韓國海洋工學會誌 第13卷 第3(Ⅱ)號, pp.106~115, 1999. 8

◉ 論 文

# 페트리네트의 계층화를 통한 시퀀스제어계의 설계 ( I )
## - SFC에 근거한 페트리네트의 분할

정 석 권* · 양 주 호**

(99년 4월 26일 접수)

# Design for Sequential Control System Using Petri Nets with Hierarchical Expression( I )
## - Division of Petri Nets Based on SFC

Seok-Kwon Jeong* · Joo-Ho Yang**

**Key Words** : Petri Net(페트리네트), Sequential Control(시퀀스제어), PLC(Programmable Logic Controller), SFC(Sequential Function Chart), Hierarchical Expression (계층화표현)


## Abstract

Modeling a discrete event system such as a sequential control system is difficult compared with a continuous system. Petri nets have been introduced as an analyzing and design tool for the discrete systems. One of the problems in its applications is that the model can not be analyzed easily in the case of large scale or complicated systems because of increase of the number of components of the system. To overcome this problem, some methods for dividing or reducing Petri nets have been suggested. In this paper, an approach for a hierarchical expression of Petri nets based on Sequential Function Chart(SFC) is proposed. A measuring tank system will be described as a typical kind of discrete systems. The system is modeled by sub Petri nets based on SFC in order to analyze and visualize efficiently about the dynamic behaviors of the system. Some numerical simulations using state equations are performed to prove the validity of the proposed method.


* 정회원, 부경대 해양산업개발연구소
** 부경대 제어기계공학과

## 1. 서 론

이산제어인 시퀀스제어는 연속제어인 피드백제어와 더불어 자동화 실현을 위한 중요한 요소기술이다. 최근, 선박을 포함한 해양플랜트의 자동화 기기 가운데 시퀀스제어용 컨트롤러로써 PLC의 사용이 점차 증가하는 추세에 있다.

PLC를 중심으로한 시퀀스제어계 설계시에는, 컨트롤러인 PLC를 포함한 전체 제어계의 거동분석과 설계된 제어로직에 대한 에러검출이 쉬운 설계법이 요구된다. 그러나 이러한 요구를 만족시킬 수 있는 표준적인 시퀀스제어계 설계법은 아직 마련되어 있지 않다. 따라서, 대부분의 설계자들은 시퀀스도와 타임챠트에 근거하여 시스템의 거동을 정성적으로 파악한 연후에 LD(ladder diagram)등의 PLC용 언어로 프로그래밍을 하게 된다. 설계된 제어칙의 설계사양 만족여부에 대한 구체적인 검증은 현장에서의 반복적인 수정작업을 통해 이루어지므로 이로 인해 많은 시간과 노력이 낭비되고 있다. 또한, LD와 같은 PLC 전용 프로그래밍 언어는 해석능력을 갖고 있지 않으므로 설계단계에서 컴퓨터를 이용한 시퀀스 추이 체크나 제어칙의 에러검출이 어려운 실정이다. 뿐만 아니라 국제적으로 규격화된 PLC의 전용 언어가 5종류에 달하고 있어서 이들 언어에 독립적인 제어계 설계법이 절실히 요구되고 있다.

이와 같은 문제점을 해결하기 위한 한 방안으로서 페트리네트(이하 PN으로 표기함)를 이용한 시퀀스제어계의 설계법이 제안되고 있다[1~4]. PN는 PLC의 전용언어에 독립적이면서 시퀀스제어계가 갖는 비동기성, 동시진행성, 병렬성을 표현하기에 용이한 그래픽 표현과 거동분석을 위한 대수방정식 표현이 가능하여 컴퓨터 상에서의 일관된 시퀀스제어계의 설계를 가능하게 한다.

그러나, 이 방법도 설계하고자 하는 대상 시스템이 복잡하거나 대규모일 경우, 시스템의 구성요소 증가가 PN 규모의 증대를 야기해 제어계 전체 흐름에 대한 시각화를 어렵게 하고 대수적 해석 또한 곤란하게 한다.

이를 해결하기 위해, Cai등은 S-invariant에 의한 PN의 분할법을 제안한 바 있다[3]. 이 방법은 시스템의 구성요소 각각을 하나의 모듈로 하여 이를 서브PN로 표현하고 이들의 합성에 의해 전체 PN가 구성될 수 있음을 보였다. 이 방법은 해석하기 쉬운 모델링 구축에는 효과적이지만, 구성기기별 지나친 세분화로 인해 특정의 시퀀스를 분석하는 데는 도리어 불편하다. 또한, 제어계의 시각화를 용이하게 하기 위해서는 대상 시스템을 필요 최소한의 서브PN로 기술함이 바람직하나 이에 대한 방안이 제시되어 있지 않다.

