Safety region: an index for evaluating the situation of RoboCup Soccer game

Junya Maeno, Akeru Ishikawa, Kazuhito Murakami and Tadashi Naruse
School of Information Science and Technology
Aichi Prefectural University, Nagakute-cho, Aichi, 480-1198 JAPAN

Abstract

In this paper, a "safety region" is proposed as an index for evaluating the situation of RoboCup Soccer games, especially for RoboCup Small Size League (SSL). The safety region is defined as the areas in the soccer field that the robots can keep the goal when an opponent robot shoots the ball in the areas. The shape of the safety region varies by the number of defense robots and defense strategy. It is difiicult to obtain the safety region analytically. We show an approximate calculation method to obtain the safety region for a cooperative shot as well as a direct shot. Since it is shown by the experiments that the computation time of the safety region is less than 2 msec using a typical PC, we can use it to evaluate the situation at a frame rate. Using the safety region index, we show an algorithm which determines the positions of defense robots. Experimental results show that the safety region index will work appropriately.

1 Introduction

In the recent RoboCup Small Size Robot League, strategies for attacking and defense are growing high, and the strategy that dynamically changes the number of defense robots depending on the game's situation is often used. In typical SSL strategy, the potential fields [1] and the playbook[2] are used for determining the action selection, and as a result, the number of defense robots. Moreover, cooperative plays such as a direct play [3] are commonly used in the SSL games. To effectively defend for such plays, we should compute the situation of the game in real time and determine the positions of the defense robots.

There are indexes for determining a mark robot[4] and deciding a passing robot[5]. We think we need an another index to determine the defense robots. In this paper, we introduce a new concept of "safety region" index for determining the action selection, especially for the number of defense robots, and discuss how we determine the number of defense robots based on the safety region. The safety region is defined as a region that the opponent's shot from the inside of the safety region can be blocked by the teammate robot(s). The shape of the safety region varies depending on the number of defense robots and the defense strategy.

The safety region can be used as an index to evaluate the defense strategy. Inversely, it can be used as an index to determine the number of defense robots according to the situation of a match. Since it is difficult to get the accurate safety region, we propose algorithms that compute the approximate safety region. The algorithms are given for a single shot and a cooperative shot, respectively. Finally, we discuss the availability of the safety region as an index of strategy evaluation and the availability of the approximate safety region.

2 Safety region

2.1 Definition

A concept of "safety region" is simple. It is defined as a region that the teammate robot(s) can keep the goal when an opponent robot shoots the ball from the inside of the safety region if teammates are positioned properly according to their defense strategy. Remaining region of the field given by removing the safety region is called "unsafety region".

In the following discussion, as a first step, we do not consider the chip shot or the curved shot in the following discussion.

2.2 Calculation of safety region

The calculation of the safety region depends on how the shot action is taken, i.e. a single shot or a cooperative shot, and how the team keeps the goal, i.e. defense strategy and the number of defense robots. It takes much time to compute the precise safety region. Therefore, in the following, we describe procedures to compute the approximate safety region.

2.2.1 Calculation of approximate safety region: single shot case

Defense robots will move according to their strategy so that we assume the right positions of the defense robots are given at any time. Let $\mathbf{b}$ be the position of the ball at time t. Let $\mathbf{r}_i$ be the position of the defense robot i at time t. Let L_r and L_l be the lines connecting $\mathbf{b}$ and the right and left goalposts, respectively. (See figure 1.) Defense robots usually stand on the inside of the region the lines L_r and L_l make. Let $\mathbf{p}_{r,i}$ be the cross-point between the line L_r and the line perpendicular to L_r through $\mathbf{r}_i$, and let $\mathbf{p}_{l,i}$ be the cross-point similarly defined for the line L_l and $\mathbf{r}_i$. Assuming $||\mathbf{r}_i - \mathbf{p}_{j,i}|| \geq R$, where $||\cdot||$ means a length of a vector $\cdot$, calculate the following equations for each defense robot i.

