동적 결함 트리 (Dynamic Fault Tree) 알고리즘을 이용한 시스템의 신뢰도 평가에 관한 연구

A Study on System's Reliability Evaluation Using DFT Algorithm

*김 진 수1) **양 성 현2) ***이 기 서3)

Kim,jin-su Yang,sung-hyun Lee,kee-seo

ABSTRACT

In this paper, Dynamic Fault Tree algorithm(DFT alrotithm)is presented. This new algorithm provides a concise representation of dynamic fault tolerance system structure with redundancy, dynamic redundancy management and complex fault & error recovery techniques.

And it allows the modeler to define a dynamic fault tree model with the relative advantages of both fault tree and Markov models that captures the system structure and dynamic behavior.

This algorithm applies to TMR and Dual-Duplex systems with the dynamic behavior and show that this algorithm captured the dynamic behavior in these systems with fault & error recovery technique, sequence-dependent failures and the use dynamic spare.

The DFT algorithm for solving the problems of the systems is more effective than the Markov and Fault tree analysis model.

제 1 장 서 론

결함 허용 시스템에서의 신뢰도 평가모델은 시스템 구조뿐만 아니라 발생 가능한 결함으로부터 유도될 수 있는 결함과 고장을 다루는 기술까지 표현되어야 한다.[1] 이를 위해 기존의 결함 트리 방법은 시스템의 고장 원인인 결함을 근거로 시스템의 구조를 간결하게 표현하여 시스템 평가를 할 수 있는 방법으로 1960년대 이후에 소개되어 신뢰도 평가 기술로 널리 이용되고있다.[5]

* 김진수 광운대학교 대학원 제어계측공학과

** 양성현 광운대학교 부교수 정회원

***이기서 광운대학교 제어계측공학과 교수 정회원

그러나, 결함 트리 모델을 이용할 때 결함과 고장의 복구기술, 순차적인 고장발생을 갖는 시스템 또는 동적 여분을 사용하는 시스템에 대한 평가는 불가능했다. 따라서 본 논문에서는 기존의 마코브 모델과 결함 트리 기법의 장점을 갖고 모델링 상의 편의와 개념적인 분석의 용이함과 계산상 효율이 좋은 알고리즘인 동적 결함 트리 기법을 제안하였다.[9]또한 이 알고리즘을 이용하여 TMR과 이중화 중복 시스템(Dual-Duplex) 성능평가를 제시함으로써, 기존의 방법보다 시스템 평가의 효율성과 유연성에 있어서의 우수함을 보였다.

제 2 장 신뢰도 정의 및 결함 모델

2.1. 신뢰도의 정의

신뢰도(Reliability) : 시간 t0 에서 t까지의 시스템이나 부품이 정상 동작 할 확률을 말한다.

$$R(t)=\frac{N_0}{N}=1-\frac{N_f}{N}=1-Q(t);\quad R(t)=e^{-\int Z(t)dt};\quad Z(t)=\lambda$$

N은 전체생산부품수 : N_0은 시간 t에서 정상동작하는 부품의 개수

R(t)는 신뢰도 ; Q(t)는 비 신뢰도(Unreliability)이다.

z(t)는 고장률(Failure rate) 또는 위험률(HazardRate), 위험함수(Hazard Function)를 나타낸다.

2.2 결함 모델

입력에 대해 회로가 의도하지 않았던 출력 발생 시 나타나는 현상을 논리적으로 모델링 함으로써 여러 종류의 결함에 대한 검출과 표현을 하게 된다. 결함모델의 표현에서는 주로 논리모델을 사용한다. 이런 결함모델에는 고착 결함 모델(stuck at model), stuck on model, 단선모델(open line model), 브리지모델(bridging model), 지연모델(delay model)로 나눌 수 있다.[8]

제 3 장 결함 트리 분석기법(Fault Tree Analysis)

3.1 결함 트리(Fault Tree)

IEC 1025 국제 표준을 따르는 일반적인 결함 트리 기법은 복잡한 시스템을 분석하기 위해 사용되어진다. 결함 트리 기법은 시스템 고장(top event)이 일어나는 원인을 찾아가는 방법으로 정해진 기호에 의해서 시스템을 표현한다.

