

축구로봇을 위한 계층적 제어기

A Hierarchical Controller for Soccer Robots

이 인 재, 백 승 민, 손 경 오, 국 태 용

(In-Jae Lee, Seung-Min Baek, Kyung-Oh Sohn, and Tae-Yong Kuc)

Abstract : In this paper we introduce a model based centralized hierarchical controller for cooperative team of soccer-playing multiple mobile robots. The hierarchical controller is composed of high-level and low-level controllers. Using the coordinates information of objects from the vision system, the high-level controller perceives the configuration of working environment represented by the state sets which are simple models of multiple mobile robots on the playground. Subsequently, the high level controller selects an action model corresponding to the perceived state transition model and generates subgoal and goal-velocity, from which the low level controller generates trajectory of each wheel velocity of the robot. This two layered control structure makes fast action transition of mobile robots possible in real time because of its computational simplicity. The feasibility of the control strategy has been demonstrated in an implementation for real soccer games at a MiroSot league.

Keywords : robot soccer, multiple mobile robots, state sets, action sets, super-state model, action model, high-level controller, low-level controller

I. 서론

다중이동로봇 시스템의 연구 목적은 작업의 형태가 복잡하여 단일 로봇으로 작업수행이 불가능하고, 위험요소가 존재하는 작업에 대해서 다중이동 로봇을 이용함으로써 인간의 위험요소를 배제하고 인간의 도움 없이 작업을 수행하는 자율 다중이동로봇의 구현에 있다. 다중이동로봇이 응용 분야로 예상되는 생산, 위험물, 폭발물제거, 탐사 등의 분야는 변화가 복잡하고 다양한 동적 환경이며, 로봇이 수행해야 할 작업은 공간적, 시간적, 기능적으로 분산되어 있다. 이러한 환경의 변화와 작업의 분산성은 로봇이 작업을 효과적이고, 일관되게 수행하는 것에 장애요소가 된다. 또한 로봇이 주어진 환경에 대한 인식과 모델링, 주어진 동적 환경에서의 복잡한 작업에 대한 분석, 단순한 작업으로 나누는 것, 로봇에게 주어진 작업을 효과적으로 이행하는 것 등의 문제는 시간적인 지연요소와 다중이동로봇 시스템의 계산량의 증가를 일으키며, 실시간 제어에 제약이 된다. 또한 최종목적을 이루기 위한 작업의 분배는 각각의 작업수행 로봇의 역할과 기능이 충분히 고려되어야 한다.

관련 연구로써 Fabrice는 [11]에서 3개의 계층 (functional, control, planner level)으로 구성된 제어기로 서로 다른 기능의 두 대의 로봇의 협동으로 Box Pushing 하는 예를 보였으며, Lynne는 [3]에서 동기부여 (motivation)를 수학적으로 모델링한 Behavior based 제어 기법을 적용하여 로봇팀의 협동작업으로 오염물을 제거하는 예를 보였다. John은 [15]에서 sliding mode 기법을 활용하여 점근적 안정성을 갖는 group behavior 관한 연구를 하였다. Miguel은 [18]에서 작업공간에 대한 로봇간의 간섭을 고려한 협동작업의 효율성에 대한 연구를 하였다

[16]에서는 단일로봇의 슛팅과 드리블 등의 간단한 동작을 강화학습을 통하여 구현하였다. Arvin은 [14]에서 로봇이 경기를 진행할 때 진화연산을 통하여 향상된 결과를 보이는 것을 모아 실험을 통하여 구현하였다.

MiroSot(Micro Robot Soccer Tournament)[1][2][4][5][7][8][12][13][15]는 다중이동로봇 연구에 있어 서로 다른 로봇시스템간의 경쟁과 팀 로봇간의 협동 작업수행을 실험할 수 있는 중요한 실험 대상이 되고 있으며, MiroSot을 포함한 로봇 축구의 연구 목적은 상대팀을 고려한 동적 환경에서 팀 구성원간의 협동과 상대로봇과의 효과적인 경쟁을 통하여 게임에서 승리할 수 있는 자율 다중이동로봇을 구현하는 것이다 MiroSot는 위에서 설명한 일반적인 다중이동로봇의 환경과 유사한 변화가 다양한 동적 환경이며, 로봇이 수행해야할 작업 역시 시간, 공간, 기능적으로 분산되어 있다. 상대팀과의 경쟁은 다중이동로봇 제어에 있어 팀 로봇간의 협동과 함께 중요한 요소가 된다. 축구 경기에 있어 서로 반대 진영의 goalpost에 공을 넣기 위해 움직이는 두 팀의 로봇과 공의 움직임은 상황을 더욱 복잡하게 만들며, 상대 로봇들이 움직이는 전략과 공의 움직임 역시 예측하기 어렵다. 위와 같은 환경의 다양한 변화와 복잡성은 동적 환경의 정확한 인식면에서 문제가 되며, 대부분의 다중 이동 로봇의 제어에 있어서 주로 고려되는 문제인 로봇간의 협동[3][11] 이외에 축구로봇의 경우에는 상대로봇과의 경쟁도 고려해야 할 또 하나의 문제가 된다. [12]-[15] 따라서 본 논문에서는 계산량의 부담이 적고 구조적으로 간단한 state-action set과 super-state에 기초한 finite state machine과 state-action model을 적용한 계층적 제어기를 소개한다. state set과 super-state model [1][3]은 경기장의 상황에 따른 로봇의 관계를 모델링하고, super-state model과 state set에 대응되는 action model은 로봇의 빠른 동작과 상황 변화에 대한 로봇간의 역할의 변화를 유도하여 복잡하고 예측하기 어려운 환경

접수일자 : 1999. 7. 27., 수정완료 : 2000. 2. 16

이인재 : (주)메디슨 X-ray 연구소

백승민, 국태용 : 성균관대학교 전기·전자 및 컴퓨터공학부

손경오 : (주)대우통신 연구소

에 효과적인 대응을 가능하게 한다. 전체 제어기의 계층구조는 상위 및 하위제어기로 구성되며 상위제어기는 동적 환경에서의 로봇의 상태를 인식. 이에 적합한 동작에 대한 목적 벡터를 생성하게 된다. 하위제어기는 상위제어기에서 전달된 목적 벡터를 이용하여 로봇의 속도를 구하게 된다.

상위 제어기는 action set[1][12]과 action model을 도입하여 로봇이 움직여야할 경로를 직접 계산하지 않고 state set을 적용한 super-state model로부터의 정보만으로 로봇이 상황에 따라 위치 해야할 좌표와 이에 도달하는 최대속도로 구성된 목적 벡터를 생성한다. 이러한 action set은 attack, wait, blocking, checking의 네 가지 동작으로 구성되며 각각의 action은 목적 벡터를 정하는 규칙을 갖는다. action set으로부터 구성된 action model은 세 가지로 구분되는 모드에 따라 공격과 수비의 적절한 행동을 결정한다. 하위 제어기에서는 상위 제어기에서 결정된 목적 벡터를 이용하여 로봇 바퀴의 속도제어를 생성하여 각각의 로봇을 구동한다. 따라서 제안된 제어기는 동적 환경을 모델링하고 동작의 기준을 제공하는 상위 제어기와 이에 따라 속도제어를 생성하는 하위제어기의 간단한 계층구조로 구분되며, 본 논문에서 제안된 제어기의 실시간 제어의 효율성과 목적달성을 위한 로봇간의 협동 동작에 있어 그 효율성을 보인다.

