# Cooperative Strategy using Dynamic Role Assignment and Potential Fields Path Planning

Paul A. Vallejos, *Member, IEEE*, Javier Ruiz-del-Solar, *Senior Member, IEEE*, Alan Duvost

**Abstract—This paper describes a cooperative strategy for a robot soccer team. The proposed strategy uses dynamic role assignment, which allows the team to have a flexible and robust strategy. Each role is defined as a state-machine, where each state triggers different behaviors. Transitions between states are triggered by internal and external robot conditions. The positions of the robots in the field are dynamically controlled using potential fields, which makes possible a better field coverage.**

**Index Terms—Cooperative Strategy, robot soccer, potential fields, multi-agents.**

## I. INTRODUCTION

Mobile robotics is a lively and expanding research field, whose main goal is the development of autonomous robots that can freely interact in dynamic environments. There are several aspects to be solved, including low-level issues like motion control, vision analysis, self-location and mobile objects tracking as well as high-level challenges, like the employed strategy. This paper presents a cooperative strategy for a multi-agent task, applied to robot soccer.

In this work, we focus on the robot soccer application, specifically on the RoboCup four-legged league challenge, where the used platform must be Sony AIBO robot. This application has some special characteristics: (i)AIBO robots have four, 3DOF legs, (ii)the main sensor is a low-resolution color camera with 3DOF, (iii)field lines and colored goals and beacons can be used for self-location, (iv)the ball is orange, (v)a wireless data network (802.11b) makes possible the communication between robots and with the game controller (a computer referee), (vi)no external sensing or processing is allowed, (vii)AIBO limited processing power imposes constraints to the algorithms that can be implemented (more information about this league in [1]), and (viii)teams are composed by four AIBO robots: one goalkeeper and three field players.

Our robot soccer library Uchile1 [2] (see block diagram in Figure 1) is designed to solve the four-legged soccer problem. Here we present a cooperative strategy using dynamic role assignment and potential fields implemented on the *High-level Strategy* module. This module communicates with: (i)*Low-level Strategy*, where all low-level and complex behaviors are implemented, (ii)*Localization*, where the self-location and the tracking of the ball are performed, (iii)*Vision*, where the image analysis and object recognition are performed, and (iv)*Motion Control*, where all robot movements are generated.

*Localization* module shares information between all robots over the wireless network. The information shared by this module corresponds to the estimated self-location, the estimated ball-location and their corresponding covariance matrices. In the same way, the proposed *High-Level Strategy* shares the robot inner-state and its current intentions through the wireless network, allowing by this way a cooperative approach.

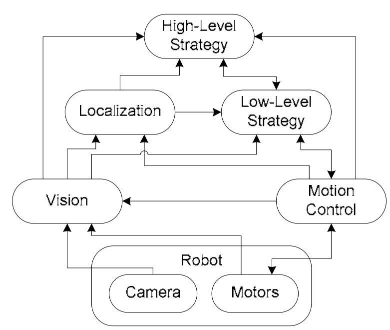


Figure 1. Uchile1 implemented software architecture for soccer playing robots.

## II. RELATED WORK

In robot soccer, flexible and robust strategy design is fundamental. Several works have been done for developing new strategy approaches. The most used strategies are fuzzy logic systems, decision trees and behavior-based strategies.

Most of the RoboCup four-legged league participating teams have chosen behavior-based strategies with different roles for the players; typically, four-legged league teams define four roles: *Attacker*, *Support Attacker*, *Defender* and *goalkeeper*. Some teams use fixed role assignment, like in [3], but most of them use dynamic role assignment. For path planning there are many methodologies implemented, like cost-objective evaluation [4], occupancy grids [5],

Department of Electrical Engineering, Universidad de Chile. E-Mail: {pavallej,jruizd}@ing.uchile.cl.
This research was funded by the research project CONICYT/BMBF 2003 – 4 – 146.

variations of occupancies trees [6], plan representation and execution through Petri Nets [7], decisions based on fuzzy logic [8], decision trees [9] and potential fields as [10] and [11].

In the technical description papers published in the RoboCup2004 proceedings [12], 37.5% of the teams mention their role management, 66.6% of them use dynamic role assignment, and only 33.3% use static roles. Also 75% of the league teams have published their path planning methodology, of them, 50% use potential fields, and the other 50% use another techniques, all based on occupancy grids. The here-proposed strategy uses a dynamic role assignment and potential fields path planning combination.

