

Fault Tree Analysis을 활용한 집진기(Bag Filter) 고장의 체계적 분석

-A System Approach to A Bag Filter Failure using Fault Tree Analysis

이 근 회*

이 동 형**

Abstract

This paper takes aim at the reliability evaluation by application of Fault Tree Analysis and its computerization.

FTA is one of the methods for evaluation of system reliability and safety analysis.

The important characteristic of this paper is that computer program is written in the package program(dBase III+) by 16Bit/AT personal computer.

The program consists of three program segments.

(1) The minimal cut sets of the system fault tree are obtained by means of "Fault tree reduction algorithm".

(2) The minimal path sets are obtained by inversion of the minimal cut sets determined from fault tree.

(3) The importance of the basic events which are presented in the minimal cut sets is obtained by means of structural importance analysis.

In this paper, a Fault Tree Analysis is applied to a BAG FILTER which is a kind of dust collector.

1. 서 론

오늘날 산업 기술이 급속하게 고도화되어감에 따라 그에 관련된 각종 시스템은 더욱 복잡해지고 대형화되는 추세에 있다. 또한 사용자는 경제성장으로 인한 부의 증대와 시민의식의 향상으로 보다 다양하고 개성있는 제품을 요구하기에 이르렀다.

이러한 변화추세에 기업이 보다 능동적으로 대응하기 위해서는 고도의 기능성을 갖는 신뢰성 높은 제품이나 설비를 생산하지 않으면 안되게 되었다. 이를 위해서는 시스템의 신뢰성 분석이나 안전성 분석을 실시할 필요가 있는데 그 유용한 방법이 Fault Tree Analysis이다.

FTA 기법은 현장에서 얻어진 발생가능한 모든 고장(또는 안전사고) mode를 조사, 분류하고 computer program을 이용하여 신뢰성 분석을 실시함으로써 다음과 같은 효과를 기대할 수 있다.

- (1) 시스템에 대한 이해증진
- (2) 시스템의 고장원인을 계통적으로 파악 가능
- (3) 시스템의 잠재적 문제점이 제기됨으로써 신뢰성 향상 도모
- (4) FMEA(Failure Mode Effect Analysis)의 보조수단으로 활용 가능

본 논문에서는 집진기(Bag Filter)에 대한 사례연구를 통해 Bag Filter의 고장요인을 조사, 분류하고 그 신뢰성분석을 행함으로써 고장대책을 강구코자 한다.

*한양대 산업공학과 교수

**대전공업대학 산업공학과








接受日字: 1989년 10월 30일

2. Fault Tree Analysis의 기본개념과 분석절차

2.1 Fault Tree Analysis의 기본 개념

FTA란 시스템의 고장(Top failure or System failure)을 수형도(tree chart)로 탐색하여 어떤 부품이 고장의 원인인가를 찾아내는 연역적 해석기법으로, 시스템의 고장요인들의 관계를 Boolean Logic Gate를 이용, 도해적으로 표현하여 분석하는 기법이다. (표 2-1 참조)

표 2-1. Fault Tree 작성에 사용하는 기호

구 분	기 호	명 칭	설 명
EVENT 기호		Fault Event	기본고장의 결합으로 이루어진 고장의 상태
		Basic Event	주어진 시스템의 기본 사상
		Diamond	더 분석이 가능하나 기본사상으로 가정
		Transfer	다른 Gate로 나가는 사상
		Transfer	다른 Gate로 부터 들어오는 사상
LOGIC 기호		AND Logic Gate	모든 입력사상이 참일 경우 출력이 참이 됨
		OR Logic Gate	한개 또는 그 이상의 입력이 참이면 출력은 참이 됨

FTA는 1961년 Bell 전화연구소의 H.A. Watson이 Minuteman Launch Control System의 안전성을 평가분석할 때 처음 사용하였으며 [26], Boeing사의 분석자들은 디지털 컴퓨터를 사용하여 정량적인 측면에서 이 기법을 발전시켰다. [14]

그 후 Hass[21]이 FTA에 관한 논문을 발표한 이래 많은 사람들이 항공학, 원자력공학 등 다른 여러 분야의 신뢰성과 안전성 분석에 대한 연구를 활발히 추진해 오고 있다.

2.2 Fault Tree Analysis의 분석절차

FTA의 분석절차는 일반적으로 (1) 시스템의 정의 (2) Fault Tree 작성 (3) Fault Tree Evaluation의 순으로 이루어진다.

우선 먼저 분석하고자 하는 시스템에 대한 정확한 정의와 충분한 기술적 검토가 선행되어야 한다. 그 다음에는 Fault Tree를 작성해야 하는데 이는 모든 정보가 정리되고 분석되는 과정이기 때문에 매우 중요하게 다루어져야 한다. FTA에 대해서는 D. Hassl[21]에 의해 일반적 개념이 논의된 이후 많은 발전을 가져왔다.

J.B. Fussel[14] [16]은 전기시스템에서의, Powers & Tompkins[33]는 화학 공정시스템에서의 FT 작성기법을 개발했다. Lapp & Power[23] [24]는 Diagraph(directed graph)을, Taylor & Hollo[38]는 CCD(Cause-Consequence Diagram)를 사용하여 FT를 작성했고 Carmarda[7]등은 거대한 시스템에 대한 신뢰도 그래프로부터 FT의 자동통합을 위한 효과적인 알고리즘을 제시했다.

마지막으로 Fault Tree Evaluation은 정성적 평가(Qualitative Evaluation)와 정량적 평가(Quantitative Evaluation)로 구분된다. 정성적 평가는 minimal cutsets과 minimal path sets 그리고 common-cause failure를 결정하는 것이다. 여기서 minimal cut sets는 고장을 유발시키는 기본고장들의 조합으로 구성되는 집합인 cut sets 중 중복되는 집합을 제거한 것으로서 이를 찾는 기본적 방법에는 Monte Carlo Simulation과 결정론적 방법(deterministic methods)이 있다.

Fussel & Vesely는 AND gate는 cut sets의 크기를 증가시키고 OR gate는 cut sets의 수를 증가시킨다는 사실을 기본으로 하여 'MOCUS'라는 유용한 알고리즘을 만들었다. [15] 이외에도 지금까지 연구되어진 정성적

분석방법에는 PREP[41], MICSUP[8][31], ELRAFT[37], FAUTRAN[43], SETS[44], FATRAM[34], DICOMIC[17], BAM-CUTS[13], BUP-CUTS[29] 등이 있다.

