

시뮬레이션 기법을 이용한 페트리 넷 모델 성능 평가

강성귀, 장석호, 남부희
강원대학교 제어계측공학과

The Performance Evaluation of Petri Net Model Using Simulation Method

S.G. Kang, S.H. Chang, B.H. Nam
Dept. of Control and Instrumentation Engineering

Abstract

We applied the simulation method using Petri Nets to a FMS model. Generally ordinary Petri Nets would be short of describing a real FMS operations. Hence we adopted the extended Petri-Nets(EPN) and timed places in order to have performance evaluation. Our simulator use data base modeled of Petri Nets in simulation. We can enhance the overall production rate of the system with the obtained results over a number of simulations.

I. 서론

현대산업의 다품종 소량생산 생산체제를 위해 FMS구축 기술이 필히 요구된다. FMS는 제조, 가공, 조립의 부시스템과 이들이 공유하는 자동 저장/출고 시스템 및 CAD 시스템, 그리고 매개체 역할을 하는 반송 시스템, 이러한 모든 부시스템들을 관리 제어해 주는 컴퓨터 시스템도 포함한다. 위와 같이 복잡한 생산 시설을 최적으로 구축하기 위해서는 실제 설치 이전에 충분한 사전 검토를 하여 최소의 효과로 최적의 결과를 얻을 수 있어야 한다. 일반적으로 FMS에 관련된 문제를 크게 설계문제와 운영문제로 나누고 있다.

FMS는 DEDES(Discrete Event Dynamic System)의 한 예로 구분할 수 있으며, DEDES의 동작은 이산, 비동기 사건의 상호관계에 의하여 결정되고, 상태전이는 오직 이산순간에 이산 사건에 의해서만 유발된다. 반면 CVDS(Continuous Variable Dynamic System)의 경우에는 적용 가능한 많은 시스템 제어이론의 발전이 있었다. 그러나 DEDES의 경우에는 좋은 해석 도구가 CVDS에 비해 빈약하다.

DEDS 모델을 논리적, 대수적, 그리고 성능에 관련된 모델로 분류할 수 있다[1]. 논리적 모델은 관점에 따라서 논리적, 시간개념이 포함된 논리적, 그리고 추체적(stochastic)단계의 모델로 다시 세분화할 수 있다. 이러한 FMS모델의 다양성은 FMS에 대한 관점과 시사의 정도 차이에 기인하며, 이러한 기준에 따라 FMS모델의 분류도 달라지게 된다. 우리는 이 모델중 논리를 기초로 한 페트리 넷을 사용하여 FMS모델링 및 성능평가를 하였다. PN(Petri Nets) 모델은 망이론(net

theory)에 기초를 둔 모델링 방법으로서, Kamath와 Viswanadham은 이 방법의 장점과 CPN(Colored Petri Nets)모델, 그리고 timed PN 모델에 관하여 서술하였다. PN모델은 DEDES를 위한 도식적 모델로서 FMS의 모델링과 분석에 가장 유용한 도구중의 하나이며, 최근에는 PN을 기초로 셀 재이기를 개발하여 사용하고 있다.

PN방법의 모델링 구조를 살펴보면, PN 모델은 프로세스 중심의 구조를 갖고 있기 때문에, 프로세스를 분해하여 각각의 동작들을 구해낼 수 있고, 각 동작에 필요한 자원의 상태나 조건이 프로세스 흐름내에 도식적으로 표현된다. 따라서, 복잡한 시스템도 한눈에 파악되기 쉽고, 계층적 모델링을 가능하게 해 준다.

본 논문은 확장 페트리 넷(Extended Petri Nets)를 사용하여 선정된 FMS에 대해 모델링하고 이 정보를 이용하여 컴퓨터 시뮬레이션을 하여 그 결과를 고찰하고자 한다.

