BahiaRT 2018: Team Description Paper for RoboCup 3D Soccer Simulation League

Marco A C Simões, Mateus Nascimento, Sérgio Souza Jr., Wellington Correia, Caroline Souza, Josemar Rodrigues de Souza, Robson Marinho, and Jorge Campos

Bahia State University (UNEB/ACSO), Salvador, BA, Brazil
teambahiart@gmail.com
http://www.acso.uneb.br/bahiart/

Abstract. This paper describes the most recent improvements for team BahiaRT in RoboCup 3D Soccer Simulation League since 2017. New developments related to kicking movement, self-localization, roles, behaviors and setplays are described. Also some work in progress about learning of new setplays and behaviors is presented.

1 Introduction

Team BahiaRT has experienced a remarkable evolution in RoboCup competitions during the last four years. In the world championship, BahiaRT ranked fifth in 2014, fourth in 2015 and 2016 and sixth in 2017. In the last three years, we were also Champions(in 2015) and runner-up(2016) of the Portuguese RoboCup Open competition. We won the last five editions of Brazilian Robotics Competition and the last four editions of Latin American Robotics Competition.

Aiming to maintain this evolution, we have decided to enhance some of our high-level AI strategies as well as increase some basic skills. We have worked with development of a new short and fast kick (see Section 2). We have also diagnosed some problems with our agent's self-localization procedure. We describe our planned enhancements on this routine in Section 3. we have developed a new strategy for dynamic roles allocation and new behaviors definitions (see Section 4) and new setplays(see Section 5).We also have begun to work in a new project to use machine learning to improve and learn new setplays(see Section 6).

Since 2013, team BahiaRT is based on a source base code named FCPBase 2013, as part of an international scientific cooperation partnership with team FC Portugal from Universities of Aveiro, Minho and Porto in Portugal.

2 Short kick

Our team has a long range kick that can reach up to 21 meters and, since we can't control the force of the kick, this distance is almost the same for all long range kicks. The robot takes a lot of time to position itself in relation to ball and target, before perform this long range kick. To deal with that situations we need a new approach to kick, making that skill capable of be performed even if there are opponents nearby. We have developed a new short range kick which takes fewer time for preparation. The main idea is to use this new kick in scenarios where there is few time available for preparation, such as, situations where there are many opponents nearby.

Before the kick, the robot needs to move himself to a defined location that is optimal for the kick to occur. The main difference is that the robot will not stop when reach the target location. He will only lowers his speed enabling kick to use part of the inertia of the walking movement.

We're using CMA-ES [1] to optimize a keyframe based [2] kick that uses the maximum speed allowed for each joint, the speed of each joint and the duration of the keyframe, as done in [3]. In the optimization, the fitness function needs to increase the kick reach, lower the kick duration and lower the angular variation of the kick, since we want an straight kick.

Since we aren't using any transition between the positioning phase and the kick phase, we need to define the correct moment to execute the kick. The force resistance perceptor (FRP) give us values of the forces applied to the bottom of the foot, allowing us to know whether one of the foots is off the ground or even the best time to kick by checking if the robot is balanced before kicking.

After the kick, the robot has to go back to the walking behavior and the Zero Moment Point based walking will find a way to keep up with the recently unbalanced robot. The foster this balance, because of the penalties applied to the fitness values if the robot falls before, during or after executing the kick.

3 Enhancing localization

In the 3D simulated soccer league there are some points in the field called flags that are used by the agents as landmarks to determine their own positions. The field has eight fixed flags as we can see in the Figure 1. The four *Goal* flags are located in a higher position (0.8 m from the floor).

Our localization system uses the triangle formation method for localization. Based on this method the agents must see the flags in the field and form a triangle as we can see in the Figure 2.

When agent can see three flags, we can get a more stable location system due to noise reduction from robot head movements. The robot should find three flags where two of them has the same coordinate value for x axis or y axis and the same value for z axis. In the Figure 2 these flags are *corner left top* and *corner left bottom.* The third flag must have the same x or y values as the other two flags, but not necessarily the same z value. In the same example, the *goal left bottom* flag is beeing considered as third flag. Assume $F_i; i = 1, 2, 3$ are the three flags the robot is seeing. Consider F_1 and F_2 as the first two flags which y and z values are the equals. In other words, F_1 and F_2 differ only in their x coordinate. The robot estimate its own localization P:

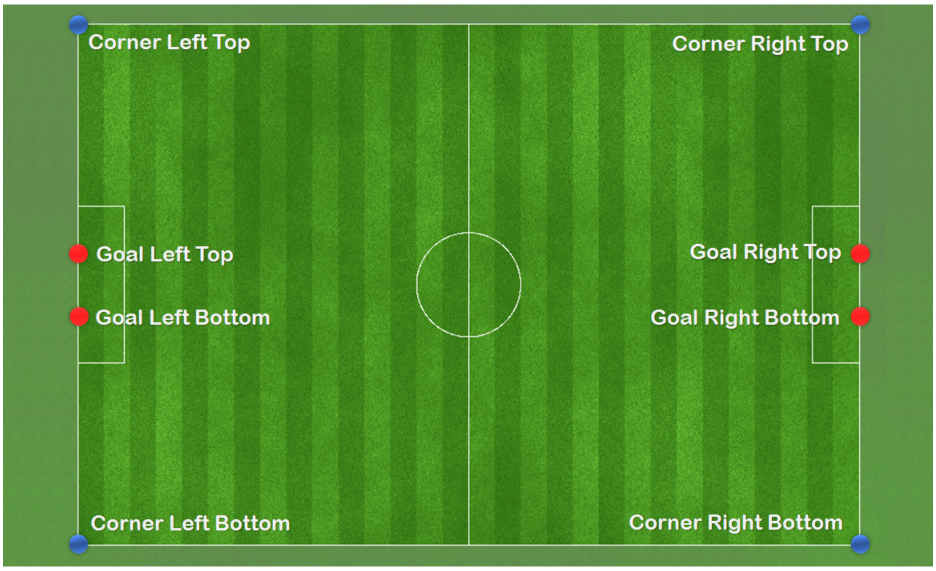

Fig. 1. Flags used as landmarks in RoboCup 3D Soccer Simulator.

$$\begin{aligned}
YY &= \max_{y} F_i - F_j; i = 1, 2; j = 3 - i \\
XX &= F_3 \times YY \\
ZZ &= XX \times YY \\
LOCHEADTOFIELD &= (XX, YY, ZZ) \\
\varDelta_1 &= F_1 - LOCHEADTOFIELD \times F_1 \\
\varDelta_2 &= F_2 - LOCHEADTOFIELD \times F_2 \\
P &= \frac{(\varDelta_1 + \varDelta_2)}{2} \qquad (1)
\end{aligned}$$

LOCHEADTOFIELD is the basis for the definition of the agent's position. It is composed by three vectors XX, YY and ZZ calculated as described in equation (1).

A more important problem diagnosed in this localization system is when the robot is near the corners or goal line. In this case, the lines of the triangle used by the agent tend to get very close and this results in a noisy location. In practice the triangle almost degenerate to a straight line.

Our proposal to solve this issue, is to keep the previous agents' location and also use information received by messages from other agents. This way e can predict the current agent position and compare it to value calculated by the robot. We are also studying other localization systems to enhance the current approach and provide better information for agent AI.

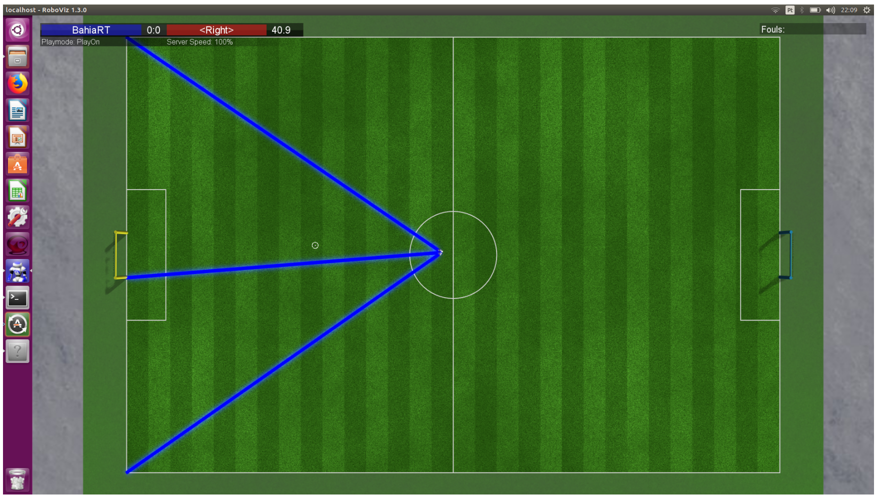

Fig. 2. Triangles used to estimate self-localization.

4 Roles and Behaviors

We developed a structure of roles and behaviors in our team. Each role is associated with a set of possible behaviors and activation rules. Figure 3 depicts all roles and behaviors in our team. The Attacker is the nearest agent to the ball; Defensive agents stay near to our own goal to avoid opponents to advance; Support agents position theirselves in strategic points preventing long kicks of the opponents and enabling the Attacker to perform passes; Keepers are agents located behind the Attacker and can replace him if he falls down or are penalized. We also have one Goalie and a Goalie substitute which can replace the Goalie when he is unavailable for any reason.

Each agent will assume specific behaviors for each role. The Attacker may assume three behavior: CarryBall (agents that holds a ball possession and aims to take him to the opponent goal), CatchBall (agents who aim to acquire a possession of the ball) and Passer (responsible for running a pass to a Support agent). A Keeper can assume the behavior of ActiveMarker which are the players that try to mar the opponent with ball possession. Support agents takes on the behavior of Receptor whose goal is to stay available to receive a pass. Defensive agents can perform the behaviors ActiveMarker and Receptor, but they also perform the. PasssiveMarker behavior. When acting as PassiveMarker, the agent will mark the opponent players which do not own the ball possession to prevent possible passes. All these behaviors are active behaviors which are fired according to specific activation rules for each role and behavior. When no activation rule is fired agents perform the strategic positioning which is named as Formation behavior. This is the unique behavior for roles Goalie and Goalie Substitute.

