

# Synchronic Time Ratio를 이용 시간 페트리 네트로 모델링된 시스템의 최적 성능에 필요한 초기 조건 결정

## Initial Requirements to the Optimal Performance of Systems Modeled by Timed Place Petri Nets Using the Synchronic Time Ratio

고 인 선, 최 정 환  
(Inseon Koh and Juang Hwan Choi)

**Abstract** : The initial token value required to the optimal performance of discrete event systems can be decided by Sum of Delay Time and Synchronic Time ratio, which are new synchronic variables in Timed Place Petri Nets. For the system consisting of two Live-and-Bounded circuits(LB-circuits) fused in common Transition-Transition-Path or common Place-Place-Path, we prove that the Synchronic Time Ratio is the initial token ratio between two LB-circuits to optimally perform system functions. These results are generalized and formulated as a theorem. The initial tokens of a specific place can imply shared resources. Using the theorem, we can decide the minimum number of the shared resources to obtain the optimal performance, and minimize the idling time of resources. As an example, an automated assembly system is modeled by Timed Place Petri Net, and the initial tokens to achieve the optimal system performance are identified. All the values are verified by simulation.

**Keywords**: timed place petri nets, synchronic time ratio, optimal performance, LB-circuits

### I. 서론

페트리 네트는 도식적이고 수학적인 모델링 도구로서 병행, 비동기, 분산적인 특성을 지닌 시스템을 모델링 하는데 이용되고 있다. 페트리 네트는 모델링된 시스템의 구조와 동적 변화에 관한 많은 정보를 담고 있으므로, 시스템의 성능을 최적화 시키고, 개량시키는데 유용하게 쓰인다. 페트리 네트는 쓰이는 용도에 따라 여러 종류로 발전해 왔다. Ordinary 페트리 네트, Colored 페트리 네트, Timed 페트리 네트 등이 대표적인 것이다. 이 중 본 논문에서 다루려고 하는 Timed Place Petri Net는 동적 시스템의 성능 분석을 위해, 각 플레이스에 지연 시간이라는 시간 개념을 도입한 것이다.

페트리 네트에서의 동기 개념은 Synchronic Distance라는 이름으로 Petri 박사에 의해 소개되었다[8]. 처음 소개되었을 때는 두 트랜지션 사이의 발화 관계만을 측정하였으나, 그 개념을 일반화한 Weighted Synchronic Distance[3]가 개발되었고, 이를 두 트랜지션 집합간의 발화 관계를 다루는 개념으로 확장, 시스템 특성을 질적인 측면에서 해석하는데 쓰이고 있다. 현재까지, 아래와 같은 4가지의 동기 변수들이 발견되어 Ordinary 페트리 네트에 사용되고 있다. Deviation Bound (DB)는 가장 기본적인 개념이고, 이를 기초로 Fairness Bound (FB) 값이 측정된다. 같은 방법으로 Synchronic Lead (SL) 개념을 기본으로 하여 Synchronic Distance (SD)가 측정된다.

동기 변수들은[11] 이산 사건 시스템에 있어서 사건 발화의 상호 의존도를 정량적으로 측정하는데 주로 사용되고, 시스템의 질적 제어(Qualitative Control)에 응용되고 있다. 예로써, 컴퓨터의 메모리 접근시 충돌 해결[4], 프로세스 제어, 컴퓨터 하드웨어, 소프트웨어 설치[3] 등에 사용되고 있다. 그러나 동기 변수는 시스템의 양적 제어(Quantitative

Control) 부문에는 전혀 사용되고 있지 않다. 그 이유는 동기 변수에 시간 개념이나 확률 개념이 고려되지 않기 때문이다.

선행된 연구에서 Hillion과 Proth[2]는 Marked Graph로 모델링된 시스템에서 Cycle Time[1,9]을 이용하여 전체 시스템의 성능을 결정하는 critical elementary circuit을 찾고, 이 circuit의 Cycle Time에 따라 초기 토큰을 분배하여 시스템을 최적화 시켰다. Hillion과 Proth의 결과는 Ramamoorthy와 Ho[9]가 성능분석을 위해 제안한 Cycle Time을 이용하므로 적용 가능한 시스템의 범위가 Marked Graph로 한정되고, 공유 자원이 포함된 시스템에는 적용될 수 없다. 본 논문에서는 공유 자원이 포함된 시스템에도 적용할 수 있는 이론을 개발하여 적용할 수 있는 시스템의 범위를 넓혔다.

본 논문에서는 동기 개념을 Timed Place Petri Net에 적용할 수 있는 방법을 연구한다. 또한 개발된 동기 변수(Synchronic Time Ratio)를 이용하여, 시스템의 성능을 최적화 하는데 필요한 조건을 구한다. Synchronic Time Ratio에서는 동기 변수에 시간 개념을 도입하였고, 이를 시스템 성능 분석에 이용하는 방법을 예를 들어(자동 조립 시스템) 설명한다. 적용된 시스템의 기본 특성은 Live, Bounded, Reversible하다고 가정하였다. 예에서도 나와 있지만, 시스템 성능의 의미를 일정 시간동안 만들 수 있거나, 조립할 수 있는 제품의 양으로 간주하였다. 로봇과 같은 고가의 공유 자원을 쓸 경우, 자원의 수가 많을수록 시스템의 성능은 증가하나, 시스템의 운용 비용 또한 높아진다. 공유 자원의 공전시간(Idle Time)을 줄이고, 시스템 최적 성능에 필요한 최소의 공유 자원의 양을 찾는 것이 본 논문의 목적이다.

본 논문은 아래와 같이 구성된다. 2장에서는 논문을 이해하는데 필요한 용어를 간단히 설명한다. 3장에서는 Sum of Delay Time(DT)과 Synchronic Time Ratio(STR)를 정의하고, 이를 이용하여 시스템 최적 성능에 필요한 최소한의 자원을 결정하는 정리를 전개하고, 이를 적용한 간단한 예를

접수일자 : 1996. 5. 29., 수정완료 : 1996. 11. 19.

고인선, 최정환 : 홍익대학교 전자공학과

\* 본 연구는 한국과학재단 핵심전문연구 (과제번호: 941-0900-088-2)에 의해 수행되었습니다.

보인다. 4장에서는 자동 조립 시스템을 Timed Place Petri Net로 모델링한다. Sum of Delay Time(DT)과 Synchronic Time Ratio(STR)를 이용하여 시스템 최적성능에 필요한 최소한의 자원 값을 구하고, 모의 실험을 통하여 결과를 검증한다. 5장에서는 결론을 맺고 향후 연구과제를 논의한다.

II. 기본 용어

본 논문은 독자가 페트리 네트를 알고 있다는 가정 하에 쓰여졌다. 이 장에서는 페트리 네트의 기본 용어와 사용된 약어에 대하여 설명한다. Timed Petri Net는 지연 시간이 트랜지션, 혹은 플레이스에 주어졌느냐에 따라 Timed Transition Petri Net(TTPN)와 Timed Place Petri Net(TPPN)로 나눌 수 있으나, 기본적인 성질은 동일하다. 본 연구에서는 Timed Place Petri Net를 사용하였고, 이후 Timed Place Petri Net를 TPPN이라고 표기한다. Timed Place Petri Net(TPPN): TPPN은 아래와 같은 6개의 구성 요소로 이루어진다.

$$TPPN=(P, T, I, O, m, D),$$

- $P=\{p_1, p_2, \dots, p_m\}$ , 플레이스의 유한집합,  $m \geq 0$ .
- $T=\{t_1, t_2, \dots, t_n\}$ , 트랜지션의 유한집합,  $n \geq 0$ .  
 $P \cap T = \emptyset$ .
- $I(p,t):(P \times T \rightarrow N)$ , p에서 t로 연결된 호선의 수를 정의하는 입력 함수.
- $O(p,t):(P \times T \rightarrow N)$ , t에서 p로 연결된 호선의 수를 정의하는 출력 함수.
- $m(p):p \rightarrow N, \forall p \in P$ , p의 토큰 수를 나타낸다. 초기마킹은  $m_0$ 으로 표시한다.
- $D=\{d_1, d_2, \dots, d_m\}$ , 각 플레이스에 주어진 지연 시간의 집합.