Compute,

$$t_s = \frac{||\mathbf{b} - \mathbf{p}_{j,i}||}{v_s}, \quad t_i = \frac{v_i}{a_i}, \tag{1}$$

where t_s means the time for the ball to move from $\mathbf{b}$ to $\mathbf{p}_{j,i}$ by the speed v_s and t_i the time for the robot i to get to the maximal speed (starting from speed 0).

If $t_i > t_s$, then compute whether equation

$$\frac{||\mathbf{b} - \mathbf{p}_{j,i}||}{v_s} > \sqrt{\frac{2(||\mathbf{r}_i - \mathbf{p}_{j,i}|| - R)}{a_i}}, \quad (j = r, l) \tag{2}$$

is satisfied or not, otherwise compute whether equation

$$t_s > t_i + \frac{||\mathbf{r}_i - \mathbf{p}_{j,i}|| - R - \frac{a_i t_i^2}{2}}{v_i}, \quad (j = r, l). \tag{3}$$

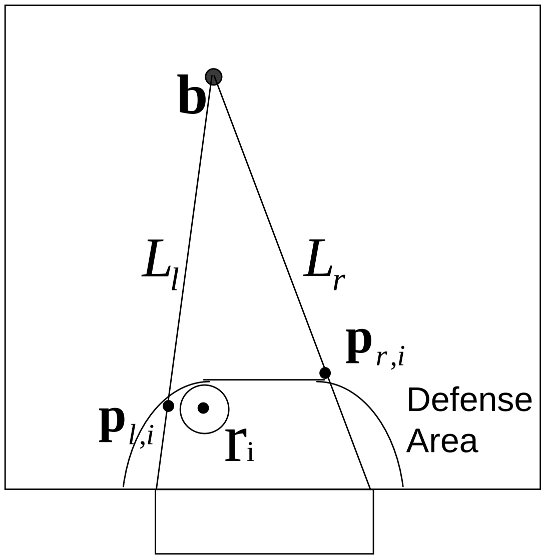

Figure 1: Definition of the safety region: single shot case

is satisfied or not, where v_s, a_i, v_i and R are the kicked speed of the ball, the maximal acceleration and velocity of the defense robot i, and the sum of radii of the defense robot and the ball, respectively.

When Equation (2) or (3) is satisfied for $j = r$ and l, the position of the ball $\mathbf{b}$ is defined as a point in the safety region. In case there are more than one defense robot, if at least one of them satisfies Eq. (2) or (3), $\mathbf{b}$ is defined as a point in the safety region. When $||\mathbf{r}_i - \mathbf{p}_{j,i}|| < R$ is satisfied, $\mathbf{b}$ is also defined as a point in the safety region.

2.2.2 Calculation of approximate safety region: cooperative shot case

When the two opponent robots make the shot cooperatively, a typical example is a direct play[3] which is an action that shooting robot kicks the ball immediately after receiving it from a passing robot (figure 2), we have to take the passing robot into consideration since the defense robots keep their goal at the positions that defend against the attack of both robots. In this case, we have to take the passing time into consideration to compute the safety region.

Let $\mathbf{b}$ and $\mathbf{e}$ be the positions of the ball and the shooting robot at time t, respectively, and $\mathbf{r}_i$ be

