
어랑 분포를 이용한 NHPP 소프트웨어 신뢰성장 모형에 관한 연구

김희철* · 최유순** · 박종구**

An Approach for the NHPP Software Reliability Model Using Erlang Distribution

Hee-Cheul Kim* · Yue-Soon Choi** · Jong-Goo Park**

이 논문은 2004년도 원광대학교의 교비 지원에 의해서 수행됨

요 약

비동질적인 포아송 과정에 기초한 모형들에서 잔존 결함 1개당 고장 발생률은 일반적으로 상수, 혹은 단조증가 및 단조 감소 추세를 가지고 있다. 본 논문에서는 잔존 결함 1개당 고장 발생률이 증가추세를 가진 어랑 분포를 이용한 어랑 모형을 제안하였다. 고장 간격시간으로 구성된 실측자료를 이용하여 기존의 모형과 어랑 모형의 모수 추정을 실시하였다. 어랑 모형의 형상모수를 선택하기 위하여 (누적)분포적합도 검정을 사용하였고 이 자료들에서 어랑 모형의 제안과 비교를 위하여 산술적 및 라플라스 검정, 적합도 검정, 편의 검정 등을 이용하였다.

ABSTRACT

The finite failure NHPP models proposed in the literature exhibit either constant, monotonic increasing or monotonic decreasing failure occurrence rates per fault. In this paper, we propose the Erlang reliability model, which can capture the increasing nature of the failure occurrence rate per fault. Equations to estimate the parameters of the Erlang finite failure NHPP model based on failure data collected in the form of inter-failure times are developed. For the sake of proposing shape parameter of the Erlang distribution, we used to the goodness-of-fit test of distribution. Data set, where the underlying failure process could not be adequately described by the existing models, which motivated the development of the Erlang model. Analysis of the failure data set which led us to the Erlang model, using arithmetic and Laplace trend tests, goodness-of-fit test, bias tests is presented.

키워드

소프트웨어 신뢰도 모형(Software Reliability Model), 적합도 검정(Goodness-of-fit Test), 비동질적인 포아송 과정(Nonhomogeneous Poisson Process), 어랑분포(Erlang Distribution), 편의 검정(Bias Tests), 소프트웨어 신뢰도(Software Reliability), 편차자승합(Sum of the Squared Errors).

I. 서 론

소프트웨어 신뢰성은 일정한 환경조건에서 일정기간

동안 고장이 나지 않고 운영 할 수 있는 확률이다. 따라서 이러한 소프트웨어 신뢰성은 시스템 신뢰도에 영향을 주는 중요한 요소가 되고 디자인 속성 측면에서는 하드웨어

* 남서울대학교 산업정보시스템공학부

** 원광대학교 컴퓨터공학과

신뢰성과는 다른 면을 가지고 있다. 소프트웨어의 다양한 기능은 소프트웨어 신뢰성 문제들에 관한 주요한 요인이 된다. 소프트웨어의 신뢰도의 일반적인 정의는 일정한 기간 동안 주어진 환경 하에서 컴퓨터 프로그램을 고장 없이 사용할 수 있는 확률을 의미한다. 어떤 프로그램이 코딩(coding)되었을 때 바로 테스트와 디버깅 작업이 수행된다. 이러한 단계에서 고장이 날 때까지 테스트된다. 테스트하다가 고장이 발견되면 즉시 고쳐진다. 이러한 고장들은 확률변수들인 T_1, T_2, \dots, T_n 의 관측치(realization)로 정의된다. 따라서 T_i 는 $(i-1)$ 번째 고장과 (i) 번째 사이의 고장간격시간을 의미한다.

본 논문에서는 수명분포가 어랑분포를 가지는 소프트웨어 신뢰성 모형을 고려하고자 한다. 어랑분포의 형상모수가 1과 2일 경우에는 각각 잘 알려진 Goel-Okumoto 모형과 Yamada-Ohba-Osaki 모형이 된다^[2,15]. 이러한 수명분포가 어랑분포를 이용할 경우에는 형상모수의 선택이 중요한 관건이다. 분석하고자 하는 소프트웨어 고장 자료를 이용하여 감마분포에서 형상모수가 정수인 어랑분포에 적합시켜 즉, 분포 적합도 검정을 시행하여 가능한 형상모수를 고려한 모형을 제시하고자 한다. 또, 와이블 형태의 특수한 형태인 랄리(Rayleigh)^[16]모형의 특성을 가진 일반 순서통계량에 기초한 소프트웨어 신뢰성 모형도 적용하였다.

이러한 모형의 적용은 소프트웨어 고장자료에 따른 강도함수(고장발생률) $\lambda(t)$ 형태에 의존된다.