논문의 구성은 II절에서 계층적 제어기의 상위제어기에서의 환경인식과 action에 대하여 설명하고, III, IV절에서 로봇간의 협동과 역할교환, 다중 이동로봇 제어기를 설명한다. V, VI절에서는 구현된 시스템을 소개하고 구체적인 실험결과에 대하여 기술함으로써 제안된 제어기의 효율성을 보인다.

II. 계층적 제어기

제안된 계층적 제어기는 상위 제어기와 하위 제어기로 구성되어 있으며, 상위 제어기에서는 축구장에서의 동적 환경을 실시간 제어에 적합하도록 빠르고 효과적으로 인식하고 판단하기 위하여 16개의 state로 각각의 로봇의 상황을 정의한다. state set을 이루며 이러한 16개의 state는 로봇의 좌표, 방향, 공과 로봇의 각도 등의 기본적인 정보를 바탕으로 동적 환경을 모델링하는 데 사용된다. 또한 로봇간의 동작의 조합으로 이뤄지는 두 대 이상의 로봇의 협동 작업에서는 state set의 조합으로 구성된 super-state set을 정의하고 이러한 super-state set은 축구경기의 공격 및 수비등 다중로봇의 협동 작업에서 로봇간의 역할분담 및 역할교환을 가능하게 한다. 네 개의 기본 동작으로 구성된 action set에서 각각의 로봇의 상황에 적합한 동작을 생성하게 되는데 action model은 이를 구현하는 바탕이 된다. 상위 제어기에서의 동작이 결정되면 하위 제어기에 전달될 목적벡터가 구해지며, 목적 벡터를 구성하는 요소는 로봇이 이동해야 할 위치와 동작에 따라 정의된 최대속도가 된다. 목적벡터는 로봇이 공을 추적하거나 슈팅, 또는 상황에 적합한 지점으로 이동하는데 사용되며, 로봇의 동작의 정확성에 영향을 미친다. 하위 제어기에서는 상위 제어기에서 전달된 목적벡터를 바탕으로 로봇의 속도제어를

생성하고 서보 및 피드백 제어기로 로봇을 구동한다. 이러한 간단한 구조의 계층적 제어방식의 적용은 경기장의 변화하는 환경에 대하여 신속한 인식과 판단, 명령 전달을 가능하게 하며, 축구 경기의 경우 로봇으로 하여금 goal을 성공시킬 수 있는 유리한 위치로 움직이도록 유도할 수 있다.

1. 동적 환경 모델

1.1. State set

축구로봇의 제어에서 팀 전술의 목적은 크게 두 가지로 나눌 수 있는 데 상대팀의 골대에 공을 넣는 것과 상대팀 공격을 효과적으로 방어하는 것이다. 따라서 로봇의 위치와 공의 위치를 빠르게 인식, 추적하고 이를 바탕으로 팀 전술을 구사해야한다. 축구장에서 움직이는 객체는 MiroSot의 경우 팀 로봇 3대, 상대팀 로봇 3대 그리고 공으로 7개인데 일반적으로 상대팀의 로봇의 이동과 운동장의 경계인 벽과의 충돌, 공의 불연속적이고 불규칙한 움직임들에 대한 예측은 매우 어려운 일이다. 따라서 복잡한 환경을 단순화하여 모델링하기 위하여 16개의 state으로 이루어진 state set를 정의하기로 한다. state set내의 각 state는 경기장, ball, 로봇의 위치에 따라 유리한 상태와 불리한 상태. 또는 공격 가능한 상태와 수비 가능한 상태로 경기장의 환경을 모델링하고 빠르게 인식하게 된다. 각각의 state는 상대의 골대와 공, 로봇간의 상대적 위치에 따라 환경을 모델링하게 되는 데 이는 positive, negative 또는 singular 구분되며, positive state는 goalpost-ball-robot (GBR)과 같이 상대적 위치 관계를 가진 경우이고 로봇이 shooting이나 상대 goalpost로 공을 물고 갈 수 있는 유리한 상태를 나타내며, 그림 1은 그 한 예를 보여주고 있다. negative state는 positive state에서 ball과 robot의 위치가 바뀐 것으로 goalpost-robot-ball (GRB)의 관계를 나타내며 로봇이 공격보다는 수세인 불리한 위치에 있음을 나타내고 그림 3에서 보여주고 있다. 그림 3에서 OR은 상대방 로봇을 나타낸다. singular state는 ball과 robot의 x축 방향좌표가 같은 경우로 goalpost-robot=ball (GR=B) 순서로 공세를 취할 수 있는 positive state와 negative state의 중간 단계를 나타내며, 그림 2와 같은 경우가 해당된다. singular state역시 로봇이 공세를 취할 수 있는 유리한 상황은 아니다.

$$\begin{aligned} S_P &= \{GBR_C, GBR_R, GBR_L\} \\ S_N &= \{GRB_C, GRB_R, GRB_L\} \\ S_S &= \{GB=R_R, GB=R_L\} \end{aligned} \quad (1)$$

여기서 S_P, S_N, S_S 는 positive, negative, singular state를 나타내며, 첨자 C, R, L은 G-B선을 기준으로 각각 중앙, 오른쪽, 왼쪽을 나타낸다. 또한 위 세 집합의 변형으로 상대측 로봇 또는 장애물(상대로봇)이 개입되는 경우를 다음과 같은 state의 집합을 정의하기로 한다.

$$\begin{aligned} S_{PO} &= \{GBOR_C, GBOR_R, GBOR_L\} \\ S_{NO} &= \{GROB_C, GROB_R, GROB_L\} \\ S_{SO} &= \{GB=O=R_R, GB=O=R_L\} \end{aligned} \quad (2)$$

O는 상대팀의 로봇을 나타내며 위와 같은 state에서는 장애물을 회피하거나 상황에 따라 상대로봇을 blocking 할 필요가 있다. 따라서 전체 state는 16개이며 3대의 로봇에 대하여 매 vision sampling 시간마다 각 로봇의 상태가 인식되고 결정된다.

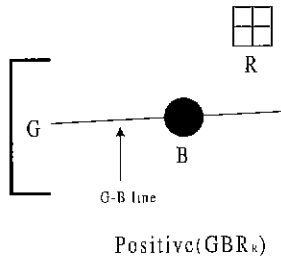


그림 1. Positive 상태.
Fig. 1. Positive state.

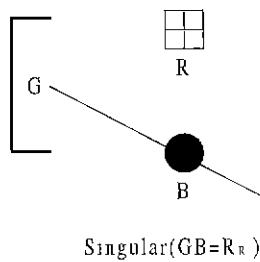


그림 2. Singular 상태.
Fig. 2. Singular state.