Timed Place Petri Net의 발화 규칙: 플레이스에 토큰이 들어오면, 플레이스에 존재하는 토큰 수에 상관없이 바로 지연이 진행되고, 지연 시간이 경과한 토큰은 발화 가능한 상태에서 대기한다.

Incidence 행렬: m개의 플레이스와 n개의 트랜지션으로 구성된 페트리 네트에서, Incidence 행렬 A는,  $A=[a_{ij}]$  ( $i=1,2, \dots, m, j=1,2, \dots, n$ ),  $a_{ij}=a_{ij}^+ - a_{ij}^-$ ,  $a_{ij}^+ = O(p_i, t_j)$ ,  $a_{ij}^- = I(p_i, t_j)$ . Incidence 행렬 A는  $A^+ - A^-$ 의 합으로 표시하기도 한다.  $A^+=[a_{ij}^+]$ ,  $A^-=[a_{ij}^-]$ , ( $i=1, \dots, m, j=1, \dots, n$ ).

P-invariant:  $X^T A = 0$ 을 만족하는 X ( $m \times 1$ ) 벡터이며, 양의 정수를 원소로 갖는다.

Liveness: 일반적으로 Liveness는 5 단계로 나누어 정의된다. 본 논문에서는 단계의 구별 없이 트랜지션이 더 이상 발화할 수 없는 상태를 Deadlock이라 하고, 시스템 내에 Deadlock이 없는 상태를 Live하다고 간주하였다.

Boundedness: 다음의 조건을 만족할 때, 페트리 네트는 Bounded되어 있다고 한다.

$$m(p) \leq k, \forall p \in P$$

$m(p) \in R(m_0)$ 이고 k는 상수이다.  $R(m_0)$ 은 초기마킹( $m_0$ )에서 도달 가능한 모든 마킹의 집합이다. 여기서  $k=1$ 일 때, 페트리 네트는 Safe하다고 한다.

Reversibility: 다음의 조건을 만족할 때, 페트리 네트는 Reversible하다고 한다.

$$m_0 \in R(m), \forall m \in R(m_0).$$

Live-and-Bounded Circuit (LB-circuit): LB-circuit은  $p_1 t_1 p_2 t_2 \dots p_n t_n$ 과 같은 Directed Path이며, 아래의 조건들을

만족한다[5][6].

- 1)  $t_i = t_{i+1}$
- 2)  $r_{1n+1} = 1, s_{1n+1} = 0$
- 3) the circuit is live.

여기서 arc ratio  $r_{in}$ 과 remainder  $s_{in}$ 의 정의는 아래와 같고, 간단한 예를 그림 1에 보였다.

1) arc ratio  $r_{in}$

$$r_{in} = INT(r_{1n-1} \times O(p_{n-1}, t_{n-1}) / I(p_{n-1}, t_n))$$

$$\text{if } I(p_{n-1}, t_n) \neq 0,$$

$$r_{in} = 0 \text{ if } I(p_{n-1}, t_n) = 0.$$

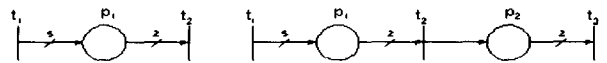
2) remainder  $s_{in}$

$$s_{in} = \sum_{i=1}^{n-1} s_{ii} + MOD(r_{1n-1} \times O(p_{n-1}, t_{n-1}) / I(p_{n-1}, t_n))$$

$$\text{if } r_{1n-1} \times O(p_{n-1}, t_{n-1}) / I(p_{n-1}, t_n) \geq 1,$$

$$s_{in} = \sum_{i=1}^{n-1} s_{ii} + r_{1n-1} \times O(p_{n-1}, t_{n-1})$$

$$\text{if } r_{1n-1} \times O(p_{n-1}, t_{n-1}) / I(p_{n-1}, t_n) < 1.$$



(a) arc ratio r12=remainder s12=1 (b) arc ratio r13=1, remainder s13=1

그림 1. arc ratio 와 remainder.  
Fig. 1. An arc ratio and a remainder.

common Place-Place-Path (PPP): LB-circuit들의 공유 path 양끝이 플레이스로 구성된 directed path.

common Transition-Transition-Path (TTP): LB-circuit들의 공유 path 양끝이 트랜지션으로 구성된 directed path.

이후 common Place-Place-Path 와 common Transition-Transition-Path를 common PPP와 common TTP로 표기한다. 그림 2는 LB-circuit, common PPP, common TTP의 한 예이다. (b)에서 LB1과 LB2가 common PPP로 연결되었고, (c)에서는 LB1과 LB2가 common TTP로 연결되었다. 두 LB-circuit이 서로 역방향의 공유 path를 결합하는 것은 허용되지 않는다.

일반적으로 초기 토큰수는 초기마킹에 존재하는 토큰수의 합이다. 본 논문에서는 초기 토큰은 각 LB-circuit 내의 하나의 플레이스에만 존재한다고 가정하고, 입출력 함수의 차이에 무관하게 LB-circuit간의 토큰 수를 비교하기 위해  $INT(\frac{m_0(p_i)}{I(p_i, t)})$ 를 LB-circuit의 초기 토큰수라고 가정한다.

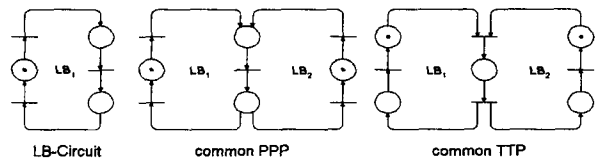


그림 2. LB-Circuit, common PPP, common TTP의 예.

Fig. 2. Examples of an LB-Circuit, a common PPP, and a commonTTP.

여기서 INT는 정수 부분을 취하는 함수이고,  $p_i$ 는 LB circuit 내에 토큰이 존재하는 플레이스이다. 위에서 소수 부분으로 표현되는 토큰들은 LB-circuit의 특성상 LB-circuit의 성능에는 영향을 미치지 않는다.

**III. Synchronic time ratio**

시스템의 성능을 최적화하기 위해서는 최소의 자원으로 최대의 성능을 얻어야 한다. 시스템을 구성하는 여러 종류의 공유 자원에서, 시스템의 성능은 부족한 공유 자원의 수에 의해 주로 결정된다. 시스템의 최적 성능은 주어진 시스템 환경에서 모든 구성 자원들의 공전 시간이 없는 상태라고 간주할 수 있다. 이를 위하여 부족한 공유 자원을 증가시키거나, 잉여 자원은 시스템에서 제거하여야 한다. TPPN에서 LB-circuit으로 모델링된 시스템의 최적 성능은 다음과 같이 정의하였다.

정의 1: TPPN에서 시스템의 최적 성능

시스템을 구성하는 LB-circuit들의 트랜지션 발화율이 같고, 토큰이 각 플레이스에서 부여된 지연시간 이상 머물지 않을때, 이 시스템을 최적성능을 갖는다고 정의한다

이러한 의미의 최적 성능을 내기 위한 시스템을 구성하는 LB-circuit의 초기 토큰 수를 구하기 위해서, 다음과 같은 동기 변수를 정의하였다.

정의 2: Sum of Delay Time(DT)

LB-circuit에서 각 플레이스에 부여된 지연시간의 합.

$$DT(LB_i) \triangleq \sum_{k=1}^n D_k$$

( $D_k$ 는 LB-circuit i의  $p_k$ 에 부여된 지연시간,  $n$ 은 LB-circuit i의 플레이스 수)

정의 3: Synchronic Time Ratio (STR)

두 LB-circuit의 Sum of Delay Time간의 비.

$$STR(LB_1, LB_2) \triangleq \frac{DT(LB_1)}{DT(LB_2)} = \frac{\sum_{k=1}^m D_{1k}}{\sum_{l=1}^m D_{2l}}$$

시스템내의 LB-circuit의 Synchronic Time Ratio간에는 아래와 같은 관계가 성립한다.

