

로그 및 지수형 결합 발생률에 따른 소프트웨어 신뢰성 모형에 관한 신뢰도 성능분석 연구

김희철*, 신현철**

The Study for Performance Analysis of Software Reliability Model using Fault Detection Rate based on Logarithmic and Exponential Type

Hee-Cheul Kim*, Hyun-Cheul Shin**

요 약 소프트웨어 개발과정에서 소프트웨어 신뢰성은 매우 중요한 이슈이다. 소프트웨어 고장분석을 위한 유한고장 비동질적인 포아송과정에서 고장발생률이 상수이거나, 단조 증가 또는 단조 감소하는 패턴을 가질 수 있다. 본 연구에서는 소프트웨어 제품 테스트 과정에서 관측고장시간에 근거한 로그 및 지수형 결합 발생률을 고려한 소프트웨어 신뢰성 모형에 대하여 연구 하였다. 신뢰성 분야에서 많이 사용되는 Goel-Okumoto모형을 이용한 새로운 로그 및 지수형 결합 확률을 반영한 문제를 제시하였다. 수명분포는 유한고장 비동질적인 포아송과정을 이용하고 모수추정법은 최우 추정법을 이용 하였다. 따라서 본 논문에서는 로그 및 지수형 결합발생률을 고려한 소프트웨어 모형분석을 위하여 소프트웨어 고장 시간간격 자료를 적용하여 비교 분석하였다. 본 연구에서 제안된 방법은 로그 및 지수형 결합발생률을 고려한모형도 신뢰성 측면에서 효율적이기 때문에 (결정계수가 80% 이상) 이 분야에서 기존 모형의 하나의 대안으로 사용할 수 있음을 확인 할 수 있었다. 이 연구를 통하여 소프트웨어 개발자들은 다양한 수명분포를 고려함으로써 소프트웨어 고장형태에 대한 사전지식을 파악하는데 도움을 줄 수 있으리라 사료 된다.

Abstract Software reliability in the software development process is an important issue. Infinite failure NHPP software reliability models presented in the literature exhibit either constant, monotonic increasing or monotonic decreasing failure occurrence rates per fault. In this paper, reliability software cost model considering logarithmic and exponential fault detection rate based on observations from the process of software product testing was studied. Adding new fault probability using the Goel-Okumoto model that is widely used in the field of reliability problems presented. When correcting or modifying the software, finite failure non-homogeneous Poisson process model. For analysis of software reliability model considering the time-dependent fault detection rate, the parameters estimation using maximum likelihood estimation of inter-failure time data was made. The logarithmic and exponential fault detection model is also efficient in terms of reliability because it (the coefficient of determination is 80% or more) in the field of the conventional model can be used as an alternative could be confirmed. From this paper, the software developers have to consider life distribution by prior knowledge of the software to identify failure modes which can be able to help.

Key Words :Software Reliability, NHPP, Logarithmic and exponential fault detection, Laplace trend test

* Corresponding Author : Division of Industrial & Management Engineering, Namseoul University, Seoul, Korea

**Division of Computer Engineering, BaekSeok Culture University, Seoul, Korea

Received June 16, 2016

Revised June 21, 2016

Accepted June 23, 2016

1. 서론

이 분야에서 특정한 환경조건에서 특정기간동안 고장이 나지 않고 운영 할 수 있는 확률이 소프트웨어 신뢰성이라고 정의 할 수 있다.

따라서 소프트웨어 신뢰성은 소프트웨어 개발과정에서 핵심과제가 된다. 이러한 사항은 사용자의 요구조건을 만족시키고 테스트 비용을 줄여야 한다. 소프트웨어의 신뢰성의 변동과 테스트 비용을 사전에 알고 있으면 소프트웨어 테스트(디버깅)면에서 비용을 줄일 수 있다. 따라서 소프트웨어 개발과정 고려사항(신뢰도, 비용 및 방출시간)을 만족하는 소프트웨어 개발 과정이 요구되어 진다. 이 분야에서 Gokhale과 Trivedi [1]은 고양된 비동질적인 포아송 과정 모형(enhanced NHPP) 모형을 제시하였고 Goel 과 Okumoto [2]은 지수적 소프트웨어 신뢰성 모형(exponential software reliability growth model)을 제안 하였다. 이 중에서 비동질적 포아송 과정(non-homogeneous Poisson process; NHPP)에 의존한 모형[1]은 여러 탐색 과정측면에서는 우수한 모형이고 이 모형은 결함이 발생하면 즉시 제거되고 디버깅 과정에서 새로운 결함이 발생되지않는다는 가정을 하고 있다[2]. 즉, 결함 탐색 비율을 일정한 상수로 간주하여 분석하는 연구가 대부분이었다.

