing is performed on a software and the number of failures of the software in the time interval $(0,t]$, denoted by $N(t)$ is observed. Furthermore, it is assumed that $\left[N(t),\ t>0\right]$ follows the NHPP with intensity function given in Equation (13). Let $0<t_1<t_2<\ldots$ be the successive failure times. Failure data is said to be failure-truncated when testing stops after a predetermined n number of failures. We denote the n failure times by $Y_{obs}^{f}=[t_i]_{i=1}^{n}$ where $0<t_1<t_2<\ldots<t_n$. Failure data is said to be time truncated if testing stops at a predetermined time t. We denote the corresponding observed failure data by $Y_{obs}^{t}=\left[n,t_1,\ldots,t_n;t\right]$, where $0<t_1<t_2<\ldots<t_n\le t$.

Let Y_{obs} represent Y_{obs}^{f} or Y_{obs}^{t}. The joint density of Y_{obs} is obtained from (5) as

$$f(Y_{obs}\mid\alpha,\beta)=\alpha^{n}\beta^{n}e^{-\beta\sum_{i=1}^{n}t_i}e^{-\alpha(1-e^{-\beta T})} \tag{14}$$

Case 1: when the shape parameter β is known, we adopt the following non informative prior distribution for α :

$$\pi(\alpha) \propto \frac{1}{\alpha}, \alpha > 0 \,. \tag{15}$$

The posterior distribution of α is thus obtained from Equation (9) as

$$h(\alpha \mid Y_{obs}) = \frac{f(Y_{obs} \mid \alpha, \beta)\pi(\alpha)}{\int_0^\infty f(Y_{obs} \mid \alpha, \beta)\pi(\alpha)d\alpha} \,.$$

Using Equation (14) and Equation (15), we obtain

$$h(\alpha \mid Y_{obs}) = \frac{\alpha^{n-1}\beta^n e^{-\beta\sum_{i=1}^{n} t_i} e^{-\alpha(1-e^{-\beta T})}}{\int_0^\infty \alpha^{n-1}\beta^n e^{-\beta\sum_{i=1}^{n} t_i} e^{-\alpha(1-e^{-\beta T})}d\alpha} \,. \tag{16}$$

Simplifying the denominator, we have

$$\int_0^\infty \alpha^{n-1}\beta^n e^{-\beta\sum_{i=1}^{n} t_i} e^{-\alpha(1-e^{-\beta T})}d\alpha = \beta^n e^{-\beta\sum_{i=1}^{n} t_i} \int_0^\infty \alpha^{n-1} e^{-\alpha(1-e^{-\beta T})}d\alpha$$

$$= \beta^n e^{-\beta\sum_{i=1}^{n} t_i} \frac{\Gamma(n)}{\left(1-e^{-\beta T}\right)^n} \int_0^\infty \frac{\left(1-e^{-\beta T}\right)^n}{\Gamma(n)} \alpha^{n-1} e^{-\alpha(1-e^{-\beta T})}d\alpha \tag{17}$$

$$= \frac{\Gamma(n)\beta^n e^{-\beta\sum_{i=1}^{n} t_i}}{\left(1-e^{-\beta T}\right)^n} \tag{18}$$

The integral part of Equation (17) integrates to 1 since it is a gamma distribution with parameters n and $1-e^{-\beta T}$. Hence the denominator of Equation (16) reduces to Equation (18). Equation (16) therefore becomes $h(\alpha \mid Y_{obs}) = \dfrac{\alpha^{n-1}\beta^n e^{-\beta\sum_{i=1}^{n} t_i} e^{-\alpha(1-e^{-\beta T})}}{\Gamma(n)\beta^n e^{-\beta\sum_{i=1}^{n} t_i} \big/ \left(1-e^{-\beta T}\right)^n}$ which reduces to

$$h(\alpha \mid Y_{obs}) = \left[\Gamma(n)\right]^{-1} \alpha^{n-1} e^{-\alpha(1-e^{-\beta T})} (1-e^{-\beta T})^{n} \tag{19}$$

Let y^{+} be the random variable being predicted. Then using Equation (10), the posterior predictive distribution of y^{+} is given as

$$f(y^{+} \mid Y_{obs}) = \int_{0}^{\infty} f(y^{+} \mid Y_{obs}, \alpha) h(\alpha \mid Y_{obs}) d\alpha \tag{20}$$

Hence the Bayesian UPL for y^{+} with level γ must satisfy

$$\gamma = \int_{-\infty}^{y_U^{(\beta)}} f(y^{+} \mid Y_{obs}) dy^{+} . \tag{21}$$

Case 2: when the shape parameter β is unknown, we consider the following non-informative joint prior density for α and β (we assume that α and β are independent).

$$\pi(\alpha, \beta) \propto \frac{1}{\alpha\beta}, \quad \alpha, \beta > 0 . \tag{22}$$

Hence the corresponding joint posterior density is obtained as

$$h(\alpha, \beta \mid Y_{obs}) = \frac{\alpha^{n-1} \beta^{n-1} e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha(1-e^{-\beta T})}}{\int_{0}^{\infty}\int_{0}^{\infty} \alpha^{n-1} \beta^{n-1} e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha(1-e^{-\beta T})} d\alpha\, d\beta} . \tag{23}$$

The denominator of Equation (23) reduces to $\Gamma(n) \int_{0}^{\infty} \frac{\beta^{n-1} e^{-\beta \sum_{i=1}^{n} t_i}}{\left(1-e^{-\beta T}\right)^{n}} d\beta$ after applying the results from Equation (17) and Equation (18). Hence Equation (23) reduces to

$$h(\alpha,\beta \mid Y_{obs}) = \frac{\alpha^{n-1}\beta^{n-1}e^{-\beta\sum_{i=1}^{n}t_i}e^{-\alpha(1-e^{-\beta T})}}{\Gamma(n)\int_0^\infty \frac{\beta^{n-1}e^{-\beta\sum_{i=1}^{n}t_i}}{\left(1-e^{-\beta T}\right)^n}d\beta}. \tag{24}$$

The integration $\int_0^\infty \frac{\beta^{n-1}e^{-\beta\sum_{i=1}^{n}t_i}}{\left(1-e^{-\beta T}\right)^n}d\beta$ in the denominator of Equation (24) has no closed form and hence the study will employ numerical technique (MCMC) to obtain its value. Let the value of this integral be denoted by a constatnt k. hence $k = \int_0^\infty \frac{\beta^{n-1}e^{-\beta\sum_{i=1}^{n}t_i}}{\left(1-e^{-\beta T}\right)^n}d\beta$. Hence Equation (24) becomes

$$h(\alpha,\beta \mid Y_{obs}) = \left[k\Gamma(n)\right]^{-1}\alpha^{n-1}\beta^{n-1}e^{-\beta\sum_{i=1}^{n}t_i}e^{-\alpha(1-e^{-\beta T})}. \tag{25}$$

Similar to Equation (20) and Equation (21), Let yU denote the Bayesian UPL of y with level γ, then the posterior distribution of y^+ is

$$f(y^+ \mid Y_{obs}) = \int_0^\infty\int_0^\infty f(y^+ \mid Y_{obs},\alpha,\beta)h(\alpha,\beta \mid Y_{obs})d\alpha\, d\beta \tag{26}$$

and

$$\gamma = \int_{-\infty}^{yU} f(y^+ \mid Y_{obs})dy^+. \tag{27}$$

4.3 Main results for single-sample prediction using non-informative prior

4.3.1 Proposition A1

The probability that the target value λ_{tv} will be achieved at time τ $(\tau > T)$ is

$$\gamma = \begin{cases} 1-\sum_{h=0}^{n-1}\dfrac{\left(\dfrac{1-e^{-\beta T}}{\beta e^{-\beta\tau}}\lambda_{tv}\right)^{h}}{h!}e^{-\frac{1-e^{-\beta T}}{\beta e^{-\beta\tau}}\lambda_{tv}} & \text{if } \beta \text{ is known} \\ 1-\dfrac{1}{k}\sum_{h=0}^{n-1}\int_{0}^{\infty}\dfrac{\left(\dfrac{1-e^{-\beta T}}{\beta e^{-\beta\tau}}\lambda_{tv}\right)^{h}}{h!}e^{-\frac{(1-e^{-\beta T})}{\beta e^{-\beta\tau}}\lambda_{tv}}\dfrac{\beta^{n-1}e^{-\beta\sum_{i=1}^{n}t_i}}{(1-e^{-\beta T})^{n}}d\beta & \text{if } \beta \text{ is unknown} \end{cases} \qquad \text{Prop. A1}$$

Proof

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 λ_{tv} will be achieved at time τ ($\tau > T$) is given by

$$\gamma = \Pr(\lambda_\tau \le \lambda_{tv} \mid Y_{obs}) = \int_{0}^{\lambda_{tv}} f(\lambda_\tau \mid Y_{obs})d\lambda_\tau \ . \tag{28}$$

When β is known, making the transformation $\lambda_\tau = \alpha\beta e^{-\beta\tau}$, we have $\alpha = \dfrac{\lambda_\tau}{\beta e^{-\beta\tau}}$ and $\dfrac{d\alpha}{d\lambda_\tau} = \dfrac{1}{\beta e^{-\beta\tau}}$. Consequently, the posterior density of λ_τ is $f(\lambda_\tau \mid Y_{obs}) = h(\alpha \mid Y_{obs})\left|\dfrac{d\alpha}{d\lambda_\tau}\right|$. This implies that $f(\lambda_\tau \mid Y_{obs}) = \left(\dfrac{\lambda_\tau}{\beta e^{-\beta\tau}}\right)^{n-1}\dfrac{1}{\Gamma(n)}e^{-\frac{\lambda_\tau}{\beta e^{-\beta\tau}}(1-e^{-\beta T})}\left(1-e^{-\beta T}\right)^{n}.\dfrac{1}{\beta e^{-\beta\tau}}$ which after simplification reduces to

$$f(\lambda_\tau \mid Y_{obs}) = \frac{\left(\dfrac{1-e^{-\beta T}}{\beta e^{-\beta\tau}}\right)^{n}}{\Gamma(n)}\lambda_\tau^{\,n-1}e^{-\lambda_\tau\left(\frac{1-e^{-\beta T}}{\beta e^{-\beta\tau}}\right)}. \tag{29}$$

We note that λ_τ from Equation (29) follows a gamma distribution with parameters n and $\dfrac{1-e^{-\beta T}}{\beta e^{-\beta\tau}}$. Noting that gamma and Poisson distributions have a relationship defined as

$$\frac{\beta^{\alpha}}{\Gamma(\alpha)}\int_{0}^{\lambda} x^{\alpha-1}e^{-\beta x}dx = 1-\sum_{h=0}^{\alpha-1}\frac{(\beta\lambda)^{h}}{h!}e^{-\beta\lambda} \tag{30}$$

and from Equation (28) and Equation (29), we obtain

$$\gamma = 1-\sum_{h=0}^{n-1}\frac{\left(\frac{1-e^{-\beta T}}{\beta e^{-\beta\tau}}\lambda_{tv}\right)^{h}}{h!}e^{-\frac{1-e^{-\beta T}}{\beta e^{-\beta\tau}}\lambda_{tv}}. \tag{31}$$

Equation (31) implies the first part of the formula in Prop. A1

When β 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 (25), the joint posterior density of (λ_{τ},β) is given as

$$f(\lambda_{\tau},\beta \mid Y_{obs}) = h(\alpha,\beta \mid Y_{obs})\left|\frac{d(\alpha,\beta)}{d(\lambda_{\tau},\beta)}\right|$$

$$= \left(\frac{\lambda_{\tau}}{\beta e^{-\beta\tau}}\right)^{n-1}\frac{\beta^{n-1}}{k\Gamma(n)}e^{-\beta\sum_{i=1}^{n}t_i}e^{-\lambda_{\tau}\frac{(1-e^{-\beta T})}{\beta e^{-\beta\tau}}}\frac{1}{\beta e^{-\beta\tau}}$$

$$= \left(\frac{1}{\beta e^{-\beta\tau}}\right)^{n-1}\frac{\beta^{n-1}}{k\Gamma(n)}e^{-\beta\sum_{i=1}^{n}t_i}\lambda_{\tau}^{n-1}e^{-\lambda_{\tau}\frac{(1-e^{-\beta T})}{\beta e^{-\beta\tau}}}\frac{1}{\beta e^{-\beta\tau}}$$

$$= \left(\frac{1}{\beta e^{-\beta\tau}}\right)^{n}\frac{\beta^{n-1}}{k\Gamma(n)}e^{-\beta\sum_{i=1}^{n}t_i}\lambda_{\tau}^{n-1}e^{-\lambda_{\tau}\frac{(1-e^{-\beta T})}{\beta e^{-\beta\tau}}}$$

$$= \frac{(\beta e^{-\beta\tau})^{-n}}{k\Gamma(n)}\beta^{n-1}e^{-\beta\sum_{i=1}^{n}t_i}\frac{\Gamma(n)}{\left(\frac{1-e^{-\beta T}}{\beta e^{-\beta\tau}}\right)^{n}}\frac{\left(\frac{1-e^{-\beta T}}{\beta e^{-\beta\tau}}\right)^{n}}{\Gamma(n)}\lambda_{\tau}^{n-1}e^{-\lambda_{\tau}\frac{(1-e^{-\beta T})}{\beta e^{-\beta\tau}}}$$

$$= \frac{\beta^{n-1} e^{-\beta \sum_{i=1}^{n} t_i}}{k(1-e^{-\beta T})^n} \frac{\left(\frac{1-e^{-\beta T}}{\beta e^{-\beta \tau}}\right)^n}{\Gamma(n)} \lambda_\tau^{n-1} e^{-\lambda_\tau \frac{(1-e^{-\beta T})}{\beta e^{-\beta \tau}}}. \tag{32}$$

From Equations (28), (30) and (32) we obtain

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

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

$$\gamma = 1 - \frac{1}{k} \sum_{h=0}^{n-1} \int_0^\infty \frac{\left(\frac{1-e^{-\beta T}}{\beta e^{-\beta \tau}} \lambda_{tv}\right)^h}{h!} e^{-\frac{(1-e^{-\beta T})}{\beta e^{-\beta \tau}} \lambda_{tv}} \frac{\beta^{n-1} e^{-\beta \sum_{i=1}^{n} t_i}}{(1-e^{-\beta T})^n} d\beta. \tag{33}$$

