

비전기반 축구로봇시스템을 위한 복합제어구조에서 행위계층설계 및 시험적 평가

Design and Experimental Evaluation of Action Level in a Hybrid Control Structure for Vision Based Soccer Robot

심현식, 성윤경, 김종환
(Hyun-Sik Shim, Yoon-Gyeong Sung, and Jong-Hwan Kim)

Abstract : A hybrid control structure for vision-based soccer robot system is considered. The structure is composed of four levels such as the role, action, behavior and execution levels. The control structure, which is a combination of hierarchical and behavioral structures, can efficiently meet the behavior and design specifications of a soccer robot system. Among the four levels, only the design of the action level is proposed in the paper and is experimentally evaluated. Design hypothesis and evaluation method are presented to improve the reliability and accomplishment of the robot system. Due to the essential element of soccer robot system design, a systematic design procedure of the action level is proposed. With the proposed structure and algorithms of the action level, the excellent result was shown at the MiroSot'98 held in France.

Keywords : vision based soccer robot, hybrid control structure

I. 서론

발전소, 공장, 이동 로봇 등의 거대 다변수시스템과 같은 시스템들은 불완전한 모델, 분산된 센서, 큰 외란, 다중 time scale, 다중 목적 함수, 복잡한 정보 패턴, 그리고 많은 양의 데이터 등의 특징들을 가지고 있다. 이와 같은 복잡한 시스템을 제어하기 위한 기본적인 연구로서 지능 제어연구가 수행되고 있으며 특히, 지능제어 시스템을 위한 계층적(hierarchical) 구조와 행동양식(behavioral) 구조를 이용한 설계에 대해서 많은 연구가 있어 왔다 [1][2][3][4]. Fayek et. al. [5]는 이러한 구조에 대해서 안정성, 신뢰성 등에 대한 여러가지 평가를 하였다. 각 구조들은 각각의 장단점을 가지고 있기 때문에 최근에는 각 제어구조의 장점들을 취합하여 새로운 형태의 구조를 제안하려는 연구가 진행되고 있다.

Shim et. al은 계층적 및 행동양식 구조를 혼합한 비전기반 축구로봇시스템 [6][7]을 위한 복합제어구조(그림 1)를 제시하였다. 제시된 제어구조 [8]는 역할(role), 행위(action), 행동(behavior) 및 실행(execution)계층으로 구성된다. 실행계층은 로봇의 구동부를 작동하기 위한 것이다. 행동계층은 행동양식 구조에 근거하여 설계되었으며, 기본적인 이동 행동과 동적인 환경 변화에 대응(장애물회피등)하기 위해 구성되었다. 행위계층은 로봇이 축구 경기를 수행하도록 구성된 부분이며 행위계층에는 자신만의 전략을 가지고 있기 때문에 역할 계층 없이도 경기가 수행되게 할 수 있다. 그러나, 지역방어 개념의 한계, 로봇고장 및 상대편 로봇에 의해 차단 당할 때를 대비하여 역할 계층을 별도로 두는 것이 보다 좋은 결

과를 얻었다 [9].

본 논문에서는 비전기반 축구로봇시스템의 복합제어 구조 중에서 행위계층의 설계 및 평가방법을 제시하고, 두 팀간의 실제 실험을 통해 제안된 제어구조의 설계 및 평가 기준들의 타당성을 검토한다.

II. 용어 및 기호의 정의

이 절에서는 이 논문에서 다루어지는 축구로봇시스템의 제어문제와 관련된 용어 및 수식적 기호들을 정의한다.

- 행동 (behavior) : 특정 지점으로 이동하거나, 멈추는 것등의 어떤 일을 목적으로 실제로 자기 몸을 움직이는, 개체가 행하는 하위 계층의 것을 말한다.

- 행위 (action) : 행동과는 달리 개체가 어느 특정한 목적을 가지고 행해지는 것을 통칭해 말하며 행동보다 상위의 개념이다.

- 역할 (role) : 개체가 응당 하여야 할 일 (구실)을 뜻하며, 축구로봇시스템에서는 골키퍼, 공격수등을 뜻한다.

- A : 행위(action)들의 집합으로서 $\{a_0, a_1, \dots, a_{n_a-1}\}$ 로 표시된다. 여기서, a_i 는 i 번째 행위, n_a 은 행위의 개수를 뜻하며 A_m^l 은 역할 l 번째에 속해있는 행위들의 집합이다.

- O : 역할 (role)들의 집합으로서 $\{ro_0, ro_1, \dots, ro_{n_r-1}\}$ 로 표시된다. 여기서, ro_i 은 i 번째 역할, n_r 은 역할의 개수이다.

축구로봇 경기에서 상대편 로봇과 불은 위치로 표시되며 우리편 로봇은 자세로 표시한다.

- $PO(k)$: 시간 k 에서의 위치 (position)로서 $(x, y)^T$ 로 표시되며 k 가 없으면 현재 시간을 의미한다. 단위는 cm 이다.

- $P(k)$ or p : 시간 k 에서의 자세 (posture)로서 $(x, y, \theta)^T$ 로 표시되며 이것은 PO 와 방향 θ 로도 나누

접수일자 : 1999. 5. 13., 수정완료 : 1999. 11. 11.

심현식 : LG 교통정보연구원

성윤경 : 한국원자력연구소 로봇연구실

김종환 : 한국과학기술원 전기 및 전자공학과

어 표시된다.

• q : 로봇의 속도 벡터를 뜻하며 선속도 v 와 각속도 ω 인 $(v, \omega)^T$ 로 표시된다.

• $Zone(x_1, x_2, y_1, y_2)$: $\{x_1 \leq x \leq x_2\}$ 과 $\{y_1 \leq y \leq y_2\}$ 에 속해있는 위치 $(x, y)^T$ 들의 집합이다.

• CR : 원 이동 궤적에 사용되는 원 궤적 명령으로 $(C_x, C_y, C_r, C_d)^T$ 로 표시된다. 여기서, C_x 와 C_y 는 원의 중심점, C_r 은 원의 반지름, C_d 는 로봇의 이동 방향을 뜻한다.

축구로봇시스템을 위해 Cover zone이라는 새로운 용어를 정의한다. Cover zone이라는 것은 지역방어 전술에 기반 하여 나온 개념으로 로봇이 맡게 되는 영역(area)을 나타내기 위해 정의된 용어다. Cover zone을 CV 라는 기호로 정의하고 그 의미는 다음과 같다.

• $CV(G)$: 위치 (PO) 들의 집합으로 G 의 영역을 의미한다.

• G : G 는 공 b, 로봇 rb 또는 역할 ro 가 될 수 있다.

• $CV(G|a_i)$: 행위 a_i 에 의해 cover된 G 의 위치들의 집합 (영역)이다.

III. 행위계층

행위계층에서는 주어진 역할에 따라 목적을 이루기 위해 여러 가지 할 수 있는 행위 (action)들 중 그 개체가 제일 적합한 것을 선택하여 하위 계층인 행동계층에 명령을 내려보내 로봇을 움직이게 한다. 본 계층을 구현하는 목적은 모든 개체에게 동일한 능력을 구현시켜서 각각이 모두 우수한 개체가 되도록 하고 신뢰도 및 일의 성취도를 배가시키기 위한 것이다.