정량적 평가는 모든 기본 사상들의 발생률과 고장지속시간(failure duration) 그리고 각 기본사상들의 통계적 종속도(statistical dependency)에 따라 Top 사상의 통계적 기대치나 확률을 구하는 방법이다.

H.E. Lambert는 FT의 기본사상이나 cut sets의 중요도를 계산하기 위해 'IMPORTANT'라는 유용한 알고리즘을 개발했다. [25] 이밖에도 정량적 분석방법에는 SAFTE[19], Crosetti.code[10], SAMPLE[35], REDIS[22], Kitt[40][41], Caldarola/Wickenhouser[6], PL-MOD[30], GO[20], WAM-BAM[13], PATREC[4], SALP[1] 등이 있다.

그러나 정량적 평가는 시스템의 고장율과 수리시간등의 자료획득이 어렵기 때문에 쉽게 적용하기가 어려운 측면이 있다.

3. FTA에 의한 사례연구

3.1 사례연구의 일반 사항

(1) 대상 업체

집진시설, 송풍기, 폐수처리시설 등 공해방지설비를 전문생산하는 서울 구로구에 소재하는 D주식회사를 대상으로 하였음.

(2) 연구일정

사례연구는 1989년 8월 7일 부터 8월 20일까지 실시함.

(3) BAG FILTER를 연구대상으로 한 이유

BAG FILTER의 고장으로 인한 집진시설의 불가동은 작업사의 호흡기 장애등 산업 등 산업재해를 유발시킬 뿐 아니라 작업능률의 저하 등 바람직하지 못한 결과를 초래할 수 있기 때문에 이 시스템의 고장요인을 체계적으로 분석, 그 대책을 강구코자 함.

(4) BAG FILTER의 구조와 원리

(a) 고체입자를 동반한 기체는 배풍기에 의해 BAG FILTER에 유입

(b) BAG FILTER의 여과포를 통과하는 동안 고체입자는 여과포에 걸리지고 청정공기만 외부로 나감

(c) 여과포에 부착된 고체기체는 Ventury를 통해 분사되는 고압의 공기에 의해 Hopper로 떨어져 Rotary Feeder(Screw Feeder)를 거쳐 외부로 이송되나 일부 미립자는 기류와 함께 상승하여 여과포에 부착되고 그로 인해 압력 손실을 초래

(d) 이 압력 손실을 조절하기 위해 Timer가 주기적으로 작동하여 전자변에 통전되면 Pulse Valve가 작동하여 고압의 공기를 Blow Tube의 Nozzle을 통하여 Ventury로 분사시킴

(e) 분사된 고압공기는 Ventury에서 고속으로 흡입하여 여과포를 급격히 팽창시키고 그 Shock와 역류하는 기류에 의해 외부에 부착된 고체 입자는 떨어져 Hopper 속으로 낙하함

(5) BAG FILTER의 기능

(a) 0.05 정도의 분진도 99.9% 포집이 가능하여 집진 효율이 높음.

(b) 소용량에서 부터 대용량까지 처리 가능

(c) 장기간 사용해도 성능은 거의 불변

3.2 사례연구의 분석절차

집진기(BAG FILTER)에 대한 신뢰성 분석은 다음과 같은 순서에 의해 실시된다.

(1) Fault Tree 작성

(2) Minimal Cut Sets의 결정

(3) Inversion에 의한 Minimal Path Sets의 결성

(4) Minimal Path Sets에 의하여 SI 분석 실시

(a) Critical Path Vector 수에 의한 분석

(b) Minimal Path Sets에 출현하는 각 기본사상의 출현회수에 의한 분석

- (5) 결과분석
- (6) PC 전산화

3.3 Fault Tree 작성

Bag Filter의 작성에는 현장에서 얻어진 고장수리 현황자료를 이용하였다. Bag Filter의 고장요인 분류코드와 Fault Tree가 표 3-1과 그림 3-1에 수록되었다. (고장요인 분류는 컴퓨터 처리시간의 단축을 위해 가급적 단순화 하려고 했음)

표 3-1. BAG FILTER의 고장요인 분류코드

분 류	분류코드	분류항목
송풍기 고장	× 1	베어링 이상
임페라 이상	× 2	먼지 낄
	× 3	unbalancing
타이머 고장	× 4	전원 공급 이상
	× 5	가변코일 파괴
밸브 고장	× 6	밸브고장
	× 7	다이아후램 밸브고장
집진력 저하	× 8	습기 참
로터리 밸브이상	× 9	설계미스
	×10	사용부주의

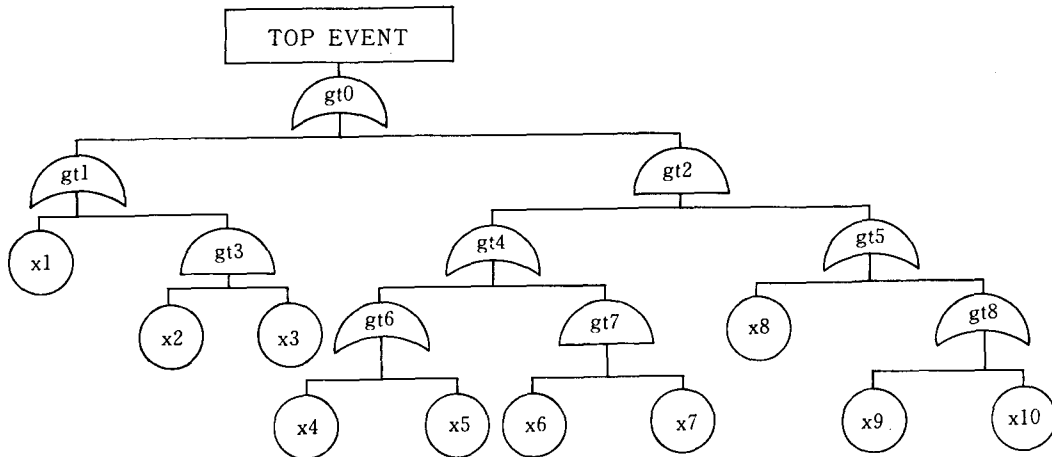


그림 3-1. Bag Filter의 Fault Tree 작성

3.4 Minimal Cut Sets의 결정

Cut Set이란 시스템의 고장을 유발시키는 기본사상들의 조합으로 구성되는 집합으로서 그 집합에 포함된 기본사상이 일어났을 때 시스템의 고장이 발생하는 것을 의미한다. Minimal Cut Set이란 Cut Sets중 중복되는 집합을 제거한 후 남은 Cut Set으로서 Top사상을 일으키기 위한 최소한의 Cut Set를 의미한다.

