2D Soccer Simulation League Team
Description Paper
KN2C (Iran)

Armin Sadreddin , Soheil Changizi , Mojtaba Moazen
Alireza Sadrai , Danial Khoshkholgh , Amirhossein Hassankhani

Khaje Nasir Toosi University of Technology, Iran

http://kn2c.ir/RCSS2D/

Abstract. In this paper, we will outline methods incorporated in preparation for IranOpen 2D Soccer Simulation 2018 by KN2C. Our team started its work in March 2017 and this is the first iranOpen competition that we want to participate in .We started our work by improving the team's defensive and offensive strategies by using Hungarian matching algorithm, Voronoi diagrams and simulating passes.

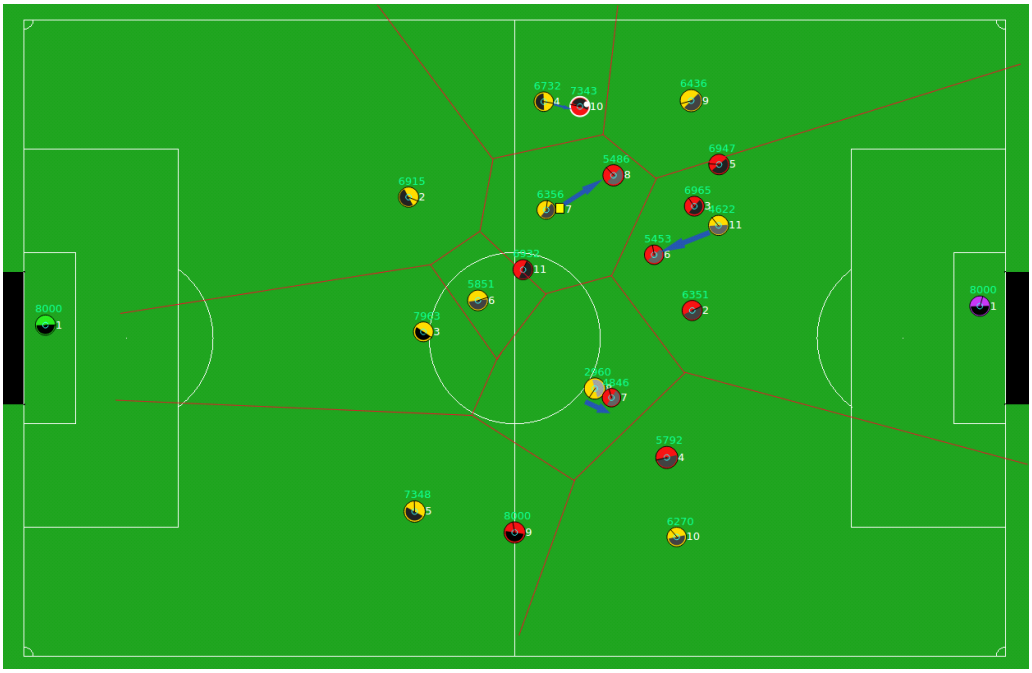
1 Introduction

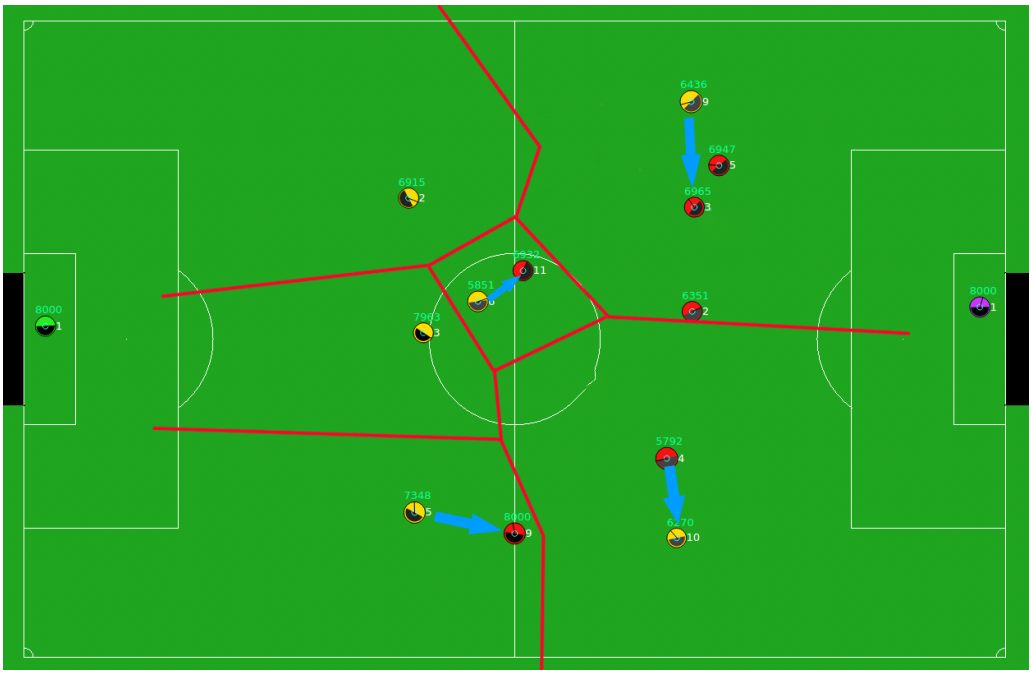
KN2C Robotic team is formed up by B.A. students currently studying at KNTU. KN2C has participated in numerous competitions with Rescue Robots, Small Size Robots and UAV leagues in the previous years. Our members are working with agent 2D base (release 3.1.1) made by H.Aakiyama .

