PARSIAN
Extended Team Description for RoboCup 2016

Mohammad Mahdi Rahimi, Mohammad Mahdi Shirazi, Seyede Parisa Dajkhosh, Alireza Zolanvari, Maziar Arfaee, Hamidreza Kazemi Khoshkijari, Amirhossein Abbasi Fashami, Alireza Saeidi Shahrivar and Mohammad Azam Khosravi.

Electrical Engineering Department
Amirkabir Univ. Of Technology (Tehran Polytechnic)
424 Hafez Ave. Tehran, Iran
parsianmembers@gmail.com

Abstract. The Parsian team placed top eight teams in the Small Size League of RoboCup 2015. In this paper, we present our robots' current design in mechanical, electrical and the team's recent work on the offensive and defensive tactics, low level skills. Among the offensive tactics, we introduce new features in our visual planner and that in both play on and play off then We present a change in defense mark system that increase reliability and robot ball-manipulation skill that keep ball till a suitable situation occurs. Finally, we describe improvements to the ball and robots state estimation algorithms by profiling robots and field.

1 Introduction

The Parsian small size team, founded in 2005, is organized by electrical engineering Department of Amirkabir University of Technology. The purpose of this team is to design and build small size soccer robots compatible with International RoboCup competition rules as a student based project. We have been qualified for ten consequent years for RoboCup SSL. We participated in 2008 - 2015 RoboCup competitions. Our most notable achievements was PARSIAN's first place in RoboCup 2012 SSL's Passing and shooting technical challenges and RoboCup 2013 SSL's Navigation challenge. In this paper we first introduce our robots' new Mechanical design (section 2), some changes in electrical design will be discussed in section 3 and control system and software will be covered in section 4 and 5.

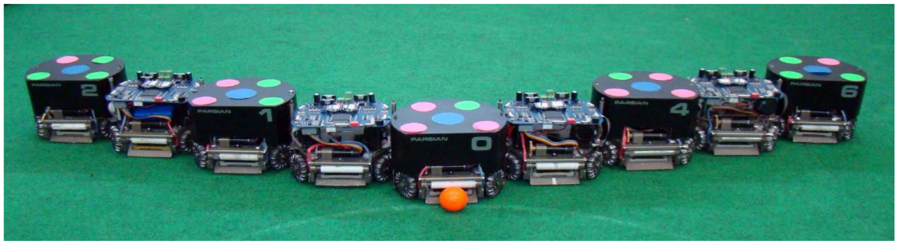

Fig. 1. Our Robots

2 Mechanical Design

In this part an introduction is going to be rendered on the mechanical structure of the new robot designed for RoboCup 2016. In this design the focus has been on two parts: the dribbler part and easy access to the different parts of the robot which will be explained in detail in the following sections.

2.1. Dribbler

the new change that we made for the dribbler is adding a new part to it that cover and route shot sensor cables in order to avoid collision with wheels or other robots that cause damage to cable.

2.2. easy access

The major change that we made for this year robots is using an integrated system that attached all electrical board, batteries and capacitors together. with this system repair and maintenance that need separation motors from electrical parts is now a lot faster and easier.

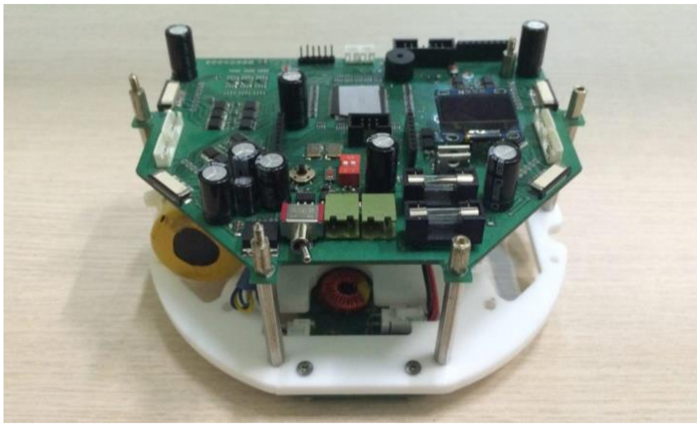

Fig.2. New integrated mechanical design

2.3. Robot's specifications

Robot Diameter	179 mm
Robot Height	130 mm
Ball Coverage	18 %
Max Linear Velocity	4.1 m/s
Weight	2.1 kg
Maximum kick speed	12 m/s
Limited kick speed	7.8 m/s
Maximum chip kick distance	6 m
Maximum ball speed catching	6.5 m/s