본 논문의 내용은 다음과 같은 내용으로 서술되었다. 2절은 비동질적인 포아송과정에 대하여 알아보고 3절에서는 수명분포가 어랑분포를 따르는 일반 순서 통계량에 기초한 소프트웨어 신뢰성모형에 대하여 서술하였고 4절에서는 최우추정법을 이용한 모수 추정을 나열하고 5절에서는 수치적인 예로서 소프트웨어 고장자료 분석에 관한 내용으로 서술하고 6절에서는 모형비교에 따른 결론을 서술한다.

II. NHPP

소프트웨어 고장이 비동질적인 포아송과정(NHPP; non-homogeneous Poisson process)에 의존하는 소프트웨어 신뢰성 모형은 시간 영역 모형으로 분류된다^[14]. 이러한 확률과정의 모수인 $\lambda(t)$ 는 소프트웨어 수행시점 t 에

관련된 강도함수를 나타낸다. $N(t)$ 을 수행시점 t 까지의 누적 고장수를 나타내고 $m(t)$ 을 이것의 평균값 함수($m(t) = E[N(t)]$)으로 나타내면 $m(t)$ 와 강도함수 $\lambda(t)$ 은 다음과 같은 관계를 만족한다^[14].

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

따라서 $N(t)$ 는 모수 $m(t)$ 를 가진 포아송 확률밀도함수(Probability density function)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 시간관련 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 나타난다. 이러한 NHPP 모형들은 유한 고장 모형과 무한 고장 범주로 분류한다^[9].

유한 고장 NHPP 모형들은 충분한 테스트 시간이 주어지면 결함들(Faults)의 기대값이 유한 값($\lim_{t \rightarrow \infty} m(t) = \theta < \infty$)을 가지고 반면에 무한 고장 NHPP 모형들은 무한 값을 가진다고 가정 된다. 유한 고장 NHPP 모형에서 충분한 테스트 시간이 주어졌을 때 탐색되어 질 수 있는 결함의 기대값을 θ 라고 표현하고 $F(t)$ 를 분포함수라고 표현하면 유한 고장 NHPP모형의 평균값 함수는 다음과 같이 표현할 수 있다^[9,14].

$$m(t) = \theta F(t) \quad (3)$$

식 (3)으로부터 순간고장 강도함수(instantaneous failure intensity)는 다음과 같이 유도된다.

$$\lambda(t) = \theta F'(t) \quad (4)$$

식 (4)를 다음과 같이 변형하여 표기 할 수도 있다.

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

단, $h(t) = f(t)/(1-F(t))$ 는

위험함수(Hazard function, 고장률 함수)로서 소프트웨어 결함당 고장 발생률을 의미하고 $[\theta - m(t)]$ 은 t 시점에서 소프트웨어에 남아있는 결함들의 기대값을 나타낸다. $[\theta - m(t)]$ 의 값은 시점 t 에 대한 단조 비증가 함수(monotonically nonincreasing function)가 된다. 즉, 시간이 지남에 따라 결함들이 탐색되어 제거되기 때문에 감소성을 가진다. 따라서 $\lambda(t)$ 는 $h(t)$ 의 값에 따라 달라지며 상수, 증가, 감소 혹은 증가하다가 감소하는 패턴을 가질 수 있다.

이 분야의 기본적 모형인 Goel-Okumoto 모형은 $h(t)$ 가 정수 패턴을 가짐으로서 시점 t 에 독립이고 잘 알려진 S 모형(Yamada, Ohba-Osaki 모형)은 증가패턴을 가진다^[14].

여기에서는 소프트웨어 테스트 시점에서 미지의 N 개의 결함들을 가지고 있고 이 N 개의 결함들로부터 실수영역내에서 임의의 확률 밀도 함수 $f(\text{pdf})$ 에 따라 발생한 n 개의 순서 통계량이 고장 시점이 된다.

충분한 테스트 시간이 경과함에 따라 평균값 함수 $m(t)$ 가 유한값을 가지는 과정 ($\lim_{t \rightarrow \infty} m(t) = \theta < \infty$)을 NHPP(fin)라고 표시하면 식(3)으로부터 $F(t) = 1 - e^{-\beta t}$ 을 가지면 Goel-Okumoto[2]과정(지수 순서 통계량)이 되고, $F(t) = 1 - e^{-\beta t^\alpha}$ 이면 Goel [3]과정(와이블 순서 통계량)이 되며, $F(t) = 1 - (1 + \beta t) e^{-\beta t}$ 이면 Yamada, Ohba-Osaki^[15]과정이 된다. 이모형은 각 수리 시점에서 새로운 결함이 발생하지 않는다는 가정을 한다.

