MRL Extended Team Description 2019

Amin Ganjali Poudeh, Vahid Khorasani Nejad, Arghavan Dalvand, Ali Rabbani Doost, Moein Amirian Keivanani, Hamed Shirazi, Saeid Esmaeelpourfard, Fatemeh Rashnozadeh, Sina Mosayebi, Meisam Kassaeian Naeini, and Aras Adhami-Mirhosseini

Islamic Azad University of Qazvin, Electrical Engineering and Computer Science Department, Mechatronics Research Lab, Qazvin, Iran
ssl@mrl.ir

Abstract. MRL Small Size Soccer team, with more than ten years of experience, is planning to participate in 2019 world competitions. In this paper, we present an overview of MRL small size hardware and software design. Having attained the third place in 2010, 2011 and 2013, second place in 2015 and first place in 2016 competitions, This year we enhanced reliability and achieved higher accuracy. Due to the major changes in the rules, We modify to the software. Finally, by overcoming electronic and mechanical structure problems, We promoted the ability of the robot in performing more complicated tasks.

1 Introduction

MRL team started working on small size robots from 2008. In 2016 RoboCup, the team was qualified to be in the final round and scored in the first place. In our last competition in Japan MRL team placed in the top 8 teams. This year, we changed the robot mechanically to improve the movement efficiency and ball possession and kicking accuracy, and along with it, we altered the electronic parts that needed to alter with this new mechanic. The main objective of our software is to play a perfect game with 8 robots so that we can dynamically perform an efficient passing play. In the other hand, we want to have the best performance in defense play. In 2019 competitions the main structure of the robots is almost intact but we have minor changes, Figure 1 shows the MRL 2019 general structure of robots.

Some requirements to reach this target are achieved by redesigning the electrical and mechanical mechanisms. Moreover, simple ML and optimization approaches will be employed in the way of more dynamic play.

This paper is organized as follows: First of all, we illustrate our defensive play details and we explain a algorithm for finding the best location for pass in section 2. the new charger board for capacitors and Electrical design including ARM micro controller, and other accessories of robots onboard brain is explained in section 3. Description of new wheels and mechanical structure, which modifies the capabilities of the robots dribbler system, is the subject of section 4.

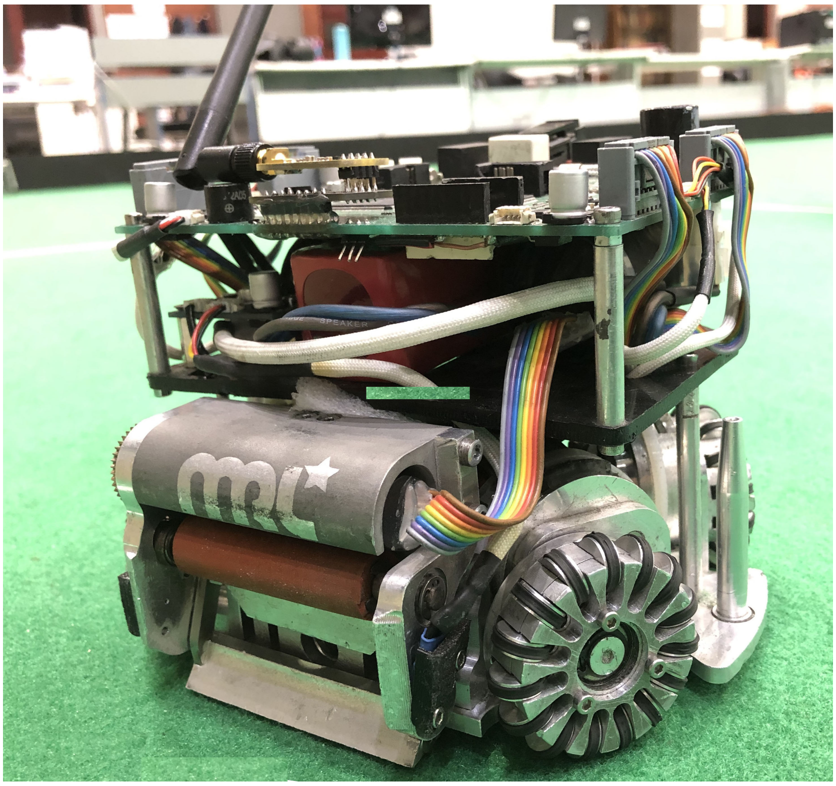

Fig. 1. MRL robot for 2019 competitions

2 Software

In this part the software main objects are presented. It is shown that how our new modifications provide us a more intelligent and flexible game. In this year MRL software team has not changed the AI main structure. The game planner as the core unit for dynamic play and strategy manager layer is not changed structurally, but some new skills and abilities are added to the whole system. In this section, after a brief review about the AI structure, short description of the unchanged parts are presented and references to the previous team descriptions are provided. Finally major changes and skills are introduced in details.

