Advanced Robotic Systems Scientific Book 2005

URL: http:\\www.ars-journal.com
E-mail: publication@ars-journal.com
Vienna University of Technology
Automation and Control Institute
Gusshausstrasse 27-29, A-1040 Wien,
Austria, EU

INSTRUCTIONS FOR AUTHORS

PUBLISHING PROJECT:
ADVANCED ROBOTIC SYSTEMS SCIENTIFIC BOOK 2005

1. Introduction

¶

In recent robotic studies, in many key areas decoupled multi-agent systems have become more popular than complex single agent systems, where the former is more robust, fast and cheap to implement. The most important reason behind this preference is to eliminate the possibility of single point of failure, which is a vital concern for single complex agents. Usage of decoupled multi-agent systems may also reduce the total cost of the entire system when it is possible to use a team of single cheap robots for performing complex tasks, instead of building up a single complex and expensive robot to satisfy all the system needs. As a result, typically when a team of robots is used, the system throughput increases while the total cost decreases. Since the robots usually have simpler physical structures, generally less complicated controller programs are necessary to manipulate the agents. The decoupled behaviors of agents can cause communication and coordination problems, however. The studies in (Dudek *et al*, 1996; Cao *et al*, 1997; Arkin and Balch, 1998; Svestka and Overmars, 1998; Stone and Veloso, 2000; Song and Kumar, 2002), refer to many different approaches to the usage of multi-agent teams in key application areas.

Before dealing with coordination problem among agents in a multi-agent system, the individual capabilities and limits of the agents should be determined. This task is usually trivial for homogeneous teams. However, in heterogeneous systems, the decoupled system should provide a feasible utilization for each agent. The most important individual action for a simple robot in a homogeneous system is usually related with motion because the path planning or trajectory planning routines depend on the capabilities of the actuators of the robot. This task becomes a challenging one even for the omnidrive robots without non-holonomic constraints.

The key problem in coordinating the team of robots is to decompose the complex task in to several simple low-level actions, and assignment of these actions among the team in an optimum way such that the tasks should be formed by combining low-level actions of the robots, while avoiding collisions and allowing all the low-level actions to be implemented synchronously, and successfully. There are numerous theoretical and practical studies about decomposing a goal into subtasks, which can then be easily performed by robots with their basic actions. However, in multi-agent systems, assigning the tasks to the agents is not an easy task because the complexity of the problem increases exponentially with the number of agents and the dimensionality of the configuration space.

In robot soccer, which is a challenging test bed for multi-agent systems, two teams of robots compete with each other to win the match. For the benefit of the team, the robots should work collaboratively, whenever possible. Designing a team, which can beat every opponent available, is certainly a hard mission. The market-driven approach applies the basic properties of free market economy to a team of robots for increasing the profit of the team as much as possible. It enables implementation of high level skills by using the “team spirit” of a group of simple robots, which is quite challenging and hard in case of classical planning and task allocation mechanisms, and while avoiding the collisions and enabling collaboration, allows gathering maximum profit from the implemented tasks.

Recently the market-driven approach was introduced as an alternative method for robot coordination in (Dias and Stenz, 2001). It is highly robust and avoids the single point failure problem, while increasing the team performance considerably. There are several applications of market-driven approach. The work in (Zlot *et al*, 2002) introduces the approach to multi-robot exploration. In (Gerkey and Mataric, 2002) a work on auction based multi-robot coordination is presented. These implementations seem to work well but are limited due to the static nature of the environment. Domains like agricultural areas are simple, static and do not require fast task allocation, planning and coordination as in robot soccer (Kose *et al*, 2004).

In order to provide a satisfactory solution to the task assignment and collaboration problem in robot soccer, several approaches have been implemented including static assignment (Kaplan, 2003), market based assignment (Kose *et al*, 2003) and reinforcement learning based extension to market based approach (Kose *et al*, 2004; Tatlidede *et al*, 2005). In this chapter, these approaches are compared and studied in detail.

In the next section, the robot soccer domain and in section 3, market-driven methodology is introduced. In the section 4, the previously developed approaches proposed in this work are described briefly. The results of the tests for analysis and comparison of the approaches are given in Section 5. In the section 6 there is a brief conclusion related to these approaches.