- 1)  $STR(LB_i, LB_j) = \frac{1}{STR(LB_j, LB_i)}$
- 2)  $STR(LB_i, LB_k) = STR(LB_i, LB_j) \times STR(LB_j, LB_k)$

Lemma 1: 두 LB-circuit이 common TTP로 연결되어 이루어진 시스템에서 최적 성능에 필요한 두 LB-circuit의 초기 토큰비는 Synchronic Time Ratio이다.

증명:  $m(\tau)$  = 시간  $\tau$ 에서의 마킹,  $A$ =Incidence 행렬,  $u(\tau)$  = 시간  $\tau$ 에서의 발화 벡터라고 놓으면,  $m(\tau) = m(\tau_0) + Au(\tau)$ 이다.  $\Delta m(\tau)$ 를  $\Delta m(\tau) \equiv m(\tau) - m(\tau_0)$ 으로 놓으면,  $\Delta m(\tau) = Au(\tau)$ 가 된다. 여기서  $\Delta\tau = \tau - \tau_0$ 라고 놓자.

$\Delta\tau \neq 0$ 인 경우,  $\frac{\Delta m(\tau)}{\Delta\tau} = A \frac{u(\tau)}{\Delta\tau} = Ai$ 라고 놓으면,  $\frac{\Delta m(\tau)}{\Delta\tau}$ 는  $\Delta\tau$ 동안 변하는 토큰수의 평균 변화를 표시하는 벡터이고,  $i$ 는  $\Delta\tau$  동안 변하는 각 트랜지션의 평균 발화율을 나타내는 벡터이다. 두 LB-circuit이 common TTP로 연결되어 있는 시스템은 consistent하므로, 일정한 주기  $T$ 를 갖는다.  $\tau = T + \tau_0$ 라고 하면,  $\frac{\Delta m(\tau)}{\Delta\tau} = Ai = 0$ 이 성립됨을 알 수 있다. 평균 마킹을  $\bar{m}$ 로 나타내면, 아래와 같은 식이 성립한다.

$$\begin{aligned} \bar{m} &= \frac{m(\tau) + m(\tau_0)}{2} = \frac{m(\tau) + m(\tau_0)}{2} + \frac{m(\tau_0)}{2} - \frac{m(\tau_0)}{2} \\ &= \frac{m(\tau) - m(\tau_0)}{2} + m(\tau_0) = \frac{\Delta m(\tau)}{2} + m(\tau_0) = 0 + m(\tau_0) \end{aligned}$$

여기서,  $\bar{m} = m(\tau_0)$ 임을 알 수 있다.

TPPN에서 평균 발화율 벡터  $i$ 는  $Ai = 0, X^T \bar{m} = X^T D$

( $A^+$ )의 두 식을 동시에 만족할 때, 최대 발화율을 가진다는 사실[10]을 본 증명에 이용하고자 한다. 여기서  $X$ 는 p-invariant 이다. 두 LB-circuit이 common TTP로 연결된 시스템의 Incidence 행렬을 구하면 다음과 같은 형태를 갖는다.

$$A = \left( \begin{array}{ccc|ccc} & A_1 & A_2 & 0 & & \\ \hline & 0 & A_3 & 0 & & \\ \hline -1 & 0 & \dots & 0 & 0 & \dots & 0 & 1 & 0 & & \\ \hline & 0 & & A_4 & & & A_5 & & & & \end{array} \right) \begin{array}{l} n-k \\ k-1 \\ 1 \\ m-k+1 \end{array}$$

n-k                  k                  m-k

$$A_1 = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ & & \ddots & & \\ 0 & \dots & 0 & 1 & -1 \\ 0 & \dots & 0 & 0 & 1 \end{pmatrix} \quad A_2 = \begin{pmatrix} 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & 0 \\ -1 & 0 & \dots & 0 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 \\ & & \ddots & & & \\ 0 & \dots & & 0 & 1 & -1 \end{pmatrix}$$

$$A_4 = \begin{pmatrix} 0 & \dots & 0 & 1 \\ 0 & \dots & 0 & \\ \vdots & & \vdots & \\ 0 & \dots & 0 & \\ -1 & 0 & \dots & 0 \end{pmatrix} \quad A_5 = \begin{pmatrix} -1 & 0 & \dots & 0 \\ 1 & -1 & 0 & \dots & 0 \\ & & \ddots & & \\ 0 & \dots & 0 & 1 & -1 \\ 0 & \dots & 0 & 0 & 1 \end{pmatrix}$$

위에서  $LB_1$ 의 Incidence 행렬은  $n \times n$ ,  $LB_2$ 의 Incidence 행렬은  $m \times m$ 인 차원을 갖고, common TTP는  $k$ 개의 공통 트랜지션과  $k-1$ 개의 공통 플레이스로 이루어져 있다. 위의 Incidence 행렬을  $A_i = 0$ 에 대입하여 평균 발화율 벡터  $i$ 를 구하면 다음과 같이 된다.

$$Ai = \begin{pmatrix} i_1 - i_2 \\ i_2 - i_3 \\ \vdots \\ i_{n-1} - i_n \\ i_n - i_1 \\ i_n - i_{n+1} \\ i_{n+1} - i_{n+2} \\ \vdots \\ i_{n+m-k-1} - i_{n+m-k} \\ i_{n+m-k} - i_{n-k+1} \end{pmatrix} = 0 \text{을 만족하는 } i = \begin{pmatrix} k_1 \\ k_1 \\ \vdots \\ k_1 \end{pmatrix} \quad (1)$$

위 (1)에서 common TTP로 연결된 LB-circuit들의 트랜지션들은 모두 같은 발화율을 가진다는 것을 알 수 있다. 연결된 시스템의 두 개의 p-invariant  $X_1$ 과  $X_2$ 를 구하여, (2)에 표시하였다.  $X_1$ 의 support는  $LB_1$ 을 구성하는 플레이스들로 이루어져 있고,  $X_2$ 의 support는  $LB_2$ 의 구성하는 플레이스들로 이루어져 있다.

$$X_1 = \left( \begin{array}{c} \left( \begin{array}{c} 1 \\ \vdots \\ 1 \end{array} \right) n \\ \left( \begin{array}{c} 0 \\ \vdots \\ 0 \end{array} \right) m-k+1 \end{array} \right), \quad X_2 = \left( \begin{array}{c} \left( \begin{array}{c} 0 \\ \vdots \\ 0 \end{array} \right) n-k \\ \left( \begin{array}{c} 1 \\ \vdots \\ 1 \end{array} \right) k-1 \\ \left( \begin{array}{c} 0 \\ 1 \end{array} \right) 1 \\ \left( \begin{array}{c} 1 \\ \vdots \\ 1 \end{array} \right) m-k+1 \end{array} \right) \quad (2)$$

$Ai = 0$  일 때 평균 마킹  $\bar{m}$ 은 (3)을 만족한다.

$$\begin{aligned} \bar{m} = DA^+i &= \begin{pmatrix} D_1 & 0 & \cdots & 0 \\ 0 & D_2 & 0 & \cdots & 0 \\ & & \ddots & & \\ 0 & \cdots & 0 & D_{n+m-k+1} \end{pmatrix} A^+ \begin{pmatrix} k_1 \\ k_1 \\ \vdots \\ k_1 \end{pmatrix} \\ &= \begin{pmatrix} D_1 k_1 \\ D_2 k_1 \\ \vdots \\ D_{n+m-k+1} k_1 \end{pmatrix} \end{aligned} \quad (3)$$

최대 발화율을 가지기 위한  $LB_1$ 과  $LB_2$ 의 평균 토큰 수를 구하기 위해, (2)의 두 개의 p-invariant  $X_1, X_2$ 를  $X^T \bar{m} = X^T D(A^+i)$ 에 대입한다.