본 논문에서는 FATRAM 원리 [34]를 이용하여 Minimal Cut Sets을 결정한다.

3.4.1 원리 및 방법

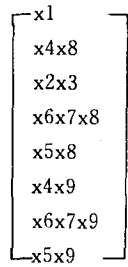
step 1. 작성된 FT의 시스템고장으로 부터 분석하며 and logic gate에 대한 input은 하나의 set으로 나열하고 or logic gate에 대한 input은 별개의 set으로 나열한다.

step 2. 기본고장만을 input으로 갖는 or logic gate를 제외한 모든 and, or gate를 분석한다.

- step 3. 중복되는 cut set를 제거한다.
- step 4. 아직 분석되지 않은 or logic gate에서 반복되는 기본고장 즉, 반복사상을 분석한다.
- step 5. 대체된 반복사상은 고려대상에서 제외시키고 나머지 반복사상을 분석한다.
- step 6. 누적된 set에 남아있는 or logic gate를 각 gate가 갖는 반복사상으로 대체한다.
- step 7. 분석되지 않는 즉, 반복사상을 포함하지 않은 or logic gate를 분석한다.
- step 8. 중복되는 set를 제거시키면 나머지 set가 minimal cut set가 된다.

3.4.2 Minimal Cut Sets의 결정

시스템고장은 Top이고 기본사상이 x1, x2, ..., x9, x10이다. 부록에 수록된 컴퓨터 프로그램을 수행시킨 결과 주어진 시스템의 Minimal Cut Set은 다음과 같이 구해진다.



3.5 Inversion에 의한 Minimal Path Set의 결정

Path Set이란 Cut Set의 쌍대미로서 시스템의 고장이 발생하지 않도록 하는 기본사상들의 조합으로 된 집합을 말한다. 즉 그 집합에 포함된 기본사상이 일어나지 않을 때 시스템의 고장이 발생하지 않는 것을 의미한다.

Minimal Path Set이란 Path Set 중 중복되는 집합을 제거한 후 남은 Path Set으로서 시스템의 고장이 일어나지 않도록 하는 최소한의 Path Set를 말한다.

본 논문에서는 Locks의 원리 [27] [28]를 이용해서 Minimal Cut Sets를 구한다.

3.5.1 원리 및 방법

step 1. 다음과 같은 De Morgan의 정리에 의하여 앞에서 구한 Minimal Cut Sets을 전개하면 Path Sets를 구할 수 있다.

$$\overline{x_1 \ x_2 \ x_3 \ \dots \ x_j} = \bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \dots + \bar{x}_j \quad (\text{De M. 1})$$

$$\overline{\bar{x}_1 + \bar{x}_2 + \bar{x}_3 \ \dots + \bar{x}_j} = x_1 \ x_2 \ x_3 \ \dots \ x_j \quad (\text{De M. 2})$$

step 2. 구해진 Path Sets 중 중복되거나 다른 Path Set에 포함되는 것을 제거하면 Minimal Path Sets이 구해진다.

$$\begin{aligned} & \overline{\overline{x_1 + x_4x_8 + x_2x_3 + x_6x_7x_8 + x_5x_8 + x_4x_9 + x_6x_7x_9 + x_5x_9}} \\ & = \bar{x}_1 \ \bar{x}_4x_8 \ \bar{x}_2x_3 \ \bar{x}_6x_7x_8 \ \bar{x}_5x_8 \ \bar{x}_4x_9 \ \bar{x}_6x_7x_9 \ \bar{x}_5x_9 \\ & = \dots \end{aligned}$$

(이런 방법으로 계산하면)

$$= \bar{x}_1 \ \bar{x}_2 \ \bar{x}_4 \ \bar{x}_5 \ \bar{x}_6 + \bar{x}_1 \ \bar{x}_3 \ \bar{x}_4 \ \bar{x}_5 \ \bar{x}_6 + \bar{x}_1 \ \bar{x}_2 \ \bar{x}_8 \ \bar{x}_9 \ \bar{x}_{10} +$$

$$\bar{x}_1 \ \bar{x}_2 \ \bar{x}_4 \ \bar{x}_5 \ \bar{x}_7 + \bar{x}_1 \ \bar{x}_3 \ \bar{x}_8 \ \bar{x}_9 \ \bar{x}_{10} + \bar{x}_1 \ \bar{x}_3 \ \bar{x}_4 \ \bar{x}_5 \ \bar{x}_7$$

따라서 Minimal Cut Sets는

x1	x2	x4	x5	x6
x1	x3	x4	x5	x6
x1	x2	x8	x9	x10
x1	x2	x4	x5	x7
x1	x3	x8	x9	x10
x1	x3	x4	x5	x7

여기서 Minimal Cut Sets의 의미는 기본사상이 5개의 조합으로 된 위의 Minimal Cut Sets이 발생하지 않을 경우 Top 사상이 일어나지 않는다는 것을 의미한다.

4.5 Minimal Path Set에 의한 SI 분석

4.5.1 Critical Path Vector의 수에 의한 SI 분석

SI(Structural Importance) 분석은 어떤 시스템의 신뢰성 또는 안전성을 제고하기 위한 대책을 강구하고자 할 때 각 기본사상의 정량적인 자료 즉, 고장율, 고장수리시간 등을 얻기 어려운 경우 어떤 기본사상이 Top 사상의 발생에 어느정도 기여하는가를 파악하거나 어떤 기본사상이 Top 사상을 일어나지 않도록 하는데 어느 정도 기여하고 있는가를 정량적으로 알기 위한 방법이다.