2. Robot Soccer Domain

Robot soccer domain is a well-defined environment for developing multi-agent strategies. The initial world model, constraints and goals are known. The nature of the game enables the implementation of different levels of team coordination, besides allows the development of challenging complex behaviors from simple low level tasks implemented by simple agents. It is also possible to test a new strategy against the existing ones in the international robot soccer competitions (FIRA, 2003; ROBOCUP, 2003).

In this work, a modified version of *Teambots* simulator (Balch, 2000) is used to develop and train the proposed controllers (See Fig. 1). Although *Teambots* is not used in any international robot soccer competition, it is a well-known multi-purpose simulator. In addition, it is an open source Java project, and enables easier development of different kinds of learning strategies.

The modification in the simulator is the implementation of free-ball for deadlock situations. The ball is moved to the center of the quarter of the field in which a deadlock situation occurs. In *Teambots* each team has five omnidrive robots. Localization information is available for each robot. The robots can communicate with any other robot via broadcasting specific types of information messages. The relative position of the ball and the other robots are sensed by the robot with a Gaussian noise.

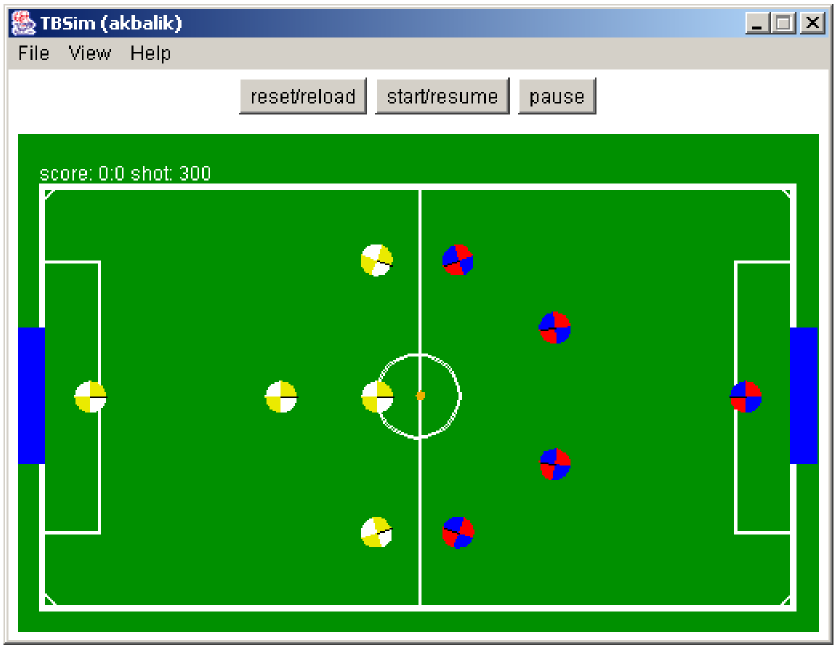


Fig. 1 ***Teambots*** **Simulator**

3. Market Methodology

The main goal in free-markets is the maximization of the overall system profit. If each participant in the market tries to maximize its profit, as a result of this, the overall profit for the system is expected to increase. The idea of the market-driven method for multi-robot teams is based on the interaction of the robots among themselves in a distributed fashion for trading work, power and information. In general, there is a main goal of the team (i.e., building the map of an unknown planet, harvesting an agricultural area, sweeping buried landmines in a particular area, etc.). Some entity outside of the team is assumed to offer a payoff for that goal. The main goal of the system is decomposed into smaller tasks and an auction is performed for each of these tasks. In each auction, the participant robots (who are able to communicate among themselves) calculate their estimated cost for accomplishing that task and offer a price to the auctioneer. At the end of the auction, the bidder with the lowest offered price will be given the right of execution of the task and receives its revenue on behalf of the auctioneer. There are many possible actions that can be taken. A robot may open another auction for selling a task that it won from another auction, two or more robots may cooperatively work and get a task which is hard to accomplish by a single robot, or for a heterogeneous system, robots with different sensors/actuators may cooperate by resource sharing (for example, a small robot with a camera may guide a large robot without a vision system for carrying a heavy load).

