Predicting the Future State of the Robocup Simulation Environment: Heuristic and Neural Networks Approaches*

Mohsen Sharifi, Hamed Mousavian, Amir Aavani
Computer Engineering Department
Iran University of Science and Technology
{msharifi, mousavian, aavani}@iust.ac.ir

Abstract - *One of the challenges in multi-agent systems is the prediction of the future state of the environment. This is because the future behavior of each agent and its relationship with other agents in the environment must be predicted according to an appropriate agent model. Such a prediction about the future of the environment in a large state space, like the Robocup simulation environment, in which most agents act on the basis of uncertain knowledge, is quite hard. Given the above context, a novel multi-agent game presentation and analysis tool has been developed. One of the agents in this tool is responsible for the prediction of individual and team behaviors. To overcome the problem of uncertain knowledge, the prediction is only made for the simulated commentator who needs to anticipate the player agent receiving the ball after a shoot by another player agent. The predictor agent has been implemented both heuristically and with neural networks. Using the logged data form Robocup 2002 simulation league, as our test data, the neural network approach had a higher success rate of true predictions than our heuristic simulation.*

Keywords: multi-agent systems, simulated commentator, Robocup simulation environment, game presentation and analysis tool, predictions. Automatic Commentary System

1 Introduction

Agents in a *Multi Agent System* (MAS) interact in order to achieve some goals. If they can predict the future state of the environment in which they act, they can expect a better chance of success in matching their individual and group objectives. To do so, they need to anticipate the intention and behavior of other agents according to an appropriate model for every other agent in the environment [4].

An appropriate model for an agent can help other agents in the way they cooperate, coordinate, assist or counteract (in an adversarial environment) with this agent [6]. To be useful, the model should contain knowledge about the *desires* and *beliefs* of the agent [15]. It is however quite hard to model the behavior of such agents and predict the future state of the environment in which these agents interact, in an uncertain environment like *Robocup Simulation Environment*, which has a huge state space and a large action space.

These impediments have triggered our research for acceptable solutions to be practical in Robocup Championship, which have not been exercised before.

We believe that the performance of a *Simulated Soccer Team* can be improved a lot if every agent has an appropriate model of the environment and knows about the intention of other player agents and can predict their behaviors beforehand.

To give an example, consider a player agent who has the ball and intends to pass it to one of its teammates. If it can anticipate how other agents, be they teammates or opponents, respond to changes in their environment, it can decide much more better on how, where and to whom to pass the ball. If other players (teammates or opponents) have an appropriate behavioral model of the player who has the ball, they can anticipate its intentions and move to a proper space either to receive the pass (teammates) or to track the ball (opponents).

Since the players' perceptions in Robocup soccer simulation are both noisy and uncertain, such anticipations by the player agents are very complex. We therefore have decided to consider the prediction of the future state of the environment for *Simulated Commentator*, since it can receive all information about the soccer game from the *Soccer Server*, without any noise.

The commentator must report some of the events in the soccer game exactly at the time they appear (i.e. real-time), e.g. when a player dribbles. Also it must be able to predict the behavior of player agents in the field. In other words, it must report future events. To give a more specific example, after the ball is kicked by a player, and before the ball reaches its destination, the commentator agent should find out the intention of the player on the basis of the behavior model of the players, in order to anticipate who will receive the ball or whether the shoot is a pass, a shoot towards the goal, or other kinds of shoots.

Since the performance evaluation and behavior analysis of agents have always been a concern in MASs [9, 7], several tools have already been developed for the analysis of the behavior and evaluation of the performance of simulated soccer teams in recent years [13, 3]. These tools have assisted the developers of simulated soccer teams in improving the performance of their teams [10].

We have developed a presentation and analysis tool, that in addition to being an *Automatic Commentary System* [2, 1], it has been used for investigating some of the MASs concerns, such as the prediction of the future state of the environment, as well as for the analysis of the individual, group and team behaviors of agents [16].

In the next section, the proposed architecture of our presentation and analysis tool is presented. Section 3 details specifically the problem of determining the player who will receive the ball after it has been kicked by another player. Section 4 presents a solution to the problem, using a heuristic simulation approach. An alternative solution using neural networks is presented too in section 5. The results under both approaches are reported separately, and are consequently compared and used to conclude the paper.

2 The Proposed Architecture

As is shown in Figure 1, our presentation and analysis tool has two parts: the *Commentary System* (providing analysis functionality), and the *Monitoring System* (providing presentation functionality). The former has the responsibility of analyzing and reporting the game, while the latter provides visual [12, 5] and auditory [2] presentation of the information given by the commentary system, as well as the information about the playing field.

