특 집

다중로봇의 성능평가를 위한 시뮬레이션 기법

A Simulation Technique for the Performance Evaluation of the Multi-Robot Interconnection Systems(MRIS)

李 起 東, 李 範 熙

嶺南大學校 電算工學科
서울大學校 制御計測工學科

요 약

생산시스템 분야에서의 다중로봇시스템에 대한 필요성이 증가되고 있기 때문에, 이에 대한 시스템 성능분석이 우선적으로 필요하다. 본 논문에서는 다중로봇시스템의 성능평가를 위한 컴퓨터 시뮬레이션 방법을 제시한다. 먼저 오류회복 기능만을 고려한 다중로봇시스템에 대하여 여러 가지의 모델링 방법 중에서 확장 페트리 네트 모델을 기초로 하여 컴퓨터 시뮬레이션 하는 과정을 제시하고 제시된 시뮬레이션 방법의 유효성을 입증하기 위하여 부가적으로 큐잉모델을 기초로 한 해석적 수식을 유도하여 두 가지의 결과를 비교해 본다. 그 다음, 오류회복 기능에 덧붙여서 충돌회피 기능을 고려한 다중로봇시스템은 해석적 수식을 유도하기 위해서는 강력한 가정들이 필요하며 이러한 가정들 때문에 실제 시스템을 정확히 모델에 반영하기 어려워진다. 따라서, 그 유효성이 입증된 컴퓨터 시뮬레이션을 이용하면 필요한 최적의 운용 변수를 용이하게 선택할 수 있을 것이다.

Abstract

Since there are increasing demands for multi-robot interconnection systems(MRIS) in industrial manufacturing system, the performance evaluation of the MRIS is first needed. This thesis presents a computer simulation technique for the performance evaluation of the MRIS. First, we consider a error recoverable MRIS. We adopt the extended Petri net model as a computer simulation model that allows an easy evaluation of the performance. To verify the significance of the proposed computer simulation method, mathematical analysis, which is based on the given queueing model, is carried out with some design issues for the MRIS. In addition to

this, it is required to analyze the MRIS considering collision avoidance as well as error recovery. In this case, mathematical analysis needs hard assumptions which are the constraints for the precise description of real environment. Thus, we present a computer simulation model and its results suggest an optimal operational strategy for the MRIS under given conditions.

I. 서 론

최근의 생산시스템 분야에서는 다품종 소량 생산에 대한 요구가 확산되고 있다. 이와 같은 생산형태로의 전환을 위하여 생산설비의 자동화, 고속화, 유연화가 필수적이라 할 수 있다. 이를 위한 요소 기술로 이용되고 있는 로봇작업은 현재 픽-플레이스(pick-and-place)작업 등과 같은 독립적이고 단순 반복적인 작업에 사용되고 있어 그 효용성을 충분히 이용하지 못하고 있는 실정이다. 따라서 두 대 이상의 로봇을 시스템화하여 사용하는 다중로봇시스템의 이용이 필수적이라 할 수 있으며 이를 위해서는 다중로봇시스템의 성능분석이 선행되어야 한다. 다중로봇시스템의 장점중의 하나는 오류회복(error recovery) 기능이며, 단점이라면 충돌회피(collision avoidance) 문제이다. 이러한 시스템을 분석하기 위하여 우리는 컴퓨터를 이용한 시뮬레이션 기법을 이용한다.

시뮬레이션에 대해서는 다양한 정의가 있을 수 있으나 이를 종합·정리한다면, 실제의 어떤 시스템 또는 순차적인 일의 어느 시간 동안의 가동 상태를 흉내내는 것이라 할 수 있다. 시뮬레이션을 하는 이유로는 어떤 조사 대상 시스템의 가동 상황이나 움직임을 인공적으로 만들어 관측해 봄으로써 실제 현실에서 그 상황이 일어났을 때 그 시스템이 일으킬 여러 가지 현상들을 유추해 보고자 하는 것을 들 수 있다. 이렇게 함으로써 실제 현실에서 가동시 일어날수 있는 피해나 사고를 예방하고 또 많은 비용이 드는 시행착오를 거치지 않고 잘못된 점을 수정할 기회를 가질 수 있게 되는 것이다.

지금까지 우리가 사용하고 있는 문제 해결 방법들을 정리해보면 대체적으로 초기에는 그래프나 그림 등을 그려서 풀어나가는 방법에서, 수학적인 해석적 방법으로 발전해 나갔음을 알 수 있다. 그러나 해석적인 방법은 해석적으로 풀 수 있는 방법론상의 제약으로 인하여 현실 문제를 가능한 한 그때에 개발되어 있는 방법으로 풀 수 있도록 간략화하여 모델을 만들어야 한다는 제약이 있다. 이것은 많은 가정들을 도입하여 현실을 간략화해야만 하는 방법으로써 모델화 한 것을 해석하여 최적해를 구할 수는 있으나, 구한 최적해가 모델화한 것에만 맞는 것이지 실제 그 대상인 현실 문제에는 잘 맞지 않는다는 결점을 가지고 있다. 이것과는 반대 개념은 발견적 기법 혹은 휴리스틱(heuristic)이라고 부르는 부류에 속하는 것들이다. 여기에 속하는 방법들의 특징은 모형화 할 때에는 현실에 아주 가깝게 되지만 답을 찾는데는 모형의 복잡성 때문에 많은 어려움이 있다는 점이다. 이 방법은 논리적인 체계성이 없어 프로그램상 많은 문제가 있고 또한 비용이 많이 든다는 단점이 있다. 따라서, 실제 상황을 가장 잘 표현할 수 있는 논리적인 모델링 방법을 선정하여 그것을 기초로 하여 컴퓨터 시뮬레이션을 수행하는 것이 더 타당하다고 할 수 있다.[1]