Equation (33) implies the formula in the second part of Prop. A1

4.3.2 Proposition B1

For given level γ, the time τ^* required to attain λ_{tv} is

$$\tau^* = \begin{cases} \frac{1}{\beta} \ln\left[\frac{\beta \chi^2(2n;\gamma)}{2\lambda_{tv}(1-e^{-\beta T})}\right] - T & \text{if } \beta \text{ is known} \\ \tau - T & \text{if } \beta \text{ is unknown} \end{cases} \qquad \text{Prop. B1}$$

Remark 1: The second part of Prop. B1 is such that τ is the solution to the equation

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

Proof

For given level γ, the time required to attain the target value λ_{tv} is $\tau^* = \tau - T$ where τ satisfies Equation (28). When β is known, from Equation (29), it can easily be seen that $2\left(\frac{1-e^{-\beta T}}{\beta e^{-\beta\tau}}\right)\lambda_\tau$ follows a chi-square distribution with $2n$ degrees of freedom. Therefore, we have

$$2\left(\frac{1-e^{-\beta T}}{\beta e^{-\beta\tau}}\right)\lambda_{tv} = \chi^2(2n;\gamma) \tag{34}$$

or

$$\frac{1-e^{-\beta T}}{\beta e^{-\beta\tau}} = \frac{\chi^2(2n;\gamma)}{2\lambda_{tv}}. \tag{35}$$

Making τ the subject from Equation (35) we obtain

$$\tau = \frac{1}{\beta}\ln\left[\frac{\beta\chi^2(2n;\gamma)}{2\lambda_{tv}(1-e^{-\beta T})}\right]. \tag{36}$$

Therefore, from Equation (36), we obtain the time required to attain the target value λ_{tv} for a given level γ as

$$\tau^* = \frac{1}{\beta}\ln\left[\frac{\beta\chi^2(2n;\gamma)}{2\lambda_{tv}(1-e^{-\beta T})}\right] - T. \tag{37}$$

Equation (37) implies the first part of the formula in Prop. B2

When β is unknown, the time required to attain the target value λ_{tv} with level γ is $\tau^* = \tau - T$ where τ is the solution to

$$\gamma = 1 - \frac{1}{k}\sum_{h=0}^{n-1}\int_0^\infty \frac{\left(\frac{1-e^{-\beta T}}{\beta e^{-\beta\tau}}\lambda_{tv}\right)^h}{h!} e^{-\frac{(1-e^{-\beta T})}{\beta e^{-\beta\tau}}\lambda_{tv}} \frac{\beta^{n-1}e^{-\beta\sum_{i=1}^n t_i}}{(1-e^{-\beta T})^n} d\beta. \tag{38}$$

4.3.3 Proposition C1

The Bayesian UPL of $\lambda_\tau = \alpha\beta e^{-\beta\tau}$ with level γ is

$$\lambda_U^{(\beta)} = \begin{cases} \dfrac{\chi^2(2n;\gamma)\beta e^{-\beta\tau}}{2(1-e^{-\beta T})} & \text{if } \beta \text{ is known} \\ \lambda_{tv} & \text{if } \beta \text{ is unknown} \end{cases} \qquad \text{Prop. C1}$$

Remark 2: The second part of Prop. C1 is such that λ_{tv} is the solution to the equation

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

Proof

For a pre-determined τ $(\tau > T)$, the Bayesian Upper Prediction Limit (UPL) for λ_τ with level γ is $\lambda_U^{(\beta)}(\tau)$ satisfying $\gamma = \Pr(\lambda_\tau \le \lambda_U^{(\beta)}(\tau) \,|\, Y_{obs})$. From Equation (28) and Equation (34) we have

$$\gamma = \int_0^{\lambda_\tau^{(\beta)}(\tau)} \frac{\left(\dfrac{1-e^{-\beta T}}{\beta e^{-\beta\tau}}\right)^n}{\Gamma(n)} \lambda_\tau^{n-1} e^{-\lambda_\tau \frac{(1-e^{-\beta T})}{\beta e^{-\beta\tau}}}\, d\lambda_\tau \;.$$

This implies that

$$2\left(\frac{1-e^{-\beta T}}{\beta e^{-\beta\tau}}\right)\lambda_U^{(\beta)}(\tau) = \chi^2(2n;\gamma). \tag{39}$$

Making $\lambda_U^{(\beta)}(\tau)$ the subject of Equation (39) we arrive at

$$\lambda_U^{(\beta)}(\tau) = \frac{\chi^2(2n;\gamma)\beta e^{-\beta\tau}}{2(1-e^{-\beta T})}. \tag{40}$$

Therefore, for a pre-determined τ $(\tau > T)$, the Bayesian UPL for λ_τ with level γ is $\lambda_U^{(\beta)}(\tau)$ as in Equation (40) which is as indicated in the first part of Prop. C1.

when β is unknown, the Bayesian UPL for $\lambda_\tau = \alpha\beta e^{-\beta\tau}$ with level γ is λ_{tv} where λ_{tv} is a solution to

$$\gamma = 1 - \frac{1}{k} \sum_{h=0}^{n-1} \int_0^{\infty} \frac{\left(\frac{1 - e^{-\beta T}}{\beta e^{-\beta \tau}} \lambda_{tv} \right)^h}{h!} e^{-\frac{(1-e^{-\beta T})}{\beta e^{-\beta \tau}} \lambda_{tv}} \frac{\beta^{n-1} e^{-\beta \sum_{i=1}^{n} t_i}}{(1 - e^{-\beta T})^n} d\beta. \tag{41}$$

Remark 3: The UPL with level γ is equal to the lower prediction limit with level $1-\gamma$

4.3.4 Proposition D1

The probability that at most k failures will occur in the time interval $(T, \tau]$ with $\tau > T$ is

$$\gamma_k = \begin{cases} \left(\frac{1 - e^{-\beta T}}{e^{-\beta T} - e^{-\beta \tau}} \right)^n \sum_{j=n}^{n+k} \binom{j-1}{n-1} \left(\frac{e^{-\beta T} - e^{-\beta \tau}}{1 - e^{-\beta \tau}} \right)^j & \text{if } \beta \text{ is known} \\ \sum_{j=n}^{n+k} \frac{\Gamma(n)}{d\,(j-n)!\,\Gamma(n)} \int_0^{\infty} \beta^{n-1} e^{-\beta \sum_{i=1}^{n} t_i} \frac{\left(e^{-\beta T} - e^{-\beta \tau}\right)^{j-n}}{\left(1 - e^{-\beta \tau}\right)^j} d\beta & \text{if } \beta \text{ is unknown} \end{cases} \quad \text{Prop. D1}$$

Proof

The probability that at most k failures will occur in the interval (T, τ) is $\gamma_k = \Pr\left[N(\tau) \le n + k \mid Y_{obs} \right]$. When β is known, we have

$$\gamma_k = \int_0^{\infty} \Pr\left[N(\tau) \le n + k \mid Y_{obs}, \alpha \right] h(\alpha \mid Y_{obs}) d\alpha \tag{42}$$

where $h(\alpha \mid Y_{obs})$ is given by (19) and

$$\Pr[N(\tau) \le n + k \mid Y_{obs}, \alpha] = \sum_{j=n}^{n+k} f(Y_{obs}, N(\tau) = j \mid \alpha) \,/\, f(Y_{obs} \mid \alpha). \tag{43}$$

From (14) we have $f(Y_{obs} \mid \alpha) = \alpha^n \beta^n e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha(1 - e^{-\beta T})}$ and

$$f(Y_{obs}, N(\tau) = j \mid \alpha) = \int_{D(j-n:T,\tau)} f(Y_{obs}, x_{n+1}, \ldots, x_j, N(\tau) = j) \prod_{\ell=n+1}^{j} dx_\ell$$

$$f(Y_{obs}, N(\tau) = j \mid \alpha) = \int_{D(j-n;T,\tau)} \alpha^j \beta^j e^{-\beta \sum_{i=1}^{j} t_i} e^{-\alpha(1-e^{\beta\tau})} \prod_{\ell=n+1}^{j} dt_\ell$$

$$= \alpha^j \beta^j e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha(1-e^{-\beta\tau})} \int_{D(j-n;T,\tau)} e^{-\beta \sum_{\ell=n+1}^{j} t_\ell} \prod_{\ell=n+1}^{j} dt_\ell . \qquad (44)$$

Solving the integral part, we proceed as follows:

$\int_0^t e^{-\beta t} dt = \frac{1}{\beta}(1-e^{-\beta t})$. Substituting the limits T and τ we obtain $\frac{1}{\beta}(1-e^{-\beta\tau}) - \frac{1}{\beta}(1-e^{-\beta T})$ which reduces to $\frac{1}{\beta}(e^{-\beta T} - e^{-\beta\tau})$. Hence the integral part of Equation (44) is obtained as

$$\int_{D(j-n;T,\tau)} e^{-\beta \sum_{\ell=n+1}^{j} t_\ell} \prod_{\ell=n+1}^{j} dt_\ell = \frac{1}{\beta^{j-n}} \frac{(e^{-\beta T} - e^{-\beta\tau})^{j-n}}{(j-n)!}. \qquad (45)$$

Substituting Equation (45) into Equation (44) we obtain

$$f(Y_{obs}, N(\tau) = j \mid \alpha) = \alpha^j \beta^j e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha(1-e^{-\beta\tau})} \frac{1}{\beta^{j-n}} \frac{(e^{-\beta T} - e^{-\beta\tau})^{j-n}}{(j-n)!}$$

From Equation (43) we have

$$f(Y_{obs}, N(\tau) = j \mid \alpha) \,/\, f(Y_{obs} \mid \alpha) = \frac{\alpha^j \beta^n e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha(1-e^{-\beta\tau})} \frac{(e^{-\beta T} - e^{-\beta\tau})^{j-n}}{(j-n)!}}{\alpha^n \beta^n e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha(1-e^{-\beta T})}}$$

which reduces to

$$f(Y_{obs}, N(\tau) = j \mid \alpha) \,/\, f(Y_{obs} \mid \alpha) = \alpha^{j-n} e^{-\alpha(e^{-\beta T} - e^{-\beta\tau})} \frac{(e^{-\beta T} - e^{-\beta\tau})^{j-n}}{(j-n)!} .$$

Therefore, Equation (43) becomes

$$\Pr[N(\tau) \le n+k \mid Y_{obs}, \alpha] = \sum_{j=n}^{n+k} \alpha^{j-n} e^{-\alpha(e^{-\beta T}-e^{-\beta\tau})} \frac{(e^{-\beta T}-e^{-\beta\tau})^{j-n}}{(j-n)!} \tag{46}$$

And Equation (42) becomes

$$\gamma_k = \int_0^\infty \sum_{j=n}^{n+k} \alpha^{j-n} e^{-\alpha(e^{-\beta T}-e^{-\beta\tau})} \frac{(e^{-\beta T}-e^{-\beta\tau})^{j-n}}{(j-n)!\Gamma(n)} \alpha^{n-1} e^{-\alpha(1-e^{-\beta T})}(1-e^{-\beta T})^n d\alpha$$

$$= \sum_{j=n}^{n+k} \frac{(e^{-\beta T}-e^{-\beta\tau})^{j-n}(1-e^{-\beta T})^n}{(j-n)!\Gamma(n)} \int_0^\infty \alpha^{j-1} e^{-\alpha(1-e^{-\beta\tau})} d\alpha$$

$$= \sum_{j=n}^{n+k} \frac{\left(e^{-\beta T}-e^{-\beta\tau}\right)^{j-n}\left(1-e^{-\beta T}\right)^n}{(j-n)!\Gamma(n)} \frac{\Gamma(j)}{\left(1-e^{-\beta\tau}\right)^j} \int_0^\infty \frac{\left(1-e^{-\beta\tau}\right)^j}{\Gamma(j)} \alpha^{j-1} e^{-\alpha\left(1-e^{-\beta\tau}\right)} d\alpha \tag{47}$$

$$= \sum_{j=n}^{n+k} \frac{\left(e^{-\beta T}-e^{-\beta\tau}\right)^{j-n}\left(1-e^{-\beta T}\right)^n}{(j-n)!\Gamma(n)} \frac{\Gamma(j)}{\left(1-e^{-\beta\tau}\right)^j}. \tag{48}$$

(Since the integral part of Equation (47) integrates to 1 from the fact that it is a gamma distribution with parameters j and $(1-e^{-\beta\tau})$

After re-arranging Equation (48) we obtain

$$\gamma_k = \left(\frac{1-e^{-\beta T}}{e^{-\beta T}-e^{-\beta\tau}}\right)^n \sum_{j=n}^{n+k} \binom{j-1}{n-1} \left(\frac{e^{-\beta T}-e^{-\beta\tau}}{1-e^{-\beta\tau}}\right)^j. \tag{49}$$

Hence, the first formula of Prop. D1 follows.

When β is unknown, noting that $\Pr\left[N(\tau) \le n+k \mid Y_{obs}, \alpha, \beta\right]$ and $h(\alpha, \beta \mid Y_{obs})$ are given by Equation (46) and Equation (25) respectively, we have

$$\gamma_k = \int_0^\infty \int_0^\infty \Pr\left[N(\tau) \le n+k \mid Y_{obs}, \alpha, \beta\right] h(\alpha, \beta \mid Y_{obs}) d\alpha\, d\beta$$

$$= \sum_{j=n}^{n+k} \frac{1}{d\left(j-n\right)!\Gamma(n)} \int_0^{\infty}\int_0^{\infty} \alpha^{j-n} e^{-\alpha(e^{-\beta T}-e^{-\beta\tau})} \left(e^{-\beta T}-e^{-\beta\tau}\right)^{j-n} \alpha^{n-1}\beta^{n-1} e^{-\beta\sum_{i=1}^{n} t_i} e^{-\alpha(1-e^{-\beta T})} d\alpha\, d\beta$$

$$= \sum_{j=n}^{n+k} \frac{\Gamma(n)}{d\left(j-n\right)!\Gamma(n)} \int_0^{\infty} \beta^{n-1} e^{-\beta\sum_{i=1}^{n} t_i} \frac{\left(e^{-\beta T}-e^{-\beta\tau}\right)^{j-n}}{\left(1-e^{-\beta\tau}\right)^{j}} d\beta. \tag{50}$$

Where $d = k$ as is used in Equation (25). Letter d has been substituted for k in Equation (50) since the summation is from n to $n+k$ and the $k's$ are not the same. Equation (50) implies the second formula in Prop. D1

4.4 Main results for single-sample prediction using informative priors

4.4.1 Some important results for the derivation of the methodologies

In this section, we will still use the joint density of Y_{obs} as is given in Equation (14).