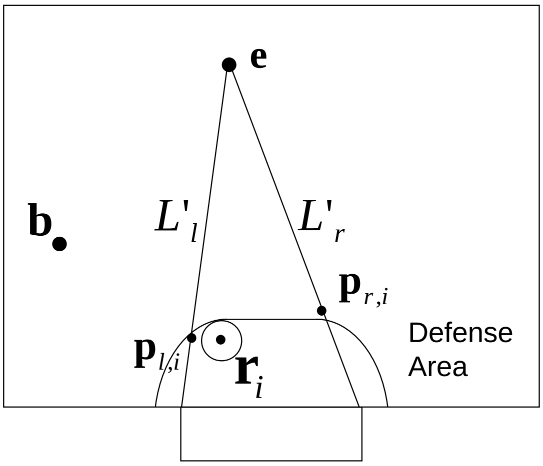

Figure 2: Definition of the safety region: cooperative shot case

the position of each defense robot i at time t. Let L'_r and L'_l be the lines connecting $\mathbf{e}$ and the right goalpost and $\mathbf{e}$ and the left goalpost, respectively. (See figure 2.) We assume the goalkeeper stands in the defense area and moves along the defense area and other defense robots stand outside defense area and move along the defense area. Then, let $\mathbf{p}_{r,i}$ be the cross-point between the line L'_r and the line perpendicular to L'_r through $\mathbf{r}_i$, and let $\mathbf{p}_{l,i}$ be the cross-point similarly defined for the line L'_l and $\mathbf{r}_i$. Assuming that the passing robot holds the ball at time t and makes the cooperative play, following equation is obtained for computing the safety region.

Compute,

$$t_p = \frac{||\mathbf{e} - \mathbf{b}||}{v_p}, \quad t_s = \frac{||\mathbf{p}_{j,i} - \mathbf{e}||}{v_s}, \quad t_i = \frac{v_i}{a_i}, \quad (4)$$

where t_p means the passing time between $\mathbf{b}$ and $\mathbf{e}$, t_s and t_i are the same ones in eq. 1.

If $t_i > t_p + t_s$, then compute whether equation

$$t_p + t_s > \sqrt{\frac{2(||\mathbf{p}_{j,i} - \mathbf{r}_i|| - R)}{a_i}}, \quad (j = r, l) \quad (5)$$

is satisfied or not, otherwise compute whether equation

$$t_p + t_s > t_i + \frac{||\mathbf{p}_{j,i} - \mathbf{r}_i|| - R - \frac{a_i t_i^2}{2}}{v_i}, \quad (j = r, l) \quad (6)$$

is satisfied or not, where, v_p, v_s, v_i, a_i and R are the speed of the ball at passing, the speed of the ball at shooting, the maximal velocity and acceleration of the defense robot i and the sum of radii of the defense robot and the ball, respectively.

If equation (5) or (6) is satisfied or there is no pass line[1] between $\mathbf{b}$ and $\mathbf{e}$, $\mathbf{e}$ is a point in the safety region. In case there are more than one defense robot, if at least one of them satisfies Eq. (5) or (6), $\mathbf{e}$ is a point in the safety region. When $||\mathbf{p}_{j,i} - \mathbf{r}_i|| < R$, $\mathbf{e}$ is also a point in the safety region.

3 Experiment

In this section, we show the experimental results of the safety region for the direct play[3]. We use the defense strategy of RoboDragons[6] here since we know all its details.

3.1 Method of experiment

The safety region should be calculated analytically, however, it is hard to do the analytic computation so that we calculate it on each of the mesh points which are given by dividing the field every 40 mm, and we give the approximate safety region using the model discussed in section 2.2.

On the other hand, to testify the correctness or preciseness of the proposed safety region, we compared the proposed safety region with the result of simulation, where simulation was done by using the game simulator embedded in the RoboDragons system. Here we show a simulation procedure for the cooperative play in the followings.

1. Divide the field into n mesh points, where $n = 14888$ in this experiment. Put the ball on the initial position $\mathbf{b}$, one of the mesh points. Also put the attacking robots on the mesh points, one around the ball and the other one a given point $\mathbf{e}_i$, a shooting position[2]. Place defense robot(s) on defending position(s) according to the strategy algorithm of the RoboDragons system.

[1] The length of the perpendicular from $\mathbf{r}_i$ to the line connecting $\mathbf{b}$ and $\mathbf{e}$ is less than the radius of the robot, it can block the pass.

[2] We assume that the robot can kick the ball toward any direction.

2. At time t, move the ball from $\mathbf{b}$ to $\mathbf{e}$ with the passing velocity v_p.

3. Move the ball on the shooting line from $\mathbf{e}$ which is the farthest line to the defense robots when the ball arrives at shooting position $\mathbf{e}$ at time t_e. (Shooting line is calculated at time t.)