The software system consists of two modules, AI and Visualizer. The AI module has three sub-modules being executed parallel with each other: Planner, STP Software (see [6]) and Strategy Manager. The planner is responsible for sending all the required information to each section. The visualizer module has to visualize each of these sub-modules and the corresponding inputs and outputs. The visualizer also provides an interface for online debugging of the hardware. Considering the engine manager as an independent module, the merger and tracker system merges the vision data and tracks the objects and estimates the world model by Filtering of the system delay using Kalman filter. Figure 2 displays the relations between different parts. In this diagram, an instance of a play with its hierarchy to manage other required modules is depicted. The system simulator is placed between inputs and outputs and simulates the entire environment's behavior and features. It also gets the simulated data of SSL Vision as an input and proceeds with the simulation.

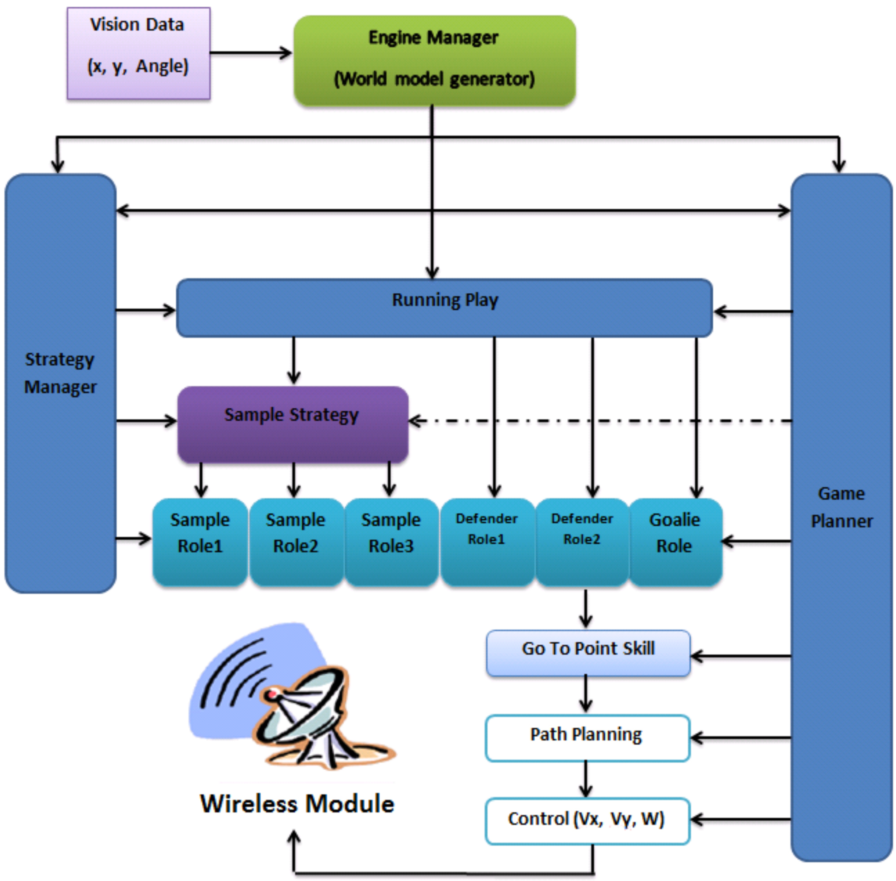

Fig. 2. Block diagram of AI structure

2.1 Defense strategy for opponents free Kicks

Free kicks are important strategies for all high ranked human and robotic football teams. There are many samples in both human and robotic tough games, that scores obtain from free kicks. So, for every team in SSL, it is important to have good strategies for free-kicks granted in different areas of the field. Also, it is important to develop good strategies to defend against opponent free kicks. From 2012 Robocup, MRL team started to design and implement the new defense algorithm. During these years, we have developed and modified the algorithm by feedbacks from Robocup and Iran-open competitions. Following points are noted about this defense algorithm:

- It is structurally static but functionally dynamic. In the other words, without any change in the algorithm, different behaviors against different teams will be shown.
- It is difficult to predict the defender behavior by other teams.
- Expert knowledge is the basis of the algorithm. This advantage gives opportunity to different team to develop their algorithm based on this structure.
- The main disadvantage of this algorithm is its high complexity. This makes the algorithm so difficult to debug/change during the competitions.
- It is highly integrated and interconnected algorithm. So, we cannot change any part or focus on any situation without care about the whole algorithm and other situations.
- Aim of this algorithm is first protecting our goal against opponent kicks, and second trying to catch/cut the ball.

The free kick defense algorithm starts with indirect or direct kick command from referee box. It ends at any of following two situations:

- New referee command is received.
- One of our robot owns the ball, i.e. ball is in front of the robot with distance less than 13 cm and speed less than 0.5 m/s.

Between the start and end of the free kick defense algorithm, following steps in three layers (Play, Roll and Skill) are repeated at each frame (16 ms), respectively. Play layer:

- Step 1: Calculating opponent robots scores. This score is a number between [0,1] where 1 shows the most important robot. For each robot, score depends on the robot position, robot velocity vector, ball position and ball velocity vector. The procedure of finding scores has been introduced in details in MRL-SSL ETDP 2011[4].
- Step 2: Finding possible opponent robots contributing in free kick attack, i.e. each robots with score more than 0.8 (adjustable).
- Step 3: Considering proper defending roles (e.g. Defender, Marker, Regional and) based on the number of attacking robots from step 2. Also for each role the corresponding opponent robot (if applicable) is consider in this step. Different roles are discussed later in this section.
- Step 4: Assigning roles of step 3 to our robots. The role assigner module works based distance between robots positions and desired target position, while considering pervious role of each robot and allowable switches. Allowable switches between roles are introduced after discussing the roles.