In order to implement the strategy, a cost function is defined for mapping a set of resources (required energy, required time, etc.) to a real number and the net profit is calculated by subtracting the estimated cost for accomplishing the task from the revenue of the task. For example, in Fig. 2, the estimated cost values are given for each task. The robots calculate their own cost values for each task. Although it seems cheaper to assign task B to robot B, when the overall profit of the team is considered, it is more profitable to assign both tasks to robot A.

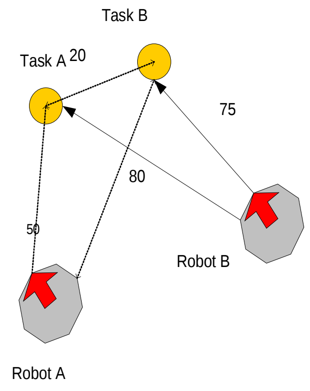


Fig. 2. Market-driven scenario

4. Role Assignment Approaches in the Robot Soccer Domain

Although the robot soccer domain is well defined, it is not a trivial task to manage a robot soccer team. The first challenge is designing the low-level actions. We use a potential fields based motion strategy. The potential fields are used both for local actions like obstacle avoidance and global actions like positioning near the ball (see Fig. 3). Each object on the field has a potential effect on the player. The weights of these fields are fine tuned by using genetic algorithms (Kaplan, 2003).

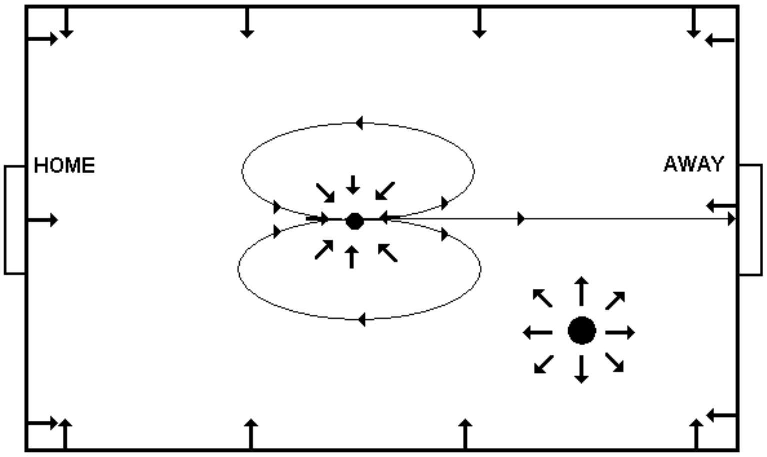


Fig. 3 Potential forces.

Although with the potential fields, specific tasks, like shooting or defending can be managed, this method is not adequate for team coordination. In other words, the roles can be performed by using potential fields; however, it is not feasible to assign roles only by using potential fields.

4.1 Static Role Assignment

The first remedy for the role assignment problem is the static role assignment, which is certainly not a good practice, since it causes system failure in case of a single agent failure. The static role assignment is used only for the goalie, since this is the most optimum choice for the team success, and as the role switching time for the goalie increases, the team performance decreases.

After assigning the goalie, there are four more agents left to manage. The roles should be assigned to the agents according to some metrics, like the distance between the agent and the ball. The "*RIYTeam*", which is the first team developed with dynamic role assignment, first chooses the agent closest to the ball as the attacker. Next, the agent, which is closest to its own goal among the unassigned agents, becomes the defender. Finally, the remaining two agents help the attacker by holding strategic positions behind the attacker (Kaplan, 2003). This method has also some drawbacks. The metrics we use for selecting the attacker or defender are rather primitive. In addition, the agent, which controls the ball, should have different options other than shooting. All these requirements introduce new metrics, which are quite hard to calculate and communicate in a multi-agent system.

