

Petri-Net 을 이용한 효과적인 다중로봇 제어알고리즘의 구현

Embodiment of Effective Multi-Robot Control Algorithm Using Petri-Net

선 승 원*, 국 태 용
(Seung-Won Sun and Tai-Young Kurk)

Abstract : A multi-robot control algorithm using Petri-Net is proposed for 5vs5 robot soccer. The dynamic environment of robot soccer is modeled by defining the place and transition of each robot and converting it into Petri-Net diagram. Once all the places and transitions of robots are represented by the Petri-Net model, their actions can be chosen according to the roles of robots and position of the ball in soccer game, e.g., offensive, defensive and goalie robot. The proposed modeling method is implemented for soccer robot system. The efficiency and applicability of the proposed multiple-robot control algorithm using Petri-Net are demonstrated through 5vs5 Middle League SimuroSot soccer game.

Keywords : petri-net, multi-robot, robot soccer system, place, transition, discrete event system, offense strategy, defense strategy, the condition event Petri-Net model

I. 서론

1. 배경

최근 로봇기술이 발전하면서 가정용 로봇이나 엔터테인먼트로봇등 인간 친화적인 로봇에 대한 관심과 연구가 증가하고 있다. 이는 산업현장에서 많이 쓰이는 고정형 로봇 매니플레이터가 제한된 작업공간을 가지는데 반하여 이동성을 갖춘 인간 친화형 로봇들이 다양하고 폭넓은 작업공간을 가짐으로써 좀 더 나은 적용가능성과 확장성을 의미한다. 이러한 이동로봇의 연구 중에서도 복잡하고 어려운 작업을 여러 대의 로봇이 협력함으로써 공동작업을 용이하게 하고 효율을 높이기 위한 다중로봇에 대한 연구도 활발하다[5, 6, 10, 19, 21]. 이러한 다중이동로봇의 연구는 단일로봇이 수행하기 어려운 작업의 수행을 가능하게하고 반복되는 동일한 작업을 여러 대의 로봇이 나누어서 수행함으로써 작업시간을 단축할 수 있도록 한다. 다중이동로봇의 응용이 가능한 분야는 원자력발전소, 폭발물제거, 위험지역 및 미 확인지역 탐사 등 매우 다양하며, 이러한 응용분야의 대부분은 주어진 문제와 주변 여건의 변화가 복잡하고 다양한 동적 환경을 가진 경우가 대부분이다. 그러므로, 다중이동로봇시스템을 구현 하고자 할 때 중요하게 고려해야할 사항은 다음과 같이 세가지로 요약할 수 있다. 첫째는 로봇이 주어진 동적인 환경들을 신속하게 인식하고, 정확히 모델링하도록 하는 것이다. 그 다음은 수행해야 할 복잡한 작업들을 분석한 후, 효율적이고 단순한 작업들로 나누어서 재구성하는 것이며, 마지막으로 단순화된 작업들을 각각의 로봇 시스템에게 나누어서 일을 처리할 수 있게 분배함으로써 주어진 작업들을 신속하면서도 효과적으로 수행할 수 있게 하는 것이다. 특히, 주어진 환경의 급격한 변화는 각각의 로봇들이 효과적으로 작업을 수행하는데 어려움을 주고, 작업의 분산성은 로봇들에게 적절한 임무를 나누어서 분배하기까지 걸리는 시간과 계산 량의 증

가로 인해서 로봇들을 실시간으로 제어하는데 많은 어려움이 따르며, 또한 정밀한 제어도 힘들게 한다. 따라서 동적 환경에서 인식된 정보를 이용하여 보다 효율적인 실시간 제어를 하기 위해서는 작업 수행 로봇들의 기능과 역할을 적절하게 고려하고, 적은 계산 량으로 각각의 로봇이 단순화된 작업을 신속하게 수행하도록 하여야 한다[13]. 이러한 다중이동로봇시스템을 효과적으로 실행할 수 있는 다중로봇제어의 대표적인 연구의 예로 로봇축구시스템을 들 수 있다.

2. 로봇축구 시스템

로봇축구는 FIRA(Federation of International Robot-soccer Association)에 의하여 1996년 첫 대회가 개최된 이후 최근의 2002년 세계로봇축구대회 및 학술행사를 거치면서, 다양한 지능제어기법 및 고급전략이 개발되어 동적환경에서 적용되는 다중로봇 플랫폼이라 할 수 있다[9, 14, 17].

FIRA의 로봇축구의 여러 종목들 중에서 3vs3 Small League MiroSot(Micro Robot Soccer Tournament)에서는 로봇들의 유기적인 협동 전략보다 각 로봇들의 움직임을 제어하는 효율적인 제어기의 우수성을 통해서 경기의 결과가 결정되는 경우가 많았다. 그러나, 최근에 새롭게 추가된 종목인 5vs5 Middle League MiroSot의 경우에도 제어기의 우수성이 요구되지만, 추가된 로봇들과 더욱 넓어진 경기장으로 인해서 로봇간의 적절한 협력을 통한 움직임과 기술적인 면을 한층 더 중요시하게 되었다[11]. 따라서, 5vs5 Middle League MiroSot은 다중이동로봇시스템 연구에 있어서 서로 다른 로봇시스템간의 경쟁과 각 팀들의 전술, 로봇간의 협동 작업수행 능력을 객관적으로 평가할 수 있는 훌륭한 실험조건을 가진 다중로봇시스템이라고 할 수 있다. 즉, 로봇축구시스템이 경기장내 상대팀 로봇들의 급격한 움직임들로 인한 동적 환경을 제공하고 로봇 상호간의 적절한 협동과 수행능력으로 상대팀 로봇들과의 경쟁을 통해서 경기 결과를 승리로 이끌어내야 하는 자율 이동로봇시스템이기 때문이다. 또한, 로봇축구시스템내 각 팀들의 로봇들이 수행해야 할 작업과 역할이 시간, 공간 상에 기능적으로 구분되어져있고, 로봇시스템과의 성능의 우수성을 축구시합의 "Goal" 수로서 객관적으로 평가, 비교할

* 책임저자(Corresponding Author)

논문접수 : 2003. 2. 28., 채택확정 : 2003. 7. 28.

선승원 : Freenex(주) 기술연구소(swsun@freenex.com)

국태용 : 성균관대학교 정보통신공학부(tykuc@yurim.skku.ac.kr)

수 있기 때문이다.

3. 본 논문의 구성 및 연구내용

일반적으로 다중로봇시스템의 각 로봇들의 성능과 역할에 대한 효율적인 작업 분배는 discrete event system을 이용하는 것이 계산 량의 부담이 적고, 구조적으로도 간단하다[2,16]. discrete event system의 state-transition 개념은 여러 가지 형태의 자동화 공정 제어에 효율적인 제어 계획을 개발하도록 제공 해 주는 역할을 한다[7,10,15,20]. 본 논문에서는 여러 가지 discrete event system을 모델링하는 여러 방법들 중에서 Petri-Net 이론을 도입하여 다중이동로봇시스템의 실험 시스템인 로봇축구시스템에서 각 로봇들의 적절한 작업분배를 위한 상위 제어를 모델링 하였다. 이러한 모델을 구현한 전체 제어기의 구조는 Petri-Net Model에 의하여 각 로봇들의 주변 환경과 상황을 판단하고, attack 및 defense, waiting과 같은 세부적인 임무를 나누어서 분배하는 상위 제어기, 로봇들의 움직임을 제어하고 속도제어를 생성하는 하위 제어기와 로봇의 정확한 속도 값을 추종하기 위한 PID-속도제어기로 구성 되어있다. 이와 같이 전체 제어기는 상위와 하위 및 로봇 제어부로 나뉘어져 있으므로, 상황에 따라 변화하는 전술을 추가하거나 삭제할 경우 수정이 필요한 제어기를 구분하여 구현할 수 있고, 오류 수정과 전략 구현 시 정확하고 신속하게 프로그래밍을 할 수 있는 장점이 있다. 또한, 일반적인 로봇 축구의 전략의 경우 다섯 대의 로봇이 한꺼번에 움직이므로 로봇 간의 충돌과 중복되는 역할 등 다중로봇 제어의 어려운 점이 있지만, 본 논문에서 설계된 Petri-Net Model은 각 로봇들에게 적절한 작업 분배를 하므로 로봇 간의 충돌이나 위치 선정, 역할 중복에 의한 비효율적인 문제가 거의 존재 하지 않는다. 본 논문에서 모델링한 다중로봇시스템의 성능평가는 2002년부터 FIRA 로봇축구대회에서 공식종목으로 채택한 로봇축구 시뮬레이터를 사용하는 SimuroSot에서 5vs5 Middle League SimuroSot과 11vs11 Large League SimuroSot 두 가지 환경에서 실험을 하였다[8].