이러한 점에 주목하여, 저자는 SFC에 근거한 PN 계층화 방안을 제안하고 이를 실제 시스템에 적용하여 그 타당성을 검토한 바 있다[4~6]. SFC는 제어의 각 단계별 동작이 제어순서에 따라 기술되므로 제어계의 전체 흐름을 파악하기에는 대단히 용이하다. 특히, SFC는 특정의 시퀀스를 스텝으로 표시하고 그때 수행해야할 동작을 정의함과 동시에 다음 스텝으로 이행하기 위한 조건을 명확히 규정하고 있어서 PN의 그래프 표현방법과 유사하다. 그러나, SFC의 경우 컴퓨터를 통한 제어대상의 거동분석은 그다지 용이하지 않다. 따라서, 본 논문에서는 SFC가 갖는 시각화 용이성과 PN가 갖는 해석의 용이성을 상호 접목시킴으로써 대규모, 복잡한 제어계의 PN표현 및 그 해석이 용이하도록 하는 서브PN 분할법을 제안한다. 또한, 분할된 서브PN를 통해 전체PN 해석결과와 동일한 결과가 도출될 수 있도록 하기 위한 서브PN의 합성칙을 정의한다.

본 논문에서 제안한 PN의 계층화 방안은 서브PN가 SFC의 액션부에 해당되어 특정의 시퀀스를 나타내게된다. 따라서, 필요 최소한의 서브PN로 전체PN가 표현되어 시퀀스제어계의 시각화가 용이하게 될뿐만 아니라, 소규모로 분할된 서브PN의 해석을 통해서도 전체PN의 거동분석이 가능하게 되므로 시각화와 해석의 용이성이 동시에 달성된다.

제안된 방법은 계량탱크 시스템을 통한 수치 시뮬레이션을 통해 그 유효성이 입증된다.

## 2. PN에 의한 시퀀스제어계의 모델링

### 2.1 시퀀스제어계의 특징

Fig. 1은 시퀀스제어계를 하드웨어 구성면에서

본 기본적 개념도이다[1].

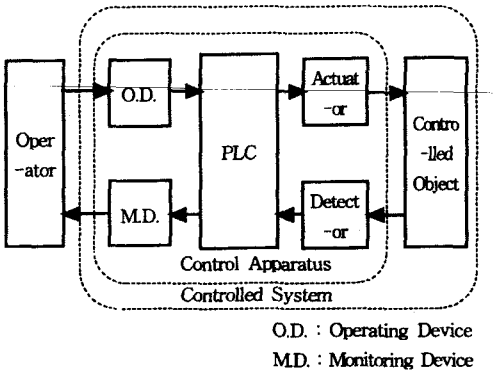



Fig. 1 Block diagram of sequential control system

그림에서 화살표는 정보의 흐름을 나타낸다. 제어대상은 구성 기기의 수에 따라 복수의 요소로 구성되며, 각 요소는 유한 값을 갖는 상태변수로 표현가능하다. 제어장치의 상태량도 이산치로 표현가능하므로 시퀀스제어계는 이산시스템으로 간주될 수 있다. 시퀀스제어계에서는 구성 요소는 대단히 많은 반면, 상태변수가 취할 수 있는 값은 그다지 크지 않은 것이 일반적이다.

어떠한 복잡, 대규모의 시퀀스제어계도 기본적으로는 Fig. 1과 같이 제어용 기기를 중심으로 분류하는 것이 가능하다.

### 2.2 PN에 의한 시퀀스제어계의 표현

PN는 구조적으로는 집합론적인 방법 또는 상태방정식과 같은 수단을 이용하여 기술되며, 그래프적으로는 유향다중그래프를 이용해서 표현되어진다. PN의 그래프 표현은 모델화된 시스템을 시각적으로 이해하기 쉽도록 하며, PN에 토큰(token)을 부여함으로써 시스템 전체의 거동파악이 가능해진다. 또한, 상태방정식을 이용한 방법은 PN의 상태천이의 기술이 가능하게 되어 시퀀스제어계의 거동분석을 가능하게 한다.

PN의 구조 $C$는 일반(ordinary) PN에서는 플레이스 $P$(place)와 트랜지션 $T$(transition)의 유한집합 및 그들의 접속관계를 나타내는 입 · 출력함수 $I$(input), $O$(output)의 4요소로 이루어져 있다[7].

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

여기서,

$$P=\{p_1, p_2, p_3, \cdots, p_m\} \quad m\geq 0$$

$$T=\{t_1, t_2, t_3, \cdots, t_n\} \quad n\geq 0$$

$p_i\in I(t_j)$인 $p_i$를 $t_j$의 입력 플레이스, $p_i\in O(t_j)$인 $p_i$를 $t_j$의 출력 플레이스 라고 한다. 또한, $I(t_j)$에 있어서 $p_i$의 다중도를 $\#(p_i, I(t_j))$, $O(t_j)$에 있어서 $p_i$의 다중도를 $\#(p_i, O(t_j))$로 나타낸다. 다중도를 행렬로 표현해서, $B^-[i,j]=\#(p_i, I(t_j))$를 입력함수행렬, $B^+[i,j]=\#(p_i, O(t_j))$를 출력함수행렬이라고 한다. $B^-$, $B^+$는 각각 $m\times n$의 행렬이고 행은 플레이스에, 열은 트랜지션에 각각 대응하고 있다. 특히, $B=B^+ - B^-$를 접속행렬(incident matrix)이라고 한다.

