

# ABOL Soccer Simulation Team Description

Mahdi Abolfazli Esfahani

Nanyang Technological University, Singapore

[Mahdi001@e.ntu.edu.sg](mailto:Mahdi001@e.ntu.edu.sg)

**Abstract.** This paper presents an overview of ABOL 2D Soccer Simulation Team. We used evolutionary based machine learning algorithms for agent's behaviors and are currently working on them for completing and improvement the actions of agents such as Shoot, Pass, Mark and etc. We worked to find best algorithms and use them in our source code. Agent2D base code is used as the based code of this team.

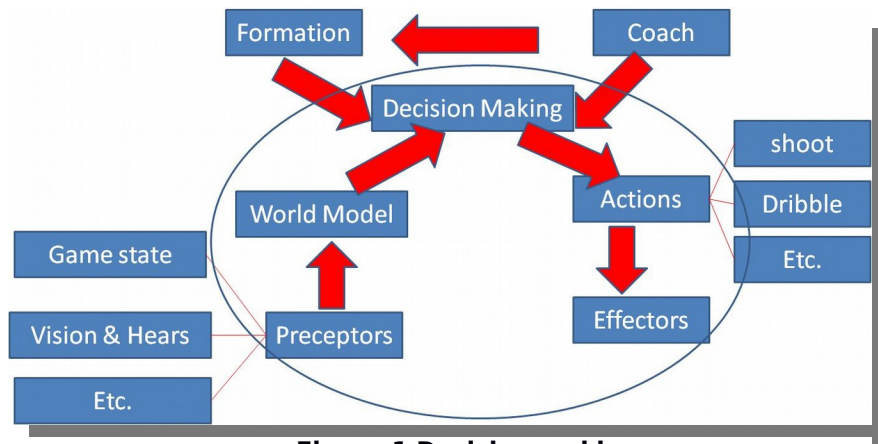
**Keywords:** Robocup, multi-agent, server, Algorithms, dummy brain

## 1. Introduction

In 2D soccer Simulation League each team have 11 players that connect to the server, the players receive some information such as distance to flags, other player, ball and etc, from server. Then the Player analyzes the information and thinks with attention to the information to find a good action to do. We are one of the teams that work in this way to simulate robots (Players). We used Agent2D Base code for our team. I was as a part of Nexus Simulation team at Ferdowsi University of Mashhad, and I have moved to NTU Singapore to continue my study. Now, we are planning to enhance the NTU simulation team and so it is the first time NTU participates in an international RoboCup 2D competition.

## 2. Decision making

In the previous section, we said that the players have the ability to receive information from server about environment and also about its own status. This information is used inside of our decision making machine to find and chose the best action. The action that find is that will send to server to do. The preceptors are the information's from the server such as game state, Vision & Hear and etc. these are the world model that we send to Decision Making function and also input of Decision making Function, then we get help from a coach, and from coach we get best Formation game and etc, then in Decision Making Function we find best action to do, now take it from actions, send it to server, that is the Effectors and robot do the action that found from Diction Making that is output of Decision Making function as shown in Fig. 1.

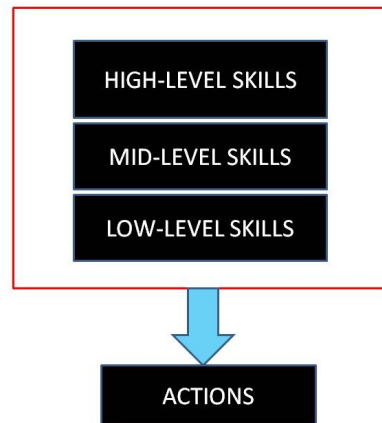


**Figure 1 Decision making process**

There are too many ways to find best action at each instant of time for agent. So that agent with attention to the own ability and priority of actions will find best action to do now. Also we recognize that the current action is finished before to do next action. We put it due to the equilibrium don't refuse.

### 3. Actions

Actions are based on the skills. Skills are divided to three different types: low-level, mid-level and high-level skills, as shown in Fig. 2 and explain briefly below [1].



**Figure 2 skills level that will give actions**

#### 3.1 Low-Level Skills

Low-Level actions are actions that need a very little calculate and they are the basic actions for the agent. For example kicking ball, turning, turning neck and etc.

### 3.2 Mid-Level Skills

In this level the skills of Low-Level Actions mix and we get a new action. For example going to point that we should do two actions: turn and dash (walk), another example is kick-to that player should kick and also turn neck and some times turn body for more kick speed.

### 3.3 High-Level Skills

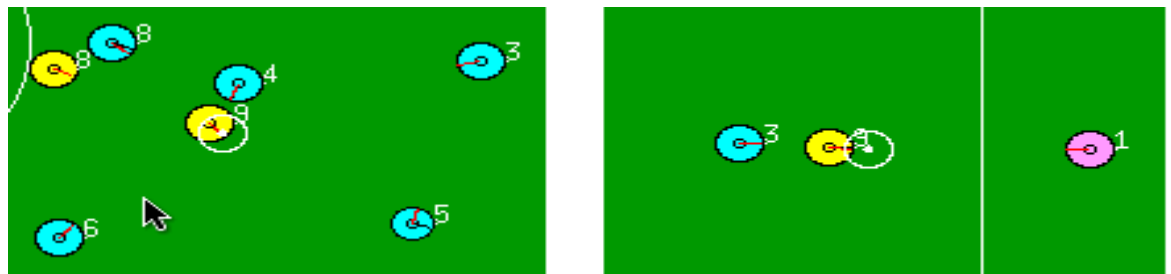
These actions are actions that need more calculate and they make with pay attention to Mid-Level Actions such as: Dribble, Pass, Positioning, Block, Marking, Intercept and etc.

