

Kappa(2) NHPP에 의한 소프트웨어 신뢰성 모형에 관한 연구 (The study for NHPP Software Reliability Model based on Kappa(2) distribution)

김희철(Hee-Cheul Kim)¹⁾

요 약

유한고장수를 가진 비동질적인 포아송 과정에 기초한 모형들에서 잔존 결함 1개당 고장 발생률은 일반적으로 상수, 혹은 단조증가 및 단조 감소 추세를 가지고 있다. 본 논문에서는 기존의 소프트웨어 신뢰성 모형인 Goel-Okumoto 모형과 Yamada-Ohba-Osaki 모형을 재조명하고 잔존 결함 1개당 고장 발생률이 단조 감소 추세를 가진 2모수 Kappa 분포를 이용한 Kappa모형을 제안하였다. 고장 간격시간으로 구성된 자료를 이용한 모수추정 방법은 최우추정법과 일반적인 수치해석 방법인 이분법을 사용하여 모수 추정을 실시하고 효율적인 모형 선택은 편차자승합과 콜모고로프 거리를 적용하여 모형들에 대한 효율성 입증방법을 설명하였다. 소프트웨어 고장 자료 분석에서는 고장수가 비교적 큰 실측 자료(고장수가 86)인 Allen P.Nikora 와 Michael R.Lyu 가 인용한 SYS2 자료를 통하여 분석하였다. 이 자료들에서 카파 모형의 비교를 위하여 산술적 및 라플라스 검정, 편의 검정등을 이용하였다.

ABSTRACT

Finite failure NHPP models presented in the literature exhibit either constant, monotonic increasing or monotonic decreasing failure occurrence rates per fault. In this paper, Goel-Okumoto and Yamada-Ohba-Osaki model was reviewed, proposes the Kappa(2) reliability model, which can capture the nonmonotonic decreasing nature of the failure occurrence rate per fault. Algorithm to estimate the parameters used to maximum likelihood estimator and bisection method, model selection based on sum of the squared errors and Kolmogorov distance, for the sake of efficient model, was employed. Analysis of failure using real data set, SYS2(Allen P.Nikora and Michael R.Lyu), for the sake of proposing two parameter of the Kappa distribution, was employed. This analysis of failure data compared with the Kappa model and the existing model using arithmetic and Laplace trend tests, bias tests is presented.

키워드(Keywords) : 소프트웨어 신뢰도 모형(Software Reliability Model), 비동질적인 포아송 과정(Nonhomogeneous Poisson Process), 카파분포(Kappa Distribution), 편의 검정(Bias Tests), 소프트웨어 신뢰도(Software Reliability), 편차자승합(Sum of the Squared Errors), 콜모고로프 거리(Kolmogorov distance)

논문접수 : 2005. 11. 15.

심사완료 : 2005. 11. 30.

1) 정희원 : 남서울대학교 산업경영공학과 전임강사

1. 서론

현대사회에 있어서 인간이 의존하고 있는 시스템의 대부분은 소프트웨어 시스템(System)을 포함하고 있으며 이러한 시스템의 기능이 확대되어 소프트웨어가 대규모화, 복잡화, 다양화하고 수효량도 급속히 증대하고 있다. 그러나 시스템의 고장(Failure)이 발생하면 그 고장을 찾아 제거하는 디버깅 과정을 거치면서 소프트웨어에 잔존하는 결함의 수는 점차 감소하며 고장이 발생하는 시간 간격이 점차 증가함으로 소프트웨어 신뢰성은 성장한다.

소프트웨어 테스트 단계에서 소프트웨어 고장수(Number of failure)와 고장간격시간에 의해 소프트웨어 고장현상을 수리적으로 모형화하면 소프트웨어에 대한 평가를 쉽게 할 수 있으며 신뢰도 모형에 의해 소프트웨어 고장수, 소프트웨어 고장발생간격시간, 소프트웨어 신뢰도 및 고장률 등의 신뢰성 평가측도들이 추정되어 미래의 고장시간을 예측할 수도 있다.