Case 1: When the shape parameter is known, the study will adopt the following informative prior distribution. The study assumes that $\alpha \sim Gamma(a,b)$, where a and b are known.

$$\pi\left(\alpha\right) \propto \alpha^{a-1} e^{-b\alpha} \tag{51}$$

The posterior distribution of α is thus obtained from Equation (9) as

$$h\left(\alpha \mid Y_{obs}\right) = \frac{f\left(Y_{obs} \mid \alpha,\beta\right)\pi\left(\alpha\right)}{\int_0^{\infty} f\left(Y_{obs} \mid \alpha,\beta\right)\pi\left(\alpha\right) d\alpha}.$$

Using Equation (14) and Equation (51), we obtain

$$h(\alpha \mid Y_{obs}) = \frac{\beta^{n} e^{-\beta\sum_{i=1}^{n} t_i} \alpha^{n+a-1} e^{-\alpha(1-e^{-\beta T}+b)}}{\beta^{n} e^{-\beta\sum_{i=1}^{n} t_i} \int_0^{\infty} \alpha^{n+a-1} e^{-\alpha(1-e^{-\beta T}+a)} d\alpha} \tag{52}$$

After evaluating the denominator, Equation (52) reduces to

$$h(\alpha \mid Y_{obs}) = \left[\Gamma(n+a)\right]^{-1} \alpha^{n+a-1} e^{-\alpha(1-e^{-\beta T}+b)} (1-e^{-\beta T}+b)^{n+a} \tag{53}$$

Case 2: When the shape parameter β is unknown, the study derives the joint informative prior distribution of α and β as follows,

We assume that $\alpha \sim Gamma(a,b)$ and $\beta \sim Gamma(c,d)$. This implies that $\pi(\alpha) \propto \alpha^{a-1}e^{-b\alpha}$ and $\pi(\beta) \propto \beta^{c-1}e^{-d\beta}$. Since α and β are independent, the joint prior density $\pi(\alpha,\beta)$ is given as $\pi(\alpha) \propto \pi(\alpha)\pi(\beta)$ which implies that

$$\pi(\alpha,\beta) \propto \alpha^{a-1}e^{-b\alpha}\beta^{c-1}e^{-d\beta} \tag{54}$$

From Equation (14) and Equation (54), we get the corresponding joint posterior density of α and β as

$$h(\alpha,\beta \mid Y_{obs}) = \frac{\alpha^{a-1}e^{-b\alpha}\beta^{c-1}e^{-d\beta}\alpha^{n}\beta^{n}e^{-\beta\sum_{i=1}^{n}t_i}e^{-\alpha(1-e^{-\beta T})}}{\int_0^\infty\int_0^\infty \alpha^{a-1}e^{-b\alpha}\beta^{c-1}e^{-d\beta}\alpha^{n}\beta^{n}e^{-\beta\sum_{i=1}^{n}t_i}e^{-\alpha(1-e^{-\beta T})}d\alpha\, d\beta}$$

$$= \frac{\alpha^{n+a-1}\beta^{n+c-1}e^{-\beta(\sum_{i=1}^{n}t_i+d)}e^{-\alpha(1-e^{-\beta T}+b)}}{\int_0^\infty\int_0^\infty \alpha^{n+a-1}\beta^{n+c-1}e^{-\beta(\sum_{i=1}^{n}t_i+d)}e^{-\alpha(1-e^{-\beta T}+b)}d\alpha\, d\beta}$$

$$= \frac{\alpha^{n+a-1}\beta^{n+c-1}e^{-\beta(\sum_{i=1}^{n}t_i+d)}e^{-\alpha(1-e^{-\beta T}+b)}}{\left\{\int_0^\infty \beta^{n+c-1}e^{-\beta(\sum_{i=1}^{n}t_i+d)}\left[\int_0^\infty \alpha^{n+a-1}e^{-\alpha(1-e^{-\beta T}+b)}d\alpha\right]\right\}d\beta}$$

$$= \frac{\alpha^{n+a-1}\beta^{n+c-1}e^{-\beta(\sum_{i=1}^{n}t_i+d)}e^{-\alpha(1-e^{-\beta T}+b)}}{\Gamma(n+a)\int_0^\infty \frac{\beta^{n+c-1}e^{-\beta(\sum_{i=1}^{n}t_i+d)}}{\left(1-e^{-\beta T}+b\right)^{n+a}}d\beta}. \tag{55}$$

Letting $p = \int_0^\infty \frac{\beta^{n+c-1} e^{-\beta(\sum_{i=1}^n t_i + d)}}{\left(1 - e^{-\beta T} + b\right)^{n+a}} d\beta$, Equation (55) becomes

$$h\left(\alpha, \beta \mid Y_{obs}\right) = \left[p\Gamma(n+a)\right]^{-1} \alpha^{n+a-1} \beta^{n+c-1} e^{-\beta(\sum_{i=1}^n t_i + d)} e^{-\alpha(1-e^{-\beta T}+b)} \tag{56}$$

4.4.2 Proposition A1.1

The probability that the target value λ_{tv} will be achieved at time τ $(\tau > T)$ is

$$\gamma = \begin{cases} 1 - \sum_{h=0}^{n+a-1} \frac{\left(\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}} \lambda_{tv} \right)^h}{h!} e^{-\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}} \lambda_{tv}} & \text{if } \beta \text{ is known} \\ 1 - \frac{1}{p} \sum_{h=0}^{n+a-1} \int_0^\infty \frac{\left(\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}} \lambda_{tv} \right)^h}{h!} e^{-\frac{(1-e^{-\beta T}+b)}{\beta e^{-\beta\tau}} \lambda_{tv}} \frac{\beta^{n+c-1} e^{-\beta(\sum_{i=1}^n t_i + d)}}{(1-e^{-\beta T}+b)^{n+a}} d\beta & \text{if } \beta \text{ is unknown} \end{cases} \tag{Prop.A1.1}$$

Proof

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 λ_{tv} will be achieved at time τ ($\tau > T$) is given by Equation (28).

When β 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 λ_τ is $f(\lambda_\tau \mid Y_{obs}) = h(\alpha \mid Y_{obs}) \left| \frac{d\alpha}{d\lambda_\tau} \right|$. This implies that $f(\lambda_\tau \mid Y_{obs}) = \left(\frac{\lambda_\tau}{\beta e^{-\beta\tau}} \right)^{n+a-1} \frac{1}{\Gamma(n+a)} e^{-\frac{\lambda_\tau}{\beta e^{-\beta\tau}}(1-e^{-\beta T}+b)} \left(1 - e^{-\beta T} + b\right)^{n+a} . \frac{1}{\beta e^{-\beta\tau}}$ which after simplification reduces to

$$f(\lambda_\tau \mid Y_{obs}) = \frac{\left(\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}} \right)^{n+a}}{\Gamma(n+a)} \lambda_\tau^{n+a-1} e^{-\lambda_\tau \left(\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}} \right)}. \tag{57}$$

We note that λ_τ from Equation (57) follows a gamma distribution with parameters $n+a$ and $\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}}$. Noting the relationship between gamma and Poisson distributions as

$$\frac{\beta^\alpha}{\Gamma(\alpha)}\int_0^\lambda x^{\alpha-1}e^{-\beta x}dx = 1-\sum_{h=0}^{\alpha-1}\frac{(\beta\lambda)^h}{h!}e^{-\beta\lambda} \tag{58}$$

and from Equations (28) and Equation (57), we obtain

$$\gamma = 1-\sum_{h=0}^{n+a-1}\frac{\left(\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}}\lambda_{tv}\right)^h}{h!}e^{-\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}}\lambda_{tv}}. \tag{59}$$

Equation (59) is exactly the same as the first formula in Prop A1.1

When β 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 (56), the joint posterior density of (λ_τ,β) is given as

$$f(\lambda_\tau,\beta\mid Y_{obs}) = h(\alpha,\beta\mid Y_{obs})\left|\frac{d(\alpha,\beta)}{d(\lambda_\tau,\beta)}\right|$$

$$=\left(\frac{\lambda_\tau}{\beta e^{-\beta\tau}}\right)^{n+a-1}\frac{\beta^{n+c-1}}{p\Gamma(n+a)}e^{-\beta(\sum_{i=1}^n t_i+d)}e^{-\lambda_\tau\frac{(1-e^{-\beta T}+b)}{\beta e^{-\beta\tau}}}\cdot\frac{1}{\beta e^{-\beta\tau}}$$

$$=\left(\frac{1}{\beta e^{-\beta\tau}}\right)^{n+a-1}\frac{\beta^{n+c-1}}{p\Gamma(n+a)}e^{-\beta(\sum_{i=1}^n t_i+d)}\lambda_\tau^{n+a-1}e^{-\lambda_\tau\frac{(1-e^{-\beta T}+b)}{\beta e^{-\beta\tau}}}\cdot\frac{1}{\beta e^{-\beta\tau}}$$

$$=\left(\beta e^{-\beta\tau}\right)^{-(n+a)}\frac{\beta^{n+c-1}}{p\Gamma(n+a)}e^{-\beta(\sum_{i=1}^n t_i+d)}\lambda_\tau^{n+a-1}e^{-\lambda_\tau\frac{(1-e^{-\beta T}+b)}{\beta e^{-\beta\tau}}}$$

$$= \frac{(\beta e^{-\beta\tau})^{-(n+a)}}{p\Gamma(n+a)} \beta^{n+c-1} e^{-\beta(\sum_{i=1}^{n} t_i + d)} \frac{\Gamma(n+a)}{\left(\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}}\right)^{n+a}} \frac{\left(\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}}\right)^{n+a}}{\Gamma(n+a)} \lambda_\tau^{n+a-1} e^{-\lambda_\tau \frac{(1-e^{-\beta T}+b)}{\beta e^{-\beta\tau}}}$$

$$= \frac{\beta^{n+c-1} e^{-\beta(\sum_{i=1}^{n} t_i + d)}}{p(1-e^{-\beta T}+b)^{n+a}} \frac{\left(\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}}\right)^{n}}{\Gamma(n+a)} \lambda_\tau^{n+a-1} e^{-\lambda_\tau \frac{(1-e^{-\beta T}+b)}{\beta e^{-\beta\tau}}} . \tag{60}$$

From Equation (28), Equation (30) and Equation (60) we obtain

$$\gamma = \frac{1}{P} \int_0^\infty \frac{\beta^{n+c-1} e^{-\beta(\sum_{i=1}^{n} t_i + d)}}{\left(1-e^{-\beta T}+b\right)^{n+a}} \left\{ 1 - \sum_{h=0}^{n+a-1} \frac{\left(\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}} \lambda_{tv}\right)^h}{h!} e^{-\frac{(1-e^{-\beta T}+b)}{\beta e^{-\beta\tau}} \lambda_{tv}} \right\} d\beta$$

$$= \frac{1}{p} \int_0^\infty \frac{\beta^{n+c-1} e^{-\beta(\sum_{i=1}^{n} t_i + d)}}{(1-e^{-\beta T}+b)^{n+a}} d\beta - \frac{1}{p} \sum_{h=0}^{n+a-1} \int_0^\infty \frac{\left(\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}} \lambda_{tv}\right)^h}{h!} e^{-\frac{(1-e^{-\beta T}+b)}{\beta e^{-\beta\tau}} \lambda_{tv}} \frac{\beta^{n+c-1} e^{-\beta(\sum_{i=1}^{n} t_i + d)}}{(1-e^{-\beta T}+b)^{n+a}} d\beta$$

$$= 1 - \frac{1}{p} \sum_{h=0}^{n+a-1} \int_0^\infty \frac{\left(\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}} \lambda_{tv}\right)^h}{h!} e^{-\frac{(1-e^{-\beta T}+b)}{\beta e^{-\beta\tau}} \lambda_{tv}} \frac{\beta^{n+c-1} e^{-\beta(\sum_{i=1}^{n} t_i + d)}}{(1-e^{-\beta T}+b)^{n+a}} d\beta \tag{61}$$

Equation (61) is the same as the equation in the second formula of Prop. A1.1

4.4.3 Proposition B1.1

For given level γ, the time τ' required to attain λ_{tv} is

$$\tau' = \begin{cases} \frac{1}{\beta} \ln\left[\frac{\beta\chi^2(2n;\gamma)}{2\lambda_{tv}(1-e^{-\beta T}+b)}\right] - T & \text{if } \beta \text{ is known} \\ \tau - T & \text{if } \beta \text{ is unknown} \end{cases} \qquad \text{Prop. B1.1}$$

Remark 3: The second formula of Prop. B1.1 is such that τ is the solution to the equation

$$\gamma = 1 - \frac{1}{p}\sum_{h=0}^{n-1}\int_0^{\infty} \frac{\left(\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}}\lambda_{tv}\right)^h}{h!} e^{-\frac{(1-e^{-\beta T}+b)}{\beta e^{-\beta\tau}}\lambda_{tv}} \frac{\beta^{n+c-1}e^{-\beta(\sum_{i=1}^{n} t_i + d)}}{(1-e^{-\beta T}+b)^{n+a}}\, d\beta .$$

Proof

For given level γ, the time required to attain the target value λ_{tv} is $\tau^{'} = \tau - T$ where τ satisfies Equation (28). When β is known, from Equation (57), it can easily be seen that $2\left(\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}}\right)\lambda_{\tau}$ follows a chi-square distribution with $2n$ degrees of freedom. Therefore, we have

$$2\left(\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}}\right)\lambda_{tv} = \chi^2(2n;\gamma) \tag{62}$$

$$\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}} = \frac{\chi^2(2n;\gamma)}{2\lambda_{tv}} \tag{63}$$