#### 3.3.1 Dribbling

Dribble is one of the most important actions that if player do it good and in the best time can flee from opponents and go to the opponent's goal fast.

The algorithm that we used for Dribbling depends on opponent's position and it run difficult:

1. Is Opponent at front of our player or not if it's true it use turning the ball around body or going to a good angle that the opponent player can't reach the ball.
2. Is opponent in back of our player (distance from our player is important) that we use a fast dribble.
3. In the near of their goal we use an smart dribble to keep the ball(use best kick speed).



**Figure 3 in the right picture we see that the closest player is at back and our player is dribbling fast and in the left Picture we see the player dribbling the front player.**

#### 3.3.2 Pass

Pass is one of the other important actions for moving ball from one place of field to another place that if do it good we can find a free place for dribbling and etc. to do this action well we should find the best player that we used an scoring model that we found by thinking about it we see the number of players around the receiver and also distance to the opponents goalie and the X and Y on the field.

To see that any opponent can reach the ball when we pass we used a following algorithm:

1. We see that how much cycles is need to ball get to the receiver.(=ball Reach Cycle)

2. We make line from 2 points, ball and player. Then we find the closest point on this line to each opponents player and See that how much is remaining that to player to go to that point.(=Player Reach Cycle)
3. Then at last we see for each opponent that if Player Reach Cycle is less than ball Reach Cycle its meaning that any player can get the ball and pass is not successful but if ball Reach Cycle is less than player Reach Cycle the pass is successful and we can pass the ball to point.

### 3.3.3 Shoot

The most important thing in 2D Soccer Simulation League is shooting the ball to goal. That we should shoot the ball to a point that the opponent goalie can't reach the ball and the ball go to the opponent goal and we win. For finding best position to kick we used the following conditions:

1. The closet point between the ball and the opponent's goal.
2. Can the goalie reach the ball or not that we used the algorithm that we used in pass.
3. The most open angel (degree) between ball and the opponent's goal.
4. The number of players around ball.

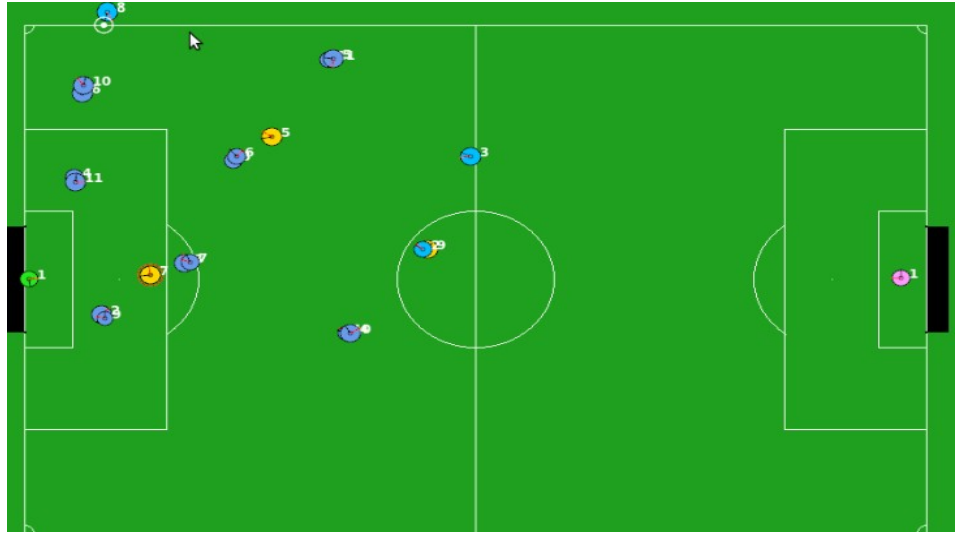


**Figure 4 kicking the ball to the best point**

### 3.3.4 Mark

When the game is not play on and the ball is for opponents or the ball is in the danger area (near our goal) we start marking opponent's player. We find the closest players to the ball that opponents can pass to it and we find best position to marking it. There are three different points to go:

1. Point between player and ball
2. Point between player and goal
3. Point between player, ball and goal



**Figure 5 Marking Model**

### 3.3.5 Goal Keeper

The algorithm of our goal keeper is that if the opponent agents don't want to kick the ball, goal keeper finds the closest way to go to new position with pay attention to the time. If the opponent agents want to kick and the ball will receive to goal, the goal keeper will find the time and position of receiving ball to goal and go fast to the point. We find the pos by attention to the following things:

1. The closest distance between ball and our goal.
2. The line between ball and our goal.
3. The angel (degree) between opponent player and our goal and also ball and our goal.

After the goalie get the ball it should pass the ball to teammate player for this we find points near the goal that player can go to them, after going to the point we find the best player to pass and pass to the player.

## 4. Conclusion and Future works

Our teams do the necessary works right now, since it is in start, and we are trying to do them faster and more intelligent. Speaking between players is one of the other material things and we are working on that. Moreover, I am planning to include machine learning techniques till IranOpen2018 and that is why I am participating in this competition.

## 5. References

1. Amin Milani Fard, Vahid Salmani, Mohammad Hossein Ansari, Mahmoud Naghibzadeh, Amir Farzad, Morteza Delgir, Amin Zamiri, Hossein Kamyar, Ehsan Saboori, Farid Seifi, Mehrdad Mozafari, "Nexus Robosoccer Development Process from 2D Simulated Agents to 3D Humanoid Soccerbots", Journal of the National Technical University "Kharkiv Polytechnical Institute" (Herald of the NTU "KhPI" Scientific Works).XIII 5'2007.
2. Peters, J., Schaal, S.: Reinforcement learning of motor skills with policy gradients. Neural Networks (in press).
3. <http://server.sourceforge.com>