

I^2SDS
The Institute for Integrating Statistics in Decision Sciences

Technical Report TR-2009-1
January 26, 2009

A Markov Modulated Poisson Model for Software Reliability

Joshua Landon
Institute for Integrating Statistics in Decision Sciences
The George Washington University, USA

Süleyman Özekici
Department of Industrial Engineering
Koc University, Turkey

Refik Soyer
Department of Decision Sciences
The George Washington University, USA

A Markov Modulated Poisson Model for Software Reliability

Joshua Landon*, Süleyman Özekici† and Refik Soyer‡

Abstract

In this paper we consider a latent Markov process governing the intensity rate of software failures. The latent process represents the behavior of the debugging operations over time and enables us to deal with the imperfect debugging scenario. We develop the Bayesian inference for the model and also introduce a method to infer the unknown dimension of the Markov process. We illustrate the implementation of our model and the Bayesian approach by using actual software failure data.

Keywords. Software reliability, Hidden Markov model, Bayesian inference

1 Introduction

Poisson process and its extensions are widely used in software reliability modeling. These point process models are generated by the interfailure times of the software. One of the original Poisson process models is the *time-dependent error detection* model of Goel and Okumoto (1980). Most other Poisson process models are extensions of the Goel-Okumoto model. Such extensions and Kuo and Yang's (1996) unification of the Poisson process models via *general order and record value statistics* are discussed in the recent review by Soyer (2011).

A Markov-modulated Poisson process (MMPP) is a Poisson process whose intensity depends on the current state of an independently evolving continuous-time Markov process. Özekici and Soyer (2006) discuss probabilistic and statistical issues related to these processes

*The George Washington University, Department of Statistics, Washington, DC 20052, USA

†Koç University, Department of Industrial Engineering, 34450 Istanbul, Turkey

‡Corresponding Author, The George Washington University, Department of Decision Sciences, Washington, DC 20052, USA (soyer@gwu.edu)

in a rather general setting. These processes are used in a variety of applications in queueing, inventory, and reliability. The use of modulating processes in these applications have gained a lot of attention because they make the models more realistic. The customer arrival process in queueing and inventory models are often affected by an external environmental process that represents economic, financial, social or other factors in which the system operates. The arrival rate of customers depend on the state of the environment as well as other stochastic or deterministic parameters. Prabhu and Zhu (1989) discuss Markov modulated queues, and Arifoğlu and Özekici (2010) analyze an inventory model operating in a partially observable random environment.

First consideration of MMPPs in software reliability applications is due to Özekici and Soyer (2003) who assume that the failures of the software depend on its operational profile. Musa (1993) defines the operational profile as the set of all operations that a software system is designed to perform and their occurrence probabilities. The modulating process now represent how the software is used and, thus, the failure process depends on it. Özekici and Soyer (2003) consider a setting where there is perfect debugging during the testing of software and the state of the environmental process is observed. In this paper, we relax both of these assumptions. More specifically, in our setup we consider software that goes through a debugging process, but we do not assume that debugging is perfect. Instead, we assume that the failure rate of the software is nonhomogeneous and modulated by the environmental process and this enables us to consider both improvements and deteriorations in the software as a result of the debugging.

Unlike Özekici and Soyer (2003), in our setup the environmental process is latent and it represents the state of the debugging process as well as the operations that the software goes through during the process. Our objective is to provide a computationally tractable Bayesian procedure to make inference on the software failure rates and the parameters of the modulating Markov process based on observed data. In our setting, since the state of the modulating process is latent or hidden, we only have information on the failure times. An interesting problem then is to make statistical inferences on the hidden process based

on failure data. Our model extends the discrete time hidden Markov model considered by Durand and Gaudoin (2005) to continuous time and develops its Bayesian analysis. Applications of our model in reliability also include hardware reliability where a device performs a stochastic mission and its failure rate depends on the stage of the mission. Çekyay and Özekici (2010) discuss issues related to mean time to failure and availability when the mission or environmental process is semi-Markovian.

The details of our model will be presented in Section 2 where the stochastic structures of the modulating and failure processes are described. In Section 3 we will assume that the number of states of the hidden process is known, and show how we can estimate the software failure rates as well as the transition rates of the Markov process. Then, in Section 4 we will assume that we do not know the number of states of the hidden Markov process, and will present an approach to obtain the marginal likelihood based on Chib (1995) that will enable us to infer the unknown number of states. Finally, our results will be demonstrated using actual software failure data in Section 5. Conclusions follow in Section 6.

2 Markov Modulated Poisson Process Model for Software Reliability

Let $N = \{N_t; t \geq 0\}$ be a modulated Poisson process such that N_t depicts the total number of software failures until time t . There is an environmental process that modulates the software failure rates. In other words, the software failure rate is random at any time depending on the state of this environmental process. In software reliability, the environmental process can have several different interpretations. In Özekici and Soyer (2003), for example, it is used to represent the operational profile or process which depicts the operations that the software performs randomly after it is released. It may also be used to represent the state of the software during testing and debugging with respect to its failure properties. At any time, the software can be classified to be in “excellent”, “good”, or “bad” state depending on the number of faults or other failure properties.

Although our analysis applies to both of these cases, we motivate it by the latter so that the environmental process represents the state of the software during testing and debugging. This state is not necessarily observed and the software failure rate depends on the unobserved state.

The environmental process is $Y = \{Y_t; t \geq 0\}$ where Y_t represents the state of the software during testing and debugging at time t . Since the effect of the changes made to the software during the debugging process is not necessarily observable, Y is a latent or hidden process. We assume that $Y = \{Y_t; t \geq 0\}$ is a continuous-time Markov process with a finite state space $E = \{1, 2, \dots, K\}$ where K is the number of states. We further assume that when the state of the software is $i \in E$, failures occur according to an ordinary Poisson process with rate λ_i . Therefore, the rate of failures at time t is λ_{Y_t} . To be more precise,

$$P[N_t = k|Y] = \frac{e^{-A_t} A_t^k}{k!} \quad (1)$$

where

$$A_t = \int_0^t \lambda(Y_s) ds \quad (2)$$

for all $k = 0, 1, \dots$ and $t \geq 0$.

