

CYRUS 2D simulation team description paper 2014

Rauf Khayami¹, Nader Zare¹, Maryam Karimi¹, Payman Mahor², Ardavan Afshar³, Mohammad Sadegh Najafi¹, Mahsa Asadi³, Fatemeh Tekrar¹, Ehsan Asali¹, Ashkan Keshavarzi¹

¹Shiraz University of technology, Shiraz, Iran
khayami.raouf@yahoo.com, nader.zare@outlook.com,
maryam.karimi69@gmail.com, msnajafi405@gmail.com,
f.tekrar@yahoo.com, ehsanasali14@yahoo.com, armanksh71@gmail.com

²Khajenasir University, Tehran, Iran
peymanmahor@yahoo.com

³Shiraz University, Shiraz, Iran
ardavan.afshar@gmail.com, mahsa.asadi@gmail.com

Abstract. This description includes some explanation about algorithms, aims and analyzer software designed by Cyrus team members. The aims of this description is to express a brief explanation about analyzer software, behavior, defensive decisions, optimizing states evaluations, offensive movement without ball and offensive decision making. It also explained about the parts that are implementing. The base code that Cyrus used is agent 3.11. We use artificial intelligence techniques to improve the team in defense, offense and evaluation.

Keywords: team description paper, analyzer software, defensive decisions, optimizing states evaluations, offensive movement, offensive decision making

1 Introduction

The Miracle robotic team formed 5 years ago by students of Shiraz University of technology. This team has already certified for competitions such as IranOpen 2012 and 2013, sharifCup 2012 and also robocup universal competitions in 2013. Team members have achieved the 5th place in Sama robocup competition, the 6th place in 4*4 IranOpen league and also the 8th place in Netherland universal competition.

We decided to use artificial intelligence and collecting information online or offline for all decision makings after universal competition 2013. At the present we are using C4.5 algorithm and Qlearning in intelligent parts. Mentioning that team members have done their undergraduate Thesis in the field of robocup learning consist of basic robocup tutorial and used algorithms in Cy-

adfa, p. 1, 2011.

rus. We intend to present this project as a robocup instruction book and then we will also publish it in English. This paper first discusses the software analyzer and its capabilities, then we explain Movement without ball, view and Behaviors including ball by explaining methods for finding time to reach point, pass and decision making and finally the future works have been declared.

2 Analyzer software

The analyzer software for 2d simulation league is designed graphically in java. This software processes and extracts useful information from the outcome logs of 2d simulation games that explained in the following.

- The discrete evaluation for different parts of field and storing them for use
- Making discrete evaluations continuous and showing them graphically
- Specifying the path to opponent goal from a demanding position, by using continuous evaluation
- Game player
- Specifying players and ball movement continuously in demanding cycles (Fig. 1)
- Presenting all passes, dribbles, shoots and unsuccessful dribbles as a whole or specifically in one picture.
- Presenting the movement of each player during the whole game in a picture
- Presenting the game abstract information such as number of successful and unsuccessful passes, dribbles, shoots, effective passes and goals.
- Presenting the ball path during the time that each team owns the ball one after the other.
- Analyzing games and getting output field evaluation for each game and also many games and changing discrete field evaluation based on output.

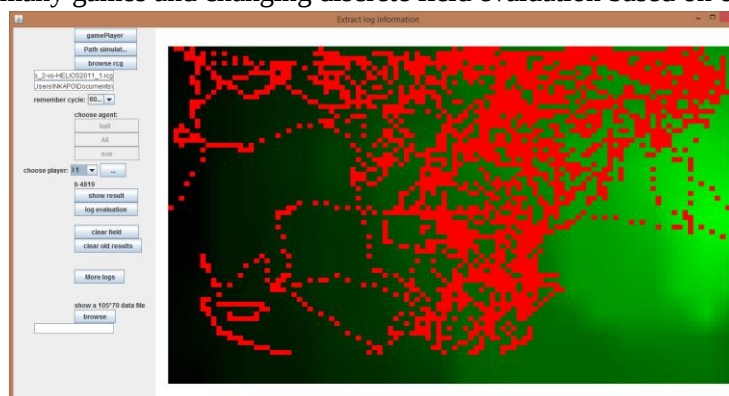


Fig. 1. ball positions during the game up to cycle 4819

2.1 Optimizing evaluation

This is a main capability of the software and by using this capability the field evaluation that is the main states evaluation can be optimized for different teams. The software can get one or more logs and gains the behavior chain of our team from the cycle we own the ball until we lose it or achieve a goal. Considering the positions that ball passes through them, we change the evaluation of cells according to the behavior chain. For example if a chain causes a goal, the path will get better evaluation. Also if a chain causes counter-strike or it didn't reach a good target, the evaluation of chain positions will decrease. This capability can be used in defense state evaluation but not implemented yet. Then by using new evaluations and old ones, an optimized evaluation will be gained. Before competition we will add this capability online by using coach. This evaluation is as follow:

$$NewEv(i, x, y) = (OldEv(x, y) + SetEv(i, x, y)) \text{invert To range } [0,100]$$

$$BestEv(x, y) = (a * OldEv(x, y) + b * Avg(NewEv(i, x, y))) / (a + b)$$

[$i=1$ To $i = \text{Number Of Games}$]

By using the mentioned formula and analyzing 10 play logs versus Yushan13 field evaluations turned out as Fig. 2:



Fig. 2. Field evaluation for

3 Movement without ball

The most important part of the 2d simulation is movement without ball, because in one moment there is at most one player owning the ball and 21 players without ball. The movement without ball consists of two parts: offensive system and defensive system

3.1 Move without ball in offense time

move without ball in offense time makes the team increase the percentage of its ball owning and decreases the opponent offense time, decays the opponent players energy, effectives the offenses and fuddles the opponent defense system such as mark and press. For this part it should be obvious initially that the player should move without ball in respect to which player. The selected player is the player owning the ball or the player who can pass the ball to the player who is executing the algorithm in case of receiving the ball. In the future, by specifying the various states and the operation done based on each state, we will choose the player who our player must move without ball in respect to him.