소프트웨어 고장시간은 수명자료가 된다. 따라서 비음(Nonnegative)의 값을 가지기 때문에 이 분야에서는 주로 지수분포, 와이블분포, 감마분포 등 일반화 감마 분포가 많이 사용되어 왔다. 그러나 이러한 감마분포 외에도 통계자료 해석시 자료를 대수변환 이 후에 정규분포로 처리하면 되는 대수정규분포(Lognormal distribution), 이 분포와 유사한 대수로지스틱 분포(Loglogistic distribution)도 이 분야에 사용이 가능하다[1]. 이 분야의 기본적 모형인 Goel-Okumoto 모형은 위험함수($h(t)$)가 정수 패턴을 가짐으로서 시점 t 에 독립이고 잘 알려진 S 모형(Yamada, Ohba-Osaki 모형)은 단조 증가패턴을 가진다[2, 3]. 따라서 본 연구는 보다 많은 분포 정보가 함축되어 있고 여러 분야에서 널리 사용되며 위험함수가 단조 감소 패턴을 가진 2모수 카파분포[4, 5]를 도입하여 NHPP 모형에 대한 신뢰성 척도를 추정하고 이를 바탕으로 카파 모형의 효율성과 그 특성을 알아보려고 한다.

본 논문의 2장에서는 관련연구로서 유한 고장 NHPP 모형에 대하여 서술하였고 3장에서는 카파 신뢰성 모형에 대하여 알아보았고 4장에서는 최우추정법을 이용한 모수추정에 대하여 설명하고 5장에서는 실측 소프트웨어 고장자료인 Allen P.Nikora 와 Michael R.Lyu 가 인용한 SYS2 자료[6]를 이용하여 각 모형에 대한 모수추정 및 모형비교를 실시하였으며 마지막으로 5장에서는 결론을 나열 하였다.

2. 관련연구

신뢰도에서 관측시간 $(0, t)$ 사이에 발견된 고장수 $N(t)$ 를 모형화 하는데 비동질적 포아송 과정(Non-homogeneous poisson process; NHPP)이 널리 사용하여 왔다. 이 과정(Process)에서 강도함수(Intensity function) 혹은 고장 발생률(Rate of occurrence of failure; ROCOF) $\lambda(t) = dE[N(t)]/dt$ 는 t 에 대한 단조(Monotonic)함수로 흔히 가정한다[6,7]. 이 범주에서 지금까지 알려진 모형들은 Goel-Okumoto 모형, Weibull 모형 그리고 Cox-Lewis 모형등이 있는데 이 모형들에 대한 강도함수는 각각 시간에 의존한 함수, 멱(Power) 함수, 대수선형(Log-linear) 함수를 가정하였다[7].

NHPP 모형에서 평균값 함수 $m(t)$ (Mean value function)와 강도 함수 $\lambda(t)$ (Failure intensity)는 다음과 같은 관계로 표현 할 수 있다.

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

$N(t)$ 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 설명

이 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[7]. 유한 고장(Finite failure) 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(Infinite failure) 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 표현 할 수 있다.

$$m(t) = \theta F(t) \tag{3}$$

(3) 식으로 부터 강도함수 $\lambda(t)$ 는 다음과 같이 유도된다.

$$\lambda(t) = \theta F'(t) \tag{4}$$

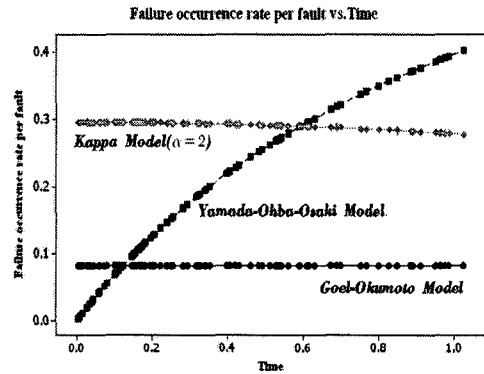
(4) 식을 다음과 같이 변형하여 표기 할 수도 있다.

$$\lambda(t) = [\theta - m(t)] \frac{F'(t)}{1 - F(t)} = [\theta - m(t)] h(t) \tag{5}$$

단, $h(t) = \frac{F'(t)}{1 - F(t)}$ 는 위험함수(Hazard function, 고장률 함수)으로서 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 그 고장을 찾아 제거하는 디버깅 과정을 거치면서 제거되기 때문에 감소성을 가진다.