4. Simulate the movement of the defense robots and judge if a goal is achieved or not. If the goal is achieved, then the point $\mathbf{e}$ is a point in the unsafety region, otherwise it is a point in the safety region.

5. Repeat steps 2 ... 4 for each point in the mesh by replacing $\mathbf{e}$ into it.

The initial position of the ball used in the simulation is selected from the logged data of the 8 games held in RoboCup Japan Open 2009 and RoboCup 2009, i.e. kick off, direct and indirect free kick points are the candidates of the initial position of the ball, and it is randomly selected from the candidates. Passing velocity and shooting velocity of the ball are 4.0 m/sec and 8.0 m/sec, respectively, which are the typical values in the SSL. The acceleration and velocity of the defense robot are 2.0 m/sec^2 and 0.6 m/sec, respectively, which are the measured values.

3.2 Experimental results

3.2.1 Coincidence rate

We compared the approximate safety region with the one obtained by the simulation for the one defense robot case and the two defense robots case under the RoboDragons's defense strategy. Figures 3 through 6 show the examples of the experimaental results.

In the figures, the white region is a safety region and the gray region is an unsafety region except the defense area in which the goalkeeper stands. There are two large gray areas in the field and the shapes of each area are similar between the proposed and simulated results. However, the down-left corner of the field does not match between the approximate results and the simulation results. (In the simulated result, there are many small gray dot-like areas. This is caused by the noise model implemented in the simulator.) These figures show that the approximate safety region is a well approximation of the safety region and can use as the safety region.

Table 1: Coincidence rate (Direct play[3])

	coincidence rate R
one defense robot	0.889
two defense robots	0.869

To show how much portion of safety and unsafety regions coincide with between the approximate method and the simulation, we define a coincidence rate R by the following equation.

$$R = \frac{\text{number of coincident points}}{\text{number of all points on the mesh}} , \qquad (7)$$

where coincident points are points that the results of approximate method and simulation coincide with. Table 1 shows the result which is the average of 10 trials.

3.2.2 Computation time

Using the three typical computers, we obtained the computation time for the proposed method. Table 2 shows the result.

In our system, given time for the computation of the safety region is at most 5 msec from our experience of the development of the RoboDragons system when we use the safety region in strategy computation in real time. From Table 2, we can realize the real time computation using the Xeon processor.

4 A defense strategy using the safety region

In this section, we propose a defense strategy using the safety region against the cooperative play.

4.1 A defense algorithm using the safety region against the direct play

When n robots have already placed in defense positions and $(n+1)$th robot is placed in defense position, the following algorithm determines the position of the $(n+1)$th robot.

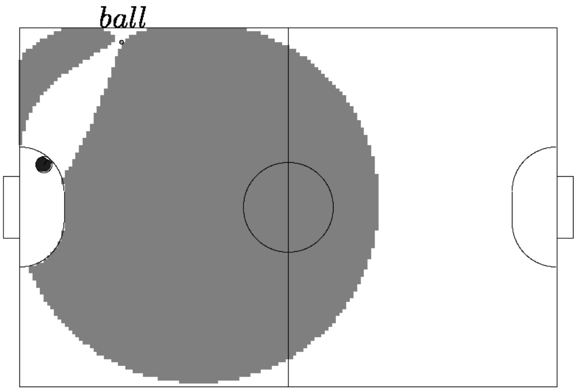

Figure 3: Safety region: one defense robot case

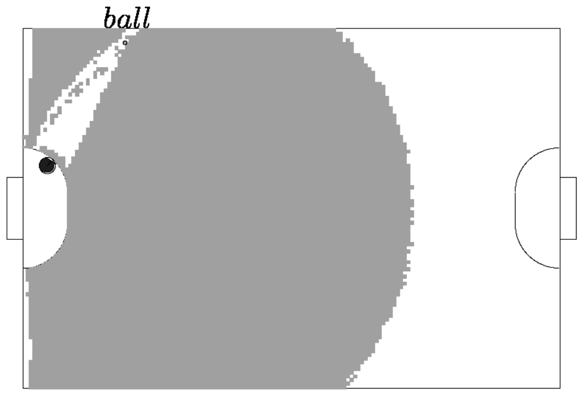

Figure 4: Safety region: simulation result corresponding to Fig. 3

Table 2: Computation time

CPU	Memory	one defense robot	two defense robots
Pentium 4 2.8GHz	1 GB	4.5 msec	5.6 msec
Athron64 X2 4200+	512 MB	4.6 msec	6.1 msec
Xeon 3.3 GHz	2 GB	1.5 msec	1.9 msec

