Borregos 3D – 2006 Team Description

Carlos Bustamante, Leonardo Garrido, and César Flores

Center for Intelligent Systems
Monterrey Institute of Technology
Monterrey NL 64849, Mexico,
{cfbh, leonardo.garrido}@itesm.mx
cesar7@gmail.com

Abstract. In this paper, we describe the architecture and features of Borregos 3D team. Our team is relatively new, we competed in Iran's RoboCup AI Games 2005 with a very basic agent design. Now we have incorporated an innovative approach for decision-making based on classification using a Hybrid Fuzzy-Bayes system. We sent a paper to the RoboCup International Symposium 2006 describing this approach. Also, we have improved some of the skills like kicking the ball to some point with a desired final speed using multiple regression and interpolation techniques. The final goal is to have a fully competitive Hybrid Fuzzy-Bayesian team for contributing with research in uncertainty and management of imprecise knowdlege in noisy environments. The soccer server 3D platform is an excelent test-bed for our purposes.

1 Introduction

RoboCup 3D simulation league provides an interesting platform to Multiagent systems and Artificial Intelligence. The soccer server 3D presents a more realistic real-time noisy environment and an excelent interface for full sense-think-act methodology on agent implementations.

In the last years, our institution has participated in RoboCup in the 2D league. The work has been focused on action factibility, emergent collaborative behavior, online planning and opponent modelling [1].

Now a new group of M.Sc. students of Monterrey Institute of Technology, Mexico, has incurred in the relatively new 3D soccer league. Although some ideas of 2D can be migrated to the new team, the 3D environment characteristics imply an extra difficulty for the agent design, mainly in the lower layers. The first attempt to our agent design was based on the *agenttest* example included in the soccer server 3D distribution. We already competed in Iran's RoboCup AI Games 2005, in which we used a very basic agent design.

Borregos 3D is now based in the TsinghuAeolus liberated code [2]. We have done a lot of research focused on improving the basic skills mainly through regression and interpolation methods and incorporated an innovative technique namely Fuzzy-Bayesian classificationfor the decision-making module of the team.

In the next sections we will show our agent architecture, the approach to some of the basic skills and the role of the Fuzzy-Bayes Classifier. Finally, we show some conclusions and future research.

2 Agent Architecture

Our agent design has a better structure compared to the last year's team. It can be represented as a module diagram (see fig. 1). There is a Main Module whose function is to serve as a layer between the communication server and the agent. It is there where the Sense-Think-Act cycle takes place. The modules that depend directly from the Main Module are the WorldModel Module, the Decision Module and the Trainer Module. The function of each module is briefly described below:

- WorldModel Module: Maintains all the relevant information of the current state and a history of past states and executed actions. Relevant information could be positions and velocities of the players and the ball and statistical information of the game.
- Decision Module: Analizes the current situation on the soccer field, taking into account world model information and decides the behavior to be executed. It is supported by the Skills Module and the Positioning Module.
- Trainer Module: Uses the Main Module to communicate with the server by the monitor port and it is used to test scenarios, understand the soccer server characteristics and learn abilities.

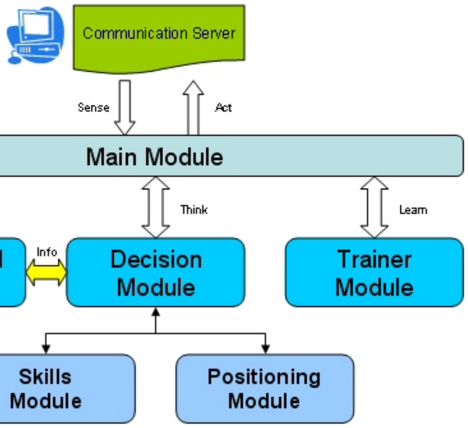

Fig. 1. Borregos3D Agent Architecture

3 Fuzzy-Bayesian Decision Making

Several approaches have been proposed for the decision making process. In our design we propose the innovative use of a Fuzzy Naive Bayesian algorithm in the RoboCup Soccer Simulation 3D domain.

