

무한고장 NHPP Lomax 수명분포의 형상모수 변화에 따른 소프트웨어 신뢰성 모형의 속성 분석에 관한 연구

민경일
남서울대학교 보건행정학과 교수

A Study on the Attribute Analysis of Software Reliability Model with Shape Parameter Change of Infinite Fault NHPP Lomax Life Distribution

Kyung-il Min
Professor, Dept. of Public Health Administration, Namseoul University

요약 본 연구에서는 무한고장 NHPP 로맥스 수명분포의 형상모수 변화에 따른 소프트웨어 신뢰성 모형의 속성을 새롭게 분석한 후 최적의 형상모수 조건을 제시하였다. 소프트웨어 고장현상을 분석하기 위하여 모수추정은 최우추정법을 사용하였고, 비선형 방정식의 계산은 이분법을 적용하였다. 그 결과, 형상모수(k)의 변화에 따른 속성을 비교하였을 때 형상모수가 작을 수록 참값에 대한 예측능력이 우수하고, 신뢰 속성이 효율적임을 알 수 있었다. 본 연구를 통하여, 소프트웨어 개발자들은 형상모수에 근거한 소프트웨어 고장형태를 사전에 파악함으로써 신뢰도를 성장시킬 수 가 있으며, 또한, 소프트웨어의 신뢰속성을 향상시키는데 필요한 기본정보로 활용할 수 있을 것으로 기대한다.

주제어 : 무한고장, 비등질적 포아송과정, 로맥스 수명분포, 소프트웨어 신뢰성모형, 신뢰성 분석기법, 형상모수

Abstract In this study, the optimal shape parameter condition is presented after analyzing the attributes of the software reliability model according to the change of the shape parameter of Loma life distribution with infinite fault NHPP. In order to analyze the software failure phenomena, the parametric estimation method was applied to the Maximum Likelihood Estimation method, and the nonlinear equation was applied to the bisection method. As a result, it was found that when the attributes according to the change of the shape parameter are compared, the smaller the shape parameter is, the better the prediction ability of the true value, and reliability attributes are efficient. Through this study, it is expected that software developers can increase reliability by preliminarily grasping the type of software failure based on shape parameter, and can be used as basic information to improve the software reliability attributes.

Key Words : Infinite failure, NHPP, Lomax life distribution, Software reliability model, Reliability analysis technique, Shape parameter

1. 서론

4차산업 혁명시대의 도래와 함께 근간이 되는 D(Data), N(Network), A(Artificial Intelligent) 기술 또한 급격히

성장하고 있다. 이들 DNA 산업을 이끄는 핵심 인프라중 대용량 데이터를 신속하게 처리할 수 있는 소프트웨어 개발이 더욱 관심을 받고 있다. 특히, 개발자들에게는 이러한 소프트웨어를 개발하는 과정에서 좀더 신뢰성 높은 소프트웨어를 만드는 것이 가장 중요한 임무가 될 것이다. 그러므로

*Funding for this paper was provided by Namseoul University year 2019.

Corresponding Author : Kyung-il Min(kimin85@nsu.ac.kr)