Role layer:

- Step 5: Finding the proper state for each considering role in step 3. Proper state depends on the situation (position and velocity vector) of the opponent target, situation of the ball and position of defending robot which is assigned to this role in step 4. Different states of each role, are defined later in this section.

Skill layer:

- Step 6: Calculating final commands for each robot. These commands are based on the skill that are selected for each robot. Skills are simple and small behaviors that are intended to be used in different parts of the whole structure of MRL software. Since the most used skill in free kick defense algorithm is simple GoToPoint skill, we do not discuss about skills here.

In the following we first introduce different roles and related states contributing in the free kick defense algorithm. Then, different plays based on the number of attackers is discussed. Finally, some notable points about the algorithm are mentioned.

2.1.1 Roles in the free kick defense algorithm

As discussed in step 3 of the algorithm, different roles are selected at each play. Following, all roles are listed with a short description about it and related states. Note that each robot can have only one role at each frame.

1. **Defenders1-3:** The behavior of this Role is positioning on the line of penalty area and try to block the straight line of the target to the empty space of goal line. This kind of blocking, results in less movement of the defender in comparison of target motion. To do this, we draw a line from the ball to the center of the blank part of the goal (considering only other defenders and Goalie roles), and the robot is placed on the line of the penalty area and at the intersection with this line. Defender2 and Defender3 contain one state, while Defender1 can select one of two states:
   - Ball: When the ball speed size is less than target robots speed size, position point of the Defender1 is calculated based on the ball position.
   - Robot: When the ball speed size is more than target robots speed size, position point of the Defender1 is calculated based on the robots position.
2. **Goalie:** The goalkeeper is the most stated role. At each state, it shows different behavior as follows:
   - Normal: In this state, Goalie treats depend on Defender1, and covers parts of the goal that are not covered by this role.
   - inPenaltyArea: When the ball is in our penalty are with margin +10 cm, Goalie tries to catch the ball.
   - KickToRobot: When the ball moves faster than 0.5 m/s and its velocity vector intersects the current position of the goalkeeper, Goalie role tries to stop and remain at the position.

- KickToGoal: When the ball moves faster than 0.5 m/s and its velocity vector intersects the goal line, Goalie tries to go to the nearest point of the ball speed line that is placed in the penalty area.
- BallInStartOfChip: When chip detector module shows that ball is in a chip kick, we active this state for 30 frames. During this state, Goalie take position at the goal center.

3. **Regional1-2:** These roles are covering the empty area of the field for possible future blocking of the moving opponents. These roles' positions are in front of the penalty area (like Defenders) in the middle of the clearest area. In the case that two regional roles are selected by the algorithm, first Regional1 is placed and then Regional2 is placed considering the position of Regional1.
4. **Markers1-3:** The task of these roles are marking the opponent robots in the man to man manner. Different states of these roles tries to find the best action based on the situation of the target robot, ball and assigned robot.
   - NearBlock: When the target robot does not move (speed size¡0.5 m/s), Marker tries to be so close to the target robot and touch it.
   - FarBlock: When the target robot moves along the width of the field, Marker tries to block the goal from distance 0.5-1 m from the target robot.
   - InTheWay: When the target robot moves toward our goal, Marker tries to go to the nearest point in its path.
   - Cut: When the target robot does not move and ball moves toward the target robot and Marker placed near the target robot, Marker forgets about blocking the goal and tries to catch/touch the ball. Simply, it moves to the point in the line of ball and target robot at distance 30 cm from target robot.
5. **Active:** In this roll, the closest robot to the ball try to catch the ball and after that moving the ball to the opponent's goal.
6. **Stop cover:** At the start of the free kick when ball is not touched by opponent team, this role stands at 60 cm from the ball and block the direct kick to our goal.
7. **Positioner:** This role is activated when the opponent team attacks with less number of robots. The role contains simple positioning in pre-defined points usually in the opponents half field. Now, this position is set as 2 meters from middle of the field to the opponent's goal, and 1 meter from the longitudinal line. we have two points in the field with this description, left or right side of the field. we choose the side that contains less opponent's robots.

2.1.2 Roles and targets selecting Here, we discuss the most important part of free kick defense algorithm mentioned in step 3. In this step, roles and their related targets (if applicable) are selects. Selection is based on the number of attackers (step 2) and state of the ball (is not moved or is moved). Roughly speaking, opponent with the highest score will defense with Defender1 and the second-high scored opponent with Defender2. After that, we select Markers and so on to the last score.

In the following, there are charts for each number of attackers in which the roles are arranged in order of concessions. The sequence is important, because if the number of our robots is reduced from 8, the roles will be removed from the end. Each block in these charts contains a role of our robots and opponent target robot by its score rank. In every part an example for more clarification is illustrated.

