

로보틱 워크셀을 위한 그래픽 시뮬레이터의 구성에 관한 연구

A Study on the Design of the Graphic Simulator for a Robotic Workcell

李 相 武* · 李 範 熙** · 高 明 三***
(Sang-Moo Lee · Bum-Hee Lee · Myoung-Sam Ko)

요 약

본 논문에서는 조립 작업 단위(assembly cell)를 대상으로한 그래픽 시뮬레이터를 구성하였다. 대상 조립작업단위는 두 대의 PUMA560 매니퓰레이터, 콘베이어, 작업 테이블, 비전 센서 등으로 구성되었고, 조립 작업 단위 안에서 진행되는 작업을 기술하여 그 제어 로직을 모델링하고 그로부터 시뮬레이터를 구성하기 위한 도구로 페트리 넷(Petri Net)이론이 도입되었다. 모델링된 작업단위를 시뮬레이션하기 위해 사건 스케줄링 방법(event scheduling approach)을 이용하였고 이 방법에 의해 진행되는 시뮬레이션 과정을 그래픽으로 나타내기 위하여 로봇 모델링, 요소의 모델링, 월드 모델링 등의 기능들이 포함되었다.

위의 방법으로 구성된 시뮬레이터를 이용하여, 시뮬레이션이 진행되면서 발생하는 시스템의 상태 변화를 시각적으로 나타낼 수 있었다. 또한 작업 단위를 구성하는 자원들의 상태를 변화 시킴으로써 발생하는 시스템의 성능 변화를 비교, 평가하여 가장 좋은 상태를 선택하는 도구로도 사용될 수 있었다.

Abstract- This paper presents the development of the graphic simulator for an assembly workcell. The assembly workcell consists of two PUMA560 manipulators, a conveyor belt system, a work table, and a vision sensor. In this study, the Petri Net theory is applied to model the assembly workcell and to construct the simulator. The event scheduling approach is used to simulate the cell. In order to show the graphic display of the simulation process, robot modelling, component modelling, and world modelling are included. The developed simulator is used to display the transition of the system state during the simulation. It is also used as a tool in selecting the best resource states by studying the performance of the system as the resource states are changed.

1. 서 론

1.1 연구 배경

산업의 규모가 커지고 복잡해짐에 따라, 다양하게 변화하는 생산 요구에 대응하기 위해, 자동화 시스템에 대한 관심이 날로 증대되어 가고 있다. 특히 공장 자동화에 있어서, 컴퓨터 기술의 발전은 로봇들과 자동화된 기계들로 구성된 무인 공장의 개념을 발전시켜 산업용 로봇의 필요성은 날로 증대되어 가고 있다.

새로운 생산 요구에 부응하기 위한 자동화된 작업 환경을 설계할 때, 설계자는 여러 가능한 대안들을 얻게 된다. 가능한 여러 대안 중에서 설계자는 각 설계를 분석, 평가하여 가장 성능이 좋은 안을 선택하게 되는데 이 때 분석, 평가의 방법 중 실제 시스템의 동작에 가깝게 접근할 수 있는 방법이 시뮬레이션을 이용하는 것이다. [7] 주어진 시스템을 시뮬레이션하고자 할 때 시스템의 동작을 정의해 주는 모델이 필요하게 된다. 이 때 모델링 방법으로는 시스템을 구성하는 요소(component)와 요소의 상태를 나타내는 변수(component state), 그리고 각 요소들간의 상관 관계(interaction)들로 구성되는 변수들로 나타내는 방법과 [8-9] 최근에 관심을 모으고 있는 Petri Net을 이용한 모델링 방법[10-17]이 있다. 주어진 시스템을 나타내는 모델은 시뮬레이션 프로그램으로 번역되어 실행되어 진다. 시뮬레이션 프로그램은 모델로부터 시스템의 상태가 변함에 따라 발생하게 될 사건들을 모은 사건 리스트(event list)를 작성하고 각 사건들은 번역 프로그램(interpreter program)에 의해 번역, 해석되어 실행되어 진다. [8-9]

본 논문에서는 그래픽 기능을 갖는, 로봇틱 워크셀(robotic workcell)을 위한 시뮬레이터를 구성하기 위한 첫단계로 모델링된 제어 로직에 의해 작업을 수행하는 조립 작업 단위에 대한 그래픽 시뮬레이터 구성에 대하여 설명한다. 조립작업을 모델링하기 위해 페트리네트(Petri Net)이론이 도입되었고 시스템 동작 중 발생 가능한 사건들을 처리하여 시뮬레이션을 행하는 사건 스케줄링(event scheduling) 방법을 이용하고, C언어를 이

용하여 시뮬레이터를 구성하였다. 또한 작업 과정을 시각적으로 나타내기 위하여 기하학적 모델링 기능, 월드 모델링 기능 및 그래픽 디스플레이 기능 등을 갖추고 있다.

1.2 시스템의 구성

본 시뮬레이터는 작업 단위에서 로봇트, 콘베이어, 작업 테이블 등 자원들의 위치 변화와 로봇트 축의 속도, 콘베이어 이동 속도, 비전 처리 속도 등 자원들의 상태 변화가 작업 단위의 성능에 미치는 영향을 시뮬레이션을 통하여 평가하고 작업 진행 과정을 그래픽을 이용하여 나타내기 위해 다음의 부분들로 구성되어 있다.

즉, 작업 단위 내에서 자원들의 위치와 상태 변화 등을 대화식으로 받아 들여 시뮬레이션하기 위한 시뮬레이션 부분, 로봇트의 궤적을 계획하고 로봇트 축 속도 정보를 이용하여 로봇트 작업 시간을 계산하고 실행하기 위한 로봇트 궤적 계획 부분, 대화식으로 입력된 관찰자 위치, 작업 단위에서의 윈도우(window), 스크린 상의 디스플레이 위치 정보 등을 이용하여 작업 단위의 변화를 시각적으로 나타내기 위한 그래픽 디스플레이 부분과 그래픽을 위해 자원들의 기하학적, 기구학적 정보를 생성, 저장하는 모델링 부분으로 구성되어 있다. 그림 1.1은 본 시뮬레이터의 블럭 다이어그램이다.

2. 작업 단위의 모델링(Workcell Modelling)

하나의 작업 단위는 전체 시스템 내에서 주어진 작업을 수행하기 위하여 여러 대의 로봇트, 자동

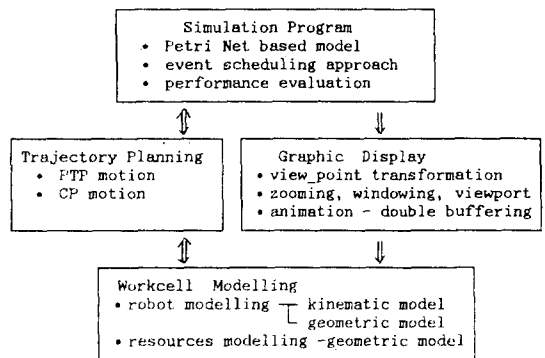


그림 1.1 시뮬레이터의 블럭 다이어그램

그림 1.1 시뮬레이터의 블럭 다이어그램
Fig. 1.1 Block Diagram of the simulator.