Making τ the subject from Equation (63) we obtain

$$\tau = \frac{1}{\beta}\ln\left[\frac{\beta\chi^2(2n;\gamma)}{2\lambda_{tv}(1-e^{-\beta T}+b)}\right] \tag{64}$$

Therefore, from Equation (64), we obtain the time required to attain the target value λ_{tv} given level γ as

$$\tau^{'} = \frac{1}{\beta}\ln\left[\frac{\beta\chi^2(2n;\gamma)}{2\lambda_{tv}(1-e^{-\beta T}+b)}\right] - T \tag{65}$$

Equation (65) is as the equation in the first formula of Prop. B1.1

When β is unknown, the time required to attain the target value λ_{tv} with level γ is $\tau^{'} = \tau - T$ where τ is the solution to

$$\gamma = 1 - \frac{1}{p}\sum_{h=0}^{n-1}\int_0^{\infty}\frac{\left(\dfrac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}}\lambda_{tv}\right)^h}{h!}e^{-\frac{(1-e^{-\beta T}+b)}{\beta e^{-\beta\tau}}\lambda_{tv}}\frac{\beta^{n+c-1}e^{-\beta(\sum_{i=1}^{n}t_i+d)}}{(1-e^{-\beta T}+b)^{n+a}}d\beta \qquad (66)$$

4.4.4 Proposition C1.1

The Bayesian UPL of $\lambda_\tau = \alpha\beta e^{-\beta\tau}$ with level γ is

$$\lambda_U^{(\beta)}(\tau) = \begin{cases} \dfrac{\chi^2(2n;\gamma)\beta e^{-\beta\tau}}{2(1-e^{-\beta T}+b)} & \text{if } \beta \text{ is known} \\ \lambda_{tv}^{'} & \text{if } \beta \text{ is unknown} \end{cases} \qquad \text{Prop. C1.1}$$

Remark 4: The second formula in Prop. C1.1 is such that $\lambda_{tv}^{'}$ is the solution to the equation

$$\gamma = 1 - \frac{1}{p}\sum_{h=0}^{n+a-1}\int_0^{\infty}\frac{\left(\dfrac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}}\lambda_{tv}^{'}\right)^h}{h!}e^{-\frac{(1-e^{-\beta T}+b)}{\beta e^{-\beta\tau}}\lambda_{tv}^{'}}\frac{\beta^{n+c-1}e^{-\beta(\sum_{i=1}^{n}t_i+d)}}{(1-e^{-\beta T}+b)^{n+a}}d\beta .$$

Proof

For a pre-determined τ $(\tau > T)$, the Bayesian Upper Prediction Limit (UPL) for λ_τ with level γ is $\lambda_U^{(\beta)}(\tau)$ satisfying $\gamma = \Pr(\lambda_\tau \le \lambda_U^{(\beta)}(\tau) \mid Y_{obs})$. From Equation (28) and Equation (62) we have

$$\gamma = \int_0^{\lambda_\tau^{(\beta)}(\tau)}\frac{\left(\dfrac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}}\right)^n}{\Gamma(n+a)}\lambda_\tau^{n-1}e^{-\lambda_\tau\frac{(1-e^{-\beta T}+b)}{\beta e^{-\beta\tau}}}d\lambda_\tau .$$

This implies that

$$2\left(\frac{1-e^{-\beta T}+b}{\beta e^{-\beta\tau}}\right)\lambda_U^{(\beta)}(\tau) = \chi^2(2n;\gamma) \qquad (67)$$

Making $\lambda_U^{(\beta)}(\tau)$ the subject from Equation (67) we arrive at

$$\lambda_U^{(\beta)}(\tau) = \frac{\chi^2(2n;\gamma)\beta e^{-\beta\tau}}{2(1-e^{-\beta T}+b)}. \qquad (68)$$

Therefore, for a pre-determined τ $(\tau > T)$, the Bayesian UPL for λ_τ with level γ is $\lambda_U^{(\beta)}(\tau)$ that satisfies Equation (68).

When β is unknown, the Bayesian UPL for $\lambda_\tau = \alpha\beta e^{-\beta\tau}$ with level γ is $\lambda_{tv}^{'}$ where $\lambda_{tv}^{'}$ is a solution to

$$\gamma = 1 - \frac{1}{p} \sum_{h=0}^{n+a-1} \int_0^{\infty} \frac{\left(\frac{1 - e^{-\beta T} + b}{\beta e^{-\beta\tau}} \lambda_{tv}^{'} \right)^h}{h!} e^{-\frac{(1-e^{-\beta T}+b)}{\beta e^{-\beta\tau}}\lambda_{tv}^{'}} \frac{\beta^{n+c-1} e^{-\beta(\sum_{i=1}^{n} t_i + d)}}{(1 - e^{-\beta T} + b)^{n+a}} d\beta. \tag{69}$$

4.4.5 Proposition D1.1

The probability that at most k failures will occur in the time interval $(T, \tau]$ with $\tau > T$ is

$$\gamma_k = \begin{cases} \left(\frac{1 - e^{-\beta T} + b}{e^{-\beta T} - e^{-\beta\tau}} \right)^n \left(\frac{1 - e^{-\beta T} + b}{1 - e^{-\beta\tau} + b} \right)^a \sum_{j=n}^{n+k} \binom{j + a - 1}{n + a - 1} \left(\frac{e^{-\beta T} - e^{-\beta\tau}}{1 - e^{-\beta\tau} + b} \right)^j & \text{if } \beta \text{ is known} \\ \sum_{j=n}^{n+k} \frac{\Gamma(j + a)}{p\,(j - n)!\,\Gamma(n + a)} \int_0^{\infty} \beta^{n+c-1} e^{-\beta(\sum_{i=1}^{n} t_i + d)} \frac{\left(e^{-\beta T} - e^{-\beta\tau}\right)^{j-n}}{\left(1 - e^{-\beta\tau} + b\right)^{j+a}} d\beta & \text{if } \beta \text{ is unknown} \end{cases} \qquad \text{Prop. D1.1}$$

Proof

The probability that at most k failures will occur in the interval

(T, τ) is $\gamma_k = \Pr\left[N(\tau) \le n + k \mid Y_{obs} \right]$. When β is known, we have

$$\gamma_k = \int_0^{\infty} \Pr\left[N(\tau) \le n + k \mid Y_{obs}, \alpha \right] h(\alpha \mid Y_{obs}) d\alpha \tag{70}$$

where $h(\alpha \mid Y_{obs})$ is given by Equation (53) and

$$\Pr[N(\tau) \le n + k \mid Y_{obs}, \alpha] = \sum_{j=n}^{n+k} f(Y_{obs}, N(\tau) = j \mid \alpha) \,/\, f(Y_{obs} \mid \alpha) \tag{71}$$

From (14) we have $f(Y_{obs} \mid \alpha) = \alpha^n \beta^n e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha(1-e^{-\beta T})}$ and

$$f(Y_{obs}, N(\tau) = j \mid \alpha) = \int\limits_{D(j-n:T,\tau)} f(Y_{obs}, x_{n+1}, \ldots, x_j, N(\tau) = j) \prod_{\ell=n+1}^{j} dx_\ell$$

$$f(Y_{obs}, N(\tau) = j \mid \alpha) = \int\limits_{D(j-n:T,\tau)} \alpha^j \beta^j e^{-\beta \sum_{i=1}^{j} t_i} e^{-\alpha(1-e^{\beta\tau})} \prod_{\ell=n+1}^{j} dt_\ell$$

$$= \alpha^j \beta^j e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha(1-e^{-\beta\tau})} \int\limits_{D(j-n;T,\tau)} e^{-\beta \sum_{\ell=n+1}^{j} t_\ell} \prod_{\ell=n+1}^{j} dt_\ell . \qquad (72)$$

Solving the integral part of Equation (72), we proceed as follows:

$\int_0^t e^{-\beta t} dt = \frac{1}{\beta}(1 - e^{-\beta t})$. Substituting the limits T and τ we obtain $\frac{1}{\beta}(1 - e^{-\beta\tau}) - \frac{1}{\beta}(1 - e^{-\beta T})$ which reduces to $\frac{1}{\beta}(e^{-\beta T} - e^{-\beta\tau})$. Hence the integral part of Equation (72) is obtained as

$$\int\limits_{D(j-n;T,\tau)} e^{-\beta \sum_{\ell=n+1}^{j} t_\ell} \prod_{\ell=n+1}^{j} dt_\ell = \frac{1}{\beta^{j-n}} \frac{(e^{-\beta T} - e^{-\beta\tau})^{j-n}}{(j-n)!} \qquad (73)$$

Substituting Equation (73) into Equation (72) we obtain

$$f(Y_{obs}, N(\tau) = j \mid \alpha) = \alpha^j \beta^j e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha(1-e^{-\beta\tau})} \frac{1}{\beta^{j-n}} \frac{(e^{-\beta T} - e^{-\beta\tau})^{j-n}}{(j-n)!}$$

From Equation (71) we obtain

$$f(Y_{obs}, N(\tau) = j \mid \alpha) / f(Y_{obs} \mid \alpha) = \frac{\alpha^j \beta^n e^{-\beta \sum_{i=1}^{n} t_i} \dfrac{(e^{-\beta T} - e^{-\beta\tau})^{j-n}}{(j-n)!}}{\alpha^n \beta^n e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha(1-e^{-\beta T})}}$$

which reduces to

$$f(Y_{obs}, N(\tau) = j \mid \alpha) \,/\, f(Y_{obs} \mid \alpha) = \alpha^{j-n} e^{-\alpha(e^{-\beta T} - e^{-\beta\tau})} \frac{(e^{-\beta T} - e^{-\beta\tau})^{j-n}}{(j-n)!}.$$

Therefore, Equation (71) becomes

$$\Pr[N(\tau) \le n + k \mid Y_{obs}, \alpha] = \sum_{j=n}^{n+k} \alpha^{j-n} e^{-\alpha(e^{-\beta T} - e^{-\beta\tau})} \frac{(e^{-\beta T} - e^{-\beta\tau})^{j-n}}{(j-n)!} \tag{74}$$

and Equation (70) becomes

$$\gamma_k = \int_0^\infty \sum_{j=n}^{n+k} \alpha^{j-n} e^{-\alpha(e^{-\beta T} - e^{-\beta\tau})} \frac{(e^{-\beta T} - e^{-\beta\tau})^{j-n}}{(j-n)!\,\Gamma(n+a)} \alpha^{n+a-1} e^{-\alpha(1-e^{-\beta T}+b)} (1 - e^{-\beta T} + b)^{n+a} d\alpha$$

$$= \sum_{j=n}^{n+k} \frac{(e^{-\beta T} - e^{-\beta\tau})^{j-n} (1 - e^{-\beta T} + b)^{n+a}}{(j-n)!\,\Gamma(n+a)} \int_0^\infty \alpha^{j+a-1} e^{-\alpha(1-e^{-\beta\tau}+b)} d\alpha$$

$$= \sum_{j=n}^{n+k} \frac{\left(e^{-\beta T} - e^{-\beta\tau}\right)^{j-n} \left(1 - e^{-\beta T} + b\right)^{n+a}}{\left(j-n\right)!\,\Gamma(n+a)} \frac{\Gamma(j+a)}{\left(1 - e^{-\beta\tau} + b\right)^{j+a}} \int_0^\infty \frac{\left(1 - e^{-\beta\tau} + b\right)^{j+a}}{\Gamma(j+a)} \alpha^{j+a-1} e^{-\alpha\left(1-e^{-\beta\tau}+b\right)} d\alpha$$

(75)

$$= \sum_{j=n}^{n+k} \frac{\left(e^{-\beta T} - e^{-\beta\tau}\right)^{j-n} \left(1 - e^{-\beta T} + b\right)^{n+a}}{\left(j-n\right)!\,\Gamma(n+a)} \frac{\Gamma(j+a)}{\left(1 - e^{-\beta\tau} + b\right)^{j+a}}. \tag{76}$$

(Since the integral part of Equation (75) integrates to 1 from the fact that it is a gamma distribution with parameters j and $1 - e^{-\beta\tau}$)

After re-arranging Equation (76) we obtain

$$\gamma_k = \left(\frac{1 - e^{-\beta T} + b}{e^{-\beta T} - e^{-\beta\tau}}\right)^n \left(\frac{1 - e^{-\beta T} + b}{1 - e^{-\beta\tau} + b}\right)^a \sum_{j=n}^{n+k} \binom{j+a-1}{n+a-1} \left(\frac{e^{-\beta T} - e^{-\beta\tau}}{1 - e^{-\beta\tau} + b}\right)^j. \tag{77}$$

Equation (77) implies the first formula of Prop. D1.1

When β is unknown, noting that $\Pr\left[N(\tau) \le n + k \mid Y_{obs}, \alpha, \beta\right]$ and $h(\alpha, \beta \mid Y_{obs})$ are given by Equation (74) and Equation (56) respectively, we have

$$\gamma_k = \int_0^\infty \int_0^\infty \Pr\left[N(\tau) \le n+k \mid Y_{obs}, \alpha, \beta\right] h(\alpha, \beta \mid Y_{obs}) d\alpha\, d\beta$$

$$= \sum_{j=n}^{n+k} \frac{1}{p\left(j-n\right)!\Gamma(n+a)} \int_0^\infty \int_0^\infty \alpha^{j-n} e^{-\alpha\left(e^{-\beta T} - e^{-\beta\tau}\right)} \left(e^{-\beta T} - e^{-\beta\tau}\right)^{j-n} \alpha^{n+a-1} \beta^{n+c-1} e^{-\beta\left(\sum_{i=1}^{n} t_i + d\right)} e^{-\alpha\left(1-e^{-\beta T}+b\right)} d\alpha\, d\beta$$

$$= \sum_{j=n}^{n+k} \frac{\Gamma(j+a)}{p\left(j-n\right)!\Gamma(n+a)} \int_0^\infty \beta^{n+c-1} e^{-\beta\left(\sum_{i=1}^{n} t_i + d\right)} \frac{\left(e^{-\beta T} - e^{-\beta\tau}\right)^{j-n}}{\left(1 - e^{-\beta\tau} + b\right)^{j+a}} d\beta \qquad (78)$$