## III. PROPOSED STRATEGY

### A. *Overview*

The proposed strategy, as in human football, is based on communication and cooperation between players with different roles. Each robot takes decisions based on its perceptions and the information it receives from teammates. In addition, each robot communicates its internal strategy state to the teammates. In this way, every robot of the team knows the current strategy state of all its partners.

A role is defined as a state machine with transitions triggered by a combination of sensorial information, internal strategy state, and messages received from other robots. The role assignment is dynamic and the necessary conditions for a role transition are of the same kind of those for state transitions. States have a set of conditions defining the behavior to be executed. The behavior output is an action executed by the *Motion Control* module; typical actions are walk to specific locations with some orientation, and different kinds of kicks.

The strategy philosophy is common to all the field players, and the goalie is the only one with a special strategy.

#### 1) *General philosophy of the goalie's strategy:*

The main task of the goalie is to protect the goal, that is, it must position itself between the ball and the goal. Two states define this behavior, which are executed slightly different depending on whether the goalie sees the ball or not. Obviously, this task can only be performed when at least one player knows the ball position. When none of the teammates knows where the ball is, the goalie stays in the middle of the goal and looks for it. If the ball comes close, and this action seems appropriate, the goalie goes to get the ball and kicks it away. After any attempt to kick the ball, the goalie, as all players do, looks towards the kick target place.

#### 2) *General philosophy of the field players' strategy*

The field players are focused on going to the ball, one player at the time. The closest robot to the ball (the distance measurement takes into account not only the absolute distance but also the angle, in order to favor players that are already looking towards the ball), goes to it and attempts to kick it towards the opposing goal. The other players take up supporting positions. After any player kicks the ball, everyone looks towards the kick target place. At any time during the game, except while attempting to kick the ball, the players attempt to locate themselves if their position uncertainty becomes too high.

### B. *Potential Fields*

A known methodology for path planning is the potential fields approach. This methodology is based on creating virtual potential fields, attractive for desired objectives and repulsive for obstacles or undesired positions. In this way, the virtual force applied over the robot is calculated as the derivate of the potential field. The proposed methodology uses a force focus, this is, given a set of objects, generating repulsive and attractive potential fields, a punctual force can be calculated just on the interest point. When a decision of path must be taken, the force on the initial point is calculated, the calculated force vector direction is used to give a step, when that step is complete, another force for that point is calculated and so on. This is a fast method to choose a path. The equation (1) shows the calculated virtual force $VF$ in a point $x$ for an object in the position $o_i$ with a factor $W_i$. If the factor $W_i$ is positive, then the virtual force is repulsive and if the factor is negative, then the virtual force is attractive.

$$VF(x) = \frac{W_i \cdot (x - o_i)}{|x - o_i|^2} \quad (1)$$

Virtual forces of lines and rectangular areas are calculated as the 1D and 2D integrals of the equation (1) respectively.

In Figure 2 is shown an example of potential field. In this case, two points are used, one attractive and the other repulsive, as well as one repulsive line. The chosen path from any point will follow the corresponding arrows (each arrow is a force evaluation).

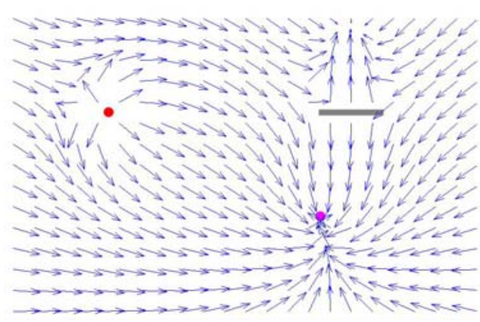

Figure 2. Virtual forces generated by an attractive point (violet), a repulsive point (red) and a repulsive line (gray).

### C. *Behaviors*

Behaviors are aimed to select a given action to be executed by the robot. In our implementation, there are three kinds of behaviors: low-level behaviors, complex behaviors and high-level behaviors.