The well-known Naive Bayes classifier works pretty well on a large set of problems, besides its condicional independence assumption is not met in most cases. In a Naive Bayes classifier the attributes or features are discrete, but in simulation 3D the variables can be considered continuous, because their domain encompasses the range of real numbers. Thus we fuzzify the variables and apply a method proposed in [3], consisting of a hybrid classifier bringing together Fuzzy Set Theory and a Naive Bayes classifier, named the *Fuzzy Naive Bayes classifier*

$$
\begin{gathered}
FNBclassify(a) = argmax_{c \in C} \\
P(c) \sum_{x_1 \in X_1} P(x_1|c)\mu_{x_1} \sum_{x_2 \in X_2} P(x_2|c)\mu_{x_2} \; ... \sum_{x_n \in X_n} P(x_n|c)\mu_{x_n}
\end{gathered}
\tag{1}
$$

where $\mu_{x_i} \in [0,1]$ is a membership fuction or degree of truth of attribute $x_i \in X_i$ in a new example a. We sent a paper describing our approach to the RoboCup International Symposium 2006 [4].

4 Skills

The server offers two primitive commands from which we must establish higher-level skills. The Decision Module will execute one of these high-level skills after the thinking process. A diagram of the agent skills is presented as a layered-design in figure 2.

4.1 PositionToKick

There are two main differences between the 2D to the 3D environment: objects have volume, so the ball cannot go through them, and there is no x-y plane angle parameter in the Kick primitive command. The direction of the force vector is radial to the ball from the agent's view point. To kick the ball to a certain position, the agent must lie in the *kickrange* area around the ball and must stand behind the ball in the line that extends from the ball to the desired destination point.

Said that, it is obvious that there are situations where we have to go around the ball to correctly positioning for kicking. We call this behavior *PositionToKick*. First, we estimate if the agent can run directly towards the kick point without colliding with the ball. If not, we calculate the best point around the ball to run to in the current timestep, so the agent will minimize its distance to the kick point. Using vector analysis we got an equation for calculating the path that minimizes the distance to the kick point avoiding collisions with the ball.

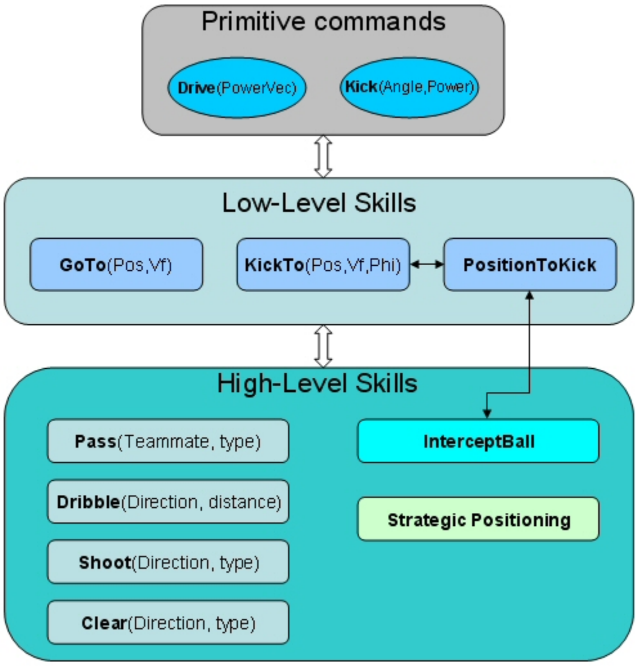

Fig. 2. Borregos3D Agent Skills

4.2 KickTo

The KickTo skill is based on the Kick primitive command and the PositionToKick low-level skill. The aim of this skill is to provide the agent with a way to kick the ball to a certain position with a certain final speed. In our current implementation, the elevation angle is zero, meaning that the KickTo skill just considers ground-level kicks.

We used multiple regression to obtain a linear equation $z = a + bx + cy$ where x is the distance that we wish the ball to travel, y is the final speed of the ball after arriving to the destination point and z is the resulting kick power to be applied. Obviusly, we assure the condition $z \in [0, 100]$, because the server does not understand values outside that range.

The best fitting curve for z has the least square error, given a traning set $\{(x_1, y_1, z_1)...(x_n, y_n, z_n)\}$ where $n \geq 3$, as shown in equation (2).

$$\Pi = \sum_{i=1}^{n} [z_i - (a + bx_i + cy_i)]^2 = min \qquad (2)$$

Under this assumption, the unknown coefficients a,b and c can be calculated making their derivatives equal to zero. Solving those equations we get:

$$\sum_{i=1}^{n} z_i = an + b\sum_{i=1}^{n} x_i + c\sum_{i=1}^{n} y_i \tag{3}$$

$$\sum_{i=1}^{n} x_i z_i = a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} {x_i}^2 + c\sum_{i=1}^{n} x_i y_i \tag{4}$$

$$\sum_{i=1}^{n} y_i z_i = a\sum_{i=1}^{n} y_i + b\sum_{i=1}^{n} x_i y_i + c\sum_{i=1}^{n} {y_i}^2 \tag{5}$$

