2000 Proceedings of the Autumn Conference, Korean Statistical Society

Markovian Perfect Debugging Model and Its Related Measures

Chong Hyung Lee[1], Kyung Hyun Nam[2] and Dong Ho Park[3]

Key Words : Software reliability model; perfect debugging; Markov process; first passage time

Abstract

In this paper we consider a Markovian perfect debugging model for which the software failure is caused by two types of faults, one which is easily detected and the other which is difficult to detect. When a failure occurs, a perfect debugging is immediately performed and consequently one fault is reduced from fault contents. We also treat the debugging time as a variable to develop a new debugging model. Several measures, including the distribution of first passage time to the specified number of removed faults, are also obtained using the proposed debugging model. Numerical examples are provided for illustrative purposes.

1. Introduction

Most of software reliability models derived under NHPP are based on the assumption that whenever a failure occurs, the fault which causes a failure is immediately removed. Such models do not consider the case when it takes a certain length of time for debugging a failure. However, the debugging of failure always takes some time in practice. Therefore, it is more realistic to treat the debugging time as a variable for developing a stochastic model which is applicable in real situations. To derive a stochastic model that can take into account the debugging time of failure, the Markov process has been widely used.

Shooman and Trivedi(1976) propose a software availability model based on a Markov process. Their model assumes that each fault causing the software failure is perfectly debugged. Goel and Okumoto(1979) propose an imperfect debugging model by modifying the one proposed by Shooman and Trivedi(1976). Goel and Soenjoto(1981) propose a model for performance assessment of a hardware-software system. They utilize the concept of imperfect debugging process to derive several measures such as the required time to attain a specified level of software purity, reliability, availability and mean numbers of software and hardware failures up to time t by using the knowledge on the number of remaining faults. Yamada, Tokuno and Osaki(1993) derive a Markovian imperfect reliability model that does not depend on the length of debugging time needed for the removal of faults. They assume that fault contents remain unchanged after debugging, and use the failure rate which was

[1]Department of Statistics, Hallym University, Chunchon, Korea.
[2]Division of Economics, Kyonggi University, Suwon, Korea.
[3]Department of Statistics, Hallym University, Chunchon, Korea.

suggested by Moranda(1975). In this case, the failure rate is geometrically decreased as the number of faults removed is increased. They also determine an optimal software release time satisfying the software reliability requirements and minimizing the expected software cost. Tokuno and Yamada(1998) derive an availability model with two kinds of restoration actions: When a software failure occurs, the restoration action with the debugging activity is performed with probability p, and the restoration action without the debugging activity is performed with probability $1-p$.

In this paper, we derive a perfect debugging model based on a Markov process. For developing such a model, we assume that a software failure is caused by either a fault that is easily detected or a fault that is difficult to detect. The former failure is referred to as 'type 1' and the latter failure is referred to as 'type 2' throughout this paper. As soon as a failure occurs, an action to correct the failure is immediately performed, and only the perfect debugging which reduces one fault from fault contents is disscussed.

In Section 2, we develop a Markovian perfect debugging model and obtain a distribution function of the first passage time to the specified number of removed faults. The expected number of faults removed up to a specified time is also derived. Section 3 provides numerical examples to illustrate our results.

2. Markovian Perfect Debugging Model

Consider a stochastic process $\{X(t), t \geq 0\}$, which represents the total number of removed faults up to time t and the state of software system at time t. The software system is classified as either working or nonworking during its testing and operation period. Nonworking state can be further classified into two types. One type of nonworking state is caused by a fault that is easily detected and the other is caused by a fault that is difficult to detect. The former type is referred to as a fault type 1 and the latter is referred to as a fault type 2. The state of software system at time t is defined as $X(t) = (x_1(t), x_2(t))$, where $x_1(t)$ is a total number of faults removed during a time interval $(0, t]$ and

$$x_2(t) = \begin{cases} 0 & \text{for working} \\ 1 & \text{for nonworking caused by a fault type 1} \\ 2 & \text{for nonworking caused by a fault type 2.} \end{cases}$$

In this model, we assume that the debugging process starts as soon as a failure occurs and that a perfect debugging is performed to remove exactly one fault regardless of the fault type.