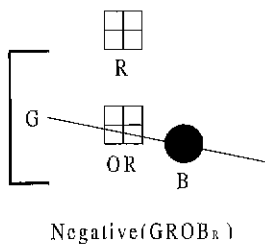


그림 3. Negative 상태.
Fig.. 3. Negative state.

1.2. Super-state model

Super-state model은 경기중에 발생할 수 있는 고정된 지역에서 움직이는 keeper로봇을 제외한 나머지 로봇의 상태의 조합이며, 공격과 수비시에 역할분담 및 교환의 조건이 된다. Super-state의 변화와 현재의 Super-state는 각각의 로봇에게 추가적인 조건이 필요 없는 반사적인 action의 변화와 서로의 역할 교환을 가능하게 한다. 따라서 공의 움직임과 상대로봇의 움직임에 따라 변화하는 환경에 대해 로봇간의 반사적인 역할 교환은 상대 로봇의 움직임에 대해 반사적인 대응을 가능하게 한다.

$$\begin{aligned}
 T_1 &= \{R_1[P] \& R_2[M|S]\} \\
 T_2 &= \{R_1[M|S] \& R_2[P]\} \\
 T_3 &= \{R_1[P] \& R_2[P]\} \\
 T_4 &= \{R_1[M|S] \& R_2[M|S]\}
 \end{aligned}
 \tag{3}$$

$$T_m = \{T_1, T_2, T_3, T_4\} \tag{4}$$

(3)에서 P, S, N은 각각 positive, singular, negative state를 나타내며, $R_i[j](i=1,2)$ 은 로봇의 현재 state, $T_j(j=1,2,3,4)$ 는 Super-state model을 정의 한 것이며, 전체 Super-state 집합(T_m)은 (4)에서 4개의 super-state로 구성된다.

2. Action model

2.1 Action set

본 절에서는 동적 환경의 모델링에 따른 로봇의 action과 action을 구성하는 세부 동작의 순환과정을 상태도를 이용하여 설명한다. 그림 4, 5, 6에서 P*는 action의 세부동작을 나타내며, t는 각각의 세부동작의 실행 조건이 된다. 표 1과 2에서 세부동작의 의미와 각각의 조건을 설명하고 있다.

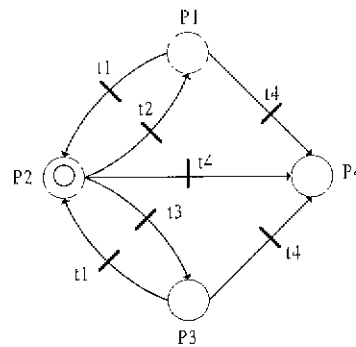


그림 4. attack의 상태도.
Fig. 4. State diagram of attack.

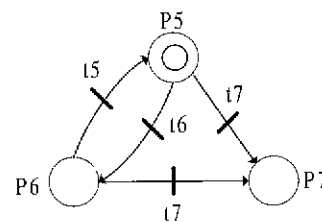


그림 5. wait의 상태도.
Fig. 5. State diagram of wait.

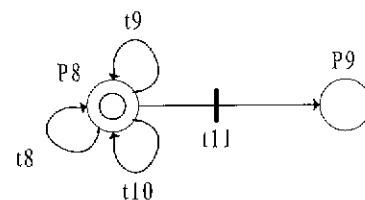


그림 6. Blocking의 상태도.
Fig. 6. State diagram of blocking.

action set 경기중에 발생할 수 있는 로봇의 동작을 각각의 특징 별로 정의한 집합이다. 여기서 action set은 attack, wait, blocking, checking, change state으로 구성되며 필요에 따라 세분화 할 수 있다. action set의 역할은 각각의 로봇의 상태에 따라 현재 상태에서 보다 바람직한 상태 즉, 공격 가능한 상태로 로봇을 동작시킬 수 있는 목적

벡터를 생성하는 것이다. 각각의 action별로 목적벡터를 정하는 규칙은 다음과 같이 정의된다. attack의 목적벡터는 shooting을 하거나 공을 상대 goalpost로 몰기 위하여 공의 좌표를 고려하여 반듯이 로봇이 위치해야 할 곳에 생성된다. attack을 이루고 있는 동작은 shooting, dribble, approach ball, change state이며, 그림 4에서 attack의 세부동작에 대한 순환과정을 나타내고 있다. wait의 목적벡터는 공격 로봇의 역할 분담 및 전환시 공격 할 수 있는 대기 지점에 생성된다. 그러나 wait에서도 shooting 조건이 만족되면 shooting을 하기 위한 목적벡터를 생성할 수 있다. wait의 순환은 그림 5에서 보여주고 있다 또한 wait는 공격시 로봇 상호간의 간섭과 충돌을 방지하고 연속적인 attack과 상대 로봇이 공을 몰고 중앙선을 통과하는 것을 대응하는 기능을 한다. blocking은 상대 로봇이 goalpost 주변에서 ball에 접근하는 것을 막거나, shooting하기 위한 경로는 차단하는 위치에 목적벡터를 생성하며 그림 6에서 과정을 보여주고 있다. checking은 상대 로봇과의 유리한 위치를 차지하기 위한 경쟁으로써, 목적벡터는 상대 로봇의 현재 위치가 된다. attack중 shooting동작의 목적벡터는 로봇과 공간의 거리와 상대 goalpost와 공과 로봇의 각도에 따라 연속적으로 생성되는 목적벡터로 이루어진다. action set에서의 주목할 것은 각각의 로봇의 이동해야할 경로를 생성하는 것이 아니라, 로봇의 순간 순간의 변화하는 환경에서 유리한 지역으로의 이동과 위치와 그 최대 이동 속도만을 결정한다는 것이다. 표1, 2에 로봇 상태에 따른 action을 결정하는 조건의 예를 볼 수 있다. 주어진 예는 MiroSot 경기에 실제 적용된 자료이다. 표 1에서 dist_RBx와 dist_RBy는 각각 로봇과 공간의 x, y축의 거리를 나타내며, R_orient는 로봇의 방향을 나타낸다.

표 1. 동작의 상태와 조건

Table 1. State and condition of action.

	condition	state	action
t1	$10 < \text{dist_RBx} < 15 \ \& \ 10 < \text{dist_RBy} < 15$	P	attack
t2	$\text{dist_RBx} < 10 \ \& \ \text{dist_RBy} < 10 \ \& \ (0 < \text{R_orient} < 80 \ \parallel \ 280 < \text{R_orient} < 360)$	P	attack
t3	$\text{dist_RBx} > 15 \ \parallel \ \text{dist_RBy} > 15$	P	attack
t4	로봇 상태가 positive에서 다른 상태로 변했을 때	S, N	attack
t5	attack 로봇의 상태에 변화가 없을 때	P	wait
t6	$(\text{dist_aRBx} > 15 \ \parallel \ \text{dist_aRBy} > 15) \ \& \ (0 < \text{R_orient} < 80 \ \parallel \ 280 < \text{R_orient} < 360)$	P	wait
t7	attack로봇의 상태가 변했을 때	P	wait
t8	공과 상대로봇이 goalpost에 가까이 있고 positive일 때	PO	blocking
t9	공과 상대로봇이 goalpost에 가까이 있고 singular일 때	SO	blocking
t10	공과 상대로봇이 goalpost에 가까이 있고 negative일 때	NO	blocking
t11	공이 goalpost로부터 멀어질 때	P	blocking

표 2. 동작에 따른 상태.

