

페트리네트의 T-invariance를 이용한 시스템의 고장진단

정석권* · 정영미** · 유삼상***

*부경대학교 기계공학부, **부경대학교 대학원 메카트로닉스, ***한국해양대학교 기계·정보공학부

Fault Diagnosis Using T-invariance of Petri Net

SEOK-KWON JEONG*, YOUNG-MI JUNG** AND SAM-SANG YOU***

*School of Mechanical Engineering, Pukyong National University, Busan 608-737, Korea

**Mechatronics Engineering, Graduate School, Pukyong National University, Busan 608-737, Korea

***School of Mechanical and Information Engineering, Korea Maritime University, Busan 606-791, Korea

KEY WORDS: Fault diagnosis 고장진단, Petri Net(PN) 페트리네트, T-invariance 트랜지션 불변성, Backward chaining 후방추론, Inter-node search 중간노드 검색, Incidence matrix 접속행렬

ABSTRACT: This paper describes a fault diagnosis method by a T-invariance of Petri Net (PN). First, a complicated fault system with some failure is modeled into a PN graphic expressions. Next, the PN model is analyzed by using the backward chaining of T-invariance to find out causes of the faults. In this step, an inter-node search technique which is suggested in this paper is applied for reducing searching area in the fault system. Also, a novel idea to compose incidence matrices which have different dimension each other in PN model is proposed. As the new knowledges which is discovered newly about faults can be added easily to conventional systems, the diagnosis system will be very flexible. Finally, the proposed method is applied to the automobile fault diagnosis system to confirm the validity of the method.

1. 서 언

고장 발생시 그 원인을 신속, 정확히 규명하는 고장진단 기술은 시스템의 운영에 있어서 생산성 향상을 위한 중요한 요소기술의 하나이다. 특히, 최근 각 시스템들의 규모가 거대화 되고, 그 기능이 다양화됨에 따라 고장은 한층 복잡한 양상을 띠게 되어, 고장진단에는 대상 시스템에 대한 전문적인 지식과 전문가의 숙련된 경험이 동시에 요구된다. 그러나 전문가 부족 및 전문인력의 잦은 직장 이동이 보편화되고 있는 현실을 감안하면, 초보자라도 전문가에 필적하는 고장진단이 가능한 고장진단 시스템의 구축이 필수적이다.

고장진단법은 온라인 상에서 고장을 실시간으로 자동 검출하는 방법과 오프라인 상에서 고장에 관한 지식 데이터베이스와 추론 로직으로 진단을 행하는 2 가지 형태로 크게 구분되어진다. 전자는 시스템이 비교적 간단할 경우의 고장진단이나 복잡한 시스템의 주요 구성 요소에 대한 고장진단에 비교적 효과적이다. 선박 엔진 등과 같이 대규모, 복잡한 시스템이면서 다년간 전문가들의 축적된 경험에 의해 고장 지식 데이터베이스를 쉽게 구축할 수 있는 경우에 있어서는 후자의 적용이 바람직하다.

오프라인 상의 고장진단 시스템은 첫째 적은 메모리 용량으로 짧은 시간 내에 정확한 고장진단이 가능할 것, 둘째 새로운 지식의 추가나 기존 지식의 수정이 용이할 것, 셋째 고장현상에 대한 인과관계가 사용자에게 시각적으로 명료히 전달될 수 있을

것 등이 요구된다. 이에 관련한 연구로서 선형대수 이론에 근거한 페트리네트의 트랜지션 불변성(이하 T불변성)을 이용하여 고장을 진단할 수 있는 후방추론 알고리즘등이 제안되었다(임, 1997). 또한, 이 후방추론을 이용하여 모터(electrical motor) 속도 제어계의 고장진단에 적용한 사례도 보고되고 있다(임, 1998). 이 방법은 고장에 관한 지식을 페트리네트(이하 PN)의 그래프 표현을 이용해 모델링하고 T불변성을 이용한 후방추론을 통해 고장의 원인을 찾아내는 진단법이다. 또한, 이 방법은 고장현상과 그 원인의 논리관계가 복수의 원인은 허용하되 한 결론만을 갖는 비교적 단순한 Horn-Clause 명제만을 대상으로 하고 있어 추론이 용이하다. 그러나, 이 방법은 고장현상과 관련한 모든 원인들을 헤로써 제공하므로 주목하고 있는 고장현상이 특정의 한 원인에 의해서만 발생한 경우에도 해집합 전체의 확인검색을 거쳐야 고장확인이 가능하게 되는 단점을 갖는다. 이 경우, 다수의 고장 원인이 존재하고, 다단계의 복잡한 인과관계를 갖는 고장 시스템에 대한 고장처치에는 엄청난 시간이 소요되게 되어 결국 비경제적인 진단법으로 된다. 뿐만 아니라 이 방법은 고장과 관련한 새로운 지식의 추가나 기존 지식을 변경하고자 할 경우, PN의 모델링을 재 구축해야 하므로 그 과정이 번거롭다는 문제점도 동시에 가지고 있다.

