2005년도 대한전기학회 하계학술대회 논문집 2005. 7. 18 - 20

제어봉제어계통의 전력제어기 소프트웨어 신뢰성 분석

권순만, 이종무, 박민국, 천종민
한국전기연구원

Software Reliability Analysis of CRCS Power Controller

Soonman Kwon, Jongmoo Lee, Min-Kook Park, and Jong-Min Cheon
Korea Electrotechnology Research Institute

Abstract - 본 논문에서는 원자로 제어봉제어계통에서의 제어 소프트웨어의 신뢰성평가 내용을 기술한다. 안전필수 플랜트의 하나인 원전에 사용되는 제어기기는 높은 수준의 신뢰성을 요구하고 있는데 지금까지 하드웨어에 대한 신뢰성은 객관적인 평가 방법이 제시되어 활용되고 있으나 소프트웨어의 타당한 신뢰성 평가는 여전히 난제로 남아 있다. 본 논문에서는 기존의 평가 모델의 하나를 적용하여 원자로 제어봉제어계통의 전력함 소프트웨어를 평가해본다. 그러나 평가방법에서 예측되는 바와 같이 평가결과의 불확실 정도의 크기로 비추어 볼 때 실제 안전필수 산업분야에서의 실적용을 판단하기 위한 객관적인 신뢰성 척도로 사용되기는 여전히 어렵다고 판단된다.

1. 서 론

원전이나 철도, 항공 등과 같은 안전필수 플랜트에서의 제어계통의 고신뢰성은 아주 중요하다. 이것은 제어계통의 고장이 인명이나 재산의 큰 손실을 가져올 수 있기 때문에 고신뢰성에 대한 객관적인 평가 검증을 필수적으로 요구하고 있다. 디지털 기술의 급속한 발달에 따라 주어진 프로그램에 따라 동작하는 디지털 프로세서가 제어계통에 많이 사용되고 있다. 이에 따라 그 제어계통의 다양한 기능은 하드웨어와 함께 그 제어기의 근간을 이루는 디지털프로세서의 소프트웨어의 신뢰성에 절대적으로 의존한다고 할 수 있다. 그러나 현재까지 하드웨어에 대한 신뢰성 평가는 군사용의 MIL 217F[1]나 상용의 Bell 6 [2] 등의 평가 방법이 제시되고 있고 이에 기반을 둔 상용 소프트웨어 평가 도구들도 개발되어 활용되고 있으나 소프트웨어에 대해서는 여전히 표준화 등과는 거리가 먼 수준에 머물고 있다. 실제 산업에 활용되는 안전필수 소프트웨어들은 개발과정에서 철저한 확이 및 검증절차를 준수하도록 요구하고 있고 개발후의 평가 기준에 대해서는 대책이 없다. 물론 다양한 이론적인 평가기법은 오래 전부터 꾸준히 제시되어 왔으나 안전필수 플랜트에서의 소프트웨어 개발기준으로 규제에 적용될 수준에는 아직 미치지 못하고 있으며 NASA 등에서는 소프트웨어의 자체 평가 및 외주 개발을 위한 개발업체의 평가 등을 위해 신뢰성 평가 기준을 자체 개발하여 사용하고 있다. 그러나 앞으로의 모든 안전필수 제어시스템은 점점 더 소프트웨어 의존도가 커지리라 예상되므로 어떤 형태로든지 현재 상용의 일반 소프트웨어에서 사용하고 있는 신뢰도 평가에 상응하는 방법론이 어떤 형태로든지 간에 안전필수 소프트웨어에도 도입되리라 생각된다. 따라서 본 논문에서는 이러한 예비적인 시도의 하나로 국내에서 개발한 안전필수 관련 제어계통[1],[2]의 소프트웨어의 신뢰성 평가에 대해 논하고 이 소프트웨어의 전체 시스템의 신뢰도에 대한 영향을 검토해 본다.

2. 본 론

2.1 제어시스템 개요

원전 계측제어설비인 제어봉구동장치제어시스템은 중성자 흡수재로 만들어진 제어봉의 상하 운동을 제어하여, 궁극적으로 원자로 출력을 조절하는 매우 중요한 설비이다. 제어시스템은 제어함(MCU)과 전력함(PCAMUs)의 두 부분으로 구성되어 있는데 제어함은 상위의 출력 조절 시스템으로부터 제어봉 속도 및 방향 명령을 수수하여 구체적인 제어봉 동작이 구현되도록 명령을 만들어 주고, 전력함에서는 제어함으로부터 오는 명령에 따라 제어봉구동장치에 공급되는 전력을 제어하게 된다. 새로이 개발된 제어봉구동장치제어시스템의 제어함은 PLC 기반으로, 전력함의 전력제어기는 DSP(Digital Signal Processor)를 기반으로 설계되어 다음 그림 1과 같은 구조로 되어 있다.