시뮬레이션 대상시스템은 크게 나누어 연속형(continuous)과 이산형(discrete) 두 가지로 분류해 볼 수 있다. 연속형은 시스템의 변화가 연속적으로 일어나는 것이고 이산형은 시스템이 특정한 사건(event)을 기준으로 변화하는 것이다. 대상시스템에 대한 모델을 선정할 때 고려해야 하는 모델의 종류에는 여러 가지가 있다. 먼저 실체적 모델(physical model)과 상징적 모델(symbolic model)로 구분할 수 있다. 실체적 모델은 물체를 사용하여 어떤 사실을 표현한 것이고, 상징적 모델은 기호나 그림을 사용하여 표현한 것이다. 또한, 시간 개념이 포함된 동적 모델(dynamic model)과 포함되지 않은 정적모델(static model)로 구분할 수 있고, 모델 내에서 사용되는 변수 값의 종류에 따라 확정적 모델(deterministic model)과 확률적 모델(stochastic model)로 구분할 수 있다.

다중로봇시스템도 컴퓨터망, 통신망, 교통시스템 등과 같이 이산사건에 의해 상태가 변하는 이산 현상 시스템(discrete event dynamic system : DEDS)으로 볼 수 있으며 모델의 종류로 볼 때 상징적 모델이며 시간개념이 포함된 동적 모델이며, 변수 값이 확률적인 확률적 모델이다. 이산 현상 시스템 성능평가를 위한 모델링 방법은 여러 가지가 있으나 Ho[2]는 시스템 모델링 방법을 시간적, 확률적, 논리적 성질에 따라 다음의 표 1과 같이 분류하였다. 이중에서 큐잉 모델은 그 쓰임새가 오래되었고, 또 다양하다. 그러나 기본조건(지수적 서비스, 포아송적 도착, 무한개의 버퍼 등)이 만족될 경우에는 효율적으로 모델링 하여 분석할 수 있으나, 그렇지 않은 이산 현상 시스템의 경우에는 모델링 하여 분석하기 힘들다. 또한 마르코프 연쇄(Markov chain)는 상당히 유용한 결과들이 밝혀져 있으나, 상태공간이 기하급수적으로 증가하고, 모델을 수정하기 힘들다는 단점이 있다. 이에 반하여 페트리 네트 모델링 기법은 동기화된 사건, 논리적 선후관계, 구조적 상관관계를 가진 이산 현상 시스템의 모델링에 적합하다. 현재까지 이용되고 있는 페트리 네트는 매우 다양하다. 따라서 이를 분류하는 방법도 각자의 관점에 따라 여러 가지가 있을 수 있으나 일반적으로, 1)고전적 의미의 페트리 네트(classical Petri net), 2)일반화된 페트리 네트(generalized Petri net), 3)칼라 페트리 네트(colored Petri net), 4)시간 페트리 네트(timed Petri net), 5)확률 페트리 네트(stochastic Petri net), 그리고 6)확장 페트리 네트(extended Petri net) 등으로 나눌 수 있다. 이중에서 실 시스템의 다양성을 표현하기에는 확장 페트리 네트가 적합하다.

본 논문에서는 여러 가지 이산형 모델링 기법 중에서 실제 상황을 가장 잘 모사할 수 있는 페트리 네트 모델을 채택하여 다중로봇시스템의 오류회복 기능과 충돌회피 문제를 모델링 하여 실제의 시뮬레이션 자료를 이용하여 시스템을 분석해 본다. 더불어 시뮬레이션의 유효성을 입증하기 위하여 큐잉 모델을 기초로 한 수학적인 모델 및 그 수식을 유도하여 그 결과를 시뮬레이션 결과와 비교해 본다.

〈표 1〉 이산 현상 시스템의 모델링 방법과 그 특성

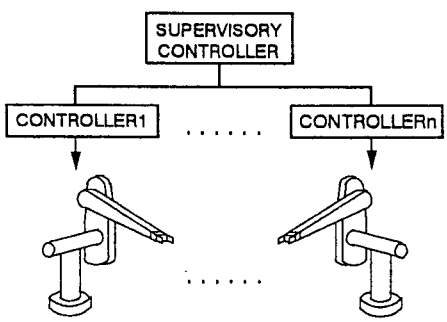
모델링 방법	성 질		
Automata/State Model Finite State Machine Extended State Machine Petri Net	untimed	deterministic	logic
Communicating Sequential Process Finitely Recursive Process	untimed	deterministic	algebraic
Markov Chain Queueing Network Generalized Semi Markov Process Generalized Stochastic Petri Net	timed	stochastic	performance

II. 다중로봇시스템의 구성 및 모델링

본 논문에서 분석하고자 하는 대상 시스템인 다중로봇시스템의 구성 형태와 그것에 대한 모델을 제시한다. 모델은 두 가지를 제시하는데 그 중에서 페트리 네트 모델은 컴퓨터 시뮬레이션용으로 사용하고 큐잉모델은 수학적인 분석을 위하여 사용한다.

1. 시스템 구성

다중로봇시스템은 아래 그림 1과 같이 각각의 로봇이 자신의 프로세서를 가지고 있으므로 로봇을 독립적으로 동작시킬 수 있고, 프로세서들을 통합 제어하는 주 프로세서를 가진 계층적 구조이다. 프로세서간의 통신은 주 프로세서가 총괄하며 주 프로세서에서 부 프로세서로 명령을 이송할 경우에는 전송시간을 고려하고, 부 프로세서에서 주 프로세서로 명령을 이송할 경우에는 역전송 시간을 고려한다. 전송 시간($f(C)$) 및 역전송 시간($g(C)$)은 Reed[3] 등이 사용한 것과 같이 전송할 명령의 개수에 비례하고 기본적으로 데이터 전송에 필요한 시간(startup time)을 첨가한 형태로 가정하여 일차함수인 다음 식(1)과 같이 나타낼 수 있다. 여기에서 k_1, k_2는 전송 상수이고, k_3, k_4는 역전송 상수이며 C는 전송할 명령의 개수이다.



