

Nexus2D Team Description Paper

Mahdi Abolfazli Esfahani, Mohsen Ghafouri, Masoud Jamili

Saeid Askari, Reza Etemadi, Hossein Mohebbi, Mohammad Dara, Masoud Erafani, Morteza Malekabadi

Nexus Simulation and Modeling Lab, Ferdowsi University of Mashhd, Iran

Nexus@um.ac.ir

Abstract. Improving our previous skills was our main concern. Advancement in blocking using Genetic Algorithm is one of these improvements which is explained bellow. Furthermore, selecting best action according to the current state is one of the other important issues which we used reinforcement learning for this. A customized log-player tool developed by our team to help visualized debugging for different situations is also explained.

1. Introduction

2D soccer simulation is one of the Robocup leagues which intends to eliminate areas concerning construction of a robot and its difficulties and instead to focus on cooperation between agents and other high-level issues. This paper is based on 2D soccer simulation server which is a physical simulation system. This server is a multi-agent system and has a dynamic environment [1]. Using simple and static methods is inefficient in a dynamic environment. Therefore, it is a good idea to use dynamic algorithms and methods in such an environment. In the following, these algorithms and methods are explained.

Nexus team was officially started in 2002, participating in different national and worldwide Robocup events. Current team members have also participated in national and international competitions as *Robotoos*, *051*, *Tempux*, and also *Nexus* teams. Our team is based on Agent-2D base code. *Nexus* achieved the first place in IranOpen2015 and third place in RoboCup IranOpen 2014 and 2013 international competitions.

2. Reinforcement Learning (RL) for decision making in penalty area

Due to the high potential of score a goal, decision making in penalty area is much more important compared to other areas. We provided an appropriate platform for test and improve the quality of decision making in this area.

2.1. Defining the problem

For action formulation, we know that we should choose an action from pass to each teammate, dribble or shoot. Earlier, we had some process to generate different type of passes, dribbles and shoots. In here, we use these process to determine the best pass to each teammate, best possible dribble and also best shoot; therefore, with RL we want to learn the best action to do in each state.

At first, we summarize the whole WorldModel information with state declaration by some factors which shown in below

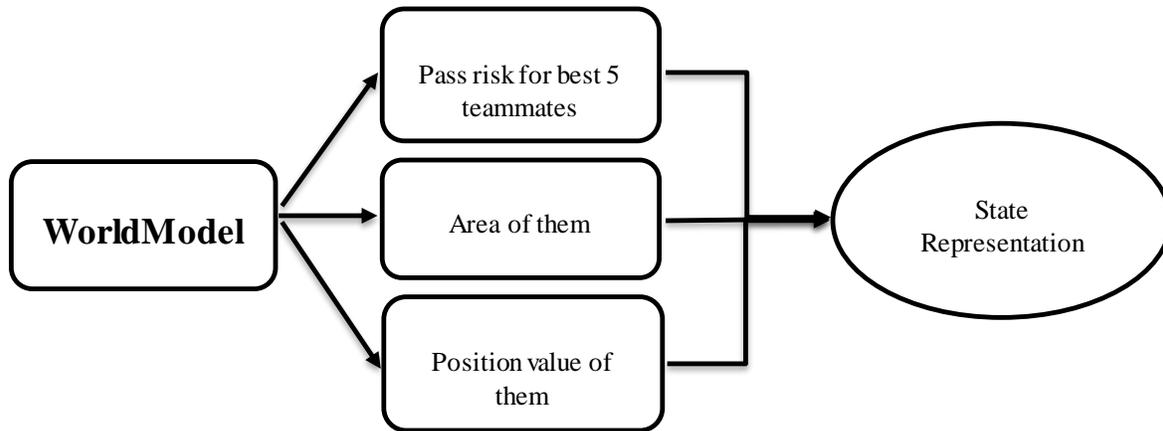


Figure 1- Summarize World Model with states declaration

2.2. Solving the problem using Reinforcement Learning

The task of reinforcement learning is to use rewards to learn a successful agent function [2]. We set reward for score a goal, transit the ball to a better place and put punishment for losing the ball. With Q-Learning algorithm, our agents calculate the function of $Q(s,a)$ which shows the quality of doing action 'a' in state 's' during offline learning. After that, we use the result of learning in penalty area in our games online.

3. Genetic Algorithm for blocking

3.1. Defining the Problem

Player blocking is performed to prevent the opponent player that carrying the ball from getting close and threatening our goal. This action is done by finding a position for our player to block the opponent player and prevent him to move forward.