### 1) Low-Level Behaviors

Low-level behaviors are the simplest behaviors, they do not use sensorial information and they have a direct connection with the action module. Implemented low-level behaviors are: (i)**Stand Still**. This behavior just keeps the robot in a stable position with its four legs on the floor. (ii)**Turn Continuously**. This behavior makes the robot turn continuously clockwise facing the head with the movement; this behavior does not involve any displacement. (iii)**Walk**. This behavior generates a walk with a given displacement. (iv)**Kick**. This behavior selects the best kick to approach the ball to a given objective. (v)**Switch Motion State**. This behavior switches between the normal actions and the actions executed when dribbling the ball. The dribbling the ball state main function is to provide movement, even displacement or kicks, while the ball is held with legs and head. Obviously, the held ball constriction makes the dribbling ball state to have a lower motion performance than normal state. (vi)**Move the Head**. This behavior just move the head to given angles. (vii)**Look an Objective**. This behavior calculates the necessary head joints angles to look an objective on a given relative position and move the head to them. (viii)**Ellipse**. This behavior moves the head on an elliptical way. (ix)**Modified Ellipse**. This Behavior moves the head following the trajectory resulting from the merge of two ellipses: one vertical and the other horizontal, generating a movement that in the higher zone is like an ellipse, but in the lower part it looks to the ground.

### 2) Complex Behaviors

Complex behaviors are basically decision trees, which, considering the inner robot state, sensorial information and the information received from teammates, choose the low-level behavior needed to accomplish the desired complex behavior. The complex behaviors implemented are: (i)**Location Help**. This behavior measures the confidence value of the self-location module and chooses between two low-level behaviors: "Look an Objective", in which the objective is one of the landmarks present on the field, and "Ellipse" to search landmarks when the confidence value is too low. (ii)**Go to an Objective**. This behavior chooses a path to achieve a specified objective and chooses the appropriated walking parameters (step length and walking speed) in each moment. This behavior can use potential fields in order to choose the path. Potential fields used by this behavior are flexible and may contain a list of any finite length of objects that produce potential fields; currently the allowed objects are points, rectangles and lines in any direction. This behavior can trigger the low-level behaviors "Stand Still" and "Walk". (iii)**Ball Tracking**. This behavior chooses the appropriated low-level behaviors to look at the estimated ball position. It can choose between "Look an Objective", "Move the Head" and "Stand Still" low-level behaviors. (iv)**Go to the Ball and Kick it to an Objective**. This behavior is on the limit of being a high-level behavior because of its complexity. Its objective is to select the corresponding low-level and complex behavior to tracking the ball, walk to the ball and once the ball is accessible, kick it to a specified objective, typically the rival goal. This behavior can trigger the "Ellipse", "Modified Ellipse", "Kick", "Switch Motion State", "Look an Objective" and "Stand Still" low-level behaviors and the "Go to an Objective" and "Ball Tracking" complex behaviors.

### 3) High-Level Behaviors

High-level behaviors are directly related to a specific state. Thus, each robot state has a corresponding high-level behavior. High-level behaviors are decision trees used for choosing the best complex or low-level behavior to be executed. The implemented high-level behaviors are: (i)**High-Level Go to Ball and Kick**. This behavior always use the complex behavior "Go to the Ball and Kick it to an Objective", changing its objective between the rival goal, a teammate (for passing) and the furthermost point from the own goal (for defending the own goal). (ii)**Look Kick Target**. This behavior always use the low-level behavior "Look an Objective", setting the objective as the same place as the objective of the last kick of any teammate. (iii)**Position in the Field**. This behavior uses the potential fields to place the robot on the field. The potential fields used are repulsive for the teammates, attractive for the ball, attractive for a positioning object (defensive line for defender player, player balancer for supporting player and ball-goal alignment for attackers) and repulsive for the field border. This behavior can trigger the complex behaviors "Go to an Objective" and "Ball Tracking". (iv)**Goalie Position**. This behavior uses potential fields to place the robot in the goal. Potential fields used include a strong attractive point at the goal and several repulsive lines corresponding to the field border. This behavior can trigger the complex behaviors "Go to an Objective" and "Ball Tracking". (v)**Field Search Ball**. This behavior performs a progressive search for the ball, first it looks around the place turning 360 degrees, then it switch between two given points of the field, at each of them it looks around the place turning 360 degrees. This behavior can trigger the complex behavior "Go to an Objective" and the low-level behaviors "Turn Continuously", "Modified Ellipse" and "Stand Still". (vi)**Goalie Search Ball**. This behavior just positions the robot in the goal and search for the ball using only the head. This behavior can trigger the complex behavior "Go to an Objective" and the low-level behavior "Modified Ellipse".