*正 會 員 : 大宇重工業(株) 中央研究所 研究員
 **正 會 員 : 서울대 工大 制御計測工學科 助教授·工博
 ***正 會 員 : 서울대 工大 制御計測工學科 教授·工博
 接受日字: 1989年 5月 16日
 1次 修正 : 1990년 3月 21日

화된 기계들, 부품(part)을 이동시키기 위한 콘베이어(conveyor), 작업 테이블(assembly table), 비전센서(vision vensor)등으로 구성될 수 있다.

본 논문에서는 이 중 조립 작업을 수행하는 조립 작업 단위(assembly cell)를 대상으로 한다. 대상 작업 단위는 두 대의 로봇, 콘베이어, 작업 테이블, 비전 센서, 부품 등으로 구성되며 이러한 자원들(resources)은 계층적인 구조를 이루어 하나의 작업 단위를 구성한다.

2.1 요소의 모델링(Component Medelling)

본 논문에서는 3차원 물체를 와이어 프레임 모델(wire-frame model) [6]로 나타내기 위하여 대상 물체에 대한 기준 좌표계를 설정하고 기준 좌표계에 대한 각 꼭지점들의 상대적인 3차원 좌표 정보와 면을 구성하는 꼭지점들에 대한 정보를 저장하는 방법을 취하였다. 이때 물체를 구성하는 꼭지점들의 순서는 우선 밀면을 바깥 방향에서 수직으로 보아 밀면을 구성하는 꼭지점들을 반 시계 방향으로 정하고 윗 면에 대해서도 같은 방향으로 순서를 설정하였다. 각 면을 구성하는 꼭지점에 대하여는 각 면을 바깥 방향에서 수직으로 보아 반 시계 방향을 이루도록 순서를 정하였다. 그림 2.1은 위의 방법에 의한 모델링 예를 나타낸다.

2.2 로봇 모델링(Robot Medelling)

로봇은 여러 모양의 링크(link)들이 축(joint)을 기준으로 특수한 형태로 결합되어 구성된다. 연속적인 두 링크는 축을 기준으로 회전하거나(revolute joint) 직선 이동(prismatic joint)을 하며 각 링크는 특수한 형태를 갖는 3차원적 요소로 이루어진다. 이러한 사실들을 바탕으로 로봇은 요소, 링크, 축 등으로 계층적(hierachical)으로 이루어지는 구조적(structural)형태로 모델링된다.

[1-5] 그림 2.2는 위의 구조로 구성된 PUMA560 매니플레이터의 모델링 예이다. 즉 로봇은 구조적인 형태로 다음과 같은 정보를 저장한다.

- 이름 ; 대상 로봇의 이름을 저장한다.

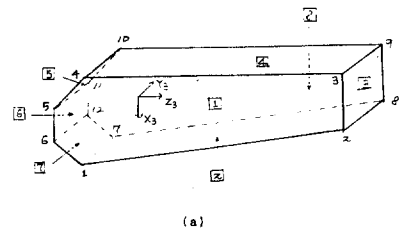
예) PUMA560

- 형태 (Type) ; 로봇 형태를 저장한다.

예) R-R-R

- 기준 좌표계 ; 로봇 기준 좌표계(base coordinate system)에 대한 정보를 저장한다. 즉 월드 좌표계(world coordinate system)에 대한 로봇 기준 좌표계의 위치(position), 방향(orientation), 기준 좌표계의 역 변환(inverse transformation)에 관한 정보를 갖는다.

- 축(joint) 수 ; 로봇을 구성하는 축의 수를 저장한다.
- 축(joint)에 대한 포인터(pointer) ; 각 축에 대한 정보를 저장하는 구조에 대한 정보를 갖는다.
- 툴(tool)에 대한 포인터 ; 로봇 핸드(hand)에 연결되는 엔드 이펙터(End-Effector)에 대한 정보를 갖는다.
- 기구학에 대한 포인터 ; 링크들이 결합된 형태에 대한 기구학적 정보를 갖는 프로그램에 대한 정보를 갖는다.
- 역 기구학에 대한 포인터 ; 역 기구학을 푸는데 필요한 프로그램에 대한 정보를 갖는다. 위와 같은 방법을 이용하여 대상 로봇을 모델링할 경우 모델링 결과를 쉽게 확인할 수 있을 뿐 아니라 같은 종류의 로봇을 여러 대 모델링할 경우 링크의 모델링 결과를 공유하여 기억 장소를 절약할 수 있는 장점이 있다.



surface	vertex no.(CCW)
1	1,2,3,4,5,6
2	2,1,7,8
3	3,2,8,9
4	4,3,9,10
5	5,4,10,11
6	6,5,11,12
7	1,6,12,7
8	12,11,10,9,8,7

vertex no.	Coordinate		
	X	Y	Z
1	127.5	-42.5	-85.93
2	42.5	-42.5	353.07
3	-42.5	-42.5	353.07
4	-42.5	-42.5	-96.93
5	7.5	-42.5	-131.93
6	77.5	-42.5	-131.93
7	127.5	42.5	-85.93
8	42.5	42.5	353.07
9	-42.5	42.5	353.07
10	-42.5	42.5	-96.93
11	7.5	42.5	-131.93
12	77.5	42.5	-131.93

그림 2.1 요소의 모델링 예
Fig. 2.1 Example of a component Modelling.

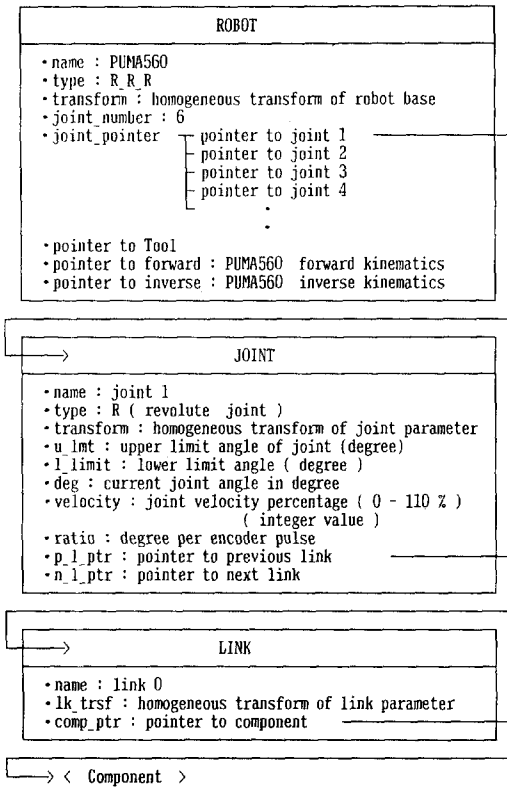


그림 2.2 PUMA560 매니플레이터의 모델링
Fig. 2.2 PUMA 560 Maniprator Modelling.

3. 셀 시뮬레이터(Cell Simulator)

3.1 이산 사건 시스템(Discrete Event Dynamic System)

미분 방정식에 의하여 표현되지 않고 불 연속적인 이산 사건(discrete event)의 복잡, 미묘한 상호 관계에 의해서만이 표현 가능한, 인간이 만든 동적 시스템을 DEDS(Discrete Event Dynamic System)라 부른다.