$\lambda(t)$ 는 $h(t)$ 의 속성에 따라 달라지며 그 추세는 상수나 증가, 혹은 감소(증가)하다가 증가

(감소)하는 패턴을 가질 수 있다. 이 분야의 기본적인 모형인 Goel-Okumoto 모형은 $h(t)$ 가 정수 패턴을 가짐으로서 시점 t 에 독립이고 잘 알려진 S 모형(Yamada, Ohba-Osaki 모형)은 단조 증가 패턴을 가진다[1].



[그림 1] 각 모형에 대한 위험함수 [Fig. 1] Hazard function of each model

3. 2 모수 카파(Kappa) 모형

소프트웨어 신뢰성이나 의료정보분야 등 여러 분야에서 널리 사용될 수 있는 분포 중에 하나가 3모수(α, β, θ) 카파 분포이다.

$$f_{kappa(3)}(t) = \frac{\alpha\gamma}{\beta} \left(\frac{t}{\beta}\right)^{\gamma-1} [\alpha + (t/\beta)^\alpha \gamma]^{-(\alpha+1)/\alpha}$$

단, 시간자료는 $t(>0)$ 이고 $\beta(>0)$ 는 척도모수이고 $\alpha(>0)$ 와 $\gamma(>0)$ 는 형상모수이다.

로그노오말(Lognormal)분포나 감마(Gamma)분포의 누적분포함수가 폐쇄형(Closed form)을 이루지 못하지만 이 카파분포는 폐쇄형을 이루고 있으므로 순서 통계량(Order statistics)의 정보를 설명하는데 편리하다는 장점을 가지고 있다[5].

따라서 본 논문에서의 소프트웨어 신뢰성 모형도 일반 순서 통계량에 의존되는 모형이므로 이 분포를 적용하는 것이 합리적이고 3모수 카파분포에서 형상모수 $\gamma=1$ 인 2모수 카파분포도 3모수 카파 분포의 성질을 그대로 유지하고

있기 때문에 본 논문에서 2모수 카파분포를 적용하고자 한다[4].

2모수 카파 분포의 확률밀도함수와 누적분포 함수는 다음과 같다[5].

$$f_{kappa(2)}(t) = \alpha/\beta [\alpha + (t/\beta)^\alpha]^{-(\alpha+1)/\alpha} \quad (6)$$

$$F_{kappa(2)}(t) = [(\alpha/\beta)^\alpha / (\alpha + (t/\beta)^\alpha)]^{1/\alpha} \quad (7)$$

(6)식과 (7)식을 유한 고장 NHPP 모형에 적용시킨 평균값 함수와 강도함수는 다음과 같다.

$$m(t) = \theta F_{kappa(2)}(t) \quad (8)$$

$$\lambda(t) = \theta F'(t) = \theta f_{kappa(2)}(x) \quad (9)$$

[그림 1] 은 각 모형에 대한 위험함수를 나타낸 그림이다. 이 모형의 모수들은 [표 1]에 있는 고장 간격 데이터(102594 × 10⁻⁵ 시간(Second) 단위에서 고장이 86번 일어남)을 이용하여 추정된 결과를 이용하였다. Goel-Okumoto 모형은 위험함수가 정수 패턴을 가짐으로서 시점 t에 독립이고 Yamada, Ohba-Osaki 모형은 단조 증가 패턴을 가지고 있고 Kappa(2) 모형은 단조 감소 패턴을 확인할 수 있다.

4. 신뢰성 모형에 대한 모수 추정

시간(0,t)까지 조사하기 위한 시간 절단(Time truncated)모형은 n번째 까지 고장시점 자료를

$$x_k = \sum_{i=1}^k t_i \quad (k=1,2,\dots,n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n) \quad (10)$$

이라고하면 데이터 집합 D_t는 {n, x₁, x₂ ... x_n; t}와 같이 구성된다. n번째까지 고장시점이 관찰된 고장 절단 모형일 경우에 데이터 집합 D_{x_n}은 {x₁, x₂ ... x_n}으로 구성된다. 이 시간 절단 모형에서의 우도함수는 다음과 같이 알려져 있