1. 행위

행위계층에는 그림 2와 같이 기본 행위(basic actions)들이 설계되어 있고 행위선택기 (action selector)가 그 기본 행위들 중에서 하나의 행위를 선택하여 하위 계층인 행동계층에 전달하여 원하는 동작을 하게한다. 행위 (action)는 볼의 위치, 속도, 로봇의 자세등에 의해서 결

정되는 것으로 행위 a_i 는 실행가능함수 (feasible function) $f_i(k, V_b, PO_b, PO_o, P_{rb})$, 실행함수 (execution function) $e_i(k, V_b, PO_b, PO_o, P_{rb})$, 실행가능영역 함수 (feasible area function) $fa_i(PO_b)$ 세 부분으로 이루어져 있다. 여기서, V_b 는 볼의 속도, PO_b 와 PO_o 는 볼과 상대 로봇의 위치 (position)이며 P_{rb} 는 home robot의 자세 (posture)이다. 실행가능함수는 현재의 상황에서 이 행위가 실행 가능한지를 검사하는 함수이고 실행 가능한 경우 실제 행위를 유발시킬 수 있는 P_r 또는 CR 을 계산하는 함수가 실행함수이다.

$$f_i(k, V_b, PO_b, PO_o, P_{rb}) = \begin{cases} 1 & \text{if } a_i \text{ is feasible} \\ 0 & \text{else} \end{cases}$$

$$e_i(k, V_b, PO_b, PO_o, P_{rb}) = \begin{cases} P_r & \text{if } a_i \text{ is for move behavior} \\ CR & \text{if } a_i \text{ is for circular_motion behavior} \end{cases}$$

실행가능영역함수는 각 행위가 공의 위치에 따라 실행 가능한 영역을 정의하는 함수로 공이 정의된 영역에 있을 때에만 실행이 가능하다. 실행가능함수와와의 차이점은 실행가능함수가 공, 로봇등의 여러 가지 변수에 의해 복잡하게 결정되는 함수인 반면, 실행 가능영역함수는 단순히 각 행위의 제한 조건에 해당된다. 실행가능영역함수 fa_i 는 공이 행위 a_i 의 고유 할당영역에 포함되는지를 나타내는 함수이며 다음처럼 정의된다.

$$fa_i(PO_b) = \begin{cases} 1 & \text{if } PO_b \in CV_i(b) \\ 0 & \text{else} \end{cases}$$

여기서 $CV_i(b)$ 는 행위 a_i 에서 실행 가능한 공의 영역을 의미하며 예를 들어 $CV(G) = Zone(x_1, x_2, y_1, y_2)$ 은 G 의 영역이 $\{x_1 \leq x \leq x_2\}$ 과 $\{y_1 \leq y \leq y_2\}$ 에 속해있는 위치 $(x, y)^T$ 들의 집합이라는 것을 뜻한다.

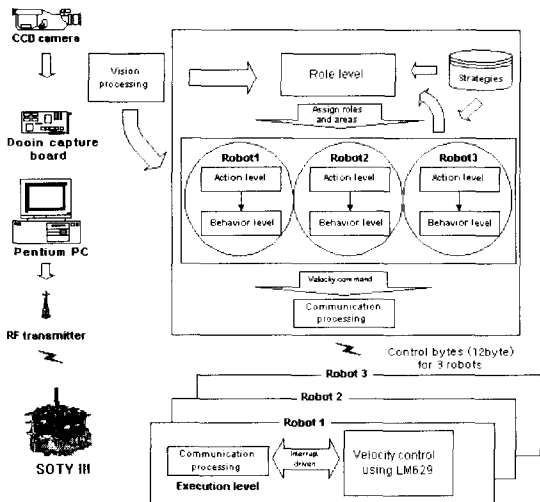


그림 1. SOTY III의 제어구조.
Fig. 1. Control structure of SOTY III.

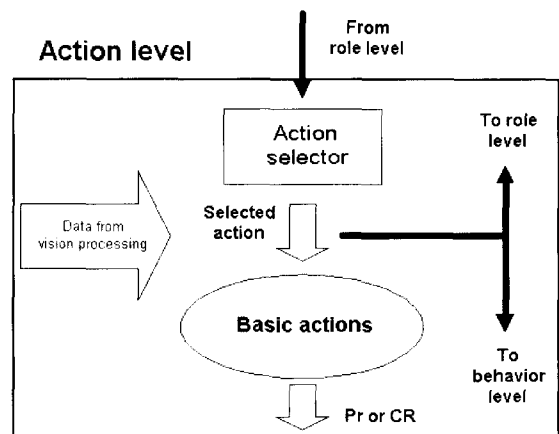


그림 2. SOTY III 의 행위계층.
Fig. 2. Action level of SOTY III.

2. 기본 가정과 설계 및 평가 기준

행위계층의 설계는 축구로봇시스템에서 핵심적인 부분으로 이 계층의 성능이 전체 시스템의 성능을 좌우한다. 그러므로, 행위계층에서는 제한된 상황에서 기대하는 목적을 달성 할 수 있게 몇 가지 가정과 설계 및 평가척도를 마련한다. 가정에 위배되는 상황은 상위 계층인 역할 계층에서 더 보완적으로 다룰 수 있게 구성한다. 행위계층의 설계를 위한 가정은 다음과 같다.

- ① 모든 개체는 동작 가능하다.
- ② 각 개체는 국소적인 관점(local viewpoint)을 갖는다.
- ③ 각 개체들은 수행해야 할 특정 역할이 주어진다.
- ④ 시스템을 위한 행위의 수는 유한개로 가능하다.

1번 가정은 행위계층의 기본 가정으로 모든 개체가 동작하고 있는 경우를 말한다. 로봇이 고장나거나 다른 이유로 인해 행위를 수행할 수 없는 경우는 이 계층에서 다루지 않고 상위 계층에서 다룬다.

2번 가정은 축구로봇시스템에 대한 정확한 예측 환경 모델이 없으므로, 현재까지의 주어진 상황을 고려하여 비교적 가까운 미래에 대해 적절한 행위를 선택하는 국소적인 관점을 갖는다.

3번 가정인 각 개체에 역할을 할당하는 장점은 역할계층의 설계시에 자세히 설명된다 [9]. 4번 가정은 로봇 축구를 위해 필요한 행위의 개수가 유한개로 가능하다는 것으로 행위의 정의에 따라 개수는 같지 않을 것이다.

축구로봇시스템 설계시 가장 중요한 것은 시스템의 하드웨어 및 소프트웨어 알고리즘의 신뢰도 및 성취도인데, 이들을 향상시키고 평가하기 위해 설계 및 평가 항목을 다음과 같이 정의한다.

- ① vision의 정보는 정확한가?
- ② 경기장 내의 모든 지역에서 정의된 행위들에 의해 공이 cover되는가?
- ③ 공이 어디에 있더라도 모든 행위들이 발생되는가?
- ④ 각 행위가 정의된 동작을 수행하는가?
- ⑤ 할당된 역할에 의해 맡아지는 영역이 중복되는가?
- ⑥ 우리편끼리의 협력 작업은 잘 수행되는가?
- ⑦ 공의 점유율은 높았는가?
- ⑧ 공은 어느 편 쪽에 많이 있었는가?
- ⑨ 경기 결과는?

축구로봇시스템에서 시각 정보는 매우 중요하고 각 행위에서의 실행가능함수와 실행함수들이 모두 비전정보에 전적으로 의존하기 때문에 다음의 비전정보에 관한 실행 조건함수를 만들어 행위선택기에서 사용하게 한다.

$$fb(k) = \begin{cases} 1 & \text{if a ball is found} \\ 0 & \text{else} \end{cases} \quad (1)$$

$$frb_i(k) = \begin{cases} 1 & \text{if robot } i \text{ is found} \\ 0 & \text{else} \end{cases} \quad (2)$$

위의 실행가능함수의 의미는 비전시스템에서 공과 로봇을 찾은 경우에만 각 행위가 실행하게 만든다.