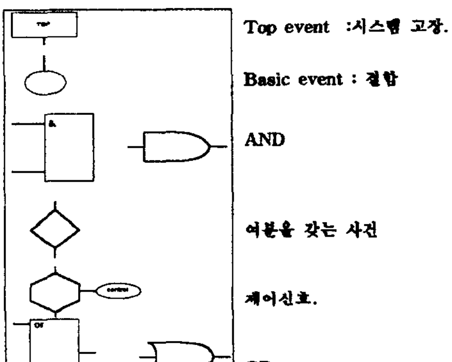

그림1 IEC1025 고장트리심볼

3.2 시스템 분석

3.2.1 구조함수(Structure Function)

시스템 구조를 표현한 상태벡터들의 함수로서 시스템을 정의한다. 그리고 시스템은 n개의 요소들의 집합으로 구성된다고 가정하고 각각 상태벡터의 결함은 시스템의 고장을 유발시킨다.

$$\Phi(X) = \begin{cases} 0 & \text{상태 벡터가 } X\text{일때 시스템이 정상 동작을 하지 않는 경우} \\ 1 & \text{상태 벡터가 } X\text{일때 시스템이 정상적인 함수를 수행하는 경우} \end{cases}$$

I) 직렬 시스템

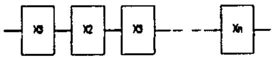

그림 2 직렬시스템의 구조함수

직렬 시스템은 모든 요소가 정상적인 함수를 수행할 때 시스템이 정상동작을 하게 된다.

$$\Phi(X) = \begin{cases} 0 & : x_i = 0 \text{ 인 상태 벡터 } i\text{가 존재하는 경우} \\ 1 & : \text{모든 } x_i = 1\text{즉 모든 상태 벡터가 정상동작을 하는 경우} \end{cases}$$

$$= \min \{x_1, x_2, x_3, x_4, \cdots, x_n\}$$

$$= \prod_{i=1}^{n} x_i$$

II) 병렬 시스템

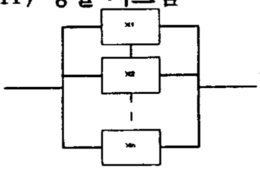

그림 3 병렬시스템의 구조함수

병렬 시스템의 경우 모든 상태 벡터가 결함상태일 때 시스템은 고장이 된다.

$$\phi(X) = \begin{cases} 1 & \text{if } x_i = 0 \text{ for } all\ i = 1, 2, 3, 4, \cdots n \\ 0 & \text{if } there\ \exists\ an\ i\ such\ that\ x_i = 1 \end{cases}$$

$$= \max \{x_1, x_2, x_3, \cdots, x_n\}$$

$$= 1 - \prod_{i=1}^{n} (1 - x_i)$$

III) k out of n

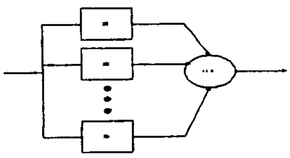

그림 4 k out of n의 구조함수

k out of n은 n개의 입력 중 k개 이상이 정상적인 함수를 수행한다면 시스템은 정상 동작하게 된다. 이것은 동적 시스템(dynamic system)의 경우에 사용되어지는 시스템 분석 기법 중의 하나이다.

$$\phi(X) = \begin{cases} 0 & \text{if } \sum_{i=1}^{n} x_i < k \\ 1 & \text{if } \sum_{i=1}^{n} x_i \geq k \end{cases}$$

IV) Bridge Structure

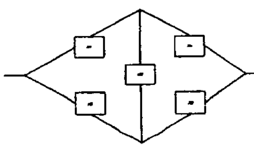

그림 5 bridge 구조함수

신호가 흐르는 경로는 {1,3,5} {1,4} {2,5} {2,3,4} 이것은 minimal path set 이라 부른다. 시스템이 복잡해져감에 따라 path를 찾아내기 어려워지기 때문에 minimal path와 minimal cutset의 개념을 도입한다.

$$\phi(X) = 1 - (1 - x_1 x_3 x_5)(1 - x_1 x_4)(1 - x_2 x_5)(1 - x_2 x_3 x_4)$$

3.2.2 Minimal path와 Minimal cutset

I) Let $P_1, P_2, P_3, \cdots P_S$ s는 minimal path 집합

$$\alpha_j(X) = \begin{cases} 1 & \text{if } all\ components\ of\ P_j\ are\ functioning \\ 0 & otherwise \end{cases}$$

j=1, 2, 3, ... , s

II) Let C1, C2, C3, ... , Ck 는 K개의 Minimal cut 집합이다.