본 논문에서는 이러한 문제점들을 보완하여 다수의 고장 원인과 다단계의 복잡한 논리구조를 갖는 고장진단 시스템에 효율적으로 적용가능한 고장진단 시스템 구축을 목적으로 한다.

이를 위해 본 논문에서는 PN의 중간노드 검색법과 PN의 접속 행렬에 대한 합성법을 제안한다. 중간노드 검색법은 기존의 추론법이 모델링된 PN의 최초, 최후의 노드만에 주목하여 추론을 행하던 것에 반해 중간노드를 적극적으로 검색함으로써 다수의 고장원인 후보 가운데 현재의 고장에 직접적인 영향을 주고 있는 원인만으로 해의 범위를 압축할 수 있는 방법이다. 또한, PN의 접속행렬의 합성법은 모델링된 PN의 접속행렬이 고장 시스템에 대한 고유정보를 갖고 있다는 점에 착안하여 고장에 관한 새로운 지식이 발견될 경우, 이를 접속행렬로 표현하여 기존 시스템의 접속행렬에 쉽게 부가, 삭제할 수 있는 규칙을 정의한 것이다. 이로써 고장진단 시스템은 고장에 관한 새로운 지식의 반영이 용이해짐으로써 유연성을 갖게 된다. 뿐만 아니라 PN의 접속행렬은 간단한 PN의 그래프 생성 프로그램에 의해 고장의 현상과 원인간의 인과관계를 용이하게 파악할 수 있는 시각정보로도 표현가능함을 보인다. 이는 사용자에게 복잡한 고장의 인과관계를 시각정보를 통해 확인할 수 있도록 한다.

제안된 방법은 자동차의 시동상태불량에 대한 고장진단 예를 통해 그 이론적 타당성이 검증된다.

2. PN의 T-invariance를 이용한 후방추론

2.1 PN와 고장 시스템의 모델링

PN의 구조 C는 일반(ordinary) PN에서는 플레이스(place) P와 트랜지션(transition) T의 유한집합 및 그들의 접속관계를 나타내는 입·출력함수 I(input), O(output)의 4 요소로 식(1)과 같이 이루어져 있다.

$$C = \{P, T, I, O\} \tag{1}$$

여기서,

$$P = \{p_1, p_2, p_3, \dots, p_m\}$$

$$T = \{t_1, t_2, t_3, \dots, t_n\}$$

이다. PN는 이와 같이 구조적으로는 집합론적인 방법 또는 후에 기술하는 상태방정식으로 나타내어지고, 그래프적으로는 유한 다중그래프로 Fig. 1과 같이 표현된다.

PN의 그래프 표현은 모델링된 시스템을 시각적으로 이해하기 쉽도록 해준다. 플레이스는 “○”로 나타내고 사건의 상태나 조건을, 트랜지션은 “—”나 “|”로 나타내며 사건의 진행이나 명령 수행을 의미한다. 플레이스와 트랜지션 간에는 사건의 흐름을 나타내는 아크(arc) “→”로써 연결하며, 시스템의 거동을 나타내기 위해 P 내부에 토큰 “●”을 표시하여 특정의 상태나 조건이 충족되어 있음을 정의한다. P 내의 토큰의 분포는 마킹

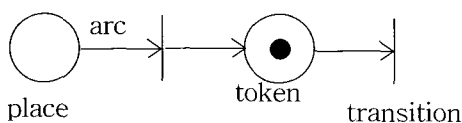


Fig. 1 Graphical expression of a PN

(marking) M으로 나타내며, 구체적으로 P 내의 토큰의 수를 나타내고 시스템 전체의 동적 상황을 모의실험하는데 사용된다. 토큰은 T의 발화규칙에 따라 P간을 이동함으로써 토큰의 재 분포가 이루어지며, M이 초기마킹 M₀에서 새로운 마킹 M_k로 변화됨으로써 상태의 천이를 나타내게 된다. 이러한 상태천이는 식 (2)의 상태방정식을 이용하여 그 거동이 수치적으로 해석된다.

$$M_{k+1} = M_k + A \cdot t_k \quad (k=1, 2, \dots) \tag{2}$$

여기서, k는 시스템에서 임의의 주목하는 시점, M_k는 k상태에서의 마킹을, M_{k+1}은 k+1상태에서의 마킹을 각각 나타낸다. t_k는 k상태에서의 T의 발화벡터로서 발화일 경우 값 “1”을, 비발화일 경우 값 “0”을 각각 가진다. T의 발화규칙은 일반 PN의 발화규칙(Peterson, 1981)을 따른다. A는 식(3)과 같이 정의되는 a_{ij}(∈A)를 그 시스템의 고유치로 갖는 접속행렬 (incidence matrix) 이다.