본 논문의 구성은 다음과 같다. 제 II장에서는 Petri-Net에 대한 기본 개념과 discrete event system을 모델링 하는데 사용되는 여러 기법의 하나인 CE(Condition Event) Petri-Net에 관한 개요와 자세한 소개를 한다. 제 III장에서는 State set과 Action set을 이용한 동적환경 인식에 대하여 간략히 기술하였으며, 제 IV장에서는 본 논문에서 제시한 5vs5 로봇축구 전략을 모델링 한 Petri-Net Model을 다룬다. 제 V장에서는 본 논문에서 구현한 Petri-Net Model에 의한 계층적 지능제어기의 구성과 역할에 관하여 기술한다. 제 VI장에서는 본 논문의 Petri-Net Model에 대한 시뮬레이션 결과를 제시한다. 제 VII장에서는 전체적인 연구 결과 및 앞으로의 연구과제에 대하여 언급한다.

II. Petri-Net

1. Petri-Net의 기본 개념

Petri-Net이론은 다른 동적 환경의 일반적인 그래프 모델 보다 비동기적이고 분산적인 시스템을 표현하는데 있어서 편리한 방법을 제공한다. Petri-Net은 동적 환경이나 모델링된

시스템에 대한 정보를 얻을 수 있는 형식적인 방법으로 분석 되어 질 수 있다[15]. 가장 간단한 Petri-Net 모델은 condition event(C/E) Petri-Net이며, 실행되는 로봇의 명령은 상호 의존하는 event들과 condition들로 표현되어진다. 여기서 event는 Petri-Net에서 transition들과 관련이 있고, condition들은 place와 관계가 있다. place와 transition은 호로 직접 연결되어 있다. place에 있는 token의 존재는 그 place에 있는 condition이 참임을 나타낸다. 따라서 place 안의 marking이 Petri-Net의 condition이 참인지 거짓인지를 구분하여 준다. transition의 모든 input place에 marking이 되어 있을 때 그 transition을 fire라고 한다. 즉, event가 발생하면 각각의 input place의 token이 제거되고 각각의 output place에 놓여지게 된다.

이러한 간단한 condition event(C/E) Petri-Net의 일반화시킨 형식이 place/transition(P/T) net이다. 여기서는 간단한 상태들을 단순히 참 또는 거짓으로 나타내는 대신에, token이 일관된 작업의 첫 부분에서 대기 할 수 있는 입력 단의 수나 버퍼 안에서 이용할 수 있는 공간의 수를 나타낸다. 이 경우에는 제한이 없더라도 각 place의 capacity > 0 인 경우가 정의되어야 한다. 여기에 덧붙여서 transition 과 place 사이에 연결되어진 호에 있는 weight는 transition의 firing후에 여러 개의 token이 더해지거나 감소되는 것을 나타낸다. 거대한 시스템의 모델링에는 colored Petri-Net이 사용되어진다. 여기서의 token은 목적물의 여러 가지 다른 계층을 나타내며 그것들의 색상은 서로 다른 색상으로 표현된다. 여기서 transition들은 색상 function에 의하여 정의된 색상 변화를 수행한다. 이러한 colored Petri-Net은 condition event(C/E) Petri-Net으로 변화되어 질 수 있다. 이러한 Petri-Net을 이용한 시스템 모델링은 time에 대해서 고려되지않고 있다.

본 논문에서는 위에서 나열한 다양한 Petri-Net 모델링 방법들 중에서, 다중로봇시스템을 모델링하기 위하여 supervisory control을 이용한 condition event(C/E) Petri-Net을 사용해서 모델링을 할 것이다. 그리고, 본 논문에서 제시한 Petri-Net Model에서 transition은 다중 입력과 하나의 출력으로 구성되어 있고 점화조건은 로봇축구의 특성상 다음과 같이 이루어진다. 로봇축구에서 Ball은 하나만 존재하고, Ball의 위치에 의한 event발생은 동시 다발적인 조건이 발생하는 것이 아니라, 시간과 관계없이 비동기적으로 공격 및 수비와 같은 사건들이 발생한다. 따라서, transition의 점화 조건은 입력 Place들 중에서 하나의 Place에 token이 존재하고, Ball의 위치에 대한 조건을 transition이 만족하게 되면 입력 place의 token은 사라지고, transition의 출력 place로 token이 옮겨가게 된다. 모델링한 결과는 5vs5 Middle League SimuroSot Program 환경에서 실험을 할 것이며, 실제 로봇축구시스템에서도 적용할 예정이다.

2. The condition event(C/E) Petri-Net Model

본 논문에서 제시한 The condition event(C/E) Petri-Net 은 다섯 개의 원소{P,T,IN,OUT, μ }로 이루어진 집합으로 정의 되어진다. 여기서 P는 유한한 place의 집합이고, T는 유한한 transition의 집합이다. IN은 transition의 입력 place들로 정의되는 함수이다. OUT은 transition의 출력 place들로 정의되는 함

수이다. 마킹 μ 는 place에 token을 할당한 것이고, token의 수는 1개로 일정하며 Petri-Net Model의 수행 중에 token의 위치는 변한다.

$$\mu = (\mu_1, \mu_2, \dots, \mu_n), n=|P|, i=1, 2, 3, \dots, n$$

$$\mu(P_i) = \mu_i$$

각각의 place가 하나의 input transition과 하나의 output transition을 가지고 있으면, 이 Petri-Net을 marked graph(or event graph)라 한다. 각각의 transition이 하나의 input place와 하나의 output place를 가지는 경우에, 이 Petri-Net를 state machine이라 한다. 여기서 marked graph는 동기화 모델링 시 아주 유용하게 사용되어지며, state machine은 자원공유와 관련되어지는 경쟁관계를 모델링하는 경우에 이점이 있다. 그 외에도, 다양한 Petri-Net으로 모델링하는 기법과 세부적인 사항들이 있다.

III. State set과 Action set을 이용한 동적 환경 인식

로봇축구 경기에서 팀 전술의 중요한 목적은 상대방 Goal에 Ball을 넣는 것과 상대팀 로봇의 공격에 대해서 효과적으로 수비를 하는 것이다. 5vs5 Middle League MiroSot 경기에서 아군 로봇 5대, 상대방 로봇 5대, Ball 이렇게 11개의 각 개체가 급격히 움직이는 상황에서 상대방 로봇의 움직임과 Ball의 불연속적이고 불규칙한 움직임, 우리 팀 로봇들의 움직임을 정확하게 제어하고 예측하기에는 매우 어려운 일이다. 따라서, 복잡하고 빠르게 변화하는 로봇축구 환경에서는 일관성 있고, 효과적인 전술을 수행하기 위해서 복잡한 동적 환경에 대한 간략화 된 모델링과 이를 기초로 한 일관성 있는 인식 기법이 요구되어진다. 이 장에서는 환경을 단순화하여 모델링하고 인식하기 위하여 8개의 state로 이루어진 state set을 정의하기로 한다. state set내의 각 state는 경기장, Ball, 로봇의 위치에 따라 유리한 상태와 불리한 상태, 또는 공격 가능한 상태와 수비 가능한 상태 등으로 경기장의 환경을 모델링하고 신속하게 인식하게 된다.

각각의 state는 상대방 Goalpost와 Ball, 로봇간의 상대적 위치에 따라 결정되는 positive, negative 또는 singular의 형태로 표현되어진다[3,4,20]. 예를 들면, positive state는 그림 1에서 goalpost-ball-robot(GBR)과 같이 상대적으로 유리한 위치 관계를 가진 경우이고 로봇이 shooting이나 상대 Goalpost로 Ball을 몰고 갈 수 있는 유리한 상태를 나타낸다. negative state는 그림 2에서 보는 바와 같이 positive state에서 Ball과 로봇의 위치가 서로 바뀐 것으로 goalpost-robot-ball(GRB)의 위치에 로봇이 있는 것을 나타내며 로봇이 공격보다는 수비를 해야 하는 불리한 위치에 있는 것을 나타낸다. singular state는 Ball과 로봇의 골대 방향좌표가 같은 경우로 goalpost-robot=ball(GR=B) 순서로 공세를 취할 수 있는 positive state와 negative state의 중간단계를 나타내며, singular state 또한, 그림 3에서 볼 수 있듯이 로봇이 공세를 취할 수 있는 유리한 상황은 아니다.

$$S_p = \{GBR_C, GBR_R, GBR_L\}$$

$$S_N = \{GRB_C, GRB_R, GRB_L\}$$

$$S_s = \{GB=R_R, GB=R_L\}$$

여기서 S_p, S_N, S_s 는 각각 positive state와 negative state, singular state를 나타내며, 첨자 C, R, L 은 각각 로봇의 위치가 Goal-line의 중앙, 오른쪽, 왼쪽을 나타낸다. 따라서 전체 state는 8개이며 5대의 로봇에 대하여 매 vision sampling 시간마다 각 로봇의 상태가 인식되고 결정된다. 아래의 그림 1 ~ 그림 3는 경기장 상황에 따른 1대의 로봇에 대한 state 예를 나타낸 것이며, OR은 상대팀 로봇을 나타낸다. State set은 간략화된

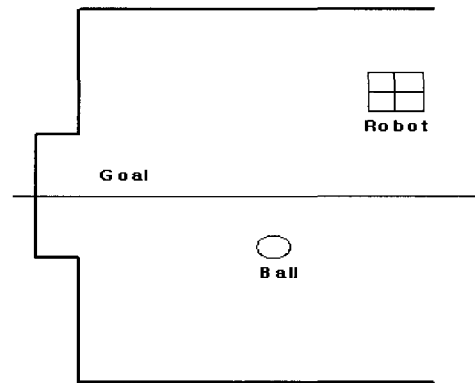


그림 1. Positive State(GBR_R).
Fig. 1. Positive State(GBR_R).