그 예로는 FMS(Flexible Manufacturing System)나, 통신 망 구조, 교통 시스템 등이 있다. 여기서 사건이라 함은 임의의 지점에서, 임의의 시간에 시스템 요소들 내에서 발생하는 불 연속적인 변화의 발생을 의미한다. 그리고, 이 시스템에서의 상태 전이(state transition)는 단지 비 동기 이산순간(asynchronous discrete instants)에만 발생한다. 그러나 오랜 연구 역사를 지닌 CVDS(continuous variable dynamic system)와는 달리

DEDS에 대한 좋은 해석적 모델이나 기술은 미비한 형편이다. [20]

최근 공장 자동화의 기본 개념인 FMS에 대한 연구가 활발히 진행되고 있으며, FMS의 모델링과 분석 그리고 성능 평가를 위해서 광범위 하고도, 실제적인 관심과 함께 몇 가지의 방법들이 제시된 바 있다.

이중 QN(Queueing Network)모델은 [7-8]시스템의 입출력 동작에 의한 분석만이 가능하고 시스템 내부 구성 요소의 동작 파악이 곤란하며, 강한 수학적 가정이 필요하다는 단점이 있다. 그러나, FMS 내에서의 시스템역학, 불 확실성 등을 고려하고 있고, 매우 효율적인 계산 알고리즘의 이용이 가능하며, 성능 평가를 위한 좋은 모델을 제공한다고 볼 수 있다.

PN(Petri Net)모델은 [12, 14] 도식법(graph method)에 의한 모델로서 FMS의 모델링과 분석을 위한 완전한 틀 구조/framework)를 제공한다. 그러나, 이모델은 생동감(liveness)과 동일 데이터 구조(concurrent data structure)에 대한 검증 방법이 부족하며, 시스템 역학으로부터 제어 메카니즘을 분리해 내기가 어려운 단점이 있다.

시뮬레이션에 의한 분석 방법은 [7-9]FMS내에서의 불 확실성을 고려할 뿐 아니라 시간 경과에 따른 시스템 내부 구성 요소의 상호 작용과 상태 전이등 동작 파악이 가능하여 어떤 시각에서의 특정 현상에 대한 연구도 가능하며, 구현하고자 하는 실제 시스템에 가장 근접한 방법으로 다양한 형태의 정보를 얻을 수 있는 장점이 있으나 비용과 시간이 많이 걸린다는 단점이 있다.

3.1.1 성능 평가 척도(Performance Measure)

시스템의 성능을 평가하기 위한 척도는 크게 다음의 두 종류, 즉 사용자 중심의 평가 척도(user-oriented measure)와 시스템 중심의 평가 척도(System-oriented measure)로 구분될 수 있다. [8]

사용자 중심의 척도는 작업(job)이 시스템 내부에 입력된 후 작업이 완료되어 결과를 사용할 수 있을 때까지 경과 시간을 나타내는 턴어라운드 시간(turnaround time), 시스템에 작업이 입력된 후 작업이 서비스를 받기 시작할 때까지의 경과 시간을 나타내는 반응 시간(response time)을 말한다.

시스템 중심의 평가 척도는 단위 시간당 처리되는 작업의 평균 수를 나타내는 단위 생산량(throughput), 특정 재원이 작업을 하는 시간의 비율을 나타내는 사용율(utilization)등을 말한다.

본 논문에서는 이 중 시스템 중심의 평가 척도인 단위 생산량(throughput)과 자원들의 사용률(utilization)의 개념을 바탕으로 다음의 정보들에 의하여 구성된 시스템의 성능을 평가하고자 한다.

- 총 생산량
- 로봇의 총 작업 시간
- 비전 센서의 총 처리 시간
- 콘베이어의 총 이동 시간

3.2 상태 모델(State Model)

3.2.1 페트리 넷(PN : Petri Net)

3.2.1.1 개 론

페트리 넷은 DEDES와 같은 비 동기적(asynchronous)이고 동시 수행적(concurrent)인 시스템의 동작을 기술하고 분석하기 위한 도식적 모델링 도구(graphic modelling tool)로 다중처리(multi-processing)시스템, 분산 처리(distributed processing)시스템, FMS(Flexible manufacturing System)등의 분야에서 시스템 모델링 및 분석 도구로서 응용되고 있다.

[10-16] 특히, FMS 응용 분야에서 페트리 넷은 FMS 모델링, FMS 제어 로직의 설계, 검증 및 구현, FMS 시뮬레이터의 설계 및 FMS 성능 분석 등 여러 분야에 걸쳐 이용되고 있다.

페트리 넷을 FMS모델링에 이용할 경우, 가장 큰 장점은 시스템의 정성적 분석뿐 아니라 정량적 분석도 할 수 있다는 것이다. 이 때 시스템의 정성적 분석을 위해 P 인베리언트(P Invariant)에 의한 구조적 분석 방법(structural analysis)이, 정량적 분석을 위해서는 페트리 넷에 시간의 개념을 도입한 타임드 페트리 넷(timed Petri Net)가 사용되고 있다.

3.2.1.2 페트리 넷의 구조적 성질(Structural Properties of Petri Net)

다음에 페트리 넷(PN:Petri Net)의 정의 몇 가지를 소개한다. [10-14]

정의 1

페트리 넷은 다음의 3요소로 이루어 진다. $N=(P, T, \Phi)$ 여기서

- $P=\{p_1, \dots, p_n\}$: n 개의 플레이스의 집합
- $T=\{t_1, \dots, t_m\}$: m 개의 트랜지션의 집합
- $\Phi \subseteq (P \times T) \cup (T \times P)$: 플레이스와 트랜지션을 연결하는 아크(arc)들의 집합

그림에서 플레이스는 원으로 트랜지션은 바(bar)로 나타낸다.

정의 2

PN의 마킹(marking) \vec{m} 은 플레이스에 위치한 토큰(token:그림에서 점으로 나타낸다.)의 수를 나타내는 정수 값을 갖는 칼럼 벡터이다.

$$\vec{m} = [m_1, \dots, m_n] \text{ 여기서 } m_j \geq 0.$$

정의 3

트랜지션 t_i 의 입력조건이 되는 플레이스들이 각각 적어도 하나 이상의 토큰을 포함하면, 트랜지션 t_i 는 인에이블(enabled)된다. 인에이블된 트랜지션은 파이어링(firing)되어 토큰 이동에 의한 새로운 마킹을 형성한다.

정의 4

$n \times m$ 인 시던스 행렬(incidence matrix) 요소 $C_{i,j}$ 는 다음과 같이 정의된다.

$$C_{i,j} \begin{cases} -1 & \text{if } (p_i, t_j) \in \Phi \\ 1 & \text{if } (t_j, p_i) \in \Phi \\ 0 & \text{otherwise} \end{cases}$$

즉 플레이스와 트랜지션 사이의 상관 관계를 나타낸다.

정의 5

다음의 관계를 만족하는 0이 아닌 로우벡터(row vector)를 \underline{p} 인베리언트(p invarient)라 한다.

$$\underline{p} C \vec{0}.$$

정의 6

주어진 초기 마킹 \vec{m}_0 에 대해 도달 가능한 마킹의 집합(set of reachable marking), $R(\vec{m}_0)$, 은 초기 마킹 \vec{m}_0 에서 생성 가능한 마킹들의 집합을 말한다.