Table 2. State of action.

	로봇의 동작	state
P1	Shooting	P
P2	Dribble	P
P3	Approach ball	P
P4	Change action	S or N
P5	Wait	P
P6	Shooting	P
P7	Change action	P
P8	Blocking position	P, N or S
P9	Change action	P

2.2 목적 벡터의 생성

각각의 action에서 attack, wait, blocking, checking, change state를 위한 목적 벡터는 $V_a, V_w, V_b, V_c, V_{cs}$ 각각 다음과 같이 정의된다. 그림 7은 축구장의 좌표계를 보여주고 있다.

$$V_a = (x_a, y_a, v_{maxa})$$

$$(x_a(k+1), y_a(k+1)) = (B_x(k) + \delta_a, B_y(k))$$

IF shooting

$$(x_a(k+1), y_a(k+1)) = (x_{optg}(k), y_{optg}(k))$$
(5)

$$V_w = (x_w, y_w, v_{maxw})$$

$$(x_w(k+1), y_w(k+1)) = (B_x(k) + \delta_w, \delta_r \text{ if } B_y(k) \geq y_0 \text{ else } \delta_r)$$
(6)

$$V_b = (x_b, y_b, v_{maxb})$$

$$(x_b(k+1), y_b(k+1)) = (x_b, B_y(k) + \delta_b)$$
(7)

$$V_c = (x_c, y_c, v_{maxc})$$

$$(x_c(k+1), y_c(k+1)) = (R_{ox}(k), R_{oy}(k))$$
(8)

$$V_{cs} = (x_{cs}, y_{cs}, v_{maxcs})$$

$$(x_{cs}(k+1), y_{cs}(k+1)) = (R_{xc}(k), R_{yc}(k))$$
(9)

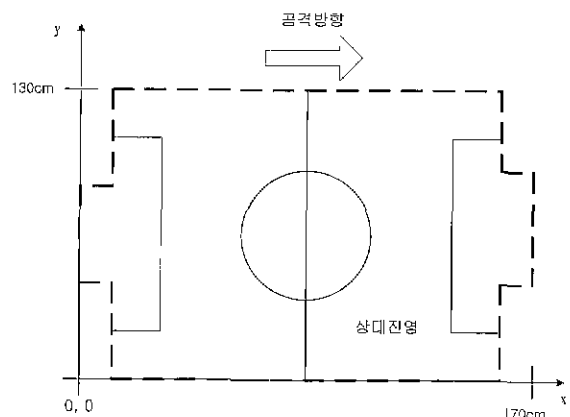


그림 7. 축구장 좌표계.

Fig. 7. coordinates of soccer play ground.

여기서 $\delta_a, \delta_w, \delta_r, \delta_l, \delta_b$ 는 조절 가능한 상수로서 x, y 좌표 임의의 값을 가지며, 이는 실험을 통해서 조정되고,

제어기의 내부의 특성에 따라 서로 다른 값을 가질 수 있다. 예를 들어 목적 벡터가 공을 가리킬 때 δ_a 는 0이 된다.

V_{oa} 는 shooting과 상대 진영으로 공을 몰고 갈 수 있는 지점에 설정되며 B_x 는 공의 x좌표이고 δ_a 는 음의 실수 값(-10)이다. x_{obpg}, y_{prg} 는 각각 V_a 가 가리키는 상대진영 goalpost의 x좌표와 현재 shooting 할 로봇의 y좌표가 고려된 상대진영 goalpost의 projection된 y좌표를 나타낸다. (7)에서 V_w 는 공의 움직임을 고려하여 x좌표는 (-40)이 된다. y좌표는 y축의 중앙(y_0)을 기준으로 공이 있는 반대쪽 좌표가 된다. δ_l, δ_r 는 y축 중앙을 기준으로 한 대칭점이다. 따라서 로봇은 공과 일정한 거리를 두고 이동하게 된다. V_b 는 goalline 내에서 움직이는 공을 중심으로 두 대의 로봇이 좌우 일정 간격을 유지하며, 상대 팀 로봇이 공에 접근하는 것을 막는다. x의 좌표는 로봇이 goalline으로 들어가지 않는 지점에 위치시킨다. V_c 는 공격 할 때 수비하는 상대로봇에 의해 공을 원하는 방향으로 몰지 못할 때 벡터는 상대로봇 좌표 $R_{ox}(k), R_{oy}(k)$ 를 가리키게 된다. (9)에서 $R_{xc}(k), R_{yc}(k)$ 는 현재 action에서 변화될 action의 목적벡터의 좌표를 말한다. 각각의 목적벡터에 포함되어 있는 $v_{max,x}$ 는 각각의 action에서의 최대 속도를 나타낸다. 따라서 동작의 목적과 특성에 따라서 정의 하게되는데, attack일 때와 change action이 일 때 $v_{max,x}$ 는 로봇이 낼 수 있는 최대속도가 되며, wait일 때 $v_{max,x}$ 는 중간속도, blocking에서의 $v_{max,x}$ 는 goalline 근처에서의 빠른 움직임으로 자살골을 막기 위해 최저속도를 갖게 된다.

그림 8은 일반적인 공격상황의 목적 벡터의 변화를 나타낸 것이다. 첫 번째의 attack 목적벡터는 공의 뒤에 위치하다 dribble 조건 t1이 만족되면 목적벡터는 공의 중심으로 바뀌고 shooting조건 t2가 만족되면 상대방의 goalpost로 subgoal이 바뀌게 된다. wait subgoal은 공의 움직임에 따라서 1-2-3의 순서로 바뀌게 된다. 그림 9는 R1이 positive에서 singular나 negative로 바뀌게 되는 상황(불리한 상황으로의 변화)에서 R1의 action은 attack에서 wait로 바뀌며 R2는 wait에서 attack으로 바뀌게 된다. 반

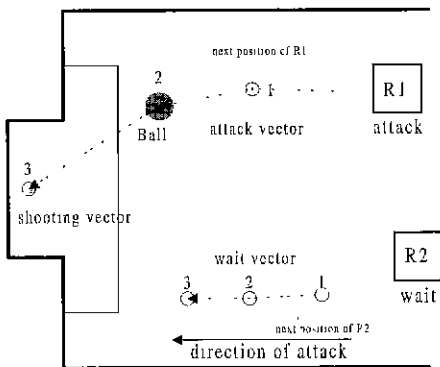


그림 8. 목적 벡터의 변화.
Fig. 8. Transition of object vector.