축구로봇은 두 팀의 로봇이 서로 상대방의 골에 공을 넣

는 경기이므로 공의 위치에 따라서 로봇이 슛을 하거나 가로막거나 하는 등의 행위를 하여야 한다. 그러므로 각 행위는 공의 위치에 전적으로 의존하여 설계되며 공이 경기장 어느 위치에 있더라도 로봇이 적절한 행위를 해야 한다. 그래서 각 행위의 설계시 다음 조건을 만족시켜야 한다.

Stop 행위를 제외한 설계된 모든 행위들에 의해 경기장 전 영역에 대해 공이 cover되어야 한다(2번 조건). 즉, 각 행위는 공집합이 아닌 $CV(b|a_i)$ 를 가지며 이 행위들의 영역의 합이 전체 영역이 되어야 한다는 것이다. 수식으로 표현하면 다음과 같다.

$$\sum_{i \neq 0}^{n_a-1} CV(b|a_i) = ALL \quad (3)$$

(3)에서 Stop 행위 a_0 가 포함되지 않는 것은 공의 위치와는 관계없이 비전시스템이 공이나 로봇을 찾지 못하는 경우에 사용된다. 그러므로, 효과적으로 행위선택기를 설계하면 a_0 행위는 선택되지 않는다.

2번 항목은 행위 설계시 중요한 평가 기준이다. $CV(b|a_i)$ 는 $CV_i(b)$ 와는 다른 개념이다. $CV_i(b)$ 는 행위의 설계시 설계자에 의해 주어진 영역을 의미한 반면 $CV(b|a_i)$ 는 행위 a_i 에 의해 커버될 수 있는 공의 영역을 의미한다. $CV(b|a_i)$ 는 행위를 설계한 후 각 실행가능함수 $e_i(k)$ 에 의해 결정되어지는 영역이다.

3번 항목은 마찬가지로 행위 설계를 위한 기준인데, 2번과는 상반된 개념을 의미하며 각 행위의 중복선택에 관한 것이다. 설계된 행위들은 $CV_i(b)$ 에 의해서는 현재 공이 실행 가능 영역에 있어도 각 행위의 실행가능함수들은 공의 위치, 공의 속도, 로봇의 자세등의 여러 변수들의 복잡한 함수이기 때문에 다른 변수들에 의해 실행 조건이 가능하지 않은 경우가 있다. 그러므로 공이 어느 영역에 있어도 로봇의 상태에 관계없이 반드시 로봇이 어떠한 행동을 하도록 하는 행위를 발생시키도록 설계하는 것이 중요하다. 이러한 조건을 만족시키기 위해 모든 변수들의 정의 구역에 대해서 로봇이 어떠한 행동을 하도록 설계하는 것은 매우 어려울 것이다. 이것을 해결하기 위해 사용한 방법이 행위의 중복 설계이다. 각 행위 하나 하나는 모든 변수들의 정의 구역에 대해서 정해진 행동을 유발시킬 수 없으나 여러 행위들이 모였을 때 이것을 만족시키도록 한다.

각 행위들의 cover zone $CV(b|a_i)$ 의 교집합들을 구했을 때 가장 많은 중첩이 이루어지는 영역이 위의 설계요소를 제일 만족시킬 것이고 이것은 시스템의 경기 수행에 대한 신뢰성을 향상 시킬 것이다. 경기장 전체의 영역에서 중첩되는 행위의 수가 많으면 많아질수록 시스템의 신뢰성을 높일 것이다. 이것을 다음처럼 수식으로 나타낼 수 있다.

$$UPO_i = ALL \text{ for } PO_i = CV(b|a_i) \text{ if } Na(PO_i) \geq n_{min} \quad (4)$$

여기서 $Na(PO_i)$ 는 $CV(b|a_i) = PO_i$ 가 되는 행위의 개수를 나타내는 함수이며 n_{min} 은 최소 행위의 수로 양수이다. 위의 수식의 의미는 중첩되는 행위의 개수가 최소 값보

다 큰 영역의 집합이 전체 경기장이 된다. 위의 기준에 의해 선택될 수 있는 행위의 수가 1개이상일때 어느 행위를 선택할 것인가는 행위선택기의 설계 문제이다. 이것은 행위선택기의 설계시에 자세히 다루겠다.

4번 항목은 행위 설계 후 선택되어진 행위에 의해 로봇이 제대로 수행하고 있는지를 평가하는 것이다.

5번 항목은 역할의 중첩을 평가하는 항목으로 행위가 중첩되는 영역이 클수록 신뢰성이 커지는 3번 항목과는 반대로 적을 수록 경기 운영에 대한 신뢰성이 커진다는 것이다. 3번 항목은 한 개체의 신뢰성을 높이기 위한 항목이고 5번 항목은 다개체 시스템의 신뢰성을 높이기 위한 항목이다. 역할을 설정하고 그 역할에 대해 영역을 설정하는 이유는 지역 방어 전술에서 나온 개념으로 각 개체간의 의견 충돌을 줄여서 원하는 협동 작업을 원활하게 수행하게 만든다.

6번 항목은 축구 경기에서의 협동 작업이 제대로 이루어졌는지를 평가하는 것으로 우리편 로봇과의 충돌여부, 한 로봇이 슈트를 하면 다른 로봇은 어떤 행위를 하는가 등을 점검한다. 전체 시스템 설계 후 경기 결과를 평가해야 할 항목으로 다음에 정의되는 두 가지 요소에 대한 것을 평가한 후 보다 향상된 경기운영을 위해 각 계층들을 변경하게 된다.

- 우리편 로봇과의 충돌 횟수
- 각 개체의 선택된 행위의 중복 횟수

제시된 것 외에 더 많은 것들이 6번 항목을 위해 추가 될 수 있다.

7, 8, 9번 항목들은 실제 축구 경기와 같은 평가 기준으로 경기 후에 평가된다. 축구 경기에서 가장 중요한 평가 기준은 결국 경기를 이겼는가인 것이다. 그리고 이것을 보조하기 위한 다른 평가 항목으로 공이 상대편 진영에 많이 있었는가와 공의 점유율일 것이다. 7번 항목은 공과 우리 편 로봇과의 거리 $dist_1$ 을 이용하여 평가식을 다음처럼 제시한다.

$$Possession_{time}(region\ occupation) = \frac{Time\ to\ satisfy\ Condition_1}{Entire\ game\ time} \quad (5)$$

여기서 $Condition_1$ 은 $dist_1 < 7$ 이다. 공을 점유하고 있는 것은 대부분은 공격시에는 벽 근처에서 공을 천천히 드리블하는 경우와 상대편 로봇과 공을 사이에 두고 대치하고 있는 경우, 또는 수비시 공을 가로막고 있는 경우이다. 슈트를 하는 경우에는 공이 순간적으로 로봇을 맞고 튕겨나가기 때문에 점유율에 영향을 끼치지 않는다. 점유율은 대부분 상대편과의 대치상태가 제일 많이 차지하며 낮을수록 공격을 많이 한 것으로 생각할 수 있다.

8번 항목은 상대편 진영에 많이 있었을 수록 그만큼 유리한 경기를 했다는 척도가 되므로 다음과 같이 평가식을 제시한다.

$$Stay_{time}(region\ occupation) = \frac{Time\ of\ ball\ staying\ in\ opponent\ region}{Entire\ game\ time} \quad (6)$$

여기서 상대편 영역에 공이 머물러 있었던 시간을 연속적으로 어떤 시간 t_c 을 넘는 시간만으로 제한함으로써 중간지점에서 짧은 시간 동안에 공이 왔다갔다하는 시간은 제외된다.