It follows from the above that, given Y , N is a nonstationary Poisson process with mean value function $E[N_t|Y] = A_t$. Letting $T = \{T_n; n = 0, 1, 2, \dots\}$ denote the software failure process so that T_n is the time of the n th failure, we have the conditional distribution

$$P[T_{n+1} - T_n > t|Y, T_n] = e^{-(A_{T_n+t} - A_{T_n})}. \quad (3)$$

The modulated process reduces to the ordinary Poisson process with rate λ if the failure rate vector is $\lambda_i = \lambda$ independent of the state of Y . In this case, $A_t = \lambda t$ deterministically.

The failure process N can be studied via the additive functional A of Y . In particular, (1) and (3) directly yield

$$P[N_t = k] = E \left[\frac{e^{-A_t} A_t^k}{k!} \right] \quad (4)$$

and

$$P[T_{n+1} - T_n > t|T_n] = E[e^{-(A_{T_n+t} - A_{T_n})}]. \quad (5)$$

Therefore, the probability law of A , thus that of the environmental process Y , plays an important role in our analysis of N and T .

Let X_n be the n th state of the environmental process and U_n denote the time when the process enters the n th state so that $Y_t = X_n$ whenever $U_n \leq t < U_{n+1}$. Since Y is a Markov process, it is well-known that the sequence of states $X = \{X_n; n = 0, 1, 2, \dots\}$ is a Markov chain with state space E and some transition matrix $P_{ij} = P[X_{n+1} = j | X_n = i]$ with $P_{ii} = 0$ for all $i, j \in E$. Moreover, the duration of any state is exponentially distributed so that $P[U_{n+1} - U_n > t | X_n = i] = e^{-\rho_i t}$ where ρ_i is the holding rate for state i . The transition rate matrix or generator of the Markov process Y is

$$G_{ij} = \begin{cases} -\rho_i, & \text{if } j = i \\ \rho_i P_{ij}, & \text{if } j \neq i \end{cases} \quad (6)$$

or $G_{ij} = \rho_i(P_{ij} - I_{ij})$ where I is the identity matrix.

Following Özekici and Soyer (2006), we define another process Y^λ such that

$$Y_t^\lambda = \begin{cases} Y_t, & \text{if } t < T_1 \\ \Delta, & \text{if } t \geq T_1 \end{cases} \quad (7)$$

where T_1 is the time of the first failure. While the process is in state i , the time to failure has the exponential distribution with rate λ_i . It is clear that Y^λ is also a Markov process on the extended state space $E_\Delta = E \cup \{\Delta\}$ and it is obtained by “stopping” the Markov process Y as soon as a failure occurs. Here, Δ is an absorbing state where the process is dumped to as soon as it is stopped. The transition matrix of the embedded Markov chain is now extended as

$$P_{ij}^\lambda = \begin{cases} \frac{\rho_i}{\rho_i + \lambda_i} P_{ij}, & \text{if } i, j \in E \\ \frac{\lambda_i}{\rho_i + \lambda_i}, & \text{if } i \in E, j = \Delta \\ 1, & \text{if } i, j = \Delta \end{cases} \quad (8)$$

and the transition rate vector is

$$\rho_i^\lambda = \begin{cases} \rho_i + \lambda_i, & \text{if } i \in E \\ 0, & \text{if } i = \Delta. \end{cases} \quad (9)$$

If we let the matrix $G_{ij}^\lambda = \rho_i^\lambda(P_{ij}^\lambda - I_{ij})$ denote the generator of Y^λ , then it is well known that the transition function $P_{ij}^\lambda(t) = P[Y_t^\lambda = j | Y_0^\lambda = i]$ for all $i, j \in E_\Delta$ is given by

the matrix-exponential solution

$$P^\lambda(t) = \exp\left(G^\lambda t\right) = \sum_{n=0}^{+\infty} \frac{t^n}{n!} \left(G^\lambda\right)^n. \quad (10)$$

A further simplification is obtained by noting that

$$G_{ij}^\lambda = G_{ij} - \Lambda_{ij} \quad (11)$$

for all $i, j \in E$ where Λ is a diagonal matrix defined as

$$\Lambda_{ij} = \begin{cases} \lambda_i, & \text{if } j = i \\ 0, & \text{if } j \neq i. \end{cases} \quad (12)$$

Since $G_{\Delta j}^\lambda = 0$ and $G_{i\Delta}^\lambda = \lambda_i$ for all $i \in E$ and $j \in E_\Delta$, we can rewrite (10) as

$$P_{ij}^\lambda(t) = \exp\left(G^\lambda t\right)_{ij} = \exp\left((G - \Lambda)t\right)_{ij} \quad (13)$$

for all $i, j \in E$.