II. FMS 모델링

A-1. 페트리 넷

페트리 넷은 5개의 원소로 이루어진 방향성 그래프이다.

$$\begin{aligned} N &= (P, T, I, O, M_0) \\ P &= \{p_1, p_2, \dots, p_n\} \\ T &= \{t_1, t_2, \dots, t_n\} \\ I &: (P \times T) \rightarrow N \\ O &: (T \times P) \rightarrow M \end{aligned}$$

여기서, P는 유한한 프레이스의 집합으로 원으로 표시하며 조건이나 상태(자원의 사용가능여부나 프로세스 상태 등)를 나타낸다. T는 막대로 표시되며 사건이나 제어시스템의 명령(프로세스의 동작 등)을 나타낸다. m과 n은 각각 프레이스와 트랜지션의 갯수이다. I는 플레이스에서 트랜지션으로의 방향성 아크를 정의하는 입력함수이다. O는 트랜지션에서 프레이스로 향하는 방향성 아크를 정의하는 출력함수이다. N과 M은 임의의 정수이다. $M_0 = (m_1, m_2, \dots, m_n)$ 는 초기상태에서 각

플레이스에 존재하는 토큰의 상황으로, m_i 는 i 번째 플레이스에 있는 토큰의 개수를 의미한다. 플레이스 내의 검은 점(token)은 플레이스가 조건을 만족함을 나타내며, 페트리 넷에서 토큰의 표시(marking)는 시스템의 상태(status)를 나타낸다. 페트리 넷에서 제어의 흐름을 나타내는 것이 토큰인데, 어떤 트랜지션의 모든 입력 플레이스에 토큰이 있는 경우가 트랜지션을 활성화 상태(enabled)에 있다고 말한다. 활성화가 된 출력 트랜지션을 가진 플레이스는 다음 상태로 전이할 수 있고, 이 변화는 트랜지션의 점화(fire)로 일어날 수 있다.

A-2. 확장된 페트리 넷(Extended Petri Nets)

FMS를 모의실험하기 위해서는 목적 대상 시스템을 되도록 정확히 묘사해줘야 한다. 페트리 넷으로는 이러한 의미에서 다소 부족한 면이 있다. 좀 더 상세히 상황을 묘사해 주고 시스템의 제어 개념이 삼입된 모델링 도구가 필요하다. 또한 성능평가를 위해서는 시간 개념이 반드시 포함되어야 한다. 일반적인 페트리 넷만으로는 실제 시스템을 정확하게 표현하기 어렵다. 이러한 단점을 보완하기 위해 발표된 연구 결과가 확장된 페트리 넷이다[4]. 다수의 연구 결과에서 제안한 다양한 기능의 기호가 있다. 우리는 그 중 다음의 몇 가지 기능을 일반 페트리 넷에 추가하여 모델링을 하기로 한다.

플레이스는 그 종류에 따라 다음의 네 가지를 사용하기로 한다. 먼저 생성 플레이스(source place)는 시스템 내로의 새로운 자원, 정보, 또는 부품이 도입됨을 나타낸다.(그림 1.a) 소멸 플레이스(sink place)는 토큰의 종료를 나타내며(그림 1.b), 이 플레이스에 이온 토큰은 소멸한다. 생성 플레이스와 소멸 플레이스는 토큰의 보존성(conservation)을 보장하며 넷내의 토큰의 수가 제한없이 증가하지 못하게 한다. 상태 플레이스(status place)는 일반적인 페트리 넷에서 사용하는 것과 동일하며 시스템의 각 상황을 나타낸다(그림 1. c). 본 논문에서는 시간의 개념을 플레이스(timed place)에 포함하였다. 상태 플레이스와 기호는 동일하게 표현하였지만 해당 데이터 베이스에 시간영역의 유무로 분별할 수 있다. 플레이스의 마지막으로 스위치 플레이스(그림 1. e)는 분기의 역할을 담당한다. 예를 들어 작업의 종료여부 등의 상황 판단을 모델링할 수 있다. 스위치 플레이스는 conflict를 해결하기 위하여 외부정보를 사용하여 해당 평가식의 조건을 검사하고 2개 이상의 작은 원의 리스트(list)중 하나를 택하여 토큰의 길(path)로 선정한다.

가능한 여러 가지 상황에서 특별한 경우에 우선권(priority)을 부여하거나 특수한 사건이 발생할 때의 어떤 작업의 시작이나 금지를 모델링 하기 위해서 금지 아크를 도입한다(그림 1.f). 그림 1.d의 P1에 토큰이 있으면 트랜지션 t는 점화 불능(비활성화) 상태가 된다. 이 경우 토큰의 흐름은 없다.

