

절단고정시간에 근거한 파레토 NHPP 소프트웨어 신뢰성장모형에 관한 비교 연구

김희철* · 신현철**

요 약

소프트웨어 시스템의 대규모자료의 적용 때문에 소프트웨어 신뢰도는 소프트웨어 개발에 중요한 역할을 해왔다. 본 연구에서는 고장시간에 관련된 소프트웨어 신뢰성장모형이 제안되었다. 이러한 검사시간은 미리 정해진 절단 고정 시간을 의미한다. 본 연구에서는 소프트웨어의 강도함수, 평균값 함수 및 신뢰도와 모수추정에 대하여 나열하고 파레토 분포를 수명분포로 적용한 비동질적인 포아송 과정을 적용하였다. 본 논문의 수치적인 예에서는 고장 간격 시간 자료를 적용하고 모수추정 방법은 최우 추정 법을 이용하고 추세분석을 통하여 자료의 효율성을 입증한 후 평균자승오차와 R_{sq} (결정계수)를 이용하고 예측 값과 실제 값의 차이에 의존한 효율적인 모형을 선택 비교하였다.

The Comparative Study for NHPP of Truncated Pareto Software Reliability Growth Model

Kim Hee Cheul* · Shin Hyun Cheul**

ABSTRACT

Due to the large scale application of software systems, software reliability plays an important role in software developments. In this paper, a software reliability growth model (SRGM) is proposed for testing time. The testing time on the right is truncated in this model. The intensity function, mean-value function, reliability of the software, estimation of parameters and the special applications of Pareto NHPP model are discussed.

This paper, a numerical example of applying using time between failures and parameter estimation using maximum likelihood estimation method, after the efficiency of the data through trend analysis model selection, depended on difference between predictions and actual values, were efficient using the mean square error and R_{sq}

Key words : Software Reliability, Non-Homogeneous Poisson Process, Time Truncated Model.

1. 서 론

소프트웨어 고장으로 인한 컴퓨터 시스템의 고장은 우리 사회에 엄청난 손실을 유발 할 수 도 있다. 따라서 소프트웨어 개발 과정에서 소프트웨어 신뢰성은 중요한 문제이다. 이 문제는 사용자의 요구조건과 테스트 비용을 만족시켜야 한다. 소프트웨어 테스트(디버깅)면에서 비용을 줄이기 위해서는 소프트웨어의 신뢰성의 변동과 테스트 비용을 사전에 알고 있어야 효율적이다. 따라서 신뢰도 모형에 의해 소프트웨어 고장 수, 소프트웨어 고장발생간격시간, 소프트웨어 신뢰도 및 고장률 등의 신뢰성 평가측도들이 추정되어 미래의 고장시간을 예측할 수도 있다.

소프트웨어 고장시간은 수명자료가 된다. 따라서 비음(Non-negative)의 값을 가지기 때문에 이 분야에서는 주로 지수분포, 와이블분포, 감마분포 등 일반화 감마 분포가 많이 사용되어 왔다[1, 2]. 이 중에서 비동질적 포아송 과정(NHPP)에 의존한 모형은 에러 탐색 과정측면에서는 우수한 모형이고 이모형은 결함이 발생하면 즉시 제거되고 디버깅 과정에서 새로운 결함이 발생되지 않는다는 가정을 하고 있다.

이 분야에서, Huang[3]은 일반화 로지스틱 테스트 노력함수(Generalized logistic testing-effort function)와 변환점 모수(Change-point parameter)를 통합하여 효율적인 소프트웨어 신뢰성 예측되는 기술을 제시하기도 하였다. 그리고 비교적 최근에는 S-형태 모형은 소프트웨어 관리자들이 소프트웨어 및 검사 도구에 익숙해지는 학습 과정을 설명할 수 있다고 하였다[4].

