

# Development of Motion Management System for the Robot Soccer Using Multiple Humanoid Robots

Jae-Man Lee, Hye-Yeun Chun, Yusupov Dilmurod, Keong-Seok Ko and Tae-Koo Kang

**Abstract** We address the motion management system to apply it to the robot soccer game. The soccer system has a vision system, a host computer which manages the strategies and position control of the soccer robots, and a communication system between a host computer and three soccer robots. The player robots can take 32 motions according to the game situation. To avoid the obstacles such as the opponent player robots, we proposed the novel motion control method which consists of CM and PL-AMS. The CM decides the avoid direction considering the position and size of the obstacles. PL-AMS selects the appropriate avoidance motion by the fuzzy logic. In addition, we have the strategy for the game situation such as the offense and defense by using the seven subregion divisions. Moreover, to prevent the ping-pong effect, we provided the buffer region between subregions.

## 1 Introduction

Robot soccer is a good example of a multi-agent system in a real-world application which can be moderately abstracted. Multi-agent systems deal with research subjects such as cooperation protocol by distributed control, effective communication and fault-tolerance while having efficiency of cooperation, adaptation, and robustness, and being in real-time.

Soccer robot control can be divided into three control schemes. The first scheme is a remote-brainless soccer robot system in which a host computer controls the robots by commanding the robots motion. The second scheme is a vision-based one which controls the robots by processing information using the vision with a

---

J.-M. Lee (✉) · H.-Y. Chun · Y. Dilmurod · K.-S. Ko · T.-K. Kang (✉)  
Department of Information and Communication Engineering, Sangmyung University,  
Cheonan-si, Chungcheongnam-do 330-720, Republic of Korea  
e-mail: ejm303@naver.com  
URL: <http://ittech.smuc.ac.kr>

T.-K. Kang  
e-mail: [tkkang@smu.ac.kr](mailto:tkkang@smu.ac.kr)

supervisor processor and sends commands to robots directly. The other scheme is a robot-based one, wherein each autonomous robot can make a decision based on information from its sensors and, if needed, can send information to others. Robots are designed for autonomous operation [1].

Among the three schemes, only the remote-brainless soccer system has been dealt with, in this paper as it is easy to develop, though the robots are designed for the robot-based system. A control structure, several behaviors, and actions are developed for a soccer robot. Modified zone defense as a basic strategy and several special strategies for fouls are applied to multiple humanoid robots for ANDROSOT'15 (Robot World Cup Soccer Tournament). A soccer game of the ANDROSOT played by 3 robots from each team is used as a test bed for the multi-agent systems. In Sects. 2 and 3, we described the soccer robot system and control structure in detail.

## 2 Soccer Robot Systems

There are various methods including the hardware, such as CPU, actuators, sensors, and software, such as control algorithm, strategies to make a soccer system for ANDROSOT which consists of multiple humanoid robots per a team. Therefore, we have to select the appropriate hardware and software among the various methods.

In this section, we focus on the overall architecture of the soccer robot system and the soccer robot system. The specific configurations and functions are explained in detail as follows.

### 2.1 Architecture of Area Soccer Robot System

Figure 1 shows the overall architecture of the soccer system for ANDROSOT. As shown in Fig. 1, the soccer robots having motion mechanism, a vision system, a host computer which manage the strategies and position control of the soccer robots according to the offense or defense situation, and a communication system between a host computer and three soccer robots are needed for robot soccer game.