Equation (78) implies the second formula of Prop. D1.1

4.5 Main results for the two-sample prediction

4.5.1 Proposition A2

The Bayesian UPL of y_r (i.e. the rth failure time of the second software system) with level γ when β is known is

$$\gamma = \Gamma(r+n)\left[\Gamma(r)\Gamma(n)\right]^{-1} \beta\left(1-e^{-\beta T}\right)^n \int_0^{y_U^{(\beta)}} \frac{e^{-\beta y_r}\left(1-e^{-\beta y_r}\right)^{r-1}}{\left(2-e^{-\beta y_r}-e^{-\beta T}\right)^{r+n}} dy_r \qquad \text{Prop A2}$$

Proof

We know that given $N(t) = n$, the n failure times $x_1, x_2, ..., x_n$ have the same distribution as the order statistics corresponding to n independent random variables with

density $f(x) = \alpha\beta e^{-\beta x} \Big/ \int_0^t \alpha\beta e^{-\beta u} du$, $0 \le x \le t$ which reduces to $f(x) = \dfrac{\beta e^{-\beta x}}{\left(1-e^{-\beta t}\right)}$. This is to say

that

$$(x_1, ..., x_n \mid N(t) = n) \sim n! \prod_{i=1}^{n} f(x_i) = n! \prod_{i=1}^{n} \left[\frac{\beta e^{-\beta x_i}}{1-e^{-\beta t}}\right] = n! \frac{\beta^n e^{-\beta \sum_{i=1}^{n} x_i}}{\prod_{i=1}^{n}\left(1-e^{-\beta t}\right)}.$$

Consequently,

$$\left(x_1, x_2, \ldots, x_n \mid x_n\right) \sim (n-1)!\,\frac{\beta^{n-1} e^{-\beta \sum_{i=1}^{n-1} x_i}}{\prod_{i=1}^{n-1} (1-e^{-\beta x_n})} \tag{79}$$

The joint density of $\left(x_1, \ldots, x_n\right)$ is also given by Equation (14). Dividing Equation (14) by Equation (79), the density of x_n is obtained as

$$f(x_n \mid \alpha, \beta) = \frac{\alpha^n \beta^n e^{-\beta \sum_{i=1}^{n} x_i} e^{-\alpha(1-e^{-\beta x_n})}}{(n-1)!\,\beta^{n-1} e^{-\beta \sum_{i=1}^{n-1} x_i} \Big/ \prod_{i=1}^{n-1} (1-e^{-\beta x_n})}$$

$$= \frac{\alpha^n \beta^n e^{-\beta \sum_{i=1}^{n} x_i} e^{-\alpha(1-e^{-\beta x_n})} \left(1-e^{-\beta x_n}\right)^{n-1}}{(n-1)!\,\beta^{n-1} e^{-\beta \sum_{i=1}^{n-1} x_i}}$$

$$= \left[(n-1)!\right]^{-1} \beta e^{-\beta x_n} e^{-\alpha(1-e^{-\beta x_n})} \left(1-e^{-\beta x_n}\right)^{n-1}. \tag{80}$$

Replacing x_n by y_r, for the second system, we have the density of y_r given as

$$f(y_r \mid \alpha, \beta) = \left[\Gamma(r)\right]^{-1} \beta \alpha^r e^{-\beta y_r} e^{-\alpha(1-e^{-\beta y_r})} \left(1-e^{-\beta y_r}\right)^{r-1}. \tag{81}$$

From Equation (20) we have

$$f(y_r \mid Y_{obs}) = \int_0^\infty \left[\Gamma(r)\Gamma(n)\right]^{-1} \beta \alpha^r e^{-\beta y_r} e^{-\alpha(1-e^{-\beta y_r})} \left(1-e^{-\beta y_r}\right)^{r-1} \alpha^{n-1} e^{-\alpha(1-e^{-\beta T})} (1-e^{-\beta T})^n \, d\alpha$$

$$= \left[\Gamma(r)\Gamma(n)\right]^{-1} \left(1-e^{-\beta y_r}\right)^{r-1} \left(1-e^{-\beta T}\right)^n e^{-\beta y_r} \beta \int_0^\infty \alpha^{r+n-1} e^{-\alpha(2-e^{-\beta y_r}-e^{-\beta T})} d\alpha$$

$$= \left[\Gamma(r)\Gamma(n)\right]^{-1}\left(1-e^{-\beta y_r}\right)^{r-1}\left(1-e^{-\beta T}\right)^{n} e^{-\beta y_r}\beta \frac{\Gamma(r+n)}{(2-e^{-\beta y_r}-e^{-\beta T})^{r+n}}\int_0^{\infty}\frac{(2-e^{-\beta y_r}-e^{-\beta T})^{r+n}}{\Gamma(r+n)}\alpha^{r+n-1}e^{-\alpha(2-e^{-\beta y_r}-e^{-\beta T})}d\alpha$$

$$= \left[\Gamma(r)\Gamma(n)\right]^{-1}\left(1-e^{-\beta y_r}\right)^{r-1}\left(1-e^{-\beta T}\right)^{n} e^{-\beta y_r}\beta \frac{\Gamma(r+n)}{(2-e^{-\beta y_r}-e^{-\beta T})^{r+n}}. \qquad (82)$$

From Equation (21) and Equation (82), we have

$$\gamma = \Gamma(r+n)\left[\Gamma(r)\Gamma(n)\right]^{-1}\beta\left(1-e^{-\beta T}\right)^{n}\int_0^{y_U^{(\beta)}}\frac{e^{-\beta y_r}\left(1-e^{-\beta y_r}\right)^{r-1}}{(2-e^{-\beta y_r}-e^{-\beta T})^{r+n}}dy_r \qquad (83)$$

Equation (83) implies the formula in Prop. A2

4.5.2 Proposition B2

The probability that the number of failures $N(t_2)$ in the time interval $(0,t_2]$ for the second software system does not exceed a pre-determined nonnegative integer m, when β is known is

$$\gamma = \sum_{k=0}^{m}\frac{\left(1-e^{-\beta t_2}\right)^{k}}{(2-e^{-\beta t_2}-e^{-\beta T})^{k}}\binom{n+k-1}{k}\frac{(1-e^{-\beta T})^{n}}{(2-e^{-\beta t_2}-e^{-\beta T})^{n}}. \qquad \text{Prop. B2}$$

Proof

The study is interested in predicting the number of failures (denoted by $N(t_2)$) of the second system occurring in the time interval $(0,t_2]$. It can be shown from Equation (4) that the distribution in the interval $(t_1,t_2]$ has a mean value of $\int_{t_1}^{t_2}\alpha\beta e^{-\beta t}dt$. After solving the integral, the mean value reduces to $m(t)=\alpha(e^{-\beta t_1}-e^{-\beta t_2})$. This implies that the mean value of the distribution in the interval $(0,t_2]$ is $m(t)=E\left[N(t)\right]=\alpha\left(1-e^{-\beta t_2}\right)$. Hence

$$\Pr\left[N(t_2)=k\mid\alpha,\beta\right]=\left[k!\right]^{-1}\alpha^{k}(1-e^{-\beta t_2})^{k}e^{-\alpha(1-e^{-\beta t_2})}, \quad k=0,1,\ldots \qquad (84)$$

For any level γ, the Bayesian Upper prediction limit for $N(t_2)$ is Nu satisfying

$$\gamma = \Pr\left[N(t_2)\le Nu \mid Y_{obs}\right]$$

Here, an equivalent problem is considered. For any given positive integer m, we want to compute the probability that $N(t_2) \le m$ i.e.

$$\gamma = \Pr\left[N(t_2) \le m \mid Y_{obs}\right] = \sum_{k=0}^{m} \Pr\left[N(t_2) = k \mid Y_{obs}\right] \tag{85}$$

When β is known, from Equation (84) and Equation (19) we have

$$\Pr\left[N(t_2) = k \mid Y_{obs}\right] = \int_0^\infty \Pr\left[N(t_2) = k \mid \alpha\right] h(\alpha \mid Y_{obs}) d\alpha$$

$$= \int_0^\infty \frac{\alpha^k (1-e^{-\beta t_2})^k e^{-\alpha(1-e^{-\beta t_2})} \alpha^{n-1} e^{-\alpha(1-e^{-\beta T})} \left(1-e^{-\beta T}\right)^n}{k!\Gamma(n)} d\alpha$$

$$= \frac{\left(1-e^{-\beta t_2}\right)^k (1-e^{-\beta T})^n}{k!\Gamma(n)} \int_0^\infty \alpha^{k+n-1} e^{-\alpha(2-e^{-\beta t_2}-e^{-\beta T})} d\alpha$$

$$= \frac{\left(1-e^{-\beta t_2}\right)^k (1-e^{-\beta T})^n}{k!\Gamma(n)} \frac{\Gamma(k+n)}{(2-e^{-\beta t_2}-e^{-\beta T})^{k+n}}$$

$$= \frac{\left(1-e^{-\beta t_2}\right)^k}{(2-e^{-\beta t_2}-e^{-\beta T})^k} \frac{\Gamma(k+n)}{k!\Gamma(n)} \frac{(1-e^{-\beta T})^n}{(2-e^{-\beta t_2}-e^{-\beta T})^n}. \tag{86}$$

Therefore, Equation (86) becomes

$$\gamma = \sum_{k=0}^{m} \frac{\left(1-e^{-\beta t_2}\right)^k}{(2-e^{-\beta t_2}-e^{-\beta T})^k} \binom{n+k-1}{k} \frac{(1-e^{-\beta T})^n}{(2-e^{-\beta t_2}-e^{-\beta T})^n}. \tag{87}$$

Equation (87) implies the formula in Prop. B2

4.5.3 Proposition C2

Given that the number of failures in $(0, t_2]$ for the second software is m, the Bayesian UPL of y_r, $(1 \le r \le m)$ with level γ is $y_U^{(\beta)}$ satisfying the equation

$$\gamma = \frac{(1-e^{-\beta y_U^{(\beta)}})^m}{(1-e^{-\beta t_2})^m} \qquad \text{Prop. C2}$$

Proof

First, we want to find the conditional density of y_r given $N(t_2)=m$, from Equation (14),

$$f(y_1,\ldots,y_m;N(t_2)=m)=\alpha^m\beta^m e^{-\beta\sum_{i=1}^{m}y_i}e^{-\alpha(1-e^{-\beta t_2})} \tag{88}$$

Expanding the exponential term with the observed variables we have,

$$e^{-\beta\sum_{i=1}^{m}y_i}=e^{-\beta y_1}e^{-\beta y_2}\ldots e^{-\beta y_{r-1}}e^{-\beta y_r}e^{-\beta y_{r+1}}\ldots e^{-\beta y_m}. \tag{89}$$

Integrating with respect to $(y_1,\ldots,y_{r-1})$, we proceed as follows

$$\int_0^{y_r}\int_0^{y_r}\cdots\int_0^{y_r}e^{-\beta y_1}e^{-\beta y_2}\ldots e^{-\beta y_{r-1}}e^{-\beta y_r}e^{-\beta y_{r+1}}\ldots e^{-\beta y_m}dy_1dy_2\ldots dy_{r-1}=\frac{1}{(r-1)!\beta^{r-1}}(1-e^{-\beta y_r})^{r-1}e^{-\beta y_r}e^{-\beta\sum_{i=r+1}^{m}y_i}$$

This implies that after integrating Equation (88) with respect to $y_1,\ldots,y_{r-1}$ we have

$$f\left(y_r,\ldots,y_m;N(t_2)=m\right)=\frac{1}{(r-1)!}\alpha^m\beta^{m-r+1}(1-e^{-\beta y_r})^{r-1}e^{-\beta y_r}e^{-\beta\sum_{i=r+1}^{m}y_i}e^{-\alpha(1-e^{-\beta t_2})}. \tag{90}$$

Further integrating Equation (90) with respect to $(y_{r+1},\ldots,y_m)$, we proceed as follows:

Expanding the exponent with the observed variables, we have $e^{-\beta\sum_{i=r+1}^{m}y_i}=e^{-\beta y_{r+1}}e^{-\beta y_{r+2}}\ldots e^{-\beta y_m}$. Now integrating, we have

$$\int_{y_r}^{t_2}\int_{y_r}^{t_2}\cdots\int_{y_r}^{t_2}e^{-\beta y_{r+1}}e^{-\beta y_{r+2}}\ldots e^{-\beta y_m}dy_{r+1}dy_{r+2}\ldots dy_m=\frac{1}{(m-1)!\beta^{m-r}}(e^{-\beta y_r}-e^{-\beta t_2})^{m-r}$$. Therefore, after integrating with respect to $(y_{r+1},\ldots,y_m)$, Equation (90) becomes

$$f(y_r; N(t_2)=m) = \left[\beta^{m-r}(m-1)!(r-1)!\right]^{-1} \alpha^m \beta^{m-r+1}(1-e^{-\beta y_r})^{r-1} e^{-\beta y_r}(e^{-\beta y_r} - e^{-\beta t_2})^{m-r} e^{-\alpha(1-e^{-\beta t_2})}$$

$$= \left[(m-r)!(r-1)!\right]^{-1} \alpha^m \beta(1-e^{-\beta y_r})^{r-1} e^{-\beta y_r}(e^{-\beta y_r} - e^{-\beta t_2})^{m-r} e^{-\alpha(1-e^{-\beta t_2})}. \qquad (91)$$