9번 항목은 경기를 이긴 것도 중요하지만 경기 스코어가 어떻게 되었는가를 평가하는 것이다. 상대편에 의해서 골을 많이 먹은 경우에는 우리편의 수비가 문제가 있다는 것을 평가 할 수 있는 척도가 되기 때문이다. 또한, 자살골로 얻은 점수는 공격해서 얻은 점수보다 낮게 평가되며 수비적인 팀에서 자살골을 넣으면 그것은 상대팀에 의한 점수보다 더 나쁘게 평가해야 한다. 이와 같은 것을 고려하여 다음의 평가식을 제시한다.

$$marks = \omega_a(\omega_s N_h + N_{hs}) - \omega_d(N_o + \omega_s N_{os}) + \omega_{win}(N_h + N_{hs} - N_o - N_{os}) \quad (7)$$

여기서 N_h 는 우리편이 넣은 점수, N_{hs} 는 자살골로 얻은 점수, N_o 는 상대편이 넣은 점수이며 N_{os} 은 자살골로 잃은 점수이다. 그리고 ω_a 는 공격에 대한 가중치, ω_d 는 수비에 대한 가중치, $\omega_s \geq 1$ 로 자살골에 대한 가중치이고 ω_{win} 은 승리한 경우에 진 경우보다 더 큰 평가를 주기 위한 가중치이다. 위의 식중 마지막 항은 일종의 바이어스 항으로 경기상 진 경우는 이긴 경우보다 좋게 평가될 수 없기 때문에 필요한 항이다. 점수차가 크면 클수록 더 많은 평가를 주게 된다. 시스템 설계시 공격적인 팀과 수비적인 팀의 전략을 구별하여 평가하기 위한 가중치로 공격적일 때는 $\omega_a > \omega_d$ 로 설정하면 된다. 만약 경기 결과에 대해서만 평가한다면 $\omega_a = \omega_d$ 로 설정하면 된다.

6, 7, 8, 9번 항목들은 어떤 객관적인 기준이 현재로서는 정의되어 있지 않다. 각 항목들의 가중치를 설정하고 평가치를 보면서 적절한지 아닌지를 판단하는 것은 설계자의 주관적인 관점에 달려있다. 이 항목들은 각 항목에 가중치를 두어 학습등의 평가함수로 사용될 수 있다.

제시된 제어구조의 형태가 신뢰성과 강인성의 향상을 위한 것이지만 단순히 제안된 제어구조를 구성하였다고 신뢰성과 강인성이 향상된 것은 아니다. 계층적인 형태로 구성된 제어구조에서는 하위 계층의 신뢰성과 안정성이 보장되어 있다는 가정하에서 설계되고 구현되고 있기 때문에 모든 계층의 신뢰성이 보장되도록 설계하는 것이 중요하다. 행동계층에서는 실행계층의 속도 제어가 안정하고 신뢰할 만하다는 가정하에서 설계를 하듯이 행위계층도 행동계층이 안정하고 신뢰성이 있다는 가정하에서 설계를 하고 있다. 행위계층도 마찬가지로 상위 계층의 설계를 위해서는 안정성과 신뢰성을 만족시키도록 설계가 되어 있어야 하고 4가지 기본 과정과 9가지 항목이 이를 위한 것이다.

3. 기본 행위

SOTY III에 사용된 기본 행위들은 18개로써 다음과 같이 분류된다.

- primitive actions : Stop(a_0), Wandering(a_1), Sweep_Ball(a_2), Ball_Find(a_3)
- offensive actions : Standard_Shoot(a_4), Cannon_Shoot(a_5), Position_To_Shoot(a_6), Turning_Shoot(a_7), Spin_Shoot(a_8), Corner_Kick(a_{17})
- defensive actions : Push_Ball(a_9), Position_To_PushBall(a_{10}), Screen_Out_Ball(a_{11})
- keeper actions : Block(a_{12}), K_Defense_Goal(a_{13}), K_Default_Position(a_{14}), K_Need_Escape_Goal(a_{15}), Keeper_Attack(a_{16}).

각 행위들은 앞에서 정의된 대로 실행가능함수와 실행함수, 실행가능영역함수로 설계된다. 각 행위들은 앞에서 정의한 평가 기준중 2, 3, 4번 기준들을 만족할 수 있도록 설계된다. 여기서는 예제로써 standard_shoot(a_4) 행위와 Block(a_{12}) 행위에 대해서만 설명한다(참조 [7][10]).

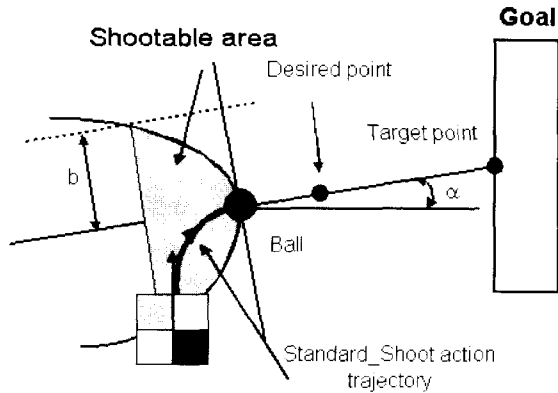


그림 3. Standard_shoot 행위의 쏠 가능 지역.
Fig. 3. Shootable region of standard_shoot action.

Standard_shoot 행위는 축구경기 중에서 가장 중요한 행위이며 골을 넣을 수 있는 행위이다. 골을 넣을 수 있는 행위는 모두 standard_shoot, spin_shoot, turning_shoot, cannon_shoot 4종류가 있으며, 공격수인 경우에 이 행위들은 다른 어떤 행위들보다도 우선 순위가 높다. 그림 3에 나온 것처럼 standard_shoot 행위는 언제나 가능한 것이 아니라 쏠 가능 지역에 로봇이 있는 경우에 만 선택된다. 이 쏠 가능한 지역은 공이 놓여 있는 위치에 따른 함수로 영역이 공의 위치에 따라 변하게 된다. 쏠 가능 지역을 판단하는 알고리즘은 다음과 같다.

$$f_4 = \begin{cases} 1 & \text{if } y_{rb} < (y_b - 2) \text{ and } |tr_x| < b \\ & \text{and } tr_y > \left(\frac{tr_x^2 a}{b^2} + tb_y \right) \\ 0 & \text{else} \end{cases}$$

$$tr_x = (x_{rb} - x_g) \cos \alpha + (y_{rb} - y_g) \sin \alpha$$

$$tr_y = -(x_{rb} - x_g) \sin \alpha + (y_{rb} - y_g) \cos \alpha$$

$$tb_x = (x_b - x_g) \cos \alpha + (y_b - y_g) \sin \alpha$$

$$tb_y = -(x_b - x_g) \sin \alpha + (y_b - y_g) \cos \alpha$$

여기서 (x_g, y_g) 는 상대편 골의 위치이며 a 와 b 는 양의 상수들로 이 값에 변화에 따라 쏠가능 지역의 영역이 바뀌게 된다. 그리고 그림 3에서 α 는 볼과 상대편 골 사이 벡터의 각도를 의미한다. 위의 함수는 공과 로봇을 그림 3처럼 공과 골대 사이의 각도 α 로 좌표 변환시켜서 로봇이 빗금친 영역에 있는 지를 표현한 것이다. a 는 포물선의 기울기이며 b 는 변환된 좌표축에서 x 축의 값을 제한하는 값이다. 이 실행가능함수를 만족하게 되면 그림 3의 desired point를 계산하기 위해 다음과 같은 실행함수를 사용한다.

$$e_4 = \begin{cases} x_r \\ y_r \\ \theta_r \end{cases} = \begin{cases} x_b + l \cos \alpha \\ y_b + l \sin \alpha \\ \alpha \end{cases}$$

l 은 양의 상수로 임의로 정해지는 값이다. 주로 10cm : 20cm사이 값을 선택한다. 위의 주어진 행위의 실행가능영역은 다음과 같다.

