n-th.com RoboCup-2002 Team

S.S.Stepanov[1] and A.Yu.Zaslavsky[1,2]

[1] N-th.com Research, Dnepropetrovsk, Ukraine
[2] Physics Department, Dnepropetrovsk University, Dnepropetrovsk, Ukraine

Abstract. The *n-th.com* robosoccer team is based upon the research carried out during the development of the team for a computer soccer simulation contest held in Eastern European countries (Robosoccer-RU League) [1]. The algorithms were considerably expanded and improved to match the RoboCup competition rules. The behavior of the team agents is governed by expert system defined in the fuzzy logic basis; the neuronet-like learning block is used to resolve the collisions between the rules. Results of extensive testing are discussed, and the main prospects for future work are outlined.

Team name: **n-th.com**
Affiliation: n-th.com Research and Dnepropetrovsk University (Dnepropetrovsk, Ukraine)
URL: http://n-th.com/robocup/
Team leader: Sergey S. Stepanov
E-mail: robocup@n-th.com
Team members: Sergey Stepanov, Alexander Zaslavsky, Igor Chavychalov, Vitaly Khit, Igor Chub

1 General Description

This team implements common concepts of the soccer game - player types, positioning, formations and others, which were introduced in robotic game in [2]. The core algorithms which determine the behavior of the agents are based upon an expert system defined in the fuzzy logic basis. If the current situation is not defined within the expert system rules, or the decision by those rules is below a certain confidense threshold, a neuronet-like learning block steps in to resolve the situation. This decision could then be used to adjust the expert rules. Team players remember their actions and world state in the past. Every player is assigned a certain role at start-up, however during the match agents can temporarily switch their responsibilities if the estimated effect on the world state is positive. All clients use a low level routines to reconstruct a current state from the incomplete information sent by the server. Various tests were run with different processing models to estimate and minimize the errors in the agents' knowledge of positions and velocities of objects on the field.

2 Defensive Tactics

The current team implementation includes 12 different defensive formations, which are used alternately by the agents. Which one is used, depends on the current positions of the ball and the actions of opponent's attack. The model of determining the current mode is stable enough, so that all defenders are in the same mode most of the time. The most important here is that every defenceman has a responsibility zone and tries to check on opposing attackers and steal the ball. Agents estimate the utility of their prospective action (pass, dribble, mark the pass line, etc) using a system of adjustable weights, and selecting the optimal one. Decision to pass, for example, is done using the fuzzy logic model, which takes into account a number of criteria with dynamically updated weights, with high priority assigned to trying to pass to a free mid-field player (counter-attack), and comparatively low to dribbling or passing back to a goalie. This tactics allows defenders to effectively prevent breakaways.

3 Goalie Behavior

Different strategies were tested for the goalie client, including running to the ball, waiting for the forward to shoot, and their combinations. As a result the following hybrid behavior was found to be the best against a wide range of teams. Goalie blocks the line between the ball and a certain point on the goal-line, in effect covering as much of the ball-goal angle as possible. A run-to-catch is only attempted if the ball is within the predefined range (which may be adjusted during the match). This works quite well provided the teammates properly fulfill their defensive tasks. On breakaways, the optimal tactics depends very much upon the behavior of the forward (which goalie tries to learn during the match). A distinctive feature of our goalie, is that it does not tries to catch the ball at every possibility, but chooses between catch and kick based on a set of rules. Also, it tries to pass forward as soon after catch as a well-positioned teammate is available. Effectiveness of not keeping the ball for long was proved in matches against itself and other teams, as it allowed for a fast counter-attack and a probable breakaway.