FMS 모델링을 하면 간단한 생산 공정의 경우에도 많은 상태가 존재하여 모델링한 결과를 일괄적으로 확인하기가 어렵다. 서브넷(subnet)를 사용하면 전체 시스템 모델을 파악하기 용이하며 제어흐름 또한 간략히 표현되어 비전문가도 쉽게 접근할 수 있다(그림 1. d). 작은 원은 포함된 복수개의 플레이스중 마지막 플레이스이고 이곳에 토큰이 진입할 때 해당 서브넷은 동작 수행을 끝낸 것이다.

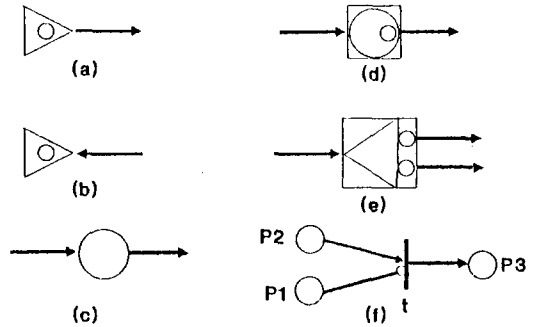


그림 1 확장 페트리 넷 (a)source place, (b)sink place, (c)status place, (d)subnet, (e)switch place, (f)inhibitor

B. 모델링 기법

Valavanis[4]는 페트리 넷을 이용하여 모델링하는 것을 체계적으로 정리하였다. 그에 의하면 확장 페트리 넷(EPN)을 조성하는 3단계를 다음과 같이 나누었다.

1단계: 시스템 분해(decomposition)

2단계: 독립적인 자원(resource) 동작 사이클의 모델링(component net)

3단계: 개별요소 넷(component net)들의 합성

1단계에서는 전체 시스템을 기능별로 분해하여 각 자원별로 부시스템을 만든다. 부시스템은 자기 독립적으로 수행될 수 있다. 자원의 예로는 로봇, 머시닝 스테이션, 치구(fixture), 작업대(workplace) 등이 있다. 이 단계에서는 부시스템 간에 상호작용은 무시한다. 각 자원의 동작 순서도를 정의하는 개념적인 도해를 마련한다. 부시스템중에는 주기능 부시스템이 있다. 여러 부시스템 가운데 전체 시스템의 주된 목표가 되는 동작을 하는 것이 주기능 부시스템이 될 수 있다. 시스템 분해가 이루어지면 각 부시스템간의 다음의 상호관계를 개략적으로 고려해야 한다. 첫째는 자원 할당 문제이다. 제한된 자원을 다수의 동작에서 요구하므로 복수개의 부시스템에서 한 자원이 다중으로 포함될 수 있다. 두 번째로는 부품이나 정보(토큰으로 표시됨)의 이동상황이다. A라는 부시스템을 빠져나온 부품은 B라는 부시스템으로 옮겨지는 것과 같은 상황이다. 마지막으로 동기화 문제이다. 부시스템들간의 선행관계를 나타내든지 동기화를 행할 때에 고려해야 한다.

2단계는 부시스템에 대한 완전한 동작을 EPN으로 나타내는 작업이다. 이 부분에서 개별요소 넷을 모델링할 때 페트리 넷의 liveness, boundedness, consistency, 그리고 error-freeness한 성질을 소유하기 위해서는 다음의 구조적인 제약조건을 지키면서 개별요소 넷을 구성해야 한다. 개별요소 넷에는 방향성 길(directed path)이 존재해야 한다(strongly connected). 개별요소 넷안에는 loop가 없어야 한다. 임의의 트랜지션을 들어가는 아크의 개수와 떠나는 아크의 개수는 일치해야 한다. 본 논문에서는 정보의 참조를 나타내는 아크만을 이 규칙의 예외로 하였다. 그리고 남은 제약 조건은 임의의 플레이스를 기준으로 하나의 입력 트랜지션과 하나의 출력 트랜지션이 연결되어야 하는 것이다. 이상의 조건들은 원하는 성능 기준을 보장하기 위하여 필수적이다.