다른 측면으로 이러한 NHPP모형들은 고장시간에 의존하기 때문에 시간소비과정(Time consuming process[5, 6])으로 알려져 있다.

따라서 본 연구는 시간소비과정에 의존한 무한고장을 가진 소프트웨어 신뢰 성장모형(SRGM, Software reliability growth model)에 대하여 수명분포로는 다소 극단적인 값들이 발생할 확률을 무시할 수 없는 파레토(Pareto distribution)[7] 분포를 사용한 NHPP에 대하여 비교 제시 하였다.

2. 관련 연구

2.1 무한 고장 NHPP와 기록값 통계량

NHPP 모형에서 평균값 함수 $m(t)$ (Mean value function)와 강도 함수(Intensity function) $\lambda(t)$ 는 다음과 같은 관계로 표현할 수 있다[1, 2].

$$m(t) = \int_0^t \lambda(s)ds, \quad \frac{dm(t)}{dt} = \lambda(t) \quad (1)$$

따라서 $N(t)$ 는 모수 $m(t)$ 을 가진 포아송 확률 밀도 함수(Probability density function; Pdf)로 알려져 있다. 즉,

$$P\{N(t) = n\} = \frac{[m(t)]^n \cdot e^{-m(t)}}{n!}, \quad n = 0, 1, 2, \dots, \infty \quad (2)$$

이처럼 시간 관련 모형(Time domain models) 들은 NHPP에 의해서 확률 고장 과정으로 설명이 가능하다. 이러한 모형들은 고장 강도 함수 $\lambda(t)$ 가 다르게 표현됨으로서 평균값 함수 $m(t)$ 도 역시 다르게 나타나고 이러한 NHPP 모형들은 유한 고장 모형과 무한 고장 범주로 분류한다[7]. 유한 고장 NHPP 모형들은 충분한 테스트 시간이 주어지면 결함들(Faults)의 기대값이 유한 값($\lim_{t \rightarrow \infty} m(t) = a < \infty$)을 가지고 반면에 무한 고장 NHPP 모형들은 무한 값을 가진다고 가정된다[8].

무한 고장 NHPP 모형의 평균값 함수와 고장 강도 함수는 각각 다음과 같이 알려져 있다.

$$m(t) = -\ln(1-F(t)) \quad (3)$$

$$\lambda(t) = m'(t) = f(t)/(1-F(t)) = h(t) \quad (4)$$

단, $F(t)$ 는 누적 분포함수이고 $f(t)$ 는 확률밀도함수, 그리고 $h(t)$ 는 위험함수를 의미한다.

시간 $(0, t]$ 까지 조사하기 위한 시간 절단(Time truncated) 모형은 n 번째 까지 고장 시점 자료를

$$x_k = \sum_{i=1}^k t_k \quad (k=1, 2, \dots, n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n) \quad (5)$$

이라고 하고 θ 을 모수공간이라고 하면 무한고장의 우도 함수는 다음과 같이 알려져 있다[8].

$$\begin{aligned} L_{NHPP_{NFR}}(\theta | \underline{x}) &= \prod_{i=1}^n \left(\frac{f(x_i)}{1-F(x_i)} \right) (1-F(x_n)) \quad (6) \\ &= \left(\prod_{i=1}^n h(x_i) \right) (1-F(x_n)) \\ &= \left(\prod_{i=1}^n \lambda(x_i) \right) e^{-m(t)} \end{aligned}$$

2.2 기존모형인 절단된 로그 포아송 실행 시간 SRGM

로그 포아송 실행시간(Log Poission execution time) 모형[9, 10]은 1984년에 Musa와 Okumoto에 의해서 소개된 무한 고장 소프트웨어 모형으로 평균값함수와 강도함수는 다음과 같이 알려져 있다.

$$m(t) = \frac{1}{\theta} \ln(\lambda_0 \theta t + 1), \quad 0 < t < T \quad (7)$$

$$\lambda(t) = \frac{\lambda_0}{\lambda_0 \theta t + 1}, \quad 0 < t < T \quad (8)$$

단, T 는 고정시간(Fixed time)이고 $\lambda_0 (> 0)$ 는 소프트웨어 내의 초기고장 강도, $\theta (> 0)$ 는 초기고장 강도 감소 모수를 의미한다.