1. Letting the positions of n defense robots be $\mathbf{r}_i$ $(i = 1, ..., n)$, obtain the safety region and the unsaftey region. Number each connected component in the unsafety region. Let it be N_k $(k = 1, ...)$

2. Search a connected component with maximal area (N_m) and compute the gravity center $\mathbf{G}$ of N_m.

3. Place $(n+1)$th robot at the cross-point of the line L_g and the defense line (a bit outside of defense area), where L_g is a bisector line of the maximal free angle toward the goal from point $\mathbf{G}$.

Figures 7 and 8 are examples of the deployment of a new robot. Fig. 7 is obtained from Fig. 3 and Fig. 8 is obtained from Fig. 5. Figure 9 is the deployment of the three defense robots under the RoboDragons' existing strategy. Table 3 shows the number of points in the unsafety region.

From Table 3, it is shown that the proposed defense strategy (deployment algorithm) greatly reduces the area of the unsafety region. This means that it is possible to keep the goal to the great extent and shows that the proposed strategy is efficient against the opponent's cooperative play.

However, the proposed defense strategy is not always efficient against the opponent's cooperative play. For example, applying the proposed strategy to Figure 10, we get Figure 11.

In Fig. 11, if the shooting robot is in the unsafety region which is shown by the circle in Fig. 11, the goal is achieved by the robot. This situation is not preferable. In the next section, we discuss how we avoid such situation.

4.2 A defense strategy considering the position of the opponent robot

In a real game, we should take the position of opponent robot into account. In this section, we discuss