따라서 디버깅 과정에서 새로운 결함(faults)이 발생하면 일반 순서 통계량 모형을 기록값 통계량 모형으로 대체된다^[9]. 이러한 상황에서 $t \rightarrow \infty$ 에 따라 $m(t) \rightarrow \infty$ 로 되는 경우를 NHPP(Inf)로 표시하면 $m(t) = -\ln(1 - F(t))$ 가 된다. 이러한 관계로부터 $\lambda(t) = \alpha / (t + \beta)$ 가 되면 Musa-Okumoto^[15,6]과정이 되고 $\lambda(t) = \alpha \beta t^{\alpha-1}$ 가 되면 the Duane 과정, $\lambda(t) = \exp(\alpha + \beta t)$ 이면 Cox & Lewis [과정^[7]이 된다.

III. 어랑 소프트웨어 신뢰성 모형

본절에서는 우선 어랑분포(Erlang distribution)^[16]의 확률밀도함수와 누적 분포함수는 각각 다음과 같다.

$$f(t) = \frac{\beta^\alpha}{\Gamma(\alpha)} t^{\alpha-1} e^{-\beta t} \quad (6)$$

$$F(t) = \left(1 - \exp(\beta t) \sum_{j=0}^{\alpha-1} \frac{(\beta t)^j}{j!} \right) \quad (7)$$

단, 평균과 분산은 각각 $\alpha/\beta, \alpha/\beta^2$ 이고

$\alpha > 0, \alpha = 1, 2, 3, \dots, 0 \leq t \leq \infty, \beta > 0$ 을 만족한다. 따라서 식 (3),(4)와 관련하여 유한 고장 NHPP모형에서의 강도함수 $\lambda(t)$ 와 평균값 함수 $m(t)$, 위험함수 $h(t)$ 는 각각 다음과 같이 나타 낼 수 있다.

$$\lambda(t) = \theta f(t) = \theta \frac{\beta^\alpha}{\Gamma(\alpha)} t^{\alpha-1} e^{-\beta t} \quad (8)$$

$$m(t) = \theta F(t) = \theta \left(1 - \exp(\beta t) \sum_{j=0}^{\alpha-1} \frac{(\beta t)^j}{j!} \right) \quad (9)$$

$$h(t) = \left(\frac{\beta^\alpha}{\Gamma(\alpha)} t^{\alpha-1} \right) / \left(\sum_{j=0}^{\alpha-1} \frac{(\beta t)^j}{j!} \right) \quad (10)$$

위 함수에서 형상모수 $\alpha = 1$ 인 경우에는 Goel-Okumoto^[2]모형 되고, $\alpha = 2$ 이면 Yamada, Ohba-Osaki^[15]모형이 된다.

와이블 형태의 특수모형인 랄리(Rayleigh) 모형은 식(3)에서 분포함수가 랄리분포^[16]인 $F(t) = 1 - e^{-\beta t^2}$ 이면 랄리 모형으로 간주 될 수 있다.

따라서 식 (3),(4)와 관련하여 유한 고장 NHPP모형에서의 강도함수 $\lambda(t)$ 와 평균값 함수 $m(t)$, 위험함수 $h(t)$ 는 각각 다음과 같이 나타 낼 수 있다.

$$\lambda(t) = \theta [1 - \exp(\beta t^2)]$$

$$m(t) = 2\theta\beta t e^{-\beta t^2}$$

$$h(t) = 2\beta t$$

그리고 테스트 시점 t 에서 소프트웨어 고장이 일어난다고 하는 가정 하에서 신뢰 구간 $(t, t+x)$ (단, x 는 임무시간(Mission Time))사이 소프트웨어의 오류가 일어나지 않을 조건부 확률인 소프트웨어 신뢰도는 다음과 같다^[12,17].

$$R(x | t) = \exp\{-\{m(t+x) - m(t)\}\}, \quad (t \geq 0, x \geq 0) \quad (11)$$

IV. 최우추정법을 이용한 모수 추정

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

$$x_k = \sum_{i=1}^k t_k \quad (k=1,2,\dots,n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n) \quad (12)$$

이라고 하면 데이터 집합 D_t 는 $\{n, x_1, x_2, \dots, x_n; t\}$ 와 같이 구성된다. n 번째까지 고장시점이 관찰된 고장 절단 모형일 경우에 데이터 집합 D_{x_n} 은 $\{x_1, x_2, \dots, x_n\}$ 으로 구성된다. 이 시간 절단 모형에서의 우도함수는 다음과 같이 알려져 있다^[9,12].

$$L_{NHPP}(\theta \mid D_t) = \left(\prod_{i=1}^n \lambda(x_i) \right) \exp(-m(t)) \quad (13)$$

단, θ 는 미지의 모수 집합을 의미하고 우도함수 (13) 식에서 t 을 x_n 으로 대치하면 유사한 형태의 고장 절단 모형의 우도함수가 된다^[9,14]. 식(13)과 (8), (9)를 연관하면 유사한 NHPP 모형에 대한 우도함수는 다음과 같은 형태로 표현할 수 있다.

$$L_{NHPP,n}(\theta, \theta \mid D_t) = \left(\prod_{i=1}^n \theta f(x_i) \right) \exp(-\theta F(x_n)) \quad (14)$$