3단계에서는 각 부시스템에 대한 EPN(개별요소 네트워크)를 합성하여 전체 시스템을 구성한다. 각 개별요소 네트워크를 상호관계를 충분히 고려하여 전체 동작 목표에 부합되도록 구성해야 한다.

C. 모델링의 예

C-1. 전체 구성도

다음은 임의의 한 FMS를 선정하여 모델링의 예를 보기로 한다. 그림 2는 FMS의 한 예이다. FMS의 동작이 시작되면 ASRS(Automatic Storage/Retrieval System)는 조립해야 할 부품을 순서대로 내 놓는다. 무인반송차(AGV)는 ASRS로부터 받은 부품을 입력 컨베이어(Input conveyor)로 옮겨 놓는다. 컨베이어는 정지와 동작(move forward)의 두 상태만이 존재한다. 광센서(L₁)는 컨베이어(C₁)에 의하여 이동되는 부품을 감지하고 이것을 신호로 하여 컨베이어는 정지한다. 로봇(R₁)은 부품을 집어(grasp)서 버퍼로 저장한다. 부품을 순서대로 입력된다고 가정했기 때문에 버퍼내의 부품 유/무를 알리는 광센서(L₂)에 신호가 감지되면 로봇(R₂)은 저장된 부품을 FCFS(First Come First Served)의 규칙에 따라 치구(fixture)로 옮겨 조립을 완료할 때까지 작업을 한다. 조립작업 후 로봇은 출력 컨베이어(C₂)로 부품을 하역하고 이동되는 완성품이 광센서(L₃)에 감지되면 무인 반송차는 자동 적재창고에 적재하여 모든 작업이 끝난다.

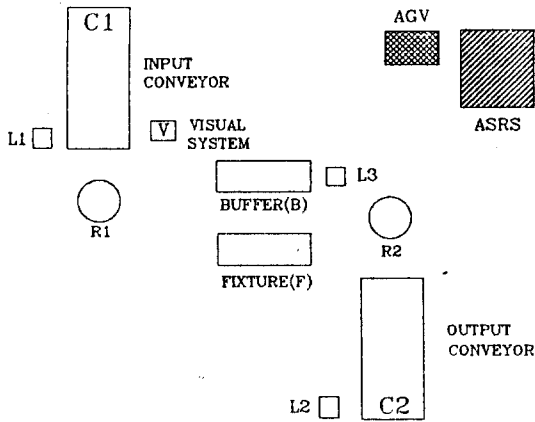


그림 2. 전체 구성도

C-2. 개별 요소 모델링

FMS 공정에 대한 routing이 확정되면 대상 시스템을 자원별로 나누어 개별 요소 모델링을 해야 한다. 토큰의 종류는 본 논문에서는 3개를 사용하였다. <p>는 부품의 이동상황이다. <s>는 일반적인 페트리 넷트에서 사용하는 자원의 이용상태 등을 나타내는 토큰이다. <i>는 정보를 나타낸다. 광센서의 부품 감지 상황을 알린다. <1>는 단순히 정보만을 제공하므로 참조만 할 뿐 토큰의 보존성에 구애받지 않는다고 가정한다.

그림 3.(a), (b)는 각각 입, 출력 컨베이어에 대한 모델링이다. 두 컨베이어는 담당하고 있는 작업의 종류가 다른 뿐 사

실상 같은 동작을 하고 있다. 단 방향으로 움직이고 있고 광센서(L₁, L₂)의 정보에 따라 정지와 동작을 반복한다. 광센서는 부품이나 조립완성품의 이동상황을 알려주는 정보를 제공한다.

그림 4는 로봇 1을 중심으로 하는 개별요소 네트워크이다. 로봇 1은 L₁의 정보제공으로 부품의 도착을 감지하고 컨베이어 1으로 이동하여 부품을 집어 버퍼에 조립순서대로 저장한다.