사적인 역할의 변화는 로봇간의 목적벡터를 변화시킴으로써 구현된다. attack을 구성하는 dribble, approach ball의 동작에서의 목적 벡터의 생성은 제어기 내부에서 정의된 δ_a 에 의해 (5)에 의해 구할 수 있다.

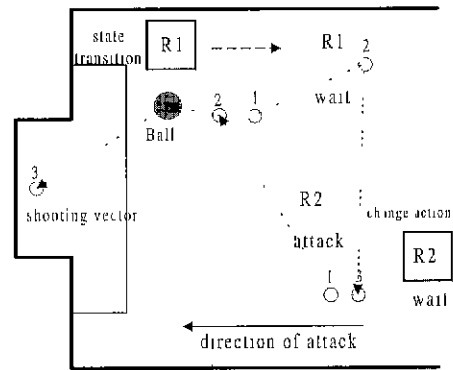


그림 9. state변화와 action의 변화.
Fig. 9. State transition and action generation.

III. 협동과 역할 교환

앞에서는 축구장의 동적 환경에 대한 인식과 이러한 인식된 환경에서의 로봇의 동작과 동작의 구현을 위한 목적 벡터의 생성에 대하여 설명하였다. 본 절에서는 로봇의 state변화에 대응하는 서로 다른 동작으로 이루어지는 역할분담, 협동 및 전환에 대하여 알아본다. 3대의 로봇이 팀을 이루는 MiroSot 경기의 일반적인 전략은 한 대의 로봇은 keeper역할을 하며 나머지 두 대의 로봇이 상황에 따라 공격과 수비를 하게 된다. keeper를 제외한 나머지 두 대의 로봇의 제어는 공격과 수비시 팀 로봇간의 간섭을 최소화하며 로봇 팀의 성능과 기능을 충분히 활용할 수 있도록 해야 한다. 먼저 공격 할 때 두 대의 로봇이 서로 가까운 지역에서 공을 목표로 움직인다면 로봇간의 충돌로 인해 공격을 하는데 장애가 될 것이다. 따라서 본 논문에서 사용되는 협동전략은 한 대의 로봇이 공격중(attack) 일 때 나머지 로봇은 대기(wait) 하도록 함으로써 팀 로봇간의 충돌을 억제하고, wait 로봇은 상대로봇의 공격에 대응하게 된다. 로봇간의 역할의 교환은 attack 로봇의 상태 변화로써 이뤄지는데, 역할 변화를 하지 않고 state 변화가 일어난 attack 로봇에게 공을 추적하게 하면, 공을 위협한 지역까지 오게 할수 있으며, 공을 몰고 공격해오는 상대로봇의 뒤를 추적하는 불리한 상황이 발생하게 된다. 역할교환의 목적은 보다 유리하고 공격이 용이한 위치에 있는 wait 로봇의 attack으로의 변화로 연속적인 공격을 가능하게 하는 것이다. 이러한 두 대의 로봇의 동작은 action model나 타내지며, 그림 9에서 그 예를 보았다.

그림 10에서 super-state를 이용한 action model의 순환 과정을 보여주고 있으며, action model은 각각 서로 다른 Super-state과 대응되어 있고, attack, wait, blocking 3개의 action 이 각각의 로봇이 수행해야 할 동작의 조합으로 keeper를 제외한 두 대의 model은 action model1(R1(a), R2(w)), action model2(R1(w), R2(a)), action model3

(R1(b), R2(b))로 나타낼 수 있다. action model은 Super-state의 상태에 따라 3가지의 action model로 반사적인 변환을 일으킨다. Super-state (T3)에서는 두 대 로봇이 공통적으로 positive 이기 때문에 공으로부터 거리가 가까운 로봇이 attack이 된다.

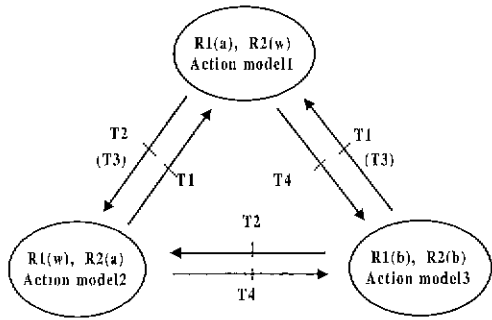


그림 10. Action model의 상태도.
Fig. 10. State diagram of action model.

a, w, b는 각각 attack, wait, blocking을 말하며, R1(*), R2(*)는 각 로봇이 수행 할 동작을 나타낸다. action model의 결정은 매 vision 샘플링 시간의 Super-state model에 따라서 결정된다.

$$\{ \{ R_1 = S_P \} \& \{ R_2 = S_P \} \} \& \{ \overline{R_1 B} \& \overline{R_2 B} \} \Rightarrow R_1(attack), R_2(wait) \quad (10)$$

$$\{ \{ R_1 = S_S \} \& \{ R_1 = S_N \} \} \& \{ R_2 = S_P \} \Rightarrow R_1(wait), R_2(attack) \quad (11)$$

$$\{ B_x < \delta_b \} \Rightarrow R_1(blocking), R_2(blocking) \quad (12)$$

R_1, R_2 는 로봇, $\overline{R_1 B}, \overline{R_2 B}$ 는 로봇과 공의 직선 거리를 나타낸다. (10)은 R_1, R_2 가 모두 positive이고 R_1 의 action은 attack이고, R_2 의 action은 wait가 된다. (11)에서는 R_1 이 singular 또는 negative이고 R_2 가 positive 일 때 R_1 의 action은 wait이고 R_2 의 action은 attack이 된다. (12)는 공의 x좌표가 δ_b (약 15cm)보다 작을 때 R_1, R_2 의 action은 동시에 blocking으로 결정되어 상대로봇이 공에 접근하는 것을 막는다 checking은 상대 진영에서 공격 상황에서 상대 로봇이 공격에 장애가 될 때 선택된다. 만약 R_1 이 attack이고 R_2 이 wait일 때 R_1 의 state에 변화가 있다면 R_2 의 action은 attack으로 바뀌며 R_1 은 wait가 된다.

IV. 계층적 제어기의 구현

1. 상위 제어기

동적 환경의 인식과 로봇의 동작의 기준과 역할을 제시 하는 상위제어기는 1/60 초 마다 갱신되는 state와 super-state에 현재 상황에 적합한 동작을 결정하며 그에 해당하는 목적 벡터를 생성하게 된다. state set은 단일 로봇의 action

결정의 판단기준이 되며, state set의 합성으로 이뤄진 super state set는 두 대 이상의 로봇의 협동작업에서의 역할 분담과 역할교환의 판단기준이 된다. 단일로봇의 action 판단과 로봇간의 협동에서 역할의 판단과 교환을 위한 state set의 사용은 동적 환경의 신속한 인식에 있어 추가적으로 또 다른 판단요소를 생성하지 않아도 되므로 효과적이다.