따라서 t 을 x_n 으로 대치한 고장절단 모형에서의 최우 추정법(MLE)을 이용하기 위한 어랑 모형 로그우도함수(Log likelihood function)는 다음과 같이 유도된다.

$$\begin{aligned} L(\theta, \alpha, \beta \mid D_{x_n}) = & n \log \theta - n \log \Gamma(\alpha) + n \alpha \log \beta + (\alpha - 1) \sum_{i=1}^n \log x_i \\ & - \beta \sum_{j=1}^n x_j - \theta + \theta e^{-\beta x_n} \left(\sum_{j=0}^{\alpha-1} \frac{(\beta x_n)^j}{j!} \right) \end{aligned} \quad (15)$$

식(15)에서 형상모수 α 는 알고 있다고(상수)가정하고 θ 와 β 에 관한 편미분 시행한 수식은 다음을 만족한다.

$$\frac{n}{\theta} = 1 - e^{-\beta x_n} \left(\sum_{j=0}^{\alpha-1} \frac{(\beta x_n)^j}{j!} \right) \quad (16)$$

$$\frac{\alpha n}{\beta} = \sum_{i=1}^n x_i - \frac{\theta \left[\theta e^{-\beta x_n} \left(\sum_{j=0}^{\alpha-1} \frac{(\beta x_n)^j}{j!} \right) \right]}{\theta \beta} \quad (17)$$

식(16)과 (17)을 수치해석적 방법을 이용하면 $\hat{\beta}_{MLE}$ 와 $\hat{\theta}_{MLE}$ 의 값을 계산 할 수 있다.

이 분야의 기본적인 모형인 Goel-Okumoto 모형^[2]은 위 식에서 $\alpha=1$ 인 경우로서 다음과 같이 알려져 있다.

$$\frac{n}{\theta} = 1 - e^{-\beta x_n} \quad (18)$$

$$\frac{n}{\beta} = \sum_{i=1}^n x_i + \theta x_n e^{-\beta x_n} \quad (19)$$

같은 방법으로 식(18)과 (19)를 수치해석적 방법을 이용하여 풀면 $\hat{\beta}_{MLE}$ 와 $\hat{\theta}_{MLE}$ 의 값을 계산 할 수 있다.

또, $\alpha=2$ 인 경우는 S형 모형(Yamada, Ohba-Osaki model^[15])인 경우는 다음과 같다.

$$\frac{n}{\theta} = 1 - (1 + \beta x_n) e^{-\beta x_n} \quad (20)$$

$$\frac{2n}{\beta} = \theta \beta x_n^2 e^{-\beta x_n} + \sum_{i=1}^n x_i \quad (21)$$

또, 본 연구에서 제시하고자 하는 $\alpha=3$ 인 경우는 다음과 같이 나타낼 수 있다.

$$\frac{n}{\theta} = 1 - \left(1 + \beta x_n + \frac{x_n^2}{2} \beta^2 \right) e^{-\beta x_n} \quad (22)$$

$$\frac{3n}{\beta} = \theta \beta^2 \frac{x_n^3}{2} e^{-\beta x_n} + \sum_{i=1}^n x_i \quad (23)$$

유사한 방법으로 식(22)와 (23)을 수치해석적 방법을 이용하여 $\alpha=3$ 인 경우에도 $\hat{\beta}_{MLE}$ 와 $\hat{\theta}_{MLE}$ 의 값을 계산 할 수 있다.

유사한 방법으로 t 을 x_n 으로 대치한 고장절단 모형에서의 최우추정법(MLE)을 이용하기 위한 랄리 모형 로그우도함수(log likelihood function)는 다음과 같이 유도된다.

$$\begin{aligned} \ln L(\theta, \beta \mid D_{x_n}) & = n \log \theta + n \log \beta + n \log 2 + \sum_{i=1}^n \log x_i \\ & - \beta \sum_{i=1}^n x_i^2 - \theta + \theta e^{-\beta x_n^2} \end{aligned} \quad (24)$$

최우추정법을 이용하기 위하여 식(24)를 θ 와 β 에 대하여 편미분을 하면 다음과 같은 식을 유도 할 수 있다.

$$\frac{n}{\theta} = 1 - \exp(-\beta x_n^2) \tag{25}$$

$$\frac{n}{\beta} = \sum_{i=1}^n x_i^2 + x_n^2 \theta e^{-\beta x_n^2} \tag{26}$$

같은 방법으로 식(25)와 (26)을 수치해석적 방법을 이용하여 풀면 $\hat{\beta}_{MLE}$ 와 $\hat{\theta}_{MLE}$ 의 값을 계산 할 수 있다.