Received July 11, 2019

Accepted August 20, 2019

Revised August 2, 2019

Published August 28, 2019

소프트웨어 개발과정에서 소프트웨어 신뢰성을 향상 시킬 수 있는 품질 평가 기술이 더욱 필요해지고 있다. 이를 위해 소프트웨어 개발자는 소프트웨어를 최종 출시하기 전에 시스템의 요구사항을 충족시키는지를 사전 신뢰성 평가 테스트를 하게 된다. 소프트웨어 개발을 효율적으로 진행하기 위해서는 소프트웨어 신뢰 속성은 물론, 결함 발생을 사전에 예측할 수 있어야 한다. 그러므로, 소프트웨어의 결함 발생을 사전에 탐색하기 위한 신뢰성 모형에 관한 연구가 활발히 진행되고 있으며, 현재까지 많은 모형들이 제안 되었다. 제안된 모형중 비동질적 포아송 과정(Non-homogeneous Poisson process: NHPP)을 적용한 모형은 결함이 발생하면 즉시 제거되며, 결함을 수리하는 도중에는 새로운 결함이 발생되지 않는다는 유한고장의 가정을 전제로 하고 있어서 효율적인 모형이다[1]. 아울러, 동일한 연구 분야에서, Goel and Okumoto은 지수 형태의 소프트웨어 신뢰성 모형을 제안 하기도 하였다[2]. 또한, Yang은 형상모수 변화에 관한 연구로서 Lomax 분포의 소프트웨어 개발비용 모형에 관한 분석을 하기도 하였다[3]. 하지만, 기존 연구들은 유한고장 기반의 신뢰성 예측 방법을 제안하거나, 여러 분포 모형들 중 가장 최적의 신뢰성 모형을 제시하는 내용이 대부분이다. 최근에는 NHPP 신뢰성 모형을 가지고 소프트웨어 고장분포의 형태를 결정하는 형상모수의 변화에 따라 신뢰도 속성을 분석하고자 하는 연구가 시도되고 있다[4].

따라서, 본 연구에서는 무한고장 NHPP Lomax 수명분포에 근거하여, 형상모수 변화에 따른 신뢰성 모형의 속성을 비교, 분석한 후 최적의 형상모수 조건을 제시하고자 한다.

2. 관련 연구

2.1 NHPP 신뢰성 모형

$N(t)$ 을 시간 t 까지 탐지된 소프트웨어의 고장의 누적 수이고, $m(t)$ 를 이에 대한 평균값(고장 발생 기대값) 함수, $\lambda(t)$ 을 강도함수(Intensity function)로 표시하면, 누적 고장수 $N(t)$ 는 모수 $m(t)$ 을 가진 포아송 확률밀도함수(Probability density function)를 따른다고 하였다. 비동질적 포아송 과정(NHPP)의 소프트웨어 신뢰성 모형은 단위 시간당 발생되어지는 고장의 횟수를 중심으로 평균 고장률 함수를 이용해서 소프트웨어의 신뢰성을 측정하는 모형이다. 즉,

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

단, $n = 0, 1, 2 \dots \infty$

NHPP 모형의 평균값 함수 $m(t)$ 와 강도함수 $\lambda(t)$ 는 다음과 같다. 여기서, 소프트웨어 신뢰성 관점에서 평균값 함수는 소프트웨어 고장발생 기댓값을 나타내고, 강도함수는 고장율 함수로서, 결합당 고장 발생율을 의미한다.

$$m(t) = \int_0^t \lambda(s) ds \quad (2)$$

$$\frac{dm(t)}{dt} = \lambda(t) \quad (3)$$

그러므로, 결함을 수리하는 시점에서도 고장이 발생한다는 무한고장의 조건을 반영하기 위하여 기록 멈춤 통계량(Record breaking statistics)을 적용하는 RVS(Record Value Statistics)모형을 사용할 수 있다. 그러므로, 무한고장 NHPP 모형의 평균값 함수는 다음과 같이 된다.

$$m(t) = -\ln(1 - F(t)) \quad (4)$$

따라서 $f(t)$ 을 확률밀도함수, $F(t)$ 을 분포함수라고 하면, NHPP의 강도함수 $\lambda(t)$ 는 다음과 같이 위험함수 $h(t)$ 가 된다.

$$\lambda(t) = m(t)' = h(t) = f(t)/(1 - F(t)) \quad (5)$$

임의 $[0, t]$ 시간까지 확인하기 위한 시간절단(Time truncated) 모형은 n 번째까지 고장자료를

$$x_n = \sum_{i=1}^n t_i \quad (6)$$

단, $i = 1, 2, \dots, n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n$

이라고 하고, θ 을 모수공간이라고 표시하면, NHPP 모형의 우도함수(Likelihood function)는 다음과 같다.

$$L_{NHPP}(\theta | \underline{x}) = \left(\prod_{i=1}^n \lambda(x_i) \right) \exp[-m(x_n)] \quad (7)$$