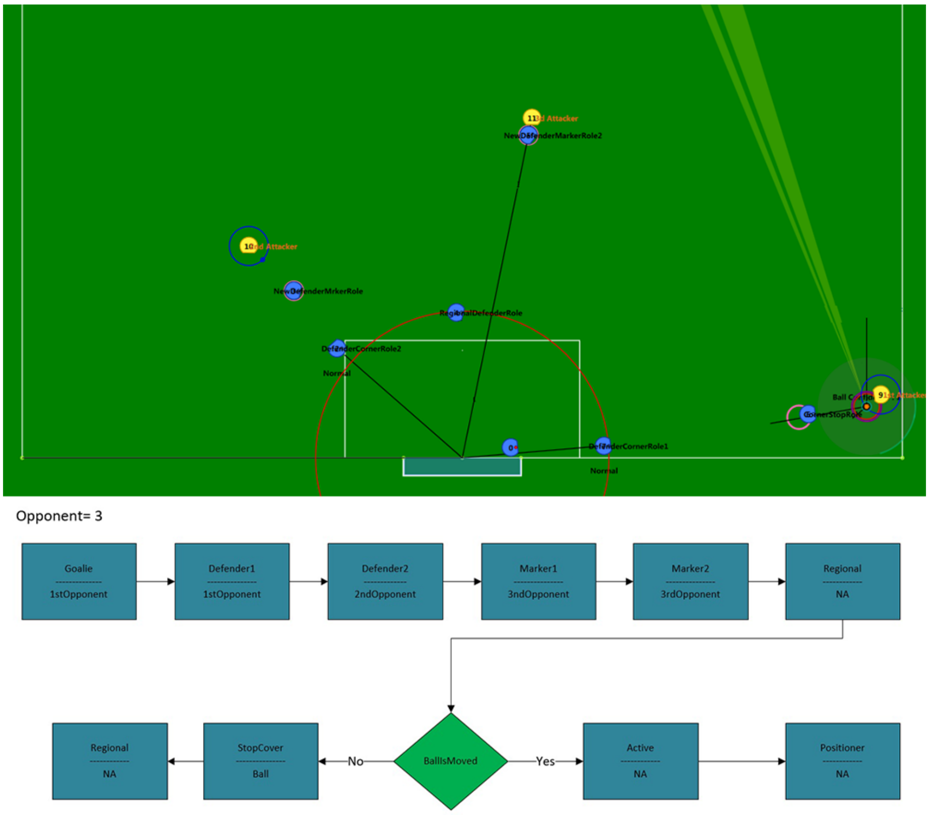

Fig. 3. Role selection if number attackers are 3

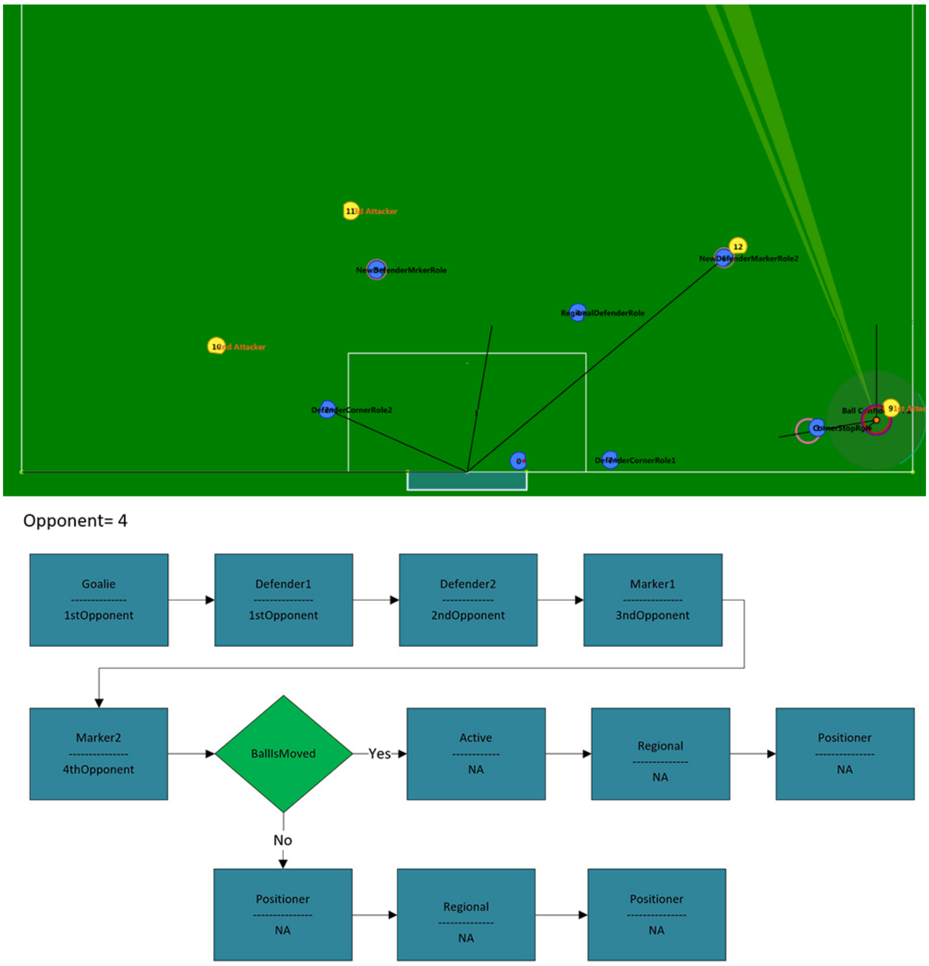

Fig. 4. Role selection if number attackers are 4

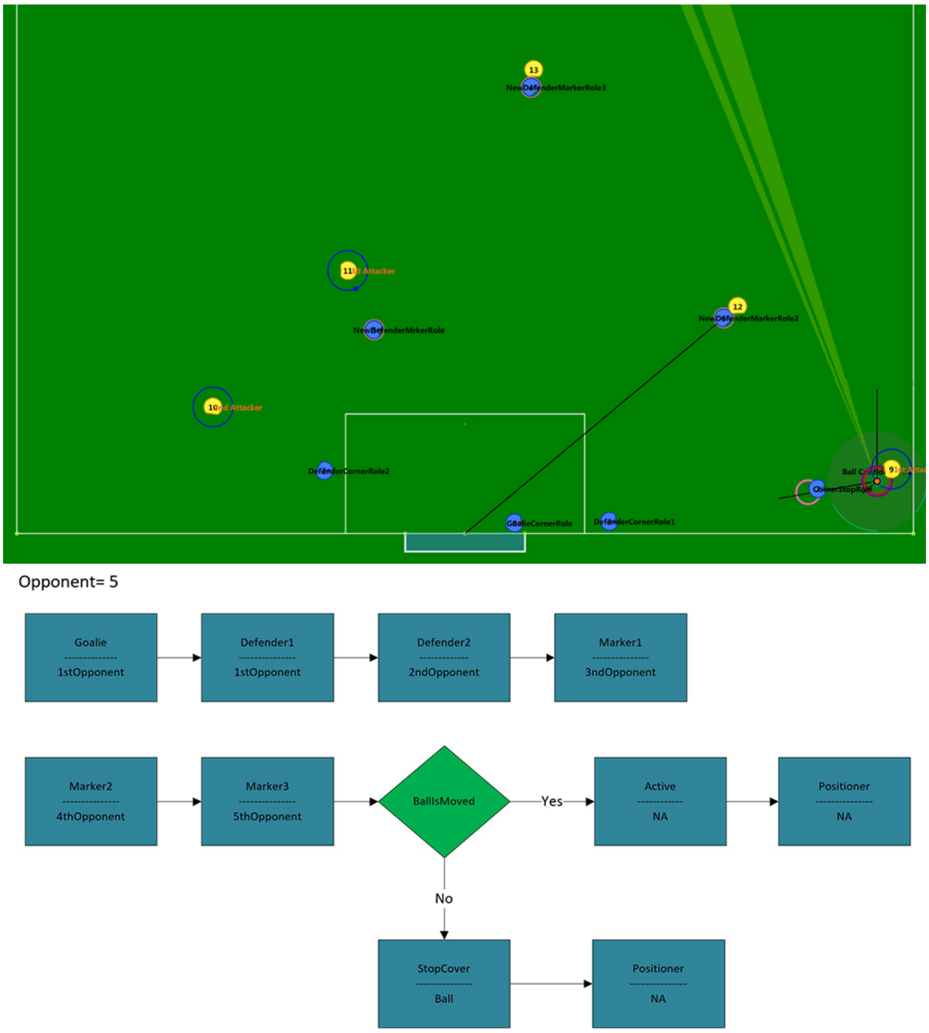

Fig. 5. Role selection if number attackers are 5

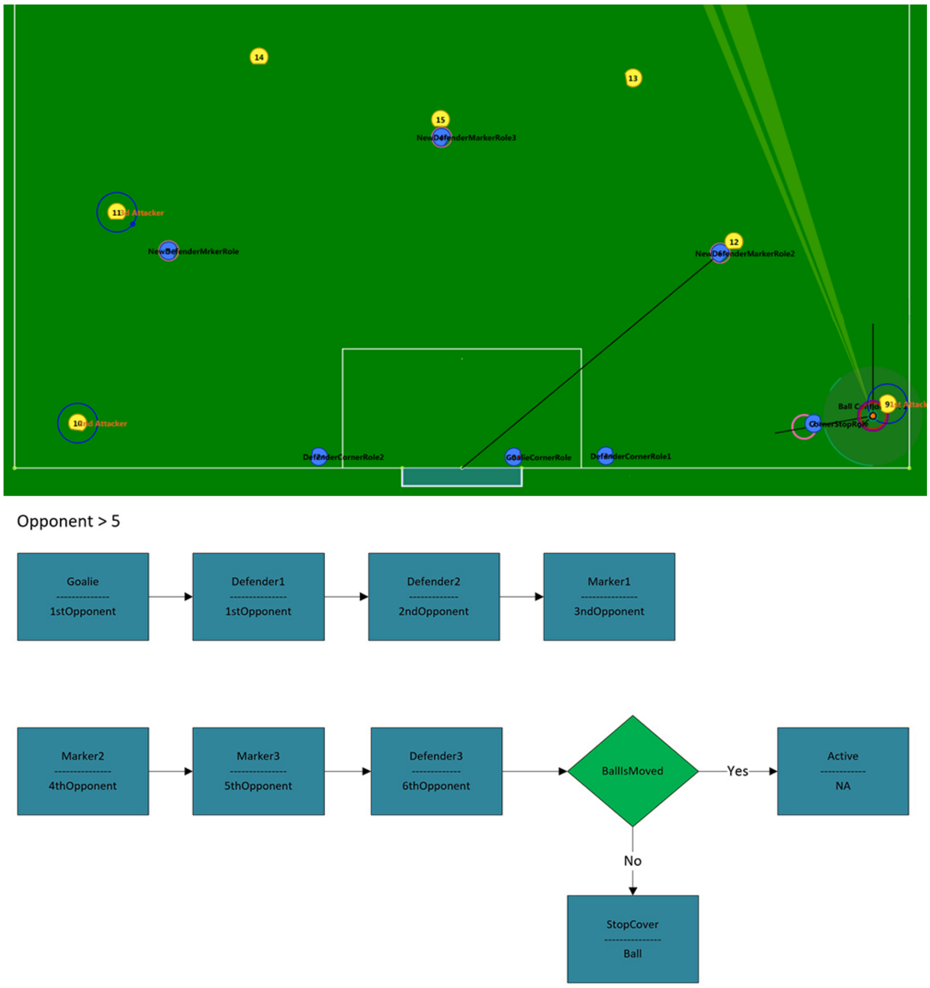

Fig. 6. Role selection if number attackers are greater than 5

2.1.3 Notable points

Roles switching rule: In step 4 of the algorithm, we assign each role from the previous sub-section to our robots. Since the algorithm is highly dynamic and at each frame roles and its targets are calculated, it is important to set a good role assigner that avoids unnecessary and conflicting movements. To this end, we select the robots for roles based on distance from the current positions. Moreover, we divide the roles based on their possible positions in two groups:

– First group contains Defenders1-3 and Regional roles.

– Second group contains Markers1-3, Stop cover, active and positioner.

At each frame role switching for a robot is constrained in its pervious group. For example, if a robot took the regional role in the previous frame it is allowed to take the same role or Defenders1-3 at this frame if the play is not changed.