The simulated commentary system is developed as a MAS in which each agent is given a specific responsibility and monitors the game in that respect. The agents cooperate to provide commentary information about the game, so that they can be presented online and real-time by the monitoring system.

The Soccer Server outputs a stream of data containing all required information about the playing filed at each cycle, without any noise in the information. The data stream at each cycle is preprocessed. Visualizable information is sent to the visualizer, and information necessary for analysis is sent to the appropriate agents of the commentary system. The agent, who is responsible for recognizing the individual and group movements, focuses on the ball movement and figures out the intention of the player agent acting on the ball in order to identify the skills or tactics like: pass, shoot to goal, or ball interception.

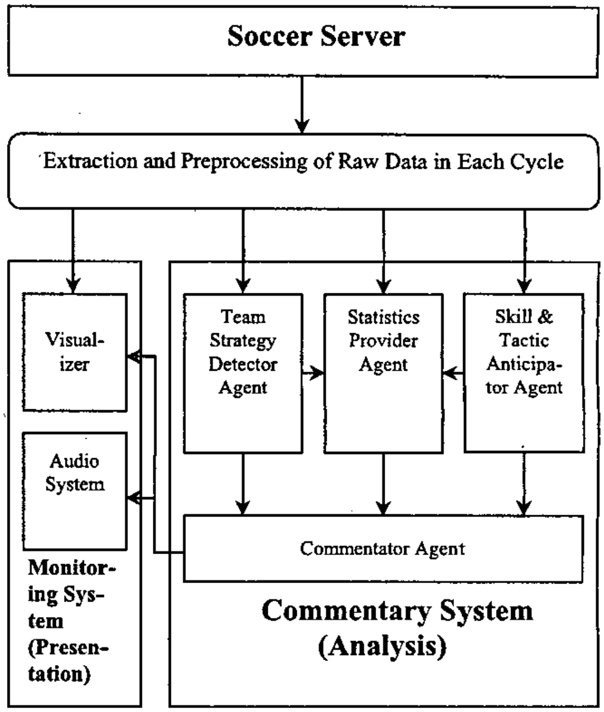

Figure 1. The architecture of game analysis and presentation tool.

The agent responsible for the detection of team strategies overlooks the whole playing field in search of strategic movements of the teams [11], such as overall team formation and offside line formation [11, 14], and reports its findings to the commentator.

There is another agent in the commentary system that provides statistics to the commentator, out of the data stream received from the Soccer Server such as data about the play mode. It also uses the information provided by other two agents in the commentary system. Some of the statistics are: goal scores, the number of off sides, the number of substitutions, the number of shoots towards the goal, and the ball possession percentages for each team. It is important to note that this agent always has the most up to date statistics about the current status of the game.

The commentator agent receives instant information from the other aforementioned three agents in the commentary system. It then reports the appropriate information to the monitoring system for presentation. The duration and content of the report is determined by the predefined configuration settings which characterize priorities.

Given the designed architecture, the next section discusses the problems encountered by the agent responsible for recognizing and predicting the skills and tactical movements of players.

3 Anticipating Who Receives the Ball

The agent responsible for detecting skills and tactics in our architecture must predict the intention of the player agent kicking the ball. According to the state of the playing field at the moment the ball is kicked, it figures out the type of shoot, i.e. whether it is a shoot towards the goal, pass to a teammate or other varieties.

This paper concentrates on how the commentator differentiates between a pass and other varieties of kicks, and how it determines the player agent receiving the ball after a kick to the ball. The player's number and team must be predicted immediately after the kick. In other words, it must predict the future state of the environment.

Two techniques have been used to provide the above functionality. The first approach is based on heuristic and the second one uses neural networks.

4 Heuristic Simulation Approach

In this approach, we simulate the playing field immediately after the ball is kicked, until a player agent receives the ball. This is done according to the algorithm shown in Figure2, which repeats the simulation for n cycles, until the ball is in the kickable area of a player agent. As in SoccerServer, the ball is moved in the direction of the kick, and the speed of the ball is multiplied by the *BallDecay* rate. It is also assumed that all player agents try *their best* to run towards the ball during these n cycles.

The algorithm associates a t_i to each player i, which represents the minimum cycles the player i can reach the ball, at best. The player with the lowest t_i is predicted to be the receiver of the ball. In this way, the number of the receiver player, as well as the success or failure of the pass, that is whether it had been a pass to a teammate or not, is determined.