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

2.2 무한고장 NHPP Lomax 신뢰성 모형

무한고장이란, 고장을 수리하는 동안에도 고장이 발생하며, 남아있는 잔존고장이 없다는 것을 전제로 한다. 본 연구에서 제안하는 로맥스(Lomax) 분포는 파레토(Pareto) 분포의 특수한 형태로서, 두꺼운 꼬리 확률 분포(heavy-tail probability)의 속성을 가지고 있기 때문에 소프트웨어 신뢰성 분야는 물론, 다양한 분야에서 모형화하는데 가장 널리 사용되는 분포이다. 로맥스 분포의 확률밀도함수와 분포함수는 다음과 같다.

$$f(x) = \frac{\lambda k}{(1 + \lambda x)^{k+1}}, \quad x > 0 \quad (8)$$

$$F(t) = 1 - (1 + \lambda x)^{-k} \quad (9)$$

단, $\lambda (> 0)$ 는 척도모수(Scale parameter)이고 $k (> 0)$ 는 형상모수(Shape parameter)을 의미한다.

무한고장 NHPP로 적용하여 풀면, (4)식의 평균값 함수와 (5)식의 강도 함수는 다음과 같이 다시 유도할 수 있다[5].

$$m(t) = -\ln(1 - F(t)) = k \ln(1 + \lambda x) \quad (10)$$

$$\lambda(t) = h(t) = f(t)/(1 - F(t)) = \frac{\lambda k}{1 + \lambda x} \quad (11)$$

또한, NHPP 모형의 로그 우도함수는 (7)식에 (10)식과 (11)식을 대입하여, 양변에 로그를 취하면 다음과 같다.

$$\ln L_{NHPP}(\lambda, k, | \underline{x}) = n \ln \lambda + n \ln k - \sum_{i=1}^n \ln(1 + \lambda x_i) - k \ln(1 + \lambda x_n) \quad (12)$$

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

여기서, 최우추정법(Maximum Likelihood Method)을 적용한 모수 $\hat{\lambda}_{MLE}$ 는 다음 식을 만족한다.

$$\frac{\partial \ln L_{NHPP}(\lambda, k, | \underline{x})}{\partial \lambda} = \frac{n}{\lambda} - \sum_{i=1}^n \frac{x_i}{1 + \lambda x_i} - k \left(\frac{x_n}{1 + \lambda x_n} \right) = 0 \quad (13)$$

본 연구에서는 고장분포의 형태를 결정하는 형상 모수 (k)의 변화에 따라서 Lomax-NHPP 신뢰성 모형의 속성 변화를 비교, 분석하기 위하여 형상모수가 $k = 30, 40, 50$ 인 경우를 고려하고자 한다. 척도모수 $\lambda = 0.5$ 일 경우 형상모수에 근거하여, 수명분포를 나타내는 확률밀도 함수 (PDF)의 분포는 Fig. 1과 같다.

Failure time Vs. probability density function

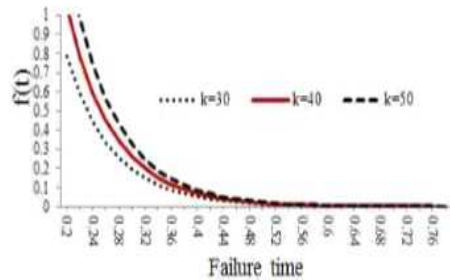


Fig. 1. Probability density function ($\lambda = 0.5$)

즉, 초기 고장 형태가 가장 높은 경우($k = 50$)와 초기 고장 형태가 가장 낮은 경우($k = 30$), 그리고 초기 고장 형태가 중간인 경우($k = 40$)를 구분하여 신뢰속성이 연구되었다.

3. 소프트웨어 고장시간 데이터분석 및 유효성 검증

본 연구에서는 소프트웨어 고장 시간 데이터를 이용하였는데, 이 자료의 고장시간은 738.68 시간 단위에 30번의 고장이 발생된 자료이며, Table 1에 요약되어 있다[6].

