ITAndroids 2D Soccer Simulation Team Description 2013

Fábio Mello, Luiz Ramos, Marcos Maximo, Rodrigo Roim, and Victor Moura

Technological Institute of Aeronautics,
São José dos Campos, São Paulo, Brazil
fabio.mello50@yahoo.com.br
{luizmramos,maximo.marcos,roim.rodrigo,vicrmoura}@gmail.com
itandroids-soccer2d@googlegroups.com
http://www.itandroids.com/

Abstract. ITAndroids 2D Soccer Simulation team was reestablished in mid-2011 by undergraduate students at Technological Institute of Aeronautics. In the past, ITAndroids was a successful robotics competition group, winning several competitions in Brazil and Latin America. Unfortunately, the team dismantled and the expertise was lost over years of inactivity. Now, the team is recovering fast and has already placed 10th in RoboCup Mexico 2012, the best position ever achieved by a brazilian soccer simulation team, and 1st in Latin American Robotics Competition 2012. This paper describes our lastest developments, including use of optimization techniques to optimize a state evaluation function, a neural network to switch between offensive and deffensive strategy, hand-coded heuristics to improve critical areas of the team, a system to detect opponent formation and a log analyzer tool. Moreover, we discuss our plans for future development.

1 Introduction

ITAndroids was created in 2005 by Jackson Paul Matsuura, who was a graduate student at Technological Institute of Aeronautics (ITA) at the time. The group rapidly consolidated itself in Brazil and Latin American, winning several competitions, including brazilian versions of 2D and 3D Soccer Simulation leagues. However, due to a personal decision of Matsuura to focus his efforts on building up the whole brazilian robotics competitions scenario, the team faded with time and dismantled.

By mid-2011, a group of undergraduate students reestablished ITAndroids 2D Soccer Simulation team, intending to participate in Brazilian Robotics Competition (CBR) 2011. Unfortunately, due to the short development time and lack of experience, the team placed last in CBR 2011.

With the experience acquired in CBR 2011 and more development time, ITAndroids 2D was able to qualify for RoboCup Mexico 2012, where it placed 10th, the best position ever achieved by a brazilian soccer simulation team in

RoboCup. Later, ITAndroids 2D participated in Latin American Robotics Competition (LARC) 2012, where the best teams from CBR 2011 also participated, and placed 1st. This last result made clear how much the team evolved in only one year.

Since the beginning, our progress was largely supported by the RoboCup community. Our current code is based on agent2d [9] and a large amount of our work was focused on improving mechanisms already present in agent2d framework. Also, many of our ideas were inspired by other teams work, specially HELIOS [1] and Nemesis [2].

This paper describes our development efforts, focusing on the ones since last RoboCup. Sec. 2 explains our most recent attempts in improving agent2d's action chain mechanism [1]. Sec. 3 presents improvements to defense achieved by hand-coded heuristics. Sec. 4 comments about a new stamina model. Sec. 5 describes a system to detect opponent formation. Sec. 6 shows a log analyzer tool we developed. Finally, Sec. 7 concludes this paper and shares our vision for future implementation.

2 Improvements to Action Chain Search Framework

Agent2d has a built-in framework to search for a sequence of actions (action chain) online as described in [1]. In our last year's TDP [4], we described our work in improving this framework, which consisted mainly in adding new features to the sample action chain evaluator and using optimization techniques, in specific Particle Swarm Optimization (PSO) [6], to optimize the weights given to each feature.

Continuing on this track, we added a new feature calculated as following: take straight lines from the player to all teammates; then, for each line compute the distance from it to the nearest opponent; finally, sum all the distances computed. After optimizing with this feature on, we observed that the player holding the ball was positioning itself better for pass. On the other hand, the feature was also making the team less offensive, spoiling its attack.

2.1 Choosing between an offensive or a deffensive strategy

Later, we developed a neural network to predict the probability of scoring a goal given a game state (considering the positions of all players and the ball). This mechanism was then used to choose between deactivate (defensive strategy) or activate (offensive) the described feature when the probability of scoring a goal was low or high, respectively. After manually tweaking the thresholds, the team with the neural network showed a better performance.

2.2 Dividing the attack in subtasks

Moving further in that direction, we assumed that the agents adaptability and the optimization procedure would benefit from dividing the attack in subtasks.

Our idea was to let the agent to think about a shorter term subtask (e.g. take the ball away from our deffensive area or move the ball to the opponent side of the field) instead of always thinking about scoring a goal.