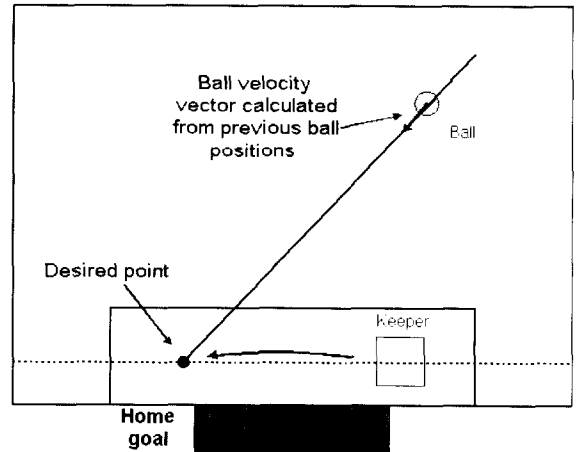


그림 4. Block 행위.
Fig. 4. Block action.

$$CV_4(b) = \text{Zone}(15, 115, 0, 90) \cup \text{Zone}(115, 130, 25, 75)$$

Block 행위는 골키퍼를 위한 행위로 공이 골대를 향해 움직일 때 공의 궤적에 따라 예측된 지점으로 이동하여 공을 막는 행동을 한다. 공의 위치 예측은 공의 위치정보로부터 추출된 공의 속도 벡터를 이용하여 계산된다. 이 Block 행위의 성능은 로봇의 응답속도와 비전 정보의 처리속도에 크게 좌우된다. 이 행위의 실행가능함수는 다음과 같다.

$$f_{12} = \begin{cases} 1 & \text{if } speed > 25 \text{ and } v_x < 0 \text{ and } 110 < \theta_v < 250 \\ 0 & \text{else} \end{cases}$$

여기서 볼의 속도 성분은 $(v_{bx}, v_{by})^T$, $speed = \sqrt{v_{bx}^2 + v_{by}^2}$ 이

며 θ_r 는 속도 성분의 방향을 나타내며 $\tan\left(\frac{v_{by}}{v_r}\right)$ 로 계산된다. 선택 조건은 볼이 빠르게 움직이고 방향이 우리 진영을 향하고 있을 때를 의미한다. 실행함수는 다음과 같다.

$$e_{12} = \begin{Bmatrix} x_r \\ y_r \\ \theta_r \end{Bmatrix} = \begin{Bmatrix} 4.5 \\ y_b + (4.5 - x_b) \frac{v_y}{v_r} \\ 90^\circ \end{Bmatrix}$$

여기서 $25 < y_r < 65$ 으로 제한된다. 이 행위의 실행가능 영역은 다음과 같다.

$$CV_{12}(b) = Zone(30, 130, 0, 90)$$

4. 행위선택기

그림 2처럼 행위계층에는 행위선택기(action selector)가 있어 각 개체가 수행할 수 있는 여러 가지 행위(action)중 주어진 상황에 따라 적절한 것을 선택하도록 한다. 행위선택기는 외부로부터 들어온 정보 및 기본 전술과 상위 계층으로부터의 정보를 이용하도록 구성한다.

이 계층에서 사용되는 전략 또는 전술은 상위 계층의 것과 유사한 목적으로 사용되지만 구성되는 방법이나 규칙등은 같지 않다. 역할 계층에서의 전술 및 전략은 경기 운영을 위한 전략이 구현되게 되며 여러 가지 전략들이 데이터베이스 형식으로 구성되어 상대팀의 전략에 따라 대응 할 수 있게 구성한다. 행위계층에서의 전술은 기본 전략이나 상위 레벨이 없는 경우에 대비하여 기본적인 협동 전술등을 구사할 수 있게 구성한다. 또한 역할에 따른 기본적인 위치, 금지될 행동같은 상세 전술 등이 구현된다. 행위선택기의 출력은 설계된 기본 행위들중 하나의 행위만 선택하여 그 선택된 행위의 실행함수의 결과를 행동계층에 전달한다.

모든 로봇에 동일하게 구현되는 제어구조이지만 각 개체의 행동을 다르게 만드는 것은 상위 계층에서의 주어진 역할과 영역에 따른다. 행위선택기에서 개체의 일을 선택할 때 주로 행할 행위의 선택 비중을 주어진 역할에 따라 더 높게 만듦으로써 각 개체가 다른 일을 선택하게 할 수 있다. 행위선택기에서 역할이 l 인 로봇 j 의 행위 a_j 의 최종적인 실행가능함수는 다음과 같다.

$$F_{ijl}(k) = f_i(k) \cap fa_i \cap fb(k) \cap frb_j(k) \cap fro_{jl} \cap fact_{il} \quad (8)$$

여기서, fro_{jl} 는 역할 계층에서 로봇 j 에 할당된 역할 l 의 영역에 공이 있는가를 나타내는 함수이다. 그리고 $fact_{il}$ 은 역할 l 에서 사용되는 행위들의 집합 A'_{ro} 에 행위 a_i 가 소속되는지를 나타내는 함수이다. fro_{jl} 의 영역은 fa_i 의 영역과는 다른 의미로 fa_i 의 영역은 행위 설계시 고려되어지는 영역이고 fro_{jl} 은 상위 레벨인 역할 계층에서 필요에 따라 제한하는 영역으로 fa_i 의 영역을 제한하는 것으로 볼 수 있다. fro_{jl} 의 영역은 상황에 따라 가변적이지만 fa_i 의 영역은 행위 설계가 끝나면 바뀌지 않고 고정되어있다.

$$fro_{jl} = \begin{cases} 1 & \text{if } PO_b^j \in CV(ro_l) \\ 0 & \text{else} \end{cases} \quad (9)$$

여기서 $CV(ro_l)$ 은 역할 계층에서 역할 l 에 할당된 영역이다.

$$fact_{il} = \begin{cases} 1 & \text{if } a_i \in A'_{ro} \\ 0 & \text{else} \end{cases} \quad (10)$$

(8)은 Ball_Find 행위와 Wandering 행위를 제외한 모든 행위에 적용된다. Ball_Find 행위와 Wandering 행위에 대한 최종 실행가능함수는 다음과 같다.

$$F_{1jl}(k) = f_1(k) \cap fa_1 \cap fb(k) \cap frb_j(k) \cap fro_{jl} \cap fact_{1l}$$

$$F_{3jl}(k) = f_3(k) \cap fa_3 \cap fb(k) \cap frb_j(k) \cap fro_{jl} \cap fact_{3l}$$

SOTY III에서는 Striker, Sweeper, Keeper, Right_Wing, Left_Wing이 사용되었다. 각각의 역할에 따라서 18개의 기본 행위중 사용되는 행위의 수는 각각 다르다. 각 역할에 따라 사용된 행위의 집합은 다음과 같다.