실제 시퀀스제어계의 모델링시에 $P$는 사건의 상태나 조건에 대응되며 기호로는 "○"로 나타낸다. $T$는 사건의 진행이나 명령 수행을 의미하고 "—","|"로써 표기한다. 그리고 사건의 흐름은 방향성을 표시하는 아크(arc: →, ↓)로써 나타내며, 플레이스와 트랜지션을 결합한다. 한편, PN에서는 시스템의 상태현황을 토큰 "●"으로써 나타내며, 플레이스 내부에 표시하여 상태의 활성(enabled)을 정의한다. 시퀀스제어계에 있어서 상태의 활성은 특정의 상태 또는 조건이 충족되고 있음을 나타낸다. 정보의 흐름을 나타내는 토큰은 시스템의 동적이고 동시발생적인 상황과 활동을 모의실험하는 데에 유용하다. 시스템의 임의의 상태를 나타내는 마킹(marking)은 $M$으로, 마킹의 초기상태는 $M_0$로 나타낸다.

$\mu(p_i)$를 플레이스 $p_i$에 존재하는 토큰의 수로 하면, 토큰의 분포를 나타내는 마킹 벡터는 $M=[\mu(p_1)\ \mu(p_2)\ \cdots\ \mu(p_m)]$과 같이 나타내어진다. 플레이스 내부의 토큰의 분포가 일정조건을 만족하면, 트랜지션이 발화(firing) 가능하게 되어 토큰의 재분포가 이루어 진다. 이는 새로운 상태로 천이함을 의미하며 이로써 시스템의 거동을 나타낼 수 있다. 본 논문에서의 PN 마킹은 다음의 트랜

지션 발화규칙을 따른다.

규칙 1] $t_j \in T$인 $t_j$의 발화는 $p_i \in P$에 대해서 $\mu(p_i) \geq \#(p_i, I(t_j))$ $(i=1,2, \cdots, m)$가 만족되어야 한다.

규칙 2] $t_j$의 발화에 의한 새로운 마킹 $M$은 $p_i \in I(t_j)$인 $p_i$는 $\mu(p_i) - \#(p_i, I(t_j))$, $p_i \in O(t_j)$인 $p_i$는 $\mu(p_i) + \#(p_i, O(t_j))$로 된다.

규칙 3] 트랜지션의 발화는 순간적이므로 다른 두개의 트랜지션이 동시에 발화할 가능성은 0이다.

시퀀스제어계의 기술은 시스템의 상태와 제어동작으로 크게 나누어 생각할 수 있다. 이를 PN로 기술할 경우, 시스템이 갖는 각 상태를 플레이스에, 제어동작을 트랜지션에 대응시켜 모델링할 수 있다. PN그래프 표현으로 모델링된 제어계의 거동은 상태방정식 표현을 이용함으로써 쉽게 해석된다.

상태방정식 표현은 시스템의 상태변수를 플레이스로 하고 임의의 플레이스와 다른 플레이스간의 천이조건 즉 제어동작을 트랜지션으로 하여 (2)식과 같이 표현된다.

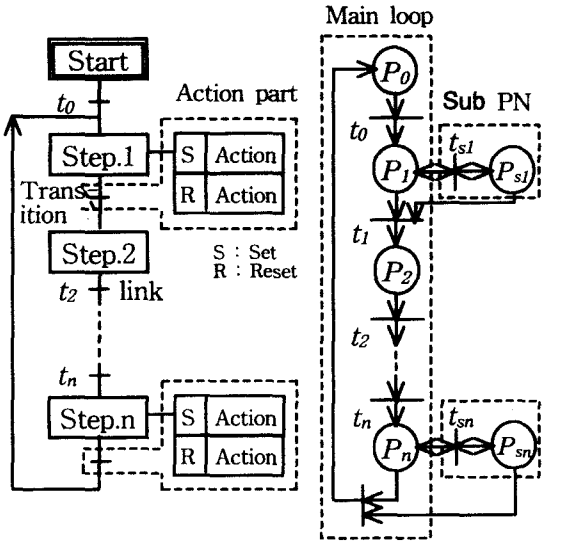
$$p(k+1) = p(k) + B \cdot t(k), \quad k=1,2,3 \ \ldots \quad (2)$$

여기서, $k$및 $k+1$은 이산시스템에서의 초기마킹 시점과 한 단계 진행된 상태의 시점을 각각 나타낸다. $p(k)$는 $p(k) = [\mu(p_1(k)) \ \mu(p_2(k)) \ \cdots \mu(p_m(k))]^T$로서, $k$상태에서의 각 플레이스가 가지고 있는 토큰수를 나타낸다. 이때 상첨자 T는 전치(transpose)를 의미한다. $t(k)$는 $t(k) = [t_1(k) \ t_2(k) \cdots t_n(k)]^T$로 정의되는 $t_j$의 발화벡터로서 발화일 경우 "1" 비 발화일 경우 "0"을 갖는다.