Now, note that our construction of Y^λ implies

$$T_1 = \inf\{t \geq 0; Y_t^\lambda = \Delta\} \quad (14)$$

and T_1 is the first-passage-time to the absorbing state Δ . So, it has a phase-type distribution and, in particular,

$$P[T_1 > t | Y_0 = i] = P[Y_t^\lambda \in E | Y_0 = i] = \sum_{j \in E} P_{ij}^\lambda(t) = \sum_{j \in E} \exp\left((G - \Lambda)t\right)_{ij}. \quad (15)$$

Note that in reliability applications where failures occur exponentially with a rate that depends on the randomly changing environmental process, (15) gives the survival function. In this case, the mean time to failure is another quantity of interest. Using the Markov property, it can be computed by solving the system of linear equations

$$E[T_1 | Y_0 = i] = \frac{1}{\rho_i^\lambda} + \sum_{j \in E} P_{ij}^\lambda E[T_1 | Y_0 = j] \quad (16)$$

for $i \in E$ so that the explicit solution is

$$E[T_1 | Y_0 = i] = \sum_{j \in E} \left[I - P^\lambda \right]_{ij}^{-1} \left(\frac{1}{\rho_j^\lambda} \right). \quad (17)$$

Following Özekici and Soyer (2006) an ergodic analysis can also be developed. Suppose that both X and Y are ergodic processes with limiting distributions $\nu_j = \lim_{n \rightarrow +\infty} P[X_n = j]$ and $\pi_j = \lim_{t \rightarrow +\infty} P[Y_t = j]$. This implies that ν is the unique solution of $\nu = \nu \mathbf{P}$ with the normalizing condition $\sum_{i \in E} \nu_i = 1$. It can be shown that

$$\pi_i = \frac{\nu_j / \rho_j}{\sum_{k \in E} (\nu_k / \rho_k)}. \quad (18)$$

Using Theorem 1 of Özekici and Soyer (2006), we can obtain

$$\lim_{t \rightarrow +\infty} E[N_t - \hat{\lambda}t | Y_0 = i] = \sum_{j \in E} \pi_j \left[\frac{\hat{\lambda} - \lambda_j}{\rho_j} \right] \quad (19)$$

where

$$\lim_{t \rightarrow +\infty} \frac{E[N_t | Y_0 = i]}{t} = \hat{\lambda} = \sum_{j \in E} \pi_j \lambda_j \quad (20)$$

is the average failure rate. It follows from Fischer and Meier-Hellstern (1992) that the expected number of arrivals until time t is

$$E[N_t | Y_0 = i] = \hat{\lambda}t + \sum_{j \in E} \left([\exp(Gt) - I] [G + \Pi]^{-1} \right)_{ij} \lambda_j \quad (21)$$

where the matrix $\Pi_{ij} = \pi_j$ has identical entries in each column.

Our primary objective is to develop statistical inference for the MMPP model using a Bayesian framework. It is important to note that in our model the Y process is latent and therefore in addition to the unknown parameters we also need to make inference about the latent states.

3 Bayesian Analysis of the MMPP Model

In this section we will illustrate how we can estimate all the parameters as well as the latent states in the MMPP model. The approach is based on the Markov Chain Monte Carlo (MCMC) method given in Fearnhead and Sherlock (2006). This method is based on a Gibbs sampler and requires a three-stage process. We first introduce some notation that will be used in describing the three stages. We define the software failure rates as $\lambda = \{\lambda_i; i \in E\}$, holding rates of the Y process as $\rho = \{\rho_i; i \in E\}$, and transition

probabilities as $\mathbf{P} = \{P_{ij}; i, j \in E\}$. Since there are K states in E and $P_{ii} = 0$ the total number of parameters to be estimated is K^2 . For example, when $K = 3$, the transition matrix for the three-state Markov chain is

$$\mathbf{P} = \begin{pmatrix} 0 & p_{12} & p_{13} \\ p_{21} & 0 & p_{23} \\ p_{31} & p_{32} & 0 \end{pmatrix}$$

where the sum of the probabilities in each row is one; so we only need to estimate one probability in each row, say p_{12} , p_{21} and p_{31} , when estimating \mathbf{P} . We assume that the system is observed until some time t_{obs} and we suppose that n failures are observed at times t_1, t_2, \dots, t_n during the observation interval $[0, t_{obs}]$.

In Stage 1, we will simulate the state of the hidden Markov process at each of the failure times given by our data set $\mathcal{D} = \{t_1, t_2, \dots, t_n\}$. Thus, all the results are conditional on the parameters λ, ρ and \mathbf{P} . In Stage 2, we will simulate the entire hidden Markov process, and in Stage 3 we will simulate a new set of parameter values using the Gibbs sampler. In the rest of this section we will outline what each of the three stage involves.

Stage 1

Since Y is a Markov process, the states $\{S_k = Y_{t_k}\}$ observed at times $0 = t_0 \leq t_1 \leq t_2 \leq \dots \leq t_n \leq t_{n+1} = t_{obs}$ form a Markov chain with transition probabilities

$$\mathbf{T}_{s_{k-1}, s_k}^{(k)} = P[S_k = s_k | S_{k-1} = s_{k-1}]$$

during the k th interval with duration $t_k - t_{k-1}$. It is well-known that $\mathbf{T}^{(k)}$ is the matrix exponential

$$\mathbf{T}^{(k)} = \exp [(\mathbf{G} - \mathbf{\Lambda})(t_k - t_{k-1})]. \quad (22)$$

We recursively define the matrices

$$\begin{aligned} \mathbf{A}^{(n+1)} &= \mathbf{T}^{(n+1)}, \\ \mathbf{A}^{(k)} &= \mathbf{T}^{(k)} \mathbf{\Lambda} \mathbf{A}^{(k+1)} \end{aligned} \quad (23)$$

for $k = n, n-1, \dots, 1$. Note that these matrices denote the likelihoods

$$\mathbf{A}_{s_{k-1}, s_{n+1}}^{(k)} = P[\{t_k, t_{k+1}, \dots, t_n\}, S_{n+1} = s_{n+1} | S_{k-1} = s_{k-1}].$$

So we first of all calculate $\{\mathbf{T}^{(k)}\}$ using (22), and then we calculate $\{\mathbf{A}^{(k)}\}$ using (23), starting with $\mathbf{A}^{(n+1)}$, and going backwards until we have

$$\mathbf{A}_{s_0, s_{n+1}}^{(1)} = P[\mathcal{D}, S_{n+1} = s_{n+1} | S_0 = s_0].$$

We will assume that we know $S_0 = s_0$ and $S_{n+1} = s_{n+1}$, the states of the Markov process at times 0 and t_{obs} , respectively. If they are unknown then we can adjust this algorithm slightly by putting a prior distribution on the state of the process at these times, but in our example we will assume that these states are known. Then, the state S_k of the Markov chain at time t_k can be simulated using the conditional distribution

$$P[S_k = s | \mathcal{D}, S_{k-1} = s_{k-1}, S_{n+1} = s_{n+1}] = \frac{T_{s_{k-1}, s}^{(k)} \lambda_s A_{s, s_{n+1}}^{(k+1)}}{A_{s_{k-1}, s_{n+1}}^{(k)}} \quad (24)$$

recursively by proceeding forwards through the observation times t_1, t_2, \dots, t_n .