$$A = O(t_j, p_i) - I(p_i, t_j) \tag{3}$$

여기서, O는 트랜지션 t_j에서 플레이스 p_i로의 아크 수에 해당하는 상태량을 나타내는 출력행렬, I는 플레이스 p_i에서 트랜지션 t_j로의 아크 수에 해당하는 상태량을 나타내는 입력행렬로서, 시스템이 결정되면 고유한 값을 가진다. O, I는 각각 m×n의 차원을 갖는 행렬이고 행은 플레이스에 열은 트랜지션에 각각 대응하고 있다. 고장진단 시스템의 모델링시에는 고장의 원인과 현상을 각각 별개의 플레이스로 표현하고 원인과 현상간의 인과관계를 트랜지션에 부여한다. 특히, 다음 절에서 자세히 설명하는 고장진단을 위한 후방추론시는 트랜지션이 플레이스의 활성을 대신하여 고장현상과 원인의 발생을 의미하게 된다.

식 (3)의 접속행렬은 PN의 구조에 관한 정보를 가지고 있으므로 적절한 프로그래밍에 의해 PN의 그래프 표현을 역으로 생성할 수도 있다. 다음과 같은 임의의 접속행렬 A₁을 갖는 시스템을 가정하면,

$$A_1 = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

이 시스템은 우선, 행과 열의 정의로부터 2개의 P, P={p₁, p₂}와 3개의 T, T={t₁, t₂, t₃}로 이루어져 있음을 알 수 있다. 다음, 접속행렬의 정의로부터 t₁에서 p₁ 및 t₂에서 p₂로의 출력아크 수가 1개이고, p₁에서 t₂ 및 p₂에서 t₃로의 입력아크 수가 각각 1개이므로 A₁의 그래프 표현은 결국 Fig. 2와 같이 얻어진다. 단, 이때 고장시스템 PN의 p_i와 t_j간에는 단일 아크만을 갖는 논리구조임과 동시에 참조 아크(↔)를 갖지 않는 것으로 제한한다.

사용자는 시스템의 이 접속행렬 정보를 통해 시스템의 모델링에 해당하는 PN의 그래프를 얻을 수 있게 되며, 이를 통해 고장 추론 과정에 대한 시각화 정보를 얻을 수 있다.

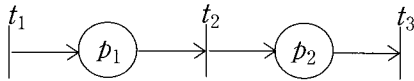


Fig. 2 Generation of PN using incidence matrix

2.2 T-invariance를 이용한 후방추론

본 논문에서는 고장과 원인의 관계가 "A→B", 또는 "A OR B → C"로서 "A→C", "B→C"로 되는 혼-클로즈(Horn-clause, HC)관계로 간주한다. HC에서 A와 B, C는 각각 하나의 명제이며 PN 모델링시 모든 명제는 각각 P로, 각 명제간의 인과관계는 T로 표현된다.

T불변성을 이용한 후방추론은 모델링된 PN에 새로운 T를 최초의 원인 P의 앞단과 최후의 결과 P의 후단에 부가하여 시스템을 해석한다. 즉, 고장이 발생하면 그 고장현상을 나타내는 P(결과P)에 새로운 T(결과T)를 부가하여 현상의 발생을 표현한다. 또한 고장의 원인을 나타내는 P(원인P)에도 새로운 T(원인T)를 부가하여 모든 원인P가 고장현상의 원인임을 표현한다. 이렇게 PN을 변형하면 선형대수 이론에 근거하여 다음에 정의된 T불변 벡터를 구할 수 있다.

[정의 1] Ax=0을 만족하는 0이 아닌 벡터 x를 T불변벡터라 한다.

정의 1에 의해 구해진 T불변벡터는 결과T와 원인T를 반드시 포함한다. 이때 불변벡터에 포함된 원인T가 구하고자 하는 고장 현상에 대한 원인이 된다. T불변성을 이용한 후방추론의 원리를 간단히 설명하기 위해 Fig. 2의 PN모델을 이용한다. 우선, 임의의 고장원인과 그로 인한 고장현상이 존재하는 것으로 하면 이는 두 플레이스 p1, p2에 각각 대응하게 된다. 다음, 후방추론 적용을 위해 p1, p2에 대응하는 트랜지션 t1과 t3를 각각 부가한다. 이때 t2는 고장의 인과관계를 나타낼뿐 별도의 의미를 갖지 않는다. 고장이 없는 초기상태에서는 초기마킹 M0는 0이며 이 상태에서 고장원인이 존재하면 원인P의 p1에 토큰이 생성되고 t2가 발화하게 되어 p1의 토큰이 소멸됨과 동시에 결과P의 p2에 토큰이 생성된다. 마지막으로 p2에 대응하는 결과T인 t3가 발화하게 됨으로써 결과P인 p2의 토큰도 소멸되어 최종 마킹 M3 또한 0으로 된다. 식 (2)의 상태방정식은 이 단계까지를 반영하면 식 (4)와 같이 된다.

$$M_0 + A(t_1 + t_2 + t_3) = M_3 \tag{4}$$

M0와 M3가 0인 상태에서의 상태방정식은 식 (4)에서 Ax=0으로 되고, 식 (3)을 통해 A≠0임이 명확하므로 결국 x≠0인 T불변벡터 x는 결과 t3와 원인 t1을 동시에 포함하게 된다. 이로써 Ax=0을 만족시키는 0이 아닌 벡터 x를 구함으로써 고장원인인 t1의 추론이 가능하게 되며, 이 추론법을 후방추론이라 한다.