We consider a software system which is just returned to the working state after removing the i th fault and let $W_{i,1}$ and $W_{i,2}$ denote the random variables representing the waiting times elapsed for the next occurrence of fault types 1 and 2, respectively. We also assume that $W_{i,1}$ and $W_{i,2}$ follow exponential distributions with means of $1/\mu_{i,1}$ and $1/\mu_{i,2}$, respectively, and they are mutually independent. It follows that

$$\Pr\{W_{i,1} < W_{i,2}\} = \frac{\mu_{i,1}}{\mu_{i,1} + \mu_{i,2}}. \tag{1}$$

Let $T_{i,j}$, $j = 1, 2$, denote the random variables representing the lengths of dubugging time needed to remove a fault of type j from the software system, which had i faults removed previously. We assume that they follow exponential distributions with means of $1/\theta_{i,j}$. It is quite resonable to assume that the debugging time becomes shorter and the length of time that the software is in working state gets longer as the number of removed faults increases. Thus, for given i and j we assume that $\mu_{i,j}$ is a decreasing function of the number of perfect debuggings and $\theta_{i,j}$ is an increasing function of the number of perfect debuggings. These assumptions are more practical, rather than assuming that the rates, $\mu_{i,j}$ and $\theta_{i,j}$, are constant, independent of the number of removed faults. Let $F_{\alpha,\beta}(t)$ be one step transition probability that the process $X(t)$ in state α will be in state β after time t. Using (1), it is easy to obtain

$$F_{(i,0),(i,j)}(t) = \frac{\mu_{i,j}}{\mu_{i,1} + \mu_{i,2}}(1 - \exp(-(\mu_{i,1} + \mu_{i,2})t)) \tag{2}$$

and

$$F_{(i,j),(i+1,0)}(t) = 1 - \exp(-\theta_{i,j}t), \tag{3}$$

where $t \geq 0$ and $j = 1, 2$.

2.1 Distribution of First Time to a Specified Number of Removed Faults

Let $T_{(i,0),(n,0)}$ be a random variable representing the first passage time of the software system with i faults already removed until the number of faults removed reaches n, where $i < n$ and let N be the number of initial faults latent in the system prior to the testing period. We use $G_{(i,0),(n,0)}(t)$ to denote the distribution function of $T_{(i,0),(n,0)}$. Thus, $G_{(i,0),(n,0)}(t)$ can be interpreted as the probability that n faults are removed during $[0,t]$, given that i faults have already been removed at time 0. If a fault of type j, $j = 1, 2$, occurs at u_1, and the perfect debugging takes the length of time u_2, the distribution function of $T_{(i,0),(n,0)}$ has the following relations.

$$\int_0^t \int_0^{t-u_1} G_{(i+1,0),(n,0)}(t - u_1 - u_2)dF_{(i,j),(i+1,0)}(u_2)dF_{(i,0),(i,j)}(u_1) = F_{(i,0),(i,j)} * F_{(i,j),(i+1,0)} * G_{(i+1,0),(n,0)}(t), \tag{4}$$

where * symbolizes a convolution. Using the relation (4), the distribution function of $T_{(i,0),(n,0)}$ can be expressed as

$$\begin{aligned} G_{(i,0),(n,0)}(t) &\equiv \Pr\{T_{(i,0),(n,0)} \leq t\} \\ &= \sum_{j=1}^{2} F_{(i,0),(i,j)} * F_{(i,j),(i+1,0)} * G_{(i+1,0),(n,0)}(t) \end{aligned} \tag{5}$$

for $i = 0, 1, 2, \cdots, n-1$, $n = 1, 2, 3, \cdots, N$, and $G_{(n,0),(n,0)}(t) = 1$. To derive the distribution function of $T_{(i,0),(n,0)}$ explicitly, we may apply the Laplace-Stieljes(L-S) transform. The L-S transform of $G_{(i,0),(n,0)}(t)$ is defined as