한편, (7)식과(8)식을 이용하면 우도함수는 다음과 같이 유도 할 수 있다.

$$L(\theta, \lambda_0 | \underline{x}) = \left(\prod_{i=1}^n \frac{\lambda_0}{\lambda_0 \theta x_i + 1} \right) \cdot e^{-\frac{1}{\theta} \ln(\lambda_0 \theta x_n + 1)} \quad (9)$$

단, $\underline{x} = (x_1, x_2, x_3, \dots, x_n)$.

모수 최우 추정 법(MLE)을 이용하기 위한 로그 우도 함수는 다음과 같이 유도된다.

$$\begin{aligned} \ln L(\theta, \lambda_0 | \underline{x}) & \quad (11) \\ &= n \ln \lambda_0 - \ln \prod_{i=1}^n (\lambda_0 \theta x_i + 1) - \frac{1}{\theta} \ln(\lambda_0 \theta x_n + 1) \end{aligned}$$

(11)식을 이용하여 최우추정치 $\hat{\theta}_{MLE}$ 와 $\hat{\lambda}_{0MLE}$ 은 다음과 같이 구할 수 있다고 하였다[9]

$$\hat{\theta}_{MLE} = \frac{1}{n} \ln(\hat{\phi} x_n + 1) \quad (12)$$

$$\hat{\lambda}_{0MLE} = \hat{\phi} / \hat{\theta}_{MLE} \quad (13)$$

단, $\phi (= \hat{\lambda}_{0MLE} \cdot \hat{\theta}_{MLE})$ 는 (14)식의 근이 된다. 즉, 이 ϕ 근을 구하기 위해서는 수치 해석적 방법으로 다음과 같은 식을 이용하여 계산할 수 있다.

$$\begin{aligned} \frac{\partial \ln L(\phi | \underline{x})}{\partial \phi} &= \quad (14) \\ \frac{n}{\phi} - \sum_{i=1}^n \frac{x_i}{\phi x_i + 1} - \frac{n x_n}{(\phi x_n + 1) \ln(\phi x_n + 1)} &= 0 \end{aligned}$$

NHPP 모형에서 테스트 시점 x_n (마지막 고장시점)에서 소프트웨어 고장이 일어난다고 하는 가정 하에서 신뢰구간 $(x_n, x_n + x]$ (단, x 는 임무시간(Mission time))동안 소프트웨어의 고장이 일어나지 않을 확률인 신뢰도 $\hat{R}(x | x_n)$ 는 다음과 같이 됨이 알려져 있다[7, 11].

$$\begin{aligned} \hat{R}(x | x_n) &= \exp\left(-\int_{x_n}^{x_n+x} \lambda(\tau) d\tau\right) \quad (15) \\ &= \exp\{-[m(x+x_n) - m(x_n)]\} \end{aligned}$$

따라서 로그 포아송 실행시간모형에 대한 신뢰도는 평균값함수 (15)식과 $t = x_n$ 을 이용하면 다음과 같이 표현된다[1, 7].

$$\begin{aligned} R(x | t) &= \quad (16) \\ \exp\left(-\frac{1}{\theta} [\ln(\lambda_0 \theta (x+t) + 1) - \ln(\lambda_0 \theta t + 1)]\right) \end{aligned}$$

2.3 관측 자료에 대한 모형비교

최근에 모형에 대한 효율성을 조사하기 위한 기준으로서 MSE(평균자승오차)와 Rsq (결정계수)를 사용한다[4].