For calculating the values for the sums we set a scenario with just one agent and the ball. When the episode e_i starts, the agent stands behind the ball and executes a kick with a power p_i and an elevation angle of zero. The agent records the value of the distance traveled and the current speed of the ball for each timestep while the ball is moving. Once the ball stops, the episode ends and the scenario is restarted, but this time with $p_{i+1} = p_i - \Delta p$. We start at $p = 100$ and decrease to $p = 0$ with intervals of $\Delta p = 0.1$. With 1000 power levels and 200 timesteps/episode we got $n \approx 200000$ examples.

For solving the equations, we represented the above set of linear equations as a matrix system

$$\begin{pmatrix} n & \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} {x_i}^2 & \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} y_i & \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} {y_i}^2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} z_i \\ \sum_{i=1}^{n} x_i z_i \\ \sum_{i=1}^{n} y_i z_i \end{pmatrix} \tag{6}$$

which we solved for coefficients a,b and c using the Gauss-Jordan Elimination method.

4.3 Pass, Dribble and Shoot

For these three skills we use the Fuzzy Naive Bayesian classifier. First of all, we set a training scenario from which we will get the data for learning the probabilities of the model. The considered features to extract from an episode depend on the skill being learned. For example, in the pass skill we set a scenario with just a passer, a receiver teammate and an opponent. We considered five variables: distance to ball, distance to teammate, distance to opponent, alignment angle and angle between teammate and opponent from the ball's perspective. We ran 1000 episodes and used that data to learn the probabilities for the classifier. In a game, the classifier is used to evaluate the probability of success of a pass and it is combined with utilities to decide which teammate is the best receiver. Dribble and shoot skills work in a similar way.

5 Positioning

We are using a partially implemented SBSP algorithm [5]. Currently we have a fixed 4-3-3 formation with defined home positions and ball attraction parameters for each player. We don't use tactics nor dynamic role exchange. We are handling four different player roles: goalkeeper, defender, midfielder and forward.

6 Future Work

There is still a lot of work to do. The Fuzzy Naive Bayes classifier actually uses exact inference and it is a little time consuming. We will improve its performance by eliminating repeated calculations using an algorithm like variable elimination or using approximate inference with likelihood weighting or Markov chain Monte Carlo methods. We plan to reuse the algorithms of Borregos 2D for online planning [6] and for opponent modelling [7]. We will be working in parallel improving the low level skills.

Acknowledgements

This work was supported in part by the research grant CAT011 and the Center for Intelligent Systems at Monterrey Institute of Technology (ITESM), Mexico. Also, we would like to thank Rujia Liu for liberating the TsinghuAeolus 3D 2005 source code. It was really helpful.

References

1. Ramón Brena, Leonardo Garrido, César Marín, Emmanuel Martínez, Mario Ochoa and Lourdes Peña: Towards a First Class Multiagent Soccer Team. 35th Mexican Congress for Technology Reseach and Development (CIDTEC), (2005)
2. Yao Hengshuai, Liu Rujia, Li Yiming, Yang Fan and Sun Zengqi: Team Description of TsinghuAeolus3D, in RoboCup Symposium 2005, Osaka, Japan.
3. Hans-Peter Störr: A compact fuzzy extension of the Naive Bayesian classification algorithm. In Proceedings InTech/VJFuzzy, (2002) 172–177
4. Carlos Bustamante, Leonardo Garrido and Rogelio Soto: Fuzzy Naive Bayesian Classification in RoboSoccer 3D: A hybrid approach to decision making. Submitted to RoboCup Symposium 2006, Bremen, Germany.
5. Luis P. Reis, Nuno Lau, Eugénio C. Oliveira: Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents. Balancing Reactivity and Social Deliberation in Multi-Agent Systems, Markus Hannebauer, Jan Wendler, Enrico Pagello, editors, LNCS 2103, Springer Verlag, (2001) 175-197
6. Ramón Brena and Emmanuel Martínez: Maximizing Future Options: An On-line Real-time planning method. In Proceedings of the 4th Mexican International Conference on Artificial Intelligence (MICAI), (2005)
7. César Marín, Lourdes Peña and Leonardo Garrido: Dynamic Adaptive Opponent Modeling: Predicting Opponent Motion while Playing Soccer. Autonomous Agents and MultiAgent Systems Conference (AAMAS), (2005)