따라서 본 연구에서는 이 분야에서 기본 모형인 지수 분포를 이용하여 개발된 Goel-Okumoto 모형을 이용하여 테스트 하는 과정에서 결함 탐색 비율을 관측시간에 의존한 로그형 및 지수 형태를 이용하고 이에 대한 소프트웨어 모형을 비교 제시 하였다

2. 제안된 로그형 및 지수형 결함발생률을 고려한 NHPP-Goel-Okumoto 모형

이 분야에서 기본 모형인 Goel-Okumoto모형은 유한 고장 상황에서 고장의 원인이 되는 결함의 기대 값 θ 을 라고 표현하고 결함 탐색비율을 β 라고 하면 NHPP 모형에서 결함 탐색비율 β 는

고정상수로 간주하여 정의되었지만[8] 본 연구에서는 결함 탐색률 β 을 관측시간에 의존한 함수 $\beta(i) (i=1,2,\dots,n)$ 로 간주하여 다음과 같은 고정상수, 로그 및 지수 형 패턴을 제시하고자 한다.

$$\beta(i) = \beta, \beta(i) = \beta \ln(d_i), \beta(i) = \beta (\exp(d_i)) \quad (1)$$

$$d_1 = x_1, d_i = x_i - x_{i-1}, (i = 2, 3 \dots, n)$$

소프트웨어 신뢰성 분야에서 널리 사용되는 모형 중에서 가장 많이 연구된 모형이 수명분포가 지수분포를 따르는 경우가 Goel-Okumoto 모형이다. 이모형에 대한 평균값함수와 강도함수는 다음과 같이 알려져 있다 [2].

$$m(t|\theta, \beta) = \theta F(t) = \theta(1 - e^{-\beta t}) \quad (2)$$

$$\lambda(t|\theta, \beta) = \theta f(t) = \theta \beta e^{-\beta t} \quad (3)$$

따라서 [그림 1]에서 보여 주듯이 결함 탐색비율 β 을 관측시간에 의존한 함수 $\beta(i) (i=1,2,\dots,n)$ 로 간주하여 로그형 패턴을 반영한 평균값함수와 강도함수는 각각 다음과 같이 표현 할 수 있다.

$$m(t|\theta, \beta) = \theta F(t) = \theta(1 - e^{-\beta \ln(d_i) t}) \quad (4)$$

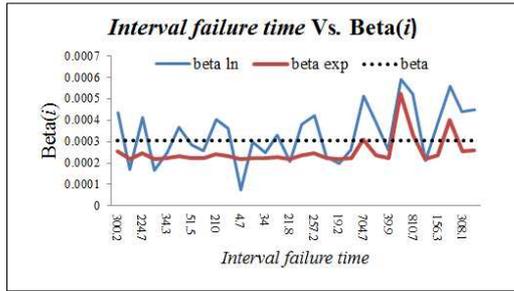
$$\lambda(t|\theta, \beta) = \theta f(t) = \theta \beta \ln(d_i) e^{-\beta \ln(d_i) t} \quad (5)$$

유사하게 지수형 패턴을 반영한 평균값함수와 강도함수는 각각 다음과 같이 표현 할 수 있다.

$$m(t|\theta, \beta) = \theta F(t) = \theta(1 - e^{-\beta \exp(d_i) t}) \quad (6)$$

$$\lambda(t|\theta, \beta) = \theta f(t) = \theta \beta \exp(d_i) e^{-\beta \exp(d_i) t} \quad (7)$$

단, $\theta > 0, \beta > 0$.