4.2 Market-driven Approach

In order to address the problems mentioned in the previous subsection a new team, "*MarketTeam*" is developed where market-based strategy is used to simplify the problem by only communicating the cost values of each agent for every action instead of communicating all metrics. As a result, every robot calculates its own bid for each action according to the following equations and broadcasts only these values (Kose *et al*, 2003; Frias-Martinez *et al*, 2004).

$$C_{ES} = \mu_1 . t_{dist} + \mu_2 . t_{align} + \mu_3 . clear_{goal} \tag{0}$$

$$C_{bidder} = \mu_4 . t_{dist} + \mu_5 . t_{align} + \mu_6 . clear_{teammate(i)} + C_{ES(i)}, i \neq robotid \tag{0}$$

$$C_{auctionerr} = C_{ES(robotid)} \tag{0}$$

$$C_{defender} = \mu_7 . t_{dist} + \mu_8 . t_{align} + \mu_9 . clear_{defense} \tag{0}$$

where $robotid$ is the id of the robot, t_{dist} is the time required to move for specified distance, t_{align} is the time required to align for specified amount, μ_i are the weights of several parameters to emphasize their relative importance in the total cost function, $clear_{goal}$ is the clearance from the robot to goal area, $clear_{ball}$ is the clearance from the robot to ball, $clear_{defence}$ is the clearance from the robot to the middle point on the line between the own goal and the ball, and similarly $clear_{teammate(i)}$ is the clearance from the robot to the position of a teammate. Each robot should know its teammates score and defense costs. In our study each agent broadcasts its score and defense costs to its teammates.

This approach increases overall performance; however, there are still problems with the role assignment strategy. The first one is the restriction of one-to-one assignment. Previous strategies assign only one agent for each role simultaneously. However, if it is not restricted in the rules of the game, more than one agent may perform the same role to increase the performance. Another problem is the training of the system. The coefficients of the cost functions, which are used in the previous strategies, are fine tuned by using genetic algorithms. However, these cost functions are manually introduced to the system by human experts. For example, while calculating the cost of defense role for each robot, we use a specific formulation. This formulation may not be the optimum one for selecting the agent for defense action. This means, that the learning phase is designed to optimize the coefficients of the cost function instead of finding the optimum cost functions.

4.3 Reinforcement-based Market-driven Approach

To solve these problems, we use Reinforcement Learning (RL) to learn the role assignment process without changing the actual implementations of the roles. RL is a learning method, which can be used when the agent is only informed about the consequences of a sequence of its actions. The RL implementation replaces the role assignment in the original market algorithm mentioned above, with a Q(λ)-Learner (Peng and Williams, 1996). Q(λ)-Learning is a variant of RL and an extension to simple Q-Learning. Q-learning algorithm uses only one step data while updating Q-values. Eligibility traces can be used to keep track of all the actions taken by the agent to reach a terminal state (Sutton *et al*, 1996). Q(λ) is widely used and it is generally believed to outperform simple onestep Q-learning, since it uses single experiences to update multiple state/action pairs (SAPs) that have occurred in the past. Generally, the Q-functions learned by the agents are represented in tabular form with one output value for each input tuple. But it is not possible to represent more realistic worlds with this approach, where the number of states can be prohibitively larger or continuous. One way of handling such problems is to use function approximation.

For function approximation and state generalization in RL, Cerebellar Model Articulation Controller (CMAC) is used. CMAC was introduced by Albus (Albus, 1975) as a simple model of the cortex of the cerebellum. It is a biologically inspired learning method similar to neural networks.

The main reason for using the CMAC is its efficiency in learning and operation, which makes it suitable for function approximation.