〈그림 1〉 시스템 구성

$$f(C)=k_1 \cdot C+k_2$$
$$g(C)=k_3 \cdot C+k_4 \quad (1)$$

Tsai[4]는 다중 로봇의 운동을 3가지로 나누었는데 1)동기운동(simultaneous motion), 2)협조운동(coordinated motion), 3)독립운동(overlap motion) 등이 그것들이다. 동기운동 및 협조운동은 로봇 한대로는 수행할 수 없거나 또는 가능하더라도 작업 절차가 복잡해지고 부수적인 환경이 요구되는 복잡한 작업들을 수행 가능케 하여 평균 실행 완료 시간을 단축시킬 수 있다. 독립운동은 여러 대의 로봇이 독립적으로 자신에게 할당된 명령들을 병렬식으로 수행하기 때문에 생산성이 증가된다. 본 논문에서는 가장 간단한 형태인 독립운동에 대하여 오류회복 기능과 충돌회피 기능을 가진 다중로봇시스템을 분석해 본다. 오류회복 방법[5]은 부하 분배, 개체 교체, 개체 수리 등 여러 가지가 있으나 본 논문에서는 한 로봇에서 오류가 생겼을 경우에 그 나머지 작업들을 남은 로봇들이 분담해서 처리하는 부하 분배 방법을 채택하며, 공동 작업영역 상호배제라는 충돌회피 방법을 채택하기로 한다.

2. 시스템 모델링

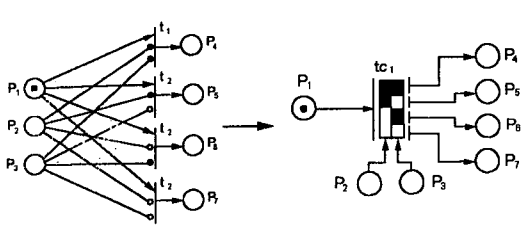
서론에서 제시한 여러 가지 모델링 기법 중에서 시뮬레이션 모델로써 페트리 네트 모델을 채택하여 다중로봇시스템의 오류회복 기능과 충돌회피 기능을 모델링 하여 분석해 본다. 더불어 시뮬레이션의 유효성을 입증하기 위하여 큐잉모델을 기초로 한 수학적인 모델 및 그 수식을 유도하여 본다. 한편, 시뮬레이션 모델은 복잡한 시스템을 현실과 근접하게 모델링할 수 있으나 이에 대한 수식적 모델링은 상당히 힘들뿐만 아니라 여러 가지 가정이 필요하게 된다. 본 논문에서는 오류회복 기능을 가지는 다중로봇시스템에 대해서는 시뮬레이션 모델과 수식적 모델을 동시에 제시하여 비교하고, 오류회복기능과 충돌회피 기능을 동시에 고려한 시스템에 대해서는 그 시스템의 복잡성 때문에 해석적 수식을 제시하기가 매우 힘들다. 따라서 시뮬레이션 모델만을 제시한다. 모델링 위해서는 필연적으로 여러 가지의 가정이 필요하게 되나 본 연구에서는 로봇의 명령 실행 시간에 대한 가정 즉, 실행시간은 지수 분포를 가진 독립 확률 변수(exponentially distributed independent random variable)로 가정하였다. 그 이유는 본 연구에서 취급하는 다중로봇시스템을 비(非)－마르코프 큐잉 시스템(non-Markovian queueing system : G/M/1, M/G/1, G/G/1)으로 생각하지 않고 마르코프 큐잉 시스템(Markovian queueing system : M/M/1 또는 M/M/m)으로 생각하여 수학적 분석결과를 유도해내기 위함이다. 이와 같이 해석적 수식을 유도해내려면 강한 가정들이 필요하게 된다. 이러한 마

르코프 성질을 가지려면 상태 시간(state time)은 반드시 지수 분포이어야 한다.[6] 따라서 본 논문에서는 각각의 로봇 운동이 서로 종속적이 아니므로 그것들의 상태 시간의 확률 분포 함수(probability distribution function)는 서로 독립적이며 지수분포를 가진다고 가정할 수 있다. 본 논문에서 고려하는 성능평가 지표는 평균실행완료 시간($\overline{T}$) 과 실행실패 확률(PS), 실행실패하여 남을 명령 개수의 기대값($\overline{PS}$) 등이다.

1) 시뮬레이션 모델

서론에서 제시한 여러 가지 모델링 방법 중에서 이산 현상을 정확하게 표현할 수 있고, 내부 변수들을 쉽게 관찰할 수 있는 페트리 네트를 다중로봇 시스템의 시뮬레이션 모델로 사용한다. 실제 시스템을 모델링하기 이전에 먼저 본 논문에서 제안한 새로운 형태의 확장 페트리 네트를 살펴본다. 그 다음, 오류회복 기능을 가진 다중로봇시스템의 모델링과 여기에다 충돌회피까지 고려한 경우의 모델링을 한다. 본 논문에서 사용한 확장 페트리 네트에서 특이한 것은 조건 트랜지션과 확률 트랜지션인데 그 내용은 다음과 같다.