Stage 2

After completing Stage 1 we will have our simulated states of the hidden Markov process $\{S_k\}$ at each of our observation times $\{t_k\}$. We will now use these to simulate the entire hidden Markov process Y . To do this we first of all simulate it over the interval (t_0, t_1) , then (t_1, t_2) and so on until (t_n, t_{n+1}) . The simulation over each interval is done using the uniformization of the Markov process Y supposing that $\rho = \max_{i \in E} \rho_i$ is finite. It is well-known (see, for example, Ross (1996)) that the Markov process Y can be represented as a Markov chain \hat{X} subordinated to a Poisson process \hat{N} with arrival rate ρ so that $Y_t = \hat{X}_{\hat{N}_t}$ and

$$\begin{aligned} P[Y_t = s_t | Y_0 = s_0] &= P[\hat{X}_{\hat{N}_t} = s_t | \hat{X}_{\hat{N}_0} = s_0] \\ &= \sum_{n=0}^{+\infty} P[\hat{N}_t = n] P[\hat{X}_n = s_t | \hat{X}_0 = s_0] \\ &= \sum_{n=0}^{+\infty} \frac{e^{-\rho t} (\rho t)^n}{n!} \mathbf{M}_{s_0, s_t}^n \end{aligned}$$

where

$$\mathbf{M} = \frac{1}{\rho} \mathbf{G} + \mathbf{I}. \quad (25)$$

is the transition matrix corresponding to the Markov chain \hat{X} . Over any interval (t_{k-1}, t_k) , we already obtained the simulated states $Y_{t_{k-1}} = s_{k-1}$ and $Y_{t_k} = s_k$ in Stage 1. Therefore, the conditional distribution of the number failures of \hat{N} during (t_{k-1}, t_k) is

$$\begin{aligned} P[\hat{N}_{t_k} - \hat{N}_{t_{k-1}} = n \mid Y_{t_{k-1}} = s_{k-1}, Y_{t_k} = s_k] & \quad (26) \\ = \left(\frac{e^{-\rho(t_k - t_{k-1})} (\rho(t_k - t_{k-1}))^n}{n!} \right) \frac{\mathbf{M}_{s_{k-1}, s_k}^n}{\exp[\mathbf{G}(t_k - t_{k-1})]_{s_{k-1}, s_k}} \end{aligned}$$

since

$$P[Y_{t_k} = s_k \mid Y_{t_{k-1}} = s_{k-1}] = \exp[\mathbf{G}(t_k - t_{k-1})]_{s_{k-1}, s_k}.$$

Therefore, the number of failures $\hat{N}_{t_k} - \hat{N}_{t_{k-1}}$ can be simulated using the distribution (26). If simulation yields $\hat{N}_{t_k} - \hat{N}_{t_{k-1}} = r$, then the r arrival times $\hat{t}_1, \hat{t}_2, \dots, \hat{t}_r$ of \hat{N} over the interval (t_{k-1}, t_k) are simulated by generating r uniform variates over (t_{k-1}, t_k) and ordering them. Now, we know that $Y_{t_{k-1}} = s_{k-1}$ and $Y_{t_k} = s_k$ and the states of hidden Markov process at $\hat{t}_1 \leq \hat{t}_2 \leq \dots \leq \hat{t}_r$ are simulated recursively by using the conditional distributions

$$P[Y_{\hat{t}_j} = s \mid Y_{\hat{t}_{j-1}} = \hat{s}_{j-1}, Y_{t_k} = s_k] = \frac{\mathbf{M}_{\hat{s}_{j-1}, s} \mathbf{M}_{s, s_k}^{r-j}}{\mathbf{M}_{\hat{s}_{j-1}, s_k}^{r-j+1}} \quad (27)$$

for $j = 1, 2, \dots, r$. For $j = 1$, one should set $\hat{t}_{j-1} = \hat{t}_0 = t_{k-1}$ and $\hat{s}_{j-1} = \hat{s}_0 = s_{k-1}$. It also follows from the conditional distribution (27) that $Y_{\hat{t}_r} = Y_{t_k} = s_k$ at the last time point when $j = r$ since \mathbf{M}^0 is the identity matrix.

Stage 3

Having completed Stages 1 and 2, we should now have the entire simulated hidden Markov process, as well as our data, the observed times of the failures. Let $\mathcal{F} = \{Y_t; 0 \leq t \leq t_{n+1}\}$ denote the environmental process generated using the procedure in stage 2. Thus, we can write out our conditional likelihood function of the parameters and then obtain the full conditionals to generate a new set of values for our parameters at each step of the Gibbs sampler. Let τ_i be the total time that the hidden Markov process spends in state i , let n_i be the total number of failures that occur while the process is in state i , and

let r_{ij} be the number of times the process makes a transition from state i to state j . It is clear that τ_i, n_i , and r_{ij} are in \mathcal{F} for all $i, j \in E$. Given data \mathcal{D} and the entire history \mathcal{F} of the Markov process, the conditional likelihood function of the parameters ρ_i 's, λ_i 's and P_{ij} 's is given by

$$\mathcal{L}(\rho, \lambda, \mathbf{P}; \mathcal{F}, \mathcal{D}) \propto \prod_{i \in E} \rho_i^{\sum_{j \in E} r_{ij}} \exp(-\rho_i \tau_i) \lambda_i^{n_i} \exp(-\lambda_i \tau_i) \prod_{j \in E} P_{ij}^{r_{ij}}.$$