Table 1. Robot's specification

3 Electrical Design

3.1 Main Board

The main board which had been used last year was fully functional, but it had some defects. For instance, the only way to represent events, problems, warnings, etc. was showing them on some LED's or send them with wireless network to the debug computer, also it didn't have any non-volatile memory to save failures that were being happened during the game.

The new main board features and technical specs will be discussed in the next part.

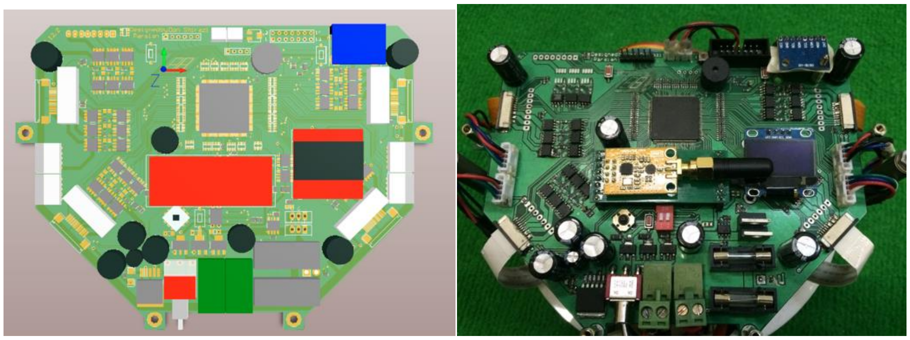

Fig3. (Left) 3D design & (Right) real hardware

3.2.1 Processing unit

Like last year an xc3s400 FPGA is used as main processor. This year an ATMEL^{TM} **ATXMEGA128A4U** micro controller is added as co-processor.

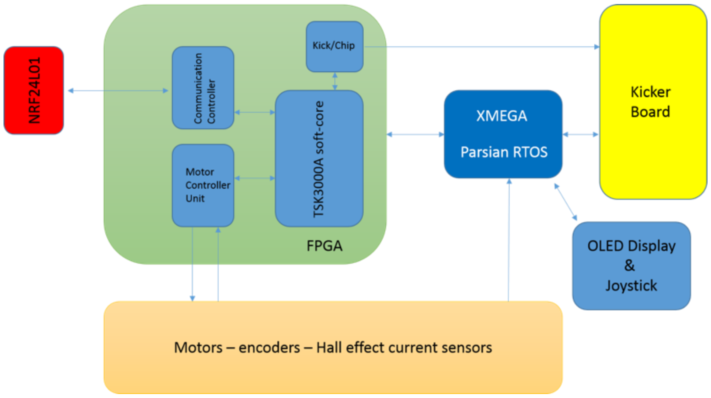

Fig. 4. New Main-Board Block Diagram

3.2.2 Current control and power management system

This year each motor has a dedicated current sensor, also two current sensors are used for measuring whole board current and kicker board current. All of the current sensors are connected to micro controller's ADC pins.

Last year the board had two keys, one for turning on logical part and the other for supply power to the motors, but this year we use a power mosfet that is being controlled by micro controller. Now the micro controller can turn on/off the power logically. For instance, when the battery is critically low the micro controller decides to turn the board off.

3.2.3 Communication system

Like last year NRF24L01 is used as wireless communicator, but this year two NRFs are used in each board to apply achieve full duplex two way communication.

Also a new industrial level communication base was constructed to prevent damages that could be applied to communicator board.