After choosing the considered player, we assume some point around the player where we assume them as the position of the player and we consider the whole possible receiving passes from that position (Fig.3). The value of each point is calculated based on the number of the passes which is received in that point or the value of the best pass received in that point. In order to decrease the process we check the usefulness the last move target and the algorithm will be executed if the last move target was not useful.

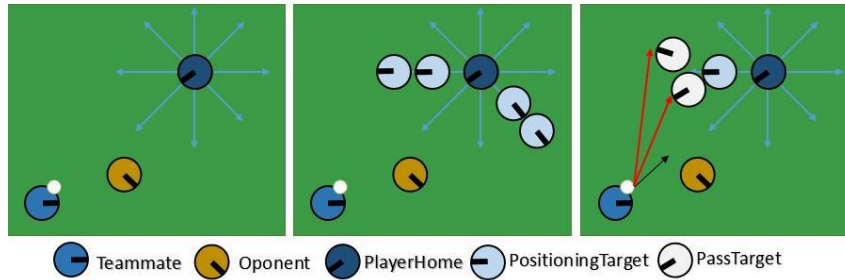


Fig. 3. move without ball in offense time

$$PosEv(i) = NumberOfPass(i) * Avg (PassEv (i)) + BestPassEv (i) + OppMarkPos (i)$$

4 Behaviors including ball

In Cyrus 2014 we tried to implement Cyrus 2013 algorithms (universal competitions) and raptor 2013 (iranOpen competition) in chain action and resolve existing bugs.

4.1 Decision making while owning ball

For offensive decision making by UCS searching algorithm a brief state of present situation will be made, then by considering this and possible behaviors like pass, dribble,... a new state will be created, then we will initiate the states evaluations by considering ball position in the created state and the effect of opponent players on the ball owner player. Then all existing states will be sort and we will do the same for the best state. At the height of more than one behavior, all processes will be light.

5 Decision making while owning ball

The basic of evaluation in Cyrus 2014 is somehow the same as 2013 but evaluation of each state will consider ball position and the effect of opponents.

5.1 Evaluation by considering ball position

In Cyrus 2014 evaluation according to ball position, is the same as Cyrus 2013 but positions around opponent's players will get a lower evaluation as shown in Fig 4. According to the results of percent of correct dribbles, gained by analyzer software, and recognition of dangerous opponent players this evaluation will be corrected and optimized.

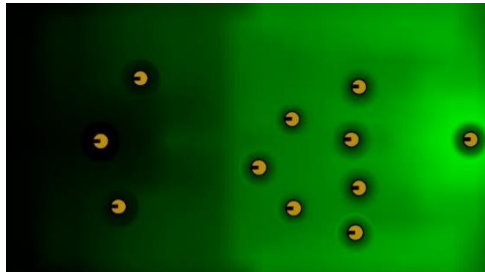


Fig. 4. Evaluation by ball position (positions around opponent's will get a lower evaluation)

5.2 The effect of opponent's players

The effect of opponents on the ball owner gained by following characteristics:

$$PressEv = \text{Sum For All Opp} (z1 * Dist + z2 * BodyAngleDif + z3 * VelAngleDif + z4 * Vel + z5 * PosCount + z6 * CycleRichDist)$$

Dist: distance of opponent player

BodyAngleDif: the difference of opponent body direction with the direction that it should go to reach ball owner player

VelAngleDif: the difference of player's movement direction with the direction that it should go to reach ball owner player

Vel: the velocity of opponent player

PosCount: the period during which the opponent player has not been seen

CycleRichDist: the number of cycles opponent player needs to reach the ball

By using the calculated values it can be specify that with which probability the opponent player intends to mark or block the ball owner player. The amount of coefficients has been gained by c4.5 algorithm.

5.3 Behaviors safety

The behaviors can get safety evaluation of 0 or 1. If a behavior has a low safety this can have effect on all of its following states in that branch. Then evaluation of each state will be calculated as the following formula:

$$EvOfState = DynamicGroundEv(BallPos) - PressEv(state) + Confidense(PathChainAction)$$

6 Future

We try to implement an algorithm that stores a created dictionary of states, gained form different games before competitions, then by considering the existing behaviors find the best behavior by light processing and execute that, and if an executed chain was useful this state will be added to the dictionary. We also intend to use Montecarlo algorithm for finding the exact position for players, and for predicting players movement we intend to predict positions by using use the existing states in the dictionary.

7 Reference

1. Akiyama, H., Noda, I.: Multi-agent positioning mechanism in the dynamic environment . In: Visser, U., Ribeiro,F., Ohashi, T., Dellaert, F., eds.: RoboCup 2007: Robot Soccer World Cup XI, Lecture Notes in Artificial Intelligence. Volume5001., Springer (2008) 377-C384.
2. Srinivas M., Patnik L.M., "Genetic Algorithms: A Survey", Journal of Computer, IEEE, 1994.
3. Satje A. Reinforcement Learning of Player Agents in RoboCupSoccer Simulation[A] . Proc of the 4th Intl Conf on Hybrid Intelligent Sus-tem[C].2004