2.3.1 평균자승오차(Mean square error)

평균자승오차는 실제 관찰 값과 예측 값에 대한 차이를 측정하는 도구로서 다음과 같이 정의 된다.

$$MSE = \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{n-k} \quad (17)$$

단, $m(x_i)$ 은 시간(0, x_i]까지 나타난 에러들의 누적 함수를 의미하고 $\hat{m}(x_i)$ 는 x_i 시점까지 평균값 함수로부터 추정된 에러의 누적개수를 의미한다. 그리고 n 은 관찰 값의 수이고 k 는 모수의 수를 의미한다.

2.3.2 Rsq

Rsq(결정계수)는 관찰 값의 차이에 대한 설명력을 나타내는 도구로서 다음과 같이 정의 된다.

$$Rsq = 1 - \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{\sum_{i=1}^n \left(m(x_i) - \frac{\sum_{j=1}^n m(x_j)}{n} \right)^2} \quad (18)$$

3. 제안한 절단된 파레토 실행 시간 SRGM

다소 극단적인 값들이 발생할 확률을 무시할 수 없는 파레토 분포 $Pateto(\alpha, \beta)$ 일 경우 확률 밀도 함수와 분포 함수는 각각 다음과 같다[7].

$$f_{Preto}(t | \alpha, \beta) = \frac{\alpha \beta^\alpha}{(\beta+t)^{(1+\alpha)}} \quad (\alpha, \beta > 0, t > 0) \quad (19)$$

$$F_{Preto}(t | \alpha, \beta) = 1 - \left(\frac{\beta^\alpha}{(\beta+t)^\alpha} \right) \quad (20)$$

(19)식과 (20)식을 이용하면 강도함수 $\lambda(t)$ 와 평균값함수 $m(t)$ 는 다음과 같이 표현된다.

$$e^{-m(t)} = 1 - F(t) = \frac{\beta^\alpha}{(\beta+t)^\alpha}, 0 < t < T \quad (21)$$

$$\lambda(t) = f(t)/(1 - F(t)) = h(t) = \frac{\alpha}{\beta+t}, 0 < t < T \quad (22)$$

단, $\alpha(>0)$ 는 형상모수(Shape parameter)이고

$\beta(>0)$ 는 척도 모수(Scale parameter), T 는 고정시간(Fixed time)을 의미한다.

(6)식과 관련하면 무한 NHPP 고장 우도 함수는 다음과 같다[11].

$$L_{Pretoi}(\alpha, \beta | \underline{x}) = \left(\prod_{i=1}^n \frac{\alpha}{\beta+x_i} \right) \left(\frac{\beta^\alpha}{(x_n + \beta)^\alpha} \right) \quad (23)$$

최우추정법(MLE)을 이용하기 위한 $Pateto(\alpha, \beta)$ 모형 로그 우도 함수는 (23)식과 관련하여 다음과 같이 유도된다.

$$\ln L(\alpha, \beta | D_{x_n}) = n \ln \alpha - \sum_{i=1}^n \ln(\beta+x_i) + \alpha \ln \beta - \alpha \ln(x_n + \beta) \quad (24)$$

(24)식에서 α 와 β 에 대하여 편미분 하여 다음과 같은 식을 만족하는 $\hat{\alpha}_{MLE}$ 와 $\hat{\beta}_{MLE}$ 을 수치 해석적 방법으로 계산할 수 있다[7].

$$\frac{\partial \ln L(\alpha, \beta | \underline{x})}{\partial \alpha} = \frac{n}{\alpha} + \ln \beta - \ln(x_n + \beta) = 0 \quad (25)$$

$$\frac{\partial \ln L(\alpha, \beta | \underline{x})}{\partial \beta} = \frac{\alpha}{\beta} - \sum_{i=1}^n \frac{1}{(\beta+x_i)} - \frac{\alpha}{(x_n + \beta)} = 0 \quad (26)$$