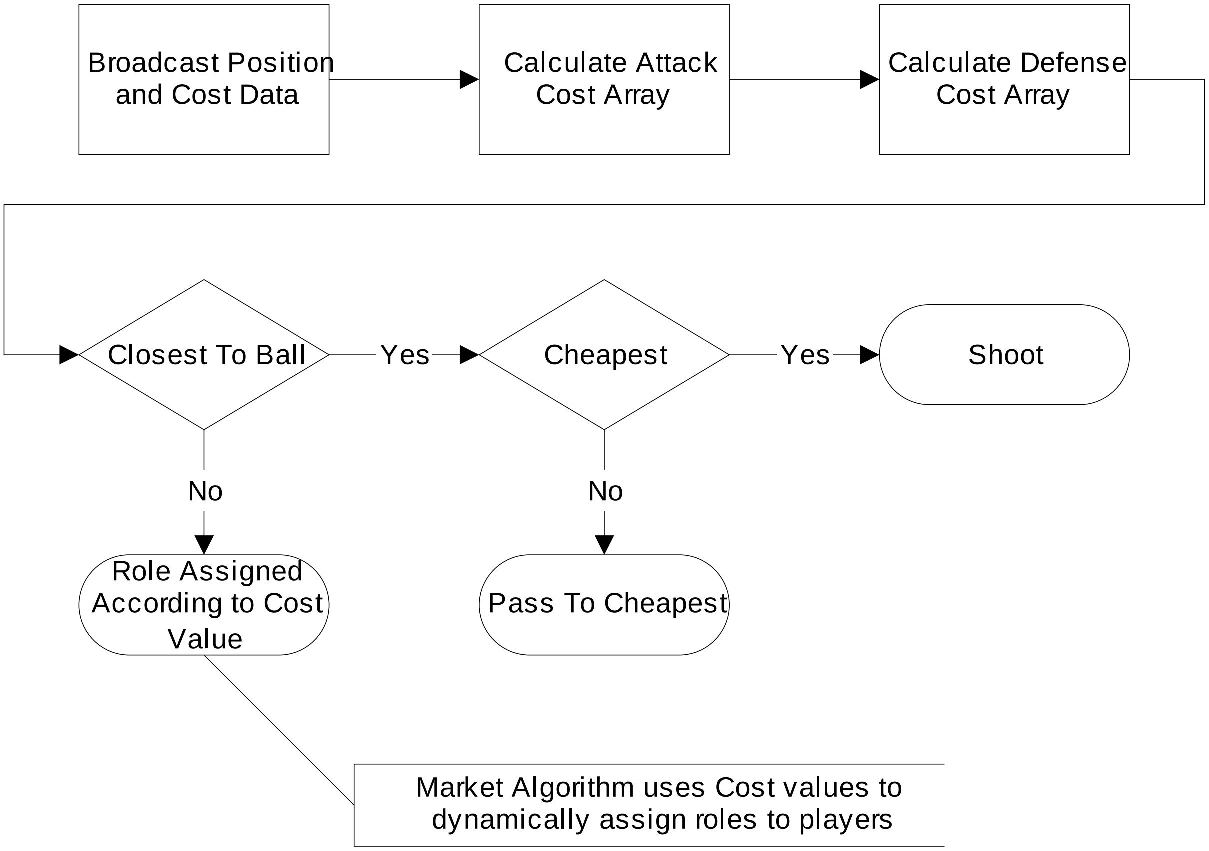

Broadcast Position and Cost Data
Calculate Attack Cost Array
Calculate Defense Cost Array
Closest To Ball
Yes
Cheapest
Yes
Shoot
No
No
Role Assigned According to Cost Value
Pass To Cheapest
Market Algorithm uses Cost values to dynamically assign roles to players

Fig. 4 Flowchart for task assignment

As in the previous strategies, the goalie role is statically assigned to an agent and does not change. Since it is always feasible to control the ball, the closest agent to the ball is assigned as the attacker and advances to the ball. The remaining three agents select the best role for themselves in the current situation according to the team policy (Fig. 4). State representation consists of perceptual and logical parameters. The perceptual parameters are relative distance to the ball, two goals, and other players (4 teammates and 5 opponents in this case). Each relative distance variable is composed of two parameters which are distance angle between the normal line and the agent. The logical parameters are the cost values (4 players' offensive and defensive cost values) and the closest player to the ball. There are 24 perceptual parameters and 9 logical parameters so totally 33 parameters are used to construct state vector. Unfortunately, this method suffers because of the large state vector. (Kose *et al*, 2004).

4.4 New Approach

The latest team developed in this study is an extension of the above described RL based approach. However, the state vector is modified here. In general, the state vector should include information about the agents and the ball. However, raw position data is not feasible to encode in the state vector. Therefore a grid decomposition for the field is proposed. In a real soccer game, the field is usually divided into three horizontal sections, where the upper and lower sections are the wings. Similarly, the field can be divided into three main vertical sections, which are forward, midfield and backward. Nevertheless, midfield can be further divided into two subsections. As a result the field is divided into 12 grids as shown in Fig. 5.