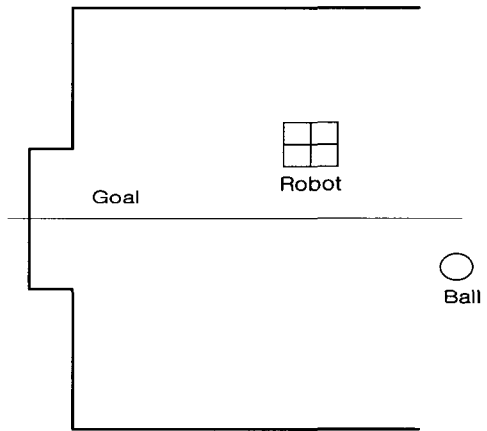


그림 2. Negative State(GRB_R).
Fig. 2. Negative State(GRB_R).

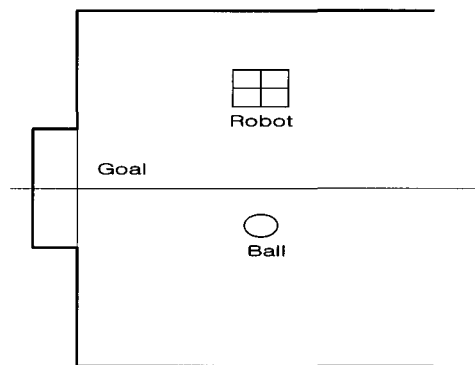


그림 3. Singular State(GB=R_R).
Fig. 3. Singular State(GB=R_R).

경기장의 모델링이지만 로봇의 상태판단과 역할판단에 핵심적인 요소로 작용하며, 복잡한 상위제어기에서 사용되는 if-else 문을 구조적으로 간략화 시키게 된다. Action set은 경기 중에 발생할 수 있는 로봇의 대표적인 5개의 동작을 각각의 특징별로 정의한 집합으로 목적된 동작을 수행하기 위해 로봇에게 전달되는 목적벡터를 생성하게 된다. 여기서 5개의 action은 attack, wait, defense, blocking, change state를 뜻하며 필요에 따라 세분화 할 수 있고, 변화된 전술에 의해 추가 및 수정도 가능하게 된다.

IV. 공격 및 수비 전략에 대한 Petri-Net Model

상위 제어기의 설계는 앞에서 기술한 state들을 이용하여 로봇의 대표적인 다섯 개의 동작인 action set 및 세부적인 동작들을 각각 모듈화해서 함수로 구현을 한 후 전체적인 로봇의 움직임을 제어하는 로봇축구 경기의 5vs5 Middle League MiroSot에서 5vs5 공·수에 관한 전략을 모델링하려고 한다. 각각의 로봇의 상태에 따른 동작은 앞장에서 기술한 state 모델 및 아군 로봇들과 Ball 사이의 거리 등 여러 조건들을 고려해서, the condition/event(C/E) Petri-Net을 이용하여 상위 전략 제어알고리즘을 모델링해서 구현한다.

Petri-Net으로 모델링한 전략은 크게 두 부분으로 나뉘어진다. 그 이유는 로봇축구시스템에서 가장 중요한 목적은 상대방 Goal 지역에 Ball을 넣는 것이고, 상대방 로봇의 공격에 대해서 Ball을 아군 Goal 지역에 넣지 못하도록 적절하게 방어하는 것이기 때문이다. 따라서, 첫 번째 전략은 Ball이 상대 진영에 있어서 공격상황인 경우와 두 번째는 Ball이 아군진영으로 넘어와서 수비상황인 경우를 고려하여 이렇게 두 부분으로 나누어서 전략을 모델링하였다.

1. 공격 상황에서의 Petri-Net Model

Ball이 상대 진영에 있을 때 공격 전술이 펼쳐지는데, 공격 상황에서는 다섯 대의 로봇들 중에서 골키퍼를 제외한 네 대의 로봇으로 각 로봇들의 상황에 맞게 적절한 위치선정과 각각의 로봇의 분업화된 역할을 고려해서, 9개의 place와 33개의 transition으로 Petri-Net Model을 설계하였다.

II장에서 제시된 Petri-Net을 이용하여 모델링한 공격 상황의 Petri-Net Model을 좀더 자세히 살펴보기로 하자. 이 모델은 앞에서 모델링 한 것을 보면 9개의 Place와 33개의 Transition으로 이루어져 있다. P1에서 표현된 attack과 wait, defense는 모듈화된 함수들이다. 모듈화된 함수들에 대해서는 V장에서 설명하기로 하겠다. 그럼, place와 transition의 관계에 관해서 알아보기로 하자.

$$\text{Place : } P = \{ P_1, P_9 \}$$

$$\text{Transition : } T = \{ T_1, T_{33} \}$$

여기에서 place는 표 1과 같이 나타내어지고, 각 상태에 대한 조건으로 나타나는 transition은 표 2와 같이 표현되어진다. 여기에서 정의된 Petri-Net을 이용한 공격 상황의 Petri-Net Model은 그림 17과 같이 나타내게 된다.

T1에서는 Ball이 왼쪽 지역에 위치하고, robot1이 robot3보다 Ball에 가깝고, robot3은 왼쪽 미드필더 지역에서 defense 상황이며, 오른쪽 지역에서는 robot2가 wait 상황이고, robot4가 오

표 1. 공격 전략에 대한 Place.

Table 1. Place of the offense strategy.

P1 : Robot1 attack, robot2 wait, robot3 defense(left), robot4 defense(right)
P2 : Robot1 wait, robot2 attack, robot3 defense(left), robot4 defense(right)
P3 : Robot1 defense(left), robot2 attack, robot3 wait, robot4 defense(right)
P4 : Robot1 defense(left), robot2 wait, robot3 attack, robot4 defense(right)
P5 : Robot1 defense(left), robot2 defense(right), robot3 wait, robot4 attack
P6 : Robot1 defense(left), robot2 defense(right), robot3 attack, robot4 wait
P7 : Robot1 attack, robot2 defense(right), robot3 defense(left), robot4 wait
P8 : Robot1 wait, robot2 defense(right), robot3 defense(left), robot4 attack
P9 : change action

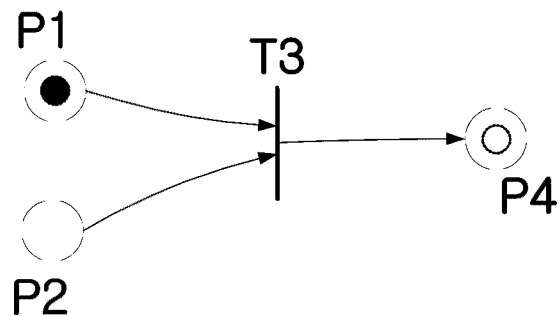


그림 4. P1 : Transition 3 fire : P4.

Fig. 4. P1 : Transition 3 fire : P4.

른쪽 미드필더 지역에서 defense 상황인 것을 의미한다. 예를 들어, 그림 4와 그림 5의 T3가 fire되는 상황과 P4로 token이 옮겨가면서 P4가 수행되는 과정을 고려해보자. 먼저, 그림 4는 token이 P1에 존재하므로 robot2는 wait, robot 3가 왼쪽 미드필더 지역에서 defense, robot4는 오른쪽 미드필더 지역에서 defense 역할을 수행하고 있는 중이고, robot1이 attack을 하는 상황이다. 그러나, robot1이 attack을 수행하다가 Ball을 지나치게 되어서, Ball이 robot3에 더 가까워지면 transition 3의 조건을 만족하게된다. 따라서, token은 P4로 옮겨가면서 P4의 상태에 맞는 임무를 4대의 로봇에게 전달하게 된다.