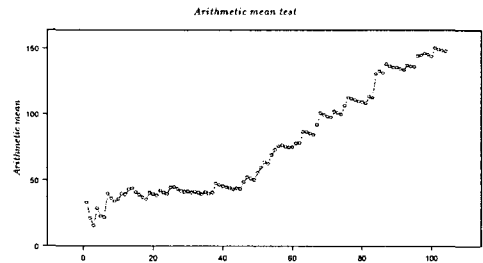
V. 소프트웨어 고장자료분석

본 절에서는 104번의 고장 간격자료인 DS3^[8](K.Kanoun 과 J.C.Laprie)를 이용하고자 한다. 이 소프트웨어 고장자료는 <표 1>과 같다.

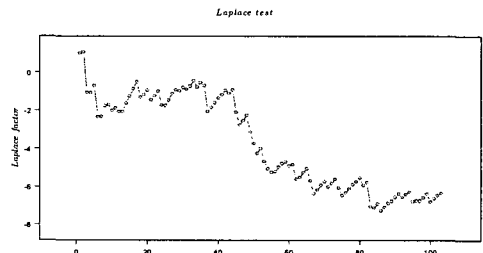
<표 1> 소프트웨어 고장 자료
<Table 1> Failure data of software

Failure number	Failure Time(hour)	Failure Time $\times 10^{-2}$	Failure number	Failure Time(hour)	Failure Time $\times 10^{-2}$
1	33	0.33	53	3305.5	33.055
2	42	0.42	54	3705.5	37.055
3	46	0.46	55	3999.5	39.995
4	112	1.12	56	4226.5	42.265
5	112.5	1.125	57	4344.5	43.445
6	130.5	1.305	58	4357.5	43.575
7	279.5	2.795	59	4404.5	44.045
8	293.5	2.935	60	4493.5	44.935
9	308.5	3.085	61	4735.5	47.355
10	358.5	3.585	62	4834.5	48.345
11	439.5	4.395	63	5441.5	54.415
12	473.5	4.735	64	5524.5	55.245
13	558.5	5.585	65	5526.5	55.265
14	612.5	6.125	66	5552.5	55.525
15	615.5	6.155	67	6138.5	61.385
16	630.5	6.305	68	6846.5	68.465
17	636.5	6.365	69	6852.5	68.525
18	644.5	6.445	70	6856.5	68.565
19	774.5	7.745	71	6911.5	69.115
20	793.5	7.935	72	7320.5	73.205
21	812.5	8.125	73	7356.5	73.565
22	924.5	9.245	74	7371.5	73.715
23	939.5	9.395	75	7944.5	79.445
24	955.5	9.555	76	8527.5	85.275
25	1109.5	11.095	77	8587.5	85.875
26	1159.5	11.595	78	8606.5	86.065
27	1169.5	11.695	79	8626.5	86.265
28	1171.5	11.715	80	8705.5	87.055
29	1193.5	11.935	81	8729.5	87.295
30	1246.5	12.465	82	9269.5	92.695
31	1265.5	12.655	83	9321.5	93.215
32	1323.5	13.235	84	10917.5	109.175
33	1343.5	13.435	85	11231.5	112.315
34	1346.5	13.465	86	11232.5	112.325
35	1438.5	14.385	87	11995.5	119.955
36	1443.5	14.435	88	12005.5	120.055
37	1509.5	15.095	89	12025.5	120.255
38	1798.5	17.985	90	12169.5	121.695
39	1801.5	18.015	91	12197.5	121.975
40	1810.5	18.105	92	12253.5	122.535
41	1822.5	18.225	93	12729.5	127.295
42	1840.5	18.405	94	12794.5	127.945
43	1849.5	18.495	95	12892.5	128.925
44	1924.5	19.245	96	13776.5	137.765
45	1939.5	19.395	97	13988.5	139.885
46	2230.5	22.305	98	14275.5	142.755
47	2442.5	24.425	99	14328.5	143.285
48	2446.5	24.465	100	14331.5	143.315
49	2451.5	24.515	101	15162.5	151.625
50	2759.5	27.595	102	15205.5	152.055
51	3028.5	30.285	103	15260.5	152.605
52	3304.5	33.045	104	15369.5	153.695

제시하는 신뢰모형들을 분석하기 위하여 우선 자료에 대한 추세 검정이 선행 되어야 한다. 추세 분석에는 산술 평균 검정(arithmetic mean test)과 라플라스 추세 검정(Laplace trend test)^[14]등이 있다. 이 검정을 실시한 결과 그림 <그림 1>에서 산술평균 검정결과 고장수가 증가함에 따라 산술 평균이 거의 증가 추세를 보이고 있으므로 신뢰성장(reliability growth)이 되고 있음을 나타내고 있고 라플라스 추세 검정의 결과도 라플라스 요인(factor)이 음수로서 감소하기 때문에 역시 신뢰성장(reliability growth)이 되고 있음을 나타내고 있다. 따라서 이 자료를 가지고 신뢰성장 모형을 제시하는 것이 효율적임을 시사하고 있다^[14,8].