한편, 접속행렬 $B$는 시스템에 의존하는 고유치이며, 트랜지션 발화로 인한 토큰 수의 증·감을 나타낸다. $B^+$의 $j$열은 트랜지션 $t_j$의 발화에 의해 각 플레이스로부터 소멸하는 토큰의 수를, $B^-$의 $j$열은 트랜지션 $t_j$의 발화에 의해 각 플레이스에서 생성되는 토큰 수를 각각 나타낸다.

### 2.3 SFC에 근거한 PN의 분할

Fig. 2(a)는 SFC의 표기를 나타낸다. SFC는 시스템의 기능을 도표에 의해 시각적으로 이해하기 쉽게 나타내는 것이 목적이다. 구성은 스텝(step), 트랜지션 및 링크(link)의 3요소로 이루어져 있다. 스텝은 시퀀스의 한 단계를 의미하고 수행되어야 할 동작은 스텝에 접속된 액션(action)부에 규정한다. 액션은 스텝 오른쪽 가로방향의 사각 블록 안에 쓰고 스텝과 횡선으로 연결된다. 링크는 그래픽 표현에 있어서 스텝간을 접속하고 스텝이 이동하는 경로를 나타내며, 위에서 아래 방향으로 진행한다. 트랜지션은 링크와 직교하는 횡선으로 표시되며, 상위 스텝에서 하위 스텝으로 상태가 이행하는 조건을 나타낸다. 이때의 조건은 스텝간을 이동하기 위한 논리적 조건을 의미한다.




(a) SFC (b) PN

Fig. 2 Division of PN based on SFC

그림에서 아크의 양쪽방향 화살표("↔")는 참조관계("⇄")[3]를 나타내며, 트랜지션 발화에 의해 토큰이 모듈 외부로는 이동하지 않음을 의미한다.

SFC의 표기는 대상의 동작이 제어순서에 따라 기술되어 전체 흐름을 파악하기 쉽다. 또한, SFC의 한 스텝이 특정의 시퀀스에 대응한다. 이점에 착안하여 거대 PN를 분할하면 필요 최소한의 서브PN로 전체PN의 표현이 가능하게 되며 아울러 서브PN가 특정의 시퀀스를 의미하게 되므로 시각

화의 용이성과 해석의 간편성을 동시에 꾀할 수 있다.

SFC에 근거한 PN의 분할을 위해 다음의 규칙을 정의한다.

규칙 1] SFC의 각 스텝 $s_i$ ($i$=0, 1, 2, ···, n) 는 PN의 플레이스 $P_i$에 일대일 대응되어 메인루프를 형성한다. 메인루프는 토큰이 실질적으로 이동하는 루프이다.

규칙 2] SFC의 각 스텝에 접속된 n개의 액션은 스텝에 대응하는 PN의 플레이스 $P_i$의 서브PN를 형성한다.

규칙 3] 서브PN의 트랜지션 $t_{si}$는 $t_{si} \approx \{t_{s1}, t_{s2}, \cdots, t_{sn}\}$ 로서 스텝 $s_i$의 각 액션의 상태를 나타내는 플레이스 $P_{si}$ 상호간 및 메인루프의 $P_i$를 접속한다. 서브PN는 메인루프의 $P_i$가 활성화되면 이에 접속된 $t_{si}$가 발화규칙에 따라 발화를 시작하여 서브PN 내의 일련의 발화시퀀스를 갖게 된다.

규칙 4] 서브PN에서 $P_{si}$는 $P_{si} \approx \{P_{s1}, P_{s2}, \cdots, P_{sn}\}$ 로서 각 액션의 활성상태를 나타내며 $t_{si}$에 의해 접속된다. 서브PN의 최종출력 $P_{si}$는 메인루프의 다음 단계 이행조건인 트랜지션 $t_i$에 접속된다.

서브PN에서 토큰은 서브PN 내부를 순환할 뿐 메인 루프에 있는 플레이스로 이동하지는 않는다. 결과적으로 서브PN는 SFC의 액션을 입력인 발화벡터로 하고 SFC의 트랜지션을 최종출력으로 하여 PN의 메인루프의 트랜지션에 접속된다.

Fig. 2에 SFC에 근거한 서브PN의 대응관계를 보인다.

## 3. 시퀀스제어계의 모델링 응용례

### 3.1 계량탱크 시스템

Fig. 3은 시퀀스제어계의 일례로서 계량탱크 시스템을 나타낸다.