Therefore, the conditional density of y_r given $N(t_2)=m$ is

$$f(y_r \mid N(t_2)=m) = \frac{f(y_r; N(t_2)=m)}{\Pr\left\{N(t_2)=m\right\}}$$

$$= \frac{\left[(m-r)!(r-1)!\right]^{-1} \alpha^m \beta(1-e^{-\beta y_r})^{r-1} e^{-\beta y_r}(e^{-\beta y_r} - e^{-\beta t_2})^{m-r} e^{-\alpha(1-e^{-\beta t_2})}}{\alpha^m \frac{(1-e^{-\beta t_2})^m}{m!} e^{-\alpha(1-e^{-\beta t_2})}}$$

$$= \frac{m!}{(m-r)!(r-1)!(1-e^{-\beta t_2})^m} \beta(1-e^{-\beta y_r})^{r-1} e^{-\beta y_r}(e^{-\beta y_r} - e^{-\beta t_2})^{m-r} \qquad (92)$$

which is independent of α. When β is known, Equation (20) can be re-written as

$$f(y_r \mid N(t_2)=m, Y_{obs}) = \int_0^\infty f(y_r \mid N(t_2)=m, \alpha) h(\alpha \mid N(t_2)=m, Y_{obs}) d\alpha$$

where $f(y_r \mid N(t_2)=m,\alpha) = f(y_r \mid N(t_2)=m)$ is given by Equation (92) and

$h(\alpha \mid N(t_2)=m, Y_{obs}) = h(\alpha \mid Y_{obs})$ is given by Equation (19). Hence

$$f(y_r \mid N(t_2)=m, Y_{obs}) = f(y_r \mid N(t_2)=m).$$

Given $N(t_2)=m$, the Bayesian UPL of y_r with level γ is $y_U^{(\beta)}$ such that

$$\gamma = \int_0^{y_U^{(\beta)}} f(y_r \mid N(t_2)=m, Y_{obs}) dy_r$$

$$= \frac{m!}{(m-r)!(r-1)!(1-e^{-\beta t_2})^m} \int_0^{y_U^{(\beta)}} \beta(1-e^{-\beta y_r})^{r-1} e^{-\beta y_r}(e^{-\beta y_r} - e^{-\beta t_2})^{m-r} dy_r. \qquad (93)$$

If $r = m$, Equation (93) becomes

$$\gamma = \frac{m!\beta}{(m-1)!(1-e^{-\beta t_2})^m} \int_0^{y_U^{(\beta)}} (1-e^{-\beta y_m})^{m-1} e^{-\beta y_m} dy_m . \tag{94}$$

Solving the integral part of Equation (94), we proceed as follows

Let $x = e^{-\beta y_m}$,$\Rightarrow \ln x = -\beta y_m$ $\Rightarrow y_m = -\frac{1}{\beta}\ln x$ and $dy_m = -\frac{1}{\beta}.\frac{1}{x}dx$. This implies that

$$\gamma = \frac{m!\beta}{(m-1)!(1-e^{-\beta t_2})^m} \int_1^{e^{-\beta y_U^{(\beta)}}} x(1-x)^{m-1}.-\frac{1}{\beta}.\frac{1}{x}dx$$

$$= \frac{m!}{(m-1)!(1-e^{-\beta t_2})^m}.-\int_1^{e^{-\beta y_U^{(\beta)}}} (1-x)^{m-1}dx.$$

Again, letting $u = 1-x$, $dx = -du$

$$\gamma = \frac{m!}{(m-1)!(1-e^{-\beta t_2})^m} \int_0^{1-e^{-\beta y_U^{(\beta)}}} u^{m-1}du$$

$$= \frac{m!}{(m-1)!(1-e^{-\beta t_2})^m} \left[\frac{1}{m}u^m \Big|_0^{1-e^{-\beta y_U^{(\beta)}}} \right]$$

$$= \frac{m!}{m(m-1)!(1-e^{-\beta t_2})^m} \left(1-e^{-\beta y_U^{(\beta)}}\right)^m$$

$$= \frac{(1-e^{-\beta y_U^{(\beta)}})^m}{(1-e^{-\beta t_2})^m} \tag{95}$$

Thus, the Bayesian UPL of y_m with confidence level γ is $y_U^{(\beta)}$ that satisfies Equation (95). The UPL with level γ is equal to the lower prediction limit with level $1-\gamma$

4.6 Real Examples for Single sample Bayesian Prediction

4.6.1 Using non-informative prior

In this subsection, real data on the time between failures in Table 1 is used to illustrate the developed methodologies for the one-sample Bayesian predictive analysis.

Table 1: Time between Failures Data (Xie et al. 2002).

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

The study used the cumulative time between failures which are the failure times $0 < t_1 < t_2 < ... < t_{30}$ where $n = 30$. Satya *et al.* (2011) argued that these data obey the Goel-Okumoto software reliability model and they obtained the MLEs of the parameters as $\alpha = 31.698171$ and $\beta = 0.003962$. In the illustration of the developed methodologies, the study has used the MLE of β as was obtained by Satya *et al.* (2011) for the case where β is assumed known.

(A1) 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 λ_{tv}. Implying that the target value cannot be achieved at time $T = 182.21$. Thus the development testing will continue. Suppose we want to find the probability that the target value λ_{tv} will be achieved at time $\tau = 277.83$, (i) when β is known, say $\beta = 0.003962$, from the first formula in (Prop. A1) we obtain $\gamma = 0$. Thus it is not likely that the target value (failure rate) will be achieved. (ii) When β is unknown, from the second formula in (Prop. A1), we obtain $\gamma = 0.0576$ based on Monte Carlo sample size of $L = 1000$.

(B1) Since the target value λ_{tv} was not achieved at time $T = 182.21$, we want to know how long it will require in order to attain λ_{tv}. (i) When β is known (i.e. $\beta = 0.003962$), Let $\gamma = 0.90$, from the first formula in (Prop. B1) we obtain $\tau^* = 268.6116h$. In other words, it will take another 268.6116h in order to achieve the desired failure rate. (ii) When β is unknown, from the second formula of (Prop. B1) and Remark 1, we obtain $\tau^* =$ 817.74h. In other words, it will take another 817.74h in order to achieve the desired failure rate when β is unknown.

(C1) Given $\tau = 900h$, (i) When β is known, from the first formula in (Prop. C1), the Bayesian Upper Prediction Limit of $\lambda_\tau = \alpha\beta e^{-\beta\tau}$ with level 0.90 is given by $\lambda_u^{(\beta)}(\tau) = 0.0051$ (ii) When β is unknown, from the second part of (Prop. C1) and Remark 2, the Bayesian UPL of $\lambda_\tau = \alpha\beta e^{-\beta\tau}$ with level 0.90 is given by $\lambda_u^{(\beta)}(\tau) = 0.131952$.

(D1) Suppose that we are interested in the probability γ_k that at most k failures will occur in the future time period $(\tau, T] = (180, 250]$ (i) When β is known (say $\beta = 0.003969$), using the first formula in (Prop. D1), 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 $\gamma_{15} = 0.9980$

(ii) When β is unknown, from the second formula in (Prop. D1), 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 β is known and when β is unknown.

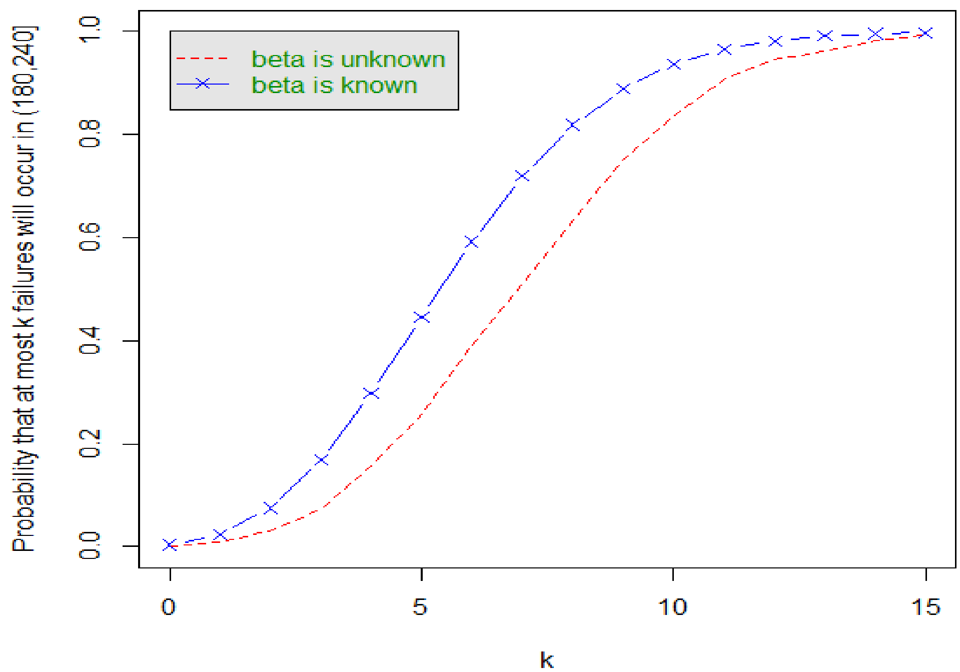

Figure 1: The graph of the probabilities γ_k that at most k failures will occur in the time interval $(180, 240]$ for the cases of known and unknown β .

4.6.2 Using informative prior

In this section, the parameters of the informative priors $Gamma(a,b)$ and $Gamma(c,d)$ are chosen arbitrarily as $a = 2,\ b = 1/2,\ c = 2,$ and $d = 1/2$.

(A1.1) The discussion in the first part of A1 follows. (i) When β is known, say $\beta = 0.003962$, from the first formula in (Prop. A1.1) we obtain $\gamma = 0.04542$. Thus, as was in the case of non-informative prior, it is unlikely that the target value (failure rate) will be achieved. (ii) When β is unknown, from the second formula in (Prop. A1.1), we obtain $\gamma = 0.0269$ where the Monte Carlo sample size is $L = 1000$.

(B1.1) The target value $\lambda_{tv} = 0.03$ has not been achieved at time $T = 182.21$. The interest now is to know how long it will require in order to attain λ_{tv}. (i) When β is known (i.e. $\beta = 0.003962$), Let $\gamma = 0.90$, from the first formula in (Prop. B1.1) we obtain $\tau^* = 97.167h$. In other words, it will take another 97.167h in order to achieve the desired failure rate. This is a significant reduction in the time that it will take to achieve the required λ_{tv} compared to the case when non-informative prior was used. (ii) When β is unknown, from the second formula of (Prop. B1.1) and Remark 3, we obtain $\tau' = 323.79h$. In other words, it will take another 323.79 hours in order to achieve the desired failure rate when β is unknown.

(C1.1) Given $\tau = 900h$, (i) When β is known, from the first formula in (Prop. C1.1), the Bayesian Upper Prediction Limit of $\lambda_\tau = \alpha\beta e^{-\beta\tau}$ with level 0.90 is given by $\lambda_u^{(\beta)}(\tau) = 0.0026$ (ii) When β is unknown, from the second formula of (Prop. C1.1) and Remark 4, the Bayesian UPL of $\lambda_\tau = \alpha\beta e^{-\beta\tau}$ with level 0.90 is given by $\lambda_u^{(\beta)}(\tau) = 0.1339$.

(D1.1) Suppose that we are interested in the probability γ_k that at most k failures will occur in the future time period $(\tau, T] = (180, 250]$ considering only the case when β is known (say $\beta = 0.003969$), using the first formula in (Prop. D1.1), we have

$\gamma_0 = 0.0438,\ \gamma_1 = 0.1744,\ \gamma_2 = 0.3750,\ \gamma_3 = 0.5867,\ \gamma_4 = 0.7592,\ \gamma_5 = 0.8748,\ \gamma_6 = 0.9412,$

$\gamma_7 = 0.9748,\ \gamma_8 = 0.9900,\ \gamma_9 = 0.9963,\ \gamma_{10} = 0.9987,\ \gamma_{11} = 0.9996,\ \gamma_{12} = 0.9999,\ \gamma_{13} = 1.0000,$

$\gamma_{14} = 1.0000$, and $\gamma_{15} = 1.0000$. The study established that the probabilities obtained will depend on the length of the time period $(\tau, T]$.

4.7 Simulation study for the two sample Bayesian prediction

In this section, two software failure data sets are generated from the Goel – Okumoto (1979) software reliability model. The two data sets are simulated using the same model and parameters. The simulated data is used to illustrate the methodologies developed for the two sample Bayesian predictive analyses. The simulation procedure was as follows. The Goel – Okumoto (1979) model is as given in Equation (13).

The values of $\alpha = 100$ and $\beta = 0.0010741$ were fixed. A value of T from the set $S = [200,\ 500,\ 10^3,\ 5\times10^3,\ 10^4]$ was selected. The study used $T = 200$. The simulation used in the study is for illustrative purposes only. Nevertheless, there is a practical interpretation to the choices of α, β and T. Case studies e.g. Musa (1987) have shown that a software fault density at the system testing stage is frequently on the order of five bugs per 1,000 lines of code. The choice of $\alpha = 100$ could be thought of as symbolizing a practically large software system that is on the order of 20,000 lines of codes. The choices for β and T together imply that most of the failures will be discovered during the simulated test period. Following the forgoing discussion, the following two data sets were simulated from the Goel – Okumoto (1979) software reliability model using the following steps:

Step 1: $t = 0,\ I = 0$

Step 2: Generate a random number U

Step 3: $t = t - \frac{1}{\lambda}\log U$. If $t > T$, stop.

Step 4: Generate a random number U.

Step 5: If $U \le \lambda(t)/\lambda$, set $I = I + 1$, $S(I) = t$.

Step 6: Go to step 2.

In the above steps, $\lambda(t)$ is known as the intensity function and λ is such that $\lambda(t) \le \lambda$. the last value of I represents the number of events time T, and $S(1), \ldots, S(I)$ are the event times. The above procedure of simulation is referred to as the thinning algorithm since it 'thins' the homogeneous Poisson points. It is the most efficient simulation procedure in the sense that it has the fewest number of rejected events times when $\lambda(t)$ is near λ throughout the interval (Sheldon, 2002). Using the above procedure, the following two data sets were generated. The first data set is assumed to be the software failure times from the first software and the second data set is assumed to be the failure times from the second software.