To test our algorithm, we have used the log files of three rounds in Robocup2002 simulation league separately, namely *Round1*, *Round2* and *Final Round*. We have compared the results of several running of the algorithm with the historical data in these log files. In separate tests, we changed the start time of our simulations from one cycle up to six cycles, from the kick time of the ball. The percentage of the cases where the ball has been received by the players who had the smallest t_i, were calculated in each test. The results of the simulations are shown in Table 1.

```
1 set t[i] = 0 for all i, 1 ≤ i ≤NumberOfPlayers
2 ReachedToBall = 0
3 n = 0
4 while ReachedToBall < 1 do
5  n = n + 1
6  BallPos = BallPos + BallVel
7  BallVel = BallVel * BallDecay
8  for each player p[i] where his t[i] = 0 do
9     Pos = position of p[i]
10    Vel = velocity of p[i]
11    Teta = the best direction to Dash for p[i]
12    for i = 1 to n do
13      U=Vel + MaxPower * PowerRate *   (cos(Teta),sin(Teta))
14      Vel = U * PlayerDecay
15      Pos = Pos + U
16    end for
17    if |Pos – BallPos| < |Pos – position of p[i]| then
18        t[i] = n
19        ReachedToBall = ReachedToBall + 1
20    end if
21  end for
22 end while
```

Figure 2. The simulation approach algorithm. This algorithm finds the number of cycles needed for the first player to reach to the ball. Note that if we change the number 1 in line 4 of the code to the number of players, this algorithm will find a t_i for each player i.

Table 1. Simulation approach results

	1 Cycle	2 Cycles	3 Cycles	4 Cycles	5 Cycles	6 Cycles
Player 1	71.38	76.81	78.99	79.49	79.75	81.07
Player 2	17.87	14.49	12.38	11.26	10.65	10.40
Player 3	4.89	3.80	3.20	3.02	3.19	2.99
Player 4	1.87	1.33	1.37	1.57	1.63	1.47
Player 5	0.83	0.77	0.92	1.01	1.10	1.08
Player 6	0.57	0.55	0.67	0.78	0.88	0.90
Player 7	0.39	0.45	0.51	0.72	0.61	0.71
Player 8	0.37	0.34	0.49	0.50	0.59	0.49

The rows in Table 1 show the average percentage of the cases where the ball has been received by the eight players who had the minimum t_i. The columns show the simulation results from one cycle up to six cycles after the ball had been kicked.

5 Neural Networks Approach

In another attempt, we used neural networks to classify the variety of shoots [8] in order to distinguish the kicks which lead to the possession of the ball by the player with the least t_i, from other types of kicks. The architecture of the deployed neural network, as well as the details of the inputs and outputs to and from the network, is described further.

The same log files from Robocup2002 simulation league, which have been used in our simulation approach, were also used here in the provision of training data set. The required information has automatically been extracted from the scenes containing the kicks to the ball. The extraction works as follows.

Whenever a shoot from player i to player j is detected in the log file, information about the playing field, one cycle after the kick to the ball, is extracted. The coordination axis is first transferred to the position of ball in line with the direction of the ball movement (as shown in Figures 3a and 3b). The Cartesian coordinates of the players are then changed to Polar coordinates.

The inputs to the neural network consisted of the polar coordinates (r, θ) of the eight players having the least t_i, as well as the speed of the ball. The output of the network was representative of whether the first player with the least t_i will receive and possess the ball or not.

Our training data were gathered from all the scenes of kicks to the ball at cycles 1 up to 1000, from all the logged games in the three rounds of Robocup2002. But the test data were derived from all the scenes which happened between cycles 1001 to 1500 in those games. Our training data set had nearly 5000 samples, and our test data set included about 2500 samples.

To simplify the network, we have not considered the speed, stamina, and direction of the head of the players in our training data; their inclusion might have misled the network.