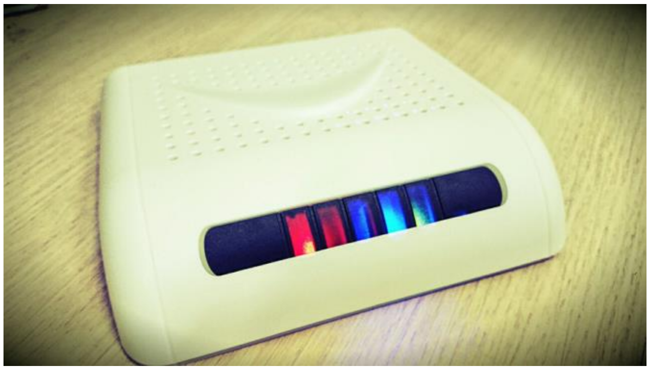

Fig. 5. Industrial Level communication board

3.2.4 User interface unit

This year a keyes^{TM} 128x64 pixels OLED display is used as main display that shows useful information like battery level, robot ID, network channel, motors current and etc. Parsian RTOS is developed for manage these data and control robot’s state. It measures motors current, battery voltage, whole board current and send over current data to FPGA, manage network channel and robot’s ID, control the kicker board voltage, and turn on/off the board intelligently.

For navigate in Parsian RTOS menus a nokia N73 joystick is used.

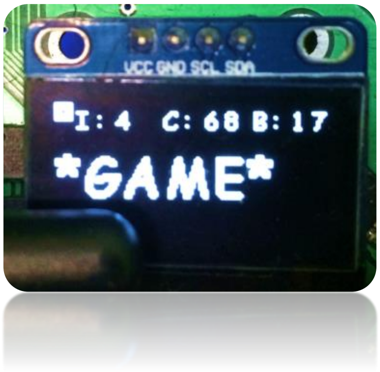

Fig. 6. Keyes OLED Display

4 Control System

Rolling over of robots is one of the most critical problems that we have in RoboCup 2015, so we should find a proper solution for this problem. For giving the most efficient solution, we must find the main reason of the event. In order to reach to this goal, we analyze the forces that affect the robot and some other effective factors like the shape of the robots and trajectory of them. We conclude that the worst case that this event may occur is the situation that the robot moves forward (direction of the robot and direction of the movement is the same) and if we control this situation all of other situations will be controlled, so we just analyze this situation and extend it for all other ones. As you see in the following figure, if we imagine that we apply force to the robot in order to move in the first direction and after a while apply force in order to move the robot in the second direction, because of inertia robot will like to continue its movement in the first direction and we apply the second force in opposite direction, so the torque is be resulted and the robot will roll over.

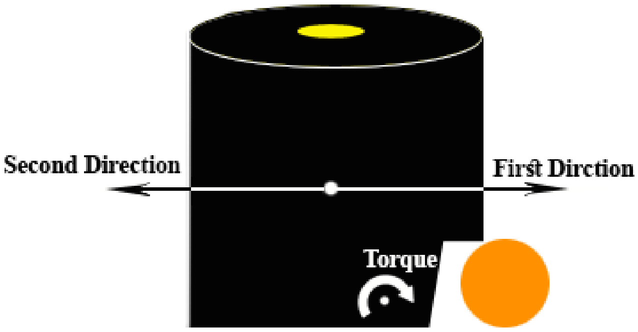

Fig. 7. Robot and applied forces

In fact in these situations we apply forces to the robot like a square wave that is shown in the following figure.

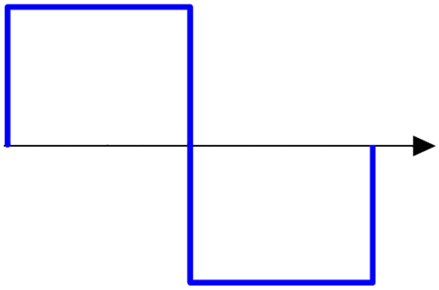

Fig. 8. Applied forces diagram

The suggested solution is to smooth the trajectories and as a result, smooth the square wave that shows the forces. One of the best way to do this, is finding the trajectory that have the minimum jerk to reduce the strike that make the robot rolling over. In 1984, Neville Hogan claim that the trajectory with the minimum strike can be a function of jerk. If we show the position by x(t), jerk can be calculated by using the following formula:

$$jerk : \dddot{x}(t) = \frac{d^3x(t)}{dt^3}$$

If we want to have minimum strike in going from x_1 in t_1 to x_2 in t_2, we should solve an optimization problem. In other words, we should minimize a cost function. Finding the best cost function, play the most important role in solving this problem.