따라서 파레토 실행시간모형에 대한 신뢰도는 평균값함수 (15)식과 $t = x_n$ 을 이용하면 다음과 같이 표현된다[5,6].

$$R(x | t) = \exp(M) \quad (27)$$

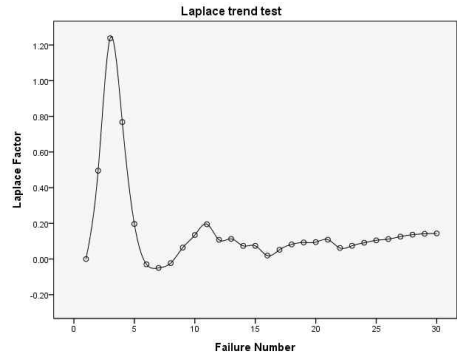
$$\text{단, } M = -\ln \frac{\beta^\alpha}{(x_n + x + \beta)^\alpha} + \ln \frac{\beta^\alpha}{(x_n + \beta)^\alpha}$$

4. 소프트웨어 고장 자료 분석

이 장에서 소프트웨어 고장 간격 시간 자료[12](Failure interval time data)를 가지고 제시하는 신뢰

모형들을 분석하고자 한다. 이 자료의 고장 시간은 1 8.735 시간단위에 30번의 고장이 발생한 자료이며 <표 1>에 나열 되어 있다. 본 연구에서는 고장수가 1 5, 20과 30인 고정시간에 대한 위험함수와 평균값 함수 그리고 신뢰도를 비교하고자 한다.

또한 제시하는 신뢰 모형들을 분석하기 위하여 우선 자료에 대한 추세 검정이 선행 되어야 한다[13]. 추세 분석에는 일반적으로 라플라스 추세 검정(Laplace trend test)을 사용한다. 이 검정을 실시한 결과 (그림 1)에서 라플라스 추세 검정의 결과는 라플라스 요인(Factor)이 -2와 2사이에 존재함으로써 신뢰성장 (Reliability growth) 속성을 나타내고 있다. 따라서 이 자료를 이용하여 신뢰 성장모형을 제시하는 것이 효율적임을 시사하고 있다[13].



(그림 1) 라플라스 추세 검정

<표 1> 고장 자료

Failure Number	Failure Interval (second)	Failure Time(second)
1	0.479	0.479
2	0.266	0.745
3	0.277	1.022
4	0.554	1.576
5	1.034	2.610
6	0.949	3.559
7	0.693	4.252
8	0.597	4.849
9	0.117	4.966
10	0.170	5.136
11	0.117	5.253
12	1.274	6.527
13	0.469	6.996
14	1.174	8.170
15	0.693	8.863
16	1.908	10.771
17	0.135	10.906
18	0.277	11.183
19	0.596	11.779
20	0.757	12.536
21	0.437	12.973
22	2.230	15.203
23	0.437	15.640
24	0.340	15.980
25	0.405	16.385
26	0.575	16.960
27	0.277	17.237
28	0.363	17.600
29	0.522	18.122
30	0.613	18.735

<표 2> 절단고정시간에 의존한 각 모형의 모수 추정값

Failure number (Fixed time T)	MLE	
	LPET	PTET
15 (8.863×10^{-1})	$\hat{\theta} = 0.0894,$ $\hat{\lambda}_0 = 18.1110$	$\hat{\beta} = 0.0618,$ $\hat{\alpha} = 1.6860$
20 (12.536×10^{-1})	$\hat{\theta} = 0.0849,$ $\hat{\lambda}_0 = 17.3881$	$\hat{\beta} = 0.0678,$ $\hat{\alpha} = 2.3912$
30 (18.735×10^{-1})	$\hat{\theta} = 0.0471,$ $\hat{\lambda}_0 = 17.1984$	$\hat{\beta} = 0.0689,$ $\hat{\alpha} = 3.6298$