- Striker : $A'_{ro} = \{a_0, a_2, a_3, a_4, a_5, a_6, a_7, a_8\}$
- Sweeper : $A'_{ro} = \{a_0, a_9, a_{10}, a_{11}\}$
- Keeper : $A'_{ro} = \{a_0, a_3, a_4, a_{12}, a_{13}, a_{14}, a_{15}, a_{16}\}$
- Right_Wing :

$$A'_{ro} = \{a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}\}$$

- Left_Wing :

$$A'_{ro} = \{a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}\}$$

평가 기준중 3번 항목은 각 행위들이 신뢰성을 높이기 위해 중복적으로 선택이 가능하도록 만들어져야 한다는 것이다. 이런 경우에 여러 선택 가능한 것중 어느 행위를 선택할 것인가가 문제가 된다. 행위선택기는 이런 선택문제를 해결하는 일종의 중재기 (coordinator)이다. 여러 가지 해결 방법중 쉽게 생각할 수 있는 방법이 행위에 우선 순위를 배정하는 것이다. 그래서 우선 순위가 가장 높은 순으로 행위를 선택하도록 구성하는 것이다. 그러면, '어떤 순서로 순위를 매기는 것이 효율적인 경기를 운영할 수 있을까?' 하는 문제가 생긴다. 역할을 사용하는 이유중의 하나가 행위선택기의 설계시 각 행위들의 우선 순위 배정을 합리적으로 할 수 있다는 것이다. 예를 들면, 공격수는 골을 넣는 것이 맡은 역할이므로 슛에 관련된 행위가 우선적으로 선택되어야 할 것이다. 반대로 골키퍼는 공을 막아야 하므로 Block 행위 같은 것이 제일 우선 순위가 높아야 할 것이다. 이러한 역할에 따른 우선 순위의 집합을 다시 정리해보면 다음과 같다.

- Striker : $A'_{ro} = \{(a_4, a_6, a_7, a_8), a_5, (a_2, a_3), a_0\}$
- Sweeper : $A'_{ro} = \{a_9, a_{10}, a_{11}, a_0\}$
- Keeper : $A'_{ro} = \{a_{12}, (a_{13}, a_{14}, a_3), a_{16}, (a_4, a_{15}), a_0\}$
- Right_Wing :

$$A'_{ro} = \{(a_4, a_6, a_7, a_8), a_5, a_9, a_{10}, a_{11}, (a_2, a_1, a_3), a_0\}$$

•Left_Wing :

$$A_{ro}^3 = \{(a_4, a_6, a_7, a_8), a_5, a_9, a_{10}, a_{11}, (a_2, a_1, a_3), a_0\}$$

여기서 각 집합은 우선 순위가 높은 것에서 낮은 것으로 나타내어졌으며 소괄호 내에 있는 것은 우선 순위가 같게 취급될 수 있다는 것을 뜻한다.

같은 우선 순위의 행위에 순위를 매기는 것은 사용자가 전술에 따라서 바꿀 수 있다. 예를 들면 4개의 슛 행위는 어느 것을 높게 해야 할지 명백하게 알 수 없다. 선택에 따라서 로봇의 이동 패턴이나 슛 정확도 정도가 달라지기 때문이다. 그래서 사용자가 경기 진행중에 순위를 바꿈으로서 다른 전술을 구현한 것처럼 유도할 수 있다. 우선 순위를 정하는 방법은 역할에 따른 것 외에 전략에 따라서도 달라질 수 있다. 예를 들면 Right_Wing, Left_Wing, Keeper의 역할을 사용하는 경우에 위의 우선 순위 배정은 공격적인 전략이다. 즉 기회가 되면 골을 넣는데에 주력한다는 것이다. 그러나 수비를 중요시하는 전략을 사용하는 경우에는 위의 우선 순위를 다음처럼 바꾸어서 사용한다.

•Right_Wing :

$$A_{ro}^3 = \{a_{11}, a_9, a_{10}, a_2, (a_4, a_6, a_7, a_8), a_5, (a_1, a_3), a_0\}$$

이처럼 행위선택기에서 행위의 우선 순위를 달리 배정함으로써 여러가지 전술을 쉽게 구현할 수 있다. 위의 최종 실행가능수와 우선 순위에 따라 행위를 결정하는 역할 l 인 로봇 j 의 행위선택기의 알고리즘은 다음과 같다.

```
if(행위 ai의 최종 실행가능함수) action=i;
else if(행위 ak의 최종 실행가능함수) action=k;
```

```
else action=0;
```

```
switch(action)
```

```
case 0 : 행위 a0의 실행함수;
```

```
case 1 : 행위 a1의 실행함수;
```

```
case 17 : 행위 a17의 실행함수;
```

위의 if-then 순위는 우선 순위 배정에 따른 것이다. 위의 알고리즘은 역할 5가지에 따라 각각 구현되며 역할 계층에서 역할이 주어졌을 때 5가지중 3가지 역할을 위해 행위선택기의 결과에 의해 로봇 3대에 대한 기준 자세 또는 원 궤적명령을 생성해 낸다.

IV. 실험 결과 및 평가

본 논문에서 제안된 축구로봇시스템을 위한 제어구조중 행위 계층을 평가하기 위해서 여러 실험을 수행하였다. 실제 경기처럼 두팀을 이용한 경기를 통해서 앞서 정의된 평가 기준에 맞추어 통계를 낸 후 각각에 대해서 평가를 내린다.

1. 실험 절차

두 팀 다 SOTY III 로봇이 사용되었으며 동일한 비전보드와 동일한 pentium 166MHz의 PC를 호스트 컴퓨터로 사용하였다. 두 시스템 모두 vision-based soccer robot system이다. 먼저 상대편 (opponent team)의 기

본 사양을 정리하면 다음과 같다.

- 시스템 샘플링 : 33msec

- 로봇 평균 최고 속도 : 60 cm/sec (Cannon_Shoot시 는 125cm/sec)

- 사용된역할 : $RO = \{Keeper, Right_Wing, Left_Wing\}$

- 사용된 행위 :

- Keeper : $A_{ro}^2 = \{a_{12}, a_{13}, a_{14}, a_3, a_{16}, a_4, a_{15}, a_0\}$

- Right_Wing :

$$A_{ro}^3 = \{a_4, a_6, a_7, a_8, a_5, a_9, a_{10}, a_{11}, a_2, a_1, a_3, a_0\}$$

- Left_Wing :

$$A_{ro}^4 = \{a_4, a_6, a_7, a_8, a_5, a_9, a_{10}, a_{11}, a_2, a_1, a_3, a_0\}$$

- 사용된 할당 영역

- CV(Keeper) = ALL

- CV(Right_Wing) = Zone(15, 130, 40, 90)

- CV(Left_Wing) = Zone(15, 130, 0, 60)

- 사용된 전략 : 공격과 수비 둘 다 같은 비중

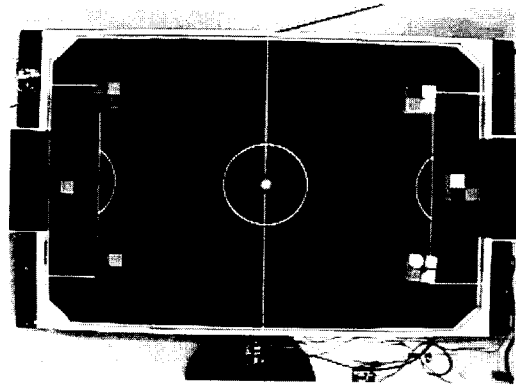


그림 5. 실험에서 각 축구로봇의 초기 위치.

Fig. 5. Initial location of each soccer robot for experiment.

실험은 3종류가 시행되었고 각 실험은 10번씩 수행되었으며 각각은 약 2분 30초씩 경기가 진행되었다. 골인되거나 대치상태가 발생하여 freeball상태가 되면 그림 5처럼 두 팀을 동일한 초기 위치에 놓아 경기를 시작하게 만들었다. 실험 종류는 다음처럼 분류된다.

- game 1 : 한 팀만 있는 경우(상대편 기본 사양 사용)

- game 2 : 우리 팀의 좌우 윙의 행위중 Sweep_Ball (a_2), Push_Ball (a_9), Screen_Out_Ball (a_{11})행위가 빠진 경우

- game 3 : 우리 팀의 행위중 Cannon_Shoot (a_6), Turning_Shoot (a_7), Spin_Shoot (a_8)이 빠진 경우

경기를 하면서 컴퓨터에 필요한 정보를 저장하고 경기가 끝난 후 저장된 정보를 이용하여 다음과 같은 평가항목들을 계산하여 결과를 정리하였다.