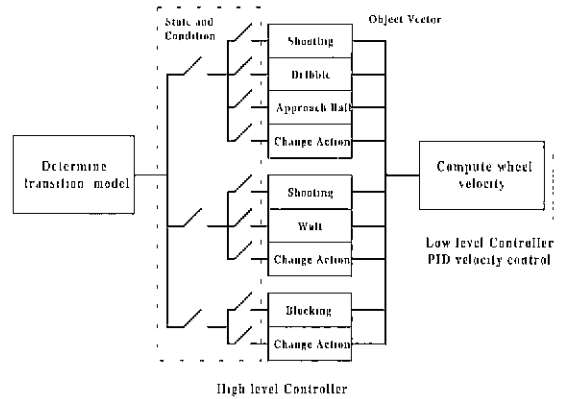


그림 11. 단일 로봇의 제어기.
Fig. 11. Controller structure of single robot.

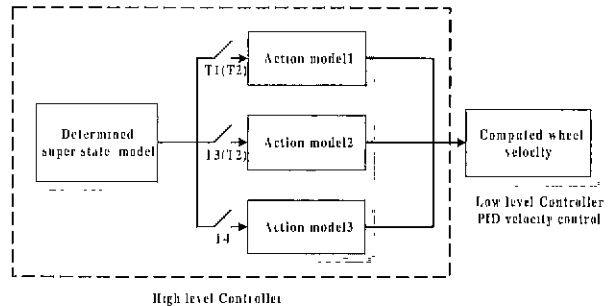


그림 12. 두 대 이상의 로봇 제어기.
Fig. 12. Controller structure of two robots.

로봇간의 협동은 공격과 수비 시 서로 다른 역할 분담과 교환으로 이뤄지는데, 공격할 때는 한 대의 로봇은 attack 동작을 수행하고 다른 한 대의 로봇은 wait를 수행한다. 이러한 전략은 공격상황에서 흔히 발생하는 attack로봇의 공격의 실패에 대해 wait로봇의 연속된 공격은 goal을 성공시킬 수 있는 가능성을 증가시킨다. 마찬가지로 공격 실패시 상대로봇이 공을 몰고 나올 때 wait 로봇은 상대 로봇의 경로를 차단하게 하므로 상대팀의 빠른 공격을 중간에서 막는 역할을 한다. blocking은 공이 상대 로봇의 공격으로 인해 goalline 안으로 들어올 때 두 로봇은 사이의 공을 중심으로 상대방로봇의 공에 대한 접근을 막는다. 제어기의 동작은 action set을 바탕으로 각각의 action model을 구성하고 각각의 action model들은 super state의 천이 모델에 따라 반사적으로 바뀌게 된다. 그림 11은 state-action set을 이용한 단일 로봇의 제어기 구조이며, 그림 12는 super-state과 action model을 이용한 두 대 이

상의 로봇제어기의 구조로써, 그림12의 제어기에서는 state-action set를 확장한 super-state과 action model을 이용함으로써 단일 로봇제어기를 확장한 형태이다.

2. 하위 제어기

하위 제어기는 상위 제어기로부터의 각각의 목적 벡터를 전달받아 로봇이 이동할 속도 궤적을 생성한다. 계산에 필요한 선가속도(v_{acc})와 각가속도(ω_{acc})는 실험을 통해 로봇이동에 무리가 없는 값으로 설정하며, 각 속도의 최대 값을 각각 결정한다. 가속도와 회전속도의 최대 값이 너무 큰 값을 갖게 되면 로봇 동작이 부정확해질 수 있으며 작은 값은 로봇의 움직임을 느리게 한다. 그림 4는 로봇이 회전해야 하는 각도를 결정하기 위한 그림이며, 로봇의 orientation을 θ , 로봇의 중심으로부터 목적벡터(subgoal) 간의 각을 ϕ 라고 하면 로봇의 회전각 ϕ 는 다음과 같다.

$$\phi = \theta - \varphi \tag{13}$$

$$\omega[t] = \omega[t-1] + \omega_a \tag{14}$$

$$v[t] = v[t-1] + v_a \tag{15}$$

$$\omega_r[t] = \frac{v[t] + W \cdot \omega[t]/2}{r} \tag{16}$$

$$\omega_l[t] = \frac{v[t] - W \cdot \omega[t]/2}{r} \tag{17}$$

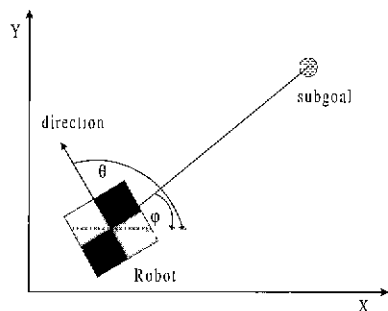


그림 13. 로봇의 이동.
Fig. 13. Movement of robot in the plane.

(13)에서 ϕ 가 결정된다. 따라서 로봇의 이동방향은 ϕ 를 0이 될 수 있는 방향으로 회전속도를 결정한다. 회전속도는 위의 기준에 의해 ω_a 의 증감으로 결정되며, 최대 각속도 ω_{max} 를 초과하지 않는다. 선속도는 현재의 선속도에 정의된 가속도 v_a 를 증감으로 결정된다. 선속도의 최대 값은 각각의 목적 벡터의 v_{max} 가 된다. (14), (15)로부터 각속도와 선속도가 결정되면 양쪽바퀴의 속도는 (16), (17)로 얻을 수 있다. ω_r , ω_l 은 양쪽 바퀴의 속도이고, W는 양쪽 바퀴간의 거리, r은 바퀴의 반지름이다.

3. 로봇 축구시스템 구현

제작된 축구시스템(KINGGO)의 전체 구성은 계층적 제어기, vision system, RF모듈, robot으로 구성되어 있으며, vision system은 처리속도 향상을 위하여 계층적 제어기를 담당하는 중앙제어부와 분리하여 독립적인 PC로 구성하였다. 그림 14는 시스템의 구성을 보여주고 있고, 그림 15는 각부분별 기능을 보여주고 있다.

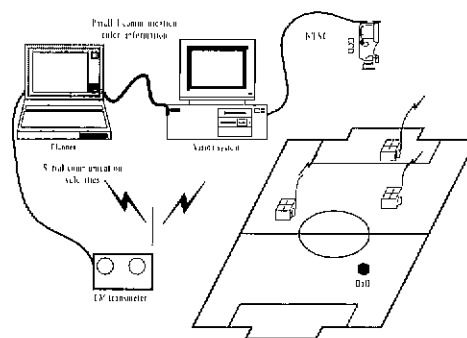


그림 14. 전체 로봇 시스템.
Fig. 14. Overall soccer system.

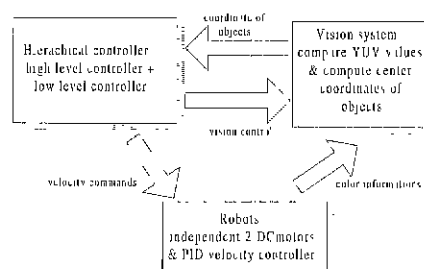


그림 15. 축구로봇 시스템의 부분별 기능.
Fig. 15. Functional flow of Robot soccer system.