$$\begin{aligned} X_1^T \bar{m} &= X_1^T D(A^+i) = \bar{m}(p_1) + \cdots + \bar{m}(p_n) \\ &= (D_1 + D_2 + \cdots + D_n)k_1 \\ X_2^T \bar{m} &= X_2^T D(A^+i) = \bar{m}(p_{n-k+1}) + \cdots + \bar{m}(p_{n-1}) \\ &\quad + \bar{m}(p_{n+1}) + \cdots + \bar{m}(p_{n+m-k+1}) \\ &= (D_{n-k+1} + \cdots + D_{n-1} \\ &\quad + D_{n+1} + \cdots + D_{n+m-k+1})k_1 \end{aligned}$$

여기서  $DT$ 의 정의에 의해,

$$\begin{aligned} DT(LB_1) &= D_1 + D_2 + \cdots + D_n \\ DT(LB_2) &= D_{n-k+1} + \cdots + D_{n-1} \\ &\quad + D_{n+1} + \cdots + D_{n+m-k+1} \end{aligned}$$

그러므로,

$$\begin{aligned} \frac{(D_1 + D_2 + \cdots + D_n)k_1}{DT(LB_1)} &= k_1, \\ \frac{(D_{n-k+1} + \cdots + D_{n-1} + D_{n+1} + \cdots + D_{n+m-k+1})k_1}{DT(LB_2)} &= k_1 \end{aligned}$$

$\bar{m}(LB_k)$ 는  $LB_k$ 의 평균 토큰수,  $m_0(LB_k)$ 는  $LB_k$ 의 초기 토큰수라 하면,

$$\frac{\bar{m}(LB_1)}{DT(LB_1)} = k_1, \quad \frac{\bar{m}(LB_2)}{DT(LB_2)} = k_1.$$

$$\frac{\bar{m}(LB_1)}{\bar{m}(LB_2)} = \frac{m_0(LB_1)}{m_0(LB_2)} = \frac{DT(LB_1)}{DT(LB_2)} = STR(LB_1, LB_2).$$

따라서, 시스템이 발화율  $k_l$ 을 가지도록 하는 두 LB-circuit의 초기 토큰비는 Synchronic Time Ratio이다. ■

위의 증명의 경우 선행 연구의 critical Cycle Time[9]을 이용하여도 유도할 수 있다.

Lemma 2 : 두 LB-circuit이 common PPP로 연결되어 이루어진 시스템에서 최적 성능에 필요한 두 LB-circuit의 초기 토큰비는 Synchronic Time Ratio이다.

증명 : 앞의 증명에서와 같이 TPPN에서 평균 발화율 벡터  $i$ 는  $Ai=0$ ,  $X^T \bar{m} = X^T D(A^+i)$ 의 두 식을 동시에 만족할 때, 최대 발화율을 가진다는 사실을 이용하여 증명한다. 두 LB-circuit이 common PPP로 연결된 시스템의 Incidence 행렬을 구하면 common TTP와 비슷한 Incidence

행렬을 얻을 수 있다. 지면 관계상 Incidence 행렬은 표시하지 않았다. Incidence 행렬을  $Ai=0$ 에 대입하여 평균 발화율 벡터  $i$ 를 구하면,

$$\begin{aligned} Ai &= \begin{pmatrix} i_1 - i_n \\ i_2 - i_1 \\ \vdots \\ i_{n-k} - i_{n-k-1} \\ i_{n-k+1} - i_{n-k} - i_{n+m-k+1} \\ i_{n-k+2} - i_{n-k+1} \\ \vdots \\ i_{n-1} - i_{n-2} \\ i_n + i_{n+1} - i_{n-1} \\ i_{n+2} - i_{n+1} \\ \vdots \\ i_{n+m-k+1} - i_{n+m-k} \end{pmatrix} = 0 \text{ 을 만족하는} \\ i &= \begin{pmatrix} \begin{pmatrix} k_1 \\ \vdots \\ k_1 \end{pmatrix} n-k \\ \begin{pmatrix} k_1 + k_2 \\ \vdots \\ k_1 + k_2 \end{pmatrix} k-1 \\ \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} 1 \\ \begin{pmatrix} k_2 \\ \vdots \\ k_2 \end{pmatrix} m-k+1 \end{pmatrix} \end{pmatrix} \end{aligned} \quad (4)$$

$Ai=0$  일 때 평균 마킹  $\bar{m}$ 은 (5)를 만족한다.

$$\begin{aligned} \bar{m} = DA^+i &= \begin{pmatrix} D_1 & 0 & \cdots & 0 \\ 0 & D_2 & 0 & \cdots & 0 \\ & & \ddots & & \\ 0 & \cdots & 0 & D_{n+m-k+1} \end{pmatrix} A^+ \begin{pmatrix} k_1 \\ \vdots \\ k_1 \\ k_1 + k_2 \\ \vdots \\ k_1 + k_2 \\ k_1 \\ k_2 \\ \vdots \\ k_2 \end{pmatrix} \\ &= \begin{pmatrix} D_1 k_1 \\ D_2 k_1 \\ \vdots \\ D_{n-k} k_1 \\ D_{n-k+1} (k_1 + k_2) \\ D_{n-k+2} (k_1 + k_2) \\ \vdots \\ D_n (k_1 + k_2) \\ D_{n+1} k_2 \\ \vdots \\ D_{n+m-k} k_2 \end{pmatrix} \end{aligned} \quad (5)$$

연결된 시스템의 p-invariant  $X$ 를 구하면,

$$X = (1, 1, \dots, 1, 1)^T \in R^{(n+m-k) \times 1} \quad (6)$$

최대 발화율을 가지기 위한  $LB_1$ 과  $LB_2$ 의 평균 토큰수를 구하기 위해, (6)의 p-invariant  $X$ 를  $X^T \bar{m} = X^T D(A^+i)$ 에 대입한다.

$$\begin{aligned} X^T \bar{m} &= X^T D(A^+i) \\ &= \bar{m}(p_1) + \cdots + \bar{m}(p_{n+m-k-1}) + \bar{m}(p_{n+m-k}) \\ &= (D_1 + D_2 + \cdots + D_{n-k})k_1 \\ &\quad + (D_{n-k+1} + D_{n-k+2} + \cdots + D_n)(k_1 + k_2) \\ &\quad + (D_{n+1} + D_{n+2} + \cdots + D_{n+m-k})k_2 \end{aligned}$$

$k_1 = k_2$  라 하면,

$$X^T \bar{m} = (D_1 + D_2 + \dots + D_{n-k})k_1 + 2(D_{n-k+1} + D_{n-k+2} + \dots + D_n)k_1 + (D_{n+1} + D_{n+2} + \dots + D_{n+m-k})k_1$$

여기서 정의에 의해,

$$DT(LB_1) = D_1 + D_2 + \dots + D_n$$

$$DT(LB_2) = D_{n-k+1} + D_{n-k+2} + \dots + D_{n-k+m}$$

그러므로,

$$\frac{X^T \bar{m}}{DT(LB_1) + DT(LB_2)} = k_1$$

$$k_1 = \frac{\bar{m}(LB_1)}{DT(LB_1)} = \frac{\bar{m}(LB_2)}{DT(LB_2)}$$

$\bar{m}(LB_k)$ 는  $LB_k$ 의 평균 토큰수,  $m_0(LB_k)$ 는  $LB_k$ 의 초기 토큰수라 하면,

$$\frac{\bar{m}(LB_1)}{\bar{m}(LB_2)} = \frac{m_0(LB_1)}{m_0(LB_2)} = \frac{DT(LB_1)}{DT(LB_2)} = STR(LB_1, LB_2)$$

따라서, 시스템이 발화율  $k_1$ 을 가지도록 하는 두 LB-circuit의 초기 토큰비는 Synchronic Time Ratio이다. ■

위의 증명에서 충돌 현상이 발생할 때 발화율  $k_1$ 을 유지하여 충돌 현상을 해결하는 스케줄러가 있다고 가정하였다. Theorem : 1개의 LB-circuit( $LB_i, 2 \leq i \leq l$ )들이 common PPP로 연결된 시스템에, 최소한 하나 이상의 LB-circuit들이 각각의  $LB_i$ 에 common TTP로 연결되면, 전체 시스템의 최적 성능에 필요한 각 LB-circuit들간의 초기 토큰비는 Synchronic Time Ratio이다.