여기서는 앞에서 구한 Minimal Cut Sets의 각 사상에 대하여 SI 분석을 실시하여 본다.

step 1. 기본사상 상태표를 작성한다.

이 상태표는 Minimal Path Sets에 포함된 각 기본사상의 발생, 미발생에 따라 Top 사상의 발생, 미발생여부를 나타내주는 표를 말한다.

step 2. CPV(Critical Path Vector)의 총수를 구한다. CPV는 기본사상이 n이 $\phi(0n, x)=1$ 일 경우를 가리킨다.

여기서 사용되는 기호에 대하여 정의를 내려보면 다음과 같다.

N개의 기본사상이 있다고 가정하자.

$X_n=1n$: 기본사상 n이 발생

$0n$: 기본사상 n이 발생 안함

$X_{top}=1$: Top 사상이 발생

0 : Top 사상이 발생 안함

$X_{top}=\phi(x)=\phi(x_1, \dots, x_n)$

이때 ϕ 를 Fault Tree의 구조함수라 함

$(0n, X)=(X_1, \dots, X_{n-1}, 0, X_{n+1}, \dots, X_n)$

$(2n, X)=(X_1, \dots, X_{n-1}, 1, X_{n+1}, \dots, X_n)$

i번째의 기본사상의 Critical Path Vector의 총수 CPV(i)는

$$CPV(i)=[\phi(1i, X) - \phi(0i, X)] \dots \dots \dots (1)$$

식 (1)의 의미는 어떤 기본사상의 상태가 1인 경우와 0인 경우를 비교하여 Top 사상이 1에서 0으로 바뀌는 수 즉, CPV가 총 몇개인가를 계산하는 것이다.

step 3. 다음 계산식에 의해 SI(Structural Importance)를 구한다.

$$SI(i)=CPV(i)/2^{n-1}$$

위와 같은 방법으로 부록에 수록된 컴퓨터 프로그램에 의해 CPV(i)와 SI를 구해보면 다음과 같다.

CPV(i):

- CPV(X 1)=111
- CPV(X 2)= 37
- CPV(X 3)= 37
- CPV(X 4)= 63
- CPV(X 5)= 63
- CPV(X 6)= 21
- CPV(X 7)= 21
- CPV(X 8)= 39
- CPV(X 9)= 39
- CPV(X10)= 39

SI(i):

- SI(X 1)=111/(2**9)
- SI(X 2)= 39/(2**9)
- SI(X 3)= 37/(2**9)
- SI(X 4)= 63/(2**9)
- SI(X 5)= 63/(2**9)
- SI(X 6)= 21/(2**9)
- SI(X 7)= 21/(2**9)
- SI(X 8)= 39/(2**9)
- SI(X 9)= 39/(2**9)
- SI(X10)= 39/(2**9)

SI(i)를 구해본 결과 X1이 Top 사상을 발생하지 않게 하는데 가장 큰 영향을 주고 있다는 것을 알 수 있다.

4.5.2 Minimal Path Sets에 출현하는 각 기본사상의 횟수에 따른 SI분석

각 기본사상이 모두 같은 확률로 발생한다고 가정하면 각 기본사상이 전체 시스템고장이 일어나지 않도록 하는데 미치는 영향은 Minimal Path Sets에 출현하는 횟수에만 관계된다. [2]

각 기본사상이 Minimal Path Sets에 출현하는 횟수에 따른 SI 분석을 MIL-STD-105D¹⁾에 기준하여 행한다. Minimal Path Sets의 수를 n이라 할 때 그 분류기준은 다음 표와 같다.

SI 분석의 분류 기준

SI	의 미	기본 사상이 Minimal Path Set에 출현하는 횟수
IE (Insignificant Effect)	전체 시스템의 가동에 아무런 영향을 미치지 않는다.	0
CE (Critical Effect)	전체 시스템의 가동에 절대적인 영향을 미친다.	n
MA (Major Effect)	전체 시스템의 가동에 중요한 영향을 미친다.	n-1
ME (Medium Effect)	전체 시스템의 가동에 비교적 많은 영향을 미친다.	n-2
MI (Minor Effect)	전체 시스템의 가동에 비교적 적은 영향을 미친다.	n-3회 이하

(n : Minimal Path Set의 총수)

1) Military-Standard-105 Defective의 약자로서 1963년 미 국방성에서 발표한 계수조정형 샘플링 검사를 위한 기준표임.

본 사례연구에 대한 SI 분석 결과는 다음과 같다.

SI	기본 사상이 Minimal Path Sets에 출현하는 횟수	기본사상
IE	0회	
CE	6회	x1
MA	5회	
ME	4회	x4, x5
MI	3회 이하	x2, x3, x6, x7, x8, x9, x10

SI 분석을 실시한 결과 x1은 시스템의 가동에 가장 큰 영향을 주고 있으며 x4, x5는 비교적 많은 영향을 주는 것으로 나타났다.

4.6 결과분석

본 사례연구의 고장분석 결과를 보면 SI 값이 가장 큰 기본사상인 x1에 대해 보다 많은 주의와 관심을 가짐으로써 x1이 발생하지 않도록 할 필요가 있다. 아울러 기본 사상 x4, x5에 대해서도 각별히 신경을 써서 이들이 발생하지 않도록 해야 할 것이다.

이와 같이 SI 분석 결과 SI 값이 큰 기본사상 x1, x4, x5에 대해서는 집중관리를 해 나갈 필요가 있다. 즉, x1은 송풍기의 베어링 이상에 관한 것으로 마손으로 인한 송풍기고장을 미연에 방지할 수 있도록 수시점검을 요하는 한편 베어링 조립시 잘못된 일이 없도록 작업자에 대한 철저한 교육과 감독을 시행할 필요가 있다.

또한 탈진장치내의 타이머의 고장요인이 전원공급상의 문제(x4)와 가변코일의 파괴(x5)가 Bag Filter의 고장에 비교적 많은 영향을 미치고 있는 것으로 나타났다.따라서이의 직접원인으로 알려지고 있는 합선이나 결선 조작의 잘못으로 인한 휴즈단선등이 발생하지 않도록 작업자에 대한 철저한 주의환기 및 안전교육실시가 요망된다.

