MRL3D Simulation Soccer Team 2017

Zahra Noury, Aref Moqadam Mehr

Mechatronics Research Lab., Dept. of Computer and Electrical Engineering,
Qazvin Azad University, Qazvin, Iran
{*z.noury, a.moqadam*}*@mrl-spl.ir*

Abstract. This paper describes the developments by MRL3D simulation soccer team entering the Robocup competitions in 2017. MRL3D is a research team of the Mechatronics Research Laboratories (MRL) under supervision of the Qazvin Azad University.

1 Introduction

Mechatronics Research Laboratory (MRL) is an active participant of the RoboCup World Champion since 2002 in different leagues. MRL3D is one of the research teams of the MRL established in 2017. It focused on researching in the field of Artificial Intelligence and Machine Learning. Currently, MRL3D is using the UT Austin Villa 2016 code release as the framework and infrastructure. Several improvements have been made on behavior base to complete hybrid automata with synchronized constraint for co-operation tasks. These improvements has previously implemented on Standard Platform League (MRL-SPL). These researches will lead the MRL3D to achieve a reliable behavior engine. Moreover, further investigations have been made on optimized path planning and kick engine to enhance the precision of reaching and kicking the ball duties. These development includes implementation of following modules: optimized path planning, optimized obstacle avoidance, multi-agent cooperative behavior strategy, and using machine learning-based methods for selecting behavior parameters dynamically. These researches will be briefly described in this paper.

2 High Level Behavior

In previous years we implemented a basic formation system, passing mechanism and task assignment algorithm which enabled us to cover our high-level decisions in gameplay on the Standard Platform League (SPL) robots [1]. Due to the essence of homogeneous teams of robots, and to have an agile cooperative team of robots we used [2] as our task assignment algorithm. With some enhancements, our implementation minimizes the time that it takes for an agent to take a position. Currently, our general strategy is to dynamically assign robots to predefined formation position while one or more of them take position according to the ball, namely leader and supporter. Other Works Toward possible improvements, the passing mechanism can be integrated into task assignment algorithm. Currently, this mechanism works in one way; an agent finds appropriate spaces to pass to. However, it can work both ways. The second agent (pass-taker) gets in position prior to the execution of the pass action by the first agent. This mechanism has been implemented in SPL and used in RoboCup 2016. We are also eager to evaluate the feasibility of using SCRAM [3]. Furthermore, we are planning to add more behavior skills to be more competitive.

3 Voting Role Selection

Since the data calculation is distributed in the robots in behavior level, a synchronization method is required to keep the team coordinated. The proposed method is to use a voting system to assign a position to each robot. In this method there are dynamic positions defined on the field which initially are assigned to each player by the player number. Afterward, during the game, players can request for another post based on their proximity to the desired position. This post switching happens through a handshaking mechanism. For instance, if robot A is assigned to a position near robot B, then robot B can ask for the privilege of the post of robot A. Finally, the robot A respond to this request with an acknowledgement and both robots switch their posts. This method ensures the robustness to the network and robot mis-coordination.

4 Head Motion

The fulfillment of the perception module has a major influence on the robot action during the match. In fact, the performance of each robot is highly dependent on its model of the environment, therefore, on how good it percepts. More useful and precise data can be collected from the environment by looking toward the directions in which it is more probable to see a desired data. A similar approach introduced by [4]. Hence, a weight is calculated for each direction with respect to the expected value of objects that would be seen. There are also other parameters influencing weight of each direction such as time cost and the time since the direction last seen [5].

5 Obstacle Avoidance

The positioning approach that is being used moves the robot directly toward the given target and this may cause collisions with the obstacles in the way. To overcome this risk, the target that is going to be used in the positioning algorithm is overwritten to an intermediate point if a collision is predicted. The intermediate point attracts the robot to a point from which it can go directly toward the original target without collisions. The first step is defining collision and predicting it. To simplify equations and processes, all obstacles are considered to be circles. Following this simplification, if the reference point of the robot is located inside an obstacle (circle), a collision would happen. Hence, a collision is going to take place if and only if the line through robot and target is crossing an obstacle. A little basic geometry hands a condition which forecasts collisions based on the position and radius of obstacles and the position of ball relative to the robot. In case of previewed collision, an intermediate point is calculated either on the left or right of the obstacle through the following steps. First, a tangent line is calculated to the obstacle which is previously considered to be a circle. Then, on the line through center and tangent point of the circle, with a safe distance from center of the circle, called safe radius, the intermediate point is chosen. The safe radius should always be a sufficient amount more than the obstacle radius. The difference between obstacle radius and safe radius causes the intermediate point to move around the obstacle as the robot walks toward it which makes the robot to walk on a curve around the obstacle. As soon as the robot is turned enough around the obstacle, no more collision is predicted and the original target is used.