계량탱크는 탱크내의 액면이 하한(L)에 있고(레벨스위치 LS1은 오프, LS2는 온), 밸브1 및 밸브2

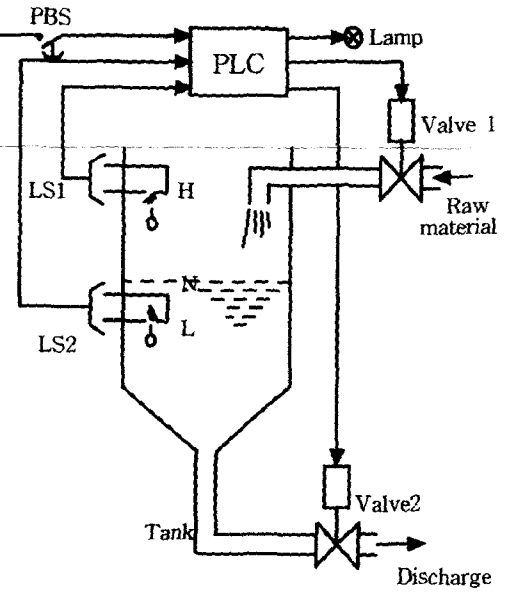



Fig. 3 Measuring tank system

가 모두 닫혀 있는 때를 정지상태로 한다. 이때, 램프1이 점등(on) 되고, 계량 준비 상태임을 나타낸다. 이 상태에서 푸쉬버턴스위치(PBS)를 누르면(on), 밸브1이 오픈(open)상태로 되어 계량상태로 들어가며 램프는 소등된다. 원료가 탱크 내로 주입되기 시작하여 액면이 중간레벨(N)을 거쳐 상한(H)에 이르게 되고 LS1이 온으로 되면, 밸브1은 닫히고, 밸브2는 열려 배출상태로 된다. 액면이 하한에 도달하고 LS2가 온 되면 밸브2가 닫혀 정지상태인 초기상태로 복귀한다. 푸쉬버턴스위치는 자기유지 기능을 갖는 것으로 하고 외부로부터 조작자에 의해 조작된다.

이 시스템은 반드시 분할을 해야만 거동분석이 가능한 시스템은 아니다. 다만, 이 시스템은 본 연구에서 제안한 서브PN의 분할칙의 적용과 그 유용성을 입증하기 위해 도입되었다.

### 3.2 PN에 의한 모델링과 PN의 계층화

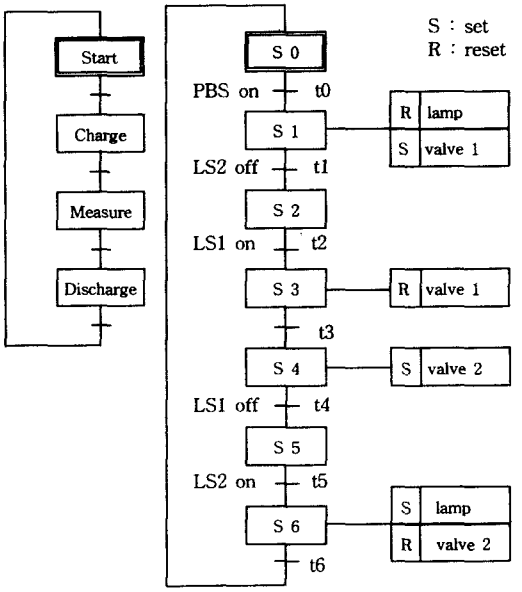
Fig. 3을 Fig. 1의 구성도에 따라 분류하면, Table 1과 같은 구성 기기별 모듈로 우선 나누어진다. 상태 수는 각 모듈이 가질 수 있는 상태치를 파악하여 규정한다.

Table 1 Component of measuring tank system

| Component | Module | State |
|---|---|---|
| controlled object | tank(liquid level) | 3 |
| operator | operator | 2 |
| controller | PLC | 7 |
| actuator | valve1, valve2 | 2 |
| detector | level switch1, 2 | 2 |
| operating device | push button switch | 2 |
| monitoring device | lamp | 2 |

Table 2 The meaning of each step

| step | state / action |
|---|---|
| 0 | waiting (initial) |
| 1 | charge start |
| 2 | measuring |
| 3 | finish of measuring |
| 4 | discharge start |
| 5 | discharging |
| 6 | finish of discharging |




(a) Macro level (b) Concrete level

Fig. 4 SFC of the measuring tank system

이 시스템의 동적 거동은 Fig. 4의 SFC표현을 통해 대략적으로 파악될 수 있다. Fig. 4(b)에 사용된 각 스텝의 의미는 Table 2와 같다.

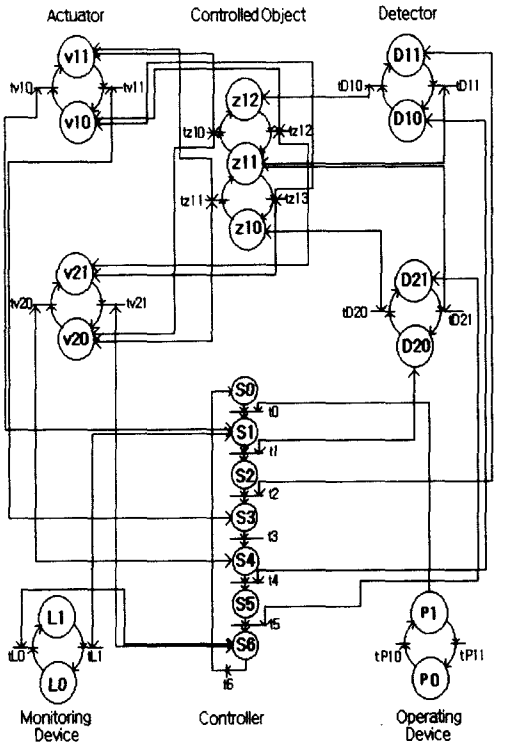



Fig. 5 PN for the measuring tank system

그러나, 이러한 SFC는 제어계의 거동파악을 위한 해석능력을 갖고 있지 아니하므로 시뮬레이션에는 적합하지 못하다. 따라서, PN에 의한 모델링을 수행한다.