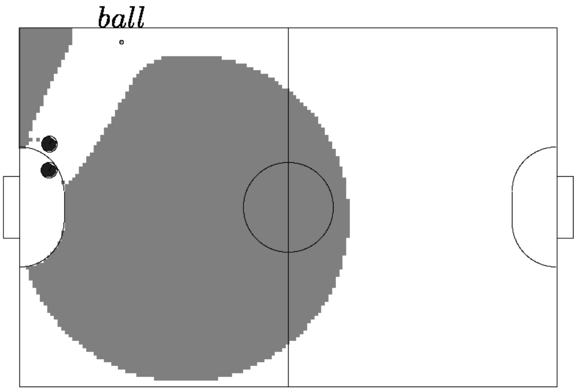

Figure 5: Safety region: two defense robots

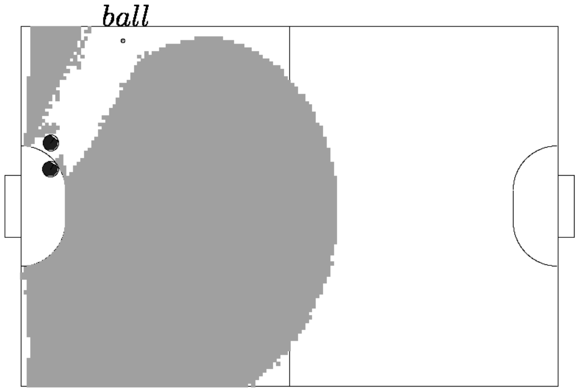

Figure 6: Safety region: simulation result corresponding to Fig. 5

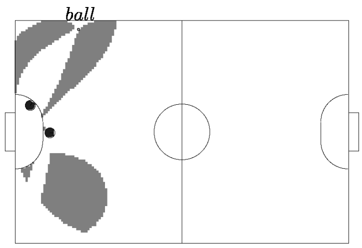

Figure 7: Safety Region: Placing a new defense robot to figure 3

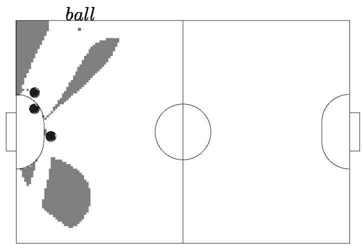

Figure 8: Safety Region: Placing a new defense robot to figure 5

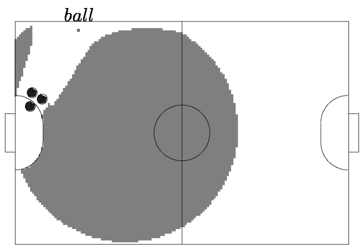

Figure 9: Safety Region: Three defense robots (existing strategy)

the weighted unsafety region. The weight is given to each point $\mathbf{e}$ in the unsafety region as a function of the distance between the position of the opponent robot $\mathbf{r}_e$ in the unsafety region and the point $\mathbf{e}$. We gave the following weighting function,

$$w(\mathbf{e}) = \max(1, 100 \times (1 - \frac{||\mathbf{r}_e - \mathbf{e}||}{\max(M, t_p \times v_r)})). \quad (8)$$

In eq. (8), v_r is a velocity of opponent robot at $\mathbf{r}_e$, and t_p is given by eq. (4). M is a threshold value to keep the weighting area wide when the value $t_p \times v_r$ is small. We used $M = 270$ in the experiment. $w(\mathbf{e})$ takes the value in the range between 1 and 100.

Computing the weighted gravity center $\mathbf{G}'$ of the unsafety region, and replacing $\mathbf{G}$ in the previous algorithm into $\mathbf{G}'$, we get an improved algorithm for computing the defense position.

Figure 12 shows a weighted unsafety region. The higher the weight, the darker the region. The weighted gravity center comes near the opponent robot in the unsafety region. Figure 13 shows obtained defense positions of the robots and an improved safety region. From the Fig. 13, it is clear that the opponent robot is in the outside of the unsafety region.

The computation time of the weighted gravity center and weighted (un)safety region[3] excluding

[3] When stating computation time, we use the term "weighted (un)safety region", since it includes the computation time of both safety and weighted unsafety region.

Table 3: the number of points in the unsafety region

	two defense robots	three defense robots
existing strategy	6256 (Fig. 5)	7086 (Fig. 9)
proposed strategy	1627 (Fig. 7)	1126 (Fig. 8)