단, LPET: log Poisson execution time model
PTET: Pareto execution time model

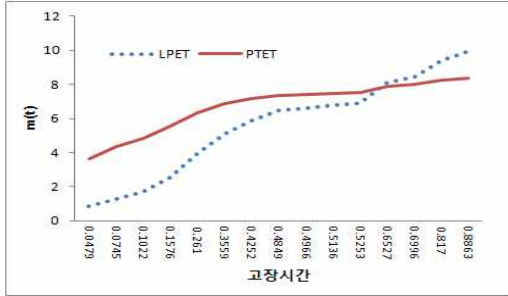
모수 추정에 대해서는 근에 대한 수렴을 보다 용이하게 하기 위하여 <표 1>의 자료를 10^{-1} 만큼 변수 변환 하여 최우 추정법을 이용하고 비선형 방정식의 계산방법은 수치 해석적 기본 방법인 이분법(Bisection method)을 사용하였다. 이러한 계산은 초기 값을 0.01 과 20을, 허용 한계(Tolerance for width of interval)는 10^{-5} 을 주고 수렴성을 확인 하면서 충분한 반복 횟수인 100번을 C-언어를 이용하여 모수 추정을 수행하였다.

<표 3> 절단고정시간에 의존한 각 모형의 MSE와 Rsq 값

Failure number (Fixed time T)	MSE		Rsq	
	LPET	PTET	LPET	PTET
15 (8.863×10^{-1})	8.353	8.024	0.965	0.968
20 (12.536×10^{-1})	17.041	14.739	0.981	0.983
30 (18.735×10^{-1})	25.265	33.257	0.983	0.848

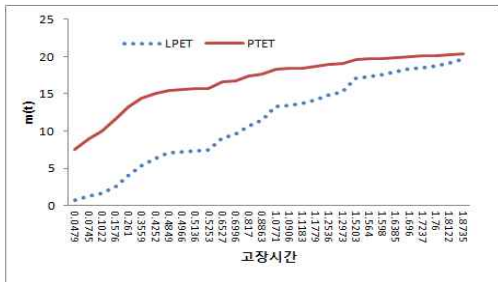
그 결과는 <표 2>에 요약되었다. 그리고 <표 3>에는 모형에 대한 효율성을 조사하기 위한 기준으로

서 MSE (평균제곱오차)와 R^2 (결정계수)도 요약되었다.



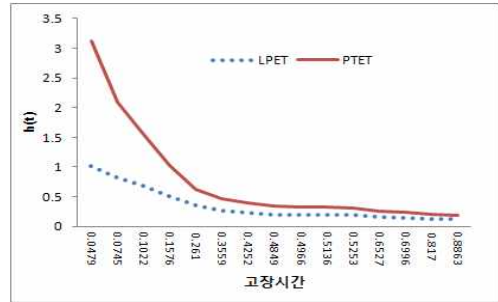
(그림 2) 고장수가 15인 경우의 평균값 함수

이 표에서 고장수가 15, 20인 경우는 MSE 는 LPET (로그 포아송 실행 시간 모형)보다 PTET(파레토 실행 시간모형)이 작고 R^2 의 값은 크기 때문에 보다 효율적인 모형이고 고장수가 30인 경우에는 LPET 모형이 효율적으로 나타나고 있다. 즉 소표 본인 경우에는 PTET이 효율적인 모형이고 대표본인 경우에는 LPET이 효율적인 모형으로 간주 될 수 있다.