Assuming conjugate independent priors for the unknown parameters the full conditional distributions can be easily obtained. More specifically, for a given state $i = 1, \dots, K$, we assume independent gamma priors for ρ_i 's and λ_i 's, denoted as $\rho_i \sim \mathcal{G}(a_i^\rho, b_i^\rho)$, and $\lambda_i \sim \mathcal{G}(a_i^\lambda, b_i^\lambda)$, respectively. For the i -th row of the transition matrix P , we assume a Dirichlet prior, independent of the other rows, as $\mathbf{P}_i \sim \mathcal{Dir}(\alpha_{i1}, \dots, \alpha_{iK})$ where $\mathbf{P}_i = (P_{i1}, \dots, P_{iK})$. Note that in \mathbf{P}_i we have $P_{ii} = 0$ and the corresponding parameter $\alpha_{ii} = 0$. Using standard Bayesian results we can show that given the full history of the hidden Markov process, the full conditional distributions of the parameters can be obtained as

$$\rho_i | \rho_i^{-i} \sim \mathcal{G}(a_i^\rho + \sum_{j \in E} r_{ij}, b_i^\rho + \tau_i) \quad \text{and} \quad \lambda_i | \lambda_i^{-i} \sim \mathcal{G}(a_i^\lambda + n_i, b_i^\lambda + \tau_i)$$

$$\mathbf{P}_i | \mathbf{P}_i^{-i} \sim \mathcal{Dir}(\alpha_{i1} + r_{i1}, \dots, \alpha_{iK} + r_{iK})$$

where $\alpha_{ii} = r_{ii} = 0$.

We then generate new values for these parameters from their posterior distribution and then repeat the whole process again, starting with Stage 1.

4 Determining the Number of States in the Hidden Markov Process

Our analysis in Section 3 assumed that the number of states K in the hidden Markov process was known. However, in general, the actual number of states may be unknown to

us, so it is important to be able to determine how many states there are. The problem of determining K can be considered as a model selection problem in the Bayesian approach where the model choice is made using Bayes factors; see Kass and Raftery (1995) for a review. The computation of the Bayes factors requires the evaluation of marginal likelihood for a given model, that is, for given value of K in our case. More specifically, if we let $\mathcal{D} = \{t_1, \dots, t_n\}$, denote our observed data, we want to obtain the marginal likelihood $p(\mathcal{D}|K)$. The model with the highest value of $p(\mathcal{D}|K)$ is the one most supported by the data and this can be used as the criterion for determining the value of K . Alternatively, assuming a support for K and specifying prior probabilities $P[K = k]$ for different models such that $\sum_k P[K = k] = 1$ we can obtain posterior model probabilities $P[K = k|\mathcal{D}]$ using the marginal likelihood.

Evaluation of the marginal likelihood $p(\mathcal{D}|K)$ analytically is not possible in many problems since it requires integrating out the unknown parameters. Since draws from prior distributions of the parameters result in unstable estimation, the use of Monte Carlo methods emphasize use of posterior Monte Carlo samples to evaluate $p(\mathcal{D}|K)$. Although this is not straightforward in many cases, when the full posterior conditional distributions are known forms, the marginal likelihood terms can be approximated using the approach proposed by Chib (1995). Since the Bayesian analysis of the MMPP in Section 3 is based on known full conditionals, we can adopt Chib's procedure to our problem as will be discussed in the sequel.

In our case, the marginal likelihood for a specific model with dimension K is given by

$$p(\mathcal{D}) = \frac{p(\mathcal{D}|\rho, \lambda, \mathbf{P}, \mathcal{F}) p(\rho, \lambda, \mathbf{P}, \mathcal{F})}{p(\rho, \lambda, \mathbf{P}, \mathcal{F}|\mathcal{D})} \quad (28)$$

where ρ and λ are K -dimensional vectors of ρ_i 's and λ_i 's and \mathbf{P} is the transition probability matrix of dimension K with zeros on the diagonal. We can rewrite (28) as

$$p(\mathcal{D}) = \frac{p(\mathcal{D}|\rho, \lambda, \mathbf{P}, \mathcal{F}) p(\mathcal{F}|\rho, \mathbf{P}) p(\rho, \lambda) p(\mathbf{P})}{p(\rho, \lambda, \mathbf{P}|\mathcal{F}, \mathcal{D}) p(\mathcal{F}|\mathcal{D})} \quad (29)$$

Equation (29) holds for any values of $(\rho, \lambda, \mathbf{P}, \mathcal{F})$ such as $(\rho^*, \lambda^*, \mathbf{P}^*, \mathcal{F}^*)$ which is typically chosen as the mean or mode values of the posterior distributions. We note that all the

terms in the numerator are available to us analytically and therefore can be evaluated at $(\rho^*, \lambda^*, \mathbf{P}^*, \mathcal{F}^*)$. The tricky part to evaluate is the denominator term

$$p(\mathcal{F}^*|\mathcal{D}) = \int p(\mathcal{F}^*|\mathcal{D}, \lambda, \rho, \mathbf{P})p(\lambda, \rho, \mathbf{P}|\mathcal{D})d(\lambda, \rho, \mathbf{P})$$

which can be evaluated using G samples from the posterior distribution $p(\lambda, \rho, \mathbf{P}|\mathcal{D})$ via

$$p(\mathcal{F}^*|\mathcal{D}) = \frac{1}{G} \sum_{g=1}^G p(\mathcal{F}^*|\rho^{(g)}, \lambda^{(g)}, \mathbf{P}^{(g)}, \mathcal{D}). \quad (30)$$

The first term $p(\rho^*, \lambda^*, \mathbf{P}^*|\mathcal{F}^*, \mathcal{D})$ can easily be written down as product of gamma and Dirichlet densities. Thus, for each value of K , we can approximate (29) and determine the model with the highest support of the data. As previously mentioned, using the marginal likelihood we can also compute posterior model probabilities $P[K = k|\mathcal{D}]$ to infer the value of K .