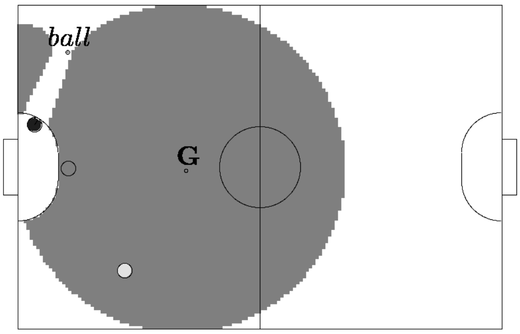

Figure 10: Safety region: one defense robot

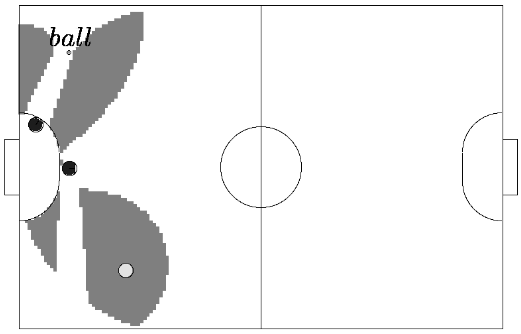

Figure 11: Safety region: two defense robots

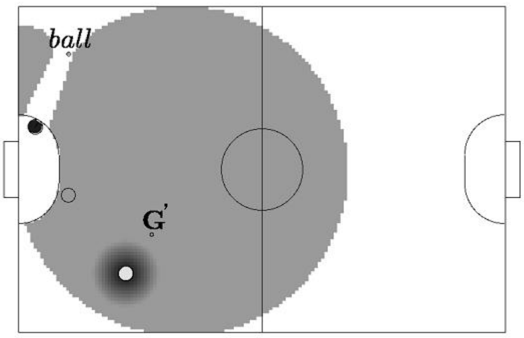

Figure 12: Weighted unsafety region

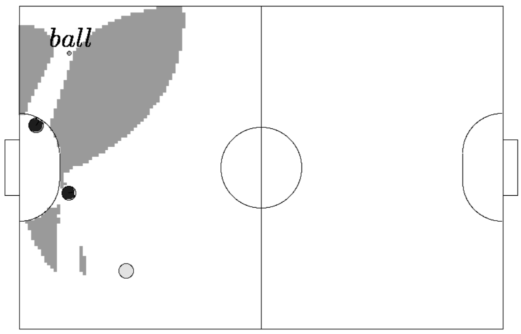

Figure 13: Improved safety region

Table 4: Computation time of weighted (un)safety region

CPU	Memory	Weighted gravity center	Weighted (un)safety region
Pentium4 2.8 GHz	1 GB	1.0 msec	5.6 msec
Athron64 X2 4200+	512 MB	1.9 msec	6.4 msec
Xeon 3.3 GHz	2 GB	0.3 msec	1.9 msec

the computation of weighted gravity center are given in Table 4. Comparing Tables 2 with 4, it is shown the increase of computation time by the weighted (un)safety region is small and the real time computation is possible.

5 Concluding remarks

We proposed the safety region as an index for evaluating the situation of the game in the SSL. The safety region is a region that the defense robots kepp the goal when an opponent robot shoots. The computation time of the safety region is around 2 msec on the typical PC. Since we can give 5 msec for the safety region computation, it is possible to use the safety region for determining the defense robots' positions in real time. We also showed the weighted unsafety region. It improves the defense ability of the team in the meaning of avoiding the chance of making the goal by the opponent robot. Experimental results support them.

Remaining problem is to improve the accuracy of the approximate safety region. It is caused by the impreciseness of the computation model of the approximate safety region. We will explore the more precise model.

acknowledgement

This work was supported by the cheif director's special study fund of Aichi Prefectural University and the president's special study fund of Aichi Prefectural University.

References

[1] D. Ball and G. Wyeth "UQ RoboRoos 2004: Getting Smarter", in RoboCup 2004 Symposium CDROM, SSL Team Description Papers, 2004

[2] M. Bowling, B. Browning, A. Chang and M. Veloso "Plays as Team Plans for Coordination and Adaptation", in RoboCup 2003: Robot Soccer World Cup VII, LNAI 3020, pp. 686 - 693, Springer, 2004

[3] R. Nakanishi, J. Bruce, K. Murakami, T. Naruse, and M. Veloso "Cooperative 3-robot passing and shooting in the RoboCup Small Size League", RoboCup 2006: Robot Soccer World Cup X, LNCS 4434 pp.418-425

[4] T. Laue, A. Burchardt, S. Fritsch, S. Hinz, K. Huhn, T. Kirilov, A. Martens, M. Miezal, U. Nehmiz, M. Schwarting and A. Seekircher "B-Smart (Bremen Small Multi Agent Robot Team) Extended Team Description", http://small-size.informatik.uni-bremen.de/tdp/etdp2009/small_b-smart.pdf

[5] S. Zickler, J. Bruce, J. Biswas, M. Licitra and M. Veloso "CMDragons 2009 Extended Team Description", http://small-size.informatik.uni-bremen.de/tdp/etdp2009/small_cmdragons.pdf

[6] H. Achiwa, J. Maeno, J. Tamaki, S. Suzuki, T. Moribayasi, K. Murakami and T. Naruse "RoboDragons 2009 Extended Team Description", http://small-size.informatik.uni-bremen.de/tdp/etdp2009/small_robodragons.pdf