We defined some factors that change blocking procedure. For example, if our player can intercept the ball in a time less than X, he goes to intercept the ball. All of the factors would affect the performance of blocking procedure.

3.2. Solving the problem using genetic algorithm

Genetic algorithm (GA) is a useful algorithm for optimizing problems, and our objective is to find sixteen optimized coefficients. Therefore, it sounds a good choice to use GA in this case. [5]

We test 10 different situations with running 10 different type of players. The opponent player attack and try to score a goal and our player should block him. The opponent player may be different for each test and may picked up from HELIOS or WrightEagle or YuShan.

We defined some scores for our test that determine the TotalScore for the test. For example, scoring a goal by opponent player has 1000 points. Duration of blocking has a score equal to number of cycles. Also Dribbling our player has some scores and etc. All of these scores affect the TotalScore. The lower this TotalScore is, the better. So, it's tried to obtain the coefficients in a way we have the least TotalScore using GA.

First, we start with a hundred randomly generated genes. To produce the next generation, fifty best genes are chosen from the previous generation. Then, fifty genes are produced by mating pairs of the fifty best. Thirty of them go into the next generation directly, but we make mutation in twenty remaining and form the next twenty genes for next generation.

Figure 2 illustrates the production of the next generation.

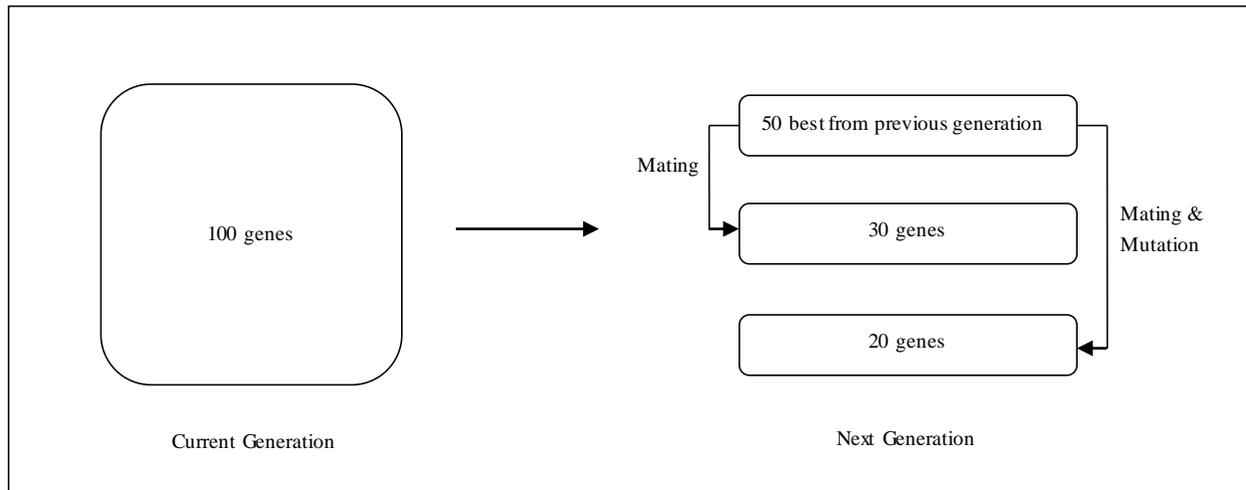


Figure 2- The production of next generation

4. Customized log player tool

4.1. Defining the problem

Different skills in 2D Soccer Simulation depend on variety of factors and there usually is a score-based engine for both making decisions and performing the actions. Multiplicity of the factors and having many options makes program debugging very hard. Normally it's usual to print the data into text files for watching the efficiency of different parameters and generally for debugging purposes. Finding the problems and improvement based on these text files is sometimes very hard. Besides, having an imagination of some text data like positions and angles is much more difficult than seeing it in real server-like environment.

4.2. Tool development

We have developed a tool that allows us to visualize information and data. It's done by making some changes in the source of "rcsslogplayer"[7] and customizing it.

Agents save considered data of every cycle in a "XML" file. At the time of playing the game logs, the program mixes data files created by different agents for different cycles. As the result, the desired information would be shown in the logplayer at suitable cycles.

4.3. Usage example

This customized logplayer helps us in improvement of many skills. As an example, the usage of this tool in improving the passing skill is explained. One good way for passing is generating different passes, evaluating them and choosing the best one (Our passing algorithm will be described in next part by details). For the evaluation part, the agent draws a small circle for every created pass in the appropriate point of field and attaches other desired data to the point. So, different passing options can easily be seen alongside each other and be compared by details. This tool is released on 15 February 2014[8].