정의 7

초기 마킹 \vec{m}_0 에 대해 $R(\vec{m}_0)$ 의 모든 마킹에서 트랜지션 t_i 가 인에이블(enable)되는 마킹으로 도달하는 파이어링(firing)순서가 있으면 트랜지션 t_i 는 살아있다(live)고 한다.

3. 2. 2 표시된 그래프(Marked Graph)[10]

PN모델중 다음의 경우를 특히 표시된 그래프라 한다. 즉 모든 플레이스에 대하여 각 플레이스가 정확히 하나의 입력 트랜지션과 하나의 출력 트랜지션을 가질 때 이러한 넷을 표시된 넷이라 한다. 다음에 표시된 넷에 대한 정리 몇가지를 소개한다.

정의 1

만약 어떤 플레이스들의 집합이 표시된 그래프에서 사이클(cycle)을 이루면, 그 특성 벡터(characteristic vector)는 \underline{P} 인베리언트(P invarient)이다.

정의 2

표시된 그래프 N 이 적어도 하나의 표시된 플레이스를 포함하고 N 의 모든 플레이스들이 정확히 하나의 표시된 플레이스를 갖는 사이클에 포함된다면, 표시된 그래프 N 은 살아있고(live) 안전하다(safe).

3. 2. 3 조립 작업 단위(Assembly Cell)의 상태 모델(State Model)

본 논문에서 개발된 그래픽 시뮬레이터는 두 대의 로봇트 매니플레이터를 갖는 자동화된 조립 작업 단위 (Assembly Cell)를 대상으로 한다. 그림 3.1에 보여진 조립 공정에서 다음의 작업이 진행된다고 가정한다. 즉 R1(Robot 1)이 C(Conveyor)에 의해 작업 단위 안으로 이동된 부품, P(Part)를 V(Vision Sensor)의 처리 결과를 이용해 잡아, 이를 A(Assembly Table)에 위치 시킨다. 이때 R2(Robot 2)는 A에서 P를 대상으로 작업을 하고, 작업이 끝난 후 P를 C로 이동시켜 조립 작업 단위 밖으로 이동시킨다. 이와 같은 조립 공정의 동작 규칙(operation rule)은 다음과 같이 정의된다.

(1) C(Conveyor)는 이동·정지 동작을 반복한다. 즉 R1이 V의 처리 결과를 이용해 P를 집으면, C는 이동하여 새로운 P를 작업 단위 안으로 이동 시킨 후 정지하여, 비전처리(vision processing)가 끝나 R1이 P를 집을 때까지 정지 상태에 있다.

이때, 다음의 사항을 가정한다.

1) 파트 피더(part feeder)는 C와 동기되어 콘베이어가 움직이기 시작할 때 하나의 부품을

콘베이어에 일정한 방향으로 위치시킨다.

2) 부품은 파트 피더에 의해 콘베이어 위의 일정한 장소에 위치되고 콘베이어에 의해 이동되어 일정한 장소에서 정지한다.

(2) V(Vision Sensor)는 C가 정지한 후 처리(processing)를 시작한다. 이때 비전 처리(vision processing)에 걸리는 시간은 일정하다고 가정한다.

(3) R1(Robot 1)이 작업 대기 중(R1-free)이고 비전 정보가 사용 가능하면 R1은 C위의 부품 P를 집는다.

(4) R1이 P를 집은 후(R1-busy), A(Assembly Table)가 비어 있으면 R1은 P를 A로 이동시킨다.

(5) R2(Robot 2)가 작업 대기 중(R2-free)이고 A에 P가 존재하면 R2는 A로 이동하여 P를 집는다.

(6) R2는 P를 집은 후 (R2-busy), 콘베이어 위로 이동하여 콘베이어에 P를 놓는다.

(7) (1)-(6)의 동작을 정해진 시간동안 반복 수행한다.

표3.1은 위와 같은 동작규칙에 의해 정의된 플레이스를 나타내고, 그림 3.2는 조립 작업 단위의 모델링 결과와 그에 따른 P 인베리언트(P invariants)와 초기 마킹(intial making), 인시던스 행렬(incidence matrix)을 나타낸다. 그림 3.2(a)에서 대상 작업단위의 PN모델이 특히 표시된 그래프가 됨을 알 수 있다. 또한 각 P인베리언트는 정확히 하나의 플레이스를 포함하고 있음을 알 수 있다. 따라서 정리 2에 의해 조립작업단위의 PN 모델은 살아있고 안전하다는 것을 알 수 있다.

3.3 로봇트 동작 모델링

MOVEP, MOVES등과 같은 작업 중의 로봇트 핸드(hand)이동에 관련된 명령어들에 의한 로봇트의 움직임을 모델링하여 작업 단위 내에서의 재원들의 위치 변화에 의한 로봇트 핸드 이동 거리 변화와 로봇트 각 축의 속도 변화에 의한 로봇트의 작업 시간 변화를 효과적으로 나타내고 그에 따른 작업 단위의 성능 변화를 파악하기 위하여 점간 운동(PIP motion) [18]과 직선 운동(CP motion) [19]에 대한 궤적 계획(trajjectory planning) 프로그램을 작성하였다. 로봇트의 궤적 계획은 작업 단위의 성능 평가외에 로봇트 움직임을 효과적으로 나타내기 위해서도 필요하다.

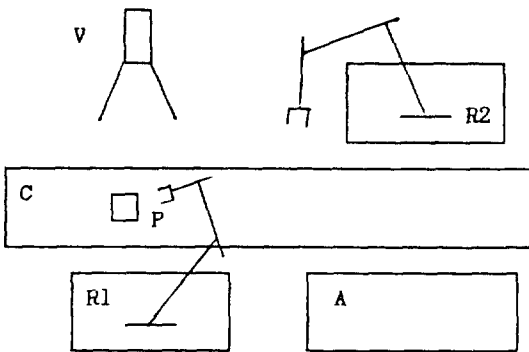
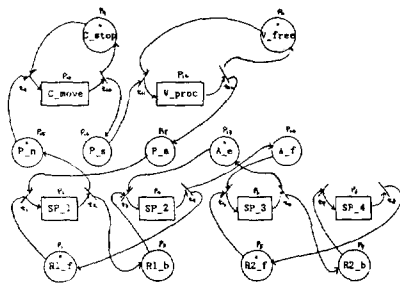


그림 3.1 조립 작업 단위의 예
Fig. 3.1 Example of an Assembly warkcell.

표 3.1 플레이스의 정의

Table 3.1 Definition of places.