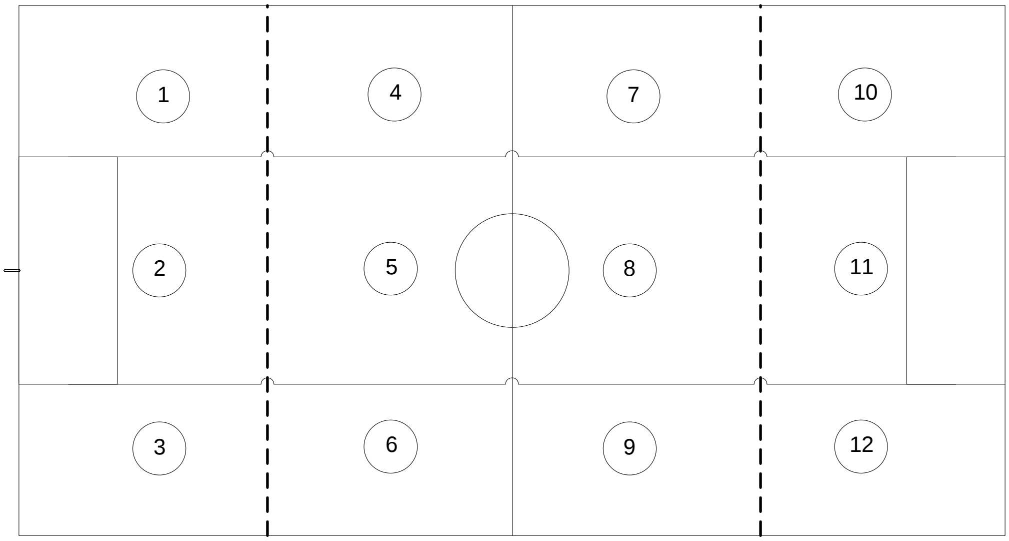
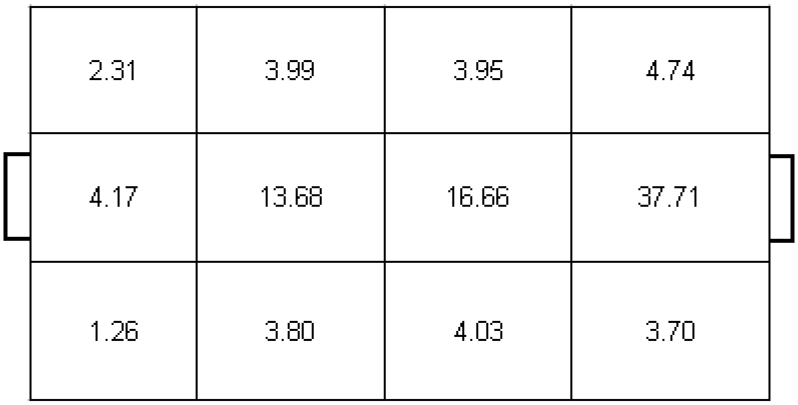


Fig 5. The grid decomposition of the field.

The state vector has the following metrics,

- **Ball position:** The number of the grid which contains the ball. (1 state vector element)
- **Ball possession:** The number of the team which possesses the ball. The possession is simply calculated by finding the agent closest to the ball. (1 state vector element)
- **Own role:** The role number assigned to the agent by the market team strategy. (1 state vector element)
- **Teammate positions:** The grid numbers of the teammate agents other than the goalie and the attacker. (3 state vector elements)
- **Opponent positions:** The number of opponents in each grid. We do not use the number of the grids because the density of the opponent agents in each grid is more important than the individual opponent agent positions. (12 state vector elements)

This state vector reduces the number of state variables from 33 to 18. The possible actions are the selection of attacker role, defender role, and secondary attacker role. The implementation details are the same as the previous reinforcement learning based team (Kose *et al*, 2004). After training, the average percentages of the positioning of the opponent robots on the playground are given in Fig. . The percentages are averages of three matches.