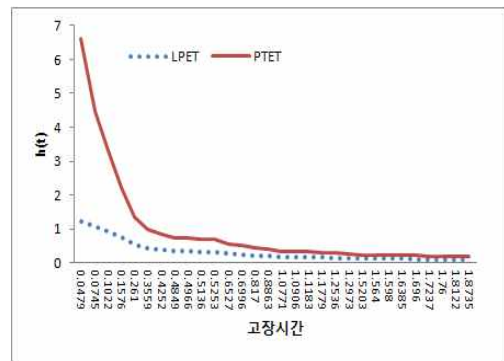


(그림 3) 고장수가 30인 경우의 평균값 함수

(그림 2) 와 (그림 3) 에서는 고장수가 15(소 표본)와 고장수가 30인(대 표본) 경우의 평균값 함수 패턴을 나타내었다. 이 그림에서 소 표본인 경우에는 PTET가 초반부에는 높으나 후반부에는 LPET가 높아 감을 알 수 있다. 대표본인 경우에는 고장시간에 대한 평균값은 항상 높은 패턴으로 나타나고 있다. 그리고 (그림 4)와 (그림 5)에서는 모두 비 증가 패턴을 가진 위험함수로 나타나고 있으며 PTET가 소 표본인 경우나 대 표본인 경우에 높게 나타나고 있다.

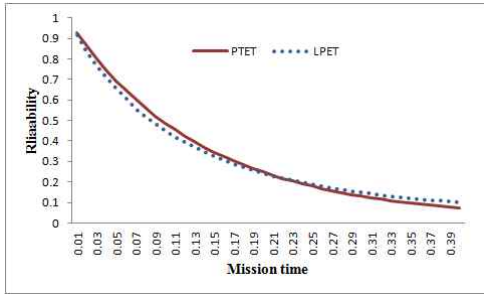


(그림 4) 고장수가 15인 경우의 위험함수

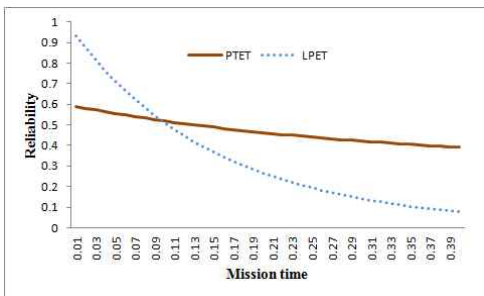


(그림 5) 고장수가 30인 경우의 위험함수

(그림 6) 에서는 고장 수가 15인 경우의 임무시간 (Mission time)에 의존한 신뢰도($\hat{R}(t | x_{15} = 0.8863)$)을 그려본 결과 모두 단조 감소형태를 나타내고 근소한 차이지만 PTET가 초반부에는 높으나 후반부에는 LPET가 높은 신뢰도를 있고 (그림 7) 에서는 고장수가 30인 경우의 임무시간(mission time)에 의존한 신뢰도($\hat{R}(t | x_{30} = 1.8735)$)에서는 단조 감소형태를 나타내고 있고 초반부에서는 LPET가 후반부는 PTET가 신뢰도가 높은 패턴으로 나타나고 있다. 이 결과는 적합 시킨 자료에 의한 결과이기 때문에 다른 자료를 적합 시키면 결과는 모수 추정 값은 달라질 수 있지만 소 표본 인 경우에는 PTET가 대체적으로 효율적인 모형임을 확인 할 수 있어서 이 분야에서 효율적 모형으로 선택 될 수 있음을 보여주고 있다.



(그림 6) 고장수가 15인 경우의 신뢰도



(그림 7) 고장수가 30인 경우의 신뢰도

5. 결 론

대용량 소프트웨어가 수정과 변경하는 과정에서 결점의 발생을 거의 피할 수 없는 상황이 현실이다. 따라서 소프트웨어 관리자들이 소프트웨어 및 검사 도구에 효율적으로 사용할 수 있도록 고장 시간 특성에 관한 NHPP 소프트웨어 모형에 대하여 연구 하였다.

따라서 적합 시킨 자료에 의한 결과이기 때문에 다른 자료를 적합 시키면 모수 추정 값의 결과는 달라질 수 있지만 소 표본 인 경우에는 기존모형의 대안으로서 파레토 실행시간 모형이 대체적으로 효율적인 모형임을 확인 할 수 있어서 이 분야에서 효율적 모형으로 선택 될 수 있음을 보여주고 있다.