Software one: 8.9345, 27.0177, 34.5816, 54.8606, 83.5715, 111.4006, 139.8851, 157.4743, 181.0868, 182.8410

Software two: 2.3159, 16.2530, 20.5721, 23.3416, 42.8030, 46.7417, 61.0926, 63.8807, 75.1330, 80.7768, 97.3435, 117.9091, 129.3157, 138.0590, 169.3410, 172.7516, 186.0293, 193.1918, 198.5999

4.8 Maximum Likelihood Estimation

Suppose the observation of the failure times occurred in the time interval $(0, T]$ where $T = 200$, and n faults were observed at the failure times $0 < t_1 < t_2 < ... < t_n < T$. The joint density of the failure times is as in Equation (14). Taking the log-likelihood function of (14) gives

$$L = n \ln \alpha + n \ln \beta - \beta \sum_{i=1}^{n} t_i - \alpha (1 - e^{-\beta T}) \tag{96}$$

Differentiating L partially with respect to α and β and equating to zero (equating the partial derivatives of L to zero optimizes the values of α and β) gives

$$\frac{\partial L}{\partial \alpha} = \frac{n}{\alpha} - (1 - e^{-\beta T}) = 0. \tag{97}$$

$$\frac{\partial L}{\partial \beta} = \frac{n}{\beta} - \sum_{i=1}^{n} t_i - \alpha T e^{-\beta T} = 0. \tag{98}$$

Solving Equation (91) and Equation (92) for α and β, we obtain

$$\hat{\alpha} = \frac{n}{1 - e^{-\hat{\beta} T}} \tag{99}$$

$$\frac{n}{\hat{\beta}} = \sum_{i=1}^{n} t_i + \frac{n T e^{-\hat{\beta} T}}{1 - e^{-\hat{\beta} T}}. \tag{100}$$

Hossain and Dahiya (1993) showed that a necessary and sufficient condition for Equation (99) and Equation (100) to have a unique and positive solution $(\hat{\alpha}, \hat{\beta})$ is if and only if $(2 \sum_{i=1}^{n} t_i / n) < T$. That is, the ML estimates of α and β will exist only and only if two times the mean failure time is less than T. In most cases, the value of α and T will be such that $\Pr[N(T) = 0]$ is negligible. As a result, there will be one fault discovered during the testing $(\text{i.e., } n \geq 1)$ and the degenerate no failure case is of little concern, (Daniel and Hoang, 2001).

In most cases, the precision in the difference $1-e^{-\hat{\beta}T}$ in the denominator of the second part in the RHS of Equation (100) will be poor since $e^{-\hat{\beta}T}$ will always be very close to unity. This brings a numerical difficulty in finding the root of Equation (100). An alternative form of Equation (100) that overcomes this difficulty is

$$\frac{n}{\hat{\beta}} = \sum_{i=1}^{n} t_i + \frac{nT}{\sum_{j=1}^{\infty} \frac{(\hat{\beta}T)^j}{j!}}. \tag{101}$$

A numerical procedure known as the Newton Raphson method can be used to solve Equation (99) and Equation (101). The Newton Raphson method requires choosing initial values of α and β. Consequently, $\alpha = 95$ and $\beta = 0.0012$ were chosen as the initial values. There is no any other explanation to the choosing of the initial values other than the fact that they are very close to the values $\alpha = 100$ and $\beta = 0.0010741$ that were used during the simulation of the two software failure data sets in section 4.6. Consequently, the ML estimates $\hat{\alpha} = 102.756$ and $\hat{\beta} = 0.001022177$ for software one were obtained.

4.9 Real example for two-sample Bayesian prediction

Here, we use the two software data sets simulated in section 4.6 to illustrate the developed methodologies in section 4.4 for two sample Bayesian prediction problems. Assuming that the two software systems were observed in the time interval $(0, 200]$, and their successive failure times are given by:

Software one: 8.9345, 27.0177, 34.5816, 54.8606, 83.5715, 111.4006, 139.8851, 157.4743, 181.0868, 182.8410

Software two: 2.3159, 16.2530, 20.5721, 23.3416, 42.8030, 46.7417, 61.0926, 63.8807, 75.1330, 80.7768, 97.3435, 117.9091, 129.3157, 138.0590, 169.3410, 172.7516, 186.0293, 193.1918, 198.5999

The two software failure times are simulated from the same Goel – Okumoto (1979) software reliability model. The three issues in the two sample prediction in Chapter Three are addressed as follows:

Issue A2: First, we assume that the failure times of the second software were not observed. Based on the failure data of software one, the maximum likelihood estimate of β is given by 0.001022177. When β is known to be 0.001022177, and from Prop. A2, the Bayesian UPL for the $15th$ failure time of the second software with level $\gamma = 0.90$ is $y_U^{(\beta)} = 33.737$ such that $\gamma = \Gamma(r+n)\left[\Gamma(r)\Gamma(n)\right]^{-1}\beta\left(1-e^{-\beta T}\right)^n \int_0^{y_U^{(\beta)}} \frac{e^{-\beta y_r}\left(1-e^{-\beta y_r}\right)^{r-1}}{(2-e^{-\beta y_r}-e^{-\beta T})^{r+n}} dy_r$.

Issue B2: If $\beta = 0.001022177$, then from Prop. B2, the probability that the number of failures in the time interval $(0, 200]$ for the second software not exceeding a pre-determined non-negative integer $m = 16$, is $\gamma = 0.9157$.

Issue C2: suppose that the number of observed failures of the second software during $(0, 200]$ is $m = 15$. Based on the failure data of the second software, if $\beta = 0.001022177$, then from Prop. C2, the Bayesian UPL for y_{15} with level $\gamma = 0.90$ is $y_U^{\beta} = 198.00$.

CHAPTER FIVE

SUMMARRY, CONCLUSIONS AND RECOMMENDATIONS

5.1 Introduction

In this chapter, the summary of the thesis is provided. Conclusions from the analysis carried out in the thesis and recommendations for further research are also discussed.

5.2 Summary

Many software reliability models have been developed by various authors in the past three decades, amongst, the Goel – Okumoto (1979) software reliability model. The models are mainly based on the history of failure of software and they can be categorized depending on the nature of the failure process studied. 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. The study has proposed four issues associated closely to software development testing process in the single sample case and three issues in two sample case. Explicit solutions to these issues have been provided. The derived methodologies for the single sample case have been illustrated by time between software failure data given by (Xie *et al.* 2002). For the two sample case, simulated data from the Goel-Okumoto software reliability model have been used to illustrate the derived solutions.

5.3 Conclusions

This research work has mainly focused on deriving explicit solutions to various prediction issues that may arise during the software development testing program using Bayesian approach. These solutions may prove to be very 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 (Phillips, 2000; Quigley and Walls, 2003). 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 (Jun-Wu *et al.* 2007). The study has used both informative and non-informative priors to explicitly derive the posterior and predictive distributions to address the discussed software prediction problems. In both cases, closed form solutions to some of the predictive inferences are not available and as such, the study has used MCMC.

Gamma distribution has been used as the prior distribution in the case of single-sample prediction using informative prior. This is because experience in Bayesian modeling has shown that gamma distributions are reasonable representations for failure rates. They provide a spread of values from which the Likelihood function can emphasize the values needed to fit the data (Allan, 2012). Choosing of the parameters of the gamma prior distribution has poised some challenges. The study thus resorted to trial and error in choosing the gamma parameters until results that were closer to those from non-informative priors were reached.

5.4 Recommendations

This thesis has only considered and derived predictive procedures on the Goel – Okumoto (1979) software reliability model. However, the procedures presented in this thesis can also be extended to other NHPPs such as the delayed S-shaped process, the Cox – Lewis process and the Musa-Okumoto process. In addition, in the investigation of two-sample prediction, this study has only considered the derivations of the methodologies for the case when the shape parameter β is known. It may be of interest to derive the methodologies for the case of unknown β . Again, for the two-sample prediction, the study has assumed that the failure times of the two software systems obey the same Goel – Okumoto (1979) software reliability model. When this assumption is violated, (e.g., $\beta_1 = \beta_2 = \beta$ but $\alpha_1 \neq \alpha_2$), it is still of research interest to study the two-sample prediction problems. This is left open for future research.

REFERENCES

Allan, T. M., (2012). *Bayesian Statistics applied to Reliability Analysis and Prediction*. Raython Missile Systems, Tucson, AZ.

Amrit, L. G., (1985). Software Reliability Models: Assumptions, Limitations and Applicability. *IEEE Transactions on Software Engineering*, **SE-11**: 1411-1423.

Andrew, G., John, B. C., Hal, S. S. and Donald, B. R., (1995). *Bayesian Data Analysis*. Texts in Statistical Science, Chapman & Hall.

Bain, L. J., (1978). *Statistical Analysis of Reliability and Life-Testing Models*. Marcel Dekker, NewYork.

Bain, L. J. and Engelhardt, M., (1980). Inference on the parameters and current system reliability for a time-truncatedWeibull process. *Technometrics*, **22**: 421–426.

Crow, L. H., (1982). Confidence interval procedures for the Weibull process with application to reliability growth. *Technometrics*, **24**: 67–72.

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.

Duane, J. T., (1964). Learning Curve Approach To Reliability Monitoring. *IEEE Transactions on Aerospace*, **AS-2**: 563-566.

Glickman, M. E. and Van D. D. A., (2007). Basic Bayesian Methods. *In: Methods in Molecular Biology: Topics in Biostatistics* (WT Ambrosius, ed.). Humana Press, Totwa, NJ, USA: **104**: 319-338.

Goel, A. L. and Okumoto, K., (1979). Time-dependent error detection rate model for software reliability and other performance measures. *IEEE Trans.Reliability*, **28**: 206–211.

Guida, M., Alabria, R., Pulcini, G., (1989). Bayes inference for a nonhomogeneous Poisson process with power intensity law. *IEEE Trans. Reliability*, **38**: 603-609.

Hossain, S. A. and Dahiya, R. C., (1993). Estimating the Parameters of a Non- homogenous Poisson-Process Model for Software Reliability. *IEEE Tranis- actions on Reliability*, **42**: 604-612.

Jelinski, Z. and Moranda, P., (1972). *Software Reliability Research. Statistical Computer Performance Evaluation*, Ed. W. Freiberger. New York: Academic.

Jun-Wu, Y., Guo-Liang, T. and Man-Lai, T., (2007). Predictive analyses for non-homogeneous Poisson processes with power law using Bayesian approach. *Computational Statistics & Data Analysis*, **51**: 4254-4268.

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 2011.

Lynn, K. and Tae Y., (1995). Bayesian Computation of Software Reliability. *Journal of Computation and Graphical Statistics*, **4**: 65-82.

Lynn, K. and Tae, Y., (1996). Bayesian Computation for Nonhomogeneous Poisson process in Software Reliability. *Journal of the American Statistical Association*, **91**: 763-773.

Meth, M., (1992). *Reliability growth myths and methodologies: a critical view*. In: Proceedings of the Annual Reliability and Maintainability Symposium.IEEE, NewYork, 230–238.

Meyfroyt, P. H. A., (2012). *Parameter Estimation for Software Reliability Models*. Masters Thesis, Eindhoven University of Technology, Eindhoven, Netherlands.

Muralidharan K., Rupal, S. and Deepak H. D., (2008). Future Reliability Estimation Based on Predictive Distribution in Power Law Process. *Quality Technology & Quantitative Management*, **5**: 193-201.

Musa, J. D. and Okumoto, K., (1984). A Logarithmic Poisson Execution Time Model for Software Reliability Measurement. *In Proceedings of the Seventh International Conference on Software Engineering*, pp. 230- 238.

Musa, J., (1987). *Software Reliability: Measurement, Prediction, Application*. New York: McGraw-Hill.

Nuria, T. R., (2011). *Stochastic Comparisons and Bayesian Inference in Software Reliability*. Ph.D. Thesis, Universidad Carlos III de Madrid.

Phillips, M.J., (2000). Bootstrap confidence regions for the expected ROCOF of a repairable system. *IEEE Trans. Reliability* **49:** 204–208.

Preeti, W. S. and Nidhi J., (2011). Reliability Prediction During Development Phase of a System. *Quality Technology & Quantitative Management*, **8**: 111-124.

Quigley, J. and Walls, L., (2003). Confidence intervals for reliability-growth models with small sample-sizes. *IEEE Trans. Reliability* **52:** 257–262.

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.

Rigdon, S.E. and Basu, A.P., (2000). The Power Law Process: a Model for the Reliability of Repairable systems. *Journal of Quality Technology*, **21**: 251-260.

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.

Shaul, K. B., Idit, L. and Benjamin, R., (1992). Bayesian Inference for the Power Law Process. *Ann. Inst. Statist. Math.*, **44**: 623-639.

Sheldon, R., (2002). Simulation. 3rd edition. *Academic Press.*

Singpurwalla, N. D and Simon, P. W., (1994). Software Reliability Modeling *International*

Statistical Review/ Revue Internationale de Statistique, **62**: 289-317.

Stringfellow, C. and Amschler, A. A., (2002). An Empirical Method for Selecting Software Reliability Growth Models. *Empirical Software Engineering*, **7**: 319-343.

Tae, Y. Y. and Lynn K., (1999). Bayesian Computation for the Superposition of Non-homogeneous Poisson Processes. *The Canadian Journal of Statistics,* **27**: 547-556.

Xie, M., Goh, T. N and Ranjan, P., (2002). *Some effective control chart procedures for reliability monitoring*. Elsevier,Reliability engineering and System safety.

Yamada, S., Ohba, M. and Osaki, S., (1983). S-shaped reliability growth modeling for software error detection. *IEEE Trans. Reliability*, **32**: 475–484.

Zhao, C., (2010). Empirical Bayes Analysis on the Power Law Process with Natural Conjugate Priors. *Journal of Data Science,* **8**: 139-149.

Zhao, C., (2004). *Bayesian and Empirical Bayes approaches to power law process and microarray analysis*. PHD Thesis, University of South Florida. USA.

Zhao, M. and Xie, M. (1996). On Maximum Likelihood Estimation for a General Non-homogeneous Poisson Process. *Scandinavian Journal of Statistics,* **23**: 597-607.

APPENDIX

The following are the R-codes that were used in the simulation and computation of the results.