4 Attack

Two type of forward tactics are implemented, for acting in the center and in the wings. Those roles are not assigned to particular players, but rather each forward takes up the role that is appropriate for it's present position. All forwards (not controlling the ball) try to keep a good position in or near the opponent's penalty area. For this purpose a grid is defined on the field, with relative weights in the nodes, which allow them to estimate their position. These coefficients are updated dynamically and reflect the current actions of the opponents, line of off-side, etc. The role of the main striker is assigned to the center forward, whose task is to take a position within shooting range from the goal and far enough from any opponent to be able to tackle the ball and shoot. Different possible ways of how and when to shoot are analyzed. Our algorithm is close to [3], in addition a notion of a “free angle” is used (kick into such angle cannot be caught by any opponent), and the optimal spot within the goal is chosen. All the calculations are made using the dynamically adjusted ideal opponent model, that is, it is assumed that the goalie's speed, for instance, is equal to kV_{max}, where $k = 1$ at the start of the game and then varied based on observations of the opponents' play.

5 Standard situations

For all the set-play situations (kick-in, corner, free kick) custom routines are implemented. The main goal here is to keep control of the ball, or to mark the most important pass lines to the opponent players. When player kicks the ball, it uses a number of measures to choose the optimal teammate to pass (this measures include the utility of the teammate position, opponent's ability to intercept the ball, general situation on the field, etc). Again, unless more specific information is available, the other side actions are supposed to be optimal, too. Also, some set combinations are used in attacking; they are applied mainly when the offensive unfolds quickly enough to allow for enough space near the opposite goal.

6 Development Tools

For this team implementation we have not used any available parsing libraries. This decision was a result of the fact that some basic routines for restoring the world state from incomplete visual information were already developed and used in the earlier competitions within the Robosoccer-RU [1]. A rather simple and general object-oriented library was written and utilized for building the intellectual agents. The low-level modules of the library parse the information from the server and assemble it into the current state of all the objects. The agent itself is a descendant module, and focuses only on decision taking. The library is a multi-platform one (Linux, Windows) and significant efforts were made to document it well.

For testing and debugging purposes a custom monitor was developed, which provides additional features and tools, extensive logging capabilities among them. It is possible, for instance, to zoom in on the active field area, to visualize the trajectories of the objects, to replay step by step writing out debug information, compare an agent's picture of the world with the real one, etc.

7 Future Work

Test matches played against varied versions of itself as well as teams-participants of the RoboCup-2001, and various experiments with agents behavior in training conditions have revealed a number of issues that need further improvement:

- Agents' notion of the current world state can still be improved. While it is quite adequate generally, increasing it's accuracy in the critical situations (e. g. forward shooting at the goal, or goalie catching the ball) would much improve the performance.
- Technical abilities of players (dribbling, stealing the ball) are far from perfect, which partially can be explained by the lack of accuracy in the world state (this is indeed shown in experiments).
- Inter-agent messages are not yet utilized at their full capacity. Although it seems implausible to try to communicate the world state with present limit on the message size, an effective use of it can be made in the situations not handled well by the existing expert system rules.
- Agents should adapt their play when the opponent's strategy is changed. Now only most basic analysis of the opponents actions is performed.
- Online coach is not used at all yet. Adding it would help in many ways, in particular, to dynamically build the models for opposing side strategy, and switching our team to the appropriate playing mode.

Apart from these tasks, further development of testing tools is now underway. A server emulator is being developed that will provide information to the clients in the form which will allow to skip the low-level phases of visual data processing (networking, parsing, restoring incomplete information). We hope that this will allow to concentrate on high-level problems of decision taking and multi-agent communication.

References

1. Stepanov, S.S. et al.: Robosoccer-RU Open Simulation League: Principles and Algorithms. Talk subm. to RoboCup International Symposium (Fukuoka, Japan), 2002.
2. Stone, P. and Veloso, M.: Task Decomposition, Dynamic Role Alignment, and Low-Bandwidth Communication for Real-Time Strategic Teamwork. Artificial Intelligence, **110** (1999) p.241-273. Defining and using Ideal Teammate and Opponent Agent Models. Proceedings of the Twelfth Innovative Applications of AI Conference, 2000.
3. Stone, P., Riley, P., and Veloso, M.: Defining and using Ideal Teammate and Opponent Agent Models. Proceedings of the Twelfth Innovative Applications of AI Conference, 2000.