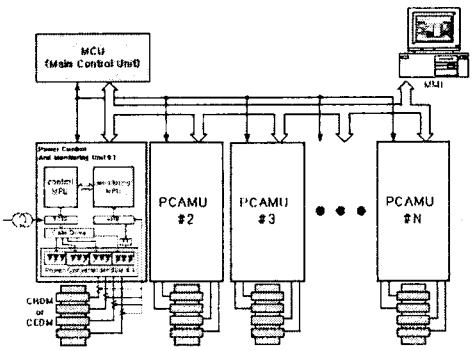

그림 1. 시스템 구조

전력제어기에서 수행하는 프로그램은 진단, 제어 및 통신 프로그램으로 구분할 수 있으며 C 코우드로 작성되어 있고 전체 사이즈는 약 13 KLOC 이다.

2.2 신뢰성 평가

신뢰성 평가를 위한 방법은 다양하나[3] 일반적으로 신뢰성 성장 모델에 기반을 둔 여러 가지 모델을 이용한 시뮬레이션 방법이 많이 이용되고 있다. 본 논문에서는 시험 중에 관측된 Failure Count에 기반한 일반적인 모델을 사용하여 평가하게 된다.

2.2.1 평가 모델 검토

신뢰성 분석을 위하여 사용하는 소프트웨어 신뢰성 모델은 Failure Count 모델의 하나로 널리 사용되는 Goel-Okumoto NHPP 모델[3]을 이용한다. 그 이유는 본 논문에서의 평가 대상 소프트웨어의 특성이 이

Goel-Okumoto 모델에서의 다음과 같은 조건과 일치하기 때문이다. 즉, failure와 fault event를 각각 Γ, Λ라고 두면

1) 테스트 전의 코우드 내의 Λ는 N개로서 정해져 있으나 미지이다.

2) Γ들은 상관되어 있지 않고 Γ가 일어나는 시간 간격들은 독립적이고 지수분포인 random 변수이다.

3) Γ가 일어났을 때의 Λ제거는 즉시 이루어지며 그것으로 인한 새로운 Λ는 야기하지 않는다.

4) 각 Λ의 hazard rate $z(t)$는 시불변인 상수 φ이다. 또 각 Λ가 Γ를 일으키는 확률은 서로 같다.

5) 시간 t 동안에 발생하는 Γ의 기대값은 bounded non-decreasing 평균값 함수 $\mu(t)$를 가지는 Poisson 분포를 가진다.

6) $E\{(\Gamma(0,\infty)\} = N$,

7) $N(\Gamma(t, t+\delta t]) \propto E\{N-\mu(t)\}$.

2.2.2 평가 모델

평균값 함수 $\mu(t)$는

$$\mu(t) = N(1-e^{-bt}), \quad N>0, \quad b>0 \tag{1}$$

로 표시된다. 여기서 b 는 failure의 발생률을 나타내는 비례상수이다.

또 failure 세기 함수 $\lambda(t)$ 는

$$\lambda(t) = \frac{\mu(t)}{dt} = Nbe^{-}bt \tag{2}$$

이다.

따라서

$$\mu(t) = \int \lambda(t)\,dt = E\{N(t)\} \tag{3}$$

로 되고

$$E\{N(\infty)\} = N \tag{4}$$

이다.

i 번째 시간 간격에서의 fault 수 f_i 는 평균값이 $\mu(t_i)-\mu(t_{i-1})$ 인 Poisson 확률변수이므로 $f_i,\ i=1,2,\cdots,n$ 의 joint 확률밀도함수는

$$\prod_{i=1}^{n} \frac{[\mu(t_i)-\mu(t_{i-1})]}{f_i!}\, e^{\mu(t_i)-\mu(t_{i-1})} \tag{5}$$

로 표시될 수 있고,

결국 N, b 의 maximum likelihood 추정값 $\hat{N}$, $\hat{b}$ 는

$$\hat{N} = \frac{\sum_{i=1}^{n} f_i}{(1-e^{-\hat{b}t_n})} \tag{6}$$

$$\frac{t_n e^{-\hat{b}t_n} \sum_{i=1}^{n} f_i}{(1-e^{-\hat{b}t_n})} = \sum_{i=1}^{n} \frac{f_i(t_i e^{-\hat{b}t_i} - t_{i-1}e^{-\hat{b}t_{i-1}})}{e^{-\hat{b}t_{i-1}} - e^{-\hat{b}t_i}} \tag{7}$$

으로부터 계산된다. 즉, (7)로부터 $\hat{b}$ 가 먼저 구해진 다음 이 값을 (6)에 대입하여 계산하면 $\hat{N}$ 이 구해진다.