$$\tilde{G}_{(i,0),(n,0)}(s) = \int_0^\infty \exp(-st)dG_{(i,0),(n,0)}(t).$$

Using the fact that the L-S transform of a convolution of two functions is equal to the product of the L-S transforms of those functions, the L-S transform of (5) can be written as

$$\tilde{G}_{(i,0),(n,0)}(s) = \sum_{j=1}^{2} \tilde{F}_{(i,0),(i,j)}(s) \times \tilde{F}_{(i,j),(i+1,0)}(s) \times \tilde{G}_{(i+1,0),(n,0)}(s), \tag{6}$$

where $\tilde{F}$ denotes the L-S transform of F. The L-S transforms of (2) and (3) are

$$\tilde{F}_{(i,0),(i,j)}(s) = \frac{\mu_{i,j}}{s+\mu_{i,1}+\mu_{i,2}} \tag{7}$$

and

$$\tilde{F}_{(i,j),(i+1,0)}(s) = \frac{\theta_{i,j}}{s+\theta_{i,j}}, \tag{8}$$

respectively, for $s \geq 0$ and $j = 1, 2$. Substituting (7) and (8) into (6), we have

$$\begin{aligned} \tilde{G}_{(i,0),(n,0)}(s) &= \left[\frac{(\mu_{i,1}\theta_{i,1}+\mu_{i,2}\theta_{i,2})(s+\theta_{i,1}\theta_{i,2}(\mu_{i,1}+\mu_{i,2})/(\mu_{i,1}\theta_{i,1}+\mu_{i,2}\theta_{i,2}))}{(s+\mu_{i,1}+\mu_{i,2})(s+\theta_{i,1})(s+\theta_{i,2})}\right] \\ &\quad \times \tilde{G}_{(i+1,0),(n,0)}(s) \end{aligned} \tag{9}$$

where $i = 0, 1, 2, \cdots, n-1$. To solve the equation (9) explicitly, we consider only the special case when $i = 0$. Solving (9) recursively, we obtain the L-S transform of $G_{(0,0),(n,0)}(t)$ as

$$\begin{aligned} \tilde{G}_{(0,0),(n,0)}(s) &= \prod_{i=0}^{n-1}\left[\frac{(\mu_{i,1}\theta_{i,1}+\mu_{i,2}\theta_{i,2})(s+\theta_{i,1}\theta_{i,2}(\mu_{i,1}+\mu_{i,2})/(\mu_{i,1}\theta_{i,1}+\mu_{i,2}\theta_{i,2}))}{(s+\mu_{i,1}+\mu_{i,2})(s+\theta_{i,1})(s+\theta_{i,2})}\right] \\ &= \sum_{i=0}^{n-1}\left[\frac{N_{n,i,1}(\mu_{i,1}+\mu_{i,2})}{s+\mu_{i,1}+\mu_{i,2}} + \frac{N_{n,i,2}\theta_{i,1}}{s+\theta_{i,1}} + \frac{N_{n,i,3}\theta_{i,2}}{s+\theta_{i,2}}\right]. \end{aligned} \tag{10}$$

Denote $x_i = \mu_{i,1} + \mu_{i,2}$. Then, the values of N's, given in (10), are defined as

$$N_{n,i,1} = \frac{\prod_{m=0}^{n-1}\left[(\mu_{m,1}\theta_{m,1}+\mu_{m,2}\theta_{m,2})((\theta_{m,1}\theta_{m,2}x_m/(\mu_{m,1}\theta_{m,1}+\mu_{m,2}\theta_{m,2}))-x_i)\right]}{x_i\prod_{\substack{m=0\\m\neq i}}^{n-1}(x_m-x_i)\prod_{m=0}^{n-1}\left[(\theta_{m,1}-x_i)(\theta_{m,2}-x_i)\right]},$$

$$N_{n,i,2} = \frac{\prod_{m=0}^{n-1}\left[(\mu_{m,1}\theta_{m,1}+\mu_{m,2}\theta_{m,2})((\theta_{m,1}\theta_{m,2}x_m/(\mu_{m,1}\theta_{m,1}+\mu_{m,2}\theta_{m,2}))-\theta_{i,1})\right]}{\theta_{i,1}\prod_{\substack{m=0\\m\neq i}}^{n-1}(\theta_{m,1}-\theta_{i,1})\prod_{m=0}^{n-1}\left[(x_m-\theta_{i,1})(\theta_{m,2}-\theta_{i,1})\right]},$$