$$\beta_j(X) = \begin{cases} 1 & \text{만약 } jth\ \text{min}imal\ cut\ set\text{이 정상동작하는 경우} \\ 0 & otherwise \end{cases}$$

j=1, 2, 3, . . . , k

∴ minimal path를 이용한 구조함수의 표현

$$\phi(X) = 1 - \prod_{j=1}^{s}(1 - \prod_{i \in p_j} x_i)$$

∴ minimal cutset을 이용한 구조함수의 표현

$$\phi(X) = \prod_{j=1}^{k}[1 - \prod_{i \in C_j}(1 - x_i)]$$

minimal path 나 cutset을 이용함으로써 보다 쉽게 구조함수를 구할 수 있다. 이렇게 하여 간소화된 구조함수는 신뢰도함수에 적용되어 우리가 원하는 신뢰도를 계산함으로서 계산량을 줄일 수 있다.

3.3 신뢰도 함수(Reliability Functions)

우리는 확률모델을 통하여 신뢰도 함수를 정의하고자한다.

① $r(p) = E[\phi(X)]$

② path vector를 이용한 정의

r(p)=P[X는 pathvector]

③ cut set vector를 이용한 정의

r(p)=1 - P[X는 cutset vector]

④ decomposition을 이용한 정의

$r(p) = r(1_i, p)p_i + r(0_i, p)(1 - p_i)$

3.4 결함트리(Fault Tree) 구조분석 예

3.4.1 Minimal Cut Set 알고리즘의 적용

그림6에 Minimal cut set 알고리즘을 적용하면,

G0는 OR Gate이므로 1 , 2, G1

G1도 OR Gate이므로 1 , 2, G2, G3

G2는 AND Gate이므로 1, 2, G4G5, G3

G3는 OR Gate이므로 1, 2, G4G5, 3, G6 . . . .

Total cut set은 {1}, {2}, {4, 6, 7}, {5, 6, 7}, {3}, {5, 6}이 된다.

여기서 {5, 6, 7} 같이 {5, 6}을 포함하는 것처럼 보이는 것을 superset이라 불리며 이것은 제거된다.

그러므로 minimal cut set은 {1}{2}{4, 6, 7}{3}{5, 6}이 된다.

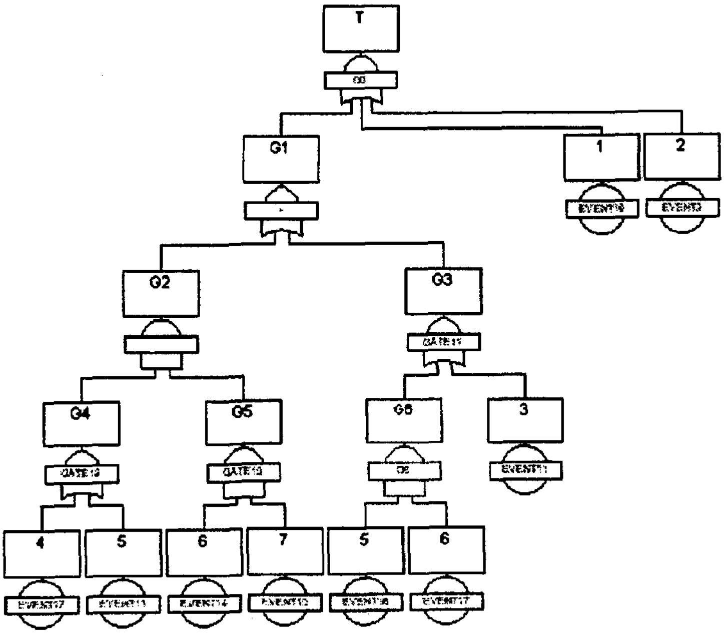

그림 6 결함트리 모델링의 cutset에 의한 분석

step	1	2	3	4	5	6
	1	1	1	1	1	1
	2	2	2	2	2	2
	G1	G2	G4,G5	G4,G5	4,G5	4,6,7
		G3	G3	3	5,G5	5,6,7
				G6	3	3
					5,6	5,6

표 1 주어진 그림에서 각단계의 cutset

$G4=X4+X5$; $G5=X6X7$; $G6=X5X6$

-> $G2=G4G5=(X4+X5)(X6X7)=X4X6X7+X5X6X7$

-> $G3=X3+G6=X3+X5X6$

-> $G1=G2+G3=X4X6X7+X5X6X7+X3+X5X6$ [Since $X5X6+X5X6X7=X5X6$]

-> $G1=X3+X5X6+X4X6X7$

-> $G0=X1+X2+X3+X5X6+X4X6X7$ 가 된다.