플레이스 1(P1)	R1이 작업 대기중(R1_f)
플레이스 2(P2)	R1이 C위로 이동하여 C위의 P를 집어올림 (SP-1)
플레이스 3(P3)	R1이 SP_1수행후 SP_2대기중(R1_b)
플레이스 4(P4)	R1이 A로 이동하여 A에 P를 위치함(SP-2)
플레이스 5(P5)	R2가 작업 대기중(R2_f)
플레이스 6(P6)	R2가 A위로 이동하여 A위의 P를 집어올림 (SP-3)
플레이스 7(P7)	R2가 SP-3 수행후 SP-4 대기중(R2_b)
플레이스 8(P8)	R2가 C위로 이동하여 C에 P를 놓는다(SP-4)
플레이스 9(P9)	C가 정지 상태(C_stop)
플레이스 10(P10)	C가 이동중(C_move)
플레이스 11(P11)	비전 처리가 일어나고 있지 않음(V_free)
플레이스 12(P12)	비전 처리중 (V_proc.)
플레이스 13(P13)	A에 P가 없음(A_e)
플레이스 14(P14)	A에 P가 있음(A_f)
플레이스 15(P15)	P가 작업단위안으로 이동중(P_n)
플레이스 16(P16)	C가 정지하여 비전처리가 진행중(P_s)
플레이스 17(P17)	비전처리가 끝나 R1이 P를 집을 수 있음(P_a)



(a) 조립 작업 단위의 Petri Net

Initial marking :
 $M_0 = [1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0]^T$
P_invariants :
(p1, p2, p3, p4)
(p5, p6, p7, p8)
(p9, p10)
(p11, p12)
(p4, p6, p13, p14)
(p2, p10, p12, p15, p16, p17)

Incidence matrix :

C =

-1	1	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
1	-1	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
1	-1	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
1	-1	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	-2	-1	:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	1	-1	:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	1	-1	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	1	-1	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	1	-1	:	:	:	:	:	:	:	:	:	:
:	-1	:	:	1	:	:	:	:	:	:	:	:	:	:	:	:
1	:	1	-1	:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	-1	:	:	:	:	:	:	:	:	:	:	:
-1	:	:	:	:	:	1	-1	:	:	:	:	:	:	:	:	:

(b) 초기 마킹, P_인베리언트, 인시던스 행렬

그림 3.2 조립 작업 단위의 모델링 결과
Fig. 3.2 Modelling Result of an Assembly warkecll.

3.4 시뮬레이터의 구성

본 논문에서는 주어진 조립 작업 단위를 대상으로 PN모형을 작성하였고, 그 모델이 타당한지를 표시된 그래프의 정리를 이용하여 정성적으로 검토하였다. PN모형을 대상으로 정량적 분석을 하기 위해 모델의 플레이스중 시간의 경과와 관계있는 동작, 즉 로봇의 궤적 계획에 의한 움직임, 컨베이어 이동등을 하나의 사건 발생에 의해 자원들의 상태에 따라 시간 경과가 변화하는 부분으로 인식하여 발생 가능한 사건 리스트를 작성하고 사건들을 처리함으로써 시뮬레이션을 행하는 사건 스케줄링 방법을 도입하여 시뮬레이터를 구성하였다. 시뮬레이터의 주요 기능들은 다음과 같다.

(1) 시뮬레이터 초기화(simulator initialization)

- 시뮬레이션에 필요한 변수들을 초기화시킨다.
- (시뮬레이터 시계, 로봇트 수, 자원 수, 초기 마킹, 인스턴스 행렬등)
- 그래픽 디스플레이에 필요한 모델을 생성, 저장한다.
- (요소의 모델링, 로봇트 모델링 등)

(2) 사건 리스트 초기화(event list initialization)

- 초기 마킹과 시뮬레이터 시계에 의해 사건 리스트를 초기화 한다.

(3) 사건 리스트 및 시뮬레이터 시계 변화

(scan event list and simulator clock update)

- 사건 리스트를 검색하여 발생 가능 시각에 따라 사건들을 재배열한다.
- 사건 리스트의 첫 사건(발생 시각이 가장 이른 사건)에 따라 시계를 변화시킨다.

(4) 시뮬레이션 종료 조건 검색

(test for terminating simulation)

- 시뮬레이터 시계의 시각으로 시뮬레이션 중단 여부를 판단한다.

(5) 사건 처리 루틴(event handling routine)

- 사건 리스트의 첫 사건 처리 루틴을 시행한다.
- 사건(event ; PN모델의 트랜지션)의 입력 플레이스에서 토큰을 제거하고 출력 플레이스에 토큰을 위치시킨다.
- 사건 발생에 의해 일어나는 동작(출력 플레이스에 해당하는 동작)을 시뮬레이션 한다.
- 작업 단위의 상태 변화를 디스플레이한다.
- 성능 평가 척도를 변화시킨다.

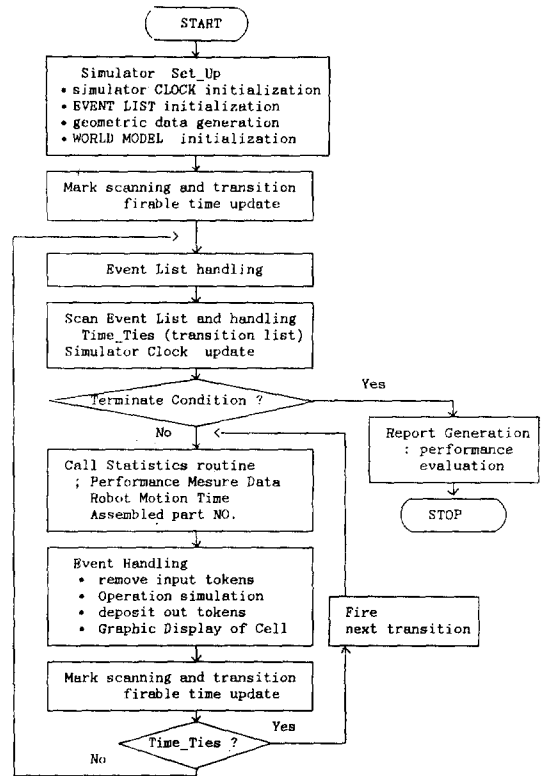


그림 3.3 시뮬레이터의 흐름도
Fig. 3.3 Simulator Flowohart.

이때 각 사건 발생에 의한 동작 처리는 궤적 계획에 의한 로봇트 핸드의 이동, 컨베이어 이동, 로봇트 작업에 의한 부품의 위치 변화, 비전 처리 혹은 단순한 자원들의 상태 변화 등의 방법으로 시뮬레이션된다.

(6) 사건 리스트 변화(marking scanning and event list update)

- 사건 처리에 따른 새로운 발생 가능 사건을 발생 가능 시각과 함께 사건 리스트에 삽입한다.

(7) 사건 동시 발생 여부 판단(check time-ties)

- 여러 개의 사건이 같은 시각에 발생하는지의 여부를 판단하고 이를 처리해 준다.

(8) 보고서 작성(report generation)

- 성능 평가 척도들을 모아 성능 분석에 필요한 정보를 출력 시킨다.

4. 셸 시뮬레이터의 그래픽 기능

구성된 시뮬레이터는 시뮬레이션을 실행하기에

앞서, 필요한 정보들, 즉 자원들의 초기 위치, 로봇 각 축의 이동 속도, 콘베이어 이동 속도 등의 초기 상태 등을 대화식으로 받아들여 그 결과를 그래픽으로 나타낼 뿐 아니라 시뮬레이션 진행 중 특정 부분의 동작을 쉽게 파악할 수 있게 하기 위한 확대, 회전 등의 기능도 포함하고 있다.