and

$$N_{n,i,3} = \frac{\prod_{m=0}^{n-1}\left[(\mu_{m,1}\theta_{m,1}+\mu_{m,2}\theta_{m,2})((\theta_{m,1}\theta_{m,2}x_m/(\mu_{m,1}\theta_{m,1}+\mu_{m,2}\theta_{m,2}))-\theta_{i,2})\right]}{\theta_{i,2}\prod_{\substack{m=0\\ m\neq i}}^{n-1}(\theta_{m,2}-\theta_{i,2})\prod_{m=0}^{n-1}\left[(x_m-\theta_{i,2})(\theta_{m,1}-\theta_{i,2})\right]},$$

where $\prod_{\substack{m=0\\ m\neq i}}^{0}\cdot = 1$. It is noted that $\sum_{i=0}^{n-1}[N_{n,i,1}+N_{n,i,2}+N_{n,i,3}] = 1$ for $n = 1, 2, 3, \cdots, N$. By applying the inverse transform to (10), the distribution function of $T_{(0,0),(n,0)}$ can be derived as

$$G_{(0,0),(n,0)}(t) = 1 - \sum_{i=0}^{n-1}\left[N_{n,i,1}\exp(-x_i t) + N_{n,i,2}\exp(-\theta_{i,1}t) + N_{n,i,3}\exp(-\theta_{i,2}t)\right] \quad (11)$$

for $t \geq 0$. In case when $n = 0$, $G_{(0,0),(0,0)}(t) = 1$.

The expectation and variance of $T_{(0,0),(n,0)}$ are obtained as

$$\mathrm{E}(T_{(0,0),(n,0)}) = \sum_{i=0}^{n-1}\left[(N_{n,i,1}/x_i) + (N_{n,i,2}/\theta_{i,1}) + (N_{n,i,3}/\theta_{i,2})\right]$$

and

$$\begin{aligned}\mathrm{Var}(T_{(0,0),(n,0)}) &= 2\sum_{i=0}^{n-1}\left[(N_{n,i,1}/x_i^2) + (N_{n,i,2}/\theta_{i,1}^2) + (N_{n,i,3}/\theta_{i,2}^2)\right] \\ &\quad - \left\{\sum_{i=0}^{n-1}\left[(N_{n,i,1}/x_i) + (N_{n,i,2}/\theta_{i,1}) + (N_{n,i,3}/\theta_{i,2})\right]\right\}^2,\end{aligned}$$

respectively.

2.2 Expected Number of Faults Removed up to a Specified Time

Let $Y(t)$ be a random variable representing the number of faults removed during $(0,t]$ and N be known. We assume that the software system had already removed i faults up to time 0 and is currently in working state. Let $\mathrm{EN}_{(i,0)}(t)$ denote the expected number of faults removed up to time t for such a software. Then, it can be written as

$$\mathrm{EN}_{(i,0)}(t) = \mathrm{E}[Y(t)|X(0) = (i,0)] \quad (12)$$

for $i = 0, 1, 2, \cdots, N$. Let $U_{(i,j),(i+1,0)}$ represent the length of transition time of the software state from (i,j) to $(i+1,0)$. Then, (12) can be rewritten as

$$\mathrm{EN}_{(i,0)}(t) = \sum_{j=1}^{2}\int_0^t\int_0^{t-u_1}\mathrm{E}[Y(t-u_1)|U_{(i,j),(i+1,0)} = u_2]dF_{(i,j),(i+1,0)}(u_2)dF_{(i,0),(i,j)}(u_1)$$

$$= \sum_{j=1}^{2} \Big[F_{(i,0),(i,j)} * F_{(i,j),(i+1,0)}(t) + F_{(i,0),(i,j)} * F_{(i,j),(i+1,0)} * \mathrm{EN}_{(i+1,0)}(t) \Big] \qquad (13)$$