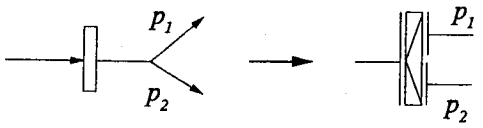
조건(conditional) 트랜지션 : 금지아크(inhibitor arc)와 구동아크(activator arc)를 사용하는 복잡한 네트를 간결하게 그리고 논리를 이해하기 쉽게 하기 위하여 도입한 트랜지션이다. 금지아크나 구동아크의 입력 플레이스들이 조건 트랜지션의 조건 플레이스가 된다. 물론 점화 이후에도 조건 플레이스에서의 토큰 흐름은 없다. 조건 트랜지션 안의 사각형 상자는 조건 플레이스들의 토큰 상황을 나타낸다. 즉, 검은 색은 조건 플레이스에 토큰이 있는 상태를 나타내고, 흰색은 토큰이 없는 상태를 의미한다. 그림 2에서 보는 바와 같이 조건



〈그림 2〉 조건 플레이스가 두 개인 조건 트랜지션

플레이스가 두개일 경우에는 트랜지션이 3개(4개⇒1개) 줄어지고, 아크는 9개(16개⇒7개) 줄어짐을 알 수 있다.

확률 스위치(probability switch) 트랜지션 : 스위치 플레이스에서 필요 없는 트랜지션의 첨가를 피하기 위하여 사용한 트랜지션이다. 그림 3에서 보는 바와 같이 점화가 일어날 때마다 주어진 확률값에 따라 분기가 된다.



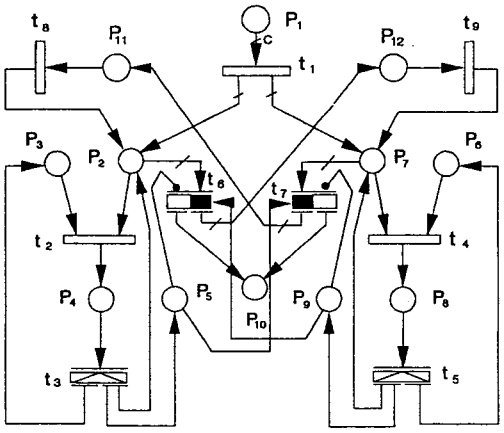
〈그림 3〉 확률 스위치 트랜지션. $p_1+p_2=1$

본 논문에서 사용하는 페트리 네트의 활성화 법칙(enabling rule)은 주어진 트랜지션 t_j의 사전조건(precondition)인 모든 플레이스에 대하여 현재의 토큰의 상태가 입력 함수값 보다 커야 되고 그리고 주어진 트랜지션의 사후 조건(postcondition)인 모든 플레이스에 대하여 현재의 토큰의 상태가 0일 때 활성화 상태로 본다. 이것을 수식으로 나타내면 아래의 식(2)와 같다. 또한 페트리 네트의 점화 법칙(firing rule)은 트랜지션 t_j가 활성화 상태일 때 점화가 가능하고 점화 결과는 주어진 트랜지션 t_j의 입력 플레이스에서 토큰을 제거하고 주어진 트랜지션 t_j의 출력 플레이스에 토큰을 새로 할당한다. 새로운 토큰의 상태(M')는 아래의 식(3)과 같이 나타낼 수 있다. 이러한 확장 페트리 네트를 이용하여 먼저 오류회복 기능만을 고려한 다중로봇시스템의 모델링 방법을 제시하고 그 다음 오류회복 기능뿐만 아니라 충돌회피 기능까지 고려한 다중로봇시스템의 모델링 방법을 제시한다.

$$M(p_i) \geq I(p_i, t_j)\ \forall p_i \in {}^* t_j,$$
$$\text{그리고 } M(p_i) = 0\ \forall p_i \in t_j^* \quad (2)$$
$$M'(p_i) = M(p_i) + O(p_i, t_j) - I(p_i, t_j)\ \forall p_i \in P \quad (3)$$

(가) 오류회복 기능을 가진 다중로봇시스템 모델링

다중로봇시스템의 한 특징인 오류회복 기능을 위주로 앞에서 제시한 시스템에 대하여 모델링을 한다. 두 로봇이 독립적으로 동작하며 명령들을 할당하거나 전송할 때 걸리는 시간을 고려하기 위하여 시간 트랜지션 t_8, t_9를 도입한다. 즉, 로봇1에서 오류가 발생하여 나머지 명령들을 로봇2로 전송해야 할 경우 그때의 전송시간을 나타내기 위하여 t_8을 도입하고, 그 반대의 경우를 위하여 t_8를 도입한다. 여기에서 시간 트랜지션 t_8, t_8의 지연 시간 분포는 확률적인 값이 아닌 확정적인 값으로 식 (1)에 나타낸 것과 같이 토큰수(명령수)와 전송상수에 의해 결정된다. 따라서 트랜지션 t_8은 역전송 지연(로봇2제어기⇒주프로세서)과 전송 지연(주프로세서⇒로봇1제어기)시간의 합을 의미한다. 그림 4에 다중로봇시스템의 확장 페트리 네트 모델이 나타나 있다.

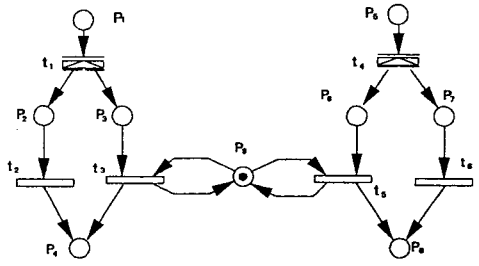


p_1 : 명령 수행 준비상태(C개의 토큰)
p_2 : 명령 버퍼1
p_3 : 로봇1 준비상태(정상상태)
p_4 : 로봇1 명령 한 개 수행완료
p_5 : 로봇1 오류상태
p_6 : 로봇2 준비상태(정상상태)
p_7 : 명령 버퍼2
p_8 : 로봇2 명령 한 개 수행완료
p_9 : 로봇2 오류상태
p_{10} : 실행실패 상태
p_{11} : 로봇1 정상상태 & 로봇2 오류상태
p_{12} : 로봇2 정상상태 & 로봇1 오류상태
t_1 : 확률 스위치 트랜지션1
t_2 : 로봇1 작업 동작
t_3 : 확률 스위치 트랜지션2
t_4 : 로봇2 작업 동작
t_5 : 확률 스위치 트랜지션3
t_6 : 로봇 오류 확인1
t_7 : 로봇 오류 확인2
t_8 : 명령 전송(로봇2⇒로봇1)
t_9 : 명령 전송(로봇1⇒로봇2)
초기상태 M_0={C,0,1,0,0,1,0,0,0,0,0,0}