4.7 PC 전산화

step 1. Minimal Cut Sets의 결정

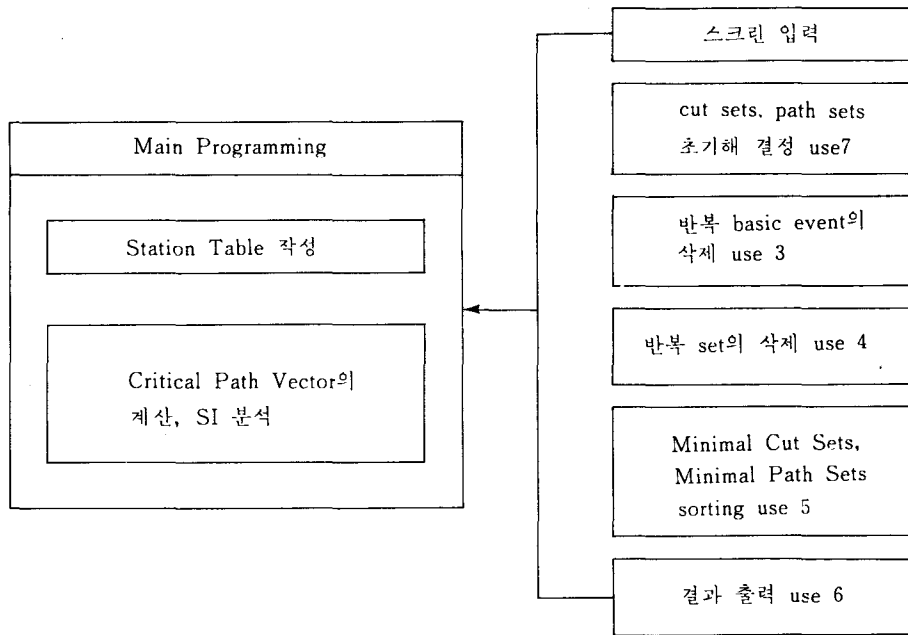
Fault Tree의 gate와 기본사상이 많은 경우에는 Minimal Cut Sets을 수작업으로 계산하기가 어렵기 때문에 컴퓨터를 이용하지 않으면 안된다.

Minimal Cut Sets을 구하는 컴퓨터 프로그램은 많이 보급되었다. 본 논문에서는 최근 개인용 컴퓨터의 급속한 보급에 따라 이 프로그램의 대중화를 도모하고자 PC로 작성하였다. 또한 기존의 프로그램은 고급언어인 FORTRAN이나 PASCAL로 쓰여졌으나 여기서는 보다 작성하기 쉽고 편리한 캣키지 언어인 dBaseIII+를 이용하였다.

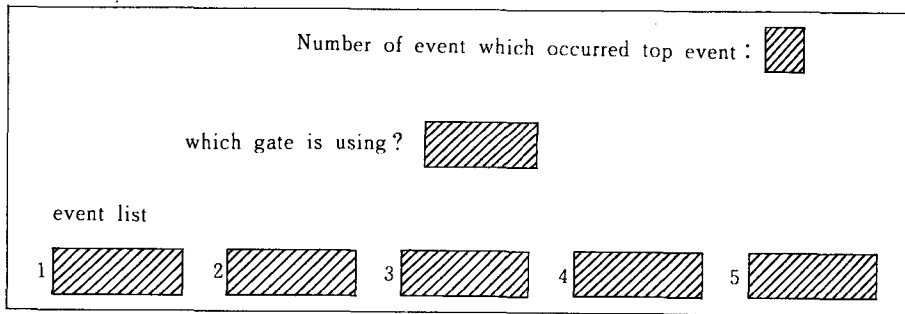
step 2. Inversion에 의한 Minimal Path Set의 결정과 Critical Path Vector의 수에 의한 SI의 결정

Minimal Path Sets의 결정과 SI 분석은 다음과 같은 과정으로 이루어졌다.

(1) Flowchart



(2) Input Model



5. 결론

본 논문에서는 PC를 이용하여 FTA 기법을 응용하는 방법을 보였다. 고장률, 고장수리시간 등 정량적인 자료를 얻기 어려운 시스템을 예제로 채택하여 Minimal Cut Sets과 Minimal Path Sets를 결정하고 SI 분석을 실시하였다.

특히 본 논문에서는 고장사상을 가동사상의 값으로 변환하여 각 기본사상이 시스템의 가동에 어느 정도 기여하는가를 정량적으로 분석해 보았다.

또한 PC의 대중화 추세에 부응하여 FTA 기법을 보다 널리 보급하기 위해서 컴퓨터 프로그램을 PC 맷키지인 dBase III +를 이용하여 작성하였다.

PC를 이용한 전산화 과정은 다음과 같은 단계로 이루어 졌다.

- step 1. FATRAM의 원리를 이용하여 Fault Tree로부터 Minimal Cut Sets를 결정한다.
 - step 2. Locks의 원리를 이용해서 step 1에서 구해진 Minimal Cut Sets를 Inversion하여 Minimal Path Sets를 결정한다.
 - step 3. step 2에서 구해진 Minimal Path Sets에 의해 SI 분석을 실시하여 각 기본사상의 SI를 산출한다.
- 끝으로 본 연구는 여러 가지 제안을 안고 있는데 앞으로 이의 해결이 요망된다.

첫째 PC의 한계성으로 인해 기본사상이 많을 경우 프로그래밍 하기가 쉽지 않을 뿐만 아니라 컴퓨터 처리시간이 너무 많이 걸린다는 점이다. 앞으로 32bit PC가 많이 보급되면 이 문제는 어느 정도 해결되리라고 보여지나 보다 처리시간 단축을 위한 근본적인 대책이 강구되어야 할 것이다.

둘째, 시스템의 상태가 가동, 고장만으로 구성되어 있다는 전제하에서 신뢰성 분석이 행하여 졌는데 그 이외의 상태 즉 중간적 상태와 인간에 의해 발생하는 고장 모드를 고려한 분석도 시도해 볼 필요가 있다고 본다. 셋째 각 고장모드에 대한 경제성 분석을 실시하여 가장 합리적인 신뢰성 향상 방안을 마련하여야 할 것이다.