P4를 수행하게 되면, robot1이 Ball과의 상태가 Negative이면서 거리가 떨어져 있으므로 defense 역할을 맡을 것이고, robot3은 Ball과의 거리가 가깝고 공격하기에 유리한 상황이므로 Ball을 드리블하거나 Shoot을 하는 attack역할로 전환되게 된다. 결과적으로 robot1과 robot3가 Ball의 위치에 따라서 서로의 역할을 change하게 된다. 그림 5는 token이 P2에 존재하므로 robot1은 wait, robot 3가 왼쪽 미드필더 지역에서 defense, robot4는 오른쪽 미드필더 지역에서 defense 역할을 수행하고 있는 중이고, robot2가 attack을 하는 상황이다. 그러나, robot2가 attack을 수행하다가 Ball이 오른쪽 지역에서 왼

표 2. 공격 전략에 대한 Transition.

Table 2. Transition of the offense strategy.

T1 : Ball left plane, robot1 is close by the ball (Positive{GBR1}, Negative{GR1B}, Singular{GB=R1}), robot3 is defense //right plane: robot2 wait(or attack) and robot4 defense
T2 : Ball left plane, robot1 is close by the ball (Positive{GBR1}, Negative{GR1B}, Singular{GB=R1}), robot3 is defense //right plane: robot4 wait(or attack) and robot2 defense
T3 : ball left plane, robot1 is far from the ball (Negative{GR1B}, Singular{GB=R1}) //right plane: robot2 wait(or attack) and robot4 defense
T4 : ball left plane, robot1 is far from the ball (Negative{GR1B} ,Singular{GB=R1}) //right plane: robot4 wait(or attack) and robot2 defense
T5 : ball left plane, robot3 is close by the ball (Positive{GBR3}, Negative{GR3B}, Singular{GB=R3}), robot1 is defense //right plane: robot2 wait(or attack) and robot4 defense
T6: ball left plane, robot3 is close by the ball (Postivie{GBR3}, Negative{GR3B}, Singular{GB=R3}), robot1 is defense//right plane: robot4 wait(or attack) and robot2 defense
...
T30 : ball center plane, robot4 is close by the ball (Positive{GBR4}, Negative{GR4B}, Singular{GB=R4}), robot2 is defense. (robot4 distance<= robot3 distance) //left plane: robot3 wait(or attack) and robot1 defense
T31 : ball center plane, robot4 is far from the ball (Negative{GR4B}, Singular{GB=R4}) robot2 is close by the ball(defense). (robot2 distance < robot3 distance) //left plane: robot1 wait(or attack) and robot3 defense
T32 : Ball center plane, robot4 is far from the ball (Negative{GR4B}, Singular{GB=R4}) robot2 is close by the ball(defense). (robot2 distance < robot1 distance) //left plane: robot3 wait(or attack) and robot1 defense
T33 : Ball moves from opponent field to our field

쪽으로 이동하면서 공격지역에서 대기하고 있는 robot1보다 왼쪽 미드필더에서 defense 역할을 하고 있는 robot3에 가깝게 위치하게 되다면 transition 3의 조건을 만족하게 된다. 그러므로, 위의 상황과 똑같이 token이 P4로 이동하게 되면서 P4의 상태에 맞는 임무를 4대의 로봇에게 분배하게 된다. 즉, robot1과 robot3 의 Ball의 위치에따른 역할 change가 일어난다.

T33은 공격전략의 모든 Place에서 Ball이 아군진영으로 넘어 오는 경우에 발생하는 조건이고, 그 조건을 만족하게 된다면 P9에 해당하는 Action으로 바뀌게 되어서 Ball의 위치에 따른 수비전략을 수행하게 된다.

2. 수비 상황에서의 Petri-Net Model

수비상황에서는 다섯 대의 로봇 중에서 골키퍼는 각각 조건상황에서 독립적으로 골키퍼의 단독임무가 수행되고 나머지 네 대의 로봇들 중에서 두 대의 로봇은 공을 중심으로 방어할 하게 되며, 다른 두 대의 로봇은 위치를 선점해서 공격

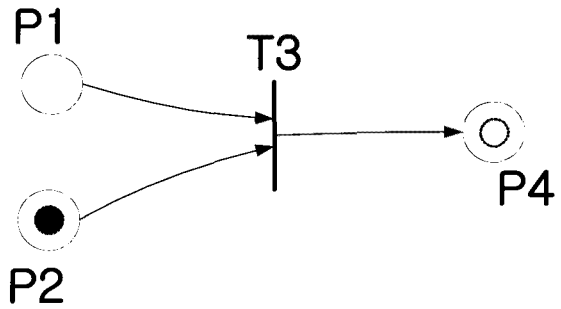


그림 5. P2 : Transition 3 fire : P4.

Fig. 5. P2 : Transition 3 fire : P4.

표 3. Transition의 입출력 관계.

Table 3. I/O relation of the transition.

Input	Output
T1 : P1, P2	T1 : P1
T2 : P7, P8	T2 : P7
T3 : P1, P2	T3 : P4
T4 : P7, P8	T4 : P6
...	...
T30 : P5, P6	T30 : P5
T31 : P7, P8	T31 : P2
T32 : P5, P6	T32 : P3
T33:P1, P2, P3, P5, P6, P7, P8	T33 : P9

표 4. 수비 전략에 대한 Place.

Table 4. Place of the defense strategy.

P1 : Robot1 attack, robot2 wait, robot3 defense(left), robot4 defense(right)
P2 : Robot1 wait, robot2 attack, robot3 defense(left), robot4 defense(right)
P3 : Robot1 attack, robot2 wait(or Goalie2), robot3 ball_touch, robot4 defense(right)
P4 : Robot1 wait(or Goalie1), robot2 attack, robot3 defense(left), robot4 ball_touch
P5 : Robot1 Goalie1, robot2 Goalie2, robot3 ball_touch, robot4 defense(right)
P6 : Robot1 Goalie1, robot2 Goalie2, robot 3 defense(left), robot4 ball_touch
P7 : Robot1 attack, robot2 attack, robot3 defense(left), robot defense(right)<center area>
P8 : Robot1 Goalie1, robot2 Goalie2
P9 : Change action

로봇과 자기의 위치에서 공을 처내는 역할을 고려하여 9개의 Place와 10개의 transition으로 Petri-Net Model을 구현하였다.

Place: $P = \{ P_1 \sim P_9 \}$
 Transition: $T = \{ T_1 \sim T_{10} \}$

수비상황에서의 place는 표 4와 같이 표현되고, 각 상태에 대한 조건으로 나타나는 transition은 표 5와 같이 나타내어진다. 여기에서 정의된 Petri-Net을 이용한 수비 상황의 Petri-Net Model은 그림 18과 같이 나타나게 된다. Ball이 아군진영에

표 5. 수비 전략에 대한 Transition.

Table 5. Transition of the defense strategy.

T1 : Ball left plane & not left wing zone, robot1 is close by the ball (Positive{GBR1}, Negative{GR1B}, Singular{GB=R1}), robot3 is defense //right plane: robot2 wait(or attack) and robot4 defense
T2 : Ball left plane & not left wing zone, robot1 is far from the ball (Negative{GR1B}, Singular{GB=R1}) //right plane: robot2 wait(or attack) and robot4 defense
T3: Ball left plane & left wing zone, robot3 is close by the ball(Positive{GBR3}, Negative{GR3B}, Singular{GB=R2}), robot1 is attack(wait-position) //right plane: robot2 wait(or attack) and robot4 defense
T4: Ball right plane & not right wing zone, robot2 is close by the ball(Positive{GBR2}, Negative{GR2B}, Singular{GB=R2}), robot4 is defense //left plane: robot1 wait(or attack) and robot3 defense
...
T9 : Ball center plane & goal zone //robot3 and robot4 defense
T10 : Ball moves from our field to opponent field

있을 때 로봇들의 기본위치 구성은 두 대의 로봇(robot3, robot4)이 골키퍼 양 옆을 방어하기 위해서 골키퍼와 일정한 거리가 떨어진 곳에서 위치를 선점하고 그 위치를 고수한다. Ball이 left, center, right area 각 지역의 존재 유무에 따라서 left 에서는 robot1과 robot3의 방어 및 공격임무가 수행되고 right 에서는 robot2와 robot4가 방어 및 공격임무를 수행하게 된다. center에서는 robot1과 robot2가 동시에 Ball을 방어와 공격의 임무를 수행하며, Ball이 Goal-area안으로 들어갔을 때는 Ball에 상대방 로봇들이 접근하지 못하도록 robot1과 robot2가 Ball의 일정한 거리 앞에서 둘러싼 형태로, Ball의 움직임에 따라 위 아래로 이동하면서 상대방 로봇들이 접근하지 못하도록 방어임무를 수행하는 역할을 한다.