A particular way of doing this division in subtasks is to think in terms of which region of the field the ball is. Thus, we divided the field in 8 regions and optimized a different action chain evaluator for each one. With this approach, it was possible to achieve good results training against specific teams, however our previous strategy that uses a neural network to do a simple strategy switch achieves an overall better robustness against differences on the defense policy of each team. Nevertheless, we still believe that a more complex way of choosing different strategies depending on game situation can led to better team performance and we expect to do more experimentation in this direction.

3 Defense Heuristics

In order to improve the team's defense we preferred to use heuristics based on statistics and domain knowledge instead of formal AI.

Basically the defender has to decide whether or not to tighten his marking. The advantage of doing that is to decrease the opponent's space and create more chances to tackle. The problem arises when the forward gets to dribble around the defender.

To make this decision properly we analyzed a lot of parameters including the position of the ball in the field, the danger of the play, the position of the offside trap, the distance between the defender and the ball, etc.

Every time a new set of conditions were added to the decision-making process, the team was tested against its old version. The tests consisted of 50 matches. After that it was possible to decide if the modifications were worth it by plotting the mean goal difference against the number of matches. This plot usually converges when the team is actually improving.

After a lot of hard working the team was able to maintain an average goal balance of 0.5 in 100 matches against its version without the modifications, which is a pretty good result.

4 New Stamina Model

The default agent2d stamina model is too simple. We analyzed this model and improved it including new conditions, specially for the defenders (that used to save needlessly too much stamina). The new model allows defenders to dash in dangerous situations (preventing through balls for example) without compromising their stamina capacity.

5 Opponent Formation Detection

A leason we learned during RoboCup 2012 was that understanding how the opponent plays is key to win. Therefore, we decided to develop a system to do opponent analysis. We considered formation to be the easiest opponent feature to analyze objectively, thus we decided to begin by focusing on it. Also, we found some work done about opponent formation modeling [1, 3], which inspired heavily our development.

Our first assumption was that the opponent players positions depend solely on ball position. This may seem like an oversimplified approach, however we believe that this simplification is reasonable given how the positioning systems of most of the teams in the league work.

We decided that the best approach was to model many team formations offline, then use our online coach to determine which known formation model was the closest to the opponent formation. We decided to develop a log analyzer tool, so we may analyze a team even we do not have its binary. The tool created extracts formation samples, which consist of the players positions for the given ball position, and stores a "model" of a formation as a list of formation samples.

To avoid runtime performance issues, the number of samples per formation must be small. Therefore, a method to select the most relevant samples is needed. As a first try, we are simply selecting a certain number of samples at random, but we expect to try some clustering algorithms, such as K-Means [7].

To detect which known formation is the best match to the opponent formation, we developed the following algorithm: given a cycle, for each formation model, extract the nearest neighbor to the current game situation in terms of ball position; then, for each player (except for the goalie) compute the distance (displacement) between its current position and the position it was in the formation sample; next, sum all distances computed and accumulate this value in a variable for each formation model; finally, chose the formation model with the smallest accumulated sum of displacements.

Given that we know the opponent formation, we still have to use this information effectively. We have two major ideas in mind. The first one is to have many action chain evaluators and select which one to use depending on the formation detected. The second idea is to use the formation model to predict the opponent players positions and use them as input to the action chain framework.

To test this system, we did an experiment using 5 teams, where 6000 formation samples from one team were compared against 6000 from the other. Table 1 summarizes the results (displacements shown are per player) using agent2d's formation model.

Table 1. Formation detection system results using a agent2d's formation model.

Team	Mean Disp.	Min. Disp.	Max. Disp.
UaiSoccer2D 2012	4.630	0.003	48.32
agent2d	4.800	0.005	58.70
ITAndroids LARC 2012	5.593	0.004	49.34
Axiom 2012	7.833	0.001	49.77
MarliK 2011	10.24	0.008	86.72

Note that the results are qualitatively consistent to what one would expect from watching the logs. Curiously, UaiSoccer2D 2012 was closer to agent2d than agent2d itself. In fact, both teams have very similar formations, if not the same.

Despite the results suggest a good detection performance, we consider that they are still too poor for prediction. Therefore, we are trying to improve our system. Currently, we are experimenting with different clustering algorithms to select a better list of samples for the model, as mentioned before. Also, we expect to change the nearest neighbor sample selection to a Delaunay Triangulation interpolation, similar to the interpolation agent2d uses for positioning [1].

6 Log Analyzer Tool

Initially, we developed a log analyzer tool to solely support our formation detection system, as described in the previous section. Later, developing a GUI for the tool to help us debug, we realized that the tool itself had potential for other uses. For example, drawing all ball positions from a given match, we could visually see the paths usually followed by a team or where it tends to stay with the ball (see Fig. 1). We believe that with this kind of visual tool, a human can analyze logs much faster and precisely than having to watch the whole match.