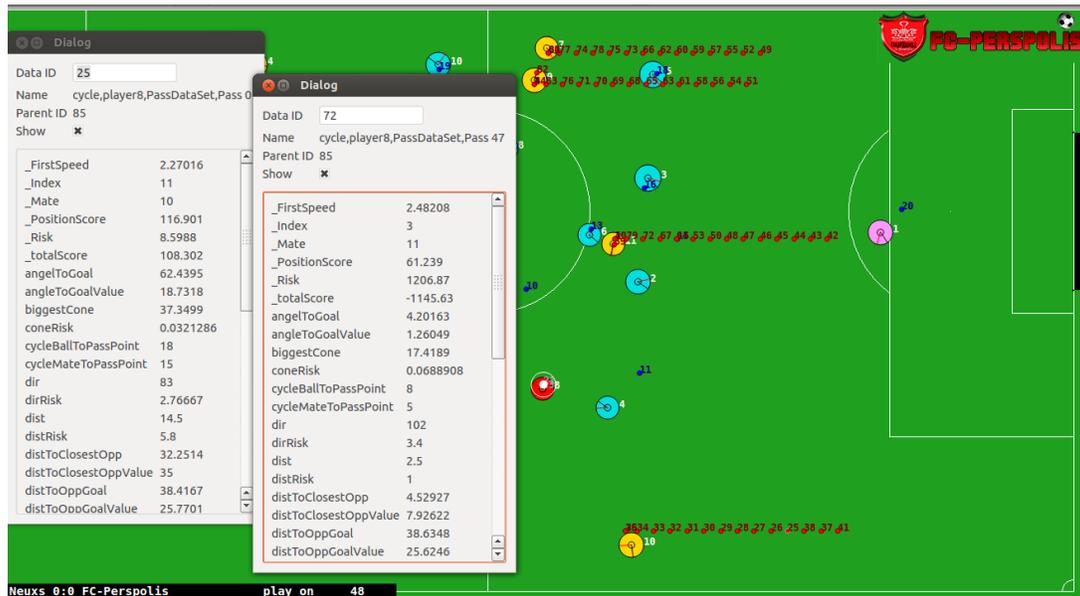


Figure 2. Customized log player tool

5. CCS Tool Added To rcssserver2D

One of the most important things in agent simulating like Soccer Robots is to find out our actions effect in physical environments which 2d soccer server makes it. Nowadays this can be done by using rcsslogplayer tool. This gets the offline log files and shows cycle by cycle game. Thus, teams should log their actions to a file and find out what is the actions problem in rcsslogplayer, which needs another tool to install near the server. We developed a tool for solving these problems, Cycle by Cycle Simulator (CCS), this simulator allows users to observe their action effects online in both standard and frameview of rcssmonitor. So, the cycle by cycle simulator tool is added to server and monitors due to the following advantages [6]:

1. Seeing our actions effect on physical environment simulated by soccer server cycle by cycle;
2. Finding suitable fitness function for optimization algorithms;
3. Making precise debugging actions online;
4. Seeing our ranking functions effect;
5. Viewing our online agent logs cycle by cycle;
6. Stopping the game for some instant to do some issues;
7. Changing the mode of game by online coach inside the game to get better results for their neural network and optimal algorithms;
8. Making better low-level executable actions.

6. References

1. Mao Chen, Klaus Dorer, Ehsan Foroughi, Fredrik Heintz, ZhanXiang Huang, Spiros Kapetanakis, Kostas Kostiadis, Johan Kummeneje, Jan Murray, Itsuki Noda, Oliver Obst, Pat Riley, Timo Steffens, Yi Wang, Xiang Yin, *Robocup Soccer Server: User Manual*, 2003
2. Frieder Stolzenburg, Oliver Obst, Jan Murry, *Qualitative Velocity and ball Interception*, 2002
3. Karen A. Kopecky, *Root-Finding*, 2007.
4. Li-Xin Wang, *A Course In Fuzzy Systems and Control*, New Jersey: Prentice Hall, 1996.
5. Stuart J. Russell, Peter Norving, *Artificial Intelligence: A Modern Approach*, 2nd edition, Practice Hall.
6. http://sourceforge.net/projects/ccsrcsserver2d/files/CCSServer_Manual.pdf/download
7. <http://sourceforge.net/projects/sserver/files/rcsslogplayer>
8. <http://sourceforge.net/projects/logplayerplusdataviewer/files/>
9. Nexus2D Soccer Simulation Team Description Paper, Mahdi Abolfazli, Masoud Jamili, Mostafa Sayahi, Saeid Askari, Mehrnaz Ghashgaei, Hadi Kalamat and Sirius Shahini, Robocup2013