5 Numerical Illustration

In this section we consider Musa’s System I data (see for example, Singpurwalla and Soyer (1992)) to illustrate our model and to discuss what type of additional insights can be obtained from the MMPP model. The data was observed over a period of $t_{\text{obs}} = 90,000$ units, and $n = 136$ failures were observed during this time. A plot of the time between failures is shown in Figure 1 below.

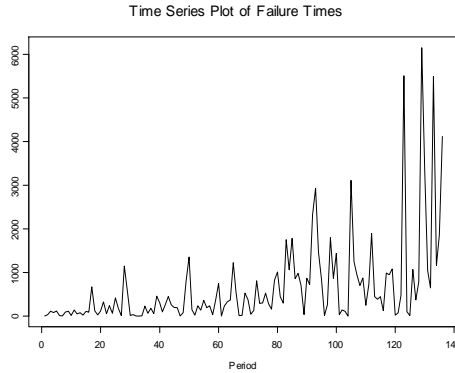


Figure 1: Musa’s System 1 Data

The failure times will be labeled t_1, \dots, t_{136} . All the numbers from Musa's System I data have been divided by 1000. We also set $t_0 = 0$ and $t_{n+1} = t_{137} = t_{\text{obs}} = 90$. So the data we are using is $t_0 = 0, t_1 = 0.003, \dots, t_{136} = 88.682, t_{137} = 90$. In our analysis we will use the standard homogeneous Poisson process model as benchmark in our analysis. Note that the marginal likelihood in this can be easily obtained as a negative binomial model by using the conjugate gamma prior for failure rate λ .

In our development we will present results from the analysis of the two-state MMPP. Note that in this case the transition matrix for the embedded Markov chain of the environmental process is given by

$$\mathbf{P} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

and therefore we only need inference on parameters $\rho_1, \rho_2, \lambda_1$, and λ_2 , as well as on latent states s_1, \dots, s_n . In our analysis we use diffused but proper priors for these parameters by setting $a_i^\lambda = b_i^\lambda = 0.01$ for $i = 1, 2$. Similarly, we use $a_i^\rho = b_i^\rho = 0.01$ for the priors of ρ_1 and ρ_2 . After a small burn-in sample, we collected 1000 simulations to obtain the posterior distributions for our four parameters. We have not witnessed any convergence problems in running the Gibbs sampler. Trace plots for posterior samples from the distributions of λ_1 , and λ_2 are shown in Figures 2 and 3.

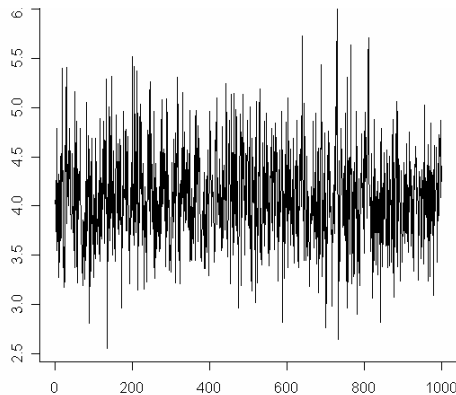


Figure 2: Trace plot of λ_1

The posterior distributions of λ_1 and λ_2 that are shown in Figures 4 and 5 imply that the failure rate in state 1 is higher than the failure rate in state 2. The posterior

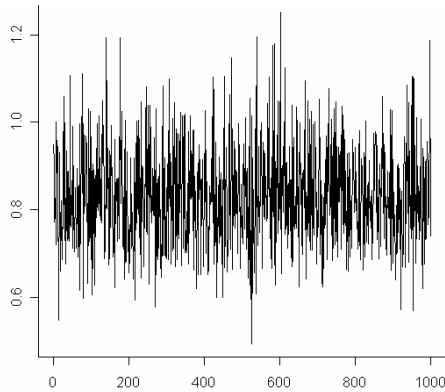


Figure 3: Trace plot of λ_2

distribution of the holding rate ρ_1 in state 1 is given in Figure 6. The posterior mean of ρ_1 is 0.0532 implying that the expected holding time in state 1 is about 18.795 time units. Our analysis has shown that state 2 is acting like an absorbing state, that is, expected holding time is infinite. Similarly, when we look at the posterior distributions of the long-run probabilities π_1 and π_2 of equation (18), we can see that the posterior expected values for π_1 and π_2 are 0.997 and 0.003, respectively. We can also infer this behavior by looking at the posterior distributions of the states at the failure times t_i 's of the software.

Table 2 shows the posterior mean values of the states at each failure time. These values seem to suggest that the hidden Markov process remains in state 1 approximately until between t_{78} and t_{79} . Looking at the data, we can see that $t_{78} = 18.728$ and $t_{79} = 19.556$. Comparing this to the posterior distribution of ρ_1 , the mean of this distribution is 0.0532, which means that the expected holding time in state 1 before changing to state 2 is $1/0.0532 = 18.795$, which also seems to suggest that, on average, the hidden Markov process remains in state 1 until between the 78th and 79th failure. Our simulated posterior density of ρ_2 is of no use to us at all. It basically just looks like a point mass at some number very close to zero. Thus, once the environmental process enters state 2 it stays there forever. This implies that during the debugging process the reliability improvement in the software is initially limited as reflected by shorter failure times and the higher failure rate λ_1 . As the debugging process continues to later periods, failure rate decreases and

the time between failures get larger. Thus, based on this we can conclude that as a result of the debugging process reliability of the software has improved. Figure 7 shows both the data and the posterior means of the states of the hidden Markov process on the same graph.