2.2 신뢰성 평가 결과 및 고찰

신뢰성 평가를 위하여 사용하는 실측 데이터는 제어봉 구동장치 제어시스템의 초기 시제품 개발시에 사용된 초기 소프트웨어 Version에 대한 자료이며 IEEE Std 1012에 제시한 소프트웨어 수명주기에 따른 개발과정 중 2 개월에 걸친 단위시험 과정을 통해 발견된 Fault 수 데이터이다. 단, 데이터에서 시간의 단위는 별도로 나타내지 않는다.

표 1. 시험 데이터

시간 slot	Fault 수
1	7
2	3
3	4
4	2
5	2
6	0
7	1

식 (6) 및 (7)을 이용하여 계산하면 $\hat{N}$, $\hat{b}$ 값은 각각 24.745, 0.103으로 된다.

평균값 함수와 failure 세기 함수는 각각 다음 그림 2와 3의 형태로 나타내어지는데 그림에서 알 수 있는 바와 같이 현재의 fault 빈도는 약 1.239로 계산된다.

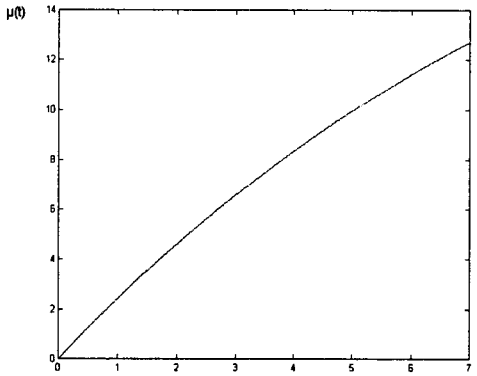

그림 2. $\mu(t)$

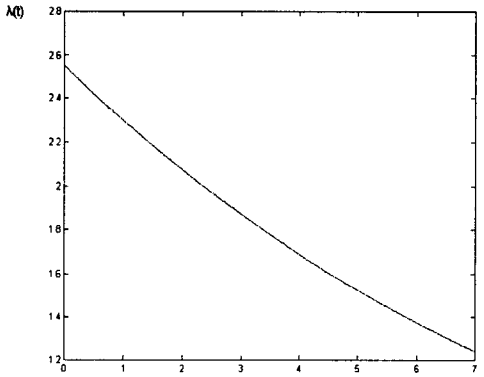

그림 3. $\lambda(t)$

앞에서 얻어진 결과를 바탕으로 안전필수 제어시스템용 소프트웨어의 신뢰성을 나타내는 척도로서의 의미를 가지기 위해서는 더 많은 검토가 필요하리라 생각된다. 왜냐하면 안전필수 소프트웨어의 생명은 무결점을 추구하며 이를 위한 노력으로 개발 및 시험과정에서의 철저한 확인 및 검증 과정[4]을 요구하고 있으므로 앞에서 논한 확률론적 결과는 확정적인(deterministic) 지표와 연결시키기가 어렵기 때문이다.

본 논문에서 결과로서 산출된 값의 의미를 정량적인 데이터로 활용하기 위해서는 다양한 신뢰성 성장 모델을

바탕으로 한 시뮬레이션을 통하여 비교 분석이 필요하다고 생각된다. 또한 추가적으로 수행되는 소프트웨어 시험을 통하여 그 모델의 타당성도 검증되어야 할 것으로 판단된다.

3. 결 론

원전에 사용되는 제어시스템의 소프트웨어의 신뢰성을 평가해 보았다. 실제 개발과정에서 추출된 데이터를 이용하여 평가를 수행하였는데 그 결과를 실제 제어시스템의 신뢰성 평가의 한 요소로서 활용하기 위해서는 실측에 위한 모델의 검증 및 특성 평가가 충분히 이루어져야 할 것으로 생각된다.

[참 고 문 헌]

[1] S. Kwon *et al.*, "A Fault-Tolerant System Design of Rod Control System for Nuclear Power Plants", ICEMS 2004, PC-39, 2004.

[2] 정대원 외, "원전 제어봉구동장치 제어계통 소프트웨어 개발 및 검증", 제1회 안전-필수 소프트웨어 개발 및 검증 워크샵, pp. 203~220, 2003.

[3] M. R. Lyu, *Handbook of Softwaxre Reliability Engineering*, IEEE Computer Society Press, 1996.

[4] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 5th Edition, McGraw-Hill, 2001.