2.31	3.99	3.95	4.74
4.17	13.68	16.66	37.71
1.26	3.80	4.03	3.70

Fig. 6 Percentages of positioning of opponent robots

According to the 3 points system, in which the winner takes 3 points and each team take 1 point for draw, the performance of the learning team is given in Fig. 5, where y-axis is the cumulative point for 50 match epochs (Tatlidede *et al*, 2005).

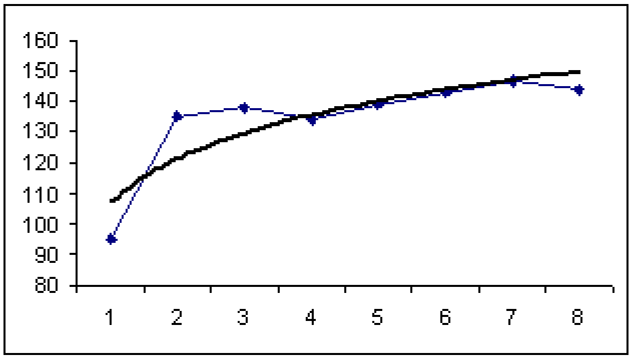


Fig. 5 Results during learning

4. Results

The new team, which is based on reinforcement learning with the reduced state vector, is compared to four other teams. The first opponent is *SchemaNewHetero*, which uses perceptual and motor schemas. This is a moderate built-in team delivered with the

Teambots simulator. The second team, *AIKHomoG* is one of the strongest built-in opponents. *AIKHomoG* team uses dynamic role assignment for strategy and potential fields for movement. The third team is the *RIYTeam*, which has very simple role assignment strategy as mentioned before. The next opponent team, is the *MarketTeam*. The final opponent is the *MarketQL*, which is the RL based team with large state vector.

Team	**Play**	**Win**	**Draw**	**Lost**	**For Goal**	**Against Goal**
SchemaNewHetero	90	60	28	2	163	19
AIKHomoG	90	78	9	3	203	38
RIYTeam	90	50	40	0	81	5
MarketTeam	90	36	48	6	56	15
MarketQL	90	33	50	7	53	17

Table 1. Results

As seen in Table 1, the proposed team defeats all other opponents. It should be stressed that the only difference between the *RIYTeam* and the *MarketTeam* and the new team is the role assignment strategy. The new team is trained to find the optimum role assignment strategy without any constraint. The previous teams suffer from the assumptions which are made by human experts. For example in *RIYTeam*, the robot closest to the ball is selected as the attacker. This assignment is subject to the assumption that the distance is the only metric which affects the role assignment strategy. However, in the new team, there is no such assumption, which means at the beginning of learning phase each agent is free to choose any role, except goalie.

5. Conclusion

In this work, the target is the coordination problem among the members of a robot soccer team. In order to solve this problem several methods which are extensions of a market-driven approach are implemented. In this work these approaches are studied and compared in detail.

The first developed method was the method with static role assignment. Since it has many drawbacks, a novel market-driven approach was implemented to increase the team success by using the full benefits of collaboration. In this first version, roles are fixed, and the agents are assigned suitable roles according to the available cost functions to increase success, in the current situation. This strategy was quite successful and takes good results in during the matches done by other teams, but there are different teams with different game strategies like in the real life case, so there is a need to change the game strategy (e.g. playing offensive or defensive) according to the opponent team strategy. So the original *MarketTeam* is extended by the addition of reinforcement-based learning method, which allows the team to learn new strategies, as it plays matches with other teams, and use a dynamic strategy to choose the roles for the players. Later this strategy which uses market-based cost values and other domain specific values in its state vector is further extended to eliminate the drawbacks, and increase success.

The results show that reinforcement learning is a good solution for role assignment problem in the robot soccer domain. However, encoding of the problem into the learner is an important issue. When the configuration space is quite large, the policy may not cover all possible states. As a result, the agent is forced to select random actions and the system performance decreases.

The communication problem is not addressed in this work. It is assumed that each agent can broadcast limited amount of data. The controller simply collects available data from any other agent. The data may be noisy. Since, at each frame the communication data is refreshed, the error is not cumulative.

