MRL-SPL
Team Description for RoboCup 2015

Omid AmirGhiasvand[1], Novin Shahroudi, Mohammad A. Sharpasand, Aref Moqadam Mehr, Pooya Sagharichi Ha, Mohammad R. Ghazeli, Mohammadreza Hasanzadeh, Mohammad A. Zakeri Harandi

Mechatronics Research Lab., Dept. of Computer and Electrical Engineering, Qazvin Azad University, Qazvin, Iran
[1] *amirghiasvand@qiau.ac.ir*

Abstract. This article gives the team description and a concise explanation of research interests of MRL-SPL team, aiming to participate in RoboCup 2015 Standard Platform League. The presented document covers various subsections including perception and world modeling, behavior control, motion planning, and future works.

1 Introduction

MRL-SPL team, under the supervision of Qazvin Azad University (QIAU), is one of the research groups of the Mechatronics Research Laboratory (MRL), dedicated to work in the field of biped and mobile robots. MRL presence in RoboCup different leagues since 2002, results in numerous successful achievements both in the areas of research and competition. MRL-SPL team has been an active participant of world RoboCup since 2009 except 2013 tournaments of the Netherlands.

The team members of MRL-SPL are:
Undergraduate members: Aref Moqadam Mehr, Mohammad Ali Sharpasand, Novin Shahroudi, Mohammad Reza Ghazeli, Pooya Sagharichiha, Koosha Zarei, Parham Sagharchiha, Mohammadreza Hasanzadeh, Behrad Kazemi, Mohammad Alikhani;
Graduate Researchers: Mohammad Ali Zakeri Harandi, Mohammad Shafiei R.N;
Post Graduate Researchers: Omid AmirGhiasvand, Sepehr Tabatabaei;
Team Leaders: Omid AmirGhiasvand, Mohammad Ali Zakeri Harandi.

In what follows, a concise explanation of ongoing research projects, with focus on the recent achievements, together with the future works are presented. This paper is structured in 5 sections including this introduction. Section 2 describes new improvements of perception module and modifications regarding the white goal post while section 3 portrays latest improvements of behavior control system. Section 4 gives an insight to the motion planning and basic functionality of robot behavior programs. Section 5 suggests the future works and a brief conclusion of works explained in this paper.

2 Perception and World Modeling

2.1 Camera Calibration

In order to estimate the position of an object precisely, the exact positions of the robot cameras are required. Here, even a slight displacement and/or disorientation of cameras, might cause a huge error in estimated distance of objects. To reduce these errors, the position of cameras has to be calibrated. To achieve this aim, a modification has been applied to B-Human calibration method [2]. The B-human module needs the entire or half of the empty field as a prerequisite. The crowded field, which is a common occurrence during RoboCup, makes MRL-SPL to propose a module whereby only several plotted squares are adequate to perform calibration precisely.

2.2 Calibration Free Image Processing

Tuning of vision modules parameters is an important task which requires an extensive time prior to each game. As a result of having white goal posts in upcoming RoboCup competitions, our previous solution for robot perception will no longer be applicable and a modified approach shall be occupied. MRL-SPL approach to overcome this difficulty is based on Bayes classification [3]. Thanks to this approach, robots and goals color features are no longer the main characteristics, therefore their shape features and positions are being used instead. Bayes classifier is supposed to classify image regions. These regions are actually a group of vertical scan-lines (line segments of homochromatic pixels). The criterion by which pixels are called homochromatic which used to be color tables are now the Euclidean distance in YUV422 color space [4]. A number of shape features in each region, are chosen as a distinction between classes. Features of each region, like size or distance from horizon are extracted and learning methods are applied on them in order to detect mentioned objects. For learning process, a supervised classification is applied on a data set of thousands random images as seen by a walking robot in game field during a competition [5].

2.3 White Goal Detection

As mentioned previously, after the image color sections are built, based on the property of each one, a feature vector is calculated. A goal post is a white section which starts above the horizon and ends at the field. It can be validated by its radius, distance to the field landmarks and other detected goal posts.