(1) 작업 단위 구성 요소들의 위치 결정

작업 단위를 이루는 자원들의 위치 선정에 의한 작업단위 구성과정을 나타내어, 자원들의 효과적인 배치와 그에의한 변환 모습을 확인할 수 있게 한다. 그림 4.1은 자원들의 위치 정보 입력에 따른 작업 단위의 형성 과정을 나타낸다.

(2) 구성 요소들의 초기 상태 입력

자원들의 위치가 결정된 후 입력되는 정보로는, 두 로봇의 각 축의 호전 속도(0-110% 사이의

표 4.1 콘베이어 이동 속도

Table. 4.1 Conveyor of velocity.

속도 백분율 (%)	10	20	30	40	50
콘베이어 속도(mm/sec)	10	25	40	60	80
속도 백분율 (%)	60	70	80	90	100
콘베이어 속도(mm/sec)	100	150	200	250	300

정수 값), 콘베이어의 이동 속도(0-100%사이의 정수 값)비전 센서의 처리 시간(sec 입력), 시뮬레이션 진행 시간(sec 입력), 로봇트 운동의 보간 종류(PTP, CP)등이 있다. 표4.1은 콘베이어 이동 속도와 백분율과의 관계를 나타낸다.

(3) 스크린 디스플레이

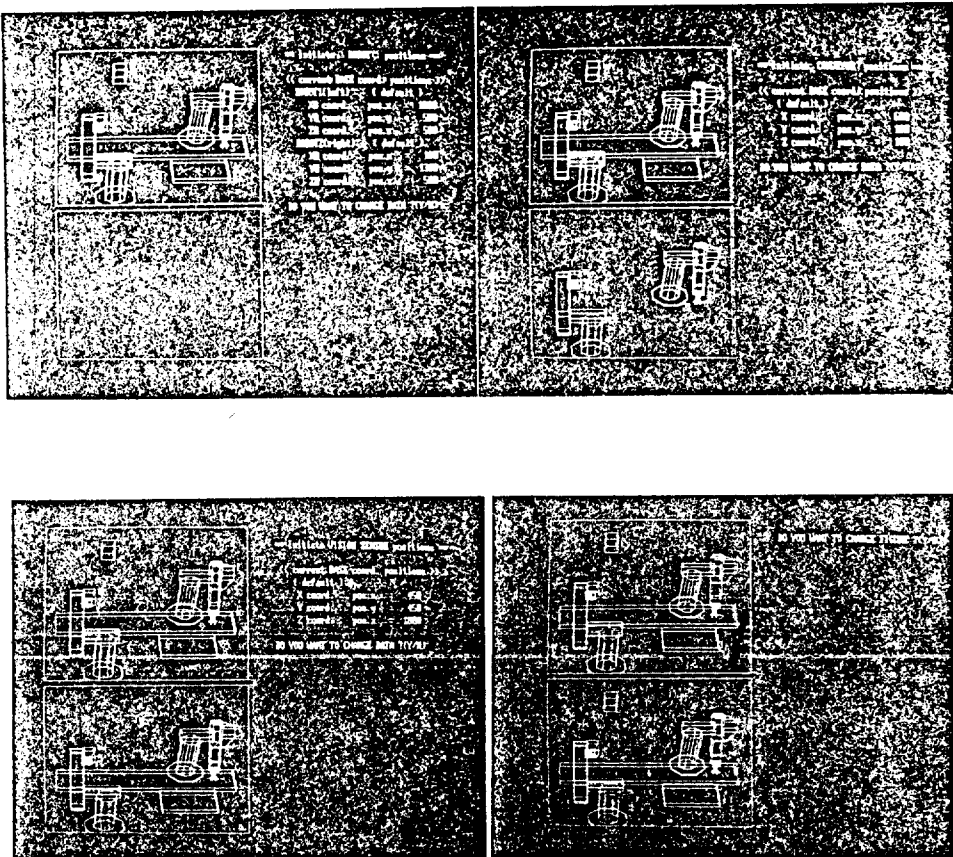
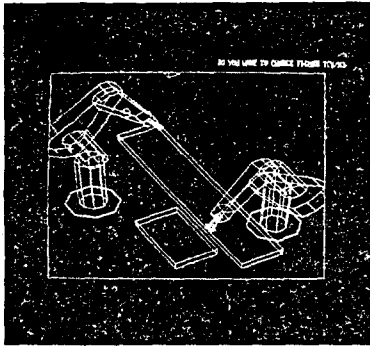
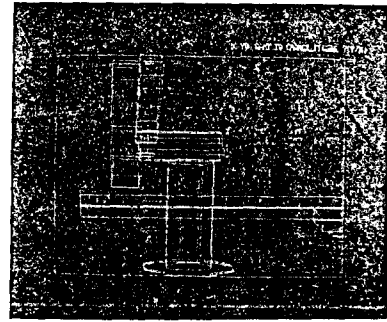


그림 4.1 작업 단위 구성 과정

Fig. 4.1 Organization process of an Assembly Workcell.



(a)



(b)

그림 4.2 스크린 디스플레이
Fig. 4.2 Screen Display.

대상 작업 단위에 대한 관찰점(viewpoint)변화, 관찰영역(window)변화등을 통해 특정 부분을 회전, 확대 시켜 시각적 확인을 돕는 기능으로 그림 4.2는 그 예를 나타낸다.

5. 실험 및 결과

(1) 시뮬레이션 및 성능 평가

시뮬레이션은 우선 시뮬레이션 진행에 의한 작업 단위 변화 모습을 디스플레이시키지 않고 진행되며 시뮬레이션 후 3.1.1절에서 언급된 작업 단위의 성능 평가 척도를 화면에 출력시켜 작업 단위 구성 요소들의 상태 변화에 의한 시스템의 성능 변화를 파악할 수 있게 한다. 그림 5.1은 구성 요

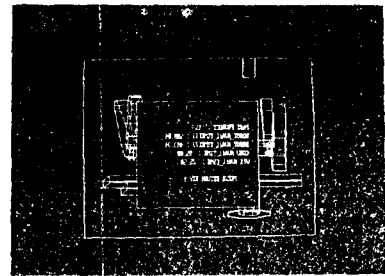


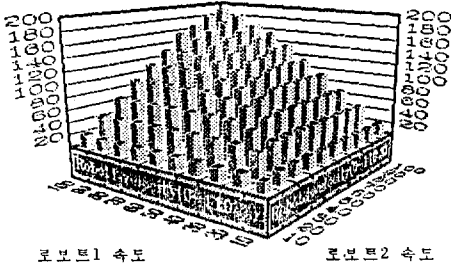
그림 5.1 성능 평가 척도 출력
Fig. 5.1 performance Measure Display.

표 5.1 로봇트 속도 변화에 따른 성능 평가 척도 변화

Table. 5.1 Performance Measure vatiution according to Robot Joing velocity change.