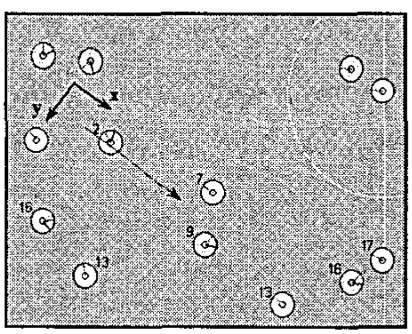

Figure 3a. Transferring coordination axis to the position of the ball and rotating them to lie in the direction of the ball movement, after the kick to the ball

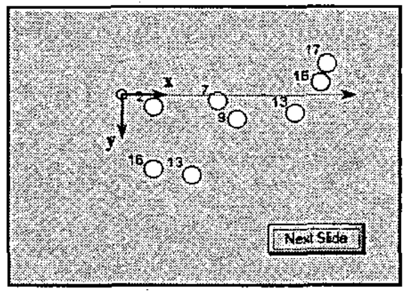

Figure 3b. Extracting polar coordinates from the new Cartesian coordinates of the eight players with the least t_i

A *Back Propagation Neural Network (BPNN)* with two hidden layers that each contained 160 neurons had been used. The network was trained by the *Gradient Decent with Momentum and Adaptive Learning Rate* method, until its *Mean Square Error* reached 10^{-1}.

As in our first simulative approach, we have repeated the training and testing of the network for one up to six cycles, after the kick of the ball. The results are shown in Table 2. The table shows the percentage of the cases that the network had predicted the receipt or not receipt of the ball by the player with the least t_i.

Table 2. Neural network approach results

	1 Cycle	2 Cycles	3 Cycles	4 Cycles	5 Cycles	6 Cycles
True Classification	71.85	77.59	78.34	80.96	82.64	81.12

6 Comparison of Results

To compare the performance of the two approaches described in the previous sections, the results under the two approaches are diagrammed in Figure 4.

The *Round1*, *Round2* and *Final Round* graphs show the success rate of our simulation approach in finding the players with the least t_i as the player receiving the ball after a kick, in the 3 rounds of Robocup2002 simulation league. The average success rate of this approach is shown by the *Average* graph, which is based on the number of data at each round.

The *Neural Net* graph shows the success rate of the neural networks approach in detecting correctly the cases where players with the lowest t_i, had received the ball after a kick.

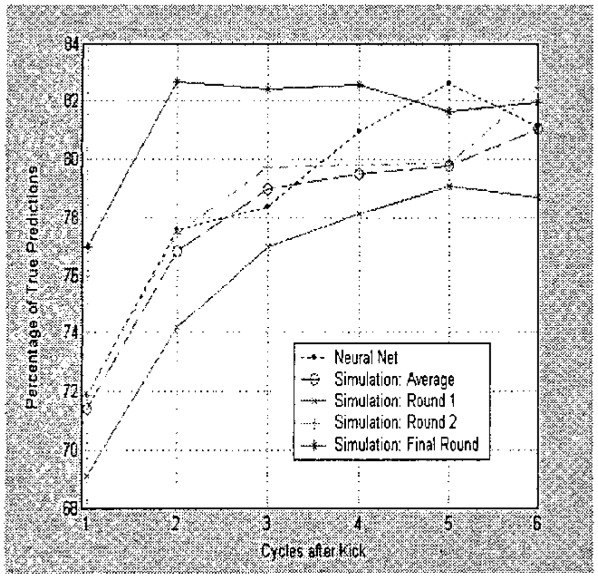

Figure 4. Comparison between neural networks approach results and heuristic simulation, for different rounds of Robocup 2002 simulation league.

An interesting point which can be implied from the results shown in these graphs is that the prediction in strong plays can be made more successfully. We can therefore be justified and hopeful to have introduced a promising MAS commentary system for strong plays.

Another point common in both approaches is that the success rate of correct prediction of the players receiving the ball after a kick, improves as more cycles are considered.

7 Conclusion and Future Works

In this paper, we presented the unattended problem of anticipating of player receiving the ball after a kick to the ball, form the viewpoint of a commentator agent in a Robocup Simulation Environment. Accordingly, we introduced our MAS-based architecture for tackling this problem, and applied both a simulation and neural network techniques to solve this problem. It was shown experimentally that an acceptable degree of correct prediction under both approaches is attainable. The results even showed better correct predictability of our approaches in strong plays.

As a future work, we are currently applying the two approaches presented in this paper to the prediction of the type of a pass and also to the prediction of the shoot to goal, by the agent responsible for predicting the individual and group movements of players in our presented architecture.

Our first simulation approach can be improved further by employing a behavior model of each player of the game by the commentator, instead of considering the best the players may achieve. The second approach can also be tested for other players in the game, other that those with the least t_i.

Acknowledgment

The work presented in this paper, is part of a continuous research activity, sponsored by the Computer Engineering Department at Iran University of Science and Technology, to prepare interested graduate and undergraduate students of the Department for Robocup championships. We would also like to thank Dr. Adel Rahmani for his support and guidance to the *Iranian* Robocup team.