표 6. Transition의 입출력 관계.

Table 6. I/O relation of the transitions.

Input	Output
T1: P1, P2, P3, P5, P7, P8	T1 : P1
T2: P1, P2, P3, P7, P8	T2 : P3
...	...
T9: P1, P2, P3, P5, P6, P7, P8	T9 : P8
T10:P1, P2, P3, P5, P6, P7, P8	T10 : P9

본 논문에서 제안한 수비전략에 대해서 대략적으로 살펴 보았다. 그림 6의 예를 들어서 수비전략에 대해서 좀 더 자세히 살펴보도록 하자.

Token이 P4에 있는 상황은 Ball이 아군 진영의 오른쪽 Goal area에 가까운 위치에 있을 때 robot2와 robot4는 Ball을 적 진영으로 보내기 위해서 협력 수비를 하고 있으며, robot1은 Goal area 앞이나 공의 일정한 거리에서 대기하고 있으면서 Ball이 넘어오기를 기다리고 있고, robot4는 Goal-Keeper robot의 왼쪽 지역에서 대기하면서 수비임무를 수행하고 있다. 로봇들이 P4를 수행하고 있는 상황에서 Ball이 robot2가

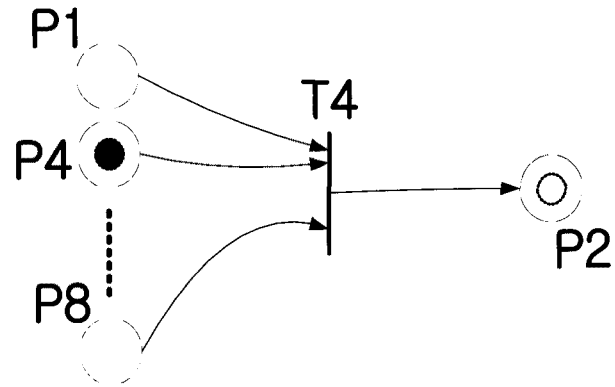


그림 6. P4 : Transition 4 fire : P2.

Fig. 6. P4 : Transition 4 fire : P2.

공격하기 좋은 위치로 이동하게 되면, robot 4는 Goal area 오른쪽, robot3는 Goal area 왼쪽에서 defense를 담당하고, robot1은 중앙에서 약간 왼쪽으로 치우쳐서 대기하고 있는 상황이므로 transition 4를 만족하게 된다. 따라서, P2로 token이 옮겨가게 되고 robot2는 적 진영으로 Ball을 드리블하거나 공격을 하는 임무를 수행하고, 나머지 로봇들은 Ball이 적 진영으로 넘어갈 때까지 수비나 대기를 하는 임무를 계속해서 수행하게 된다.

V. 계층적 지능제어의 구성

IV.장에서 살펴본 Petri-Net Model에서 공격과 수비 전략에 따라서 각 로봇들이 수행해야 할 역할들을 적절하게 나누어서 분배하는 것을 보았다. 여기에서는 로봇들에게 전달되는 세부적인 역할에 대해서 좀 더 자세히 살펴보기로 하자.[1,3,4,20] 그림 7은 Petri-Net Model의 바로 아래 계층의 제어기 구조이다. 먼저, Ball의 위치를 인식하고, 그 위치에 따라서 공격 전략이나 수비 전략을 선택한다. 공격 전략에서는 대표적인 세 가지 세부동작을 볼 수가 있다. attack 명령은 III 장에서의 positive, negative, singular 상태를 고려하고, 반사 행동 제어기를 사용해서 Motion Control함수의 조합으로 이루어져 있다[1]. 이 명령은 shooting, approach ball, dribble, change action 네 가지 역할을 포함하고 있으며, attack 명령이 내려지면 로봇은 어떠한 상황에서도 공격을 수행하도록 구성되어 있다. waiting_a 명령은 Ball의 움직임에 따라서 일정한 거리를 두고 이동하면서 공격상황을 기다리는 역할을 수행한다.

defense_waiting은 로봇이 중앙선 근처의 정지한 위치에서 Ball의 방향을 향하여 계속 움직이며, 공격명령을 내리면 즉시, Ball에게 접근해서 공격임무를 수행하게 된다. 수비 전략에서는 attack_d, waiting_d, defense, goalie 이렇게 네 가지가 대표적인 수비전략으로 구성되어있다. attack_d는 공격 전략에서와 마찬가지로 공격적인 동작을 취하는 명령이다. waiting_d는 right 나 left 지역에 Ball이 존재하는 경우 두 대의 로봇이 협력해서 방어 임무를 수행하고, 중앙에 있는 로봇이 Ball의 갑작스런 접근을 방어 하기 위해서 Ball의 일정한 거리를 유지하며 대기하는 명령이다. defense 명령은 골키퍼 양 옆의

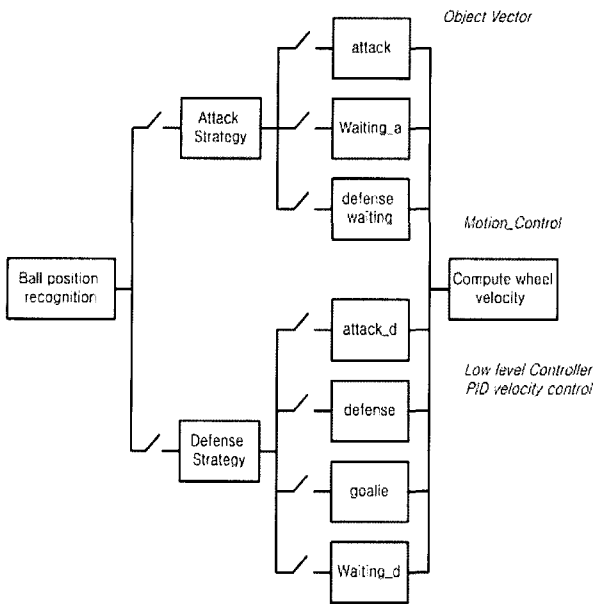


그림 7. 상위 레벨의 제어 구조.
Fig. 7. Control structure of High level.

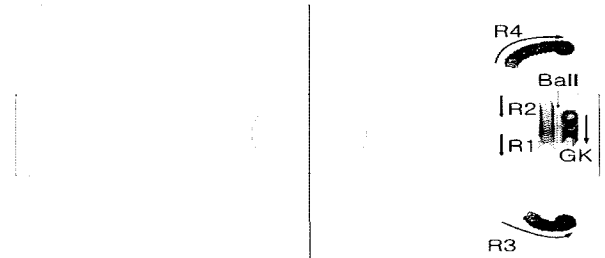


그림 10. T9 fire → P8 (Defense Strategy).
Fig. 10. T9 fire → P8 (Defense Strategy).

상위 제어기의 구조는 위에서 설명한 것과 같이, Ball의 위치에 따른 각각의 로봇들이 수행해야 할 행동들을 Petri-Net Model에 의해서 정해주고, 하위 제어기로 로봇들이 이동해야 할 위치를 전달해준다. 하위 제어기는 로봇들이 이동해야 할 각 위치들을 반사 행동 제어기로서 로봇들의 이동해야 할 속도를 생성하게 되고, 그 속도 값을 PID 속도 제어기로서 Motor를 제어하게 된다[1]. 따라서, 상위제어기는 Petri-Net Model로 이루어져 있고, 하위제어기는 반사 행동 제어기와 PID 속도 제어기로 이루어져 있다.

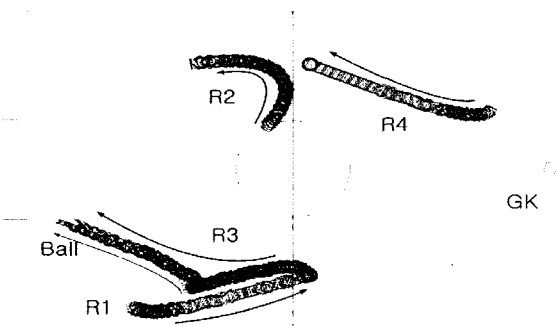


그림 8. T3 fire → P4(Offense Strategy).
Fig. 8. T3 fire → P4(Offense Strategy).

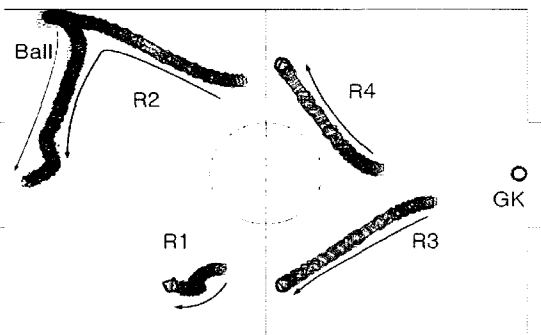


그림 9. T9 fire → P2(Offense Strategy).
Fig. 9. T9 fire → P2(Offense Strategy).