〈그림 4〉 다중로봇시스템의 확장 페트리 네트

(나) 오류회복 기능과 충돌회피 기능을 가진 시스템 모델링

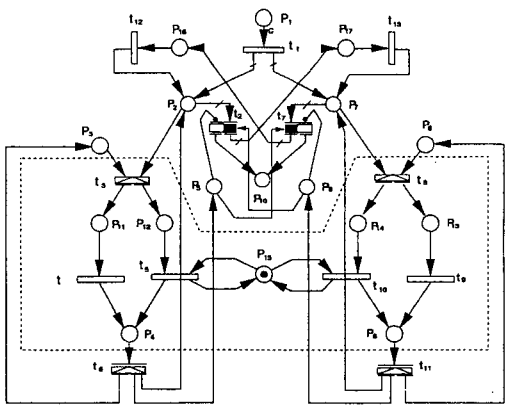
두 대의 로봇이 동시에 구동될 때 각 로봇들의 작업영역이 겹치지 않으면 로봇들 사이의 충돌은 생기지 않는다. 그러나 중복되는 작업영역이 있다면 충돌이 생길 소지가 있다. 따라서 충돌회피 문제가 발생하게 되는데 일반적으로 충돌회피 방법은 여러 가지가 제안되어 있으나 여기에서는 로봇의 특정 작업을 대상으로 분석하는 것이 아니므로 한 로봇이 공동 작업영역을 통과하는 작업을 하고 있을 경우에는 다른 로봇이 작업을 중단함으로써 충돌을 근원적으로 회피한다. 이것을 확장 페트리 네트로 표현하면 다음 그림 5와 같다. 그림 5의



p_1 : 로봇1 준비상태(정상상태)
p_2 : 로봇1 독립영역 작업
p_3 : 로봇1 공동영역 작업

p_4 : 로봇1 명령 한 개 수행 완료
p_5 : 로봇2 준비상태(정상상태)
p_6 : 로봇2 독립영역 작업
p_7 : 로봇2 공동영역 작업
p_8 : 로봇2 명령 한 개 수행 완료
p_9 : 작업영역 점유 상태
t_1 : 확률 스위치 트랜지션1
t_2 : 로봇1 독립영역 작업 동작
t_3 : 로봇1 공동영역 작업 동작
t_4 : 확률 스위치 트랜지션2
t_5 : 로봇2 독립영역 작업 동작
t_6 : 로봇2 공동영역 작업 동작
초기 상태 $M_0=\{1,0,0,0,1,0,0,0,1\}$

〈그림 5〉 충돌회피에 대한 페트리 네트 모델



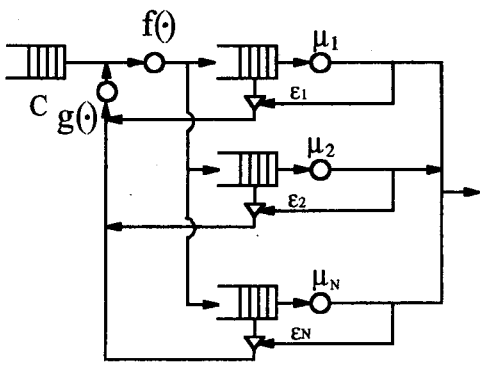
p_1 – p_{10}는 그림 4의 정의와 동일
p_{11} : 로봇1 독립영역 작업
p_{12} : 로봇1 공동영역 작업
p_{13} : 로봇2 독립영역 작업
p_{14} : 로봇2 공동영역 작업
p_{15} : 작업영역 점유 상태
p_{16} : 로봇2 정상상태 & 로봇1 실패상태
p_{17} : 로봇1 정상상태 & 로봇2 실패상태
t_1 : 명령 할당
t_2 : 조건 트랜지션1
t_3 : 확률 스위치 트랜지션1 초기상태
t_4 : 로봇1 독립영역 작업 동작
t_5 : 로봇1 공동영역 작업 동작
t_6 : 확률 스위치 트랜지션2
t_7 : 조건 트랜지션2
t_8 : 확률 스위치 트랜지션3
t_9 : 로봇2 독립영역 작업 동작
t_{10} : 로봇2 공동영역 작업 동작
t_{11} : 확률 스위치 트랜지션4
t_{12} : 명령 전송(로봇2⇒로봇1)
t_{13} : 명령 전송(로봇1⇒로봇2)

〈그림 6〉 충돌회피를 고려한 확장 페트리 네트 모델

확장 페트리 네트를 앞에서 제시한 다중로봇시스템에 적용한 것이 아래 그림 6에서 보여주고 있다. 여기에서 확률 스위치 트랜지션 t_1과 t_4의 확률은 두 로봇의 공동 작업영역의 비율에 따라 결정된다.

(2) 해석적 모델

이산 현상 시스템의 성능평가를 위한 모델링 기법 중에서 큐잉 모델 기법을 기초로 하여 수식적 모델을 유도해 본다. 다중로봇시스템에 대한 큐잉 모델은 아래 그림 7과 같고 이것을 기초로 유도된 수식은 아래 식(4)–(6)과 같으며 그 유도과정 및 정의는 많은 분량 관계로 생략하니 이기동[7] 등의 논문을 참조하기 바란다.