결과적으로 고장관측 시간이 길지 않은 작업환경인 경우에는 파레토 실행시간 모형으로 긴 경우에는 로그 포아송 실행시간모형을 적용하면 바람직한 작업 결과를 얻을 수 있다.

경우에 따라서는 왜도와 첨도 측면에서 효율적인

카파분포, 지수화지수분포 등 업데이트된 분포에 대한 적용 문제를 비교 분석하는 연구도 가치 있는 일이라 판단되고 이 연구를 통하여 소프트웨어 개발자들은 수명분포에 의존한 고장시간 특성을 파악 하는데 어느 정도 도움을 줄 수 있으리라 사료 된다.

참고문헌

- [1] Gokhale, S. S. and Trivedi, K. S. "A time/structure based software reliability model", Annals of Software Engineering. 8, pp. 85-121. 1999.
- [2] Goel AL, Okumoto K, " Time-dependent fault detection rate model for software and other performance measures", IEEE Trans Reliab 28, pp.206-11, 1978.
- [3] Huang C-Y. "Performance analysis of software reliability growth models with testing-effort and change-point". J Syst Software 76, pp. 181-194, 2005.
- [4] Kuei-Chen, C., Yeu-Shiang, H., and Tzai-Zang, L. "A study of software reliability growth from the perspective of learning effects". Reliability Engineering and System Safety 93, pp. 1410 - 1421, 2008.
- [5] Prince, D. R. and Vivekanandan, P., "TRUNCATED SOFTWARE RELIABILITY GROWTH MODEL", performance measure, Korean J. Comput. & Appl. Math.(Series A) Vol. 9, pp. 761-769, 2002.
- [6] A. K. Sheikh, J. K. Boah, M. Younas., "Truncated Extreme value model for pipeline Reliability", Reliability Engineering and system safety, Vol. 28, pp.1-14, 1989.
- [7] 간광현, 이재근, 김희철, "기록값 통계량 모형기 초한 NHPP 소프트웨어 신뢰성에 관한 연구", 한국 통신학회논문지, 30권12T호, pp.356-363, 2005년 12월
- [8] L. Kuo and T. Y. Yang."Bayesian Computa

tion of Software Reliability". Journal of the American Statistical Association, Vol.91, p p. 763-773, 1996.

- [9] Hee-Cheul KIM and Hyoung-Keun Park, " The Comparative Study of Software Optimal Release Time Based on Burr Distribution", International Journal of Advancements in Computing Technology, Volume 2, Number 3, pp. 119-128, 2010.
- [10] Musa, J. D and Okumoto, K. "A Logarithmic Poisson Execution Time Model for Software Reliability Measurement," Proceeding the 7th International Conference on Software Engineering. pp. 230-238, 1984.
- [11] J. F. Lawless. Statistical Models and Methods for Lifetime Data. John Wiley & Sons, New York, 1981.
- [12] Y. HAYAKAWA and G. TELFAR "Mixed Poisson-Type Processes with Application in Software Reliability", Mathematical and Computer Modelling, 31, pp. 151-156, 2000.
- [13] K. Kanoun and J. C. Laprie, "Handbook of Software Reliability Engineering", M.R.Lyu, Editor, chapter Trend Analysis. McGraw-Hill New York, NY: 1996; p.401-437.

[저 자 소개]



김 희 철 (Hee-cheul Kim)

1992년 2월 동국대학교 통계학과 졸업(이학석사)

1998년 8월 동국대학교 통계학과 졸업(이학박사)

email : kim1458@nsu.ac.kr



신 현 철 (Hyun-cheul Shin)

1990년 2월 광운대학교 전자계산학과 졸업(공학석사)

2002년 2월 원광대학교 컴퓨터공학과 졸업(공학박사)

email : hcshin@bcc.ac.kr