위치를 고수하면서 상대방 로봇이 Ball 을 물고 접근하는 것을 막아주고, Ball 을 상대 진영으로 물고 가는 역할을 하는 명령이다. goalie 는 1,2 번 로봇이 goal area 에 Ball 이 존재하는 경우에 Ball 의 움직임에 따라서 이동하면서 방어를 하도록 하는 명령이다. 이 모든 명령들의 조합 역시 작은 Petri-Net Model 이고, 각각의 명령은 Module 화 된 함수로 이루어져 있다.

VI. 시뮬레이션

이 장에서는 IV장에서 설계한 Petri-Net Model을 시뮬레이션 프로그램에 적용해서 설계한 의도대로 효율적으로 수행되는 시뮬레이션 결과들을 보여준다. 시뮬레이션은 실제 로봇축구대회에서 정식종목으로 사용되고 있는 프로그램인 5vs5 Middle League SimuroSot과 11vs11 Large League SimuroSot 두 종목에서 시뮬레이션을 하였다[8,11].

그림 8은 공격 전략에서 robot3이 robot1 보다 Ball에 가까운 상태이므로, robot1과 임무를 교대해서 robot3이 공격을 하고 robot1이 defense_waiting 상태로 들어간다. robot2는 Ball과 일정한 거리를 유지하면서 대기한다. 그림 9는 상대방 진영에 Ball이 있으므로 robot2가 Ball을 물고 가면서 상대방 goal 지역에 Goal을 넣는 것을 보여준다. 그림 10은 Ball이 골키퍼 앞에 있을 때 robot1과 robot2가 Ball을 감싼 상태에서 일정한 거리를 유지하고, Ball의 움직임에 따라 움직이면서 방어하는 것을 나타내고 있다. 그림 11은 Transition 7이나 8이 fire 되었을 때, P7에 해당하는 방어역할을 수행하는 것을 볼 수 있다. 그림 12는 공격 전략을 수행하면서 두 가지 상황이 연속으로 변하는 것을 보여 주고 있다. robot2가 공격하려는 상황에서 robot1이 더 신속하게 Ball에 가깝게 접근 하였으므로, robot1이 Ball을 물고 가면서 공격을 하고, 공격을 하던 robot2는 waiting_a 상황으로 전환 한다. 그림 13은 Ball이 야군진영에 있을 때, 방어 임무를 수행하다가 robot1이 Ball을 상대 goal 지역으로 Goal을 넣는 공격 상황으로 바뀌면서 임무를 수행하는 것을 보여준다. 그림 14는 Ball이 벽에 부딪혀서 상대 진영에서 야군진영으로 넘어오는 여러 상황 속에서, 로봇들은 방어에서 공격, 공격에서 방어를 수행하는 3가지 상황을 연속적으로 보여주고 있다. 그림 15는 방어에서 두 가지 공격상황을 나타내고 있으며, 그림 16은 방어에서 공격 전략을 수행하는 것을 연속적으로 보여주고 있다.

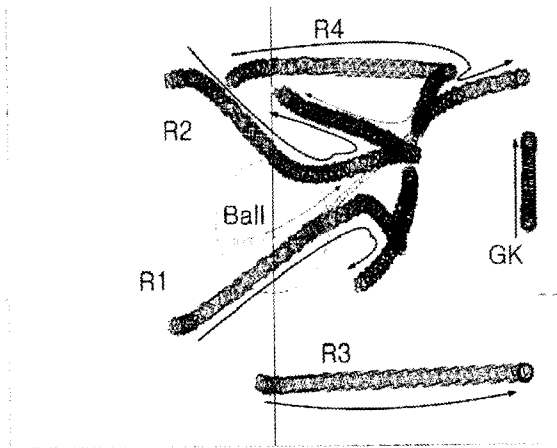


그림 11. T7, T8 fire → P7(Defense Strategy).
 Fig. 11. T7, T8 fire → P7(Defense Strategy).

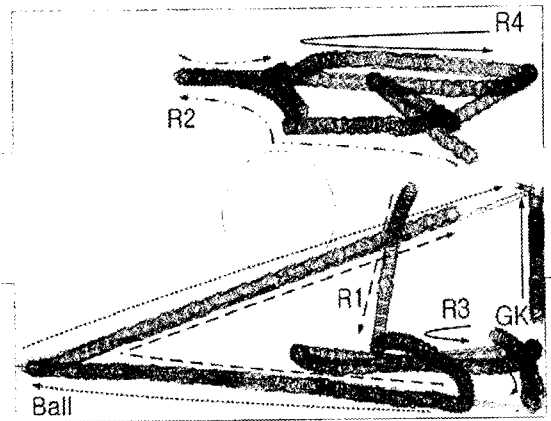


그림 14. T2 → P3(Defense) ⇒ T1 → P1(Offense) ⇒
 T9 → P8(Defense).
 Fig. 14. T2 → P3(Defense) ⇒ T1 → P1(Offense) ⇒
 T9 → P8(Defense).

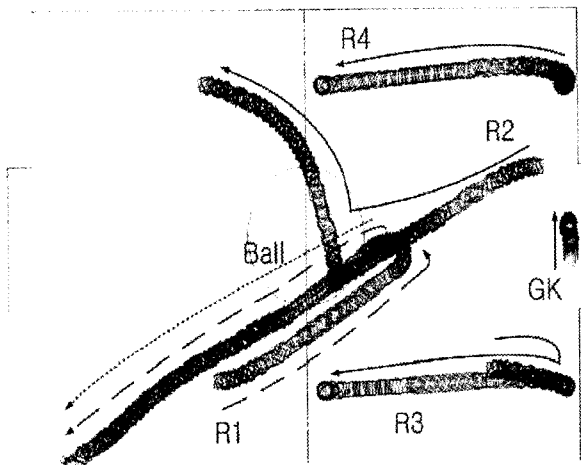


그림 12. T25 fire → P2 ⇒ T17 fire → P1(Offense Strategy).
 Fig. 12. T25 fire → P2 ⇒ T17 fire → P1(Offense Strategy).

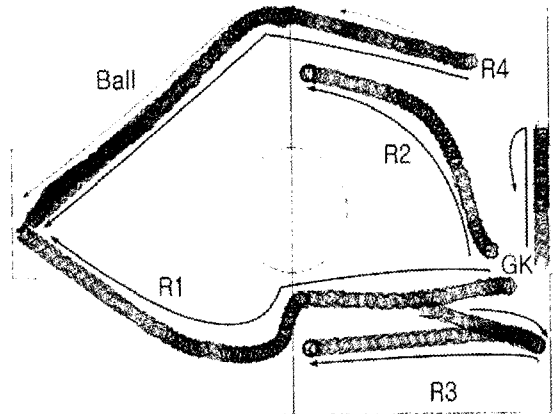


그림 15. T5 → P4(Defense) ⇒ T13 → P8(Offense) ⇒
 T18 → P7(Offense).
 Fig. 15. T5 → P4(Defense) ⇒ T13 → P8(Offense) ⇒
 T18 → P7(Offense).

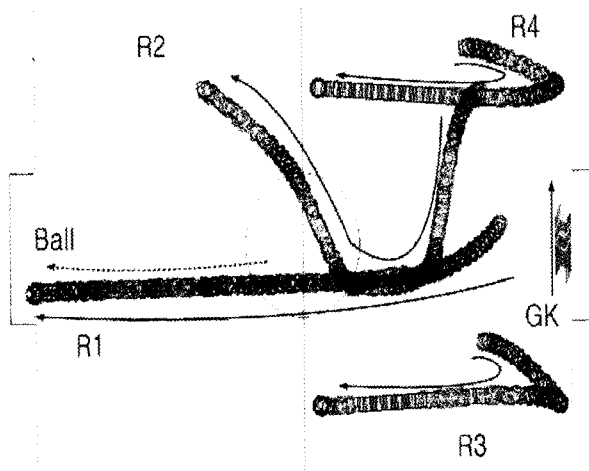


그림 13. T7 → P7(Defense Strategy) ⇒ T1 → P1(Offense Strategy).
 Fig. 13. T7 → P7(Defense Strategy) ⇒ T1 → P1(Offense Strategy).