- marks : (8)을 사용하였으며 $\omega_{in} = 1$, $\omega_{out} = 2$, $\omega_{miss} = 5$ 로 하여 평가하였다.

- 점유율 : (5)을 사용하였다.

- 진점점유율 : (6)을 사용하였으며 공의 x 좌표가 75cm를 넘었을 때만 계산하였고 $t_c = 0.5sec$ 를 이용하였다.

- 충돌율 : 우리편 로봇끼리 셋중의 어느 둘이라도 거

리가 9cm이내일 때 충돌한다고 보고 계산하여 전체 시간에 대해서 백분율로 표시하였다.

●공격율 : 우리편 로봇 중의 어느 하나라도 슛 행위를 하면 계산하여 전체 시간에 대해 백분율로 표시하였다.

현재 사용된 역할에 의하면 공격과 수비 둘 다 중요하게 여기고 설계된 것이므로 marks의 평가시 함께 비중을 두었으며 자살골에 대해서는 2배의 가중치, 그리고 이긴 것을 강조하기 위해서 5배의 가중치를 두었다. 점유율, 진영점유율, 충돌율 및 공격률등은 전체 경기 시간에 대해서 백분율로 표시하였다.

●game 1은 기본 사양을 가진 팀의 능력을 검사하는 것으로 앞서 설명한 제어구조 및 전술이 모두 구현되었다. game 2와 3는 action level의 효용성을 알아보기 위한 것으로 game 2는 action level의 2번 조건을 만족시키지 못한 경우에 대해서 game 3는 3번 조건을 만족시키지 못한 경우에 대해서 각 조건의 중요성을 평가하기 위한 것이다.

2. 실험 결과

game 1의 실험은 한 팀만 있는 경우로 현재 상대방으로 사용되는 기본 사양의 공격기능에 대한 평가를 위한 것이다. 설계된 행위들의 성취도는 어느 정도인지, 할당 영역은 적당인지, 또는 장애물 회피가 잘 되는지 등을 평가한다.

표 1. game 1 : 상대팀이 없는 경우

Table 1. game 1 : Case without opponent team.

횟수	득점 (실점)	marks	점유율	진영 점유율	충돌율	공격율
1	7(0)	49	39.1	86.4	2.0	42.6
2	8(0)	56	31.7	78.6	2.8	36.1
3	6(0)	42	28.7	68.2	2.3	31.8
4	10(0)	70	11.4	72.1	2.3	38.2
5	6(0)	42	25.2	60.1	1.7	31.5
6	10(0)	70	28.8	79.9	2.1	36.1
7	12(0)	84	10.9	71.3	0.3	40.3
8	6(0)	42	32.3	81.2	2.9	31.0
9	7(0)	49	28.5	69.9	1.9	37.6
10	9(0)	63	26.1	72.5	2.0	41.0
평균	8.1(0)	56.7	26.3	74.0	2.03	36.6

평균 2분 30초 경기에 득점 8.1점의 공격 득점력을 가지고 있고 충돌율이 낮은 편이다. 겹쳐진 영역이 현재 10cm인데 장애물 회피를 사용하여 평균 2.03% 정도의 낮은 충돌율을 보이고 있다.

골을 많이 넣고 슛을 하는 공격율이 높은 관계로 점유율은 낮다. 특히 점수가 많이 난 7번 경기 같은 경우에 점유율은 특히 낮다. 점유율이 낮은 이유는 상대팀이 없기 때문에 대치상태가 없기 때문이다. 공은 주로 상대편 진영에만 머물러 있기 때문에 진영점유율은 매우 높은 편이다.

game 2의 실험은 같은 속도와 제어능력을 가지고 있을 때 action level의 행위 설계 평가를 한다. 공은 진영역에 걸쳐 모든 행위에 의해 cover되어야 한다는 2번 조건은 (3)을 만족하지 못하는 경우에 대한 실험이 행해졌다. Sweep_Ball, Push_Ball 및 Screen_Out_Ball 행위를 사용하지 않음으로써 벽 주위에서 공이 있을 때와 골 영역 근처에서 처리를 못하게 된다. 실점이 증가한 이유는 우리편의 수비 행위가 사용되지 않았기 때문이다. 그리고 득점이 낮아진 이유는 우리편이 벽이나 코너에서 공격할 수 있는 능력이 없어서 공이 슛 가능지역인 가운데 영역에 있어야만 슛 또는 상대진영으로 공을 쳐내기 때문이다. 대부분의 경기가 벽에서의 처리가 많이 필요하게 되는데 우리 팀은 이것을 위한 행위가 없기 때문에 불리하다. 실점이 현저히 증가하지 않은 이유는 골키퍼의 능력이 동일하기 때문이다.

game 3는 action level의 평가 기준 3번에 대한 것을 알아보기 위한 것으로 4개의 슛 행위를 더 사용함으로써 그만큼 슛을 위한 신뢰도가 증가하는 것을 평가한다. 평가 기준 3번 항목은 행위가 중첩되게 함으로써 신뢰성을 높이는 것인데 이 실험의 결과로는 행위의 수에 정비례해서 결과가 좋아지는 것은 아님을 보여준다.

표 2. game 2 : 우리 팀 좌우 윙의 행위중 Sweep_Ball(a_2), Push_Ball (a_9), Screen_Out_Ball (a_{11})이 빠진 경우.

Table 2. game 2 : Case without Sweep_Ball (a_2), Push_Ball (a_9), and Screen_Out_Ball(a_{11}) of left and right wing plays in friendly team.

횟수	득점 (실점)	marks	점유율	진영 점유율	충돌율	공격율
1	0(3)	-18	27.4	37.0	3.0	20.1
2	0(2)	-12	34.9	50.9	3.8	23.3
3	1(3)	-11	29.8	25.3	1.1	24.8
4	0(1)	-6	36.1	32.9	3.3	25.1
5	1(2)	-5	32.4	37.8	0.2	27.7
6	0(3)	-18	28.1	36.1	3.1	24.4
7	0(2)	-12	31.4	24.3	2.8	21.0
8	0(1)	-6	34.3	34.0	1.8	23.3
9	0(1)	-6	38.9	39.7	3.9	22.5
10	0(3)	-18	25.1	44.3	1.1	26.8
평균	0.2 (2.1)	-11.2	31.8	36.2	2.03	23.9

이 실험은 4개의 슛 행위중 1개만 사용하더라도 다른 실험과 비교해서 이기는 경우가 많이 발생했고 실점율이 낮아졌으며 득점력도 증가하였다. 또한 두 팀의 전력이 비슷하여 공격의 기회가 많이 낮아졌으며 대치상태로 인해 점유율도 높았다. 상대팀은 슛 행위를 3개를 더 사용함으로써 경기력이 더 나으나 슛을 위한 행위의 추

가에도 불과하고 낮은 득점력은 하위 계층인 circular_motion behavior의 완전하지 못한 제어능력에 기인할 수도 있다. Standard_Shoot은 move behavior를 이용하는데 여기에 사용된 알고리즘은 안정성 및 강인성이 보장되며 케환제어가 되고 있기 때문에 action level에서 전달되는 명령이 수행됨을 보장할 수 있다. 그러나 circular_motion behavior는 케환제어가 되는 것이 아니기 때문에 action level의 원 케칙을 따라간다는 것을 보장할 수 없다. 이것은 behavior level의 수행에 대한 신뢰도가 높지 않은 것을 보여준다. 그러므로 3번 평가 기준에 맞추어 더 높은 신뢰도 및 성취도를 얻기 위해서는 하위 계층의 신뢰도를 더 높여야 할 것이다.