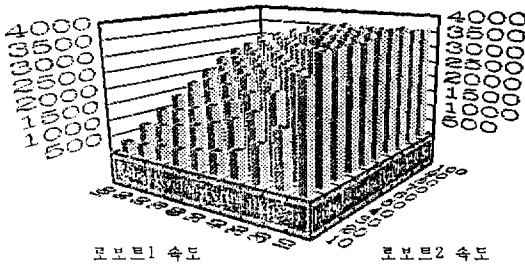
Robot1 속도 백분율	Robot2 속도 백분율	Part product number	Robot1 job time(초)	Robot2 job time(초)	Conveyor move time(초)	Vision processing time(초)
10	10	17	3041.32	2952.32	95.0	28.5
30	30	55	2962.01	3021.37	285.0	85.5
50	50	94	2953.94	3031.56	480.0	144.0
70	70	133	2959.46	3034.69	675.0	202.5
30	40	64	3413.82	2538.65	330.0	99.0
40	30	64	2500.32	3510.75	335.0	100.5
50	70	110	3450.78	2509.84	560.0	168.0
70	50	110	2460.41	3566.36	565.0	168.0

Part Product No. (per 3600 sec)



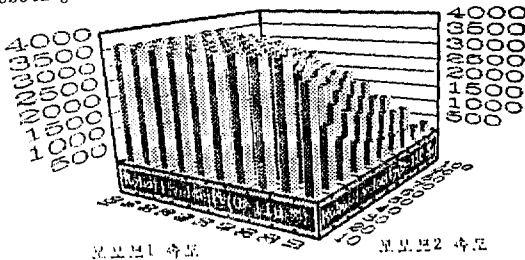
(a) 총 생산량

Robot1 job time (per 3600 sec)



(b) 로봇1 작업 시간

Robot2 job time (per 3600 sec)



(c) 로봇2 작업 시간

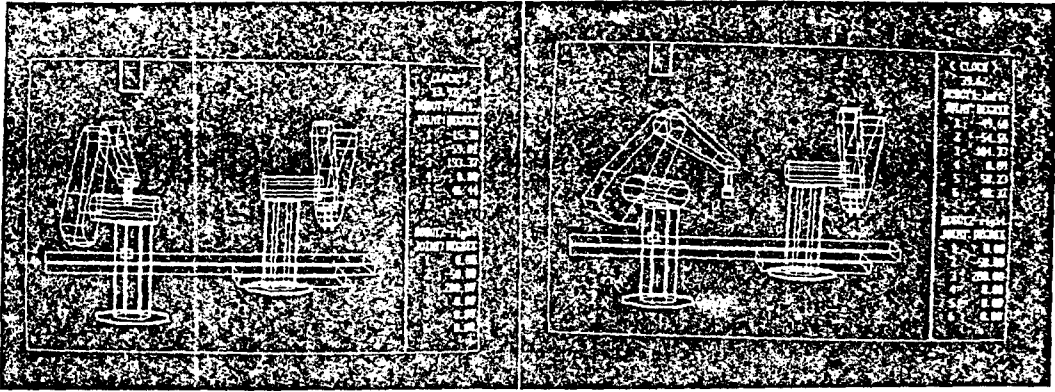
그림 5.2 성능 평가 척도의 결과
Fig. 5.2 Performance Measures.

소들의 상태 변화에 의한 시스템 성능 평가 척도 변화 출력을 나타낸다. 그림 5.1에 나타나는 성능 평가 척도, 즉 파트의 총 생산량, 로봇의 총 작업 시간, 콘베이어의 총 이동 시간, 비전 센서의 총 처리 시간 등의 정보를 이용하여 사용자는 본 시뮬레이터를 이용하여 구성된 작업 단위의 최적의 작업 조건을 판단할 수 있게 된다. 표 5.1은 콘베이어 이동 속도와 비전 처리 시간을 고정시키고 두 로봇의 속도를 변화시켰을 때의 성능 평가 척도들의 변화 결과를 나타낸다.

표 5.1의 조건과 같이 콘베이어의 이동 속도를 50%, 비전 센서의 처리 시간을 1.5초로 고정시키고 두 로봇의 축 속도를 10%에서 100%로 변화시키면서 3600초 동안 조립 작업 단위에 대한 시뮬레이션을 진행했을 때, 변화되는 성능 척도의 결과는 그림 5.2와 같다. 그림 5.2(a)에서 정해진 시간 동안의 작업 처리 결과, 즉 파트 생산량 (part product number)은 두 로봇의 축 속도가 빠를수록 많다는 것을 알 수 있다. 파트 생산량이 두 로봇 중 어느 한 로봇의 축 속도에만 비례하지 않고 두 로봇의 축 속도에 비례하는 것은 두 로봇이 같은 작업을 동시에 수행하는 것이 아니라 순차적인 작업을 서로의 작업 완료 여부에 의존하면서 각 각 수행하기 때문이다. 또한 그림 5.2(b), (c)에서 로봇1과 로봇 2는 축 속도가 느릴수록 정해진 시간 동안에 작업에 관여하는 시간이 많아 진다는 것을 알 수 있다. 이것은 축 속도가 느린 로봇이 작업을 하는 동안 상대적으로 축 속도가 빠른 로봇과 콘베이어, 비전 센서 등이 작업을 마치고 속도가 느린 로봇이 작업을 끝내기를 기다리고 있다는 것을 의미한다. 이와같은 관계는 특히 조립 작업 단위에서 각 자원들의 작업 처리가 서로 독립적이지 아니라 서로 일정한 관련을 갖으면서, 즉 어느 한 자원의 작업 처리 결과가 다른 자원의 작업 처리에 영향을 미치면서 작업이 진행되기 때문이다.

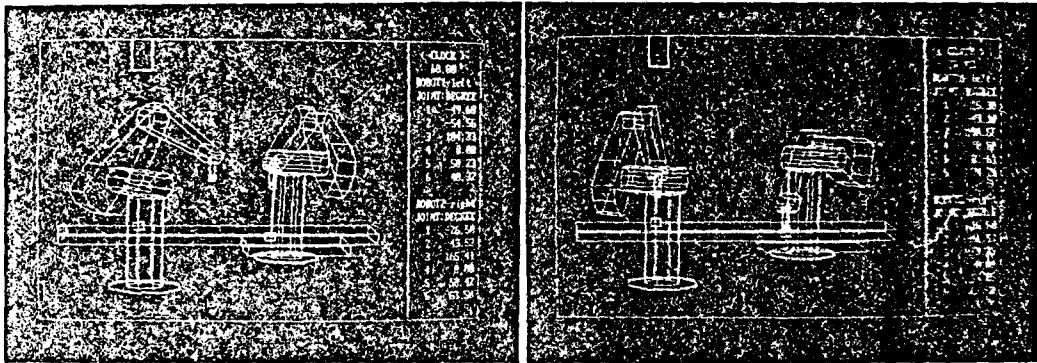
(2) 그래픽 디스플레이

시뮬레이션이 진행되어 성능 평가 척도를 출력시킨 후 시뮬레이션 진행 과정을 그래픽을 이용하여 화면에 디스플레이시키게 된다. 이 때 디스플레이되는 부분 오른쪽에 시뮬레이터 시계의 시간, 즉 시뮬레이션 시간 변화를 출력시키고 이와 함께 두 로봇의 각 축의 값들을 디스플레이시킨다. 그림 5.3는 그래픽을 이용한 시뮬레이션 진행 과정 디스플레이를 나타낸다.