## D. Roles

The possible roles of the players are Attacker, Supporting Attacker, Defender, and Goalie. Only the goalie has a fixed role. The others players may change their roles depending on their position on the field mainly. Employed rules for changing roles are the following:

- The defender cannot become directly an attacker, or the reverse; they must first take up the intermediate position of being a supporting attacker.
- The defender becomes a supporting attacker if it is no longer the last player, i.e. the nearest to his own goal.
- The supporting attacker becomes a defender if it finds itself located nearest to its own goal.

- The supporting attacker becomes an attacker if it goes for the ball while being the nearest to the opposing team's goal.
- The attacker becomes a supporting attacker if another player is going for the ball, and that other player took the position of an attacker.

This system should not allow two players to take up the same position (for any lapse of time), but if they would, it would not affect the decisions regarding state-changing, as the three field players share the same state-system. In the Figure 3 is shown the role transition diagram.

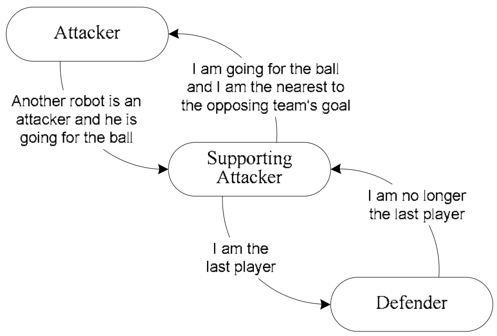


Figure 3. Role transitions allowed for the field players.

### *E. States*

Each role is a state-machine with several states and transitions, described in the following paragraph.

#### *1) Goalie Strategy*

The goalie strategy is composed by five states (see Figure 4): (i)**Searching the ball**. The goalie walks to the middle of the goal, and searches for the ball moving its head. Once the goalie has reached the middle of the goal, it stops moving the body and only moves the head. This state executes the high-level behavior "Goalie Search Ball". (ii)**Defending goal, ball seen**. The goalie walks, placing itself on the line between the goal and the ball, 70cm from the goal. This state executes the high-level behavior "Goalie Position". (iii)**Defending goal, ball not seen**. The goalie places itself along that same line, but nearer the goal, covering the case when the ball position estimation is inaccurate. This state executes the high-level behavior "Goalie Position". (iv)**Going to ball**. The goalie goes to the ball, using its own visual information and the information received from its teammates. When it gets close enough to the ball, it uses its own visual information only, and tries to hit the ball. It always tries to reach the ball from the closest side to its goal, in order to avoid blunders leading to home goals. When kicking the ball, the goalie aims directly ahead of where the ball is. This state executes the high-level behavior "High-Level Go to Ball and Kick". (v)**Looking to objective**. If the goalie kicks the ball, it looks in the direction where the ball should be going, and then, after a short time (about two seconds), it goes to the search ball state. This state executes the high-level behavior "Look Kick Target".

The transitions between the states of the goalie are specified by the next conditions: (i)**Ball lost**. The ball is "lost" when none of the team players knows where the ball is. (ii)**Ball found**. This means a player (any except the player itself) sees the ball, which position is received through the communication network. (iii)**Ball kicked**. This condition is true immediately after the goalie has kicked the ball. (iv)**Ball seen**. In this case, it means the goalie sees the ball. (v)**After short time**. There is an automatic state change after about 1 second. (vi)**Ball reachable**. This condition is a complex one. The goalie has a lined area around the goal where no defender can enter. If the goalie sees the ball, and estimates its position within the area or just outside (within 10cm), then the condition is true and the goalie goes for the ball. We also consider that the goalie can leave its area, but only in special circumstances, and not too far away. It will consider that the ball is reachable and it will go for it if it is at least 50cm closer to the ball than any other player, and if the ball is within 130cm from the goal. This is the second way that the condition we mark here as "ball reachable" becomes true. (vii)**Ball unreachable**. This actually means that the previous condition becomes false. The condition, however, cannot interrupt the goalie if it is attempting to hit the ball.