Table 1. Software Failure Time Data

Failure number	Failure time (hours)	Failure time (hours)×10 ⁻²
1	30.02	0.30
2	31.46	0.31
3	53.93	0.53
4	55.29	0.55
5	58.72	0.58
6	71.92	0.71
7	77.07	0.77
8	80.90	0.80
9	101.90	1.01
10	114.87	1.14
11	115.34	1.15
12	121.57	1.21
13	124.97	1.24
14	134.07	1.34
15	136.25	1.36

16	151.78	1.51
17	177.50	1.77
18	180.29	1.80
19	182.21	1.82
20	186.34	1.86
21	256.81	2.56
22	273.88	2.73
23	277.87	2.77
24	453.93	4.53
25	535.00	5.35
26	537.27	5.37
27	552.90	5.52
28	673.68	6.73
29	704.49	7.04
30	738.68	7.38

우선, 제안된 고장시간 데이터에 대한 신뢰성을 확보하고자 라플라스 추세검정(Laplace trend test)을 실시하였다. 일반적으로 라플라스 요인 추정 값이 -2와 2사이에 분포하면 극단적인 값이 존재하지 않고 안정적이기 때문에 자료에 대한 신뢰성을 가진다고 하였다[7]. 그 결과는 Fig. 2에 요약되었다. 이 그림에서 Laplace Factor가 “0과 +2” 사이에 안정적으로 존재함으로써 Table 1과 같은 데이터를 이용하는 것이 유효하다는 것을 검증하였다[8].

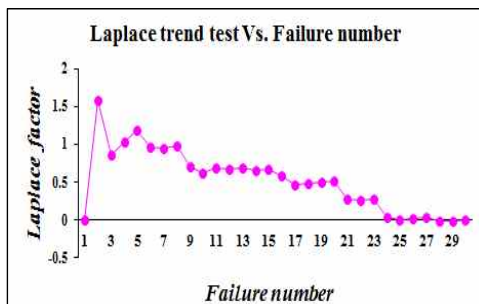


Fig. 2. Laplace Trend Test

4. 제안된 신뢰성모형의 속성분석 기법

본 연구에서 제안된 소프트웨어 신뢰성 모형의 속성분석 기법은 다음과 같다.

1 단계 : 최우추정법을 사용하여 제안 모형에 대한 모수 ($\hat{\lambda}_{MLE}$)를 계산한다.

2 단계 : 효율적인 모형 선택을 위한 평균제곱오차 (MSE) 및 결정계수(R^2)를 계산한다.

3 단계 : 참값에 대한 예측능력 $\lambda(t)$, $m(t)$ 및 임무시

간에 대한 신뢰도 $R(t)$ 를 분석한 후 최적의 형상모수 조건을 제시한다.

본 연구에서는 위에서 제안한 분석기법을 사용하여 다음과 같이 속성을 분석한다.

가. 1단계 : 최우추정법을 사용하여 제안 모형에 대한 모수 ($\hat{\lambda}_{MLE}$)를 계산한다.

먼저, 모수 ($\hat{\lambda}_{MLE}$)에 대한 추정은 최우추정법을 이용하였고, 비선형 방정식은 이분법(Bisection method)을 사용하여 계산하였다. 또한, 모수추정을 쉽게 하기위하여 Table 1과 같이 고장시간 데이터를 변수변환($Failure\ Time \times 10^{-2}$)하여, 적용하였다. 모수의 추정은 C-언어 프로그래밍을 이용하였고, 수렴성을 확인하면서 수행하였다. 또한, 최우추정법의 결과와 MSE (평균제곱오차)와 R^2 (결정계수)가 Table 2에 요약되었다[9].

Table 2. Parameter estimation of the each model and MSE , R^2

Shape Parameter	MLE	Model comparison	
	$\hat{\lambda}_{MLE}$	MSE	R^2
$k = 30$	0.8214	6.47404	0.97948
$k = 40$	0.5606	10.57867	0.96647
$k = 50$	0.4218	13.63771	0.95678

Note) MLE : Maximum likelihood estimation;

MSE : Mean square error;

R^2 : Coefficient of determination

나. 2단계 : 효율적인 모형 선택을 위한 평균제곱오차 (MSE) 및 결정계수(R^2)를 계산한다.