다[7,8].

$$L_{NHPP}(\theta | D_t) = \left(\prod_{i=1}^n \lambda(x_i) \right) \exp(-m(t)) \quad (11)$$

단, θ은 미지의 모수 집합을 의미하고 우도함수 (11) 식에서 t을 x_n으로 대체하면 유사한 형태의 고장 절단 모형의 우도함수가 된다. (11)식과 (8), (9)식을 연관하면 유한 NHPP 모형에 대한 우도함수는 다음과 같은 형태로 표현할 수 있다.

$$L_{NHPP_{FN}}(\theta, \theta | D_t) = \left(\prod_{i=1}^n \theta f(x_i) \right) \exp(-\theta F(x_n)) \quad (12)$$

감마분포의 경우의 NHPP 모형에 적용시킨 평균값 함수와 강도함수는 다음과 같다[3].

$$m(t) = \theta F_{Ga}(t | \alpha, \beta) = \theta \left[1 - e^{-\beta t} \left(\sum_{i=0}^{\alpha-1} \frac{(\beta t)^i}{i!} \right) \right]$$

$$\lambda(t) = \theta f_{Ga}(t | \alpha, \beta) = \theta \frac{\beta^\alpha}{\Gamma(\alpha)} t^{\alpha-1} e^{-\beta t}$$

따라서 최우추정법(MLE)을 이용하기 위한 Gamma(α, β) 모형 로그우도함수(Log likelihood function)는 다음과 같이 유도된다.

$$\ln L(\theta, \alpha, \beta | D_{x_n}) = n \ln \theta - n \log \Gamma(\alpha) + n \alpha \ln \beta + (\alpha - 1) \sum_{i=1}^n \ln x_i - \beta \sum_{j=1}^n x_j - \theta + \theta e^{-\beta x_n} \left(\sum_{j=0}^{\alpha-1} \frac{(\beta x_n)^j}{j!} \right) \quad (13)$$

형상 모수 α 값은 상수(알고 있다고)라고 가정했을 때 최우추정법을 이용하기 위하여 (13)식을 θ와 β에 대하여 편미분을 하면 다음과 같은 식을 유도할 수 있다.

$$\frac{n}{\theta} = 1 - \exp(-\beta x_n) \sum_{i=0}^{\alpha-1} \frac{(\beta x_n)^i}{i!} \quad (14)$$

$$\frac{\alpha n}{\beta} = \sum_{i=1}^n x_i - \frac{\theta \left[\exp(-\beta x_n) \sum_{i=0}^{\alpha-1} \frac{(\beta x_n)^i}{i!} \right]}{\theta \beta} \quad (15)$$

(14) 식과 (15) 식을 비선형 연립방정식(수치해석

적 방법을 이용하여 풀면 최우추정치 $\hat{\beta}_{MLE}$ 와 $\hat{\theta}_{MLE}$ 의 값을 구할 수 있다.

(14)과 (15) 식에서 $\alpha=1$ 이면 Goel-Okumoto 모형이 되고 $\alpha=2$ 은 Yamada, Ohba-Osaki 모형이 된다.

유사한 방법으로 (8)식과 (9)식을 연관하여 2모수 카파(kappa(2)) 성장 모형에 대한 로그우도 함수는 다음과 같이 표현된다[4].

$$\ln L(\theta, \alpha, \beta | D_{x_n}) \quad (16)$$

$$= n \ln \theta + n \ln \frac{\alpha}{\beta} - \left(\frac{\alpha+1}{\alpha} \right) \sum_{i=1}^n \ln(\alpha + (x_i/\beta)^\alpha)$$

$$- \theta \left(\frac{(\alpha/\beta)^\alpha}{(\alpha + (x_n/\beta)^\alpha)} \right)^{1/\alpha}$$

본 연구에서는 $\alpha=2$ 인 경우를 고려하면 즉, 형상 모수 $\alpha=2$ 값은 상수(알고 있다고)라고 가정했을 때 최우추정법을 이용하기 위하여 (16)식을 θ 와 β 에 대하여 편미분을 하면 다음과 같은 식을 유도할 수 있다.

$$\frac{n}{\theta} = \left(\frac{(2/\beta)^2}{(2 + (x_n/\beta)^2)} \right)^{1/2} \quad (17)$$

$$\frac{2n}{\beta} = 3/2 \sum_{i=1}^n \left(\frac{4\beta}{2\beta^2 + x_i^2} \right) + \theta \left(\frac{4\beta}{\sqrt{2\beta^2 + x_i^2}} \right) \quad (18)$$

(17) 식과 (18) 식을 비선형 연립방정식(수치해석적 방법)을 이용하여 풀면 최우추정치 $\hat{\beta}_{MLE}$ 와 $\hat{\theta}_{MLE}$ 의 값을 구할 수 있다.