### 2.2 Soccer Robot System

Figure 2 illustrates a robot system we used in the soccer game. As shown in Fig. 2b, the soccer robot has 34 cm × 18 cm × 12 cm dimension and 17 DOF [3]. It involves a 2-axis gyro sensor, an IR sensor, and a tilt sensor to detect the

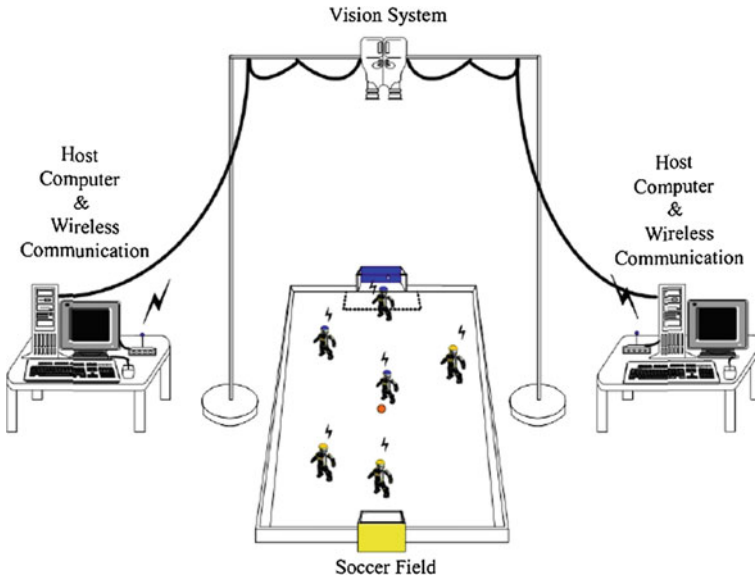


Fig. 1 Overview of the soccer system including a vision system [2]

inclination fall down of the robot body. To control the servo motor, we used the MR-C3024FX (Fig. 2a) which can control 24 servo motors and has various control interface, such as a gyro sensor, a Bluetooth module, and a remote control.

Basically, each robot system can make over 120 motions and in the case of our soccer game, we use the 32 various motions including the block motions for a

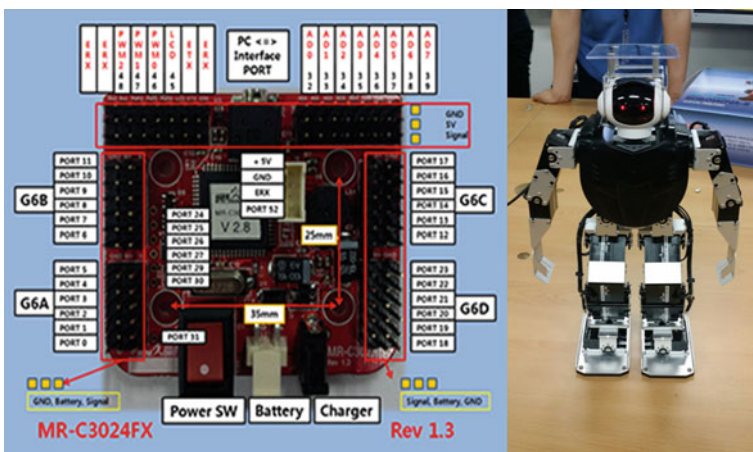


Fig. 2 A soccer robot system and its controller

goalkeeper robot. Each robot system selects the appropriate motion corresponding to the game situation.

### 3 Motion Control and Path Planning Algorithm

In this section, we deal with the control algorithms for the obstacle avoidance method and the optical path generation. Detailed methods for the control structure are described as follows.

#### 3.1 Motion Control for Obstacle Avoidance

To avoid collisions between our player robots and the obstacles such as the opposite player robots, we need to determine the avoidance direction and the avoidance motion of our player robots. Therefore, we proposed novel method called Complex Measure (CM) and the Fuzzy Logic-based Avoidance Motion Selector (FL-AMS) to solve the given problem.

##### 3.1.1 Decision of Avoidance Direction Using Complexity Measure (CM)

To calculate the avoidance direction, we use the environment information such as obstacle's position and size as the input data. To decide the avoidance direction, we propose the complexity measure (CM) as follows:

$$CM = \begin{cases} \sum_{l=1}^n k_l + 2^{(\pi/2 - \theta_l)} + w_l, & \theta_l > 0 \\ - \sum_{l=1}^n k_l + 2^{(\pi/2 + \theta_l)} + w_l, & \theta_l < 0 \\ \sum_{l=1}^n 2k + w_l^k \sum_{l=1}^n, & \end{cases} \quad (1)$$