for $i = 0, 1, 2, \cdots, N-1$ and $\mathrm{EN}_{(N,0)}(t) = 0$. Applying the L-S transform to both sides of (13), we have

$$\widetilde{\mathrm{EN}}_{(i,0)}(s) = \sum_{j=1}^{2} \Big[\widetilde{F}_{(i,0),(i,j)}(s) \times \widetilde{F}_{(i,j),(i+1,0)}(s) + \widetilde{F}_{(i,0),(i,j)}(s) \times \widetilde{F}_{(i,j),(i+1,0)}(s) \times \widetilde{\mathrm{EN}}_{(i+1,0)}(s) \Big], \qquad (14)$$

where $\widetilde{\mathrm{EN}}_{(i,0)}(s)$ denotes the L-S transform of $\mathrm{EN}_{(i,0)}(t)$. Substituting (2) and (3) into (14), we obtain

$$\widetilde{\mathrm{EN}}_{(i,0)}(s) = \frac{(\mu_{i,1}\theta_{i,1} + \mu_{i,2}\theta_{i,2})(s + \theta_{i,1}\theta_{i,2}(\mu_{i,1} + \mu_{i,2})/(\mu_{i,1}\theta_{i,1} + \mu_{i,2}\theta_{i,2}))}{(s + \mu_{i,1} + \mu_{i,2})(s + \theta_{i,1})(s + \theta_{i,2})} \times \Big[1 + \widetilde{\mathrm{EN}}_{(i+1,0)}(s) \Big]. \qquad (15)$$

It is in general extremely difficult to obtain the explicit expressions for $\mathrm{EN}_{(i,0)}(t)$ from (15) by applying the inverse L-S transform. Thus, as a special case we consider only the case when $i = 0$. If (15) is solved recursively, the L-S transform of $\mathrm{EN}_{(0,0)}(t)$ is obtained as

$$\widetilde{\mathrm{EN}}_{(0,0)}(s) = \sum_{n=1}^{N} \widetilde{G}_{(0,0),(n,0)}(s). \qquad (16)$$

Applying the inverse L-S transform to (16), we have

$$\mathrm{EN}_{(0,0)}(t) = \sum_{n=1}^{N} G_{(0,0),(n,0)}(t). \qquad (17)$$

3. Numerical Examples

In this section we compare the patterns of $G_{(0,0),(n,0)}(t)$'s and $\mathrm{EN}_{(0,0)}(t)$ and obtain the software reliability for various choice of n. When the i th fault is removed as the result of perfect debugging, the rates, $\mu_{i,j}$ and $\theta_{i,j}$, are defined as follows. Such definition of $\mu_{i,j}$ was suggested by Moranda(1975), and its form is given by

$$\mu_{i,j} = \mu_{0,j} k_j^i,$$

where $\mu_{0,j} > 0$ and $0 < k_j < 1$ for $j = 1, 2$. This implies that the failure rate is geometrically decreased as the number of fault removals increases. Similarly, we define $\theta_{i,j}$ as

$$\theta_{i,j} = \theta_{0,j} \Big[1 + [1 - \exp(-\sqrt{i})] l_j \Big],$$

where $\theta_{0,j} > 0$ and $l_j > 0$. The l_j is a learning factor, which shortens the elngth of debugging times as the number of fault removals increases.

We assume that the the mean time to remove a fault of type 1 is smaller than the mean time to remove a fault of type 2. Set $N = 10$, $\mu_{0,1} = 0.15$, $\mu_{0,2} = 0.1$, $\theta_{0,1} = 0.9$, $\theta_{0,2} = 0.5$, $l_1 = 0.1$, $l_2 = 0.05$, $k_1 = 0.8$ and $k_2 = 0.8$. FIG. I shows the graphical behaviors of $G_{(0,0),(n,0)}(t)$ for $n = 4, 6, 8, 10$. The figure shows that the greater the required number of removed faults is, the longer it takes to reach such a goal. FIG. II shows the graphical behavior of $\mathrm{EN}_{(0,0)}(t)$.