Ax=0 (x≠0)을 만족하는 T불변벡터 x는 선형변환을 이용하여 쉽게 구할 수 있다. 여기서 A는 m×n의 행렬, x와 0은

n×1 및 m×1의 열벡터이다. Ax=0을 전치시키면 xT A T=0로 되며, 이때 구하고자 하는 불변벡터 xT는 n×n의 차원을 갖는 단위행렬(unit matrix) I를 이용한다. 이 단위행렬에 대해 한 행의 상수배를 다른 행에 더하는 기본 행연산을 통해 A T와의 곱이 0으로 되는 행을 포함하는 행렬 D를 구함으로써 식 (5)와 같이 얻어진다.

$$D A T = 0 \tag{5}$$

이때, D는 n×n의 차원을 가지며, 행렬 A T와의 곱이 0으로 되는 I의 선형조합 행렬로서 0과 1만을 그 요소로하는 행렬이다. 이 행렬 D의 행 가운데 식 (5)를 만족하는 n개 이하의 행벡터가 구하고자 하는 T불변벡터로서 xT의 해이다. 이는 컴퓨터를 이용한 연산 프로그램에 의해 용이하게 구해질 수 있으며 구해진 해 가운데 대응되는 고장현상을 제외한 고장원인이 최종적으로 구해지게 된다.

3. 중간노드 검색 및 접속행렬의 합성

3.1 중간노드 검색법

고장 p1과 원인의 관계가 Fig. 3과 같이 다단계의 복잡한 인과관계를 갖는 보다 일반적인 형태를 가정하면, 이상에서 논의한 고장 진단법은 비효율적이다. 왜냐하면, 구해진 해들은 현재 주목하고 있는 현상 p1에 관련된 모든 원인 pi (i=3,4,6)를 포함하고 있기 때문이다. 어떤 고장을 일으키는 요인이 다수 개 존재하더라도 이들이 모두 동시에 고장의 원인으로 작용하고 있지 않을 일반적인 경우를 상정해 보면, 이들을 항상 모두 해로 진단하는 추론법은 비경제적이다. 이 경우 고장치료를 위해서는 구해진 모든 해에 대해 확인, 점검을 행해야만 하므로 막대한 시간이 소요될 것이기 때문이다. 따라서, 현재의 고장에 직접적으로 작용하고 있는 원인만으로 해의 범위를 최대한 압축할 수 있는 방안이 필요하다. 본 논문에서는 이에 대한 해결책의 하나로써, 중간노드 검색법을 제안한다. 중간노드는 최초의 원인T와 최후의 결과T를 제외한 트랜지션들을 의미한다. 이 방법은 구하고자 하는 최종 결과T에 대한 최초 원인T를 직접적으로 구하지 않고, 결과T 직전의 중간노드들을 순차적으로 확인해 가는 방법이다.

Fig. 3과 같이 PN로 모델링된 시스템에 중간노드 검색법을 적용하는 과정을 이하에 기술한다. 우선 Fig. 3에서의 단계별 구분은 명제의 표현상 사실 원인 P와 그에 대한 결과 P를 포함하도록 구획되어야 하지만 후방추론법이 트랜지션에 주목하고 있는 관계로 최종 T와 직전의 중간노드 T를 하나의 단계로 구분하고 있다. 그림에서는 최초 원인 p3, p4에 의해 p2의 고장이, p6에 의해 p5의 고장이 발생하고, p2와 p5로 인해 최종적으로 고장 p1이 발생하는 다수의 고장원인을 갖는 다단계 구조로 되어 있다.