증명 :  $l$ 개의 LB-circuit( $LB_i, 2 \leq i \leq l$ )들이 common PPP들로 연결된 시스템과  $m(m \geq l)$ 개의 LB Circuit이 최소한 하나 이상 각각의  $LB_i$ 에 common TTP로 연결되면, common PPP들로 연결된 시스템과  $m$ 개의 LB-circuit에 존재하는 각각의 토큰수의 합은 항상 일정하다. common PPP로 연결된 시스템과  $m$ 개의 LB-circuit의 구성 플레이스들은 각각 p-invariant가 되므로,  $1+m$ 개의 p-invariant를 구할 수 있다. 따라서, 시스템의 Incidence 행렬을 구하여 Lemma와 같은 식으로 풀면 다음의 식을 구할 수 있다.

$$\frac{X_1^T \bar{m}}{\sum_{i=1}^l DT(LB_i)} = k_1, \frac{X_2^T \bar{m}}{DT(LB_{l+1})} = k_2, \dots, \frac{X_{1+m}^T \bar{m}}{DT(LB_{l+m})} = k_{1+m}$$

( $k_i$ 은 common PPP들로 연결된 시스템의 발화율,  $k_{l+i}(i=1, 2, \dots, m)$ 는  $LB_i$ 의 발화율)

$k_0 - k_1 = \dots = k_m - k_{m+1} = \dots = k_{0+m} - k_1$  라 가정하면,

$$\frac{X_1^T \bar{m}}{\sum_{i=1}^l DT(LB_i)} = k_0, \frac{X_2^T \bar{m}}{DT(LB_{l+1})} = k_0, \dots, \frac{X_{1+m}^T \bar{m}}{DT(LB_{l+m})} = k_0$$

Lemma 2에서 common PPP들로 연결된 시스템을 구성하는 LB-circuit  $l$ 개의 토큰비는 STR이므로,

$$\frac{\bar{m}(LB_1)}{DT(LB_1)} = k_0, \dots, \frac{\bar{m}(LB_{l+m})}{DT(LB_{l+m})} = k_0$$

$$\frac{m_0(LB_a)}{DT(LB_a)} = \frac{m_0(LB_b)}{DT(LB_b)} \quad (1 \leq a, b \leq (l+m))$$

$$\frac{m_0(LB_a)}{m_0(LB_b)} = \frac{DT(LB_a)}{DT(LB_b)} = STR(LB_a, LB_b)$$

따라서, 발화율  $k_0$ 을 가지도록 하는 각 LB-circuit들간의 초기 토큰비는 Synchronic Time Ratio이다. ■

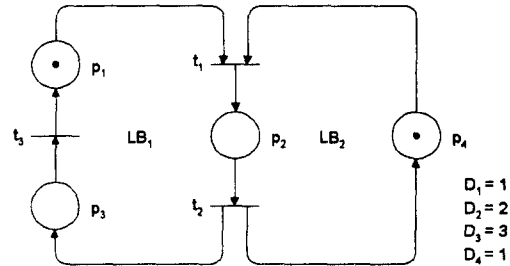


그림 3. Timed place petri net.  
Fig. 3. Timed place petri net.

그림 3과 4에서는 증명한 Lemma와 정리를 예를 들어 설명하였다. 그림 3의 TPPN는 Lemma 1을 적용한 예로서, 두개의 LB-circuit,  $LB_1(p_1t_1p_2t_2p_3)$ 과  $LB_2(p_4t_2p_2)$ 가 common TTP( $t_1p_2t_2$ )로 연결되어 구성되어 있다. 각 플레이스의 지연시간은 그림 안에  $D_i$ 로 표시되어 있다. 이 TPPN는 Live, Bounded, Reversible하다. 초기에는 트랜지션의 발화 흐름이 불안정한 상태이나, 일정한 시간이 지난 뒤 정상 상태가 되면, 발화 시퀀스는 일정한 주기를 가지고 안정된다. 그림 4는 이 시스템의  $m_0=(2 \ 0 \ 0 \ 1)$ 인 경우, 시간의 흐름에 따라 각 플레이스에 토큰의 상태를 나타낸 그래프이다. 가로축은 시간을 나타내고, 세로축은 각 플레이스의 토큰수를 표시한다. 그림 4에 나타나듯이, 6 단위 시간 후부터 시스템은 안정되고, 3 단위 시간의 주기로 발화 시퀀스가 반복된다.

각 LB-circuit의 DT를 구하면,

$$DT(LB_1) = D_1 + D_2 + D_3 = 6, \quad DT(LB_2) = D_2 + D_4 = 3.$$

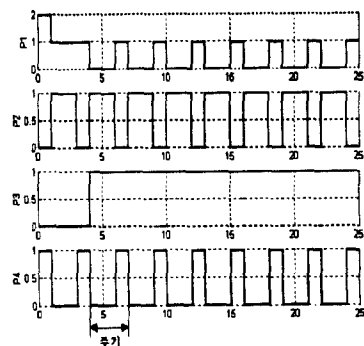


그림 4. TPPN의 마킹 변화.

Fig. 4. A flow of markings of TPPN.

두 LB-circuit의 STR는 다음과 같다.

$$STR(LB_1, LB_2) = \frac{DT(LB_1)}{DT(LB_2)} = 2$$

Lemma 1에 의하여,  $LB_1$ 과  $LB_2$ 의 토큰수가 2:1이 되도록,  $m_0(p_1):m_0(p_4) = 2:1$ 이 되면 이 시스템은 최적 성능을 가지게 된다. 여기서 시스템의 성능이란  $t_1$  혹은  $t_2$ 의 단위 시간당 평균 발화율을 말한다.  $m_0(p_1)=2, m_0(p_4)=1$  일 때  $t_1$ 의

평균 발화율은 0.33/단위시간 이다.  $m_0(p_1)$ 이나  $m_0(p_4)$ 의 어느 한쪽만 증가시킬 경우, 잉여 토큰은 공전하고  $t_1$ 의 평균 발화율은 0.33/단위시간 이상 향상되지 않는다. 여기서 토큰의 공전 시간이란 한 플레이스에서 주어진 지연시간 이상 머물 때의 잉여 시간을 의미한다.

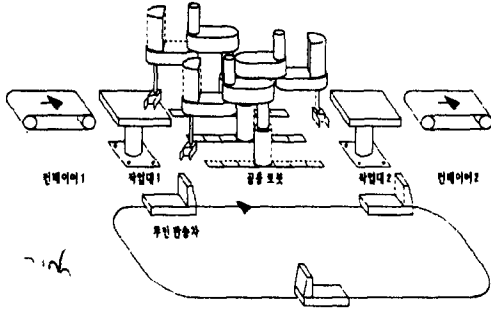


그림 5. 자동 조립 시스템.  
Fig. 5. An automated assembly system.

IV. 자동 조립 시스템에 적용

그림 5는 자동 조립 시스템의 구성을 나타낸다. 시스템은 두대의 작업대와 여러 대의 공용로봇, 두대의 컨베이어와 여러 대의 무인 반송차로 구성되어 있다. 작업물을 컨베이어1로부터 공급받아, 작업대1에서 부품을 1차 조립하고 무인반송차로 옮겨진다. 무인반송차는 작업물을 작업대2로 이동시킨다. 무인반송차로부터 작업물을 공급받아, 작업대2에서 2차 조립된 완제품은 컨베이어2로 옮겨져 출고된다. 작업대1, 2에서 여러 대의 로봇이 동시에 작업이 가능하다고 가정하였다.