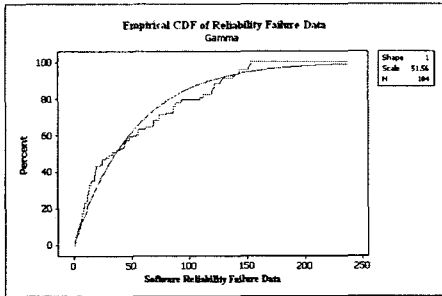
<그림 1> 산술평균 검정
<Fig. 1> Arithmetic mean test



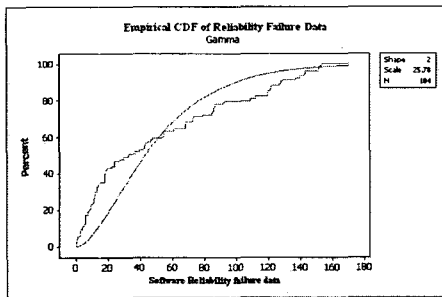
<그림 2>라플라스 추세 검정
<Fig. 2> Laplace trend test

모수추정을 용이하게 하기 위하여 척도 변경자료(고장시간 $\times 10^{-2}$ 만큼 이동 시킨 자료)를 이용하여 분석하였다. 이 자료를 이용하여 어랑 분포의 형상모수를 선택하기 위하여 어랑 분포를 가정한 분포 적합성 검정은 <그림 3>, <그림 4> 및 <그림 5>에 나타나 있고 이 그림에서 사용된 형상모수 $\alpha = 1, 2, 3$ 일 경우는 척도모수는 다르지만 분포유형이 비슷한 경향을 보이고 있으므로 본 분석에서는 형상모수가 1, 2, 3인 경우를 고려 하고자 한다.

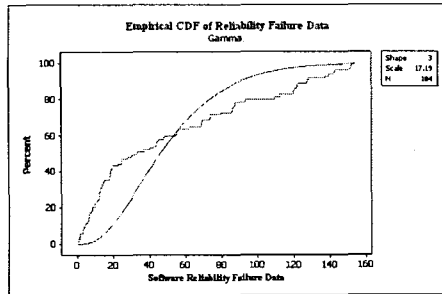
따라서 소프트웨어 신뢰성 모형의 모수 추정에는 최우추정법을 이용하였고 비선형 방정식의 계산방법은 수치해석적 기본 방법인 이분법(bisection method)^[12]을 사용하였다. 따라서 소프트웨어 신뢰성 모형의 모수 추정에는 최우추정법을 이용하였고 비선형 방정식의 계산방법은 수치해석적 기본 방법인 이분법(Bisection method)을 사용하였다. 이러한 계산은 초기값을 10^{-6} 와 10 을, 허용 한계(tolerance for width of interval)는 10^{-10} 을 주고 수렴성을 확인 하면서 충분한 반복 횟수인 100번을 C-언어를 이용하여 모수 추정을 수행한 모수의 추정값들의 결과는 <표 2>와 같다.



<그림 3> 형상모수 $\alpha=1$ 인 경우의 적합성 검정
<Fig. 3> Goodness-of-fit test of $\alpha = 1$



<그림 4> 형상모수 $\alpha=2$ 인 경우의 적합성 검정
<Fig. 4> Goodness-of-fit test of $\alpha = 2$



<그림 5> 형상모수 $\alpha=3$ 인 경우의 적합성 검정
<Fig. 5> Goodness-of-fit test of $\alpha = 3$

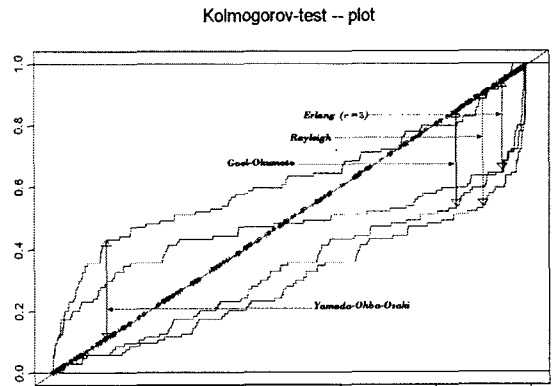
<표 2> 각 모형에 대한 모수의 추정값
<Table 2> Parameter estimation of each models

Model	$\hat{\theta}_{MLE}$	$\hat{\beta}_{MLE}$
Goel-Okumoto	102.761	0.0045
Rayleigh	103.104	0.0028
Yamada-Ohba-Osaki	107.294	0.0078
Erlang($\alpha = 3$)	103.012	0.0092

모형 선택의 하나의 방법으로 편차자승합(SSE^[12])을 이용할 수 있는데 이 편차자승합이 작으면 상대적으로 효율적인 모형이 된다. 주어진 자료를 이용하여 제시된 모형들에 대한 편차자승합의 값은 <표 3>에 나타내었다. <표 3>에서 Erlang($\alpha=3$)모형이 이 분야에서 기존에 알려진 모형인 Yamada-Ohba-Osaki 모형이나 Goel-Okumoto 모형에 비해 상대적으로 효율적 모형인 것을 확인할 수 있다.