Table 2에서 평균제곱오차는 실제 관찰 값과 예측 값에 대한 오차를 나타내는 측정 값으로서, 이값이 작을수록 효율적인 모형이 된다[10].

$$MSE = \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{n - k} \tag{14}$$

여기서, $m(x_i)$ 은 시간 $[0, x_i]$ 까지 나타난 고장(결합)들의 누적합수이고, $\hat{m}(x_i)$ 는 x_i 시점까지 평균값 함수로부터 예측된 고장의 누적개수를 의미한다. 그리고 n 은 관찰 값의 수, k 는 모수의 개수를 의미한다. 실제 관찰 값의 차이에 대한 예측 값의 설명력을 나타내는 결정계수(R^2)는 값이 클수록 효율적인 모형이 된다.

$$R^2 = 1 - \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{\sum_{i=1}^n \left(m(x_i) - \frac{\sum_{j=1}^n m(x_j)}{n} \right)^2} \quad (15)$$

그러므로, Table 2와 같이 평균제곱오차와 결정계수의 결과 값을 보면, 형상모수 $k = 30$ 일 때 평균제곱오차가 제일 작고, 결정계수도 크게 나타나기 때문에 제안한 형상모수 값들 중에서, 가장 효율적이라고 결정할 수 있다[11].

여기서, Fig. 3은 평균제곱오차(MSE)의 추이를 나타내고 있다. Table 2에서 계산한 결과 값과 같이 형상모수 $k = 30$ 일 때 평균제곱오차가 제일 작게 나타나고 있음을 확인하였다.

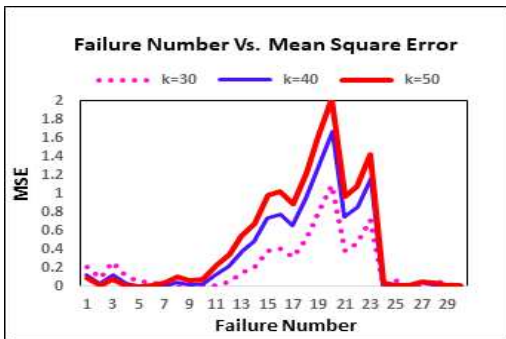


Fig. 3. Transition of Mean Square Error(MSE)

다. 3단계 : 참값에 대한 예측능력 $\lambda(t)$, $m(t)$ 및 임무 시간에 대한 신뢰도 $R(t)$ 를 분석한 후 최적의 형상모수 조건을 제시한다[12].

Fig. 4는 고장이 발생하는 강도를 나타내는 강도함수의 속성이 분석되었다. 이 그림에서 속성을 분석해보면, 형상모수 k 가 30, 40, 50일 때 모두 고장시간이 흘러 갈수록 감소하는 추세를 보이고 있으나, $k=30$ 일 때가 고장발생 강도가 가장 작기 때문에 효율적임을 알 수 있다.

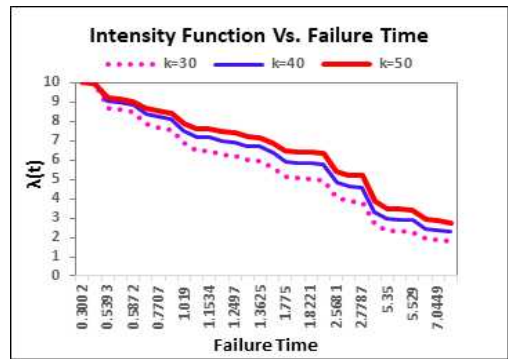


Fig. 4. Transition of Intensity Function $\lambda(t)$

Fig. 5는 참값에 대한 예측능력을 나타내는 평균값 함수에 대한 속성이 분석되었다. 이 그림에서 속성을 분석해보면, 형상모수 k 가 30, 40, 50일 때 모두 참값에 대해 과소 평가 패턴을 보이고 있으나, $k=30$ 일 때 참값에 대한 예측오차의 폭이 가장 작기 때문에 우수함을 알 수 있다.