* $r(p) = 1-(1-p1)-(1-p2)-(1-p3)-(1-p5p6)-(1-p4p6p7) \le 1-Q$

제 4 장 동적 결함트리 방법

4.1 동적 결함트리 알고리즘

일반적인 결함트리 방법에서는 시스템이 순차적인 종속요소 갖고 있을 때 분석할 수 없다. 순차 종속 고장의 예로써 스위치 제어기에 연결된 하나의 대기 여분과 동작 시스템 요소로 구성 된 시스템을 고려 할 수 있다. 만약 스위치 제어기가 동작중인 시스템의 고장 후에 고장난다면 그 때 시스템은 계속 동작할 것이다. 그러나 만약 스위치 제어기가 동작 시스템 고장 전에 고장이 난다면 대기 여분은 동작상태로 스위칭 되지 않기 때문에 시스템은 동작하고 있는 모듈 또는 시스템 고장과 동시에 전체 시스템 고장으로 발전한다. 따라서 이러한 경우 시스템

고장은 부 시스템 또는 부품의 결합사이의 조합에 따라 일어나는 것이 아니라 그들의 순서에 의존하여 일어난다.

본 절에서는 이러한 결함 트리 방법을 개선한 동적 결함 트리 알고리즘을 제시하기 위하여 기본적인 다음 사항을 가정한다.

1) 고장률은 동일한 상수로 고려한다.
2) 동작 중 보수는 고려하지 않는다.(즉 고장상태인 사건을 동작 중 정상상태로 복구하는 것을 고려하지 않는다.

따라서 결함과 고장의 복구기술이나 순차적인 고장발생을 갖는 시스템을 평가하기 위해 결함트리에 동적 게이트(FDEP) 알고리즘을 추가한다.

4.2 FDEP(Functional dependency gate)

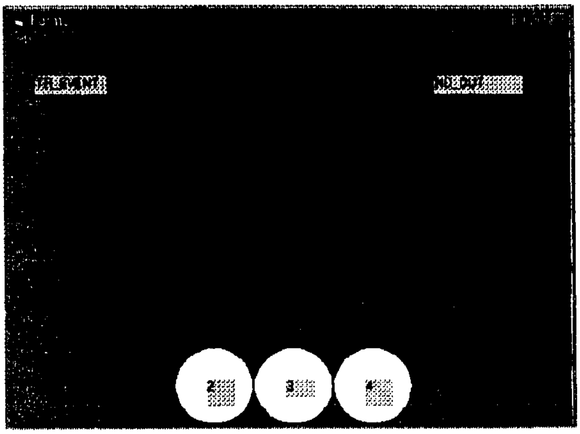

그림 7 FDEP 블록 다이아그램

1) FDEP의 트리거 사건은 기본 사건이거나 다른 게이트의 출력이다.
2) FDEP의 출력은 독립적이다.
3) 각 사건 2,3,4는 종속사건들이며 nd_out은 독립적인 출력이다.
   종속 기본 사건은 함수적으로 트리거 사건에 의존한다.
   트리거 사건이 일어났을 때 종속사건이 따라 일어난다.

```
/*visual basic 5.0을 이용한 FDEP 재구성 프로그램*/
Private Sub Command1_Click()
message17 = " time dependent 시에 t값을 입력하시요"
input17 = InputBox(message17, Title, Default)
End Sub
Private Sub L2_Click(Index As Integer)
input4 = 0
message1 = "다음은 l2의 고장률을 입력해 주십시요"
input4 = InputBox(message1, Title, Default)
End Sub
Private Sub L3_Click(Index As Integer)
input5 = 0
message3 = " 다음은 l3의 고장률을 입력해주십시요"
input5 = InputBox(message3, Title, Default)
End Sub
Private Sub L4_Click(Index As Integer)
input6 = 0
message4 = "다음은 l4의 고장률을 입력해 주십시요."
input6 = InputBox(message4, Title, Default)
End Sub
Private Sub ND_OUT_Click()
Form9.Show
Form9.Print out3
Form9.Print out0
Form9.Print out1
Form9.Print out2
Form9.Print out4
End Sub
Private Sub TR_EVENT_Click()
input3 = 2  ; default
Message2 = "트리거 값을 입력하시요. 입력값의 범위는 2-4"
input3 = InputBox(Message2, Title, Default)
If input3 = 2 And Not input4 = 0 Then out0 = input4
If input3 = 2 And Not input4 = 0 Then out1 = 1 - input4
If input3 = 2 And input4 = 0 Then input3 = 3  ; 다음입력으로 교체
If input3 = 2 And input4 > 0 Then out3 = 1   ; 정상동작
If input3 = 2 And input4 = 0 Then out3 = 0   ; 고장
If input3 = 2 And Not input4 = 0 Then out2 = 1 - (input4) * (input5) * (input6)
```

```
If input3 = 3 Then out4 = Exp(-input4 * input17) ; 신뢰도 계산  exp(-lamda*t)
If input3 = 3 And Not input5 = 0 Then out0 = input5
If input3 = 3 And Not input5 = 0 Then out1 = 1 - input5
If input3 = 3 And input5 = 0 Then input3 = 4
If input3 = 3 And input5 > 0 Then out3 = 1
If input3 = 3 And input5 = 0 Then out3 = 0
If input3 = 3 Then out2 = 1 - (input5) * (input6)
If input3 = 3 Then out4 = Exp(-input5 * input17)
If input3 = 4 And Not input6 = 0 Then out0 = input6
If input3 = 4 And Not input6 = 0 Then out1 = 1 - input6
If input3 = 4 And Not input6 = 0 Then out2 = 1 - input6
If input3 = 4 And input6 > 0 Then out3 = 1
If input3 = 4 And input6 = 0 Then out3 = 0
If input3 = 3 Then out4 = Exp(-input6 * input17)
End Sub
```