그림 5는 로봇 2를 중심으로 하는 개별요소 네트워크이다. 로봇 2는 L₃의 상태로 버퍼에 조립될 부품의 저장 유무를 알 수 있다. 로봇 2는 부품을 집어 치구로 옮기고 조립작업을 한다. 조립이 마칠 때까지 필요한 부품을 버퍼에서 가져와 이 동작을 반복한다. 조립이 완료되면 완성된 부품을 컨베이어 2(출력 컨베이어)에 하역한다.

그림 7은 무인 반송차의 모델링이다. 무인 반송차는 부품을 자동저장 창고에서 가져와 입력 컨베이어에 하역하고, 조립된 완성품을 자동저장창고에 저장하는 두 가지 일을 담당한다. 작업을 마친 무인 반송차는 무인 반송차 스테이션에서 대기 상태에 있다.

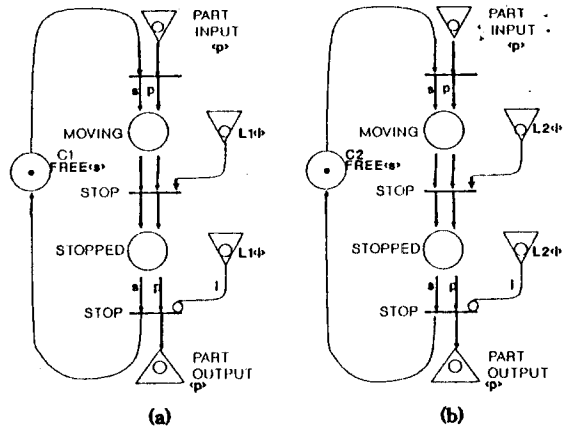


그림 3. (a)입력 컨베이어 모델링, (b)출력 컨베이어 모델링

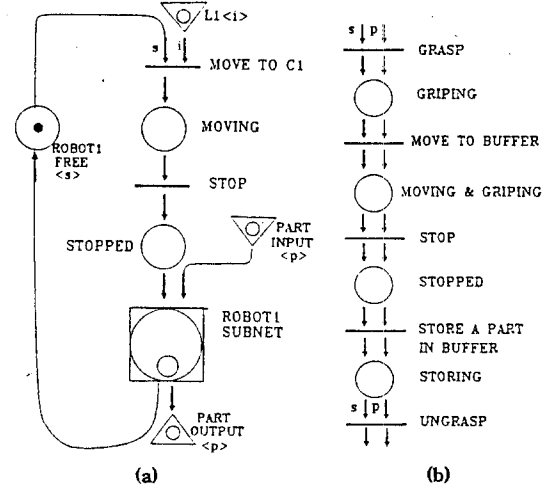


그림 4. (a) 로봇 1 모델링, (b) 로봇 1 서브넷

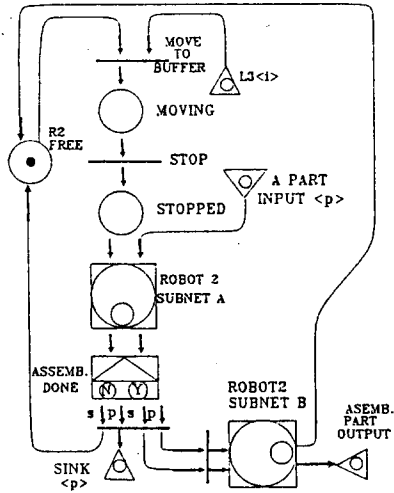


그림 5. 로봇 2 모델링

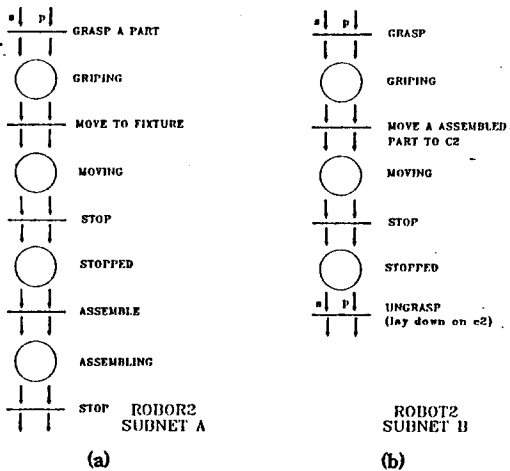


그림 6. (a) 로봇 2 서브네트 A, (b) 로봇 2 서브네트 B

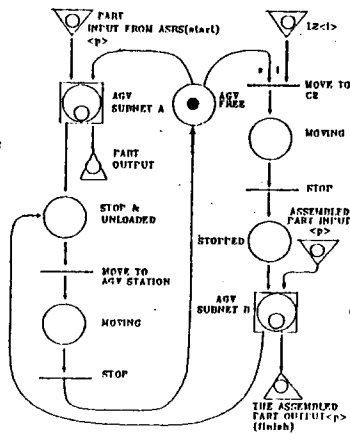


그림 7. 무인 반송차 모델링

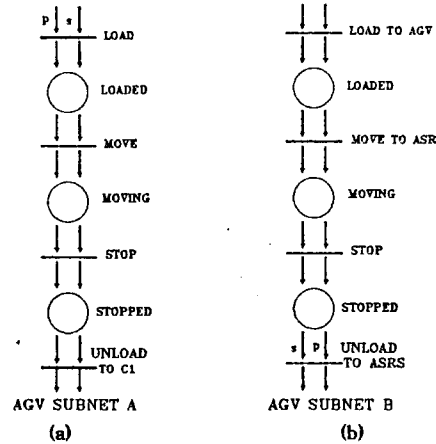


그림 8. (a) 무인 반송차 서브네트 A, (b) 무인 반송차 서브네트 B

C-3. 전체 시스템 모델링

그림 9는 개별요소 네트를 종합하고 상호관계를 고려하여 구성한 전체 시스템 모델링이다. 상황파악이 용이하도록 서브네트를 이용하여 표현하였다. 주목할 것은 토큰의 불변성을 유지하기 위하여 스위치 플레이트로부터 조립작업이 끝나지 않았을 때 분기하는 곳에 소멸 플레이트를 위치시켰다. 이것은 완성품은 1개이고 조립될 부품은 다수이기 때문에 생긴 결과이다.