우리가 주목하는 고장현상이 p1이면, 임(1998)의 방법에 의한 진단결과는 p3, p4, p6 모두가 고장원인으로 주어지게 된다. 이

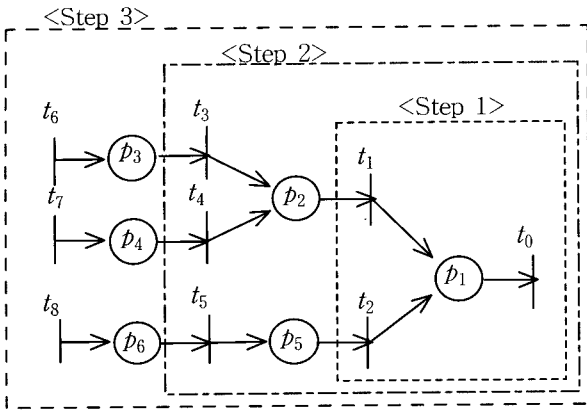


Fig. 3 Concept of the inter-node search

들 3 요소는 p_1 의 고장요인임이 분명하지만, 이들 모두가 항상 동시에 p_1 의 고장요인으로 작용하는 것은 아니다. 만일, p_1 의 고장원인으로 p_6 만이 직접적으로 작용하고 있는 경우라면, 기존의 방법으로 구해진 3 요소 모두를 점검하는 것은 명확히 비효율적이다. 이 경우에 전문가는 적어도 다수의 최초 원인들을 직접적으로 추정하여 이들 모두를 확인하기에 앞서 이 고장현상이 필연적으로 수반하게 될 몇 가지 중간단계의 고장현상에 대한 발생유무를 확인함으로써 해의 범위를 점차 압축해 갈 것이다. 즉, p_2 와 p_5 의 발생유무를 먼저 확인한 연후에 그 결과에 따라 다음 단계로의 고장을 순차적으로 확인해 가는 방법을 택하게 될 것이다.

중간노드 검색법은 이와 같이 최종 결과T인 t_0 로부터 직전의 중간노드 t_1, t_2 를 1단계로 먼저 검색하고 점차적으로 단계2와 단계3을 검색하는 방법이다. 이때, p_2 에 해당하는 현상이 없을 경우, 해에 포함되었던 p_2 와 p_1 과의 인과 관계를 나타내는 t_1 (이하 대응T로 기술)은 제거된다. 그리고, t_2 가 새로운 결과T로 간주되어 이 T와 t_8 에 이르는 시스템에 대한 진단을 계속적으로 행하게 된다. 이로써 t_8 이 최종적인 해로서 얻어지게 되며, 이의 점검만을 통해 고장요인의 해소가 가능하게 되므로 효율적인 고장진단이 수행되게 된다.

한편, 중간노드 검색법을 적용하기 위해서는 단계별 접속행렬이 요구되므로 PN로 모델링된 시스템의 전체 접속행렬인 식(3)의 일부 변경이 필요하게 된다. 단계별 접속행렬은 이미 구해진 전체 접속행렬로부터 다음의 정의에 의해 용이하게 구해진다.

[정의 2] P_x 는 결과T로 간주되는 트랜지션 t_c 에 대하여 $A(p_i, t_c) = -1 (i=1, 2, \dots, m)$ 을 만족하는 $x \times 1$ 차원의 플레이스 집합이다.

[정의 3] T_y 는 P_x 의 플레이스 p_c 에 대하여 $A(p_c, t_j) \neq 0 (j=1, 2, \dots, n)$ 을 만족하는 $1 \times y$ 차원의 트랜지션 집합이다.

[정의 4] 단계별 접속행렬 $A_s (x \times y)$ 는, P_x 를 행, T_y 를 열로 하며 그 원소 $a_{ij} (\in A)$ 를 고유값으로 하는 $x \times y$ 차원의 행렬이다.

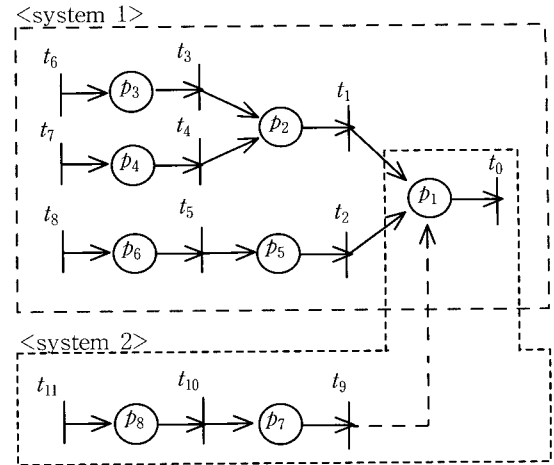


Fig. 4 Composition of a new system

중간노드 검색법을 통한 해의 도출은 정의 4에 의해 생성된 단계별 접속행렬 A_s 를 이용하여 식 (5)를 통해 T불변벡터를 구한다. 실제 시스템 적용시, 중간노드 검색은 MMI(Man Machine Interface)에 의해 사용자에게 “예”, 또는 “아니오”의 형태로 확인을 구하게 하면 된다. 고장상황에 따른 사용자의 응답에 의해 불필요한 노드를 검색대상에서 순차 배제하면서, 최초의 원인T까지 이 과정을 반복하게 된다.

고장의 전체 경로를 나타내는 고장현상에서 최초 원인까지의 과정은 각 단계에서 해가 아닌 T항을 제거한 접속행렬들을 모두 합성함으로써 얻어진다. 이때 차원이 다른 행렬들 간의 합성은 다음 절에서 기술하는 접속행렬의 합성법을 통해 얻어진다.

3.2 접속행렬의 합성

고장진단 시스템은 고장에 대한 추가적인 새로운 지식이 발견되었을 시 이의 반영이 용이해야 한다. 본 논문에서 제안하는 고장진단법은 PN로 모델링된 고장진단 시스템의 구조에 대한 정보를 식 (3)과 같은 접속행렬의 형태로 관리하고 있으므로 이 시스템에 대한 수정은 접속행렬의 변경을 요하게 된다. 따라서, 고장진단 시스템의 유연성은 접속행렬의 수 정용이성에 달려 있다. 차원이 자기 다른 접속행렬간의 합성은 다음의 정의를 이용하여 용이하게 구할 수 있다.