In addition, to achieve even more percise data a Neural Network has been employeed to learn the difference between real goal posts and other objects such as robot arms. The network is trained offline via a supervised sample set then the parameters are used in the main module in order to distinguish between the real goal posts and other objects.

2.4 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 simiiliar approach introduced by [6]. 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.

2.5 Object Tracking

A variety of algorithms are applied to achieve object tracking. Some objects such as ball are unique. The position of some others is initially known, such as teammate robots. Consequently, several object-specified methods have been utilized to track the objects correctly. The goal post positions are extracted out of localization module. However, occasionally the localization might be misleading. If the measured position of a goal post is unexpected, that goal post would be resampled. Besides, when a goal is observed, the position of its posts and the other goals are refined. Since the position and the target of the teammates are known through the network communication, there would be no need to track them, except for the situation where wireless communication is lost. In that case, the previous data is chosen as the best known observation.

3 Behavior Control

For the last year's RoboCup competitions we had started developing a behavior control system from scratch. Our current behavior infrastructure is inspired by hierarchical architectures [7] [8] and patterns like episodic memory [9] and dynamic border. Dynamic Border is a replacement for decision conditions to maintain consistency on a noisy continuous input. This year our high level behavior uses Hierarchical Finite State Machine (HFSM) paradigm instead of FSM and uses C++ for describing high level behaviors instead of Python. New features include a gui interface which helps modifying behavior parameters, new dynamic role/post assignment algorithm and a pass planner method.

3.1 Architecture and Structure

Our structure divides behavior of the robot into two main parts including low level and high level since our 2013 developments. Low level is responsible for reactive behaviors of the robot. In contrast, high level performs the deliberative aspects. Each low level behavior is a hierarchy of task and/or skills. A task is a special form of skill which is callable by the high level. High level behavior of the robot is always run at the top of the hierarchy so that it determines the appropriate task to be run in current execution cycle.

3.2 Task Selection

Task selection is part of the high level behavior which uses HFSM paradigm to model robot high level behaviors. It uses output of the strategy modules such as passing, dynamic role/post assignment to select an appropriate low level task.

3.3 Strategy and Planning

The goal in a multi-robot coordination scenario is to divide a task or set of tasks among robots to reach a common goal with efficiency and cooperation. A formation system is developed which is a means to define robot tasks in a soccer game and coordinate robots. Cooperation and team play achieves higher performance with a pass planner module. Strategy is a set of rules and parameters that affects all underlying mechanisms such as formation and passing modules. For example an offensive strategy affects number of supporter roles (quantity matter) or how solid a player should act (quality matter).

Formation & Dynamic Role/Post Assignment Formation consists of a set of positions called role/post that describes each robot task in a soccer game. Each role/post is a target position. Mapping n robots (R) to n targets (T) is a linear assignment problem aiming to find minimum sum of costs Eq. **??** where cost function C is $C : R \times T \to \mathbb{R}$ and $f : R \to T$ is a bijection of robots to targets. We utilize a heuristic method [11] to solve this assignment problem. One of the motivations to use this method was the similiarity of the SPL and the environment it is developed for.

$$\sum_{r \in R} C(r, f(r)) \tag{1}$$

In our implementation, assignements are time optimal. Fig. **??** depicts the time cost calculation criteria such as distance and rotation to the target.

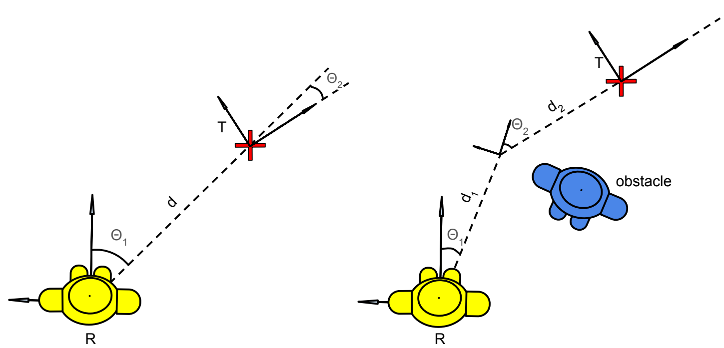

Fig. 1. Calculation of time cost for robot R to target T in which d_1 and d_2 are the modified distance with respect to the obstacle, θ_1 and θ_2 are rotations required.