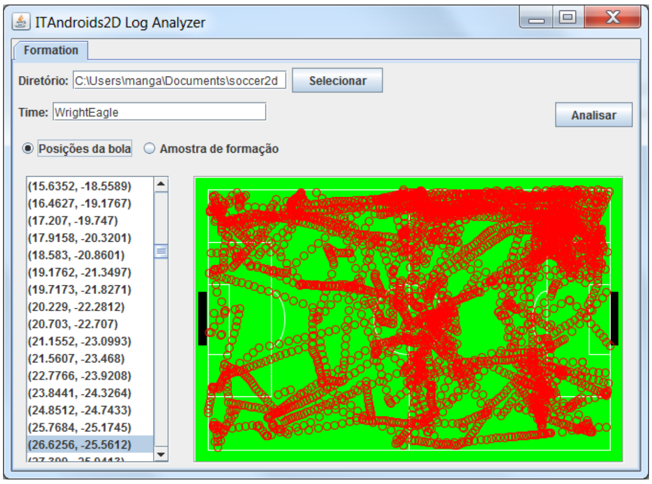

Fig. 1. ITAndroids 2D log analyzer tool.

7 Conclusions and Future Work

This paper presented the most recent efforts of team ITAndroids Soccer 2D. Despite being reestablished in mid-2011, the team has already placed 10th in RoboCup Mexico 2012 and 1st in Latin American Robotics Competition 2012.

Our current code is based on agent2d. We have been experimenting with many formal AI techniques: heuristic-guided search, Particle Swarm Optimization [6], Neural Networks, K-Means Clustering [7] etc. Also, there is a lot of improvement in our team achieved by specific hand-coded heuristics.

Currently, our efforts are focused on: improving action chain performance by building better action chain evaluators and generators, developing a log analyzer tool, creating a opponent formation detection system, doing opponent positioning prediction and improving our goalie against specific strategies. We expect to have these modifications done and integrated in our team until the competition.

Acknowledgements

We would like to acknowledge the RoboCup community for sharing their developments and ideas. Specially, we would like to acknowledge Hidehisa Akiyama for agent2d [9], librcsc [10], soccerwindow2 [11] and fedit2 [12] softwares. Also, we would like to acknowledge Luís Paulo Reis for his PhD thesis [8] and UvA Trilearn [13] for their base code; both played a great role in introducing our team to the league.

References

1. Akiyama, H., Shimora, H.: HELIOS2010 Team Description (2010)
2. Norouzitallab, M., Javari, A., Noroozi, A., Salehizadeh, S.M.A., Meshgi, K.: Nemesis Team Description 2010 (2010)
3. Bai, A., Zhang, H., Lu, G., Jiang, M., Chen, X.: WrightEagle 2D Soccer Simulation Team Description 2012 (2012)
4. Mello, F., Ramos, L., Maximo, M., Ferreira, R., Moura, V.: ITAndroids 2D Team Description 2012 (2012)
5. Lau, N., Reis, L.P., Mota, L., Almeida, F.: FC Portugal 2D Simulation: Team Description Paper (2011)
6. Eberhart, R.C.; Kennedy, J.: A New Optimizer Using Particle Swarm Theory. In: Sixth International Symposium on Micro Machine and Human Science (MHS'95). Nagoya, Japan (1995)
7. Alsabti, K., Ranka, S., Singh, V.: An Efficient K-Means Clustering Algorithm. Electrical Engineering and Computer Science, Paper 43 (1997)
8. Reis, L.P.: Coordenação em Sistemas Multi-Agente: Aplicações na Gestão Universitária e Futebol Robótico, PhD Thesis, FEUP (2003)
9. agent2d-3.1.1, 2012, online, avaliable at: `http://pt.sourceforge.jp/projects/rctools/downloads/51943/agent2d-3.1.0.tar.gz/`, consulted on February 2012.
10. librcsc-4.1.0, 2011, online, avaliable at: `http://pt.sourceforge.jp/projects/rctools/downloads/51941/librcsc-4.1.0.tar.gz/`, consulted on February 2012.
11. soccerwindow2-5.1.0, 2011, online, avaliable at: `http://pt.sourceforge.jp/projects/rctools/downloads/51942/soccerwindow2-5.1.0.tar.gz/`, consulted on February 2012.
12. fedit2-0.0.0, 2010, online, avaliable at: `http://pt.sourceforge.jp/projects/rctools/downloads/48791/fedit2-0.0.0.tar.gz/`, consulted on February 2012.
13. UvA Trilearn 2003, 2005, online, avaliable at: `http://staff.science.uva.nl/~jellekok/RoboCup/2003/index_en.html`, consulted on February 2012.