In (1), the CM is determined by three factors: distance from the robot, width of the obstacle, and angle from the robot to the obstacle. From the CMs, we can get the final avoidance direction as

$$Direction = \text{sign}\left(\sum_{all\ obstacle} CM\right) = \begin{cases} Direction < 0, & \text{turn left} \\ Direction > 0, & \text{turn right} \end{cases} \quad (2)$$

The avoidance direction is decided by the summation of all CMs for all obstacles. In (2), the CMs for obstacles located on the left side of the robot have

negative values, and the CMs for obstacles located on the right side of the robot have positive values. Therefore, if the direction is a positive value, then the robot turns to the left, and if the direction is a negative value, then the robot turns to the right.

### 3.1.2 Decision of Avoidance Motion Using Fuzzy Logic-Based Motion Selector

After we decide the avoidance direction for the obstacle, avoidance motion has to be selected depending on the environment information. Avoidance motions have to be varied under width of obstacle and open space that a robot can avoid. Therefore, we propose the fuzzy logic-based avoidance motion selector (FL-AMS) [4].

In order to perform the fuzzification, a width difference between an obstacle width and a robot width  $w_i$  and space difference between avoidance space width and robot width  $S_i$  are set up as the input variables. The term sets of the input and output variables are selected as follows:

$$T(w_i) = \{A_1, A_2, A_3, A_4, A_5\} = \{NL, NS, Z, PS, PL\}, \quad (3)$$

$$T(s_i) = \{B_1, B_2, B_3, B_4, B_5\} = \{NL, NS, Z, PS, PL\}, \quad (4)$$

$$T(f_i) = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7\} = \{ST, SS, SF, RS, RF, TS, TF\}, \quad (5)$$