5. 소프트웨어 고장 자료 분석

이 장에서 실제적인 자료를 가지고 카파(2) 신뢰도 모형을 분석하고자 한다. 고장자료는 Allen P.Nikora 와 Michael R.Lyu 가 인용한 SYS2 자료[6]을 이용하고자 한다. 제시하는 신뢰모형들을 분석하기 위하여 우선 자료에 대한 추세 검정이 선행 되어야 한다[1,9]. 추세 분석에는 산술평균 검정(Arithmetic mean test)과

라플라스 추세 검정(Laplace trend test)등이 있다.

<표 1> 고장 간격 자료
<Table 1> Failure Interval Time Data

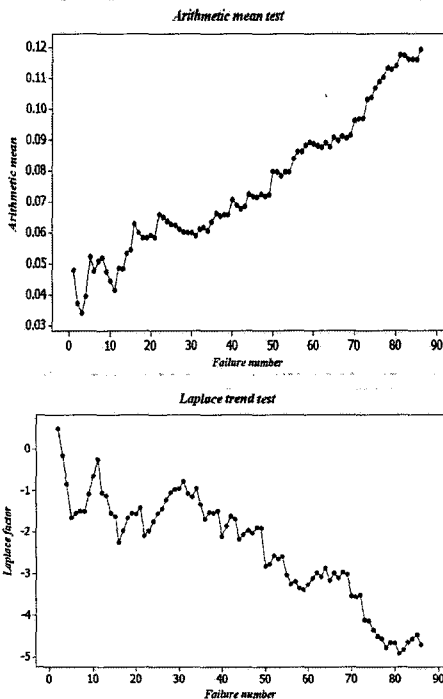
Failure number	Failure interval(second)	Failure time(second)	Variate transformation time(second)	Failure number	Failure interval(second)	Failure time(second)	Variate transformation time(second)
1	479	479	0.00479	44	2441	31886	0.31886
2	266	745	0.00745	45	460	32346	0.32346
3	277	1022	0.01022	46	565	32911	0.32911
4	554	1576	0.01576	47	1119	34030	0.34030
5	1034	2610	0.02610	48	437	34467	0.34467
6	249	2859	0.02859	49	927	35394	0.35394
7	693	3552	0.03552	50	4462	39856	0.39856
8	597	4149	0.04149	51	714	40570	0.40570
9	117	4266	0.04266	52	181	40751	0.40751
10	170	4436	0.04436	53	1485	42236	0.42236
11	117	4553	0.04553	54	757	42993	0.42993
12	1274	5827	0.05827	55	3154	46147	0.46147
13	469	6296	0.06296	56	2115	48262	0.48262
14	1174	7470	0.07470	57	884	49146	0.49146
15	693	8163	0.08163	58	2037	51183	0.51183
16	1908	10071	0.10071	59	1481	52664	0.52664
17	135	10206	0.10206	60	559	53223	0.53223
18	277	10483	0.10483	61	490	53713	0.53713
19	596	11079	0.11079	62	593	54306	0.54306
20	757	11836	0.11836	63	1769	56075	0.56075
21	437	12273	0.12273	64	85	56160	0.56160
22	2230	14503	0.14503	65	2836	58996	0.58996
23	437	14940	0.14940	66	213	59209	0.59209
24	340	15280	0.15280	67	1866	61075	0.61075
25	405	15685	0.15685	68	490	61565	0.61565
26	535	16220	0.16220	69	1487	63052	0.63052
27	277	16497	0.16497	70	4322	67374	0.67374
28	363	16860	0.16860	71	1418	68792	0.68792
29	522	17382	0.17382	72	1023	69815	0.69815
30	613	17995	0.17995	73	5490	75305	0.75305
31	277	18272	0.18272	74	1520	76825	0.76825
32	1300	19572	0.19572	75	3281	80106	0.80106
33	821	20393	0.20393	76	2716	82822	0.82822
34	213	20606	0.20606	77	2175	84997	0.84997
35	1620	22226	0.22226	78	3505	88502	0.88502
36	1601	23827	0.23827	79	725	89227	0.89227
37	298	24125	0.24125	80	1963	91190	0.91190
38	874	24999	0.24999	81	3979	95169	0.95169
39	618	25617	0.25617	82	1090	96259	0.96259
40	2640	28257	0.28257	83	245	96504	0.96504
41	5	28262	0.28262	84	1194	97698	0.97698
42	149	28411	0.28411	85	994	98692	0.98692
43	1034	29445	0.29445	86	3902	102594	1.02594