The solution can also be used in other highly dynamic environments where it is possible to introduce some reinforcement measures for the team. In the robot soccer domain, the reinforcement measures are the goals scored by either our team or the opponent team.

8. References

Albus, J.S. (1975). Data storage in the cerebellar model articulation controller (CMAC), *Transactions of the ASME: Journal of Dynamic Systems, Measurement, and Control,* pp. 228-233

Arkin, R. C. and Balch, T. (1998). Cooperative multi-agent robotic systems, *Artificial Intelligence and Mobile Robots,* D. Kortenkamp, R. P. Bonasso, and R. Murphy, Eds. Cambridge, MA: MIT/AAAI Press

Balch, T. (2000). *Teambots*, *Available from:* http://www.cs.cmu.edu/~trb/*Teambots*/ Domains/SoccerBots, *Accessed:* 2000

Cao, Y.U.; Fukunaga, A.S. and Kahng, A.B. (1997). Cooperative mobile robotics: antecedents and directions, *Autonomous Robots*, 4, pp.2–7

Dias, M. B. and Stenz, A. (2001). A Market Approach to Multi-robot Coordination, *CMU-RI-TR-01-26*, August

Dudek, G.; Jenkin, M.R.M.; Milios, E.and Wilkes, D. (1996). A taxonomy for multi-agent robotics, *Autonomous Robots*, 3(4), pp. 375–397

FIRA (2003). *Available from:* http://www.fira.net/, *Accessed:* 2003.

Frias-Martinez, V.; Sklar, E. and Parsons, S. (2004). Exploring auction mechanisms for role assignment in teams of autonomous robots, *Proceedings of the RoboCup Symposium,* Lisbon, 2004

Gerkey, P.B. and Mataric, M.J. (2002). Sold!: Auction methods for multi-robot coordination, *IEEE Transactions on Robotics and Automation, special issue on Advances in Multi-Robot Systems,* 18(5), 758-786

Kaplan, K. (2003). Design and Implementation of fast controllers for Mobile Robots, Master Thesis, January

Kose, H.; Mericli, C.; Kaplan, K. and Akin, H. L. (2003). All Bids for One and One Does for All: Market-Driven Multi-Agent Collaboration in Robot Soccer Domain, *Proceedings of Computer and Information Sciences-ISCIS 2003, 18th International Symposium*, pp. 529-536, LNCS 2869, Antalya, Turkey, November 2003

Kose, H.; Tatlidede, U.; Mericli, C.; Kaplan, K. and Akin, H. L. (2004). Q-Learning based Market-Driven Multi-Agent Collaboration in Robot Soccer, *Proceedings of TAINN 2004, Turkish Symposium On Artificial Intelligence and Neural Networks*, pp.219-228, Izmir, Turkey, June 10-11, 2004.

Peng, J. and Williams, R. (1996). Incremental multistep qlearning, *Machine Learning*, pp. 283-290

ROBOCUP (2003). *Available from:* http://www.robocup.org/, *Accessed:* 2003.

Song, P. and Kumar, V. (2002). Potential field based approach to multi-robot manipulation, *In Proceedings of the 2002 IEEE Int'l Conf. on Robotics and Automation,* Washington D.C., May

Stone, P. and Veloso, M. (2000). Multi-agent systems: A survey from a machine learning perspective, *Autonomous Robots*, 8(3)

Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding, *Advances in Neural Information Processing Systems,* MIT Press

Svestka, P. and Overmars, M. (1998). Coordinated path planning for multiple robots, *Robotics and Autonomous Systems*, 23, pp:125 – 152

Tatlidede, U.; Kaplan, K.; Kose, H and Akin, H. L. (2005). Reinforcement Learning for Multi-Agent Coordination in Robot Soccer Domain, *Fifth European Workshop on Adaptive Agents and Multi-Agent Systems*, Paris, March 21-22, 2005 (accepted)

Zlot, R.; Stenz, A.; Dias, M. B. and Thayer, S. (2002). Multi-Robot Exploration Controlled by a Market Economy, *Proceedings of the IEEE International Conference on Robotics and Automation*, May