단 $beta = \beta$, $beta \ln = \beta \ln(d_i)$, $beta \exp = \beta (\exp(d_i))$,
 그림 1. 결함 탐색비율
 Fig. 1. Fault Detection Rate

본 연구에서는 t 을 최종고장시점 x_n 으로 대체하고 최우추정법(MLE)을 이용하기 위한 로그우도함수를 구하면 다음과 같다.

$$\ln L_{NHPP}(\theta | \underline{x}) = n \ln \theta + n \ln \beta - \beta \sum_{i=1}^n x_i - \theta (1 - e^{-\beta x_n}) \quad (8)$$

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

따라서 고장절단모형에서의 모수 θ 와 β 에 관한 편미분식은 다음과 같이 유도 할 수 있다 [1, 2, 3].

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - 1 + e^{-\beta x_n} = 0 \quad (9)$$

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \beta} = \frac{n}{\beta} - \sum_{i=1}^n x_i - \theta x_n e^{-\beta x_n} = 0 \quad (10)$$

따라서 각 모수에 대한 최우추정량 $\hat{\theta}_{MLE}$ 와 $\hat{\beta}_{MLE}$ 은 다음 식을 만족한다.

$$\frac{n}{\hat{\theta}} = 1 - e^{-\hat{\beta} x_n} \quad (11)$$

$$\frac{n}{\hat{\beta}} = \sum_{i=1}^n x_i + \hat{\theta} x_n e^{-\hat{\beta} x_n} \quad (12)$$

Goel-Okumoto모형과 유사한 방법을 동원하면 로그형 패턴을 반영한 각 모수에 대한 최우추정량

$\hat{\theta}_{MLE}$ 와 $\hat{\beta}_{MLE}$ 은 다음 식을 만족한다.

$$\frac{n}{\hat{\theta}} = 1 - e^{-\hat{\beta} \ln(d_n) x_n} \quad (13)$$

$$\frac{n}{\hat{\beta}} = \sum_{i=1}^n \ln(d_i) x_i + \hat{\theta} \ln(t_n) x_n e^{-\hat{\beta} \ln(d_n) x_n} \quad (14)$$

유사하게 지수형 패턴을 반영한 각 모수에 대한 최우추정량 $\hat{\theta}_{MLE}$ 와 $\hat{\beta}_{MLE}$ 은 다음 식을 만족한다.

$$\frac{n}{\hat{\theta}} = 1 - e^{-\hat{\beta} \exp(d_n) x_n} \quad (15)$$

$$\frac{n}{\hat{\beta}} = \sum_{i=1}^n \exp(d_i) x_i + \hat{\theta} \exp(d_n) x_n e^{-\hat{\beta} \exp(d_n) x_n} \quad (16)$$

3. 소프트웨어 고장시간 분석

이 절에서는 소프트웨어 고장 시간자료 [4] (Failure time data)를 이용하여 본 논문에서 제시하는 소프트웨어 신뢰모형들을 이용한 소프트웨어 고장시간을 분석하고자 한다. 이 자료의 고장시간은 738.68 시간단위에 30번의 고장이 발생한 자료이며 [표 1]에 나열 되어 있다.

또한 제시하는 신뢰모형들을 분석하기 위하여 우선 자료에 대한 추세검정이 선행 되어야 한다 [5, 6]. 추세분석에는 일반적으로 라플라스 추세검정(Laplace trend test)을 사용한다. 이 검정을 실시한 결과 [그림 1]에서 라플라스요인(Factor)이 -2와 2사이에 존재함으로써 즉, 극단값(Extreme value)이 존재하지 않으므로 이 자료를 이용하여 신뢰성장모형을 제시하는 것이 효율적임을 시사하고 있다[5, 7, 8]. 그리고 로그 및 지수형 결함탐색률은 지수 와 로그 성질을 만족하도록 다음과 같은 패턴을 이용하여 모수를 추정하였다.