참 고 문 헌

1. Astolfi, M., et al. "Fault Tree Analysis by List-Processing Techniques," IEEE Trans. Reliability, Vol. R-27, Oct., 1978, pp. 5-31
2. Barlow, R. E. and F. Proschan, "Statistical theory of reliability and life twsting, Holt, Rinehart Winston, INC., 1975.
3. Bennetts, R., "On the Analysis of Fault Tree," IEEE Trans. on Rel., Vol R-24, No. 3, Aug., 1975.
4. Blin, A. and A. Carline, et al, "PATREC, A Computer Code for Fault Tree Calculation," IEEE Trans, Reliability, Vol. R-27, Oct., 1978, pp. 33-43.
5. Burdick, G. R., "COMCAN-A computer code for common-cause analysis," IEEE Trans. on Rel., Vol. R-26, No. 2, June, 1977.
6. Caldarol, L., "Coherent Systems with Multistate Components," Nuclear Engineering and Design, Vol. 58, 1980, pp. 127-139.
7. Camarda, F. Corsi, A. Trentadue, "An Efficient Simple Algorithm for Fault Tree Automatic Synthesis from Reliability Graph," IEEE Trans. Reliability, Vol, R-27, Aug., 1978, pp. 215-221.
8. Chatterjee, P., "Fault Tree Analysis:Minimal Cut set Algorithm," ORC 74-2, Operations Research Center, University of California, Berkeley, California, Jan., 1974.
9. Crosetti, P. A., "Fault Tree Analysis with Probability Evaluation," IEEE Trans. Nuclear Science, Vol. NS-18, No. 1, Feb., 1971.
10. Crosetti, P. A., "Fault Tree Analysis for Systems Reliability," Instrumentation Technology, Aug., 1971, pp. 52-56.
11. Dhillon, B. S. and C. Singh, "On Fault Trees and Other Reliability evaluation Methods, Microelectronics and Reliability, Vol. 19, 1979, pp. 57-63.
12. Dhillon, B. S. and C. Singh, Engineering Reliability, John Wiley and Sons, Inc., 1981.
13. Erdmann, R. C., et al, "A Method for Quantifying Logic Models for Safety Analysis," Nuclear Science and Engineering, 1973, pp. 732-754.
14. Fussell, J. B., "A Formal Methodology for Fault Tree Construction," Nuclear Science and Engineering, 1973.
15. Fussell, J. B. and E. B. Henry, et al, "MOCUS-A Computer Program to obtain Minimal Sets from Fault Trees," ANCR-1156, Aerojet Nuclear Company, IDAHO, March, 1874.
16. Fussell, J. B., et al, "Fault Tree-A State of The Art Discussion," IEEE Trans. Reliability, Vol. R-23, No. 1, 1974.
17. Garribba, S., et al., "DICOMICS, An Algorithm for Direct Computation of Minimal Cut Sets of Fault Trees," EUR-5481E, 1975.
18. Garribba, S., and P. F., Mussio, et al, "Efficient Construction of Minimal Cut sets from Fault Trees," IEEE Trans. on Rel., Vol. R-26, June, 1977.
19. Garrick, B. J., "Principles of Unified System Safety Analysis", Nuclear Engineering and Design, Vol. 13, 1970, pp. 245-321.
20. Gately, W. Y., et al, "GO, A Computer Program for the Corporation, Colorado Springs, Colorado, KN-67-704(R), Apr., 1968.

21. Haasl, D. F., "Advanced Concepts in Fault Tree Analysis," System Safety Symposium, June, 1965.
22. Kongsoe, H. E., "REDIS, A Computer Program for System Reliability Analysis by Direct Simulation," Int'l Symp. Reliability of Nuclear Powers Plants, Innsbruct, Austria, April, 1975, pp. 14-18.
23. Lapp, S. A., G. J. Powers, "Computer-aided Synthesis of Fault Trees," IEEE Trans. Reliability, 1977, Vol. R-28, April, 1979, pp. 6-9.
24. Lapp, S. A., G. J. Powers, "Update of Lapp-Powers Fault Tree Synthesis Algorithm," IEEE Trans. Reliability, Vol. R-28, Apr. 1979, pp. 12-14.
25. Lambert, H. E., "Fault Trees for Decision Making in System Analysis," Lawrence Livermore Laboratory, University of California, Livermore, UCRL-51829, Oct., 1975.
26. Le, W. S., et al, "Fault Tree Analysis, Methods, and Applications," IEEE Trans, Reliability, Vol. R-34, No. 3, Aug., 1985, pp. 194-203.
27. Locks, M. O., "Inverting and Minimalizing Path Sets and Cut Sets," IEEE Trans, Reliability, Vol. R-27, No. 2, June, 1978, pp. 107-109.
28. Locks, M. O., "Relationship Between Minimal Path Sets and Cut sets," IEEE Trans, Reliability, Vol. R-27, No. 2, June, 1978, p. 106.
29. Nakashima, K. and Y. Hattori, "An Efficient Bottom-Up Algorithm for Enumerating Minimal Cut sets of Fault Trees," IEER Trans. Relia., Vol. R-28, Dec., 1979, pp. 353-357.
30. Olmos, J. and L. wolf, "A Modular Representation and Analysis of Fault Trees," Nuclear Engineering and Design, Vol.
31. Pande, P. K., et al., "Computerized Fault Tree Analysis," TREEL and MICSUP, ORC 75-3, Operations Research Center, University of California, Berkely, April, 1975.
32. Power, G. J., "Computer aided Synthesis of Fault Trees," IEEE Trans. on rel., April, 1977.
33. Power, G. J. and dF. C. Tompkins, "Fault Tree Syndthesis for Chemical Process," AIChE Journal, Vol. 20, Mar., 1974.
34. Rasmuson, D. M. and N. H. Marshall, "FATRAM-A Core Efficient Cut set Algorithm," IEEE Trans. Reliability, Vol. R-27, Oct., 1978, pp. 250-253.
35. Reactor Safety Study-An Assessment of Accident Reisk in U. S. Commercial Nuclear Powerplants, WASH-1400 (NUREG-75/014), US Nuclear Regulatory Commission, Washington, DC., Oct., 1975.
36. Salvendy, G., Handbook of Industrial Engineering, John Wiley and Sons, 1982.
37. Semanderes, S. N., "ELRAFT, A Computer Program for the Efficient Logic Reduction Analysis of Fault Trees," IEEE Trans. Nuclear Science, Vol. NS-18, Feb., 1971, pp. 481-487.
38. Taylor J. B., E. Hollo, "Algorithm and Programs for Consequence Diagram and Fault Tree Construction, Roskilde, Denmark, 1977.
39. Van Slyke, W. J and D. E. Griffing, "ALLCUTS, A Fast Comprehensive Fault Tree Analysis Code," Atlantic Richfield Hanford Company, Richard, Washington, ARH-ST-112, July, 1975.
40. Vesely, W. E., "Reliability and Fault Tree Applications at NRT," Proc. 1970 Reliability and Maintainability conf., Vol. 9, 1970, pp. 472-480.
41. Vesely, W. E. and R. E. Narum, "PREP and Kitt Computer Code for The Automatic Evaluation of a Fault Tree," Idaho Nuclear Corporation, Idaho Falls, Idaho, IN-1349, 1970.
42. Wheeler, D. B. and J. S. Hsun, et al., "Fault Tree Analysis using Bit Manipulation," IEEE Trans. on Rel., Vol. R-26, No. 2, June, 1977.
43. Wong, P. Y., "FAUTRAN-A FAULT Tree Analyzer," AECL-5182, Atomic Energy of Canada Limited, Chalk River Nuclear Lab. Chalk River, Ontario, Canada, 1975.
44. Worrell, R. B., "Set Equation Transformation System(SETS)," SLA-73-0028A Sandia Laboratories, Albuquerque, New Mexico, May, 1974.