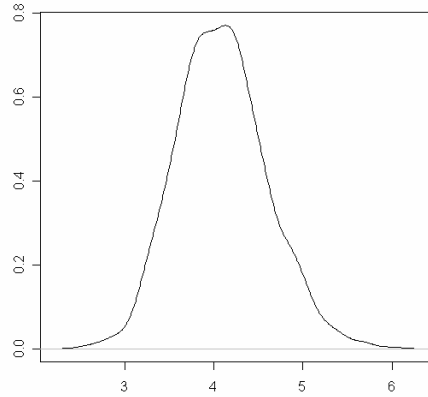


Figure 4: Posterior density of λ_1

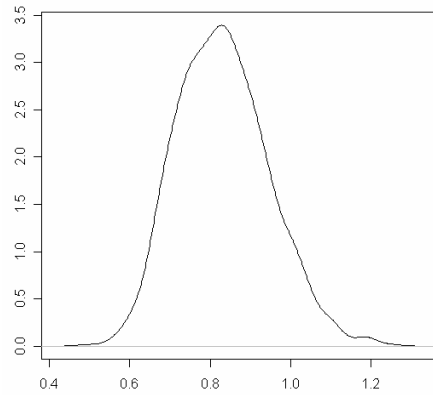


Figure 5: Posterior density of λ_2

Furhermore, using the posterior samples of parameters (ρ, λ) we can obtain the posterior predictive reliability using (15). We note that (15) is a function of (ρ, λ) and therefore we need to evaluate the integral

1	...	57	58	59	60	61	62	63	64
1.000	...	1.000	1.101	1.002	1.002	1.002	1.002	1.005	1.017
65	66	67	68	69	70	71	72	73	74
1.025	1.025	1.025	1.025	1.020	1.036	1.037	1.048	1.142	1.155
75	76	77	78	79	80	81	82	83	84
1.191	1.262	1.308	1.362	1.755	1.885	1.899	1.920	1.997	1.999
85	86	87	88	89	90	91	92	93	94
2.000	2.000	2.000	1.999	1.999	1.999	1.999	2.000	2.000	2.000
95	96	97	98	99	100	101	102	...	136
2.000	2.000	2.000	2.000	2.000	1.999	1.999	2.000	...	2.000

Table 1: Mean value of states

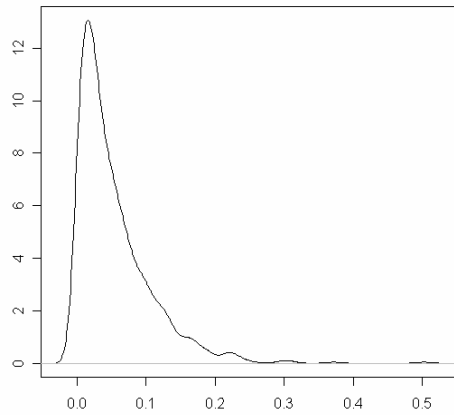


Figure 6: Posterior distribution of ρ_1

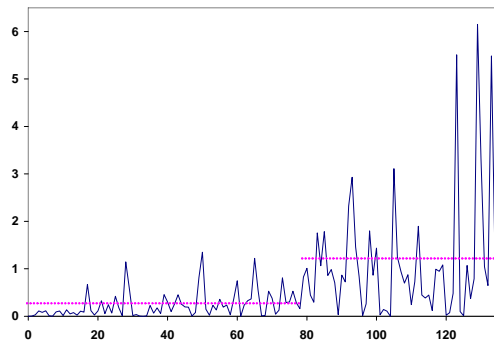


Figure 7: Data with hidden Markov chain when $K = 2$

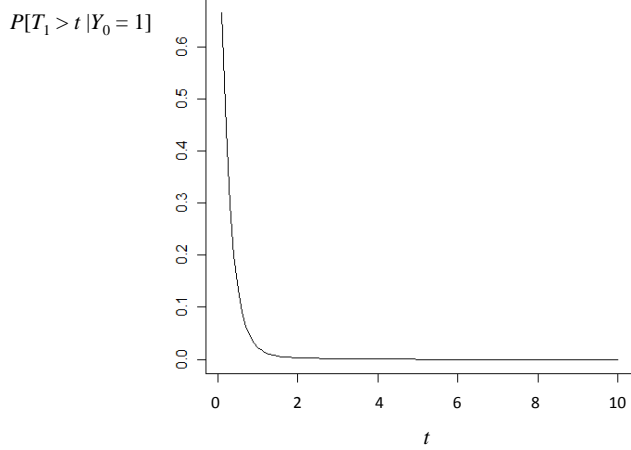


Figure 8: Posterior Predictive Reliability when $Y_0 = 1$.

$$P[T_1 > t | Y_0 = i, \mathcal{D}] = \int P[T_1 > t | Y_0 = i, \rho, \lambda] p(\rho, \lambda | \mathcal{D}) = \int \sum_{j \in E} \exp((G - \Lambda)t)_{ij} p(\rho, \lambda | \mathcal{D}) d\rho d\lambda. \quad (31)$$

The above can be approximated by the Monte Carlo average

$$P[T_1 > t | Y_0 = i, \mathcal{D}] \approx \frac{1}{S} \sum_s \sum_{j \in E} \exp((G^s - \Lambda^s)t)_{ij} \quad (32)$$

using posterior samples $\{\rho, \lambda\}_{s=1}^S$ from $p(\rho, \lambda | \mathcal{D})$. In Figures 8 and 9 we present the posterior predictive reliability functions for $Y_0 = 1$ and $Y_0 = 2$, respectively. Since environmental state 1 has a higher failure rate, if testing starts in state 1, then the posterior predictive reliability goes to 0 around $t = 2000$ time units. On the other hand, if the initial state is state 2, the reliability reaches to 0 around 6000 time units.