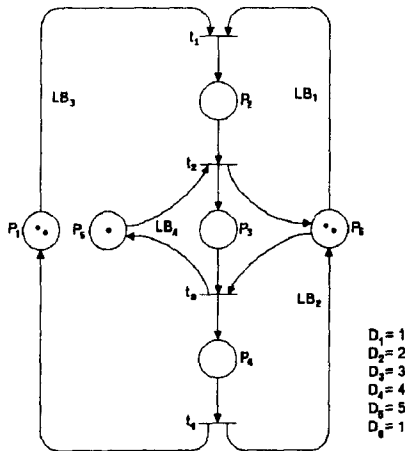


그림 6. 자동 조립 시스템에 대한 TPPN 모델.  
Fig. 6. TPPN model of the automated assembly system.

그림 6은 자동 조립 시스템을 TPPN으로 모델링한 것이다. 일반적으로 플레이스는 시스템 상태를 나타내고, 트랜지션은 프로세스의 시작과 끝을 의미한다. 토큰은 사용 가능한 장비의 수와 시스템 동작여부를 나타낸다. 각 트랜지션, 플레이스, 토큰의 상세한 의미는 다음과 같다.

- t1: 컨베이어1에서 작업물이제, 작업대1에서 작업시작.
- t2: 작업대1에서 작업완료, 무인반송차에 작업물적재.
- t3: 무인반송차에서 작업물이제, 작업대2에서 작업시작.
- t4: 작업대2에서 작업완료, 컨베이어2에 완제품적재.
- p1: 완제품 출고, 새로운 작업물 보급

- (초기 토큰수: 동시에 작업할 수 있는 작업물의 수)
- p2: 작업대1에서 작업
- p3: 작업대1에서 작업대2로 작업물 이동
- p4: 작업대2에서 작업
- p5: 무인반송차를 대기위치로 되돌린다  
(초기 토큰수: 무인반송차의 수)
- p6: 공용 로봇을 대기위치에 놓는다  
(초기 토큰수: 공용 로봇의 수)
- D1=1: 완제품 출고, 새 작업물 보급에 걸리는 시간.
- D2=2: 작업대1에서 작업시간.
- D3=3: 작업대1에서 작업대2로 무인반송차 이동시간
- D4=4: 작업대 2에서 작업시간
- D5=5: 무인반송차가 대기위치로 돌아오는 시간
- D6=1: 공용로봇이 대기위치로 돌아오는 시간

$p_3$ 와  $p_5$ 에 주어진 지연시간의 합은 이 시스템에서 무인반송차가 한 주기를 수행하는데 걸리는 평균적인 시간을 의미한다. 무인반송차 대수의 증가와 감소에 의한 이의 증감은 없다고 가정하였다.

그림 6의 시스템은 4개의 LB-circuit ( $LB_1=p_6t_1p_2t_2$ ,  $LB_2=p_6t_3p_4t_4$ ,  $LB_3=p_1t_1p_2t_2p_3p_4t_4$ ,  $LB_4=p_6t_2p_3t_3$ )으로 구성된다. 각 LB-circuit의 DT를 구하면,

$$DT(LB_1)=D_6+D_2=3 \quad DT(LB_2)=D_6+D_4=5$$

$$DT(LB_3)=D_1+D_2+D_3+D_4=10 \quad DT(LB_4)=D_5+D_3=8.$$

시스템은  $LB_1$ 과  $LB_2$ 가 각각 common TTP ( $t_1p_2t_2$ ,  $t_3p_4t_4$ )를 통하여  $LB_3$ 과 연결되었다.  $LB_1$ 과  $LB_2$ 는 common PPP ( $p_6$ )를 통하여 연결되었고,  $LB_4$ 는  $LB_3$ 과 common TTP ( $t_2p_3t_3$ )를 통하여 연결되었다. 여기서  $LB_1$ 과  $LB_2$ 가  $p_6$ 를 통하여 연결된 네트워크( $LB_{12}$ )는 Shared Resource Net로, LB-circuit이 common PPP로 연결된 플레이스에 초기 토큰이 존재하므로,  $p_6$ 에 존재하는 초기 토큰수는  $LB_1$ 과  $LB_2$ 의 초기 토큰수의 합으로 생각할 수 있다.

위에서 구한 DT를 이용하여, 각 LB-circuit간의 STR을 구하면,

$$STR(LB_1, LB_2) = \frac{DT(LB_1)}{DT(LB_2)} = \frac{3}{5}$$

$$STR(LB_2, LB_3) = \frac{DT(LB_2)}{DT(LB_3)} = \frac{5}{10}$$

$$STR(LB_3, LB_1) = \frac{DT(LB_3)}{DT(LB_1)} = \frac{10}{3}$$

$$STR(LB_3, LB_4) = \frac{DT(LB_3)}{DT(LB_4)} = \frac{10}{8}$$

Lemma 2에 의해, 두 개의 LB-circuit,  $LB_1$ 과  $LB_2$ 로 구성된 시스템은 토큰비를 3:5로 해줄 때 최적 성능을 가진다. 여기에  $LB_3$ 을 연결한 시스템에서 두 LB circuit,  $LB_2$ 와  $LB_3$ 의 토큰비가 1:2 일 때 두 LB-circuit,  $LB_2$ 와  $LB_3$ 으로 구성된 시스템이 최적 성능을 가지므로, 정리에 의해 세 개의 LB-circuit,  $LB_1$ ,  $LB_2$ ,  $LB_3$ 으로 구성된 시스템은 토큰비가 3:5:10 일 때 최적 성능을 가진다. 여기에  $LB_4$ 를 연결한 시스템에서 두 LB-circuit,  $LB_3$ 과  $LB_4$ 의 토큰비를 5:4로 해줘야 두 LB-circuit,  $LB_3$ 과  $LB_4$ 로 구성된 시스템이 최적 성능을 가지므로, 네 개의 LB-circuit,  $LB_1$ ,  $LB_2$ ,  $LB_3$ ,  $LB_4$ 로 구성된 시스템은 토큰비를 3:5:10:8로 해줄 때 최적 성능을 가진다.  $LB_{12}$ 에 속한  $p_6$ 의 초기 토큰수와  $LB_3$ 에 속한  $p_1$ 의 초기 토큰수의 비가 4:5,  $LB_3$ 에 속한  $p_1$ 의 초기 토큰수와  $LB_4$ 에 속한  $p_5$ 의 초기 토큰수의 비가 5:4가 되도록 하여,  $m_0(p_1):m_0(p_5):m_0(p_6)=5:4:4$ 이면 시스템은 최적 성능을 가지

게 된다.

0:0000/500024	1:1000/410023	2:1000/320022	3:1000/230021	4:1000/140020
5:0100/131011	6:1100/032001	7:0000/032001	8:0110/031110	9:0000/031110
10:0000/031110	11:0000/031110	12:0001/131011	13:1100/032001	14:0010/031110
0:0000/500034	1:1000/410033	2:1000/320032	3:1000/230031	4:1000/140030
5:0100/131021	6:1100/032011	7:0100/032002	8:0010/032211	9:0010/032220
10:0000/032220	11:0000/032220	12:0001/121121	13:1101/122012	14:1100/032211
15:0000/032211	16:0110/031120	17:0000/032220	18:0001/121121	19:1100/032211
20:0011/121121	21:1100/032211	22:0010/032220	23:0000/032220	24:0001/121121
0:0000/500044	1:1000/410043	2:1000/320042	3:1000/230041	4:1000/140040
5:0100/131031	6:1100/032021	7:0100/032012	8:0110/121121	9:0010/032221
10:0010/113300	11:0000/011330	12:0001/111231	13:1101/112122	14:1011/121131
15:1100/032212	16:0110/012221	17:0000/012221	18:0011/111231	19:1100/012221
20:0011/111231	21:1100/012221	22:0010/111231	23:1100/012221	24:0011/111231
0:0000/500054	1:1000/410053	2:1000/320052	3:1000/230051	4:1000/140050
5:0100/131041	6:1100/032031	7:0100/032022	8:0110/013122	9:0110/032222
10:0010/032331	11:0010/031140	12:0001/101341	13:1001/111241	14:1001/121141
15:1101/122032	16:1110/032231	17:0100/013122	18:0110/032222	19:0010/032331
20:0011/101341	21:1000/011340	22:0001/111241	23:1101/112132	24:1011/121141
25:1100/032231	26:0110/012331	27:0100/032222	28:0011/102232	29:1010/011340
30:0011/111241	31:1100/012231	32:0011/111241	33:1101/112132	34:1010/021340
35:0100/012231	36:0111/102232	37:1000/012231	38:0011/111241	39:1100/012231
40:0011/111241	41:1100/012231	42:0011/111241	43:1100/012231	44:0011/111241

그림 7.  $m_0(p_5)$  변화에 따른 발화 시퀀스.  
Fig. 7. The firing sequences made by  $m_0(p_5)$ .