〈부록 1〉 OUTPUT LIST

** Minimal cut sets list **	
minimal cut set index	basic event
1	x1
2	x4 * x8
3	x2 * x3
4	x6 * x7 * x8
5	x5 * x8
6	x4 * x9
7	x6 * x7 * x9
8	x5 * x9

** Minimal path sets list **	
minimal path set index	basic event
1	x1 * x2 * x4 * x5 * x6
2	x1 * x3 * x4 * x5 * x6
3	x1 * x2 * x8 * x9 * x10
4	x1 * x2 * x4 * x5 * x7
5	x1 * x3 * x8 * x9 * x10
6	x1 * x3 * x4 * x5 * x7

** Structural importance analysis using critical path vector **	
structural importance of x 1=	111/(2**9)
structural importance of x10=	39/(2**9)
structural importance of x 2=	37/(2**9)
structural importance of x 3=	37/(2**9)
structural importance of x 4=	63/(2**9)
structural importance of x 5=	63/(2**9)
structural importance of x 6=	21/(2**9)
structural importance of x 7=	21/(2**9)
structural importance of x 8=	39/(2**9)
structural importance of x 9=	39/(2**9)

〈부록 2〉 PROGRAM LIST

```

• type main.prg
set bell off
*
*Fault tree analysis using PC program
*
set talk off
set safety off
clear all
select b
use exam1
select a
use sample index isample

```

```

h="top event"
*
*screen edit
*
do use 1
append blank
replace event with "top", calcul with top. gate with g select b
a=at("+", top)
if a<>0
  b=(len(top)+1)/5
  x=0
  do while x<b
    append blank
    replace cut with substr(top,5*x+1,4), len with 4
    x=x+1
  enddo
else
  append blank
  replace cut with top. len with len(top)
endif
go top
clear
no=""
@ 5,5 say "Total number of basic event?" get no
read
no=val(no)
kk="+"
do use 7
*
*basic event elimination of duplication for minimal cut sets
*
do use 3
*
*minimal cut sets sorting
*
select a
use sort index isort
select b
do use 5
*
*basic event elimination of redundancy for minimal cut sets
*
do use 4
*
*inversion of minimal cut sets
*

```

```

select a
use sample index isample
find "top"
aa=5*(int(len(trim(calcul))/5))+4-len(trim(calcul))
top=trim(calcul)+space(aa)
select b
use exam 2
a=at("*", top)
if a<>0
  b=(len(top)+1)/5
  x=0
  do while x<b
    append blank
    replace cut with substr(top, 5*x+1,4), len with 4
    x=x+1
  enddo
else
  append blank
  replace cut with top, len with len(top)
endif
go top
kk="*"
do use 7
*
*basic event elimination of duplication for minimal path sets
*
do use 3
*
*basic event elimination of redundancy for minimal path sets
*
do use 4
*
*minimal path sets sorting
*
select a
use sort index isort
select b
do use 5
*
*making of station table
*
select a
use state index istate
a=2**no-1
x=0
do while x<a+1

```

```

z=x
y=0
do while .t.
  b=int(z/2)
  c=z-2*b
  y=y+1
  if y=1
    d=str(c,1)
  else
    d=str(c,1)+d
  endif
  if b=0
    exit
  else
    z=b
  endif
enddo
e=no-y
if e<=0
  z=0
  do while z<0
    d="0"+d
    z=z+1
  enddo
endif
append blank
replace st with d
x=x+1
enddo
go top
*
*calculation of critical path vector
*
do while .not. eof( )
  select b
  go top
  do while .not. eof( )
    d=(len+1)/5
    x=0
    do while x<d
      e=substr(cut, 5*x+1, 4)
      f=val(substr(e,2,3))
      if substr(a->st, f, 1)="0"
        exit
      endif
      x=x+1
    enddo
  enddo

```

```

        if x=d
            replace a->t with "1"
            exit
        endif
        skip
    enddo
    if eof( )
        select a
        replace t with "0"
        select b
    endif
    select a
    skip
enddo
go top
x=1
do while x<no+1
    y=0
    d=0
    locate next 20000 for substr(st, x, 1)="1"
    do while .t.
        if eof( )
            exit
        endif
        b=recno( )
        if t="1"
            if x=1
                c="0"+substr(st, 2, no-1)
            else
                if x=no
                    c=substr(st, 1, no-1)+"0"
                else
                    c=substr(st, 1, x-1)+"0"+substr(st,x+1, no-x)()
                endif
            endif
            find & c
            if t="0"
                if y=0
                    d=1
                    y=y+1
                else
                    d=d+1
                endif
            endif
        endif
    enddo
    go b
    continue