PN 그래프는 Fig. 1과 Table 1을 참조하여 각각의 입력요소와 상태요소, 출력요소의 상호 논리식에따라 표현된다. 즉, 각 구성기기를 하나의 모듈로 하여 논리조건에 맞는 PN를 작성하고 이들의 상호 입·출력관계를 고려하여 연결하면 Fig. 5와 같은 계량탱크 시스템의 전체PN가 작성된다.

그림에서 컨트롤러부를 제외한 각 모듈의 플레이스의 의미는 Table 3과 같다.

Table 3 The meaning of each place

| Place | State | Place | State |
|---|---|---|---|
| v11 | valve 1 open | z12 | H level |
| v10 | valve 1 close | z11 | N level |
| v21 | valve 2 open | z10 | L level |
| v20 | valve 2 close | D11 | Ls1 on |
| L1 | lamp 1 on | D10 | Ls1 off |
| L0 | lamp 1 off | D21 | Ls2 on |
| P1 | PBS on | D20 | Ls2 off |
| P0 | PBS off | | |

이때, 컨트롤러인 PLC의 플레이스 S0에서 S6까지는 Table 2의 SFC 상세레벨에 대한 각 스텝내용과 동일하다.

이하에서는 Fig. 5를 대상으로 본 연구에서 제안한 PN의 분할규칙에 따른 계층화를 행한다.

Fig. 6은 Fig. 5를 Fig. 4(b)의 SFC에 근거하고 앞서 정의한 분할규칙에 따라 분할한 계층화PN 표현이다. 그림에서 플레이스 Su1과 트랜지션 tsu1은 한 조로서 서브PN를 나타내고 있다. Su2와 tsu2, Su3와 tsu3, Su4와 tsu4도 각각 서브PN를 나타낸다. 또한 SFC의 각 스텝은 PN표현에서 메인루프를 형성하고 있다.

서브PN에 대응하는 상세한 PN의 일례가 점선 속에 도시되어 있다. 서브PN의 트랜지션 tsu1은 메인루프의 S1의 활성으로 인한 출력으로서 서브PN 내의 트랜지션 tL1과 tv10에 대응되며 서브PN의 최초 입력으로 작용한다. 또한, 서브PN의 플레이스 Su1은 서브PN의 최종 출력 플레이스인 D20에 대응하고 있다.

계층화PN에서는 전체 흐름이 PLC의 입 · 출력에 대응되어 나타나므로 시각화가 보다 용이하고 각 서브PN별 접속행렬과 초기마킹을 쉽게 정의할 수 있어 거동해석도 용이하다.

한편, 규모가 크로 복잡한 시스템일 경우 전체 PN에 대한 접속행렬 $B$의 파악은 용이하지 않다. 본 연구에서는 이를 해결하기 위해 각 모듈별 접속행렬 $B_k$를 구한 후 이를 다음에 정의한 합성칙을 통해 전체PN의 $B$를 구할 수 있음을 보인다. 여기서 모듈이란 Table 1에 정의된 시퀀스제어계의 각 구성기기를 의미하며, 토큰이 순환하는 단위루프에 대응한다.

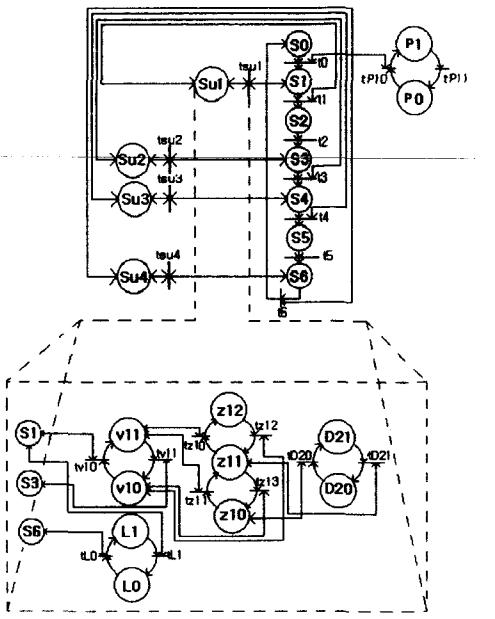



Fig. 6 Hierarchical description of PN

정의 1] $b_{ij}(x\times y)$는 $i$행 $j$열을 초기요소로 하고 크기가 $x\times y$인 모듈 $k$에 대한 접속행렬 $B_k$이다. 이때, $x$는 플레이스, $y$는 트랜지션에 각각 대응한다.