이 검정을 실시한 결과 [그림 2]에서는 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 있음

을 나타내고 있다. 따라서 이 자료를 가지고 신뢰성장 모형을 제시하는 것이 효율적임을 시사하고 있다[9].

실제 분석에서는 원래의 자료를 변량변환(Variate transformation) 시킨 고장 간격 데이터 (102594×10^{-5} 시간(Second) 단위에서 고장이 86번 일어남)을 이용하였고 <표 1>에 자료가 나열 되어 있다.

소프트웨어 신뢰성 모형의 모수 추정은 최우추정법을 이용하였고 비선형 방정식의 계산방법은 수치해석적 기본 방법인 이분법(Bisection method)을 사용하였다. 이러한 계산은 초기값을 10^{-1} 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 수행하였다 카파 모형을 포함한 모수의 추정값들의 결과는 <표 2>에 요약되었다.



[그림 2] 산술평균 검정과 라플라스 추세 검정
[Fig. 2] Arithmetic mean and Laplace trend test

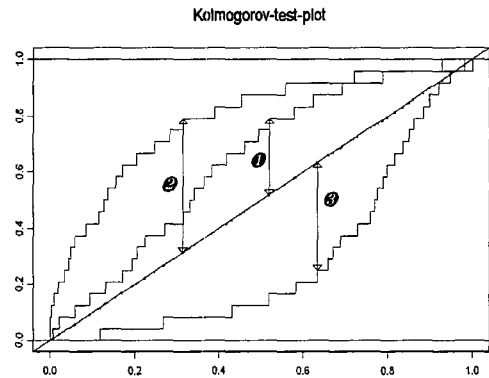
<표 2> 각 모형의 모수 추정값
<Table 2> Estimator of each model

모형	$\hat{\beta}_{MLE}$	$\hat{\theta}_{MLE}$
Goel-Okumoto Model	0.08215	84.87
Yamada-Ohba-Osaki Model	0.82564	85.43
kappa Model($\alpha = 2$)	3.79872	89.78

<표 3> 모형들에 대한 SSE의 값
<Table 3> SSE of each model

Model	Kappa(2) Mode	Goel-Okumoto Model	Yamada-Ohba-Osaki Model
SSE	7926.23	6057.38	5682.12

모형 선택의 하나의 방법으로 편차자승합(SSE [10])을 이용할 수 있는데 이 편차자승합이 작으면 상대적으로 효율적인 모형이 된다. 주어진 자료를 이용하여 제시된 모형들에 대한 편차자승합의 값은 <표 3>에 요약되었다. 이 표에서 카파모형이 기존에 알려진 모형인 Yamada-Ohba-Osaki[11]모형이나 Goel-Okumoto[8] 모형에 비해 상대적으로 효율적 모형으로 나타나고 있다.



단. ① Kappa(2) Model ② Yamada-Ohba-Osaki Model ③ Goel-Okumoto Model

[그림 3] 모형에 대한 Kolmogorov 검정
[Fig. 3] Kolmogorov test of each model

<표 4> 모형에 대한 Kolmogorov 거리
 <Table 4> Kolmogorov distance for each

Model	Kolmogorov distance
Goel-Okumoto Model	0.159593
Yamada-Ohba-Osaki Model	0.155121
Kappa(2) Model	0.132452

6. 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 평가 기술이 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 성장 시킬 수 있다. 이러한 과정을 소프트웨어 성장 과정이라고 볼 수 있다[13].