```



```
    enddo
  if x<10
    e="x"+str(x,1)
  else
    if x<100
      e="x"+str(x,2)
    endif
  endif
  & e=d
  x=x+1
  go top
enddo
use out index iout
x=1
do while x<no+1
  if x<10
    e="x"+str(x,1)
  else
    if x<100
      e="x"+str(x,2)
    endif
  endif
  append blank
  replace xn with "&e", nn with &e
  x=x+1
enddo
set print on
set console off
set device to printer
select b
*
*report generation
*
do use 6
set print off
set console on
set device to screen
delete all
pack
use state index istate
delete all
pack
close database
.type use 1. prg
*
*screen edit
*
```

```

public top, g, f
store " " to a, b, c, d, e
f=" "
g=" "
@5,5 to 16,60
@7,7 say "Number of event which occurred "+h+": get f: picture "⚡"
@9,7 say "Which gate is using?" get g picture "aax"
@12,7 say "event list"
@14,10 say "1" get a
@14,20 say "2" get b
@14,30 say "3" get c
@14,40 say "4" get d
@14,50 say "5" get e
read
if upper(g)="AND"
  do case
    case f="1"
      top=a
    case f="2"
      top=a+"*" + b
    case f="3"
      top=a+"*" + b+"*" + c
    case f="4"
      top=a+"*" + b+"*" + c+"*" + d
    case f="5"
      top=a+"*" + b+"*" + c+"*" + d+"*" + e
  endcase
else
  case
    case f="1"
      top=a
    case f="2"
      top=a+"+" + b
    case f="3"
      top=a+"+" + b+"+" + c
    case f="4"
      top=a+"+" + b+"+" + c+"+" + d
    case f="5"
      top=a+"+" + b+"+" + c+"+" + d+"+" + e
  endcase
endif
return
. type use 3. prg
*
*elimination of duplication
*
```

```

go top
do while .not. eof( )
  c=(len+1)/5
  store 0 to x,y
  if c<>1
    ct=cut
    do while c>x+1
      e=substr(ct, 1, 4)
      f=substr(ct, 6, len-5)
      if at(e,f)=0
        y=y+1
        if y=1
          y=1
          d=e
          in=4
        else
          d=d+"*"+e
          in=in+5
        endif
      endif
      ct=f
      x=x+1
    enddo
    if y<>0
      d=d+"*"+ct
      in=in+5
    else
      d=ct
      in=4
    endif
    replace cut with d, len with in
  endif
  skip
enddo
go top
return

```

```
. type use 4= .prg
```

```
*
```

```
*elimination of redundancy
```

```
*
```

```

do while .not. eof( )
  a=cut
  b=recno( )
  go top
do while .not. eof( )

```

```

    if recno( ) <> b
        i = at(trim(cut), a)
        if i <> 0
            go b
            delete record b
            pack
            go top
            exit
        else
            skip
        endif
    else
        skip
    endif
enddo
if eof( )
    go b
    skip
endif
enddo
go top
return

. type use 5. prg
*
*basic event sorting
*
do while .not. eof( )
    select a
    c = (b->len+1)/5
    x = 0
    do while x < c
        d = substr(b->cut, 5*x+1, 4)
        e = val(substr(d, 2, 3))
        append blank
        replace seq with d. record with e
        x = x + 1
    enddo
    go top
    x = 0
    do while .not. eof( )
        if x = 0
            e = seq
        else
            e = e + "*" + seq
        endif
    enddo
enddo

```

```

        skip
        x=x+1
    enddo
    select b
    replace cut with e
    select a
    delete all
    pack
    select b
    skip
enddo
go top
return

. type use 6. prg
*
*report1 generation
*
clear
@5,20 say "*** Minimal cut sets list ***"
@7,10 say "minimal cut set index"
@7,40 say "basic event"
@8,5 say " .....
..... "

use exam1
x=9
do while .not. eof( )
    @x,16 say str(x-8,3)
    @x,35 say trim(cut)
    @skip
enddo
@x,5 say " .....
..... "

eject
*
*report2 generation
*
clear
@5,20 say "*** Minimal path sets list ***"
@7,10 say "minimal path set index"
@7,40 say "basic event"
@8,5 say " .....
..... "

use exam 2
x=9
do while .not. eof( )

```

```

    @x,16 say str(x-8,3)
    @x,35 say trim(cut)
    x=x+1
    skip
enddo
@x,5 say " .....
.....

eject
*
*reprot3 generation
*
clear
select a
@5,10 say "** Structural importance analysis using critical path; vector **"
@7,5 say " .....
.....

x=8
go top
do while .not. eof( )
    @x,17 say "structural importance of"+xn+;
    "="+str(nn,7)+"/(2* *+str(no-1,2)+" )"
    x=x+1
    skip
enddo
@x,5 say " .....
.....

?
return

. type use 7. prg
*
*support program
*
do while .not. eof( )
    if at("g", cut)=0
        skip
        loop
    else
        i=at('g', cut)
        h=substr(cut, i, 4)
        j=len
        select a
        find &h
        if eof( )
            clear
            do use1

```

```

append blank
replace event with h, calcul with top, gate with g
else
  aa=5*(int(len(trim(calcul))/5))*(+4-len(trim(calcul)))
  top=trim(calcul)+space(aa)
  g=gate
endif
select b
*
*event distribution
*
a=at("&kk", top)
if a<>0
  d=recno( 0
  b=*(len(top)+1)/5
  x=0
  ct=cut
  ln=lken
  do while x<b
    c=substr(top, 5*x+1,4)
    if x<>0
      append blank
    endif
    if i=1
      replace cut with c+substr(ct, 5, j-4), len with in
    else
      if i=j-3
        replace cut with substr(ct, 1, i-1)+c, len with in
      else
        replace cut with substr(ct, 1, i-1)+c+substr(ct, i+4, j-i-3),:
        len with in
      endif
    endif
    x=x+1
  enddo
  go d
else
  if i=1
    replace cut with top+substr(cut, 5, j-4), len with len+len(top)-4
  else
    if i=j-3
      replace cut with substr(cut, 1, i-1)+top, len with len+len(top)-4
    else
      replace cut with substr(cut, 1, i-1)+top+substr(cut, i+4, j-i-3),:
      len with len+len(top)-4
    endif
  endif
endif

```