정의 2] $\widetilde{B_{ijk}}$는 $m\times n$의 크기와 모듈 $k$에 대한 행렬 $b_{ij}(x\times y)$를 소행렬로 갖고, $b_{ij}(x\times y)$이외는 모두 0인 행렬이다. 이때 $m$과 $n$은 구하고자 하는 접속행렬 $B$의 플레이스 및 트랜지션에 각각 대응한다.

정의 3] $D=\#\widetilde{B_{ijk}}$의 $D$는 양의 정수로서 동일 모듈 $k$의 중복도이다.

정의 4] $B_{ijk}$는 $m\times n$의 크기를 갖고 모듈 $k$의 중복도를 고려하여 $B_{ijk}=\frac{1}{D}\widetilde{B_{ijk}}$로 주어지는 행렬이다.

이상의 정의를 이용하면, 전체PN의 접속행렬 $B$는 서브PN의 접속행렬들을 이용하여 다음 식과 같이 표현된다.

$$B=\sum_{k=1}^{N}B_{ijk}=\frac{1}{D}\sum_{k=1}^{N}\widetilde{B_{ijk}} \tag{3}$$

여기서, $N$은 $B$에 속한 모듈의 전체 수를 나타낸다.

이 정의에 의해 22×23의 크기를 갖는 전체PN의 접속행렬은 분할에 의해 그 최대크기가 13×13인 행렬로 축소되었다. 또한 이 서브PN의 접속행렬을 이용하여 전체PN의 접속행렬로 합성이 가능하며 따라서 거동해석도 쉽게 될 수 있음을 알 수 있다.

## 4. 수치 시뮬레이션

이 장에서는 Fig. 6에 대해 상태방정식을 이용한 수치 시뮬레이션을 행함으로써 주어진 시퀀스제어계의 설계사양 만족 여부를 확인한다.

Fig. 6의 메인 루프에서의 플레이스와 트랜지션은 각각 다음과 같다.

$$p=\{S_0, S_1, S_2, S_3, S_4, S_5, S_6, P_0, P_1\} \quad (4)$$

$$t=\{t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_{p10}, t_{p11}\} \quad (5)$$

상태벡터의 초기상태 $M_0$는 (6)식, 발화벡터는 (7)식과 같이 각각 주어진다.

$$p(k)=[100000010]^T \quad (6)$$

$$t(k)=[100000000]^T \quad (7)$$

(7)식에서의 $t0$를 1로 설정한 것은 푸쉬버튼스위치 $P_1$을 온(발화) 시킨 것에 대응한다. 이때, 접속행렬 $B$는 입·출력함수 행렬로부터 (8)식과 같이 얻어진다.

$$B=\begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \quad (8)$$

$p(k+1)$은 (6), (7), (8)식을 이용하여 상태방정식 (2)의 해를 구함으로써 (9)식과 같이 얻어진다.

$$p(k+1)=[010000001]^T \quad (9)$$

(9)식은 $P_1$의 입력에 의해 플레이스 $S_0$에서 $S_1$으로 상태가 이행하였음을 보여주고 있다.

다음은 서브PN의 거동에 관해 기술한다. $S_1$이 활성 상태이므로, 플레이스 $S_2$의 활성을 위한 필요충분조건은 트랜지션 $t_1$의 발화이다. 이를 위해서는 서브PN의 플레이스 $Su_1$의 활성이 전제되어야 한다. (9)식에 의한 $S_1$의 활성은 $t_{su1}$을 발화시키고, 이는 상세PN의 $t_{L1}$ 및 $t_{v10}$의 발화에 대응되어 상세PN에 대한 입력인 발화벡터로 작용한다. 상세PN의 플레이스 및 트랜지션은 (10), (11)식과 같고 이에 대한 초기상태와 발화벡터는 (12), (13)식과 같다.

$$p_{s1}=\{L_1, L_0, V_{11}, V_{10}, Z_{12}, Z_{11}, Z_{10}, D_{21}, D_{20}\} \quad (10)$$

$$t_{s1}=\{t_{L0}, t_{L1}, t_{V10}, t_{V11}, t_{Z10}, t_{Z11}, t_{Z12}, t_{Z13}, t_{D20}, t_{D21}\} \quad (11)$$

$$p_{s1}(k)=[100100110]^T \quad (12)$$

$$t_{s1}(k)=[0110000000]^T \quad (13)$$

여기서 하첨자 $s$는 서브PN임을 나타낸다. 서브PN의 접속행렬 $B_{s1}$은 점선 속의 상세PN에 대한 입·출력함수 행렬로부터 (14)식과 같이 얻어진다.

$B_{s1}=$

| | $t_{L0}$ | $t_{L1}$ | $t_{V10}$ | $t_{V11}$ | $t_{Z10}$ | $t_{Z11}$ | $t_{Z12}$ | $t_{Z13}$ | $t_{D20}$ | $t_{D21}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| L1 | 1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| L0 | -1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| V11 | 0 | 0 | 1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 |
| V10 | 0 | 0 | -1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Z12 | 0 | 0 | 0 | 0 | 1 | 0 | -1 | 0 | 0 | 0 |
| Z11 | 0 | 0 | 0 | 0 | -1 | 1 | 1 | -1 | 0 | 0 |
| Z10 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 1 | 0 | 0 |
| D21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -1 |
| D20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 1 |

(14)

(14)식에서는 대응되는 플레이스와 트랜지션을 행과 열에 각각 표시하였다.