Importance of hysteresis: In this algorithm like any other algorithms that contains if conditions, it is so important to use suitable hysteresis for each state transition. In the proposed algorithm, one should care about changing number of attackers, states of each role and role assignments. Using proper hysteresis for each condition, avoid multiple switching especially at the borders of the conditions.

2.2 Algorithm for finding the best location for pass

Sometimes in competitions, under some circumstances, ball owner robot prefers pass to other actions. Finding a position for pass is very meaningful and valuable. In order to find the position for pass, first we select a rectangular area in play field then we make a list of point in that area by creating a grid with Width Step and Height Step parameters, we calculate two score for each of that points, first one is a score for catching ball and then rotating and shoot called CRScore, second one is for shooting the ball with one touch called OTScore.

$$CRScore = \frac{\alpha \ \beta \ e \ f \ h \ i}{time} \tag{1}$$

$$OTScore = \frac{K_\theta \ \alpha \ \beta \ e \ f \ h \ i}{time} \tag{2}$$

where: $shooter$ = the robot that receive the pass
$shootTarget$ = target of shooter (usually center of the goal)
$oppMaxDis$ = the furthest distance of opponent robots to the point, in meter
$oppMinDis$ = the nearest distance of opponent robots to the point, in meter
$shooterDis$ = distance of the shooter to the point in meter
$ballPointDis$ = distance of ball to the point in meter
$pointTargetDis$ = distance of the point to shootTarget in meter
α = the angle between the ball and the two sides of the maneuverable area for the robot (in the direction of perpendicular line of the ball speed vector to that point) in radian
β = available angle for shoot (Including obstacles) from the point in radian
θ = angle between the point to shootTarget and ball to passTarget in degree
e $= \frac{oppMaxDis}{oppMinDis}$
f $= 1 - ((\frac{minimum(10,shooterDis)}{10})$
i $= (\frac{maximum(minimum(ballPointDis,3),0.7)-0.7)}{2.3}$
$time$ $= \frac{ballPoint}{passSpeed} + \frac{pointTargetDis}{shootSpeed}$
K_θ $= 1$

if $\theta < 45 \longrightarrow K_\theta = \frac{\theta\ K_\theta}{45}$
else if $\theta < 90 \longrightarrow Kc = 1$
else $\longrightarrow Kc = 0$

we choose the point with maximum amount of scoreOt or scoreCr, then pass task will be executed.
for performing pass we need to calculate pass speed and time of reaching the shooter to the pass target, these calculations was explained in synchronization algorithm [1].