〈그림 7〉 다중로봇시스템의 큐잉 모델

전체 평균 실행완료 시간($\overline{T}(X,Y,Z)$)은 식(4)와 같이 반복형태(recursive form)로 표현되고,

$$\overline{T}(X,Y,Z)) = \frac{P_{no}}{P_{no} + \sum_{i=1}^{N} P_i} T_{no} + \sum_{k=1}^{N} \frac{P_k}{P_{no} + \sum_{i=1}^{N} P_i}$$

$$T_k + f(C) + \overline{T}_{sch} \quad (4)$$

실행실패가 일어날 확률($PS(X,Y,X)$)과 실행 실패로 인하여 실행 완료하지 못할 명령수의 기대값($\overline{CR}(X,Y,X)$)은 식(5)와 식(6)을 이용하여 구할 수 있다.

$$PS(X,Y,X)=\sum_{k=1}^{N}\frac{\varepsilon_k}{1-\varepsilon_k}\sum_{i=1}^{C_k}\left[\prod_{j=1}^{N}(1-\varepsilon_j)^{c_{kj}(i)}\cdot PS(X_k, Y_k, Z_k(i))\right] \quad (5)$$

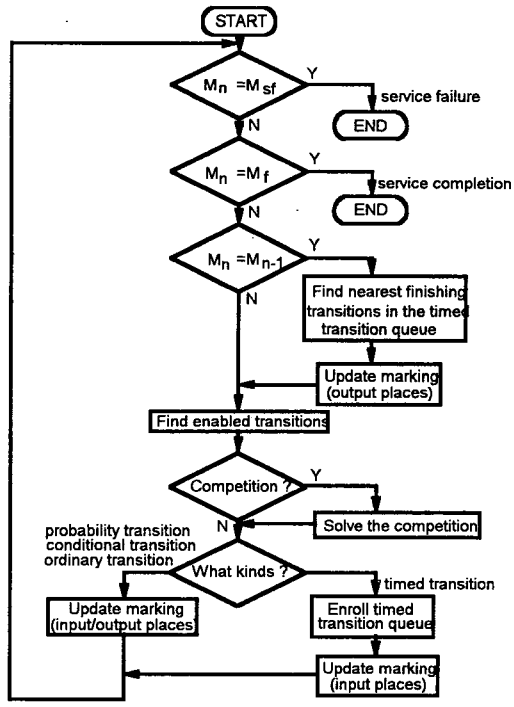
$$\overline{CR}(X,Y,Z)=\sum_{k=1}^{N}\frac{\varepsilon_k}{1-\varepsilon_k}\sum_{i=1}^{C_k}\left[\prod_{j=1}^{N}(1-\varepsilon_j)^{c_{kj}(i)}\cdot PS(X_k, Y_k, Z_k(i))\cdot\overline{CR}(X_k, Y_k, Z_k(i))\right] \quad (6)$$

III. 시뮬레이션

앞 장에서 제시한 페트리 네트 모델을 기초로 하여 실제적인 결과를 살펴보고 분석해 본다. 일반적인 컴퓨터 시뮬레이션의 절차를 Gordon,[8] Law와 Kelton,[9] 그리고 Banks[10] 등의 내용을 바탕으로 간략히 정리하면 다음과 같다. 먼저 대상시스템을 명확히 이해하고 그것을 모델화해야 한다. 모델이 선정되면 해당 모델을 프로그램으로 작성해야 하는데 프로그램 작성시 사용할 수 있는 언어로는 여러 가지가 있다. 일반 목적의 고급언어인 FORTRAN, C 등을 쓸 수도 있고 시뮬레이션 전용언어인 C-SIM, GPSS, SLAM, SIMSCRIPT, SIMAN 등을 사용할 수도 있는데 직접 작성할 경우에는 많은 노력과 경비가 요구된다는 점이 단점이지만 반면에 필요한 내부 관찰 변수를 임의로 선정할 수 있다는 장점이 있다. 본 연구에서는 전용 시뮬레이션언어를 이용하지 않고 직접 C 언어로 작성하여 사용하였다. 시뮬레이션의 그 다음 절차로는 작성된 프로그램이 의도한대로 만들어졌는가에 대한 입증 작업(정확성 검증)이 필요하고 모델이 시뮬레이션 대상시스템을 정확히 표현(모델의 타당성 확인)하고 있는지를 확인해야 한다. 본 연구에서는 정확성 검증과 모델의 타당성 검증을 위하여 부가적으로 해석적인 수식을 유도하여 그 결과를 상호 비교하도록 하였다.

앞장에서 제시한 확장 페트리 네트 모델을 기초로 한 시뮬레이션 프로그램의 작성 방법은 여러 가지가 있을 수 있으나 본 논문에서 사용한 방법의 흐름도는 아래 그림 8에 나타나 있다.

실제 다중로봇 시스템에 대한 시뮬레이션을 할 경우에 제일 중요한 요소는 로봇의 실행 시간을 나타내는 변수의 확률 분포이다. 실행시간에 대한 확률분포를 정확히 얻어내기 위해서는 실제 시스템을 통하여 자료수집을 해야한다. 이렇게 수집된 자료를 이용하여 적정한 분포를 선정한다. 선정된 분포가 적정한지를 알아보기 위하여 적합도검정(goodness-of-fit test)을 하게 된다. 적합도 검정에는 χ^2 검정(Chi-square test)과 K-S 검정



〈그림 8〉 페트리 네트 모델의 시뮬레이션 프로그램 흐름도

(Kolmogorov-Smirnov test) 등이 있는데 그 자세한 내용은 [1]을 참조하기 바란다. 선정할 수 있는 확률분포의 종류도 여러 가지가 있다. 포아송분포(Poisson distribution), 감마분포(gamma distribution), 베타분포(beta distribution), 일양분포(uniform distribution), 기하분포(geometric distribution), 이항분포(binomial distribution), 정규분포(normal distribution), 지수분포(exponential distribution) 등이 있는데 본 연구에서는 실행시간의 확률분포를 지수분포로 선정하였다.