$p_{s1}(k+1)$ 은 식 (12), (13), (14)식을 (2)식에 대입하여 (15)식과 같이 구해진다.

$$p_{s1}(k+1)=[0\,1\,1\,0\,0\,0\,1\,1\,0]^T \quad (15)$$

다음 단계에서는 tZ11이 발화하고 (15)식을 초기 마킹으로 하여 $p_{s1}(k+2)$가 다음과 같이 구해진다.

$$p_{s1}(k+2)=[0\,1\,1\,0\,0\,1\,0\,1\,0]^T \quad (16)$$

최종적으로 tZ10와 tD21이 발화, $p_{s1}(k+3)$가 (17)식과 같이 구해진다.

$$p_{s1}(k+3)=[0\,1\,1\,0\,1\,0\,0\,0\,1]^T \quad (17)$$

결국, 상세PN의 최종 출력은 플레이스 D20의 활성으로 귀착되며, 이는 서브PN의 플레이스 Su1의 활성에 대응된다. 결과적으로 메인루프의 t1이 발화하게 되고 상태는 S1으로부터 S2로 이행하게 된다.

이상과 같은 방법으로 트랜지션을 순차적으로 발화시켜 상태방정식 (2)에 대한 수치 시뮬레이션을 통해 계량탱크 시스템에 대한 동적 거동이 설계 사양을 만족함을 확인할 수 있었다. 또한, Fig. 6의 각 서브PN에 대한 상태방정식 표현 및 이를 통한 수치 시뮬레이션 결과는 Fig. 5의 전체 PN의 해석 결과와 일치함이 확인되었다.

## 5. 결 언

본 연구에서는 PN를 이용한 시퀀스제어계의 설계시, 제어계 구성요소의 증가로 인한 해석상의 곤란과 시각화의 어려움을 해결하기 위한 PN의 계층화 방안이 제안되었다. 제안된 방법은 SFC에 주목한 PN 분할법으로서 분할을 위한 규칙을 정의하고 아울러 분할된 PN를 통해 전체PN의 해석이 가능할 수 있도록 하기 위한 분할된 서브PN의 접속행렬에 대한 합성칙도 정의하였다.

제안된 방법은 서브PN가 특정의 시퀀스에 대응하게 되어 해석상의 용이성은 물론 제어계 전체 흐름의 시각화에도 대단히 유효한 수법이다. 이 방법을 계량탱크 시스템에 적용하여 상태방정식 표현을 이용한 수치 시뮬레이션을 행함으로써, 제안된 방법의 유효성을 확인할 수 있었다.

제안된 방법은 복잡하고 거대한 시퀀스제어계의 거동해석시 효과적인 해석수단으로써 시퀀스제어계의 설계에 기여할 수 있을 것으로 기대된다.

## 후 기

본 연구는 부경대학교 해양산업개발연구센터의 연구비 지원(과제번호 97K4-1506-00-02-3)으로 수행된 과제의 일부로서, 이에 관계자 여러분께 감사드립니다.

## 참고문헌

1) 일본전기학회, シーケンス制御工學, 1988
2) 丁國駿, "ペトリネットによるシーケンス制御システムの表現と解析に關する研究", 横浜國立大學博士論文, 1988
3) Cai, Y., Nishii. I. and T. Sekiguchi, "Modeling by Petri Net with Place Invariants for Sequential Control Systems"(in Japanese), T.IEE Japan, Vol. 114-D, No. 7/8, pp.784~ 791, 1994
4) 정석권, "페트리네트에 의한 탈팬장치의 시퀀스제어계 모델", 한국해양공학회지, Vol. 12, No. 4, pp.83~91, 1998
5) 정석권, 양주호, "페트리네트의 구조화를 통한 시퀀스제어계의 모델링", 한국정밀공학회 춘계학술대회 논문집, pp.252~257, 1998
6) 정석권, 양주호, 이동길, "페트리네트의 계층화를 통한 시퀀스제어계의 설계", 한국해양공학회 춘계학술대회논문집, pp.91~98, 1999
7) Peterson, James Lyle, "Petri Net Theory and The Modeling of Systems", Prentice-Hall, 1981
8) Kayama, M. and H. Nagase, "A High Speed

Control Method for Automated Machine Systems Based on Token Transition of Petri Net Theory"(in Japanese), T.IEE Japan, Vol. 109-D, No. 7, pp.463~469, 1989

9) Arai, H., A. Fujimori and T. Hisamura, "Applications of Time Petri Net to Scheduling Problems of Repetitive Process and Their Restoring Strategies in Emergency Stops(in Japanese)", SICE, Japan, Vol. 22, No. 9, pp.35~41, 1986