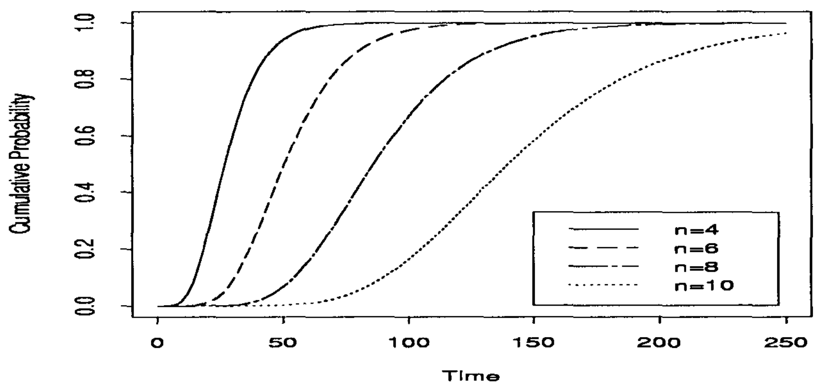

FIG. I. Distribution functions of $T_{(0,0),(n,0)}$ for various n with $\mu_{0,1} = 0.15$, $\mu_{0,2} = 0.1$, $\theta_{0,1} = 0.9$, $\theta_{0,2} = 0.5$, $l_1 = 0.1$, $l_2 = 0.05$, $k_1 = 0.8$ and $k_2 = 0.8$.

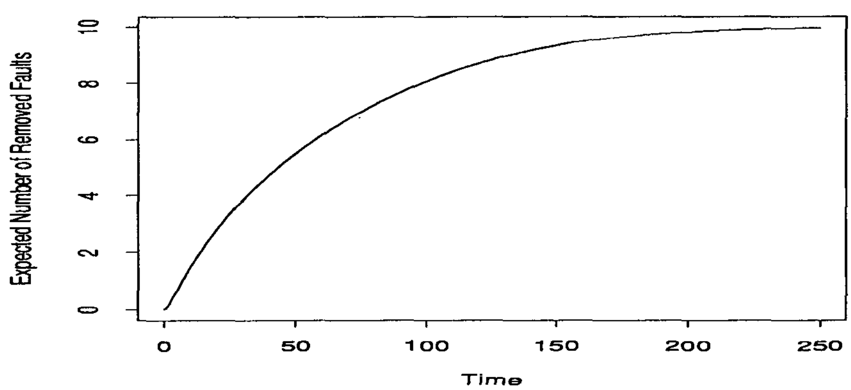

FIG II. Expected number of removed faults $\mathrm{EN}_{(0,0)}(t)$ with $N = 10$, $\mu_{0,1} = 0.15$, $\mu_{0,2} = 0.10$, $\theta_{0,1} = 0.9$, $\theta_{0,2} = 0.5$, $l_1 = 0.1$, $l_2 = 0.05$, $k_1 = 0.8$ and $k_2 = 0.8$.

References

1. Goel, A.L. and Okumoto, K. (1979). "A Makovian Model for Reliability and Other Performance Measures of Software System", *Proceedings National Computer Conference*, 769-774.

2. Goel, A. L. and Soenjoto, J. (1981). "Models for Hardware-Software Operational-Performance Evaluation", *IEEE Transactions on Reliability*, 30, 232-239.

3. Moranda, P. B. (1975). "Prediction of Software Reliability during Debugging", *Proc. 1975 Annual Reliability and Maintainability Symposium*, 327-332.

4. Shooman, M. L. and Trivedi, A. K. (1976). "A Many-State Markov Model for Computer Software Performance Parameters", *IEEE Transactions on Reliability*, R-25, 66-68.

5. Tokuno, K. and Yamada, S. (1998). "Operational Software Availability Measurement with Two Kinds of Restoration Actions", *Journal of Quality in Maintenance Engineering*, 4, 273-283.

6. Yamada, S., Tokuno, K. and Osaki, S. (1993). "Software Reliability Measurement in Imperfect Debugging Environment and Its Application", *Reliability Engineering and System Safety*, 40, 139-147.