One Sample Bayesian prediction using non-informative prior

Proposition A1

i) When β is known, the program used was:

```
n<-30
lamtv<-0.03
T<-182.21
tao<-277.83
beta<-0.003962
h<-0:(n-1)
d<-((1-exp(-beta*T))*lamtv)/(beta*exp(-beta*tao))
t<-(d^h)/factorial(h)
z<-t*exp(-d)
y<-1-sum(z)
y
```

ii) When β is unknown, the program used was:

```
n<-30
st<-7190.86 # sum of the ti's
lamtv<-0.03
T0<-182.21
tau<-277.83
g1<-function(b){
        (1-exp(-b*T0))^-n*factorial(n-1)/st^n
       }
g2<-function(b,h){
```

```
        b1<-(1-exp(-b*T0))*lamtv/(b*exp(-b*tau))
        b2<-1-exp(-b*T0)
        (b1^h/factorial(h))*exp(-b1)*factorial(n-1)/(b2*st)^n
        }
H<-0:(n-1)
vec1<-rep(0,n)
for (j in 1:n){
        h<-H[j]
        G<-1000
        vec2<-rep(0,G)
        vec3<-rep(0,G)
        for (i in 1:G){
                br<-rgamma(1,n,st)
                vec2[i]<-g1(br)
                vec3[i]<-g2(br,h)
                }
        vec1[j]<-sum(vec3)/sum(vec2)
        }
1-sum(vec1)
```

Proposition B1

i) When β is known, the program used was:

```
beta<-0.003962
n<-30
T<-182.21
lambdatv<-0.03
chi<-qchisq(0.10, df =  60)
```

```
x<-(1-exp(-beta*T))
y<-(beta*chi)/(2*lambdatv*x)
taostar<-(log(y)/beta)-T
taostar
```

ii) When β is unknown, the program used was:

```
n<-30
st<-7190.86 # sum of the ti's
lamtv<-0.03
T0<-182.21
vectau<-890:1090
p<-0.7
g1<-function(b){
        (1-exp(-b*T0))^-n*factorial(n-1)/st^n
        }
g2<-function(b,h,tau){
        b1<-(1-exp(-b*T0))*lamtv/(b*exp(-b*tau))
        b2<-1-exp(-b*T0)
        (b1^h/factorial(h))*exp(-b1)*factorial(n-1)/(b2*st)^n
        }
k<-1
vecp<-c()
while(p<=0.9){
tau<-vectau[k]
H<-0:(n-1)
vec1<-rep(0,n)
for (j in 1:n){
```

```
        h<-H[j]
        G<-1000
        vec2<-rep(0,G)
        vec3<-rep(0,G)
        for (i in 1:G){
                br<-rgamma(1,n,st)
                vec2[i]<-g1(br)
                vec3[i]<-g2(br,h,tau)
               }
        vec1[j]<-sum(vec3)/sum(vec2)
        }
vecp[k]<-p
p<-1-sum(vec1)
k<-k+1
}
which(vecp==max(vecp))+min(vectau)
```

Proposition C1

i) When β is known, the program used was:

```
beta<-0.003962
tau<-900
n<-30
T<-182.21
chi<-qchisq(0.10, df =  60)
x<-(1-exp(-beta*T))
UPL<-(chi*beta*exp(-beta*tau))/(2*x)
UPL
```

ii) When β is unknown, the program used was

```
n<-30
st<-7190.86 # sum of the ti's
veclamtv<-seq(0.03, 1,length=1000)
T0<-182.21
tau<-277.83
g1<-function(b){
        (1-exp(-b*T0))^-n*factorial(n-1)/st^n
        }
g2<-function(b,h,lamtv){
        b1<-(1-exp(-b*T0))*lamtv/(b*exp(-b*tau))
        b2<-1-exp(-b*T0)
        (b1^h/factorial(h))*exp(-b1)*factorial(n-1)/(b2*st)^n
        }
k<-1
vecp<-c()
p<-.7
while(p<=0.9){
lamtv<-veclamtv[k]
H<-0:(n-1)
vec1<-rep(0,n)
for (j in 1:n){
        h<-H[j]
        G<-1000
        vec2<-rep(0,G)
        vec3<-rep(0,G)
        for (i in 1:G){
```

```
                    br<-rgamma(1,n,st)
                    vec2[i]<-g1(br)
                    vec3[i]<-g2(br,h,lamtv)
                   }
          vec1[j]<-sum(vec3)/sum(vec2)
          }
vecp[k]<-p
p<-1-sum(vec1)
k<-k+1
}
veclamtv[which(vecp==max(vecp))]
```

Proposition D1

i) When β is known, the program used was:

```
T0<-180
tau<-250
n<-30
b<-0.003962
const1<-(1-exp(-b*T0))/(exp(-b*T0)-exp(-b*tau))
const2<-(exp(-b*T0)-exp(-b*tau))/(1-exp(-b*tau))
k<-0:15
vecyk<-rep(0,length(k))
for (i in 1:length(k)){
             J<-n:(n+k[i])
             a<-rep(0,length(J))
             for (j in 1:length(J)) a[j]<-choose(J[j]-1,n-1)*const2^J[j]
             vecyk[i]<-const1^n*sum(a)}
```

```
vecyk
```

ii) When β is unknown, the program used was:

```
T0<-180
tau<-240
st<-7190.86
n<-30
k<-0:15
G<-1000
vecyk2<-rep(0,length(k))
fb<-function(b0)(exp(-b0*T0)-exp(-b0*tau))^(j0-n)/((1-exp(-b0*tau))^j0*st^n)
g1<-function(b){
          (1-exp(-b*T0))^-n*factorial(n-1)/st^n
         }
for (i in 1:length(k)){
              J<-n:(n+k[i])
              a<-rep(0,length(J))
              for (j in 1:length(J)){
                j0<-J[j]
                vecb0<-rgamma(G,n,st)
                d<-mean(g1(vecb0))
                k0<-mean(fb(vecb0))
                a[j]<-factorial(J[j]-1)*k0/(d*factorial(J[j]-n))}
vecyk2[i]<-sum(a)
}
vecyk2
```

One-Sample Bayesian prediction using informative prior

Proposition A1.1

i) When β is known, the program used was:

```
n<-30
lamtv<-0.03
T<-182.21
tao<-277.83
beta<-0.003962
a<-2
b<-1/2
c<-2
d<-1/2
h<-0:(a+n-1)
d<-((1-exp(-beta*T)+b)*lamtv)/(beta*exp(-beta*tao))
t<-(d^h)/factorial(h)
z<-t*exp(-d)
y<-1-sum(z)
y
```

ii) When β is unknown, the program used was:

```
n<-30
st<-7190.86 # sum of the ti's
lamtv<-0.03
T0<-182.21
tau<-277.83
```

```
a<-2
b<-1/2
c<-2
d<-1/2.
g1<-function(x){
        (1-exp(-x*T0)+b)^-(n+a)*factorial(n+c-1)/(st+d)^(n+c)
        }
g2<-function(x,h){
        k1<-(1-exp(-x*T0)+b)*lamtv/(x*exp(-x*tau))
        k2<-factorial(n+c-1)/((1-exp(-x*T0)+b)^(n+a)*(st+d)^(n+c))
        (k1^h/factorial(h))*exp(-k1)*k2
         }
H<-0:(n+a-1)
vec1<-rep(0,(n+a))
for (j in 1:(n+a)){
        h<-H[j]
        G<-1000
        vec2<-rep(0,G)
        vec3<-rep(0,G)
        for (i in 1:G){
                br<-rgamma(1,n+c,st+d)
                vec2[i]<-g1(br)
                vec3[i]<-g2(br,h)
               }
        vec1[j]<-sum(vec3)/sum(vec2)
        }
1-sum(vec1)
```

Proposition B1.1

i) When β is known, the program used was:

```
n<-30
lamtv<-0.03
T<-182.21
tao<-277.83
beta<-0.003962
t<-qchisq(0.10,2*n)
a<-2
b<-1/2
c<-2
d<-1/2
r<-(beta*t)/(2*lamtv*(1-exp(-beta*T)+b))
taoprime<-((1/beta)*log(r))-T
taoprime
```

ii) When β is unknown, the program used was:

```
n<-30
st<-7190.86 # sum of the ti's
lamtv<-0.03
T0<-182.21
p<-0.7
vectau<-300:1090
#values of the parameters of prior distribution
a<-2
b<-1/2
```

```
c<-2
d<-1/2
g1<-function(x){
        (1-exp(-x*T0)+b)^-(n+a)*factorial(n+c-1)/(st+d)^(n+c)
        }
g2<-function(x,h,tau){
        k1<-(1-exp(-x*T0)+b)*lamtv/(x*exp(-x*tau))
        k2<-factorial(n+c-1)/((1-exp(-x*T0)+b)^(n+a)*(st+d)^(n+c))
        (k1^h/factorial(h))*exp(-k1)*k2
         }
k<-1
vecp<-c()
while(p<=0.9){
   tau<-vectau[k]
   H<-0:(n+a-1)
   vec1<-rep(0,(n+a))
   for (j in 1:(n+a)){
          h<-H[j]
          G<-1000
          vec2<-rep(0,G)
          vec3<-rep(0,G)
          for (i in 1:G){
             br<-rgamma(1,n+c,st+d)
             vec2[i]<-g1(br)
             vec3[i]<-g2(br,h,tau)
            }
          vec1[j]<-sum(vec3)/sum(vec2)
```

```
        }
vecp[k]<-p
p<-1-sum(vec1)
k<-k+1
}
vecp
which(vecp==max(vecp))+min(vectau)
```

Proposition C1.1

i) When β is known, the program used was:

```
n<-30
lamtv<-0.03
T<-182.21
tao<-277.83
taoprime<-900
beta<-0.003962
t<-qchisq(0.10,2*n)
a<-2
b<-1/2
c<-2
d<-1/2
r<-(beta*t*exp(-beta*taoprime))/(2*(1-exp(-beta*T)+b))
r
```

ii) When β is unknown, the program used was:

```
n<-30
st<-7190.86 # sum of the ti's
```

```
veclamtv<-seq(0.03, 1,length=1000)
T0<-182.21
tau<-277.83
p<-0.7
#values of the parameters of prior distribution
a<-2
b<-1/2
c<-2
d<-1/2
g1<-function(x){
        (1-exp(-x*T0)+b)^-(n+a)*factorial(n+c-1)/(st+d)^(n+c)
        }
g2<-function(x,h,tau){
        k1<-(1-exp(-x*T0)+b)*lamtv/(x*exp(-x*tau))
        k2<-factorial(n+c-1)/((1-exp(-x*T0)+b)^(n+a)*(st+d)^(n+c))
        (k1^h/factorial(h))*exp(-k1)*k2
         }
k<-1
vecp<-c()
p<-.7
while(p<=0.9){
lamtv<-veclamtv[k]
H<-0:(n+a-1)
vec1<-rep(0,n+a)
for (j in 1:(n+a)){
        h<-H[j]
        G<-1000
```

```
        vec2<-rep(0,G)
        vec3<-rep(0,G)
        for (i in 1:G){
                br<-rgamma(1,n+a,st+d)
                vec2[i]<-g1(br)
                vec3[i]<-g2(br,h,lamtv)
               }
        vec1[j]<-sum(vec3)/sum(vec2)
        }
vecp[k]<-p
p<-1-sum(vec1)
k<-k+1
}
veclamtv[which(vecp==max(vecp))]
```

Proposition D1.1

i) When β is known, the program used was:

```
T0<-180
tau<-240
n<-30
x<-0.003962
e<-2
b<-1/2
c<-2
d<-1/2
const1<-(1-exp(-x*T0)+b)/(exp(-x*T0)-exp(-x*tau))
const2<-(exp(-x*T0)-exp(-x*tau))/(1-exp(-x*tau)+b)
```

```
const3<-(1-exp(-x*T0)+b)/(1-exp(-x*tau)+b)
k<-0:10
vecyk<-rep(0,length(k))
for (i in 1:length(k)){
        J<-n:(n+k[i])
        a<-rep(0,length(J))
        for (j in 1:length(J+a)) a[j]<-choose(J[j]+e-1,n+e-1)*const2^J[j]
        vecyk[i]<-const1^n*const3^e*sum(a)}
vecyk
```

Two Sample-Bayesian Prediction

Simulation

For simulation and calculation of the MLE for the simulated data, the following R-codes was used:

```
t<-0; I<-0; T<-200; alpha<-100; beta<-0.0010741
lambda0<-alpha*beta*exp(-beta*T)
lambda<-function(t)   alpha*beta*exp(-beta*t)
S0<-c()
while(t<=T) {
  u<-runif(1)
  t<-t-log(u)/lambda0
  u1<-runif(1)
    if(u1<=lambda(t)/lambda0){
          I<-I+1
          S0[I]<-t  } else {I<-I}
}
S<-S0[-length(S0)]
```

```
n<-length(S)
Loglik<-function(par){
        a<-par[1]
        b<-par[2]
        -n*log(a)-n*log(abs(b))+abs(b)*sum(S)+a*(1-exp(-abs(b)*T))

}
par0<-c(147,0.0012)
2*sum(S)/n
mle<-optim(par0,Loglik)
mle
est<-mle$par
est
```

Proposition A2

The following program was used to illustrate Proposition A2:

```
n<-10
r<-15
T<-200
b<-0.001022177
const<-(factorial(r+n-1)*b*(1-exp(-b*T)))/(factorial(r-1)*factorial(n-1))
g1<-function(yr){
      const*(exp(-b*yr)*(1-exp(-b*yr))^(r-1))/((2-exp(-b*yr)-exp(-
b*T))^(r+n))
     }
g<-integrate(g1,0,33.73700)
g
```

Proposition B2

The following program was used to illustrate the Proposition B2:

```
T<-200
t2<-200
n<-10
b<-0.001022177
k<-0:16
z<-((1-exp(-b*t2))^k)/(2-exp(-b*t2)-exp(-b*T))^k
g<-((1-exp(-b*T))^n)/(2-exp(-b*t2)-exp(-b*T))^n
r<-choose(n+k-1,k)
y<-z*r*g
sum(y)
```

Proposition C2

The following program was used to illustrate the Proposition C2

```
m<-15
yu<-1:200
b<-0.001022177
t2<-200
t<-(1-exp(-b*yu))^m
k<-(1-exp(-b*t2))^m
y<-t/k
y[198]
```