

# ESKILAS

## Soccer 2D Simulation Team

### Description Paper 2011

Mehrdad Bakhtiari (mehrdad.baxtiari@gmail.com)  
Mohammadreza Montazeri (monta2009@gmail.com)

Team leaders:

Sina Saharkhiz (sinas1991@gmail.com)  
Pooria Kaviani (pooria.kaviani@gmail.com)

Department of Computer Science and Information Technology  
Allame Helli High School (NODET), Iran

**Abstract.** ESKILAS is a 2D soccer simulation team that has been participating in many robocup competitions since 2008[1]. In this paper we describe our researches and efforts in our team. Some of the new strategies and algorithms will be explained.

## 1 Introduction

ESKILAS team researches begun in 2008 with the name HelliBASH. We have been participating in robocup competitions and we have achieved many successes in this years, starting 5<sup>th</sup> place in China Open 2008. Later on we earned many different places such as 2<sup>nd</sup> place in IranOpen 2009, 1<sup>st</sup> place in AUTCUP 2009 and some different places in local competition. But our main accomplishment was 7<sup>th</sup> place in RoboCup 2009 Graz and in the following year 4<sup>th</sup> place in RoboCup 2010 Singapore which was a great success. We also achieved 3<sup>rd</sup> place in both ChinaOpen 2009 and IranOpen 2010. Our team still uses the Mersad base as base code. We made many major changes since last year, so now we should explain some of these strategies in seperated sections.

## 2 Defense System

Defense system is divided into two parts. Default system is a commander-based system which is used when defenders' commander is ready to determine defenders' roles. Another system is an independent system for each defender; it includes regional mark system and guard system.

### 2.1 Commander-based system

In this system defenders pay attention to the commander and that central player determines their roles, some of them are markers and others are called guardian.

First, the commander should find dangerous opponents, then he specifies the number of markers according to the number of dangerous opponents. Next step is generating mark-table, Which is achieved by using a weighted graph matching algorithm. In this model every player is a node of the graph and between each opponent node and each teammate defender node there is an edge with a weight, formerly that weight was proportional to the two players' distance before, but now there are many factors involved in assigning weight to edges. These factors include the distance, last markers and danger ratio of each opponent. After generating mark-table, the commander defines a few defensive areas, then matches each guardian to an area. Guardians take their positions in those areas.

## 2.2 Regional mark system

Our commander is one of offenders, it has many advantages but also has a problem, sometimes after an attack, commander player can not come back to our field as quick as needed, so the defenders will have to take an individual decision. In first step each defender defines an area according to his and ball's position. They keep their position until an opponent gets in their area;at this moment the defender starts marking him. The other defenders keep guarding. If this situation continues too long and ball enters our penalty area, defenders pay attention to our goalie and he becomes the new commander and defenders do the marks and guards due to his commands. In this status goalie specifies defensive style while awaiting our main commander (Player No.11). When he reaches an acceptable position goalie says his last command and then defenders turn their attention to main commander.

## 3 Defensive Skills

### 3.1 Block

Block is a defensive skill and its goal is to cut opponent's dribble and stop the opponent, and obtain the ball's possession if its possible. Block is divided into two section, The first issue is to decide which teammate player is the best blocker at the time; Sometimes no blocker is needed and sometimes we need more than one (even more than two) blockers. After determining the blocker(s), the second issue arises, "How to block?".

**Choose blocker(s)** There are several ways for each player to compute if he is blocker or not. Defenders can find blockers according to the central strategies, which were received from defenders' commander. It has two kinds of blockers: sometimes the commander determines the blockers and inform the defenders by "say" facility. Also, we may have another blocker if opponent attacker is counted as dangerous players;This blocker is marker of the opponent. Defenders have also an individual system for finding blockers; this system is for guardians and they should block when the opponent gets in their area or his blockers are far from him.

**How to block (also known as physical blocking)** Blocking process is divided into two main parts: Closing on the opponent and Possessing the ball.