$$\beta(i) = \beta \ln(d_i \times 5 \times 10^{-1}) \quad (17)$$

$$\beta(i) = \beta \exp(d_i \times 5 \times 10^{-4}) \quad (18)$$

표 1. 소프트웨어의 고장시간자료
Table 1. software failure time data

Failure number (<i>i</i>)	Failure time (hours)	Failure time (hours) × 10 (<i>x_i</i>)	Failure interval (hours) (<i>d_i</i>)
1	30.02	0.3002	300.2
2	31.46	0.3146	314.6
3	53.93	0.5393	539.3
4	55.29	0.5529	552.9
5	58.72	0.5872	587.2
6	71.92	0.7192	719.2
7	77.07	0.7707	770.7
8	80.9	0.809	809
9	101.9	1.019	1019
10	114.87	1.1487	1148.7
11	115.34	1.1534	1153.4
12	121.57	1.2157	1215.7
13	124.97	1.2497	1249.7
14	134.07	1.3407	1340.7
15	136.25	1.3625	1362.5
16	151.78	1.5178	1517.8
17	177.5	1.775	1775
18	180.29	1.8029	1802.9
19	182.21	1.8221	1822.1
20	186.34	1.8634	1863.4
21	256.81	2.5681	2568.1
22	273.88	2.7388	2738.8
23	277.87	2.7787	2778.7
24	453.93	4.5393	4539.3
25	535	5.35	5350
26	537.27	5.3727	5372.7
27	552.9	5.529	5529
28	673.68	6.7368	6736.8
29	704.49	7.0449	7044.9
30	738.68	7.3868	7386.8

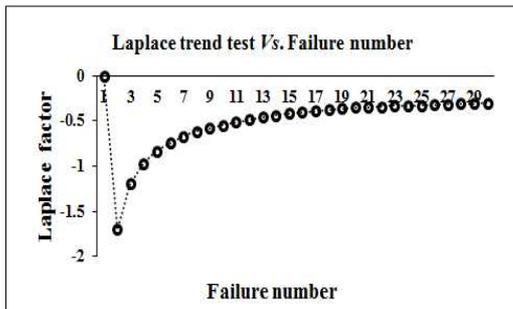


그림 2. 라플라스 추세검정
Fig. 2. Laplace trend test

모수추정은 최우추정법을 이용하고 모수 추정을 용이하게 하기 위하여 원래의 고장시간 데이터를 변수변환($Failure\ time \times 10$)하여 적용하였다 [9]. 비선형 방정식의 계산방법은 수치 해석적 기본 방법인 이분법(Bisection method)을 사용하였다. 이러한 계산은 초기 값을 10^{-5} 와 5을, 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. 최우추정법의 결과와 모형에 대한 효율성을 조사하기 위한 기준으로서 MSE (평균제곱오차)와 R^2 (결정계수)가 [표 2]에 요약되었다. 여기에서 평균제곱오차는 실제 관찰 값과 예측 값에 대한 차이를 측정하는 도구로서 다음과 같이 정의 된다 [5, 7, 10].

$$MSE = \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{n - k} \quad (19)$$

단, $m(x_i)$ 은 시간(0, x_i]까지 나타난 에러들의 누적합수를 의미하고 $\hat{m}(x_i)$ 는 x_i 시점까지 평균값 함수로부터 추정된 에러의 누적개수를 의미한다.

결정계수는 관찰 값의 차이에 대한 설명력을 나타내는 도구로서 다음과 같이 정의 된다 [3, 5].

$$R^2 = 1 - \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{\sum_{i=1}^n (m(x_i) - \sum_{j=1}^n m(x_j)/n)^2} \quad (20)$$

표 2. 모수 추정값 및 MSE , R^2
Table 2. Parameter estimation of the each model and MSE , R^2

Fault Detection Rate	<i>MLE</i>	Model Comparison	
		<i>MSE</i>	R^2
$\beta(i) = \beta$	$\hat{\theta}_{MLE} = 33.4086$ $\hat{\beta}_{MLE} = 3.06 \times 10^{-4}$	6.08061	0.9371
$\beta(i) = \beta \ln(d_i \times 5 \times 10^{-1})$	$\hat{\theta}_{MLE} = 31.1439$ $\hat{\beta}_{MLE} = 8.70 \times 10^{-5}$	134.99	0.8475
$\beta(i) = \beta \exp(d_i \times 5 \times 10)$	$\hat{\theta}_{MLE} = 35.2113$ $\hat{\beta}_{MLE} = 2.18 \times 10^{-4}$	17.991	0.9065