3.1 Robots

축구로봇은 경기중에 벽과의 충돌이나 상대 로봇과의 충돌등 비정상적인 상황을 견뎌야한다. 따라서 로봇의 제어기와 기타 장치들의 설계와 제작은 위와 같은 문제를 고려하여 하는 것이 바람직하다. KINGGO 로봇은 CPU로는 80196KC를 사용하였으며, CPU에서는 2ms의 샘플링 타임을 갖는 PID 속도제어기를 동작시키며, 정격 1.6W의 모터는 14:1의 감속비와 1회전당 16pulse의 분해능의 엔코더를 갖고 있다. 로봇의 제어 가능한 속도는 157cm/s로 설계되었다.

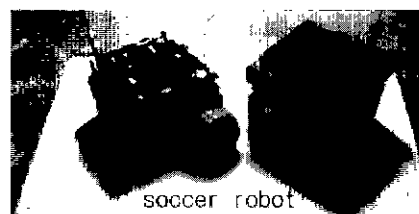


그림 16. KINGGO 로봇.
Fig. 16. KINGGO robots.

3.2 Vision System

KINGGO vision system은 240x180의 해상도에서 60fields/sec속도로 영상의 YUV값을 비교하여 팀칼라, 각각의 로봇 칼라, 상대팀의 칼라를 찾아 각각의 객체의 중심좌표를 추출하여 상위 제어기에 전달한다. vision system의 특징은 PC기반의 software로 구현되었으나, 비교적 정확하고 빠른 속도를 갖고 있다. 또한 칼라를 추출 알고리즘을 최적화하여 vision 데이터에 노이즈가 섞인

상황에서도 찾고자 하는 색을 정확하게 찾을 수 있도록 하였다. 전체 vision system은 Pentium PC와 overlay카드, windows capture driver를 이용한 프로그램으로 구성되었다.

V. 실험

그림 17, 18, 19, 20에서 사각형은 로봇의 이동 궤적이고, 중심으로부터의 직선은 목적벡터를 나타내며, 원은 공을 나타낸다. 로봇과 공의 위치는 매 10 비전 샘플마다 표시했으며, 비전의 처리 속도는 60fields/sec이다 그림 17, 18에서 로봇은 positive 상태이고 상대 goalpost에 shooting을 할 수 있는 위치이다. 따라서 목적 벡터는 공에 접근하고, 이어 shooting 방향으로 생성되었으며 로봇의 shooting동작에서는 공을 들고 최대 속도로 goalpost를 향하는 것을 보여주고 있다. 그림 19, 20은 로봇이 negative state로 불리한 상황이므로 이를 회피하기 위한 목적 벡터의 생성과 state가 바뀌면서 목적 벡터는 그림 17, 18에서의 목적벡터 생성과 같게 생성된다. 그림 21은 negative state에서 공이 이동하고 있을 때 공을 추적하고, state에 따라 변화하는 목적벡터를 보여주고 있다 그림 17~21의 결과 그림들을 통해 로봇과 공의 위치에 따른 상태판단과

행동결정이 합리적으로 이루어지고 있음을 알 수 있었으며, 이 밖에도 로봇과 공을 임의의 위치에 두었을 때 로봇이 공을 건드리는 시간을 측정해 보았다. 실제 축구경기에서는 이 시간이 공격과 수비를 하는 반응 시간으로 볼 수 있으므로, 이 시간이 짧을수록 경기의 주도권을 우리 팀이 가질 확률이 높다고도 볼 수 있다. 측정결과 공을 가장 늦게 치는 경우가 3초 정도로 측정됐다 즉 3초 이내에는 공을 막거나 슛동작을 할 수 있다는 결과다.

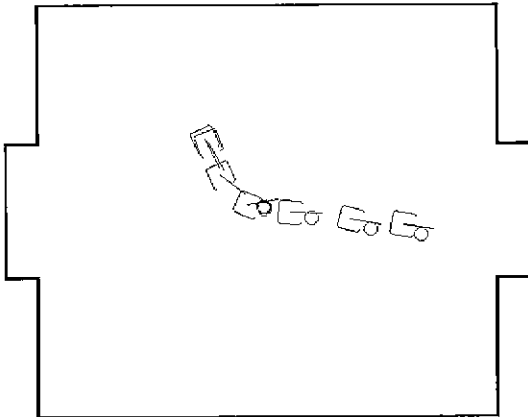


그림 17. Positive state 일 때 로봇의 동작(1).
Fig. 17. Robot motion for positive state(1)

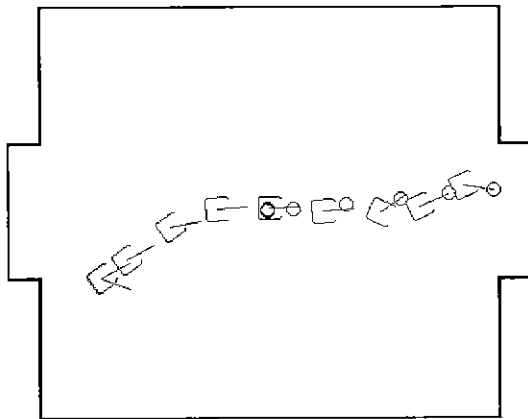


그림 18. Positive state 일 때 로봇의 동작(2).
Fig. 18. Robot motion for positive state(2).

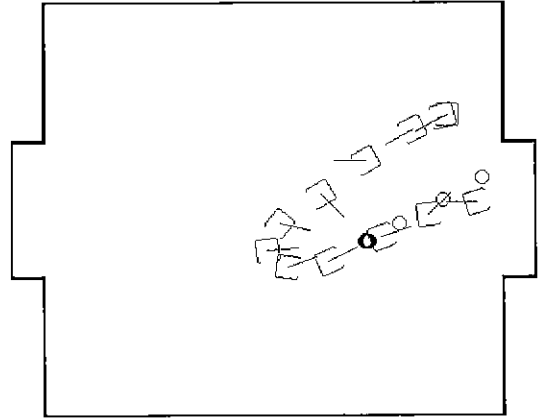


그림 19 Negative state 일 때 로봇의 동작(1).
Fig. 19. Robot motion for negative state(1).

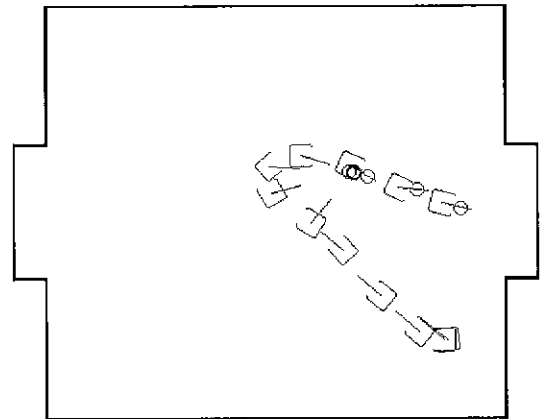


그림 20. Negative state 일 때 로봇의 동작(2).
Fig. 20. Robot motion for negative state(2).