<표 3> 모형들에 대한 SSE 의 값
<Table 3> SSE value of each models

Model	Goel-Okumoto	Rayleigh	Yamada-Ohba-Osaki	Erlang($\alpha=3$)
SSE	35621.58	54452.72	31835.49	30215.05
AIC	814.9768654	859.1125069	803.2903928	797.8572705



<그림 6> Kolmogorov 검정
<Fig. 6> Kolmogorov test

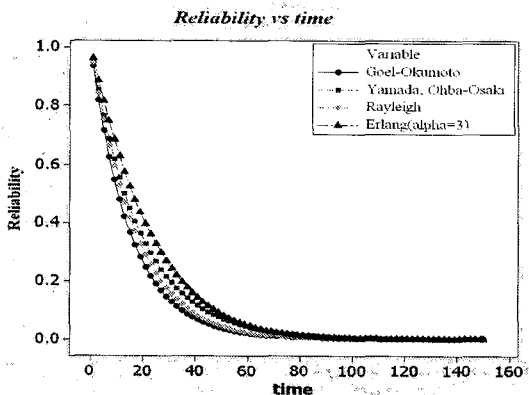
예측 오류(prediction error)의 비정상성(nonstationarity)에 대한 측도는 Kolmogorov 거리(distance)^[14,4,8]로 측정되는데 이 거리가 클수록 상대적으로 비정상성을 내포하고 있다. <그림 6> 은 S-Plus 소프트웨어^[13]를 이용하여

Kolmogorov 검정에 대한 그림을 보여주고 있고 이 그림에서도 전체적으로 Erlang($\alpha=3$)모형이 상대적으로 비정상성 속성이 덜 내포하고 있음을 알 수 있다. <표 4>는 Kolmogorov 거리를 요약한 표로서 모형에 대한 치우침(bias)을 나타내고 있다. 이 표에서도 Erlang($\alpha=3$)모형의 형태가 상대적으로 비정상성 속성이 덜 내포하고 있음을 알 수 있다.

<표 4> Kolmogorov 거리
<Table 4> Kolmogorov distance

Model	Kolmogorov distance
Goel-Okumoto	0.351321
Rayleigh	0.361672
Yamada-Ohba-Osaki	0.321314
Erlang ($\alpha = 3$)	0.317219

<그림 7>의 신뢰도 그림에서도 Goel-Okumoto 모형, Yamada-Ohba-Osaki 모형, Rayleigh 모형 및 Erlang($\alpha=3$)모형에 대한 신뢰도를 추정한 결과도 신뢰도는 처음시간에는 비슷하다가 임무 시간이 흐를수록 근소한 차이지만 Erlang($\alpha=3$)모형이 높아지는 패턴을 볼 수 있다.



<그림 7> 임무시간에 따른 신뢰도
<Fig. 7> Reliability of mission time

VI. 결론

소프트웨어 신뢰성은 개발의 최종단계에 있는 테스트

공정이나 실제 사용단계에 있어서 소프트웨어 내에 존재하는 고장 수나 고장 발생 시간에 의해서 효과적으로 평가할 수 있는 상황으로 그 평가기술이 중요하게 된다. 따라서 소프트웨어 개발의 테스트 공정이나 실제사용단계에 있어서 고장 발생 환경이나 고장 발생 현상을 수리적으로 모형화가 가능하면 평가를 할 수 있다. 테스트 시간이나 혹은 실행 시간, 발생된 고장 수와 고장 발생시간과의 관계를 효율적으로 관리함으로써 소프트웨어 신뢰도를 성장시킬 수 있다. 이러한 과정을 소프트웨어 성장과정이라고 볼 수 있다.

본 논문에서는 잔존 결함 1개당 고장 발생률이 증가추세를 가진 어랑 분포를 이용한 Erlang($\alpha=3$) 모형을 제안하였다. 고장 간격시간으로 구성된 자료를 이용하여 기존의 모형과 어랑 모형 및 랠리 모형에 대하여 최우 추정법을 이용하여 모수 추정을 실시하였다. 어랑 모형에 대한 형상모수의 가능성을 위하여 분포의 적합도 검정을 실시하고 고장수가 비교적 큰 자료(고장수가 104)를 통하여 분석하였다.

분석의 결과에서는 인용한 자료를 바탕으로 편차자승합을 이용한 결과로서 Erlang($\alpha=3$)모형이 기존에 잘 알려진 Yamada-Ohba-Osaki 모형이나 Goel-Okumoto보다 우수함을 보이고 있고 콜모고로프 거리의 결과도 Erlang($\alpha=3$) 모형이 상대적으로 정상성에 가까운 모형으로 간주할 수 있다. 분석된 자료에 대한 평가에서도 산술평균 검정과 라플라스 추세 검정을 실시한 결과도 신뢰성장이 되고 있음을 확인하였다.