그 다음 선정된 확률분포로부터 실제 값을 발생시키는 방법에는 크게 역변환법(inverse transform method)과 기각법 등이 있다.

(1) 역변환법 : 확률밀도함수(probability density function)가 $f(x)$인 어떤 통계적 모집단으로부터 임의의 확률변수 X_i를 발생하고자 할 경우에는, 우산 누적분포함수(probability distribution function) $F(x)$를 구한다. 그러면 이 $F(x)$는 일양분포를 가지게 된다. 따라서 구하고자하는 값인 x는 역함수를 이용하여 구한다.

(2) 기각법 : 만일 역변환함수에 의해서 기술하기가 어렵거나 불가능한 경우에는 기각법에 의하여 처리하거나 혹은 확률변수의 성질을 이용하여 쉽게 발생시킬 수 있는 변수들을 이용하면 가능하다. 그러나 임의의 변수 발생시 기장 기본적인 것은 난수 발생부분이며 좋은 난수를 발생시키지 못하면 좋은 변수 발생이 불가능하다.

단계1 : $f(x)$의 범위를 상수 c를 이용하여 아래 식(7)과 같이 표준화시킨다.

$$c \cdot f(x) \leq 1, a \leq x \leq b \qquad (7)$$

단계2 : $x=a+(b-a)\cdot r$로 x를 정의한다.

단계3 : 서로 독립인 한 쌍의 난수 r_1, r_2를 생성한다.

단계4 : r_1을 이용하여 x_0를 구한다. 즉, $x_0 = a + (b-a)\cdot r_1$

단계5 : 만약 $r_2 \leq c \cdot f(x_0)$이면, x_0가 임의의 변수로 선택된다. 그렇지 않으면 단계3으로 가서 다시 새로운 r_1, r_2를 생성한 후 반복한다.

본 연구에서는 역변환법을 이용하여 확률변수를 발생시킨다. 따라서 지수확률분포를 가지는 실행시간에 대한 변환값은 $x=-\beta\ln[1-F(x)]$와 같다. 여기서 $F(x)$는 난수 발생기에서 발생된 수가된다. 그러나 앞에서 언급한 바와 같이 좋은 난수를 발생시키는 것이 무엇보다도 중요하다. 난수를 발생시키는 방법에도 여러 가지가 있는데 컴퓨터 시뮬레이션용으로는 수식을 이용하여 난수를 발생시키는 방법을 이용한다. 이 방법은 사용하는 수식에 따라 그 장단점이 달라지는데 대표적인 것으로 중앙제곱법(mid-square method), 중앙승산법(mid-product method), 상수승수법, 가산콘그루엔셜법(additive congruential generator) 및 선형콘그루엔셜법(linear congruential generator) 등이 있는데 이중에서 선형콘그루엔셜법이 가장 많이 사용되고 있다.[1]

이와 같은 시뮬레이션 방법을 이용하여 그 결과를 구해보면 다음과 같다. 먼저 오류회복 기능만을 고려한 시뮬레이션 경우의 결과는 표 2와 같으며 이 모델과 같은 조건을 가진 해석적 수식의 계산결과는 표 3에 나타나 있다.

〈표 2〉 페트리 네트를 기초로 한 컴퓨터 사뮬레이션 결과

$\overline{T}_{sch}=0$, $\mu_1=\mu_2=1$, $\alpha_1:\alpha_2=1:1$, $\varepsilon_1=\varepsilon_2=0.0001$, $C_l/C=0.5$, $k_1=k_2=k_3=k_4=0$

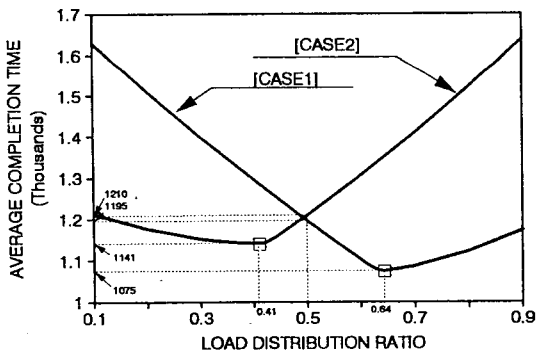
C	$\overline{T}$	PS	$\overline{CT}$	반복 횟수
1000	528.09	0.005	1.64	2000
5500	3244.13	0.108	222.33	2000
10000	6190.94	0.263	1058.67	4000

〈표 3〉 큐잉 모델 해석 결과

$\overline{T}_{sch}=0$, $\mu_1=\mu_2=1$, $\alpha_1:\alpha_2=1:1$, $\varepsilon_1=\varepsilon_2=0.0001$, $C_l/C=0.5$, $k_1=k_2=k_3=k_4=0$

C	$\overline{T}$	PS	$\overline{CT}$
1000	522.77	0.005	1.59
5500	3238.09	0.106	212.20
10000	6250.25	0.264	1036.38

두 표의 결과를 비교해 보면 제시된 시스템에 대한 큐잉 모델을 기초로 한 수학적 해석 결과와 페트리 네트를 이용한 컴퓨터 시뮬레이션 결과와의 오차는 최대 6% 이내이다. 오차가 생기는 이유는 지수 분포를 표현하기 위해 사용한 난수 발생기 오차의 영향이 크며, 컴퓨터 시뮬레이션의 반복 시행 횟수를 많이 하지 않은 원인도 있다. 그러나 6%의 오차는 본 연구에서 제시한 컴퓨터 시뮬레이션 방법의 유효성을 입증하는 결과라고 볼 수 있다. 그 다음 오류회복 기능에 덧붙여서 충돌회피 기능을 고려한 다중로봇시스템은 해석적 수식을 유도하기 어려운 점이 많으므로 그 유효성이 입증된 컴퓨터 시뮬레이션을 이용하면 그림 9에서와 같이 두 로봇의 수행속도와 공동영역 작업 비율이 다를 경우([경우1] : $\mu_1=1$, $\mu_2=2$, [경우2] : $\mu_1=2$, $\mu_2=1$)에 최소 실행시간을 가지는 최적의 분배율 찾을 수 있다.