**Closing on the opponent** The purpose of Closing on the opponent is to make a good situation to start Possessing the ball stage. First, the blocker assumes a virtual target for the opponent, then he calculates the opponent's dribble route according to the opponent's position, his distance from the ball, other opponents' positions, and teammate guardians' positions. After finding the opponent's dribble route, the next step is to find the best interception point on the opponents dribble route through an algorithm similar to ball-intercept.

**Possessing the ball** Possessing the ball stage is performed in one of the following ways:

- Greedy block

The greedy block is usually used in dangerous areas. In this method, the blocker directly approaches the opponent in minimal cycles.

- Direct close block

This method is safer than greedy block and faster than side-dash-enabled block, in this method in the most of cycles the blocker directly approaches the opponent and in some cycles adjusts his path by using side dash.

- Side-dash-enabled block

In this method, when the blocker reaches the opponent's dribble route, turns and continues his moving via side-dash, so the opponent can't move easily and is forced to stop. Thus the blocker can approach the opponent carefully. When the blocker reaches the opponent, he still can continue his movement to possess the ball, but in some dangerous situations he might prefer to turn toward the opponent and tackle the ball.

If we have two or three blockers, blockers use different methods.

### 3.2 Mark

Mark consists of two stages, first is approaching the opponent and the second one is called "Physical mark".

**Closing on the opponent** When the mark target is so far from the marker, our marker should approach the opponent as fast as possible. It's like closing on the opponent in block algorithm. It's difference is in locating the opponent's virtual route.

**Physical Marking** Physical mark means take the best position to cross opponents' pass routes. The marker calculates this position according to the ball's and target opponent's positions. The distance from the target opponent depends on the situation of the target opponent, the ball and the agent himself. For instance in dangerous situations (e.g. when the ball is in penalty area), the agent prefers to stick to the opponent. In this condition the agent uses side dash to make small moves.

## 4 Dribbling

A good dribble is the one which besides having a sufficient speed, can have a good security. We achieved this important features with using side dash. That helps the agent to dribble through opponent's without turning (which reduces the dribbles speed).

### 4.1 Path Finding

One of the most important part's of a good dribble skill, is path finding. If there is no opponent in the field, the obvious path for a dribbling agent would be a direct line towards the opponent's goal. But, in a real game a dribbling agent should decide about its path so that it can maximize the probability of having good new opportunities for the team and minimizes the probability of getting trapped between opponents and thus losing the ball.

So the agent considers some directions around the default direction. Then, he compares them with each other to find the best direction. In a simple procedure that used in ESKILAS team, the agent gives a score to each direction that shows how good that direction is (for being selected as the dribble path). This score is computed by a heuristic formula that has three parameters:

```
Score(dir) = Safety(dir) * safety_weight +  
            DeltaDir(dir, default) * target_weight +  
            NoTurnNeeded(dir) * turn_wieght;
```

Where Safety(dir) is a function that regarding opponents' positions, computes their danger in the direction and concludes how safe the direction is. DeltaDir(dir, default) simply computes the absolute angle between the input directions. The more the DeltaDir(dir, default) is, the more that direction is deviated from the goal direction, and the worse that path is. NoTurnNeeded(dir) is again a simple function that is either 1 or 0. It is 0 if the agent will have to do a turn for dribbling in the dir direction; and it is 1 if the DeltaDir(dir, Body-Direction) is less than a threshold and the agent can dribble in the dir direction with no need for a turn. The third parameter encourages the agent to prefer its current path.

In our new solution for path finding, we design a model in which a trainee can learn to compare different choices. By this idea, we train a function such that, given two different choices by their three features (Safety, DeltaDir, No-TurnNeeded), it can distinguish which one is better. Actually, we are training the "<" operator to compare two choices. To train the comparator, in first step, we build a dataset that contains some cycles and best direction for dribbling which has been recognized by a human expert. Then another program reads the log file of the game and extracts other information of those cycles (opponents positions, agents position, agent body direction, and etc.). For any instance of our informations (that describes situation of the field in a specified cycle), we consider 18 directions around the dribbling agent and compute three parameters for all 18 direction choices. Then, knowing the best answer for that instance; we