The role of each agent is key to defining their positioning. Currently the system used for team positioning is SCRAM (Scalable Collision-avoiding Role

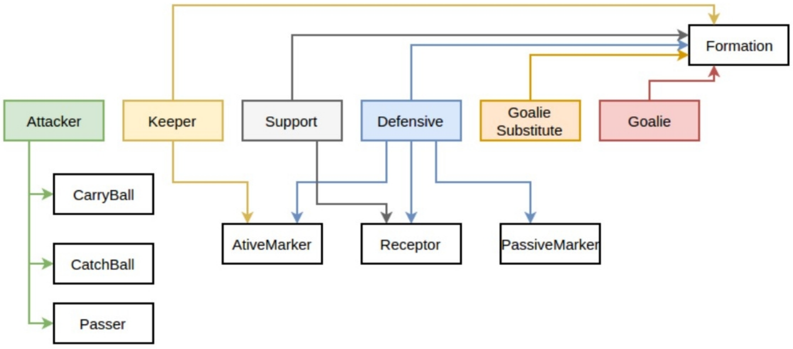

Fig. 3. Behaviors in black and roles in colors

Assignment with Minimal-makespan)[4]. The analysis of the positioning of all teammates is carried out, ensuring that each agent walks the least possible path to perform strategic positioning.

5 Setplays

To provide an intelligent and cooperative behavior for the team, it is necessary to develop a setplay module where the agents cooperatively executes sequences of behaviors. The module is divided in two pieces, the first one being the setplay planner, and the second one the framework which allows the connection between the generated setplay and the team code.

To create a new setplay, we use the Splanner[5], a graphic application tool easily create and maintain a setplay. Using the graphical tool, it is possible to configure the conditions to execute each setplay, besides the actions the agents should run depending on the current step. SPlanner generates a text files containing all setplays definition.

The framework reads the file with the setplay and convert all the instruction code into series of behaviors that the agents should execute according to their position and the global time.

Using the Case-Based Reasoning method (CBR), it is possible to define if a setplay should be executed or not. When occurs a change in the world state, like a change of playmode, or certain position of the agents, the CBR engine chooses a setplay to execute. After the conclusion step the engine logs the data, as if it was a success or a failure, to allow a better result when a similar situation is repeated in the future.

When a setplay is activated, an agent is defined as setplay leader, in order to maintain the coordination between agents. The leader choose the players that will participate on the setplay, based on their position on the field and send a message to them. After that, the leader continuous to send a step message informing the current and next step, until the end of the setplay.

6 Learning new behaviors

SPlanner reduces the coding effort but it does not solve the problem to predict all possible situations a team of agents should preview to be efficient in a complex domain such as robotic soccer. To build an efficient set of setplays many researches have been considering using machine learning.

We propose a solution to learn new setplays considering existing basic behaviors and simpler setplays. The idea is to learn complex setplays for real time situations. Our proposal will deal with real time play on gaming situations where agents requires complex setplays to achieve their goals. Our proposal is also concerned with both defensive and attacking setplays. We will use task decomposition to reuse existing knowledge (basic behaviors, basic setplays, internal agents' states and decision making policies) to optimize learning convergence and give appropriate bias to the learning process. We will consider recent results using learning from demonstration as good option. To deal with uncertainty we plan to investigate using of bayesian networks and fuzzy learning. We expect to use this to provide our solution to learn setplays. At this moment, this project is in project in a very early stage.

Acknowledgements

This project is partly funded by CNPq/PIBIC, FAPESB/IC, UNEB/PICIN. We thank team FCPortugal for their cooperation and partnership.

References

1. Nikolaus Hansen. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.
2. Mike Depinet, Patrick MacAlpine, and Peter Stone. Keyframe sampling, optimization, and behavior integration: Towards long-distance kicking in the robocup 3d simulation league. In *Robot Soccer World Cup*, pages 571–582. Springer, 2014.
3. Klaus Dorer. Learning a humanoid kick with controlled distance. In *RoboCup 2017: Robot World Cup XXI*, pages 100–111. Springer, 2017.
4. Patrick MacAlpine, Eric Price, and Peter Stone. SCRAM: Scalable collision-avoiding role assignment with minimal-makespan for formational positioning. In *AAMAS Autonomous Robots and Multirobot Systems Workshop (ARMS 2014)*, May 2014.
5. Fernando Almeida João Guilherme Bettencourt Cravo, Luís Paulo Reis. Uma aplicação gráfica de definição flexível de jogadas estudadas no robocup. Retrieved January 11, 2018, from https://repositorio-aberto.up.pt/bitstream/10216/62120/1/000149781.pdf, 2011.