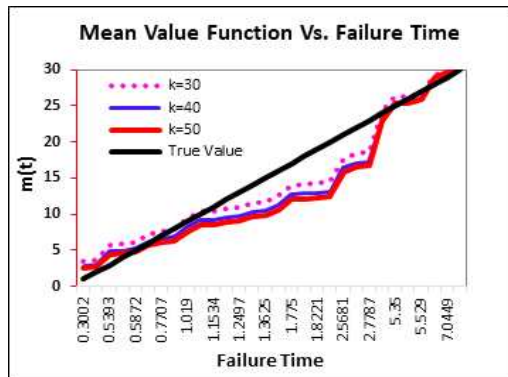


Fig. 5. Pattern of Mean Value Function $m(t)$

Fig. 6은 고장이 발생할 수 있는 미래의 임무시간에 대한 신뢰도 능력이 분석되었다. 이 그림에서 속성을 분석해보면, 형상모수 k 가 30, 40, 50일 때 임무시간이 흘러 갈수록 동일하게 유사한 추세를 보이고 있어서, 모두 효율적임을 알 수 있다[13].

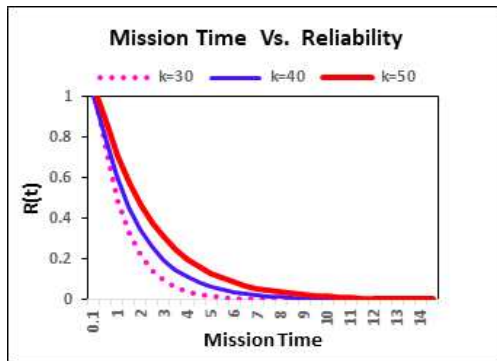


Fig. 6. Transition of Reliability $\hat{R}(t)$

즉, 단계별로 분석한 속성을 종합적으로 비교한 결과, 형상모수의 값이 작을수록 최적의 조건임을 확인할 수 있다.

5. 결론 및 향후 연구방향

소프트웨어 개발과정에서 소프트웨어 테스트시간이나 혹은 실행시간, 발생된 고장수와 고장발생 시간과의 관계를 효율적으로 관리할 수 있다면 소프트웨어 신뢰도를 성장시킬 수 있다. 그러므로, 소프트웨어 개발자들은 개발과정에서 이러한 속성들을 모형화한 후 사전에 분석할 수 있다면 좀더 신뢰성이 높은 제품들을 만들어 낼 수 있을 것이다. 따라서, 본 연구에서는 효율적인 신뢰성 평가와 신뢰성 향상을 위해 기존 연구사례가 없는 무한고장 NHPP Lomax 모형의 형상모수 변화에 따른 소프트웨어 신뢰성의 속성을 분석하였다.

본 연구의 결과는 다음과 같다.

첫째, 강도함수 측면에서 속성을 분석해보면 형상모수 k 가 30, 40, 50일 때 모두 고장시간이 흘러 갈수록 감소하는 추세를 보였으며, $k=30$ 일 때가 가장 작아서 효율적임을 알 수 있었다.

둘째, 평균값 함수의 측면에서 속성을 분석해보면 형상모수 $k=30$ 일 때 참값에 대한 오차의 폭이 가장 우수함을 알 수 있었다.

셋째, 미래의 임무시간에 대한 신뢰도 측면에서 속성을 분석해보면 형상모수 k 가 30, 40, 50일 때 고장시간이 흘러 갈수록 유사한 추이를 보이고 있어서 모두 효율적임을 알 수 있었다.

즉, 본 연구를 통해 형상모수 k 값이 작을수록 신뢰도 속성이 우수함을 확인할 수 있었다.