제 5 장 실험 및 고찰

5.1 TMR 시스템

시간 종속인 TMR의 구조함수에 의한 신뢰도함수는 식(1)과 같다.

$$R(t)=1-(1-e^{-\lambda_1 * t}e^{-\lambda_2 * t})(1-e^{-\lambda_1 * t}e^{-\lambda_3 * t})(1-e^{-\lambda_2 * t}e^{-\lambda_3 * t}) \quad (1)$$

5.2 이중화 중복 시스템

시간 종속인 이중화 중복 시스템(dual-duplex system)의 구조함수에 의한 신뢰도 함수는 식(2)과 같다.

$$\begin{aligned} M_1 &= e^{-\lambda_1 t}e^{-\lambda_2 t} \\ M_2 &= e^{-\lambda_3 t}e^{-\lambda_4 t} \\ R(t) &= 1-[(1-M_1)(1-M_2)] \end{aligned} \quad (2)$$

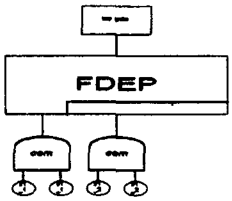

그림 8 Dual Duplex의 결합트리모델링

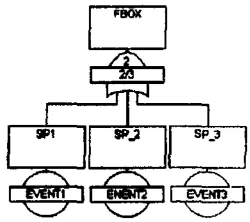

그림 9 TMR의 결함트리 모델링

고장률이 0.00001인 경우와 0.0001인 경우에 대해 제작된 분석기(그림 10)를 사용하여 시뮬레이션한 결과를 그림 (10,11)에서 보여준다.

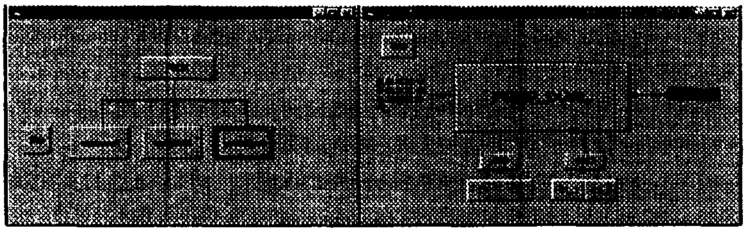

그림 10 Visual Basic을 이용한 TMR 과 Dual Duplex 신뢰도분석기

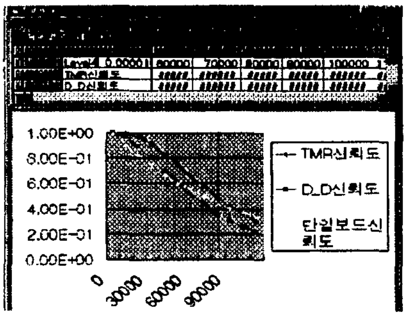

그림 11 고장률이 0.00001일때의 세보드의 신뢰도 비교

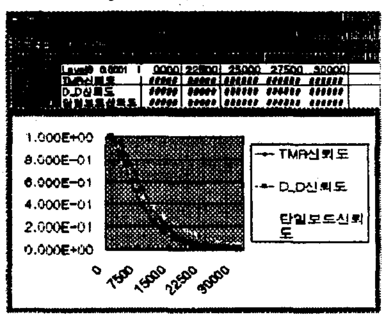

그림 12 고장률이 0.0001일 때 세 보드의 신뢰도의 비교

따라서 우리는 DFT 알고리즘을 사용하여 수동여분을 가지는 TMR과 순차적 종속 여분 시스템인 이중화 중복시스템의 평가가 가능함을 알 수 있다.