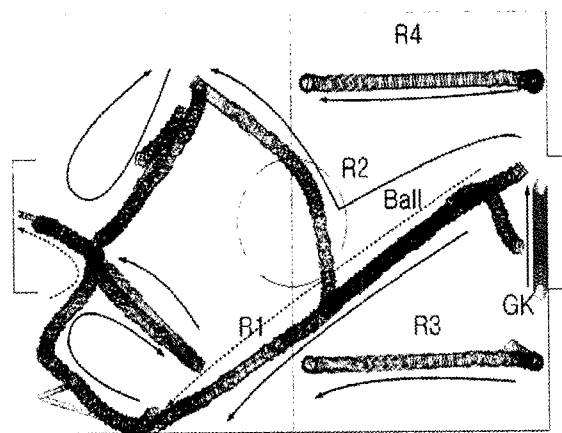


그림 16. Defense Strategy ⇒ Offense Strategy.
 Fig. 16. Defense Strategy ⇒ Offense Strategy.

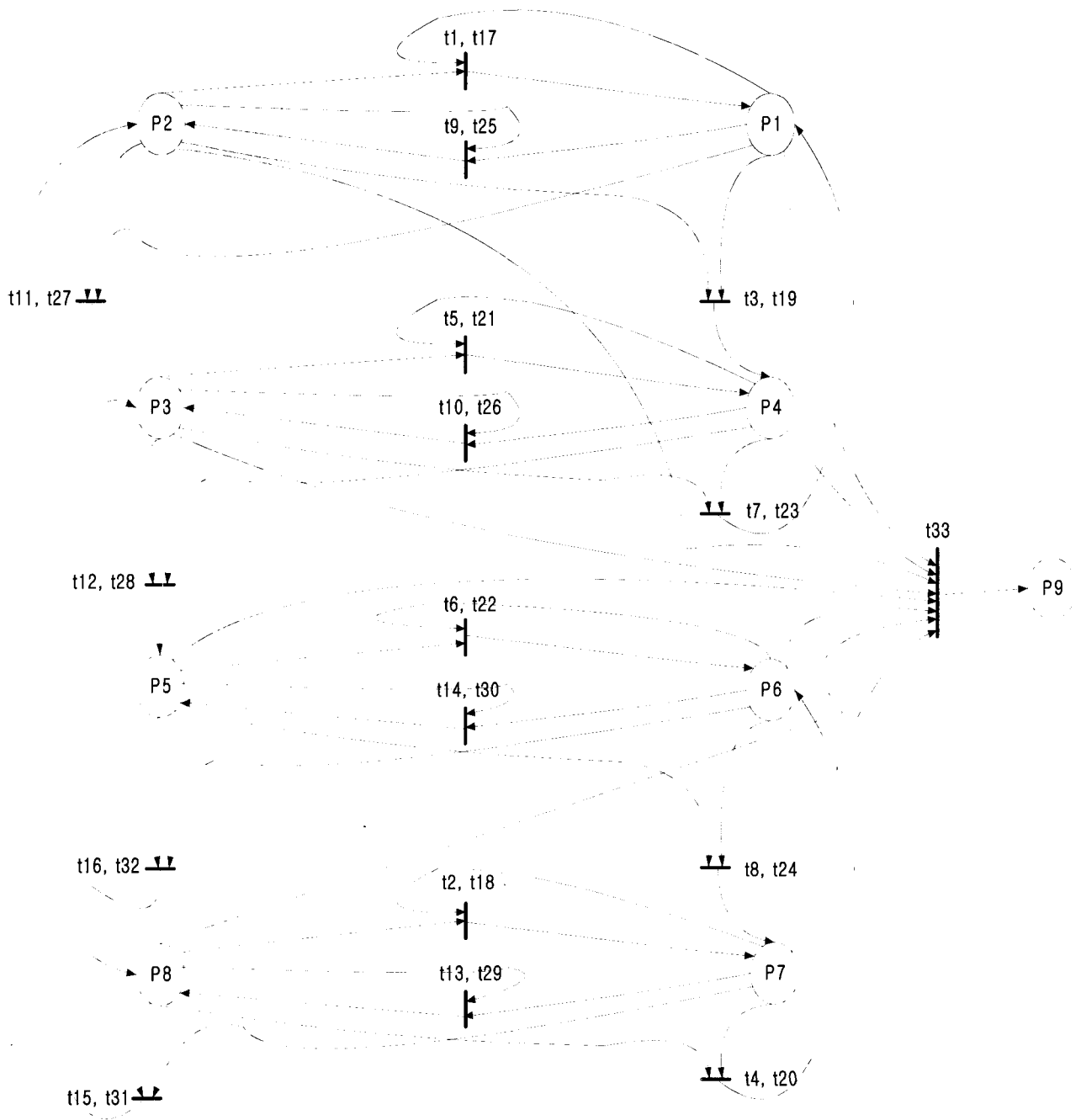


그림 17. 공격 전략에 대한 Petri-Net Model.

Fig. 17. Offense strategy Petri-Net Model.

여러 번에 걸쳐서 실행한 시뮬레이션 결과들을 보게 되면 Petri-Net Model을 설계한 의도에 따라서, 특정한 상황뿐만 아니라 공·수 두 가지의 계속되는 상황변화와 Ball의 불연속적인 움직임에 따라 연속적으로 바뀌는 상태에서도, 로봇들이 각각의 임무에 따른 역할들을 훌륭하게 수행하는 것을 확인할 수 있었다.

VII. 결론

여러 개의 개체가 급격한 움직임과 불규칙적으로 계속적인 변화를 만들어 내는 다중로봇시스템 환경에서 로봇들을 정확

하게 제어하기에는 많은 어려움이 있다. 그 이유는 동적 환경과 여러 개체들의 상황을 고려할 수 많은 계산 량과 조건들은 시간지연으로 인해서 여러 대의 로봇들을 실시간으로 제어하기 힘들게 만들기 때문이다. 본 논문에서는 로봇들의 효율적인 움직임을 위하여 계산 량과 조건들을 줄이고, 복잡하고 연속적인 행동에 대한 모델링을 하기위해서 Petri-Net 이라는 모델링 Tool 을 사용하였다. Ball의 위치에 따라서 로봇들의 수행해야할 작업들을 효율적으로 분배를 하기위하여 동적인 환경에 대한 인식과 모델링을 통해 상태는 place 로 조건은 transition 으로 나누어서, Petri-Net Model 을 설계하였다. 본 논문