References

[1] Andre, E., Herzog, G., Rist, T. Generating Multimedia Presentations for Robocup Soccer Games. In: *Robocup-97: Robot Soccer World Cup I*, 1997.

[2] Binsted, K.: Character Design for Soccer Commentary. In: *Proceedings of the Second International Workshop on Robocup*, pages 22-36, 1998.

[3] Frank, I., Tanaka-Ishii, K., Okuno, H., Akita, J., Nakagawa, Y., Maeda, K., Nakadai, K., Kitano, H.: And the Fans Are Going Wild! SIG Plus MIKE. In: Stone, P., Balch, T., Kraetzschmar, G., editors, *RoboCup-2000: Robot Soccer World Cup IV*, pages 139-148. Springer-Verlog, 2001.

[4] Howe, A.E., Cohen, P.R.: Understanding Planner Behavior. *Artificial Intelligence 76* (1995) 125-166.

[5] Jung, B., Oesker, M., Hecht, H.: Virtual Robocup: Real-Time 3D Visualization of 2D Soccer Games. In Veloso, M., Pagello, E., Kitano, H., editors, *RoboCup-99: Robot Soccer World Cup III*, pages 331-344. Springer-Verlog, 2000.

[6] Kaminka, G., Fidanboylu, M., Chang, A., Veloso, M.: Learning the Sequential Coordinated Behavior of Teams from Observations. In Kaminka, G., Lima, P., Rojas, R., editors, *RoboCup-2002: Robot Soccer World Cup VI*. Springer-Verlog, 2003 (forthcoming).

[7] Kaminka, G., Pynadath, D., Tambe, M.: Monitoring Teams by Overhearing: A Multi-Agent Plan-Recognition Approach, Journal of Artificial Intelligence Research, Volume 17, pp. 83-135, 2002.

[8] Maire, F., Taylor, D.: A Quadratic Programming Formulation of a Moving Ball Interception and Shooting Behaviour, and Its Application to Neural Network Control. In: Stone, P., Balch, T., Kraetzschmar, G., editors, *RoboCup-2000: Robot Soccer World Cup IV*, pages 327-332. Springer-Verlog, 2001.

[9] Pynadath, D.V., Tambe, M., 2002. *Multiagent Teamwork: Analyzing the Optimality and Complexity of Key Theories and Models.* In Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems, pp. 873-880.

[10] Raines, T., Tambe, M., Marsella, S.: Automated Assistants to Aid Humans in Understanding Team Behaviors. In Veloso, M., Pagello, E., Kitano, H., editors, *RoboCup-99: Robot Soccer World Cup III*, pages 85-102. Springer-Verlog, 2000.

[11] Riley, P., Veloso, M., Kaminka, G.: An Empirical Study of Coaching. In: *Proceedings of Distributed Autonomous Robotic Systems 6*, . Springer-Verlog, 2002.

[12] Shinjoh, A., Yoshida, S.: Autonomous Information Indication System. In Veloso, M., Pagello, E., Kitano, H., editors, *RoboCup-99: Robot Soccer World Cup III*, pages 469-480. Springer-Verlog, 2000.

[13] Tanaka-Ishii, K., Noda, I., Frank, I., Nakashima, H., Hasida, K., Matsubara, H.: MIKE: An Automatic Commentary System for Soccer. In: *Proceedings of ICMAS-98*, pages 285-292, 1998.

[14] Visser, U., Drucker, C., Hubner, S.: Recognizing Formations in Opponent Teams. In: Stone, P., Balch, T., Kraetzschmar, G., Editors, *RoboCup-2000: Robot Soccer World Cup IV*, pages 391-396. Springer-Verlog, 2001.

[15] Wendler, J., Hannebauer, M., Burkhard, H., Myritz, H., Sander, G., Meinert, T.: BDI Design Principles and Cooperative Implementation in RoboCup. In Veloso, M., Pagello, E., Kitano, H., editors, *RoboCup-99: Robot Soccer World Cup III*, pages 531-541. Springer-Verlog, 2000.

[16] Wunstel, M., Polani, D., Uthmann, T, Perl, J.: Behavior Classification with Self-Organizing Maps. In: Stone, P., Balch, T., Kraetzschmar, G., editors, *RoboCup-2000: Robot Soccer World Cup IV*, pages 108-118. Springer-Verlog, 2001.