그림 7, 8은  $m_0(p_5)$ ,  $m_0(p_6)$ 의 변화에 의하여 생긴 발화한 트랜지션과 각 플레이스의 토큰수를 나타내는 시뮬레이션 결과이다. 시뮬레이션 결과 데이터의 포맷의 예는 다음과 같다.

- ①3 : ②1001 / ③500054  
 ①: 경과 시간, 예: 3 단위 시간 경과.  
 ②: 발화한 트랜지션, 예: 1번, 4번 트랜지션이 발화.  
 ③: 각 플레이스의 토큰수,  
 예:  $m(p_1)=5$ ,  $m(p_5)=5$ ,  $m(p_6)=4$ .

그림 7은  $m_0(p_1)=5$ ,  $m_0(p_6)=4$ 로 고정하고,  $m_0(p_5)$ 을 변화시켰을 때, 시스템 성능과 토큰의 공전시간이 변하는 것을 보여주고, 줄친 부분에서 주기를 알 수 있다. 그림 7의 세 번째 그림에서 보면, 17 단위시간까지 트랜지션 발화와 플레이스 토큰수가 불규칙적이다가 18 단위시간에서 19 단위시간의 트랜지션 발화와 플레이스 토큰수가 20 단위시간부터 계속 반복되는 것을 알 수 있다. 이때, 18 단위시간부터 19 단위시간까지가 주기가 된다. 평균 공전시간은 그림 7의 경우 시뮬레이션 결과를 이용하여, 주기가 나타난 18 단위시간 이후에  $p_5$ 에 들어온 토큰이 몇 단위시간 후에 나가는가를 따져 보면, 세 번째 그림까지는 지연시간인 5 단위시간 후에 바로 나가는 것을 알 수 있고, 네 번째 그림의 경우 7 단위시간후에 나가는 것을 알 수 있다. 따라서 공전시간은 7 단위시간에서 지연시간을 뺀 2 단위시간이다. 18 단위시간 이후에  $p_5$ 에 들어온 모든 토큰이 2 단위시간의 공전시간을 가지므로 평균 공전시간은 2 단위시간이 된다. 각 초기 조건에 해당하는 시스템 성능과  $p_5$ 에 존재하는 토큰의 평균 공전시간은 다음과 같다. 여기서 시스템 성능은 생산제품수 / 단위시간 으로 간주하였다. 이것은  $t_4$ 의 발화결과 동일하다.

- $m_0(5\ 0\ 0\ 0\ 2\ 4)$ : 시스템 성능 =  $\frac{1}{6}$ , 평균 공전시간 = 0  
 $m_0(5\ 0\ 0\ 0\ 3\ 4)$ : 시스템 성능 =  $\frac{1}{3}$ , 평균 공전시간 = 0  
 $m_0(5\ 0\ 0\ 0\ 4\ 4)$ : 시스템 성능 =  $\frac{1}{2}$ , 평균 공전시간 = 0  
 $m_0(5\ 0\ 0\ 0\ 5\ 4)$ : 시스템 성능 =  $\frac{1}{2}$ , 평균 공전시간 = 2  
 시뮬레이션 결과로부터  $m_0(p_1)=5$ ,  $m_0(p_6)=4$ 인 경우,  $m_0$

( $p_5$ )=4일 때, 이 시스템의 성능이 최대로 되고,  $p_5$ 의 토큰이 공전하는 시간(무인반송차의 공전 시간)이 0이 됨을 알 수 있다.  $p_5$ 의 토큰수를 Synchronic Time Ratio 이상으로 증가시키면 시스템의 성능은 변화가 없으나, 공전시간이 증가함을 시뮬레이션 결과는 보여준다.

0:0000/500042	1:1000/410041	2:1000/320040	3:0000/320040	4:0000/320040
5:0100/131031	6:1100/212021	7:1000/122020	8:0100/113011	9:1100/014001
10:0010/013110	11:0000/013110	12:0000/013110	13:0000/013110	14:0001/130111
0:0000/500043	1:1000/410042	2:1000/320041	3:1000/230040	4:0000/230040
5:0100/131031	6:1100/122021	7:1100/032011	8:0110/012220	9:0010/012220
10:0000/012220	11:0000/012220	12:0001/112121	13:1101/113012	14:1010/022120
15:0100/013111	16:0010/012220	17:0000/012220	18:0001/112121	19:1100/013111
20:0011/112121	21:1100/013111	22:0010/012220	23:0000/012220	24:0001/112121
0:0000/500044	1:1000/410043	2:1000/320042	3:1000/230041	4:1000/140040
5:0100/131031	6:1100/032021	7:0100/032012	8:0110/013112	9:0010/012221
10:0010/011330	11:0000/011330	12:0001/111231	13:1101/112122	14:1011/121131
15:1100/022121	16:0110/012221	17:0000/012221	18:0011/111231	19:1100/012221
20:0011/111231	21:1100/012221	22:0010/111231	23:1100/012221	24:0011/111231
0:0000/500045	1:1000/410044	2:1000/320043	3:1000/230042	4:1000/140041
5:1100/041031	6:0100/032022	7:0100/032013	8:0110/013113	9:0010/012222
10:0010/113311	11:0010/104040	12:0001/110341	13:1101/111232	14:1001/121132
15:1101/122032	16:1110/032212	17:0100/013113	18:0010/012222	19:0010/011331
20:0011/110341	21:1100/011331	22:0001/111232	23:1101/112132	24:1011/121132
25:1100/022122	26:0110/012222	27:0000/012222	28:0011/111232	29:1100/011331
30:0011/111232	31:1100/012222	32:0011/111232	33:1101/112132	34:1010/021231
35:0100/012222	36:0011/111232	37:1000/012222	38:0011/111232	39:1100/012222
40:0011/111232	41:1100/012222	42:0011/111232	43:1100/012222	44:0011/111232

그림 8.  $m_0(p_6)$  변화에 따른 발화 시퀀스.  
Fig. 8. The firing sequences made by  $m_0(p_6)$ .

그림 8은  $m_0(p_1)=5$ ,  $m_0(p_6)=4$ 로 고정하고,  $m_0(p_5)$ 을 변화시켰을 때의 발화 시퀀스이다. 이때의 시스템 성능과 토큰의 공전시간은 다음과 같다.