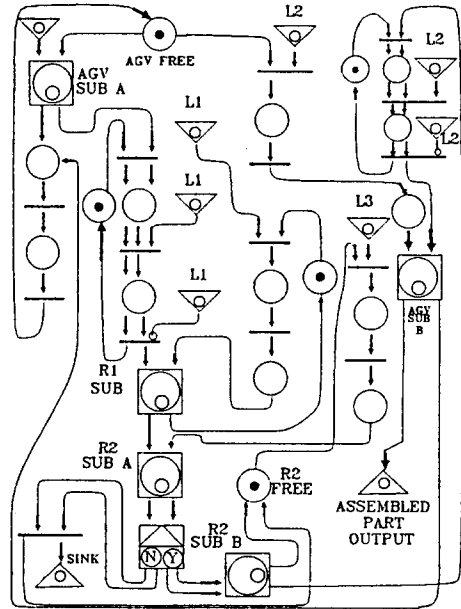


그림 9. 전체 시스템 모델링

III. 모의실험 기법

PN 모델에 대하여 모의실험을 하기 위해서는 우선적으로 입력파일을 만들어야 한다. 입력파일의 구성은 다음과 같다.

o place의 경우

모든 place의 갯수				
ID	Descriptor	출력 transition 갯수	Output transition list	Marking
·	·	·	·	·
·	·	·	·	·

ID : place의 ID

Descriptor : place에 대한 조건이나 정보 표시

Output transition list : 각 place에서 출력되는 transition들의 list

Marking : 각 place들이 가지고 있는 token의 갯수

o transition의 경우

모든 transition의 갯수						
ID	Descriptor	Inhibit list	입력 place 갯수	Input place list	출력 place 갯수	Output place list
·	·	·	·	·	·	·
·	·	·	·	·	·	·

ID : transition의 ID

Descriptor : transition에 대한 사건이나 정보 표시

Inhibit list : transition으로 입력되는 inhibit list

Input place list : transition으로 입력되는 place들의 list

Output place list : transition에서 출력되는 place들의 list

위와같이 입력화일을 만든 후 모의실험을 수행한다. 그림 10에 대한 입력화일 구성 예는 다음과 같다.