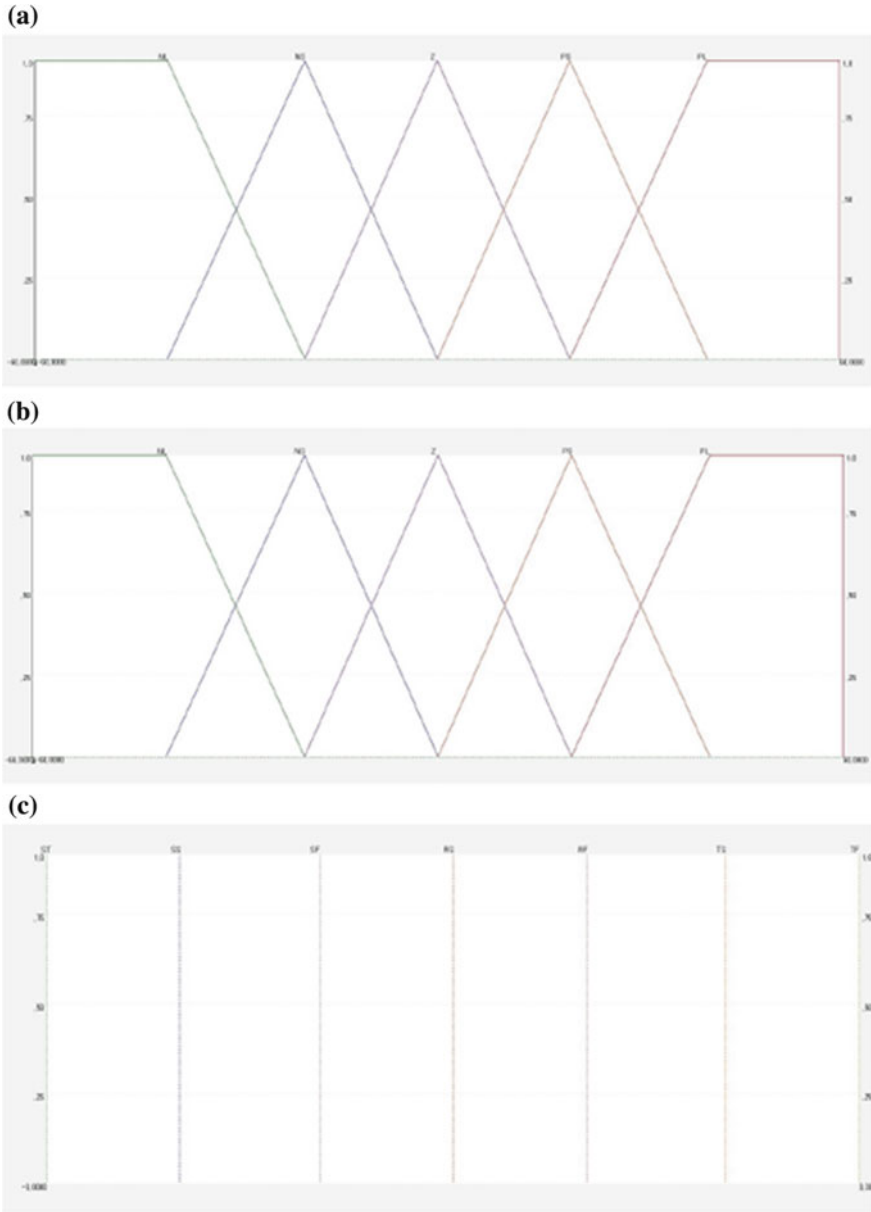
where fuzzy sets  $A_1, A_2, A_3, A_4$ , and  $A_5$  are, respectively, denoted as negative large (NL), negative small (NS), zero (Z), positive small (PS), and positive large (PL) for the input variable  $w_i$ . Fuzzy sets  $B_1, B_2, B_3, B_4$ , and  $B_5$  are also, respectively, denoted as negative large (NL), negative small (NS), zero (Z), positive small (PS), and positive large (PL) for the input variable  $d_i$ . Fuzzy sets  $C_1, C_2, C_3, C_4, C_5, C_6$ , and  $C_7$  are, respectively, denoted as Stop (ST), Slip step + Slip step (SS), Slip step + Forward step (SF), Rotation step + Slip step (RS), Rotation step + Forward step (RF), Turning step + Slip step (TS), and Turning step + Forward step (TF) which are the combinations of two motions in Slip step (S), Forward step (F), Rotation step (R), and Turning step (T) excluding the Stop motion (ST) for the output variable  $f_i$ .

The partitions and the shapes of the membership functions are shown in Fig. 3, where triangular-type membership functions and fuzzy singleton-type membership function are, respectively, used for the input and output variables. As shown in Fig. 3, each input is divided into five subsets, so that two inputs will construct 25 fuzzy rules and output value of fuzzy logic determining the walking motion type is constructed in seven membership functions. The fuzzy rule base used for the proposed FL-AMS system is described in Table 1, where each rule can be described by

Rule  $R(j_1, j_2)$ :

$$\text{IF } w_i \text{ is } A_{j_1} \text{ and } d_i \text{ is } B_{j_2}, \text{ then } f_i \text{ is } C_{f(j_1, j_2)} \quad (6)$$

$$j_1 \in \{1, 2, 3, 4, 5\}, j_2 \in \{1, 2, 3, 4, 5\}, f(j_1, j_2) \in \{1, 2, 3, 4, 5, 6, 7\}$$



**Fig. 3** Membership functions of FL-AMS: **a** input variable  $w_1$ , **b** input variable  $w_2$ , and **c** output variable  $w_3$

**Table 1** Fuzzy rule for FL-AMS

| $f_i$ |    | $S_i$ |    |    |    |    |
|-------|----|-------|----|----|----|----|
|       |    | NL    | NS | Z  | PS | PL |
| $w_i$ | NL | ST    | SS | SF | SF | SF |
|       | NS | ST    | SS | SF | SF | SF |
|       | Z  | ST    | RS | RF | RF | RF |
|       | PS | ST    | RS | RF | RF | RF |
|       | PL | ST    | TS | TF | TF | TF |