제 6 장 결 론

본 논문에서는 시스템 평가를 위해 이용되고 있는 결함 트리 방법과 마코브 방법의 단점을 지적하고 그것을 극복하기 위한 새로운 알고리즘인 DFT를 제시하였다. 기존의 결함 트리 방법이 순차적인 종속여분을 가지는 시스템이나 동적 여분을 가지는 시스템, 결함의 검출 및 오류 복구 기능을 갖게 설계된 시스템에 대한 신뢰도 평가는 할 수 없었고 마코브 방법을 이용할 때에는 시스템의 상태 증가에 따른 계산량과 모델링상의 복잡도가 증가한다. 따라서 본 논문에서는 결함 트리 기술을 근거로 이러한 단점을 해결할 수 있는 알고리즘을 설계하였다. 제시한 DFT 알고리즘은 결함트리 방법으로서 평가가 불가능했던 모델 중 동적 여분을 갖는 TMR 과 시스템 복구 기능을 위해 스위치 기능을 갖는 이중화 중복 시스템에 대한 신뢰도 평가를 수행함으로써 순차적 종속 여분 시스템의 신뢰도 평가가 가능함을 보였다. 그러나 위의 알고리즘을 범용화 하기 위해서는 부품 레벨에서의 신뢰성 평가방법의 모듈화와 시스템 구조에 따른 유연성을 갖게 하기 위한 알고리즘 개발에 대한 연구가 필요할 것으로 생각된다.

참 고 문 헌

[1] Joanne Bechta Dugan "Fault Trees and Imperfect Coverage" IEEE 1989

[2] Kelvin S.Brown "Evaluating Fault Trees(AND & OR Gate Only) with Repeated Events" IEEE Trans on Reliability vol.39 NO2 1990 JUNE

[3] "Empirical Failure Analysis and Validation of Fault Models in CMOS VLSI Circuit"1992 IEEE DESIGN & TEST OF COMPUTER

[4] F.A.Patterson-Hine and B.V.Koen "Direct Evaluation Fault Tree Using Objected-Oriented Programming Techniques" IEEE Trans on Reliability vol.38 NO2 1989 JUNE

[5] Oliver Coudert and Jean Christophe Madre "An Interactive Fault-Tree Analyzer IEEE Trans on Reliability vol.43 NO1 1994 MARCH

[6] Pankaj Jalote "Dynamic Reconfiguration of CSP Programs for Fault Tolerance" IEEE 1992

[7] Antonio Bogarin Geymayr and Nelson Francisco Favilla Ebecken "Fault-Tree Analysis : A Knowledge-Engineering Approach" IEEE Trans on Reliability vol.44 NO1 1995 MARCH

[8] Jacob A Abraham "Fault and Error Models for VLSI" IEEE 1996

[9] Joanne Bechta Dugan, Salvatore J.Bavuso "Dynamic Fault-Tree Models for Fault-Tolerant Computer Systems" IEEE Trans on Reliability, vol 41,NO 3,1992 SEPTEMBER

[10] R.Ramakumar "Engineering Reliability : Fundamentals and Application" PRENTICE HALL 1993

[11] Lawrence M.Leemis "RELIABILITY Probabilistic Models and Statistical Methods" Prentice-Hall International 1995

[12] Dhiraj K. Paradhan "Fault-Tolerant Computer System Design" Prentice-hall PTR 1996

[13] Dimitri Kececioglu "Reliability Engineering Handbook" volume 2 Prentice hall 1991

[14] 양성현,이기서 "제어 가능한 자체검사 특성검사기 설계" 한국통신학회지 VOL.2

[15] 양성현,이기서 "Fault-Tolerance를 위한 시스템 동작 방식에 대한 비교 연구" 한국통신학회 논문지 제17권 제11호 pp 1297-1289 1992

[16] 양성현,이기서 "TMR 시스템의 설계 및 신뢰도 측정 알고리즘" 대한 전기 학회 논문지 제43권 제3호 pp 515-527 1992