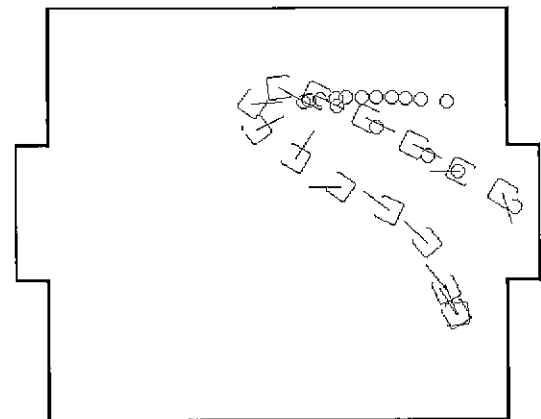


그림 21. 이동하는 공에 대한 로봇의 동작.
Fig. 21. Robot motion for moving ball.

VI. 결론

로봇 축구경기에서의 환경은 매우 빠르고 불규칙하게 변하므로 인식과 판단이 느리게 되면 누적되는 지연효과로 인하여 로봇을 원하는 대로 움직일 수 없게 된다. 그러나 본 논문에서 적용된 state-action과 super state-action model을 적용하면 복잡한 환경에 대한 인식과 로봇 동작 결정을 위한 추론을 합리적이고 신속하게 할 수 있다. 이러한 계층적 제어구조는 실시간 제어에 적합함을 알 수 있었으며, 또한 복잡하게 변화하는 로봇의 경로 탐색과 로봇이 목적된 동작에 대해 반사적인 반응을 가능하게 한다. 이러한 축구로봇 시스템으로 98~99년도에 개최된 여러 축구로봇대회에서 상위에 입상할 수 있었다는 것은 제안한 제어방식이 효용성을 나타내는 좋은 예이다. 추후 연구과제는 로봇팀의 협동 동작과 상대팀과의 경쟁에 있어 학습을 위한 객관적인 평가 기준을 설계하고 이를 바탕으로 변화하는 환경에 대해 적응 및 학습 능력을 갖는 제어 구조를 구현하는 것이다.

참고문헌

- [1] Tae-Yong Kuc, Seung-Min Baek, In-Jae Lee, and Kyung-Oh Sohn "Intelligent control strategy for robot soccer" *4th Int. Symp. on Artifi. & Robotics*, Oita, Japan, 1999.
- [2] Tae-Yong Kuc, Woong-Gie Han, and Seung-Min Baek "Learning control of soccer playing mobile robots" *IFAC Conf. on SSC*, July, 1998.
- [3] Lynne E. Parker "ALLIANCE: An architecture for fault tolerant multirobot cooperation" *IEEE Trans. Robotics and Automat.* vol. 14, no. 2, April, 1998.
- [4] Seung-Min Baek, Woong-Gie Han, and Tae-yong Kuc "Cell-based motion control of mobile robots for soccer game" *Proceeding of Micro-Robot World Cup Soccer Tournament* pp. 77-82, KAIST, Taejon, Korea, July, 1997.
- [5] Woong-Gie Han, Seung-Min Beak and Tae-Yong Kuc "Genetic algorithm based path planning and dynamic obstacle avoidance of mobile robots" *IEEE Int. Conf. on SMC* vol. 3, pp. 2747-2751, Orlando, Florida, USA, 1997.
- [6] Atsushi Fujimori, Peter N. Nikiforuk, and Madan M. Gupta "Adaptive navigation of mobile robots with obstacle avoidance" *IEEE Trans. Robotics and Automat.* vol. 13, no. 4, August, 1997.
- [7] Micro-Robot World Cup Soccer Tournament (MIROSOT '97), Call for participation, KAIST, Daejon, Korea, 1997.
- [8] Woong-Gie Han, Seung-Min Baek, and Tae-yong Kuc "GA based on-line path planning of mobile robots playing soccer games" *Mid West Symposium on Circuits and Systems*, Sacramento, CA, USA, August, 1997.
- [9] Woong-Gie Han, Seung-Min Baek, and Tae-Yong Kuc "Path planning of visual-served multiple mobile robots using the genetic algorithms" *Proceedings 96 Micro-Robot World Cup Soccer Tournament*, pp. 57-63, November, 1996.
- [10] David B. Reister and Francois G. Pin "Time-optimal trajectories for mobile robots with two independently driven wheels" *The International Journal of Robotics Research*, vol. 13, no. 1, February, 1994.
- [11] Fabrice R. Noreils "Toward a robot architecture integrating cooperation between mobile robots: Application to Indoor Environment" *The International Journal of Robotics Research* vol. 12, no. 2, February, 1993.
- [12] Micro-Robot World Cup Soccer Tournament (MIROSOT '96), proceedings, KAIST, Daejon, Korea, 1996
- [13] 이인재, 백승민, 손경오, 국태용 "다중이동 로봇의 계층적 제어" 한국자동제어 학술회의 논문집, 1권, pp. 252-255, 1998.
- [14] Arvin Agah and Kazuo Tanie "Robots playing to win: Evolutionary soccer strategies" *IEEE Int. Conf. Robotics & Automation*, pp. 632-637, Albuquerque, New Mexico, April, 1997.
- [15] John T. Feddema, Rush D. Robinett, and Brian J. Dressen "Designing stable finite state machine behaviors using phase plane analysis and variable structure control" *IEEE Int. Conf. Robotics & Automation*, pp. 1134-1141, Leuven, Belgium, May, 1998.
- [16] Judy A. Franklin, Tom M. Mitchell, and Sebastian Thrun "Recent advances in robot learning" Kluwer Academic Publishers, pp. 163-187, 1996.
- [17] John S. Bay "Design of the "Army-Ant" cooperative lifting robot" *IEEE Robotics & Automation Mag.*, pp. 36-43, March, 1995.
- [18] Miguel Schneider Fontan and Maja J. Matraric "Territorial multi-robot task division" *IEEE Trans. Robotics and Automat.* vol. 14, no. 5, pp. 815-822, October, 1998.
- [19] <http://icon.skku.ac.kr/> 축구로봇 실험 동영상.



이 인 제

1995년 호서대 제어계측공학과 졸업 (공학사). 1999년 성균관대학교 전자공학과 졸업 (공학석사). 1999년 8월~2000년 1월 삼성종합기술원 연구원 2000년 2월~현재 메디슨 연구원재직 중 관심분야는 지능제어, 로봇틱스, 메디컬 로봇시스템 등.



백 승 민

1996년 성균관대학교 전자공학과 졸업 (공학사) 동대학원 석사(1998). 1998년~현재 성균관대학교 전기·전자 및 컴퓨터공학부 박사과정. 관심분야는 지능제어, 로봇틱스, 디지털 제어 등.



손 경 오

1998년 성균관대학교 전자공학과 졸업 (공학사). 동대학원 석사(2000). 2000년~현재 (주)대우통신 연구원. 관심분야는 지능제어, 로봇틱스, 디지털 제어 등.

국 태 용

제어·지동화·시스템공학 논문지 제4권, 제2호, 참조.