To determine the final output of the FL-AMS, we use the weighted average method described by

$$f_i = f(w_i, s_i) = \frac{\sum_{j_1=1}^{j_1=5} \sum_{j_2=1}^{j_2=5} u(j_1, j_2) \cdot v(C_{f(j_1, j_2)})}{\sum_{j_1=1}^{j_1=5} \sum_{j_2=1}^{j_2=5} u(j_1, j_2)} \tag{7}$$

where  $v(C_{f(j_1, j_2)})$  is the crisp value of the fuzzy set,  $C_{f(j_1, j_2)} \cdot u(j_1, j_2)$  is the fire strength of the rule  $R(j_1, j_2)$  and can be described by

$$u(j_1, j_2) = \min(\mu_{A_1}(w_i), \mu_{B_2}(s_i)) \tag{8}$$

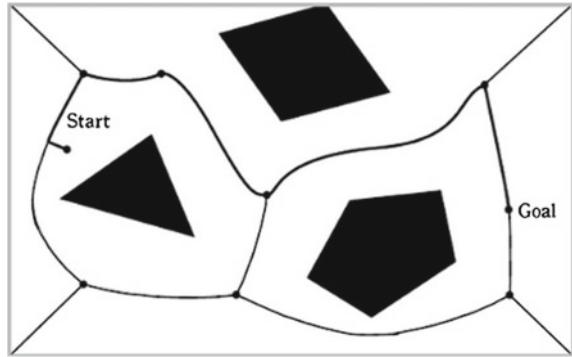
Based on the width difference between obstacle width and robot width  $w_i$  and space difference between avoidance space width and robot width  $s_i$ , seven evaluation values  $f_i, i \in \{1, 2, 3, 4, 5, 6, 7\}$  for seven avoidance motions can be determined by the FL-AMS. The walking motion type consists of seven motions, ST, SS, SF, RS, RF, TS, and TF which are the combinations of two motions in sidestep walking (S), forward walking (F), rotation walking (R), and turning (T) excluding the stop motion (ST). Table 1 shows the fuzzy rule for the walking motions.

### 3.2 Determination of Optimal Path

To reach the objective point, the optimal path generation from current location to target location is necessary. We can examine this problem in terms of efficiency and safety.

The path planning method can be largely divided into two parts: (1) Pint-To-Point (PTP) and (2) Complete Coverage (CC). PTP path planning is searching the optimal path from a start point to an end point through points. The objective of PTP path planning method is that we find the path having efficiency in terms of minimum energy consumption or risk reduction. On the other hand, the objective CC path planning is that a robot covers the all sections as far as possible for a given area. Therefore, we use the PTP path planning method because it is suitable for our purpose playing the soccer game.

**Fig. 4** Voronoi diagram-based path planning method



In addition, an algorithm conducting the PTP path planning is needed. To achieve this purpose, we consider algorithms in terms of efficiency and safety. In terms of efficiency, the robots playing the game can move using the relatively short course to reach the target position under offense or defense situations. In terms of safety, the robot in the soccer game does not foul like colliding with the robots of the opposite team.

To satisfy the above-mentioned conditions, we use a Voronoi diagram to generate the optimal path. Figure 4 shows an example of the Voronoi diagram-based path planning. In mathematics, a Voronoi diagram is a partitioning of a plane into regions based on distance to points in a specific subset of the plane. That set of points (called seeds, sites, or generators) is specified beforehand, and for each seed, there is a corresponding region consisting of all points closer to that seed than to any other. These regions are called Voronoi cells.

Moreover, the PTP path planning can be used in the free-ball situation such as free-kick, goal kick, and penalty kick due to the non-interference property of the Voronoi diagram between the robots belonging our team and those of the opposite team.

### ***3.3 Strategies for the Game Condition***

The basic strategy for the soccer robots was the zone defense or offense. However, the applied strategies were needed to the complicated situations in the game.