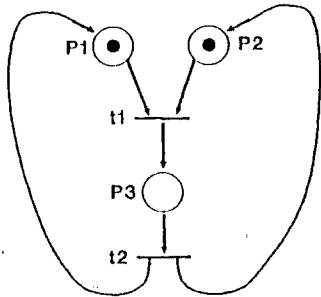


그림 10. 페트리 네트의 간단한 예

```

3
p1 NULL 1 t1 1
p2 NULL 1 t1 1
p3 NULL 1 t2 0
2
t1 NULL NULL 2 p1 p2 1 p3
t2 NULL NULL 1 p3 2 p1 p2
    
```

PN 모델에 대하여 모의실험을 수행하기 위한 알고리즘은 다음과 같다.

1. 모든 place들에 대하여 token의 유무를 결정한다. 한개 이상의 token을 갖는 marked place만에 대한 데이터베이스를 만든다.

2. marked place를 입력 place로 하는 enable transition만의 데이터베이스를 만든다. enable transition은 firable하다.

3. fired transition의 출력 timed place는 필요한 기간동안 공정을 수행한다.

4. transition이 fire되면 입력 place에서 token을 하나 빼고, 출력 place에는 token을 하나 더한다.

5. marked place와 enable transition에 대한 각각의 데이터베이스를 update시킨다.

하지만, 위의 알고리즘에서 중요한 문제가 발생한다. 그것은 동시에 여러개의 transition이 enable될 경우 어느 transition을 선택할 것인가에 대한 문제이다. 이런 경우가 바로 "conflict"가 발생한 경우이다. 이런 경우 fire되지 않은 transition들은 "wtrans"라는 waiting transition 데이터베이스로 삽입된다. 이 transition들은 fire되지 않은 한 이 데이터베이스에서 제거되지 않는다. 이 데이터베이스의 구조는 waiting transition이 발생하면 순서대로 데이터베이스의 가장 아래부분부터 삽입된다. 이것은 waiting transition들에 우선권을 할당하게 된다. 따라서 이 데이터베이스의 가장 상위에 있는 transition은 가장 오랫동안 waiting된 것이고, 가장 높은 우선권을 가지게 된다.

한편, enable transition이 없다면 그것은 "deadlock" 상태

IV. 결과 및 고찰

본 논문에서는 그림 2의 FMS 모델에 대하여 모의실험을 하였다. 그림 2에 대한 APT(Average Production Time)와 Throughput에 대한 결과는 그림 11,12,13에 나타내었다. 그림 11은 AGV의 속도를 변화시키면서 수행한 Throughput을 보인다. 그림 11에서 AGV의 속도가 100에 가까울수록 Throughput이 크게 증가함을 알 수 있다. 그림 12는 conveyor의 속도를 변화시키면서 수행한 Throughput을 나타낸 것으로 AGV에 대한 Throughput보다 상대적으로 그 값이 작으며 속도변화에 대해서도 영향을 적게 받음을 알 수 있다.