add 17 (that is 18 - 1) pairs to the final dataset. The first choice of these pairs is features of the best direction of that instance, and for all other 17 directions in that instance a new pair is obtained. And all these 17 pairs belong to class 'Better' in final dataset. we also add the same 17 pairs again to the final dataset with the difference that we transpose the order of the choices in each instance and the resulting instance will belong to the class 'Worse'. Now a classifier will train our comparator with this dataset. An example of final dataset is shown below:

```
@RELATION dribble

@ATTRIBUTE firstturnvalue REAL
@ATTRIBUTE firsttargetvalue REAL
@ATTRIBUTE firstsafetyvalue REAL
@ATTRIBUTE secondturnvalue REAL
@ATTRIBUTE secondtargetvalue REAL
@ATTRIBUTE secondsafetyvalue REAL
@ATTRIBUTE class {better,worse}

@DATA
0,2.52088,10,1,160.951,2,better
1,160.951,2,0,2.52088,10,worse
0,2.52088,10,1,150.951,2,better
1,150.951,2,0,2.52088,10,worse
0,2.52088,10,1,140.951,2,better
1,140.951,2,0,2.52088,10,worse
0,2.52088,10,1,130.951,2,better
1,130.951,2,0,2.52088,10,worse
0,2.52088,10,1,120.951,2,better
1,120.951,2,0,2.52088,10,worse
...
```

After our comparator is successfully trained, for recognizing the best path direction for any unseen dribbling situation, We just need to first compute the three parameters for different direction choices in that situation; and then, just compare all pairs of choices using the comparator in a tournament structure. And finally, the choice with the maximum winning comparisons will be the answer[2].

## 5 Look Ahead Pass (LAP)

We have introduced *LAP* in our last year's TDP[3]. This method mainly implemented to decrease the communication time and improve group works. Since last year, we have improved our team's pass skills with this method and it's derivatives, especially *Forced LAP* and *Without Ball LAP*.

### 5.1 Forced LAP

The main reason of developing LAP was to be able to pass the ball to other players with higher speed and without a time difference of communication. But

we may want to execute LAP in some situations (like 'Non play on') to obtain a better situation (generally better visual information) and to inform the receiver of the pass. This ability is called "Forced LAP". Forced LAP is usually a 2 cycles long LAP.

## 5.2 Without Ball LAP

When the agent is receiving a teammate's pass (or in a more general case, intercepting the ball), he can estimate the situation of the field after the interception and doing so he can find a good pass for his next action. That would save the ball in dangerous situations. Another advantage of this method is vision refreshing; After calculating passes, agent can check the area again to refresh visual information and make changes to the selected pass. Even if new visual information shows that the selected pass can't be executed, agent can calculate new passes with lower chance of failure. Composition of this method with *Normal LAP* significantly increase the chance of safe pass receiving.

## 6 Possession Freezing

In RoboCup 2010 competition we discovered that our surprising attack policy may cause many problems in our team. One of the main problems was insufficient number of offenders. Because most of our players were defenders and they couldn't reach our offenders to help them in attacking. Another problem was players' stamina; our defense commander (central offender) and play-makers who play between defenders and offenders were losing their stamina while dashing through length of the pitch. Our solution to these problems is "Possession Freezing". In this method we keep the possession of the ball by our team to advance the ball more slowly. To achieve this goal, we made small changes in dribble skill. We decreased dribble speed in some situations and changed the dribble path finding. In a major change, we decided to reduce our breaker passes. In the field of positioning, we need a coordinated positioning that could be found by ball owner, as we explained in our last year TDP. We are now working on this special method to build a more effective offense system.

## 7 Summary

In this paper, we described our work in the field soccer 2D simulation in our team ESKILAS. We are planing to bring our new ideas in our team with new algorithms and methods in the near future.

## References

1. Eskilas soccer 2d simulation official site <http://eskilas.allamehelli.ir>.
2. Vosoughpour M. Some supervised learning schemes for sbd path-finding problem.
3. Amini Zanjani M. Saharkhiz S. Bakhtiari M. Montazeri M. Vosoughpour M. Kaviani P. Eskilas soccer 2d simulation team description paper. RoboCup 2010 Singapore.