Each robot has its own role such as a striker, defender, and a goalkeeper. Firstly, we divided the ground into the three parts according to its role. At the beginning of development, the concept of zone defense was that the robot selected an appropriate action among basic actions if only the ball is located in its own area and the other robots did not select any actions because the ball was out of their areas. But, this concept had two problems: (1) motion conflict of nearby two robots when the ball is located in one of the boundary areas and (2) ping-pong effect [5]. The corresponding formation will be formed according to the position of the ball, which



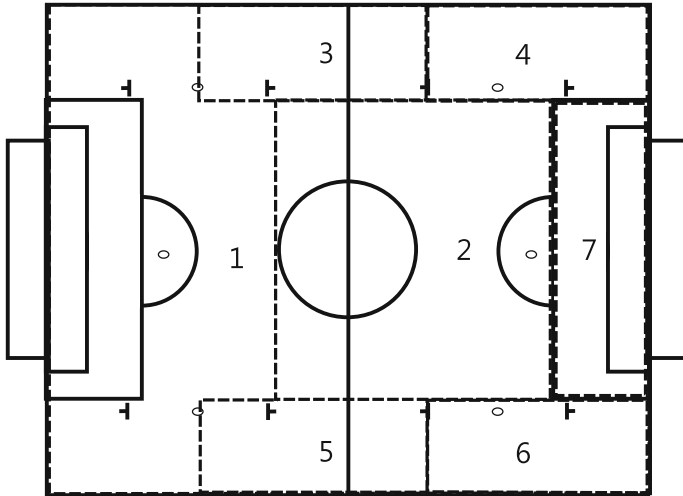


Fig. 5 Subregions for formation and play mode

causes a serious problem when the ball moves back and forth between two regions. The player robots will keep changing formation and consume their energy unnecessarily.

To solve the problem, we made a play mode and a formation through the field division according to the position of the ball. The play mode consists of offense, defense, and free-ball mode and depends on the relative position among the positions of our opposite robots and the location of the ball except the free-ball mode.

Player robots change the formation according to the position of the ball within seven subregions as shown in Fig. 5. If the ball locates within the defender goal area (region 2, 4, 6, and 7), then the major mission of the robots in these regions is that kick the ball out of defender goal area. When the ball is close to the opponent goal area or the position of our player robot with respect to the ball position is closer than that of the opponent player robot, we regard this situation as the attack mode and form the attack formation. As mentioned above, to prevent the ping-pong effect, we provide the buffer region between subregions, which keep the formation of the team although the ball moves back and forth between two subregions.

## 4 Conclusion

We address the motion management system and its application to the robot soccer game. The soccer system has a vision system, a host computer which manages the strategies and position control of the soccer robots, and a communication system between a host computer and three soccer robots. The player robots can take 32 motions according to the game situation.

To avoid the obstacles such as the opposite player robots, we proposed the novel motion control method which consists of CM and PL-AMS. The CM decides the avoidance direction considering the position and size of the obstacles. PL-AMS selects the appropriate avoidance motion by the fuzzy logic.

In addition, we have the strategy for the game situation such as the offense and defense by using the seven subregion divisions. Moreover, to prevent the ping-pong effect, we provide the buffer region between subregions. In the demonstration tournament, we played games with two periods of up to seven minutes (or until a goal was scored), and on average 4–5 goals were scored in each match, which was essential for us to achieve a spectacular game.

## References

1. Kim, J.-H., Shim, H.-S., Kim, H.-S., Jung, M.-J., Choi, I.-H., Kim, J.-O.: A cooperative multi-agent system and its real time application to robot soccer. *Int. Conf. Robotic Autom.* **1**, 638–643 (1997)
2. [www.fira.net/](http://www.fira.net/)
3. <http://minirobot.co.kr/>
4. Kang, T.-K., Choi, I.-W., Park, G.-T., Lim, M.-T.: Local environment recognition system using modified SURF-based 3D panoramic environment map for obstacle avoidance of a humanoid robot **10** 1–13 (2013)
5. Wu, S.L., Tung, K.C., Liou, Y.R., Lin, W.H., Wu, M.H.: A multi-agent algorithm for robot soccer games