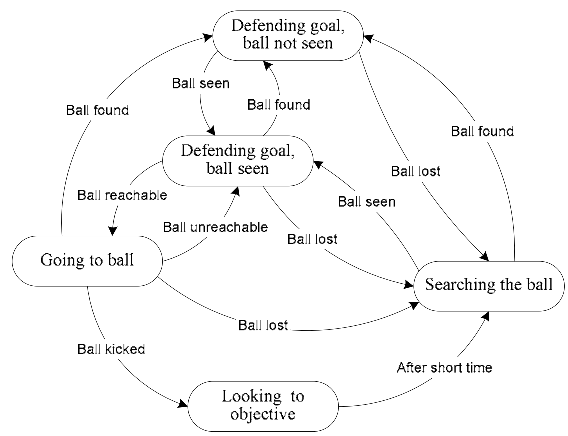


Figure 4. Transition-state map for the goalie.

#### *2) Field players strategy.*

The Field players strategy is composed by four states (see Figure 5): (i)**Searching ball.** The players look around to see if they can find the ball, and then rotate, looking all around, and if they still do not find the ball, they will go to search in two predefined locations on the field. First, they walk to the first of those two positions, and then they go to the other, and move back and forth until they find the ball. Of course, with all four players searching, the search rarely lasts long enough for the players to use those predefined positions. This state executes the high-level behavior "Field Search Ball". (ii)**Positioning.** This state is the one for playing without the ball. The idea is to spread the team on the field, and for this purpose, virtual potential fields are used. They repel the players from the edges of the field, from the defense zone, and from the other players. In Figure 6 is shown the virtual forces present on the attacker while positioning. An additional virtual potential field attracts the defender towards the line between the goal and the ball. This state executes the high-level behavior "Position in the Field". (iii)**Going to ball.** Same as for the goalkeeper, except that the kicks are always aimed at the opposite goal. This state uses potential fields to choose the path for approaching to the ball with an appropriated alignment. In Figure 7 is shown the virtual forces present while going to ball. This state executes the high-level behavior "High Level Go to Ball and Kick". (iv)**Looking to objective.** If any robot kicks the ball, then the robot looks at the direction where the ball should be going, and then after a short time (about two seconds) goes to the search ball state. This state executes the high-level behavior "Look Kick Target".

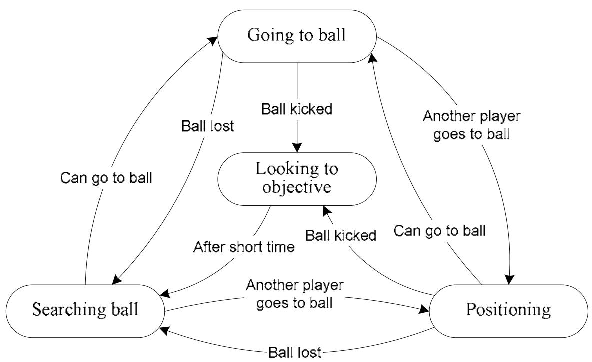


Figure 5. Transition-state map for a field player.

The field players transitions are equal to those of the goalie, except the conditions "Can go to the ball", and "ball kicked". The condition "ball kicked", in this case, includes kicks by other players. The condition "Can go to the ball" almost only depends on who is the closest robot to the ball. If the player estimates that its position is at least 50cm closer to the ball than the next player position, then it goes. If it is the closest, but only slightly, he checks that nobody else is going to the ball in order to go. This does not include the goalie, meaning that another player will go to the ball at the same time as the goalie.
The condition of "another player goes to ball" is actually the opposite of the previous. Another player is closer to the ball and going to play it.

The transitions between the field players states are determined by the next conditions: (i)**Ball lost**. The ball is "lost" when none of the team players knows where the ball is. (ii)**Ball kicked**. This condition is true immediately after any player has kicked the ball. (iii)**After short time**. There is an automatic state change after about 1 second. (iv)**Can go to ball**. If the player estimates that its position is at least 50cm closer to the ball than the next player position, then it goes. If it is the closest robot to the ball, but only slightly, it checks that no other field player is going to the ball in order to go. (v)**Another player goes to ball**. This actually means that the previous condition becomes false. The condition, however, cannot interrupt the player if it is attempting to hit the ball.