- $m_0(5\ 0\ 0\ 0\ 4\ 2)$ : 시스템 성능 =  $\frac{1}{6}$ , 평균 공전시간 = 0  
 $m_0(5\ 0\ 0\ 0\ 4\ 3)$ : 시스템 성능 =  $\frac{1}{3}$ , 평균 공전시간 = 0  
 $m_0(5\ 0\ 0\ 0\ 4\ 4)$ : 시스템 성능 =  $\frac{1}{2}$ , 평균 공전시간 = 0  
 $m_0(5\ 0\ 0\ 0\ 4\ 5)$ : 시스템 성능 =  $\frac{1}{2}$ , 평균 공전시간 = 1

시뮬레이션 결과에도 나타나듯이, 시스템의 성능은 각 플레이스의 초기토큰값이  $m_0=(5\ 0\ 0\ 0\ 4\ 4)$ 일 때 최적임을 확인할 수 있다. 이 결과는 정리의 결과와 일치한다. 플레이스의 초기 토큰이 STR로 주어졌을 때, 토큰의 공전 시간은 0이고, 최적 성능을 가진다. 어떤 플레이스가 STR 이하의 토큰을 가질 경우, 시스템 성능은 해당하는 토큰수가 최적비에 도달할 때까지 토큰수에 비례하여 증가하고, 어떤 플레이스가 최적비 이상의 토큰을 가질 경우, 시스템은 최적 성능을 가지나, 토큰의 공전 시간은 잉여 토큰수에 비례하여 증가함을 알 수 있다.

0:0000/500044	1:1000/410043	2:1000/320042	3:1000/230041	4:1000/140040
5:0100/131031	6:1100/032021	7:0100/032012	8:0110/013112	9:0010/012221
10:0010/011330	11:0000/011330	12:0001/111231	13:1101/112122	14:1011/121131
15:1100/022121	16:0110/012221	17:0000/012221	18:0011/111231	19:1100/012221
0:0000/100008	1:1000/910087	2:1000/820086	3:1000/730085	4:1000/640084
5:1100/541074	6:1100/442084	7:1100/343054	8:1110/243153	9:1110/143252
10:1110/043351	11:0110/033451	12:0111/123452	13:1111/123452	14:1111/123452
0:0000/15001212	1:1000/141001211	2:1000/132001210	3:1000/123001209	4:1000/114001208
5:1100/10401118	6:1100/0420108	7:1100/043008	8:1110/743197	9:1110/643296
10:1110/543395	11:1110/443494	12:1111/443494	13:1111/443494	14:1111/443494

그림 9.  $m_0(p_1)$ ,  $m_0(p_5)$ ,  $m_0(p_6)$  변화에 대한 발화 순서.  
Fig. 9. The firing sequence made by  $m_0(p_1)$ ,  $m_0(p_5)$ ,  $m_0(p_6)$ .

그림 9는 시스템의 최대 성능(발화율=1)을 얻기 위해, STR을 유지하면서 모든 플레이스의 초기 토큰수를 증가시킨 경우의, 시스템 성능 변화를 보인 것이다. 각 초기 조건에 해당하는 시스템 성능은 다음과 같다.

$$m_0(5\ 0\ 0\ 0\ 4\ 4): \text{시스템 성능} = \frac{1}{2}$$

$$m_0(10\ 0\ 0\ 0\ 8\ 8): \text{시스템 성능} = 1$$

$$m_0(15\ 0\ 0\ 0\ 12\ 12): \text{시스템 성능} = 1$$

시스템 성능은 플레이스의 초기 토큰수가 각 플레이스가 속해 있는 LB-circuit의 Sum of Delay Time에 도달할 때까지, 초기 토큰수에 비례하여 증가함을 알 수 있다.  $LB_1$ 의 토큰수가 DT인 3,  $LB_2$ 의 토큰수가 DT인 5,  $LB_3$ 의 토큰수가 DT인 10,  $LB_4$ 의 토큰수가 DT인 8일 때, 즉, 초기마킹  $m_0=(10\ 0\ 0\ 0\ 8\ 8)$ 이 될 때, 시스템 성능은 1이 된다.

**IV. 결론**

현재까지의 연구결과들은 동기 개념을 패트리 넷으로 모델링된 시스템의 질적 분석에만 사용하였으나, 본 논문에서 Timed Place Petri Net에 Synchronic Time Ratio라는 이름으로, 동기 개념을 성능 분석에 이용하는 방법을 소개하였다. Synchronic Time Ratio를 이용하여 LB-circuit들이 common TTP와 common PPP로 연결되어 있는, Bottom-up 방식으로 디자인 된 Timed Place Petri Net 시스템의 성능을 최적으로 유지하는 각 LB-circuit의 초기 토큰수를 구하였다. 정리를 적용한 예로서, Synchronic Time Ratio를 자동 조립 시스템을 모델링한 Timed Place Petri Net에 적용하여 최적 성능에 필요한 최소한의 공유 자원의 개수를 구하고, 이를 모의 실험을 통하여 검증하였다.

본 연구 결과의 문제점은 Synchronic Time Ratio가 시스템의 최적 성능에 필요한 초기 토큰수를 알아 낼 수 있지만, 모의 실험을 통하지 않고는 시스템의 성능을 양적으로 정확하게 표현 할 수는 없다는 것이다. 이 문제는 Synchronic Time Ratio 이외의 동기 변수들을 설정함으로써 풀 수 있다고 생각한다. 또한, 본 논문의 결과는 Bottom-up 방식으로 디자인되지 않는 시스템에 적용 할 때는 그 시스템을 구성하는 LB-circuit들을 찾는 알고리즘이 필요하다. 현재 이들에 관한 연구를 진행하고 있다.

**참고문헌**

[1] F. Commoner, A. W. Holt, S. Even and A. Pnueli,

“Marked directed graphs,” *J. Comput. Sci.*, vol. 5, no. 5, pp. 511-523, 1971.  
 [2] H. Hillion and J. M. Proth, “Performance evaluation of job shop systems using timed event-graphs,” *IEEE Trans. Automatic Control*, vol. 34, no. 1, pp. 3-9, Jan., 1989.  
 [3] U. Goltz and Y. Chong Yi, “Synchronic structure,” *LNCS: Advanced in Petri Nets*, pp. 233-252, 1985.  
 [4] W. E. Kluge and K. Lautenbach, “The orderly resolution of memory access conflicts among competing channel processes,” *IEEE Trans. Computer*, vol. C-31, no. 3, pp. 194-207, 1982.  
 [5] I. Koh and F. DiCesare, “Modular transformation methods for generalized petri nets and their applications in automated manufacturing systems,” *IEEE Transactions on Systems, Man, Cybernetics*, vol. 21, no. 6, pp. 1512-1522, Nov., 1991.  
 [6] I. Koh and F. DiCesare, “Synthesis methods using deviation bounds in automated manufacturing systems,” *IEEE Conference on Robotics and Automation*, pp. 1025-1030, 1992.  
 [7] S. Laflit, J. M. Proth and X. Xie, “Optimization of invariant criteria for event graphs,” *IEEE Trans. Automatic Control*, May, 1992.  
 [8] C. A. Petri, “Introduction to general net theory,” *LNCS: Net Theory and Application*, pp. 1-19, 1979.  
 [9] C. V. Ramamoorthy and G. S. Ho, “Performance evaluation of asynchronous concurrent systems using Petri nets,” *IEEE Trans. Software Eng.*, vol. SE-6, no. 5, pp. 440-449, Sept., 1980.  
 [10] J. Sifakis, “Performance evaluation of systems using nets,” *LNCS*, pp. 307-319.  
 [11] M. Silva, “Toward a synchrony theory for P/T nets,” *Eds. Concurrency and Nets, Special volume in the series “Advances in Petri nets,”* New York: Springer-Verlag, pp. 435-460, 1987.



**고 인 선**  
 1979년 서울대학교 전자공학과 졸업 (B.S.). 1987년 Marquette University, 졸업(M.S.). 1991년 Rensselaer Polytechnic Institute, Dept. of ECSE 졸업 (Ph.D.). 1981년~1985년 대우전자 근무. 1991년~1992년 대우전자 근무.

1992년~현재 홍익대학교 전자공학과 조교수. 1996년~현재 시스템자동화 연구센터 소장. 관심분야는 이산 사건 시스템 제어, 공장 자동화, 지능 제어, CIM, Petri Net.



**최 정 환**  
 1972년 3월 16일생. 1995년 홍익대학교 전자공학과 졸업. 1995년 3월~현재 동대학원 전자공학과 석사과정 재학중. 주 관심분야는 이산 사건 시스템 제어, 공장 자동화, Petri Net.