2 Decision making for marking opponents using voronoi diagrams

To prevent opponent from passing the ball effortlessly, we tried marking all their players while having a base position for our agents and not letting them get far from their positions at the same time. So we implemented what is called zonal marking in soccer:

Every player has a zone to mark within it and this zone is determined by their base positions. We give these base nodes as input to our voronoi diagram and every agent marks the most important opponent in his zone. Although this algorithm seemed to be an improvement upon base's simple defence, soon it proved to be inefficient due to the fact that there are lots of situations where there is more than one opponent in a zone and by using mentioned method we will never mark them all. To overcome this, we ran this method recursively every time that a player became in charge of an agent by omitting both the defender and his corresponding attacker for the sub voronoi calculations.





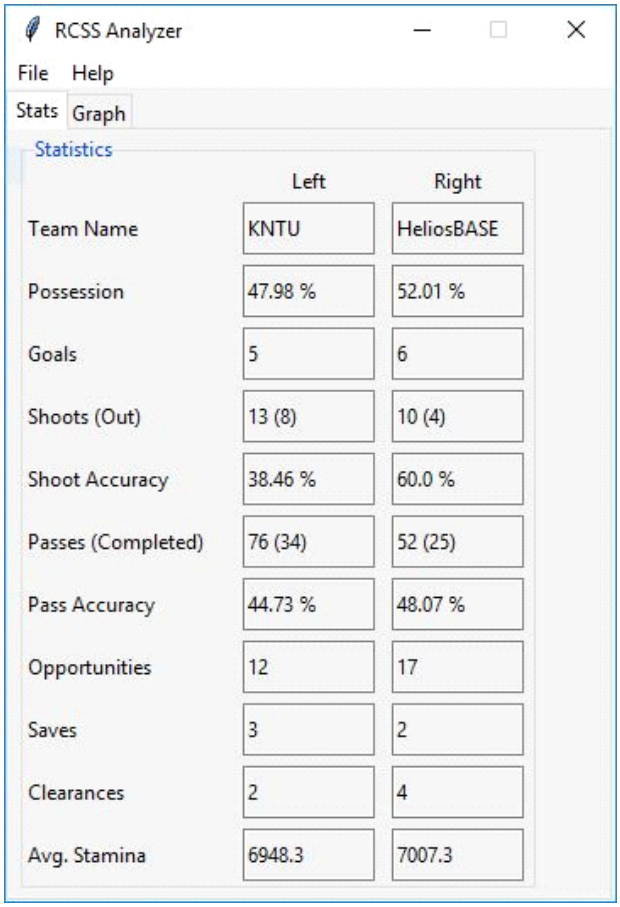
3 Decision making for marking opponents using Hungarian matching algorithm

In this method we aimed to minimize whole team displacement which has an adverse effect on our agents' stamina and marking opponents productively. We implemented Hungarian matching algorithm to achieve this purpose. First we defined two sets of values: mate_val which is a value given to a teammate based on his effectiveness as a defender; the other is opp_val which is given to an opponent depending on how dangerous he is at the moment. There were a number of factors affecting each of these values (e.g. defender's distance from opponent for former, opponent's distance from our goal for latter and various other factors). Next we used Hungarian matching to find the best set of pairs each containing an opponent and his corresponding marker. This way we could accomplish our 3 main concerns: Keeping the original team formation, saving our stamina and marking opponents in time.

4. Rcss2D Analyzer

The need for a way of evaluating our performance and measuring the improvements was clear from the beginning. At first this was done by running a few matches simultaneously each time and calculating the win ratio and average scores through a bash script. As time went by, this method proved to be insufficient as simply knowing the final result was just not enough. So we started to develop our own way of measurements: an analyzer that can

calculate more than just the results. It takes in game and log files created by RcssServer(i.e. .rcg and .rcl files) and analyzes different aspects of game such as ball possession, pass and shoot accuracy and etc.



5. Offensive positioning by simulating passes for different positions

In offensive situations players not holding the ball need to be in a place easily reachable by ball to receive a productive pass and they should be able to get to that point in time. The second important consideration was that sometimes there is no need for an agent to be in position to get a pass directly from the teammate currently holding the ball, but rather receive ball through a second pass from another teammate. In order to do that we first determine which opponent should each of our agents escape from. Next we have to find the best place to go in order to receive a pass. To do this we simulated each pass our player could receive around his current position and tried to find out whether opponent would intercept each of these passes or not and then we chose the best point from the possible choices. After that we did the same for our second receiver with the difference being our first receiver is now the considered to be the current ball holder. This process can be done as many times as possible but further we go down the layers, less reliable the results become and heavier the calculations turn out to be so we decided to go with as

many as 3 layers (3 passes).

6. Conclusion

By implementing methods mentioned, we achieved the following results:

We ran 50 matches against agent 2D base (release 3.1.1).

- After implementing voronoi diagram method for defense, we achieved 0.7 goals received per match.
- After implementing Hungarian matching method for defense, we achieved 0.4 goals received per match.
- After implementing our offensive method, we achieved 3.4 goals scored per match.

After comparing both our defense methods & their results, we decided to go with the latter for competitions.

7. References

1. Hidehisa Akiyama, Tomoharu Nakashima: HELIOS Base: An Open Source Package for the RoboCup Soccer 2D Simulation, Proc. of 2013 RoboCup Symposium, 8 pages (2014)