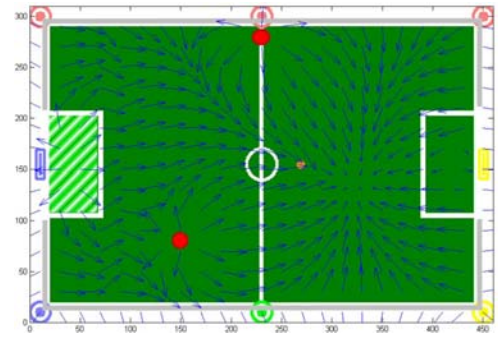

Figure 6. Virtual forces calculated for the attacker while it is positioning. Gray lines represent repulsive lines for repelling from the field edges, red circles represent two teammates generating repulsive fields, an orange circle represents the ball generating an attractive field and a light green square represents the defending area generating a repulsive field.

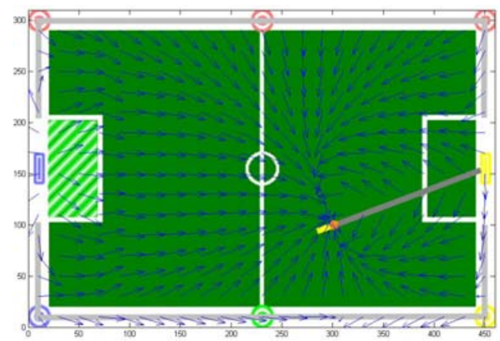

Figure 7. Virtual forces calculated while the player is going to ball. An orange circle represents the ball generating an attractive field, gray lines represent repulsive lines (six for repelling from the field edges and one for repelling the player from the line between the ball and the rival goal). A light green square represents the defending area generating a repulsive field for preventing the player to enter the forbidden area and a yellow line generating an attractive field to align the player on their approach to the ball.

## IV. Experiments and Results

It is required to evaluate three aspects of the proposed strategy: (i)the behavior of the robots when they use the potential fields for placing themselves on the field or for going to the ball, (ii)the dynamic role assignment and (iii)the state transitions. To evaluate the potential fields used for placing robots on the field, there were placed two standing still robots on the field and a third one was placed on several positions on the field. The position reached due to the potential fields was evaluated qualitatively from the soccer point of view, considering as a good position one where the system compound by the three robots maximize the field coverage given the positions of the standing still robots. Five different sets of positions were tested for the standing still robots, with ten different positions for the moving robot each one. All positions were chosen as the positions that typically adopt the robots on the field during a game. Only two of the tests gave undesirable stable positions, the others 48 times were evaluated as good positions.

Next experiment was the same as the previous test, but this time all three robots were placing themselves using potential fields, this time only ten trials were made. Five of them resulted in good robot positioning, maintaining dynamic changes of their position. The robots never reach a static equilibrium position, due probably to errors on the self-localization module. The other five trials delivered two robots on desirable static equilibrium positions, but the third robot was pushed to reach their static equilibrium on the field edge, which is a highly undesirable position.

Ten trials were made to test the ball approaching application of the potential fields. In this case, one robot and one ball were used, both placed arbitrary on the field, the results was a successfully approach to the ball improving the angle of the robot with respect to the rival goal, but the approaching time was seriously compromised, almost duplicating it.

The dynamic role assignment and the state transitions were tested together. In this case, there were made five trials, each of them consisting on ten minutes of game between one team formed only by a goalie and the other one compound by three field players. The evaluation was again qualitative from the soccer point of view. This time the role assignment was almost perfect, the only problem appeared on the states transitions. It was some indecision between the *Attacker* and the *Supporting Attacker* when both were near the ball, and both players were switching a few seconds the *going to ball* state, until one assumed definitely that state and the roles were reassigned giving the *Attacker* role to the one going to ball and the *Supporting Attacker* role to the other.

One problem observed during all the experiments was the imprecision of our *localization* module. This problem sometimes made the potential fields calculation to fail and it generated some oscillations around the equilibrium places. Probably this is the cause of the player pushed to the field edge.

The proposed strategy was implemented before the participation of the UChile1 team in the RoboCup2004. During the competitions, there were serious problems with the communications over the wireless data network, which affected not only our team but also others teams. In the particular case of our proposed strategy, the result of the communication problem was catastrophic. The *Positioning* state has no sense without communication, since all robots goes to the same equilibrium point (near the center), giving place to the exactly opposite behavior to the desired one (the desired behavior is to cover the maximum area of the field), the ball get lost easily and all the behavior is undesirable. All these problems make the proposed strategy unfeasible under those circumstances, and show the high communication dependency of the proposed strategy. All RoboCup2004 games were played using our old strategy implementation without potential fields and with static role assignment.