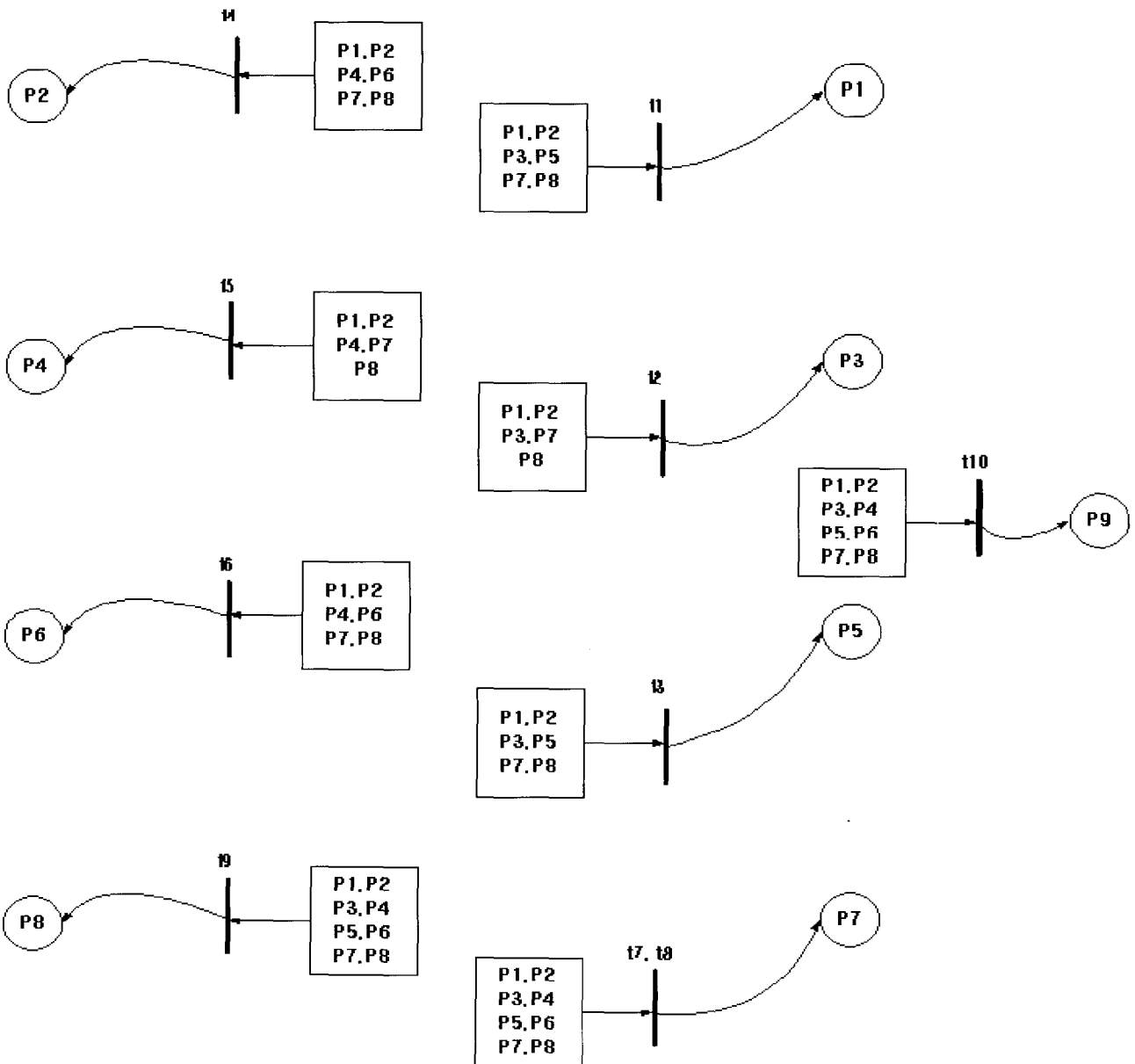


그림 18. 수비 전략에 대한 Petri-Net Model.
 Fig. 18. Defense strategy Petri-Net Model.

에서 제시한 Petri-Net Model 의 장점은 공격상황에는 적절한 역할 분담을 통한 공격을 하면서, Ball 에 대한 위치변화에 따라서 공격하기 유리한 로봇을 인식한 후 서로 역할 교대를 통한 효율적인 공격과 수비라고 할 수 있겠다. 여러 가지 임무에 대해서 로봇들의 협동에 의한 수행능력은 VI 장의 시뮬레이션 결과들을 통하여 충분히 알 수가 있었다.

본 논문에서 제시한 Petri-Net Model은 축구 로봇의 guide를 이용한 dribble과 shooting기능을 고려하여 설계하였으나, 시뮬레이터 환경에서 로봇이 평면으로 설계되었으므로 Petri-Net Model에 대한 전략의 일부만이 시뮬레이터 환경과 잘 맞지 않은 문제점을 나타냈다.

추후 연구과제로는 실제 로봇축구시스템에 대한 Petri-Net Model의 적용과 본 논문에서 제시한 Petri-Net Model을 보완하여 공격로봇이 상대로봇에 막혀 있거나 벽에 갇힌 경우 공격

하기에 유리하기 위해서, 대기하고 있던 로봇들이 이러한 상황을 인식하여 역할을 바꾸어서 공격을 할 수 있도록 세부적인 부분을 고려해서 진술을 수정하고 추가해야 하는 것이다. 또한, 공격이나 수비 등 모든 상황에서, 네 대의 로봇 모두가 상황에 따른 교대시스템 구축과 패스를 통한 효율적인 공격과 수비를 할 수 있도록 보완하는 것이고, 더 나아가서 11vs11 환경에서 Petri-Net Model을 설계하고 11대의 로봇들에 대한 제어알고리즘을 효과적으로 구현하는 것이다.

참고문헌

[1] 백승민, “이동로봇의 지능형 복합제어,” 성균관대학교 정보통신공학부 전기전자 및 컴퓨터 공학과 박사학위청구 논문
 [2] 박세웅, “패트리 네트를 이용한 다중 로봇 제어,” 성균관대학교 정보통신공학부 전기전자 및 컴퓨터 공학과 석사

학위청구 논문

- [3] 이인재, "다중이동로봇의 계층적 지능제어," 성균관대학교 정보통신공학부 전기전자 및 컴퓨터 공학과 석사학위청구 논문
- [4] 이인재, 백승민, 손경오, 국태용, "축구로봇을 위한 계층적 제어기," 제어·자동화·시스템공학 논문지, 제 6 권, 제 9 호, pp. 803-812, 2000. 9.
- [5] A. Agah and K. Tanie, "Robots playing to win : evolutionary soccer strategies," *IEEE Int. Conf. Robotics & Automation*, PP. 632-637, Albuquerque, NewMexico April 1977
- [6] B.-J. Lee and G.-T. Park, "A robot in intelligent environment : soccer robot," *Proceeding of the 1999 IEEE/ASME International Conference on Advaced Intelligent Mechatronics*, September 19-23, 1999 · Atlanta, USA
- [7] D.-H. Kim, Y.-J. Kim and J.-H. Kim, "Nonlinear field based path planning and petri-net based role selection mechanism which Q-Learning for soccer robot system," *2000 robot soccer engineering workshop*
- [8] FIRA-KAIST Cup Workshop, "Robot soccer simulation competition platform based on multi-agent," *Federation of International Robot-soccer Association Harbin Institute of Technology, 2000*
- [9] F. R. Noreils, A. A. Recherche, and R. D. Nozay, "An architecture for cooperative and autonomous movile robots," *IEEE Proc. Int. Conf. on Robotics and Automation*, 2703-2710, 1992
- [10] I. H. Suh, H. J. Yeo, J. H. Kim, J. S. Ryoo, S. R. Oh, C. W. Lee, and B. H. Lee, "Design of a supervisory control system for multiple robotic systems," *Proc. IROS 96, 1996 IEEE*
- [11] J. Jo, *SimuroSot 5vs.5 : Robot Soccer Simulation Game*.
- [12] L. Shi, C. Jiang, Y. Zhen, and S. Zengqi, "Learning competition in robot soccer game based on an Adapted neuro-fuzzy inference system," *2001 IEEE. International Symposium on Intelligent Control*. September 5-7 Mexico City
- [13] L. Shi, Y. Jinyi, Y. Zhen, and S. Zengqi, "Multiple rewards fuzzy reinforcement learning algorithm in robocup environment," *Proceeding of the 2001 IEEE International Conference on Control Applications September 5-7, 2001 Mexico City, Mexico*
- [14] M.-W. Han and P. Kopacek, "Neural networks for the control of soccer robots," *2000 IEEE, ISIE/2000*, Cholula, Puebla, Mexico
- [15] P. Freedman, "Time, petri nets, and robotics," *IEEE transactions on robotics and automation*, vol. 7, no.4, August 1991, PP. 417-433
- [16] P. H. Lewis, "Introducing discrete-event control concepts and state-transition methodology into control system curricula," *IEEE transactions on education*, vol. 37, no. 1, February 1994, PP. 65-70
- [17] Q. Meng, X. Zhuang, C. Zhou, J. Xiong, Y. Wang, T. Wang, and B. Yin, "Game strategy based on fuzzy logic for soccer robots," *This research is supported by the project of National Natural Science Foundation of China 0-7803-6583-6/00/\$10.00 © 2000 IEEE*.
- [18] S.-W. Sun, J.-H. Choi, C.-W. Chung, and T.-Y. Kuc, "Embodiment of effective multiple-robot control algorithm using petri-net", *ICCAS 2002..*
- [19] S.-W. Park, J.-H. Kim, E.-H. Kim, and J.-H. Oh, "Development of a multi-agent system for robot soccer game," *Proceeding of the 1997 IEEE International Conference on Robotics and Automation Albuquerque*, New Mexico - April 1997.
- [20] T.-Y. Kuc, I.-J. Lee, and S.-M. Baek, "Perception, reasoning and learning of multiple agent systems for robot soccer," *IEEE Int. Conf. on SMC*, vol. 6, pp. 728-733, Tokyo, Japan, 1999.10.12
- [21] T. Arai and J. Ota, "Motion planning of multiple mobile robots," *IEEE Proc. Int. Conf. on Intelligent Robots and Systems*, 1761-1768, 1992



선승원

1974년 5월 22일생. 2001년 경일대학교 제어계측공학과 졸업 (공학사). 2003년 성균관대학교 정보통신공학부 전기전자 및 컴퓨터공학과 졸업 (공학석사). 현재 Feenix(주) 기술연구소 연구원으로 재직. 관심분야는 지능제어 및 Mobile로봇, 다

중로봇제어, Embedded System 등임.



국태용

1961년생. 1988년 서울대학교 제어계측공학과 졸업 (공학사). 1990년 포항공대 전자전기공학과 졸업 (공학석사). 1993년 포항공대 전자전기공학과 졸업 (공학박사). 1993년4-8월 삼성항공(주) 정밀기기연구소 주임 연구원. 1993년 9월-1995년

2월 목포대학교 전기공학과 전임 강사. 1995년-현재 성균관대학교 정보통신공학부 교수. 관심분야는 지능제어 및 지능로봇, 센서네트워크 등임.