Graph-based Pass/Kick Beside individual skills like dribbling, the ability of passing to teammates, and kicking towards the goal, are two important capabilities that every robot should possess during the game. The mentioned skills should be executed fast and accurate with high dependency on the suitable point or area for destination of pass or kick. According to these requirements, Graph-Based Pass Planner module has been developed, with two major subsections. The first one, called Analyzing and Planning subsection, generates 40 predefined position variables, called Nodes. The Nodes are sampled directly from the state space, to represent the posterior probability, and are updated by involving the new observation distribution. To evaluate the posterior probability of each Node, other world model data like position of opponent robots, position of teammates relative to each node, etc are also gathered. After finding the best node, a kick with the highest accuracy at distance and orientation should be executed. This objective is achieved by the second subsection, Motion Control, which determines the required hit power to transfer the ball toward the best Node area. To reach an accurate pass/kick, it is decided to employ a lookup table. For each initial ball location with respect to robots foot, a cubic polynomial regression is derived, which estimates ball range after being hit by the robot, as a function of motion duration of swing foot. Fig. 3World state and pass nodes. **Red crosses**: our teammates; **Yellow cross**: Best node; **Black polygon**: opponent team agents shadows; **Blue circle**: pass nodes; **x1**: distance to goal; **x2**: distance to nearest teammate; **x3**: angle between node and strikerfigure.caption.6 shows a sample output of this module.

4 Motion Planning

When the robot decides to kick the ball to a target, for example to the goal, it computes a local point relative to its current position and the ball, which is the point robot is able to kick. The procedure is divided into four steps:

1. Calculation of all of such points, e.g. for kicking with left or right foot
2. Selection of the most feasible point
3. Calculating an obstacle-avoiding intermediate point
4. Positioning algorithm which calculates final commands to walk engine

4.1 Calculation of Possible Kicking Spots

Kicking spots are positions in which the robot can kick the ball to a target with only executing kick action. For each kicking action, there is an expected ball position relative to robot and an orientation on which the ball is intended to move after the kick. Having these data as well as relative positions of the ball and target, spots are located easily. Localization data is avoided which possesses a much lower certainty and accuracy.

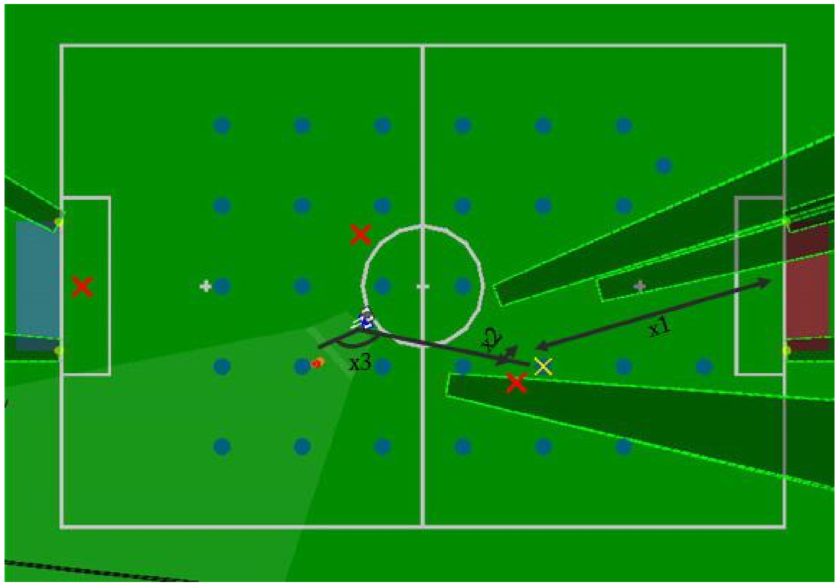

Fig. 2. World state and pass nodes. **Red crosses**: our teammates; **Yellow cross**: Best node; **Black polygon**: shadow of opponent team robots; **Blue circle**: pass nodes; **x1**: distance to goal; **x2**: distance to nearest teammate; **x3**: angle between node and striker

4.2 Selection of the Most Feasible Kick Spots

In this step, it is necessary to choose one of the spots to walk toward it. This choice should not be changed frequently. If so, the robot may tolerate between several orientations and slow down dramatically. Therefore, a consistent measure of comparison is needed. The used measure is the amount of rotation around the ball necessary for going to each spot.