본 논문에서는 잔존 결함 1개당 고장 발생률이 감소추세를 가진 카파(2)분포를 이용한 카파모형을 제안하였다. 고장 간격시간으로 구성된 자료를 이용하여 기존의 모형과 카파 모형에 대하여 최우 추정법을 이용하여 모수 추정을 실시하였다. 소프트웨어 고장분석 자료는 고장수가 비교적 큰 실측 자료(고장수가 86)인 Allen P.Nikora 와 Michael R.Lyu 가 인용한 SYS2 자료를 통하여 분석하였다. 효율적인 모형 비교를 위한 편차자승합의 결과는 카파모형이 기존에 잘 알려진 Yamada-Ohba-Osaki 모형이나 Goel-Okumoto보다 우수함을 보이고 있고 콜모고로프 거리의 결과도 카파모형이 상대적으로 정상성에 가까운 모형으로 간주할 수 있다.

분석된 자료에 대한 평가에서도 산술평균 검정과 라플라스 추세 검정을 실시한 결과도 신뢰성장이 되고 있음을 나타내고 있다.

참고문헌

- [1] S. S. Gokhale and K. S. Trivedi(1999). A time/structure based software reliability model, *Annals of Software Engineering*, 8, pp. 85-121.
- [2] W. Farr(1996). *Handbook of Software Reliability Engineering*, M.R.Lyu, Editor, chapter *Software Reliability Modeling Survey*, pp. 71-117. McGraw-Hill, New York, NY.
- [3] J. F. Lawless(1981). *Statistical Models and Methods for Lifetime Data*. John Wiley & Sons, New York.
- [4] Hosking, J. R. M(1994). The Four-parameter Kappa distribution". *IBM J. RES, DEVELOP*, Vol. 38, No 3, May.
- [5] Mielke, M. I.(1973), Another Family Distribution for Describing and Analyzing Precipitation Data, *J. Appl. Meteorol* 12, pp. 275-280.
- [6] A. P. Nikora and M. R. Lyu.(1996), *Handbook of Software Reliability Engineering*, M.R. Lyu, Editor, chapter *Software Reliability Measurement Experience*, pp. 255-301. MacGraw-Hill, New York.
- [7] L. Kuo and T. Y. Yang.(1996), Bayesian Computation of Software Reliability. *Journal of the American Statistical Association*, Vol. 91, pp. 763-773.
- [8] A. L. Goel(1985). Software Reliability Models: Assumptions, Limitations and Application, *IEEE Trans. on Software Engineering*, SE-11(12): pp. 1411-1423, Dec.
- [9] K. Kanoun and J. C. Laprie(1996). *Handbook of Software Reliability Engineering*, R.Lyu, Editor, chapter *Trend Analysis*, pp.401-437. MacGraw-Hill, New York.
- [10] H. Pham and L. Nordmann and X. Zhang (1999), A General Imperfect-Software Debugging Model with S-Shaped Fault-Detection Rate. *IEEE Trans. on reliability*, VOL, 48, NO 2, 1999.
- [11] S. Yamada, M. Ohba and S. Osaki(1983),

S-Shaped Reliability Growth Modeling for Software Error Detection". *IEEE Trans. on Reliability*. R-32(5): pp. 475-485, Dec.

[12] S. Selvin(1998), *Modern Applied Biostatistical Methods Using S-Plus*, pp. 141-184.

Oxford University Press, New York.

[13] 김희철, 이상식, 송영제(2005), 소프트웨어 신뢰도 모형에 대한 모수 추정 비교, 한국정보처리학회D, 제11-D권 제6호 pp. 1269-1276.

김희철



1992년 동국대학교 통계학과 석사

1998년 동국대학교 통계학과 박사

2000년3월 ~ 2004년 2월

송호대학 정보산업계열 조교수

2005년 ~ 현재 남서울대학교

산업경영공학과 전임강사

관심분야 : 소프트웨어 신뢰성공학, 웹프로그래밍, 전산통계, 인터넷 비즈니스