참고문헌

- [1] A.A.Abdel-Ghally, P.Y.Chan, and B.Littlewood. "Evaluation of competing Software Reliability Predictions". *IEEE Trans. on Software Engineering*, SE-12(9):538-546, Sept. 1989.
- [2] A. L. Goel and K. Okumoto. "Time-Dependent Error-Detection Rate Models for Software Reliability and Other Performance Measures". *IEEE Trans. on Reliability*, R-28(3):206-211, Aug. 1979.
- [3] A. L. Goel. "Software Reliability Models: Assumptions, Limitations and Applicability". *IEEE Trans. on Software Engineering*, SE-11(12):1411-1423,

Dec. 1985.

[4] A. P. Nikora and M. R. Lyu. *Handbook of Software Reliability Engineering*, M.R.Lyu, Editor, chapter *Software Reliability Measurement Experience*, pp.255-301. MacGraw-Hill, New York, 1996.

[5] J. D. Musa. "A Theory of Software Reliability and its Application". *IEEE Trans. on Software Engineering*, SE-1(1):312-327, Sept. 1975.

[6] J. D. Musa, A. Iannino and K. Okumoto. *Software Reliability: Measurement, Prediction, Application*. McGraw Hill, New York, 1987.

[7] J. F. Lawless. *Statistical Models and Methods for Lifetime Data*. John Wiley & Sons, New York, 1981.

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

[9] L. Kuo and T. Y. Yang. "Bayesian Computation of Software Reliability". *Journal of the American Statistical Association*, Vol.91, pp.763-773, 1996.

[10] M. R. Lyu. *Handbook of Software Reliability Engineering*, M.R.Lyu, Editor, chapter *Introduction*, pp.3-25. MacGraw-Hill, New York, 1996.

[11] O. Gauodin. "Optimal Properties of the Laplace Trend Test for Software-Reliability Models". *IEEE Trans. on Reliability*, 41(4):525-532, Dec. 1992.

[12] H. Pham and L. Nordmann and X. Zhang "A General Imperfect-Software -Debugging Model with S-Shaped Fault-Detection Rate". *IEEE Trans. on reliability*, VOL, 48, NO 2, 1999.

[13] S. Selvin. *Modern Applied Biostatistical Methods Using S-Plus*, pages 141-184. Oxford University Press, New York, 1998.

[14] S. S. Gokhale and K. S. Trivedi. "A time/structure based software reliability model". *Annals of Software Engineering*, 8, 85-121. 1999

[15] S. Yamada, M. Ohba and S. Osaki. "S-Shaped Reliability Growth Modeling for Software Error Detection". *IEEE Trans. on Reliability*. R-32(5):475-485, Dec. 1983.

[16] V. K. Rohatgi. *Statistical inference*, pages 398-416. JOHN WILEY & SONS, INCI, New York, 1984.

[17] W. Farr. *Handbook of Software Reliability Engineering*, M.R.Lyu, Editor, chapter *Software Reliability Modeling Survey*, pages 71-117. McGraw-Hill, New York, NY, 1996.

[18] W. Q. Meeker and L. A. Escobar. *Statistical Methods for Reliability Data*, pages 98-101. JOHN WILEY & SONS, INCI, New York, 1998.

[19] <http://www.answers.com/gamma%20distribution> (site visited 2005-5-15)

저자소개

김희철(Hee-CheulKim)



1992년 동국대학교 통계학과 석사
1998년 동국대학교 통계학과 박사
2000년3월 ~ 2004년 2월 송호대학
정보산업계열 조교수
2005년 ~ 현재 남서울대학교 산업정
보시스템공학부 전임강사

※ 관심분야 : 소프트웨어 신뢰성공학, 웹프로그래밍,
전산통계, 인터넷 비즈니스

최유순(Yue-SoonChoi)



1986년 원광대학교 컴퓨터공학과 학사
1990년 원광대학교 컴퓨터공학과 석사
2000년 원광대학교 컴퓨터공학과
박사과정 수료
2001 ~ 현재 원광대학교 강의전담교수

※ 관심분야 : 소프트웨어공학, 컴포넌트 소프트웨어,
웹프로그래밍, 웹서비스

박종구(Jong-GooPark)



1969년 동국대학교농업경제학과 학사
1975년 동국대학교 전자정보처리학
과 석사
1999년 동국대학교 통계학과 박사
1981년 ~ 현재 원광대학교
컴퓨터공학과 교수

※ 관심분야 : 전문가시스템, 소프트웨어공학, 소프트웨
어 신뢰성 공학, 모바일 프로그래밍