[정의 5] 접속행렬 A_i 는 $P_i = \{p_1, p_2, \dots, p_m\}$,

$T_i = \{t_1, t_2, \dots, t_n\}$ 를 각각 행과 열로 하고, 시스템의 고유치 $a_{ij} (\in A)$ 를 그 요소로 하는 $m \times n$ 차원의 행렬이다.

[정의 6] 차원이 서로 다른 접속행렬 A_i 간의 합 행렬 $A^+ (P^+, T^+)$ 는 OR 연산을 나타내는 기호 “ \oplus ”를 이용하면, $P^+ = P_1 \oplus P_2 \oplus \dots \oplus P_n, T^+ = T_1 \oplus T_2 \oplus \dots \oplus T_n$ 을 각각 행($p \times 1$)과 열($1 \times q$)로 하고 그 요소는 $a_{ij} \in A_i$ 일 경우 a_{ij} 를, $a_{ij} \notin A_i$ 일 경우 0을 그 고유치로 하는 $p \times q$ 차원의 행렬이다.

[정의 7] 행렬 A와 차원이 서로 다른 행렬 $A_i (x \times y)$ 간의 차

행렬 $A^-(P^-, T^-)$ 는 $A^- = A - A_i$ 로서, $P^- = P - P_i$, $T^- = T - T_i$ 를 각각 행과 열로 하고 그 요소는 $a_{ij} \in A_i$ 일 경우 a_{ij} 를, $a_{ij} \notin A_i$ 일 경우 0을 그 고유치로 하는 $(m-x) \times (n-y)$ 차원의 행렬이다. 이때, 삭제 대상의 PN모델의 최소단위는 적어도 하나 이상의 플레이스와 트랜지션을 가져야하며, 기존의 고장현상에 완전 독립적이어야 한다.

이상에서 정의한 행렬의 연산 A^+ , A^- 는 접속행렬의 부가와 삭제시에 각각 적용되어지며 간단한 컴퓨터 연산프로그램을 통해 시스템의 수정이 용이하게 이루어진다.

Fig. 4에 PN의 접속행렬의 합성법을 설명하기 위한 PN모델의 한 예를 보인다.

우선, 시스템 1의 접속행렬 A_1 은 식 (2)에 의해 $P_1 = \{p_i\}$ ($i=1\sim 6$), $T_1 = \{t_j\}$ ($j=0\sim 8$)를 각각 행과 열로 하여 다음과 같이

$$A_1 = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

구해진다.

다음, 추가되는 새로운 시스템 2의 접속행렬 A_2 는 $P_2 = \{p_7, p_8\}$, $T_2 = \{t_9, t_{10}, t_{11}\}$ 를 행과 열로 하여 다음과 같이

$$A_2 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

얻어진다. 이 두 시스템의 합성행렬은 정의 6을 이용하여 각 접속행렬을 합성함으로써 다음과 같이 얻어진다. 이때, $P^+ = \{p_i\}$ ($i=1\sim 8$), $T^+ = \{t_j\}$ ($j=0\sim 11$)가 A^+ 의 행과 열에 각각 대응하고 있다.

$$A^+ = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \end{bmatrix}$$

한편, 전체시스템 A^+ 로부터 시스템 2 부분을 삭제할 경우를 고려한다. 이때의 삭제대상 행렬의 접속행렬은 기존의 고장현상에 완전 독립적인 부분이어야하므로 t_{11} 에서 t_9 에 이르는 PN모델로 된다. 따라서, 삭제대상의 접속행렬은 A_2 가 아닌, A_2' 로 주어진다. 이때, 행과 열은 각각 $P_2' = \{p_7, p_8\}$,

$T_2' = \{t_9, t_{10}, t_{11}\}$ 로서 다음과 같다.

$$A_2' = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}$$

삭제된 후의 수정된 접속행렬 A^- 는 정의 7을 적용하여 구해지며, 행은 $P^- = \{p_1, p_2, p_3, p_4, p_5, p_6\}$, 열은

$T^- = \{t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8\}$ 로 되고 A^- 를 다음과 같이 얻는다.

$$A^- = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

이때 얻어진 행렬은 당연한 결과로서 시스템 2가 합성되기 이전의 A_1 과 동일하게 됨을 알 수 있다.

4. 고장진단의 적용 예

Fig. 5는 A/T 자동차에 있어서 Table 1과 같은 시동시의 고장현상을 PN로 모델링한 것이다.

우선, 해석상의 편의를 위하여 시동이 걸리지 않는 고장현상에 주목한다. 그리고 사용자가 확인한 중간단계의 원인으로서는 스타팅 모터는 작동되지만 스파크가 발생하고 이에 대한 최초원인은 컨넥터의 접촉불량에 의한 것으로 가정한다.