Note. *MLE* : Maximum likelihood estimation;
MSE : Mean square error;
 R^2 : Coefficient of determination
 $d_1 = x_1, d_i = x_i - x_{i-1}, (i = 2, 3, \dots, n)$

이 표에서 결함탐색비율이 고정된 상수모형이 지수와 로그 모형 모형보다 실제 값과 예측 값에 대한 차이를 측정하는 평균제곱오차가 낮고 예측 값의 차이에 대한 설명력을 의미하는 결정계수도 높게 나타나기 때문에 결함탐색비율이 고정된 상수 모형이 다른 모형보다 효율적인 모형으로 간주할 수 있다. 그러나 실제 상황에서는 관측시간에 의존하는 결함탐색비율 모형이 더 현실적모형으로 판단된다. 그러나 지수 및 로그 모형도 결정계수가 80%이상 나타나 이 분야에서 효율적 모형임을 확인 할 수 있다[5, 6].

[그림 2]은 강도함수에 대한 패턴이 요약 되었다. 이 그림에서 제시한 모형들은 고장시간이 지남에 따라 완만한 감소형태를 보이고 있다.

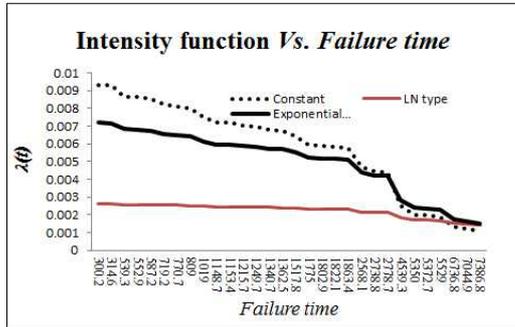


그림 3. 강도함수의 형태
Fig. 3. Pattern of intensity function

[그림 3]은 평균값 함수에 대한 패턴이 요약 되었다. 이 그림에서 제안된 모형은 참값과의 차이에서 과소평가 추정이 이루어졌고 고정상수 결함탐

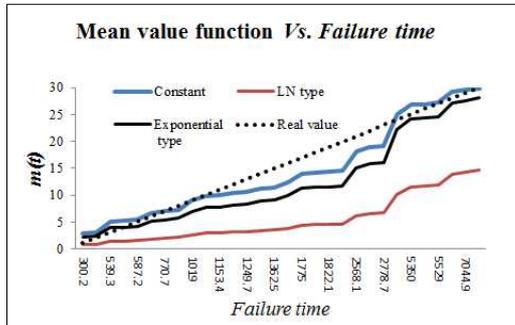


그림 4. 평균값 함수의 형태
Fig. 4. Pattern of mean value function

색비율 모형이 지수 및 로그모형보다 참값과의 차이에서 가깝게 추정되었음을 보여주고 있다.

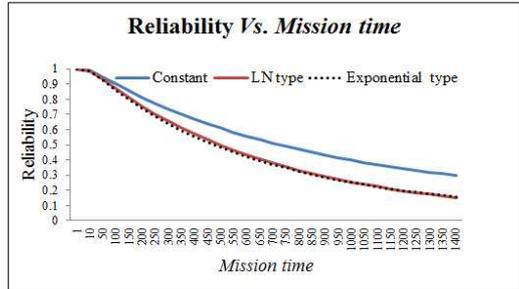


그림 5. 신뢰도의 형태
Fig. 5. Pattern of reliability

[그림 4]에서 보여 주듯이 임무시간에 대한 신뢰도 그림에서도 고정상수 결함탐색률 모형이 신뢰도의 상승으로 나타나고 있다. 지수 및 로그모형 신뢰도는 거의 유사한 패턴을 보이고 있다.