3 Electronics

The electronic part of the robot consists of 2 boards. Main board and charger board. The main board contains a STM32 ARM microcontroller. The primary task of this board is to communicate with the software server, controlling and driving motor and ball sensor implementation. Charging and discharging of kick capacitors are goals of the charger board. the charger board and how it works will be explained.

3.1 Capacitor charger board

For every kick we need a process,The charger board is an important part of this process.
This board should handle these tasks:

- Charge the capacitors and keep them charged
- Discharge them in the right way

In the old range, capacitors were fully charged at 8 or 9 seconds. Which is a long time and caused trouble in consecutive kicks that takes less than 9 seconds.
One of the most important factors in kicking the ball is the equality of practical velocity of kick and the theoretical velocity from software. It was not good enough in our older charger board. in order to solve these problems, a new PCB was designed, features of the new board is explained below:

- Due to the low quickness in charging capacitors, in the new board we use a transformers called DA2033, which reduces charging time from 8-9 seconds to 4 seconds. This transformers works with a charging controller called LT3751. Through this piece, we can control the amount of charge and charge process functions.
- We used MOSFET IXYS for depletion the capacitors in the old board, in order to optimize this depletion, the IGBT-IRG4PC50 transistor with better guidance and switching is replaced.
- By changing the mainboard micro-controller from the LPC2378FB to the STM32F746, the velocity of the robot kicks became very close to the same command.
- By using the LT3751 we can get feedbacks, e.g, beginning of charge cycle, Reaching the maximum voltage of capacitor, etc.

Changes made to the micro-controller of mainboard, caused approaching the practical velocity and theoretical velocity to each other, with these changes, we will able to control the depletion of capacitors very well.
These achievements gives us the advantages of having accurate kicks in the competition. Figure 7 shows the comparison of the old board and the new board
.

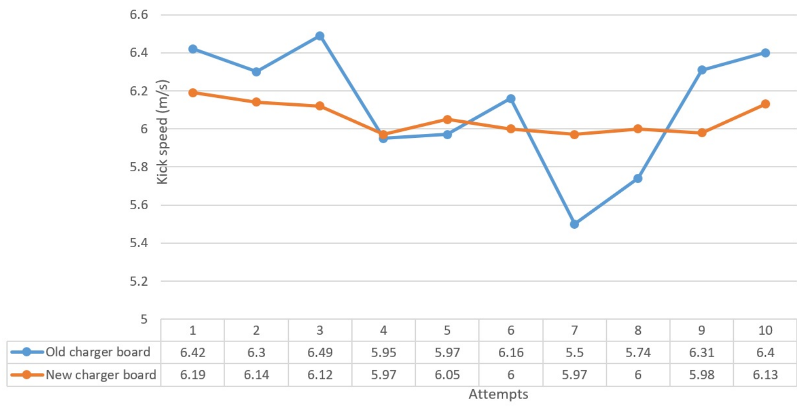

Fig. 7. Comparison of theoretical data and practical data in old board and new board

4 Mechanical Design and construction

Typically, the main portions of the mechanical structure of a small size robot, include 4 wheels, two kickers, a dribbler and the motion transformer system. Regarding the league rules, the diameter of the robot is $179mm$ and the height is $140mm$. The spin back system conceals 20% of the ball diameter in the maximum situation.

Due to some drawbacks in the previously proposed design, we have decided to improve both the mechanical design and the construction materials. Main changes in the mechanical structure of the robot are described in the following paragraphs. The other parts are the same as 2014 robot described in MRL-SSL ETDP2014[1].

4.1 Railway mechanism for Dribbling system

A piece that was used to move the Dribbling system(shows in Figure 8) forward and backward at the desired angle, was a non-industrial and hand-crafted mechanism, in the form of a frame in which the rhombus-like rails rolled inside.

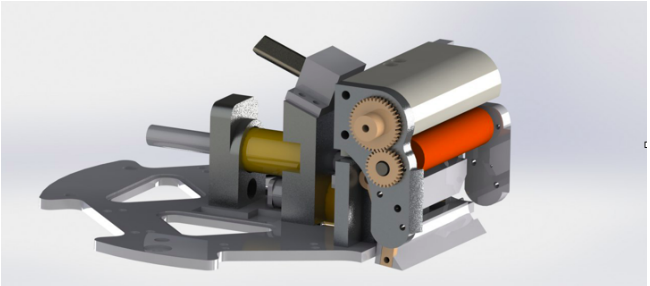

Fig. 8. old Dribbling system and kicker

In order to lower friction, we put linear bearings inside the frame and by doing so, recoil arising between parts of the mechanism, and the entire system did not work arbitrarily.
It also had problems, such as when the dribbling system hit the barrier, friction prevented the Dribbling system from returning to the right place.
The backlash also caused the Dribbling system to move to the sides and this action will change the position of the sensor which was connected to the dribbling system and creating problems in identifying the ball.
To solve these difficulties, we applied linear guideways, these parts with low friction resistance and a small amount of advance force will cause wagons moving along the rails, Each of them was replaced by the frame and the rhombus-like rails, respectively. One of the advantages of this new system is that the low friction of the system makes it quick to fit into its proper place.
Due to the small size of the robot, we selected MGN12 Miniature Linear Guideway.