〈그림 9〉 분배율의 변화에 따른 평균 실행완료 시간의 비교

$C=2000$, $\varepsilon_1=\varepsilon_2=0.0005$, 로봇1의 CWR=0.5, 로봇2의 CWR=0.25, $\overline{T}_{sch}=0$, $k_1=k_2=k_3=k_4=0$)

IV. 결 론

본 논문에서는 다중로봇시스템의 성능평가를 위한 컴퓨터 시뮬레이션 방법을 제시하였다. 먼저 오류회복 기능만을 고려한 다중로봇시스템에 대하여 여러 가지의 모델링 방법 중에서 확장 페트리 네트 모델을 선정하여 컴퓨터 시뮬레이션 하는 과정을 제시하였고 제시된 시뮬레이션 방법의 유효성을 입증하기 위하여 부가적으로 큐잉모델을 기초로 한 해석적 수식을 유도하여 두 가지의 결과를 비교하였다. 그 결과 비교는 표 2와 표 3에서 보는 바와 같이 오차가 최대 6% 이내이므로 오류회복 기능만을 포함한 실제적인 문제에서는 제시된 시뮬레이션 방법이 타당성이 있다는 것을 알 수 있다. 따라서 오류회복 기능에 덧붙여서 충돌회피 기능을 고려한 다중로봇시스템은 해석적 수식을 유도하기 어려운 점이 많으므로 그 유효성이 입증된 컴퓨터 시뮬레이션을 이용하면 그림 9에서와 같이 두 로봇의 수행속도와 공동영역 작업 비율이 다를 경우([경우]」: $\mu_1=1$, $\mu_2=2$, [경우2] : $\mu_1=2$, $\mu_2=1$)에 최소 실행시간을 가지는 최적의 분배율 찾을 수 있다. 본 연구는 다중로봇시스템을 분석하기 위하여 실제 시스템을 가장 정확하게 모사할 수 있는 페트리네트 모델을 기초를 하여 컴퓨터 시뮬레이션 하는 기법을 제시하였다. 이것을 이용하면 다중로봇시스템의 설치 및 운용의 최적화에 도움이 될 것이고 나아가서는 작업 일정계획을 고려한 다중로봇시스템의 성능 평가를 가능하게 할 것이다.

참 고 문 헌

〔1〕 문기수, 산업 시뮬레이션, 생능출판사, 1994.

〔2〕 Y.C.Ho, "Performance Evaluation and Perturbation Analysis of Discrete Events Dynamic Systems," *IEEE Trans. on Automatic Control*, vol.AC-32, pp.563~572, 1987.

〔3〕 D.A.Reed and D.C.Grunwald, "The Performance of Multicomputer Interconnection Network," *IEEE Computer*, vol.20, no.6, pp.63~73, Jun. 1987.

〔4〕 Chi-Keng Tsai, "Multiple Robot Coordination and Programming," *IEEE Int'l Conf. on Robotics & Automation*, pp.978~985, 1991.

〔5〕 Liebowitz, "Multiple Processor Minicomputer Systems Part1 : Design Concepts," *Computer Design*, pp.87~95, Oct. 1978.

〔6〕 L.Kleinrock, *Queueing Systems*, John Wiley &

Sons, New York, pp.141～143, 1975.
〔7〕 이기동, 이범희, 고명삼, "다중로봇 시스템의 비교 성능 분석", 대한 전기학회 논문집 42권 2호 pp.130～141, 1993
〔8〕 Gordon G., *System simulation(second edition)*, Prentice-Hall, 1978.
〔9〕 Law A.M. and Kelton W.D., *Simulation modeling and analysis*, McGraw-Hill Book Company, 1982.
〔10〕 Banks J. and Carson J.S.II, *Discrete-event system simulation*, Prentice-Hall, 1984.

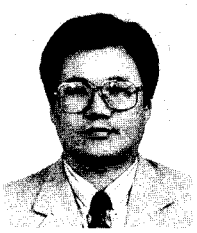
저 자 소 개

李 起 東

1962年 10月 5日生
1985年 2月 서울대학교 공대 제어계측공학과 졸업(학사)
1987年 2月 서울대학교 공대 대학원 제어계측공학과 졸업(석사)
1994年 2月 서울대학교 공대 대학원 제어계측공학과 졸업(박사)

1994年 3月～1995年 2月2 서울대학교 부설 제어계측신기술연구센터 연구원
1995年 3月～현재 영남대학교 전산공학과 전임강사

주관심분야 : 로보틱스 및 자동화



李 範 熙

1955年 5月 29日生
1978年 서울대학교 공대 전자공학과 졸업
1980年 서울대학교 공대 대학원 전자공학과 졸업
1985年 Univ. of Michigan 전기과 졸업

1981年 중앙대학교 전자공학과 전임강사
1985年～1987年 Purdue大 전기공학과 조교수
1987年～1991年 서울대 제어계측공학과 조교수
1991年～현재 서울대 제어계측공학과 부교수

주관심분야 : 로보틱스 및 자동화