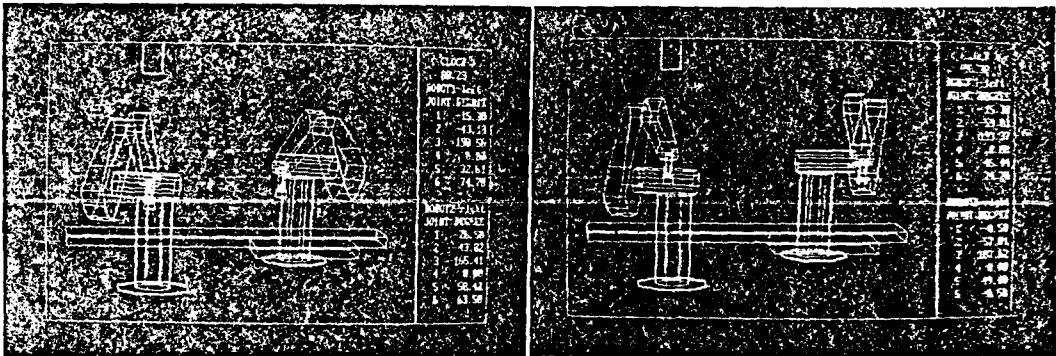
(a)

(b)



(c)

(d)



(e)

(f)

그림 5.3 시뮬레이션 진행 과정 디스플레이
Fig. 5.3 Simulation process Display.

6. 결 론

본 논문에서는 주어진 조립 작업 단위(assembly cell)를 대상으로한 그래픽 시뮬레이터를 구성하였다. 대상 조립작업단위 안에서 진행되는 작업을 기술하고 그 제어 로직을 모델링하기 위하여 페트리 넷(Petri Net)이론이 도입되었고, PN의 구조적 성질을 이용하여 작업단위의 모델이 타당함을 보여주었다. 또한 구조적으로 타당한 모델의 정량적 분석을 위해 발생 사건 처리에 시간 경과 개념을 갖는 사건 스케줄링 방법을 이용하여 시뮬레이션을 진행하였다.

시뮬레이션 과정을 그래픽을 이용하여 나타내기 위하여 로봇트 모델링, 요소의 모델링, 월드모델링 등의 기능들이 포함되었으며, 로봇트는 정해진 모양의 링크들이 축을 기준으로 특수한 형태로 결합되어 이루어진 구조로 인식되어, 링크를 구성하는 요소의 모델링, 링크 모델링, 그리고 축 모델링의 순서로 계층적, 구조적으로 모델링되었다.

위의 방법으로 구성된 시뮬레이터를 이용하여, 조립 작업 단위 안에서 정해진 제어 로직에 의해 작업이 진행되면서 발생하는 시스템의 상태 변화를 시각적으로 나타낼 수 있었을 뿐 아니라, 자원들의 상태 변환에 의한 시스템의 변화도 파악, 비교할 수 있었다.

본 논문에서 개발된 시뮬레이터에 작업 진행 중 발생할지도 모를 두 로봇트 사이의 충돌 가능성, 로봇트와 다른 자원들 사이의 충돌 가능성을 검증하는 기능 추가가 요구된다. 또한 PN모델의 인스턴스 행렬, P인베리언트, 초기 마킹등의 정보에 의해 주어진 모델의 타당성 여부를 검증하는 기능이 추가되면 본 시뮬레이터는 모델링된 작업 단위를 정성적, 정량적으로 분석하는 기능과함께 그래픽 기능을 갖는 시뮬레이터로서 일반 작업단위를 대상으로 한 그래픽 시뮬레이터 개발에 대한 하나의 방향 제시가 될 수 있을 것이다.

참 고 문 헌

[1] 상광제, "로봇트 시뮬레이터", 서울대학교 제어 계측 공학과 특수설계 프로젝트 보고서, 1985.
 [2] 장 원, "컴퓨터 그래픽스를 이용한 로봇트 시뮬레이터의 설계", 한국 과학 기술원 석사 학위 논문, 1986.
 [3] D.K. Pai, M.C. Leu, "INEFFABLE-An Environment for Interactive Computer

Graphic Simulation of Robotic Application", IEEE Int'l Conf. on Rob. and Auto. pp. 897-903, 1986.
 [4] D.W. Wloka, "ROBSIM-A Robot Simulation System", IEEE Int'l Conf on Rob. and Auto. pp. 1859-1864, 1986.
 [5] T. Stata의 2인, "Robot Simulation System as a Task-Programming Tool", 11th ISIR pp. 595-602, 1981.
 [6] A. Baer, C. Eastman and M.Henrion, "Geometric Modelling : a Survey", Computer-Aided Design, September, 1979.
 [7] R. Suri, "Quantitative Technique for robotic Systems Analysis", Handbook of Industrial Robotics(S.Y.Nof ed.) chap. 31, Wiley, 1985.
 [8] Kobayashi, Modelling and Analysis, Addison-Wesey, 1978
 [9] R.F. Garzia, M.R. Garzia, B.P. Zeigler, "Discrete-Event Simulation", IEEE SPEC-TRUM pp. 32-36, DECEMBER 1986.
 [10] Wolfgang Reisig, Petri Nets : an introduction, Springer Verlag, 1985.
 [11] Sifakis, "Structrual properties of Petri Nets", LNCS vo 164., 1978.
 [12] M. Kamath, N. Viswanadham, "Application of Petri net Based Models in the Modelling and Analysis of Flexible Manufacturing Systems," IEEE Int'l Conf. on Rob. and Auto. pp. 312-317, 1986.
 [13] N. Viswanadham, Y. Narahari, "Coloured Petri Net Models for Auto-mated Manufacturing Systems", IEEE Int'l Conf. on Rob. and Auto. pp. 1985-1990, 1987.
 [14] C.L. Beck, B.H. Krogh, "Models for Simulation and Dicrete Control of Manufacturing Systems," IEEE Int'l Conf. on Rob and Auto pp. 305-310 1986.
 [15] 황희수, 김현기, 우광방, "자동화된 생산 시스템의 유연한 제어 구조의 모델링과 시뮬레이션," 대한 전기학회 창립 40주년 기념 학술대회, 1987.
 [16] B.H. Krogh, R.S. Sreenivas, "Essentially Decision Free Petri Nets for Real time Resource allocation", IEEE Int'l Conf. on Rob and Auto. 1987.
 [17] Y.Kakazh의 2인 "A Sturdy on Cooperative

- Motion Simulator,” 15th ISIR pp.691-698, 1985.
- [18] M. Vukobratovic, M. Kircanski, kinematics and Trajectory Synthesis of Manipulation Robots, Scientific Fundamentals of Robotics 3. Springer-Verlag, 1986.
- [19] R.H. Taylor, “Planning and Execution of Straight Line manipulator Trajectory”, Tutorial on Robotics, 1983.
- [20] 김 대원, 고 명삼, 이 범희, “지식 베이스를 이용한 로봇틱 조립 셀의 모델링과 운영 관리를 위한 프레임 워크”, '88한국 자동 제어 학술 회의, pp.374-379, 1988.
-