We can reach the same conclusion by computing the posterior expected time to failure (17). Again we note that (17) is a function of unknown parameters and it can be

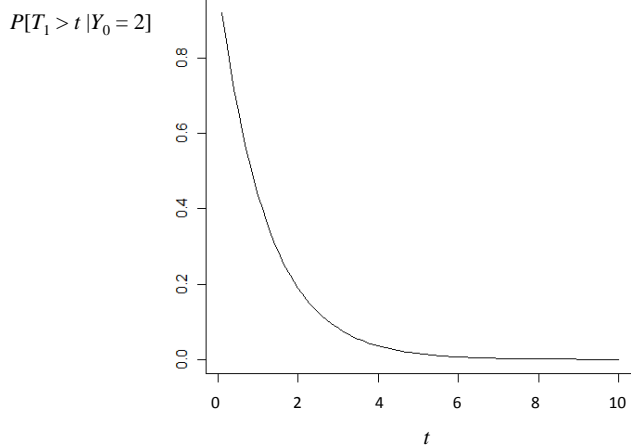


Figure 9: Posterior Predictive Reliability when $Y_0 = 2$.

approximated as a Monte Carlo average given by

$$E[T_1|Y_0 = i, \mathcal{D}] \approx \frac{1}{S} \sum_s \sum_{j \in E} \left(\left[I - P^\lambda \right]_{ij}^{-1} \frac{1}{\rho_j^\lambda} \right)^s \quad (33)$$

using posterior samples $\{\rho, \lambda\}_{s=1}^S$ from $p(\rho, \lambda|\mathcal{D})$. Note that P^λ is given by (8) which also depends on \mathbf{P} and therefore in general one needs to use the posterior samples of \mathbf{P} to evaluate (33). But since \mathbf{P} is known for the two dimensional case, we do not need it in our calculations. It can be shown that the posterior expected time to failure can be obtained as $E[T_1|Y_0 = 1, \mathcal{D}] = 0.83$ and $E[T_1|Y_0 = 2, \mathcal{D}] = 4.15$ when testing starts at states 1 and 2, respectively.

We have repeated a similar analysis for the MMPP with $K = 3$ states. Finally, we computed the marginal likelihoods using approach introduced in Section 4. The comparison of the marginal likelihoods indicate that the model with $K = 2$ is preferred over the others.

6 Concluding Remarks

In this paper we introduced a Markov modulated Poisson process to describe the failure of a software that goes through a debugging process. The model can capture imperfect debugging and allows us to make inferences about the performance of the debugging process at each stage of testing. We presented a Bayesian analysis of the model and illustrated how Bayesian inferences can be made about reliability of the software. Furthermore, we introduced an approach to assess the dimension of the hidden Markov process of the model by using the marginal likelihood. In doing so, we showed how the marginal likelihood can be computed using the approach proposed by Chib (1995). We illustrated the implementation of our approach using actual software failure data and discussed the additional insights that can be obtained from our model.

References

- [1] K. Arifoğlu and S. Özekici, *Optimal policies for inventory systems with finite capacity and partially observed Markov-modulated demand and supply processes*, European Journal of Operational Research **204** (2010), 421–483.
- [2] B. Çekyay and S. Özekici, *Mean time to failure and availability of semi-Markov missions with maximal repair*, European Journal of Operational Research **207** (2010), 1442–1454.
- [3] S. Chib, *Marginal likelihood from the gibbs output*, Journal of the American Statistical Association **90** (1995), 1313–1321.
- [4] J.B. Durand and O. Gaudoin, *Software reliability modelling and prediction with hidden Markov chains*, Statistical Modelling **5** (2005), 75–93.
- [5] P. Fearnhead and C. Sherlock, *An exact Gibbs sampler for the Markov-modulated Poisson process*, Journal of the Royal Statistical Society: Series B **68** (2006), 767–784.

- [6] W. Fischer and K. Meier-Hellstern, *The Markov-modulated Poisson process cookbook*, Performance Evaluation **18** (1992), 149–171.
- [7] A. Goel and K. Okumoto, *Optimal release time for software systems based on reliability and cost criteria*, The Journal of Systems and Software **1** (1980), 315–318.
- [8] R. E. Kass and A. E. Raftery, *Bayes factors*, Journal of the American Statistical Association **90** (1995), 773–795.
- [9] L. Kuo and T.Y. Yang, *Bayesian computation for nonhomogeneous Poisson processes in software reliability*, Journal of the American Statistical Association **91** (1996), 763–773.
- [10] J.D. Musa, *Operational profiles in software reliability engineering*, IEEE Software **10** (1993), 14–32.
- [11] S. Özekici and R. Soyer, *Reliability of software with an operational profile*, European Journal of Operational Research **149** (2003), 459–474.
- [12] ———, *Semi-Markov modulated Poisson process: Probabilistic and statistical analysis*, Mathematical Methods of Operations Research **64** (2006), 125–144.
- [13] N.U. Prabhu and Y. Zhu, *Markov-modulated queueing systems*, Queueing Systems **5** (1989), 215–246.
- [14] S. Ross, *Stochastic processes*, 2. ed., Wiley, New York, 1996.
- [15] N.D. Singpurwalla and R. Soyer, *Nonhomogeneous autoregressive processes for tracking reliability growth, and their bayesian analysis*, Journal of the Royal Statistical Society, Series B **54** (1992), 145–156.
- [16] R. Soyer, *Software reliability*, Wiley Interdisciplinary Reviews: Computational Statistics **3** (2011), 269–281.