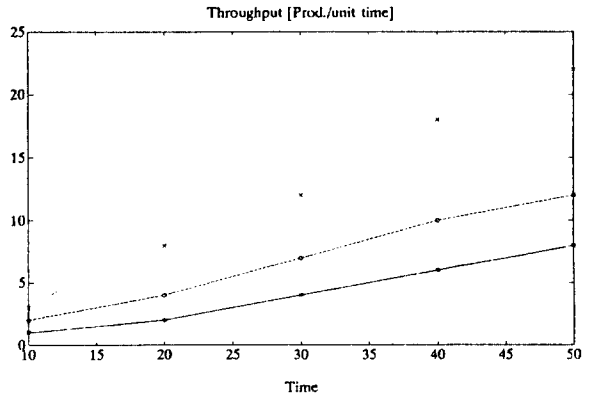


그림 11. 무인운반차 속도와 시간변화에 따른 Throughput

· : 10% ○ : 50% × : 100%

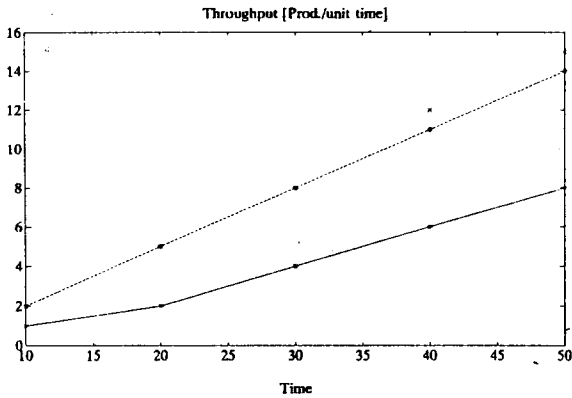


그림 12. 컨베이어 속도와 시간변화에 따른 Throughput

* : 10% o : 50% x : 100%

그림 13은 AGV, conveyor, robot에 대한 APT(Average Production Time)를 나타낸다. 그림 13에서 conveyor와 robot는 속도변화에 대하여 APT가 적게 변화하지만, AGV는 그 속도가 변화함에 따라 APT가 크게 증가됨을 알 수 있다. 따라서, 그림 2의 FMS 모델은 그림 11,12,13에서 알 수 있듯이 AGV의 속도변화에 크게 의존함을 알 수 있다.

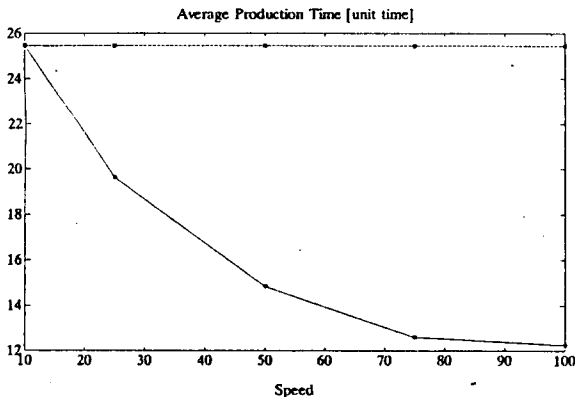


그림 13. 무인운반차, 컨베이어, 로봇 속도변화에 따른 APT

* : AGV o : Conveyor x : Robot

V. 결론

본 논문에서는 확장 페트리 넷를 사용하여 FMS 모델링 및 성능평가를 하였다. 모의실험 및 성능평가를 위하여 시뮬레이터를 개발하였다. 시뮬레이터가 갖는 의의는 임의의 시스템에 대해서 최적의 상태를 예측할 수 있다는 것이다. 최적의 상태를 예측하기 위해 시뮬레이터를 이용하여 로봇, AGV, 컨베이어의 속도나 조립시간등의 요소를 변화시켜 시스템의 최적상태를 예측할 수 있다.

앞으로 수행해야 할 연구과제는 발생가능한 여러가지 상황(자원의 고장, 오동작 등)을 고려하고, 복잡한 구조의 대규모 시스템에 대한 적용이 되어야 하며, 통신을 이용하여 실시간으로 실제 시스템을 제어하고, 사용자 인터페이스를 위해 그래픽 기능을 보강하여야 한다.

참고문헌

- [1] Y. C. Ho, "Performance Evaluation Analysis of Discrete Event Dynamic System", IEEE Trans., Automatic Control, Vol. AC-32, pp.563-572, 1987
- [2] 홍지민, "로봇적 조립선의 모델링 및 시뮬레이션에 관한 연구", 서울대학교 제어계측공학과 석사학위 논문, 1991
- [3] 김대원, "지식 베이스를 이용한 로봇적 조립선의 모델링 방법", 서울대학교 제어계측공학과 박사학위 논문, 1990
- [4] K. P. Valavanis, "On the Hierarchical Modeling Analysis and Simulation of Flexible Manufacturing Systems with Extended Petri Nets", IEEE Trans. Systems, Man and Cybernetics, Vol.20, No. 1, pp.94-110, 1990
- [5] 안인석, 서기성, 우광방, "확장된 Petri-nets를 이용한 FMS의 모델링 및 스케줄링", 1991, 한국 자동제어 학술회의 논문집, pp.488-492, 1991
- [6] 임용희, 김대원, "페트리 넷를 이용한 유연 로봇 조립선의 시뮬레이션에 관한 연구", 1993