다음, 중간노드 검색법 적용을 위해 단계를 구분하면, Fig. 5는 4단계로 구분되고 단계별 접속행렬은 정의 4에 따라 구해진다.

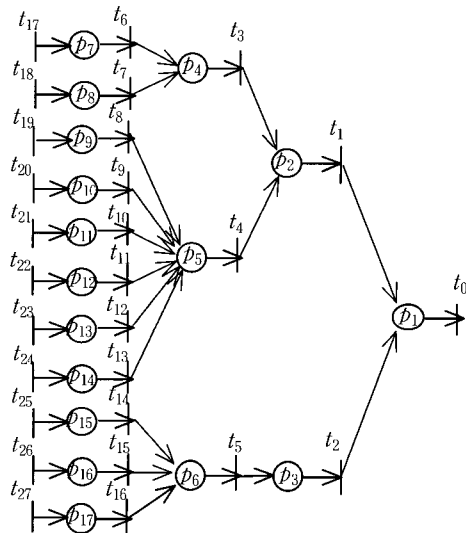


Fig. 5 PN model of the starting trouble in the engine system

Table 1 Definition of each place

place	state	place	state
p_1	abnormal condition of starting	p_{10}	abnormal of power TR
p_2	normal condition of starting motor	p_{11}	abnormal contact of CPS or distributor
p_3	impacting sound	p_{12}	abnormal state of control relay
p_4	spark	p_{13}	abnormal grounding of main wiring
p_5	without spark	p_{14}	abnormal state of ECU
p_6	battery discharge	p_{15}	abnormal running of starting motor
p_7	abnormal contact of connector	p_{16}	electrical leakage
p_8	fall off or leakage of vacuum hose	p_{17}	lack of charge time
p_9	abnormal state of ignition coil		

A=

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
P1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P2	0	-1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P3	0	0	-1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P4	0	0	0	-1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P5	0	0	0	0	-1	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P6	0	0	0	0	0	-1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
P7	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
P8	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
P9	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
P10	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P11	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P12	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P13	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P14	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	1	0	0	0
P15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	1	0	0
P16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	1	0
P17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	1

먼저 Fig. 5의 PN에 대한 전체 접속행렬 A는 식(6)과 같이 구해진다. 고장진단은 1단계로 고장현상인 시동이 걸리지 않음을 표시하는 결과T가 t_0 이므로 정의 4에 따라 결과P인 $P_1 = \{p_1\}$ 을 행으로, $T_1 = \{t_0, t_1, t_2\}$ 을 열로 하는 접속행렬 A_1 을 구하면, $A_1 = [-1 \ 1 \ 1]$ 로 된다. 이때, T불변벡터는 후방추론에 의해 $[1 \ 1 \ 0]$, $[1 \ 0 \ 1]$ 로 구해진다. 이는 결과T인 t_0 에 대해 그 원인T가 t_1, t_2 임을 나타내므로 t_1, t_2 에 대응하는 각 플레이스 p_2, p_3 의 상태를 확인함으로써 중간 단계의 원인T를 구할 수 있다. Table 1에서 p_2 는 스타팅 모터는 작동되고 있는 상태를, p_3 는 부딪치는 소리만 나는 상태를 각각 나타낸다. 스타팅 모터는 작동된다고 가정하였으므로, 확인과정을 통해 이 단계에서의 원인T는 t_1 으로 정해진다.

2단계에서는 1단계에서 확정된 t_1 을 결과T로 간주하여 1단계의 과정과 동일하게 정의 4에 따라 접속행렬 A_2 는 $A_2 = [-1 \ 1 \ 1]$ 로 구해진다. 이때, A_2 는 $P_2 = \{p_2\}$ 를 행

으로, $T_2 = \{t_1, t_3, t_4\}$ 를 열로 가진다. T불변벡터는 $[1 \ 1 \ 0]$, $[1 \ 0 \ 1]$ 로 구해지므로 이 단계에서의 원인T는 t_3 와 t_4 로 된다. 이 원인T에 대응하는 플레이스가 p_4, p_5 이며, 그 의미는 Table 1로부터 스파크의 유·무임을 알 수 있다. 본 논문에서는 스파크 현상이 있는 것으로 가정하였으므로, 이 단계에서의 원인T는 p_4 에 대응하는 t_3 로 정해진다.

이 t_3 를 결과T로 하여 3단계의 검색을 수행하면, A_3 는 $P_3 = \{p_4\}$, $T_3 = \{t_3, t_6, t_7\}$ 를 각각 행과 열로 하여 $A_3 = [-1 \ 1 \ 1]$ 로 얻어진다. 이로부터 T불변벡터 $[1 \ 1 \ 0]$, $[1 \ 0 \ 1]$ 을 얻을 수 있고, 원인T는 t_6, t_7 임을 알 수 있다. 원인T는 대응하는 p_7, p_8 의 상태인 컨넥터의 접촉불량 또는 진공호스의 빠짐이나 누설 상태를 확인함으로써 알아낼 수 있다. 고장의 원인은 컨넥터의 접촉불량으로 가정하였으므로, t_6 이 결과T로 되고 그에 대응하는 플레이스 p_7 이 고장원인으로 최종 확정된다.