4.3 Calculating an Obstacle Avoidance Intermediate Point

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.

4.4 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.

5 Conclusion and Future Works

SPL is one of the fast growing leagues in RoboCup. There are new challenges introduced every year in this league. One of these new challenges were white goal posts. Our perception and modeling modules are adapted to this new change by the use of calibration free image processing method and neural networks. Every year new strategies are introduced by different teams regarding the behavior control and multi-agent cooperation. To keep up with other teams pace, our high level behavior is reimplemented and a new role/post assignement algorithm is employed. Besides we tried to make our strategies adabtable to get benefit of any opponent we are going to play against in 2015 tournaments. With some modifications in low level behavior algorithms our team is going to be more agile and stable in performing tasks such as dribbling, kicking, obstacle avoidance, etc.

In the future, the graph-based path/kick planner, can be revised using a Fast Multi-Swarm Optimization. The aforementioned method can increase the functionality of the planned spots having the freedom to resample. Also, a step based path planning is to be developed for low distances of the ball to precisely calculate a position for the robot foot to put on the ground. This can greatly increase effectiveness of positioning algorithm near the ball and prevent over-shoots and under-shoots which may lead to loss of the ball or an unsuccessful kick. As well, a scripting language for describing strategies and scenarios would be usefull in current behavior structure. Learning methods and statediagram-to-code generator can also be employed for task selection of the high level behavior.

References

[1] T. Rfer, T. Laue, J. Mller, A. Burchardt, et al, B-Human Team Report and Code Release 2010, http://www.b-human.de/downloads/bhuman10_coderelease.pdf

[2] Thomas Rfer, Tim Laue, Judith Mller, Michel Bartsch, et al, B-Human Team Report and Code Release 2013, http://www.b-human.de/downloads/publications/2013/CodeRelease2013.pdf

[3] Majid Lashgarian, Mohammad Shafiei R. N., Mohammad Ali Zakeri Harandi, et al, MRL-SPL Team Description 2013, Standard Platforf League, http://www.mrl-spl.ir/downloads/MRL-SPL_TDP_2013.pdf

[4] J. Bruce, T. Balch and M. Veloso, "Fast and Inexpensive Color Image Segmentation for Interactive Robots," in IEEE/RSJ International Conference on Intelligent Robots and Systems., 2000.

[5] M. Shafiei Rezvani Nejad, "Color Calibration Free Image Processing Using Bayes Classifier On Humanoid Robots Soccer Scenario", Bachelor Thesis, Qazvin Azad University, 2014.

[6] Andreas Seekircher, Tim Laue, and Thomas Rofer: Entropy-Based Active Vision for a Humanoid Soccer Robot, 2010.

[7] Max Risler Thesis, Behavior Control for Single and Multiple Autonomous Agents Based on Hierarchical Finite State Machines, 2009.

[8] Monica N. Nicolescu and Maja J. Mataric, A Hierarchical Architecture for Behavior-Based Robots, In Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems Bologna, ITALY, July 15-19, 2002, 227-233

[9] Endel Tulving, Organization of Memory, Chapter 10, Episodic and Semantic Memory, Department of Psychology, Yale University, New Haven, Connecticut, 1972.

[10] Reis, L. P. and Lau, N., COACH UNILANG A Standard Language for Coaching a (Robo) Soccer Team, in Andreas Birk, Silvia Coradeschi and Satoshi Tadokoro, editors, RoboCup2001: Robot Soccer World Cup V, Springer Verlag LNAI, Vol. 2377, pp. 183192, Berlin, 2002, ISBN 3540439129.

[11] 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.