## V. Conclusions

The potential fields' technique is very useful for placing the robots on the field, avoiding obstacles and aligning to the ball before going, but this technique must be calibrated very well to minimize the effect on the required time to develop the actions, since time is a very important issue on soccer application. Our potential fields approach shows itself as a good and simple solution to the robot-positioning problem, but it is also clear that more tuning and experimentation is required to exploit the full capacity of this methodology.

Dynamic role assignment is highly needed in the robot soccer application and it showed very good preliminary results. However, it must be combined with other techniques as fuzzy logic and some learning techniques as in [13].

Due to the presence of network problems on the current application (robot soccer on the four-legged league of RoboCup), the communication dependency of the proposed strategy should be corrected, so tactical changes should be implemented to allow the robots objective accomplishment even if the communication is poor or absent.

## VI. References


[1] RoboCup Four-Legged League Official Web Site: http://www.openr.org/robocup/index.html.

[2] Ruiz-del-Solar, P. Vallejos, J. Zagal, R. Lastra, G. Castro, C. Gortaris, I. Sarmiento, "UChile1 2004 Team Description Paper", *Proc. of the RoboCup 2004 Symposium*, July 4 – 5, Lisbon, Portugal.

[3] A. Shinohara, S. Takano, S. Miyamoto, M. Toyomasu, "JollyPochie in the Four Legged Robot League", *Proc. of the RoboCup 2004 Symposium*, July 4 – 5, Lisbon, Portugal.

[4] H. Najima, Y. Yasutake, T. Ishimura, K. Oda, T. Ohashi, "ASURA: The Kyushu United Team 2004 in the Four Legged Robot League", *Proc. of the RoboCup 2004 Symposium*, July 4 – 5, Lisbon, Portugal.

[5] N. Mitsunaga, M. Hirota, H. Fukumoto, T. Nishi, H. Toichi, M. Asada, T. Toriu, "BabyTigers 2004: Osaka Legged Robot Team", *Proc. of the RoboCup 2004 Symposium*, July 4 – 5, Lisbon, Portugal.

[6] M. Veloso, S. Chernova, D. Vail, S. Lenser, C. McMillen, J. Bruce, F. Tamburrino, J. Fasola, M. Carson, A. Trevor , "CMPack'04: Robust Modeling through Vision and Learning", *Proc. of the RoboCup 2004 Symposium*, July 4 – 5, Lisbon, Portugal.

[7] L. Iocchi, D. Nardi, F. Patrizi, "S.P.Q.R. Legged Team", *Proc. of the RoboCup 2004 Symposium*, July 4 – 5, Lisbon, Portugal.

[8] K. LeBlanc, S. Johansson, J. Malec, H. Martínez, and A. Saffiotti, "Team Chaos 2004", *Proc. of the RoboCup 2004 Symposium*, July 4 – 5, Lisbon, Portugal.

[9] C. Sammut, W. Uther, B. Hengst, "rUNSWift 2003", *Proc. of the RoboCup 2004 Symposium*, July 4 – 5, Lisbon, Portugal.

[10] D. D. Lee, D. Cohen, P. Srivastava, Y. Ooi, L. Saul, C. J. Taylor and V. Kumar, "University of Pennsylvania's RoboCup Legged Soccer Team (the UPennalizers)", *Proc. of the RoboCup 2004 Symposium*, July 4 – 5, Lisbon, Portugal.

[11] T. Röfer, R. Brunn, I. Dahm, M. Hebbel, J. Hoffmann, M. Jüngel, T. Laue, M. Lötzsch, W. Nistico, M. Spranger , "GermanTeam 2004: The German National RoboCup Team", *Proc. of the RoboCup 2004 Symposium*, July 4 – 5, Lisbon, Portugal.

[12] D. Nardi, M. Riedmiller, C. Sammut, J. Santos Victor (Eds.), Proc. of the RoboCup 2004 Symposium, July 4 – 5, Lisbon, Portugal.

[13] K. Hwang, S. Tan, C. Chen, "Cooperative Strategy Based on Adaptive Q-Learning for Robot Soccer Systems", IEEE Transaction on Fuzzy Systems, Vol. 12, No 4, August 2004.