표 3. game 3 : 우리 팀의 행위중 Cannon_shoot (a_6), Spin_Shoot (a_8), Turning_Shoot (a_7)이 빠진 경우.

Table 3. game 3 : Case without Cannon_shoot(a_6), Spin_Shoot (a_8), Turning_Shoot (a_7) of actions in friendly team.

횟수	득점 (실점)	Marks	점유율	진영 점유율	총 돌출율	공격율
1	0(0)	0	49.8	48.3	3.8	12.7
2	0(1)	-6	45.7	57.8	4.2	19.3
3	1(0)	7	49.7	26.7	6.4	2.7
4	0(1)	-6	32.6	42.7	4.2	11.4
5	1(2)	-5	28.0	40.9	3.8	24.6
6	2(2)	-2	38.3	45.2	3.9	21.3
7	0(0)	0	32.5	40.1	2.4	12.7
8	0(1)	-6	42.3	52.3	3.8	20.5
9	1(0)	7	40.8	28.9	4.5	3.9
10	1(1)	1	42.1	43.1	3.2	19.2
평균	0.6 (0.8)	-0.6	40.2	42.6	4.02	14.8

3. 실험 평가

위의 세가지 실험을 통해서 행위계층의 설계시 중요하게 고려해야 할 요소들을 정리하면 다음과 같다.

game 2를 통해 알 수 있듯이 모든 지역에서 공을 cover할 수 있게 행위가 설계가 되어야 하며 로봇의 속도도 대응해야 한다는 것이다. 이것은 로봇의 개인 능력(개인 전술)의 중요성을 의미한다.

행위의 중첩이 필요하다. game 3을 통해 알 수 있듯이 같은 상황에서도 선택될 수 있는 여러 행위를 설계하여 시스템의 신뢰도, 성취도를 향상시키도록 하는 것이 필요하다.

모든 계층의 신뢰도 향상이 필요하다. game 3의 실험에서 알 수 있는 것은 행위계층의 신뢰도를 향상시켜도 하위 계층인 행동계층의 신뢰도가 낮으면 좋은 성능을 얻을 수 없다는 것이다. 계층적인 구조의 설계시 기

본 가정은 각 계층이 제대로 작동하고 있다는 것이다. 행위 계층의 하위 계층이 잘 작동하고 있다는 가정하에서 설계되었기 때문에 설계가 잘되었어도 하위 계층에 문제가 있다면 전체 시스템의 성능은 크게 향상되지 않는다.

상대편에 대한 고려가 필요하다. game 1처럼의 성능을 가지고 있어도 단순히 상대편이 존재한다는 것만으로도 월등한 결과를 얻을 수 없었던 것은 상대편에 대한 고려가 없었기 때문일 것이다. 예를 들면 상대편 골키퍼의 위치에 따라서 골대로 슛을 할 수 있는 기회는 줄어들기 때문에 다른 행위를 선택해야 하는데 현재는 그런 고려를 안하고 있다. 상대편이 골대를 가로막고 있어 슛을 할 수 없다면 반칙을 유도하든지 다른 어떤 방법을 강구해야 할 것이다. 이의 사항은 pass 같은 행위의 필요성을 강조한다.

위처럼 실험결과로부터 정리된 설계 필요 요소들은 본 논문에서 제안된 행위계층의 설계 및 평가기준이 적절하다는 것을 설명하고 있다.

V. 결론

영상 정보를 이용한 축구로봇시스템을 위한 복합제어 구조는 4가지 계층으로 구성되는데 즉, 역할, 행위, 행동 및 실행계층으로 되어 있다. 본 논문에서는 복합제어구조중 가장 중요한 요소로 행위계층에 대한 설계 및 평가에 관한 것을 제시하였다.

행위 계층에서는 주어진 역할에 따라 목적을 이루기 위해 여러 가지 할 수 있는 행위(action)들 중 그 개체가 제일 적합한 것을 선택하여 하위 계층인 행동 계층에 명령을 내려보내 로봇을 움직이게 한다. 본 계층을 구현하는 목적은 모든 개체에게 동일한 능력을 구현시켜서 각각이 모두 우수한 개체가 되도록 하고 신뢰도 및 일의 성취도를 배가시키기 위한 것이다. 행위 계층은 로봇이 축구를 수행할 수 있도록 행동을 유발시키는 18개의 기본 행위들과 제시된 이 기본 행위들중 공의 위치등, 환경에 따라서 적절하게 행위를 선택하는 행위 선택기로 구성되어 있다. 기본 행위들과 행위 선택기의 설계는 본 논문에서 제안된 설계 기준과 평가 방법에 의해서 이루어졌으며 효용성은 실험을 통해서 증명되었다.

참고문헌

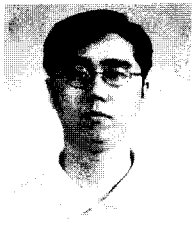
- [1] A. M. Meystel, "Intelligent control : issues and perspectives," *Proc. IEEE Workshop on Intelligent Control*, pp. 1-15, 1985.
- [2] G. N. Saridis, "Intelligent controls for advanced automated processes," NASA CP-2180, May, 1980.
- [3] K. M. Passino, "Bridging the gap between conventional and intelligent control," *IEEE Control Systems*, pp. 12-18, 1993.
- [4] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 1, March, 1986.
- [5] R. E. Fayek et. al., "A System architecture for a

mobile robot based on activities and a blackboard control unit," *IEEE Conf. of Robotics and Automation*, pp. 267-274, 1993.

- [6] J.-H. Kim, H.-S. Shim, H.-S. Kim, M.-J. Jung, I.-H. Choi and K.-O. Kim, "A cooperative multi-agent system and its real time application to robot soccer," *IEEE Int. Conf. on Robotics and Automation*, Albuquerque, New Mexico, pp. 638-643, 1997.
- [7] H.-S. Shim, H.-S. Kim, M.-J. Jung, I.-H. Choi, J.-H. Kim and J.-O. Kim, "Designed distributed control architecture for cooperative multi-agent system and its real-time application to soccer robot," *Journal of Robotics and Autonomous System* vol. 21,

no. 2, pp. 149-165, September, 1997.

- [8] H.-S. Shim, M.-J. Jung, H.-S. Kim, J.-H. Kim, and P. Vadakkepat, "A hybrid control structure for vision based soccer robot system," *Journal of Intelligent Automation and Soft Computing*, to appear in 1999.
- [9] H.-S. Shim, M.-J. Jung, H.-S. Kim, J.-H. Kim, and P. Vadakkepat, "Role level design in a hybrid control structure for a vision-based soccer robot system," AROB '99, Bebbu, Japan, Feb., 1999.
- [10] H.-S. Shim, "Design and implementation of multi-robot cooperation system using BIOS : its application to robot soccer, Ph.D. Thesis, KAIST, 1998.



심 현 식

1969년 2월 2일생, 1991년 한국과학기술원 전기 및 전자공학과 졸업, 동대학원 석사(1993년), 박사(1998년), 1999년~현재 LG교통정보연구원, 선임연구원, 관심분야는 다개체시스템,

ITS 시스템 설계.



성 윤 경

1961년 3월 1일생, 1982년 영남대학교 기계설계과 졸업, 1985년 연세대학교 기계공학(석사), 1997년 알라바마주립대학교 기계공학(박사), 1998년~현재 한국원자력연구소, 관심분야는 동력학 및 제어, 다변수 비선형 강인제어, 지능구조물.

분야는 동력학 및 제어, 다변수 비선형 강인제어, 지능구조물.



김 중 환

1981년 서울대학교 전자공학과 졸업, 동대학원 석사(1983), 박사(1987), 1988년~현재 한국과학기술원 전기 및 전자공학과 교수, IEEE, KITE, KIEE, KCAS 회원.