4. 결론

소프트웨어 신뢰성은 개발의 최종단계에 있는 테스트 공정이나 실제 사용단계에 있어서 소프트웨어 내에 존재하는 고장 수나 고장 발생시간에 의해서 효과적으로 평가할 수 있는 상황으로 그 평가 기술이 중요하게 된다. 따라서 소프트웨어 개발의 테스트공정이나 실제사용단계에 있어서 고장 발생 환경이나 고장 발생현상을 수리적으로 모형화가 가능하면 평가를 할 수 있다.

따라서 본 연구에서는 대부분 연구에서는 결함 탐색비율을 일정한 상수로 간주하여 분석하는 연구가 대부분이었다. 본 연구에서는 결함 탐색비율을 관측시간에 의존한 로그형 및 지수 형태를 이용하고 이에 대한 소프트웨어 모형을 비교 제시하였다.

그 결과 결함탐색비율이 고정된 상수모형이 다른 모형보다 효율적인 모형으로 간주할 수 있다. 그러나 실제 상황에서는 관측시간에 의존하는 결함 탐색비율 모형이 더 현실적인 모형으로 판단된다.

따라서 본 연구에 제안된 지수 및 로그 모형도 신뢰성 측면에서 효율적이기 때문에 (결정계수가 80% 이상) 이 분야에서 기존의 모형의 하나의 대

안으로 사용할 수 있음을 확인 할 수 있었다. 이 연구를 통하여 소프트웨어 개발자들은 다양한 결함탐색비용을 고려함으로써 소프트웨어 결함탐색 비용에 대한 사전지식을 파악하는데 도움을 줄 수 있으리라 사료 된다.

REFERENCES

[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 A L, Okumoto K, "Time-dependent fault detection rate model for software and other performance measures", IEEE Trans. Reliab. 28, pp. 206-11, 1978.

[3] Hee-Cheul KIM, "The Assessing Comparative Study for Statistical Process Control of Software Reliability Model Based on Rayleigh and Burr Type", Journal of korea society of digital industry and information management, Volume 10, No.2, pp. 1-11. 2014.

[4] K,H Rao, R. S, Prasad and. R.L.Kantham "Software Reliability Measuring using Modified Maximum Likelihood Estimation and SPC", International Journal of Computer Applications(0975 - 8887), Volume 21, No.7, pp. 1-5., May 2011.

[5] Tae-Hyun Yoo, "The Infinite NHPP Software Reliability Model based on Monotonic Intensity Function", Indian Journal of Science and Technology, Vol. 8, No. 14, pp. 1-7, 2015.

[6] K. Kanoun and J. C. Laprie, "Handbook of Software Reliability Engineering", M.R.Lyu, Editor, chapter Trend Analysis. McGraw-Hill New York, NY, pp. 401-437, 1996.

[7] 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.

[8] J. F. Lawless. Statistical Models and Methods for Life time Data. John Wiley & Sons, New York, 1981.

[9] Kuo L, Yang TY., "Bayesian computation of software reliability" .Journal of the American Statistical Association. Vol.. 91, pp.763 - 773,, 1999.

[10] Kim H-C. The Property of Learning effect based on Delayed Software S-Shaped Reliability Model using Finite NHPP Software Cost Model, Indian Journal of Science and Technology 8(34), pp.1-7, 2015.

저자약력

김 희 철 (Hee-Cheul Kim) [정회원]



- 1992년 2월 : 동국대학교 통계학과 (이학석사)
- 1998년 8월 : 동국대학교 통계학과 (이학박사)
- 2005년 3월 ~ 현재 : 남서울대학교 산업경영공학과 교수

<관심분야>

소프트웨어신뢰성 공학, 전산 통계, 웹프로그래밍

신 현 철 (Hyun-Cheul Shin) [정회원]



- 2002년 2월 : 원광대학교 컴퓨터공학과 (공학박사)
- 2005년 3월 : 한국정보처리학회 이사
- 1994년 3월 ~ 현재 : 백석문화대학교 컴퓨터공학부 교수

<관심분야>

통신공학, 컴퓨터과학, 소프트웨어공학, 소프트웨어공학