결론적으로, 기존 연구보다, 본 연구결과를 통하여 소프트웨어 개발자들은 형상모수에 근거한 소프트웨어 고장 형태를 고려함으로써 고장형태에 대한 사전정보를 파악하여 소프트웨어 신뢰도를 향상시킬 수 있다. 또한, 기존 연구사례가 없는 무한고장 NHPP Lomax 모형의 형상모수의 변화에 따른 신뢰도 속성에 관한 새로운 분석 기법을 제시하였으며, 소프트웨어 개발자들이 사전에 기본적인 정보로 활용할 수 있는 신뢰도 속성 정보도 제시할 수 있었다. 아울러, 같은 형태의 소프트웨어 고장시간 및 수명분포를 갖는 다른 무한고장 NHPP 신뢰성 모형들과의 속성 분석을 통해 가장 최적의 신뢰성 모형을 찾아내는 후속 연구가 필요하겠다.

REFERENCES

- [1] S. S. Gokhale, & K. S. Trivedi. (1999). A time/structure based software reliability model. *Annals of Software Engineering*, 8(3), 85-121.
- [2] A. L. Goel & K. Okumoto. (1978). Time-dependent fault detection rate model for software and other performance measures. *IEEE Trans. Reliab.*, 28(1), 206-211.
- [3] T. J. Yang. (2016). A Software Reliability Cost Mode Based on the Shape Parameter of Lomax Distribution. *The Journal of Korea Institute of Information Electronics and Communication Technology*, 9(2), 171-177.
- [4] H. C. Kim. (2018). A Study on the Reliability Attributes of the Software Reliability Model Following the Shape Parameter of Minimax Life Distribution. *The Journal of Korea Institute of Information Electronics and Communication Technology*, 11(4), 325-330.
- [5] T. J. Yang. (2018). A characteristic study on the software development cost model based on the lifetime distribution following the shape parameter of Type-2 Gumbel and Erlang distribution. *The Journal of Korea Institute of Information Electronics and Communication Technology*, 11(4), 460-466.
- [6] K. H. Rao, R. S. Prasad & R. L. Kantham. (2011). Software Reliability Measuring using Modified Maximum Like-lihood Estimation and SPC. *International Journal of Computer Applications*, 21(7), 1-5.

- [7] H. C. Kim. (2015). 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), 1-7.
- [8] T. J. Yang. (2018). A Study on Comparative Evaluation of Application of Software Reliability Model Dependent on Various Hazard Functions. *The Journal of Korea Institute of Information Electronics and Communication Technology*, 11(6), 800-806.
- [9] X. Teng and H. Pham. (2006). A new methodology for predicting software reliability in the random field environments. *IEEE Transactions on Reliability*, 55(3), 458-468.
- [10] K. S. Kim, & H. C. Kim (2014). The Comparative Study for Software Reliability Model Based on Finite and Infinite Failure Property using Rayleigh Distribution. *Journal of Digital Convergence*, 12(12), 277-284.
DOI : 10.14400/jdc.2014.12.12.277
- [11] K. S. Kim & H. C. Kim. (2013). The Assessing Comparative Study for Statistical Process Control of Software Reliability Model Based on Logarithmic Learning Effects. *Journal of Digital Policy & Management*, 11(12), 319-326.
DOI : 10.14400/jdpm.2013.11.12.319
- [12] H. C. Kim, & K. S. Kim (2015). Finite Time Prediction Capability Comparative Analysis of Software NHPP Reliability Model. *Journal of Digital Convergence*, 13(12), 143-149.
DOI : 10.14400/jdc.2015.13.12.143
- [13] H. J. Jung, & G. H. Han (2019). The Software Reliability Growth Model base on Software Error Datal. *Journal of the Korea Convergence Society*, 10(3), 59-65.

민 경 일 (Kyung-il Min)

[정회원]



- 1988년 2월 : 서울과학기술 대학교 기계공학과(공학사)
- 1990년 8월 : 인하대학교 생산공학과 (공학 석사)
- 2010년 8월 : 서경대학교 경영학과 (경영학 박사)

- 1983년 6월 ~ 2017년 2월 : 한국산업인력공단 지사장
- 2017년 3월~현재 : 남서울대학교 보건행정학과 교수
- 관심분야 : 정보통신 정책, 컴퓨터시스템 분석
- E-Mail : kimin85@nsu.ac.kr