6 Formation

There are several methods for global team positioning during the game. A simple method to overcome this problem is to use static position during the game; however, this method lack the response to environment (ball and other robots). In order to provide enough flexibility to the game condition we have designed a mechanism for adjusting these static positions. Each static post is defined in a a normalized space. Then, the distance to the ball and other robot in addition to the robot's global position itself is considered. Based on these data an adjustment applied to the original posts. Finally, the corrected positions convert to the field coordination and used as positioning.

7 Positioning Algorithm

Last but not least, an algorithm is required to move the robot to a target pose, known relatively to the robot. It is important to be accurate enough to have the ball in the right position for a good kick. On the other hand, it is necessary to be fast enough to own the ball left on the field in a duel with an opponent. Consequently, methods used in far and close distances are separated. When the robot is far from the target, it does not notice the target orientation and moves using forward and rotational movements i.e. no sidewalk. Near the target, however, it moves more deliberately and considers the orientation. In far distances, rotational speed is set proportional to heading error with the target, and forward speed is set proportional to both the heading error and distance to the target. When the robot is close to the target, a minimum time is calculated for each of the three errors, in x and y axes and rotation around z-axis, to be made zero. Assuming that the speeds and their maxima are independent of each other, maximum of the three calculated minima is the actual minimum time for all the errors to diminish to zero. Having the minimum time, we can calculate a combination of speeds, which will reduce errors to zero in the same duration. This method makes the robot move on an almost straight line to the target.

8 Future Works

There are several parameters used in the behavior engine that can have a tremendous effect on the gameplay. How much offensive or defensive our strategy must be, how often should agents pass the ball, at what level of confidence they should kick the ball toward the goal are all examples of these parameters. These parameters usually tuned prior to each game depending on the opponent team. We have planned to implement a mechanism to set these parameters dynamically during the game. The idea is to maximize these configurable parameters and train a Deep Neural Network to use the game data and offers these parameters. These data includes opponent strategy and formation, the average time the ball was in our field, the number of robots, time left, and goal differences can all be strategically important. In addition, knowing how defensive or offensive robots should play can specify how many defenders we need, how robots can score a goal, and so forth. Our formation positions are a function of game states, the number of field players, and the ball position. This function can be improved by taking into account the free spaces of the field.

References

1. Aref Moqadam Mehr, Mohammad Ali Sharpasand, Novin Shahroudi, Farzad Fathali Beyglou, and Mohammadreza Hasanzadeh. "MRL-SPL."
2. Patrick MacAlpine, Francisco Barrera, and Peter Stone. Positioning to Win: A Dynamic Role Assignment and FormationPositioning System. In Xiaoping Chen, Peter Stone, Luis Enrique Sucar, and Tijn Van der Zant, editors, RoboCup-2012: Robot Soccer World Cup XVI, Lecture Notes in Artificial Intelligence, Springer Verlag, Berlin, 2013.
3. Patrick MacAlpine and Eric Price and Peter Stone, SCRAM: Scalable Collision-avoiding Role Assignment with Minimal-makespan for Formational Positioning, AAMAS Autonomous Robots and Multirobot Systems Workshop (ARMS 2014), Paris, France, May, 2014.
4. Andreas Seekircher, Tim Laue, and Thomas Rofer: Entropy-Based Active Vision for a Humanoid Soccer Robot, 2010.
5. Omid Amirghiasvand, et al. MRL-SPL TDP Description for RoboCup 2015. RoboCup 2014: Robot World Cup XIX. Lecture Notes in Artificial Intelligence, Springer