Model No.	Dimensions of Assembly (mm)			Dimensions of Block (mm)									Dimensions of Rail (mm)							Mounting Bolt for Rail	Basic Dynamic Load Rating	Basic Static Load Rating	Static Rated Moment			Weight	
	H	H_1	N	W	B	B_1	C	L_1	L	G	G_M	Mxl	H_2	W_R	H_R	D	h	d	P	E	(mm)	C(kN)	C_0(kN)	M_R N-m	M_P N-m	M_Y N-m	Block kg	Rail kg/m
MGN 12C	13	3	7.5	27	20	3.5	15	21.7	34.7	-	Ø2	M3x3.5	2.5	12	8	6	4.5	3.5	25	10	M3x8	2.84	3.92	25.48	13.72	13.72	0.034	0.65

MGN12

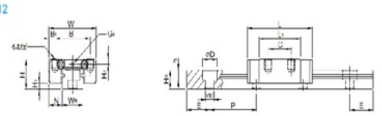

Fig. 9. MGN12 Miniature Linear Guideway details

4.2 Wheels gearbox

The old wheel gearbox was designed with a ratio of 72:20, a module of 0.5 and a backlash of 0.2. In order to increase the precision of the robot movement, we designed a new gearbox with an almost-zero backlash. By making this change, accuracy increases in the movements with directional change.
enhancing backlash will cause lost motion between motor and gearbox, which makes it difficult to achieve precise positioning and certainly reduces the accuracy of movement.
We changed the material of both gears from phosphor bronze to steel, because it is more resistant to corrosion.

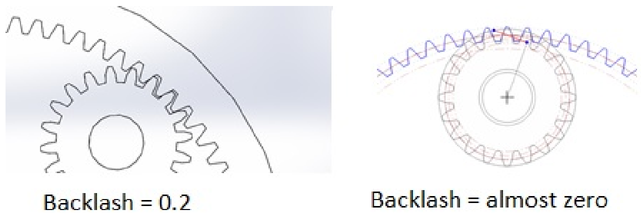

Fig. 10. old gearbox against new gearbox

4.3 Direct kicker

The former plunger did not have a groove or guide and with the arrival of the solenoid force, it spins around its axis due to its high backlash. This issue itself causes problems during the game, e.g., after the force enters while the iron part return into the Solenoid, it stuck to the front of the Dribbling system.
This issue raises problems during the competition, e.g., when plunger wanted to come back into the Solenoid, it stuck to the front of the Dribbling system and creating problems in the next kick.
So, to prevent these movements of the Plunger around its axis, we used two grooves as a guide around Plunger, preventing excessive spin and Plunger rotation.

References

1. Ganjali Poudeh, A., Beik Mohammadi, H., Hosseinikia, A., Esmaeelpourfard, S., Adhami-Mirhossein, A.: MRL Extended Team Description 2014. Proceedings of the 17th International RoboCup Symposium, Jao Pesoa, Brazil, (2014).
2. Ganjali Poudeh, A., Asadi Dastjerdi, S., Esmaeelpourfard, S., Beik Mohammadi, H., Adhami-Mirhossein, A.: MRL Extended Team Description 2013. Proceedings of the 16th International RoboCup Symposium, Eindhoven, Netherlands, (2013).

3. Adhami-Mirhosseini, A., Bakhshande Babersad, O., Jamaati, H., Asadi, S., Ganjali, A.: MRL Extended Team Description 2012. Proceedings of the 15th International RoboCup Symposium, Mexico city, Mexico, (2012).
4. Ahmad Sharbafi, M., Azidehak, A., Hoshyari, M., Bakhshande Babersad, O., Esmaeely, D., Adhami-Mirhosseini, A., Zareian, A., Jamaati, H., Esmaeelpourfard, S.,: MRL Extended Team Description 2011. Proceedings of the 14th International RoboCup Symposium, Istanbul, Turkey, (2011).
5. Zarghami, M., Fakharian, A., Ganjali-Poudeh, A., Adhami-Mirhosseini, A.,: Fast and Precise Positioning of Wheeled Omni-directional Robot With Input Delay Using Model-based Predictive Control. Proceedings of the 33th Chinese Control Conference (CCC), pp. 7800–7804, Nanjing, china, (2014).
6. Browning, B., Bruce, J.; Bowling, M., Veloso, M.M.: STP: Skills, Tactics and Plays for Multi-Robot Control in Adversarial Environments. Robotics Institute, (2004).
7. InvenSense Inc: MPU-6000 and MPU-6050 Product Specification Revision 3.4, 2013.
8. Utilizing a Vishay IrDA Transceiver for Remote Control. Application Note by Vishay Semiconductors, Doc. No. 82606, (2014).
9. Nordic Semiconductor: nRF24L01+ Product Specification v1.0,(2008)
10. LLC Allegro MicroSystems: A3930 and A3931, Automotive Three Phase BLDC Controller and MOSFET Driver, A3930 and A3931 Datasheet
11. LLC Allegro MicroSystems. A3930 and A3931 Automotive 3-Phase BLDC Controller and MOSFET Driver, A3930-1 Demo board Schematic/Layout