전체적인 고장경로는 각 단계별 접속행렬을 정의 5와 정의 6에 따라 각각 합성함으로써 알 수 있다. 마지막 4단계의 접속행렬은 $A_4 = [-1 \ 1]$ 로 구해지므로 고장경로인 전체 접속행렬 A는 A_1 에서 A_4 에 이르는 각 단계별 접속행렬을 합성함으로써 다음과 같이 구해진다.

$$A = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

이때, A_4 의 행과 열은 $P_4 = \{p_7\}$, $T_4 = \{t_6, t_{17}\}$ 이고, A는 $P = \{p_1, p_2, p_4, p_7\}$, $T = \{t_0, t_1, t_3, t_6, t_{17}\}$ 를 각각 행과 열로 한다. 여기서, T불변벡터를 구하면 $[1 \ 1 \ 1 \ 1 \ 1]$ 을 얻게 되며, 이를 통해 지금까지의 전체 추론 과정을 확인할 수 있다. 또한 PN 그래프 생성프로그램을 통해 도식화가 가능하므로 고장과 원인 간의 전 인과관계를 용이하게 확인할 수 있다.

이 적용 예에서 가정한 고장의 경우, 기존의 임(1998)의 방법으로 T불변벡터를 구하면 주목하는 동일의 고장현상에 대해 총 11가지의 원인이 해로서 주어진다. 그러나 중간노드검색법의 적용시에는 이 고장에 직접적으로 작용하는 한가지 원인만으로 해를 제한해 준다. 따라서, 전체 원인을 모두 검색함으로써 야기되는 시간적 낭비와 인력의 소모를 최소화할 수 있다. 또한, 이 방법은 원인의 추론 과정 전체를 PN의 그래프 표현을 통해 시각화도 가능하므로 구축된 고장진단 알고리즘을 사용자가 쉽게 이해할 수 있다는 장점을 갖고 있다.

5. 결 언

본 논문에서는 복잡한 고장 시스템의 원인을 효과적으로 압축함으로써 정확한 고장진단이 가능한 중간노드 검색법을 제안하였다. 또한, 제안된 방법은 기존의 방법과는 달리 고장진단 시스템에 새로운 지식의 추가, 삭제가 용이하도록 유연성을 갖는 방

안도 제안하였다. 이를 위해, 기존의 PN 접속행렬에 각기 차원이 다른 새로운 PN의 접속행렬을 용이하게 합성할 수 있는 규칙을 정의하였다. 뿐만 아니라 제안된 방법은 복잡한 시스템을 접속행렬의 간단한 형태로 정보를 저장한 후 필요시 이를 통해 PN 그래프를 재 생성할 수 있으므로, 사용자의 진단 시스템에 대한 이해를 용이하게 한다. 특히, 모든 연산이 행렬정보로 처리됨으로써 컴퓨터를 통한 고속연산이 가능하고 소용량의 메모리만으로도 진단시스템 구현이 가능한 장점을 가진다. 마지막으로 제안된 방법은 자동차의 시동상태불량에 대한 고장진단에 적용되어 그 유효성이 확인되었다.

제안된 방법은 향후 선박 엔진의 고장진단과 같은 복잡한 시스템에 있어서 초보자도 전문가와 같이 정확하고 신속한 고장처리를 하게 하는 고장 전문가 시스템 구축에 유효하게 적용될 수 있을 것으로 기대된다.

참 고 문 헌

- 임재결(1997). “논리 프로그램의 페트리넷 모형에서 후방추론 방법”, 한국정보과학회 97추계학술대회논문집, 제24권 제2호, pp 23~26.
- 임재결(1998). “페트리넷 후방 추론을 이용한 모터 제어장치 고장 진단 시스템 구현”, 한국정보과학회 춘계학술대회논문집 제25권.
- 정석권, 양주호(1999). “페트리네트의 계층화를 통한 시퀀스 제어 계의 설계(I) -SFC에 근거한 페트리네트의 분할”, 한국해양 공학회지 제13권 제3호, pp 106 ~115.
- 정영미, 정석권, 양주호(2000). “PLC베이스의 시퀀스 제어 시스템 해석을 위한 페트리네트의 분할과 합성”, 한국동력기계 공학회 춘계학술대회논문집, pp 57~63.
- 정영미, 정석권, 유삼상(2001). “T-invariant의 후방추론 기법을 이용한 시스템의 고장진단”, 한국해양공학회 춘계학술대회논문집, pp 32~37.
- J. L Peterson(1981). “Petri Net Theory and The Modeling of Systems”, Prentice-Hall.

2001년 7월 25일 원고 접수

2001년 10월 10일 수정본 채택