Michael Arbib Et al find the most proper cost function in order to find the minimum jerk trajectory. The following control rule ensure us to go from origin to the target by minimum strike and in fact calculate the smoothest trajectory.

$$\dot{\mathbf{q}} = \begin{pmatrix} \dot{\mathbf{x}} \\ \ddot{\mathbf{x}} \\ \dddot{\mathbf{x}} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{-60}{D^3} & \frac{-36}{D^2} & \frac{-9}{D} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \\ \ddot{\mathbf{x}} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{60}{D^3} \end{pmatrix} \mathbf{x_f}$$

In this equation, X_f is the target and D is the time that we need to go from origin to the goal point. By knowing the position, velocity and acceleration, this rule will calculate the changes that we need to apply to these parameters in order to have the smoothest trajectory. In the following figure, a square wave before and after applying this control rule is shown.

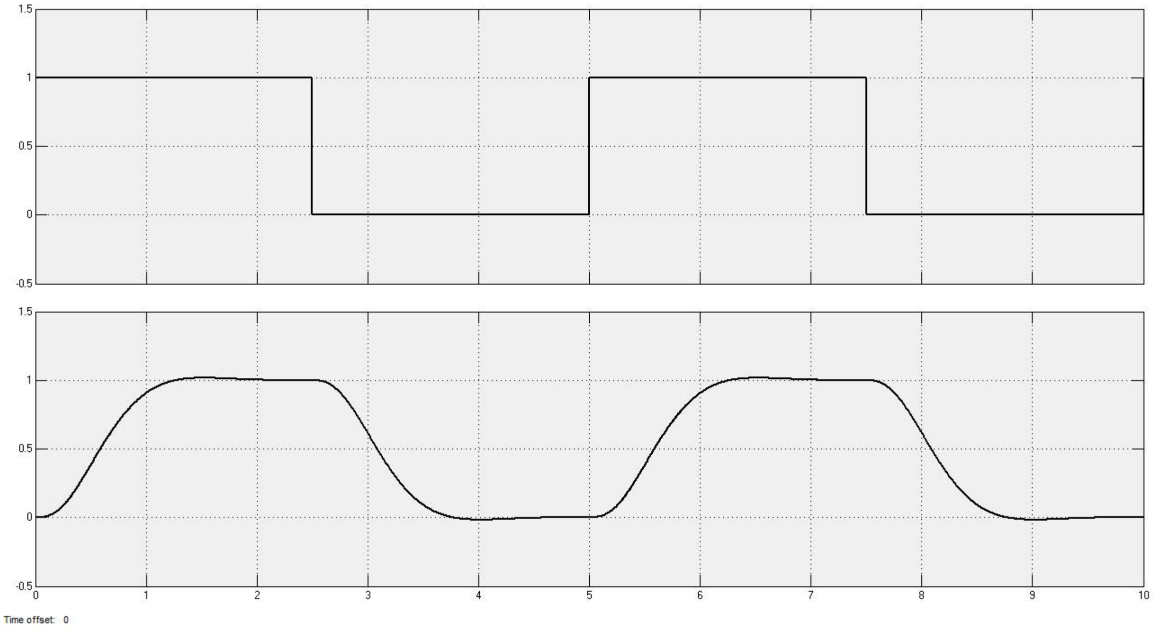

Fig. 9. Applied forces diagram before and after applying minimum jerk trajectory control rule

5 Software

5.1. Architecture

This year the software architecture has some minor changes that will be discussed in the next part. Here is The Parisan Software architecture chart (Fig.10).

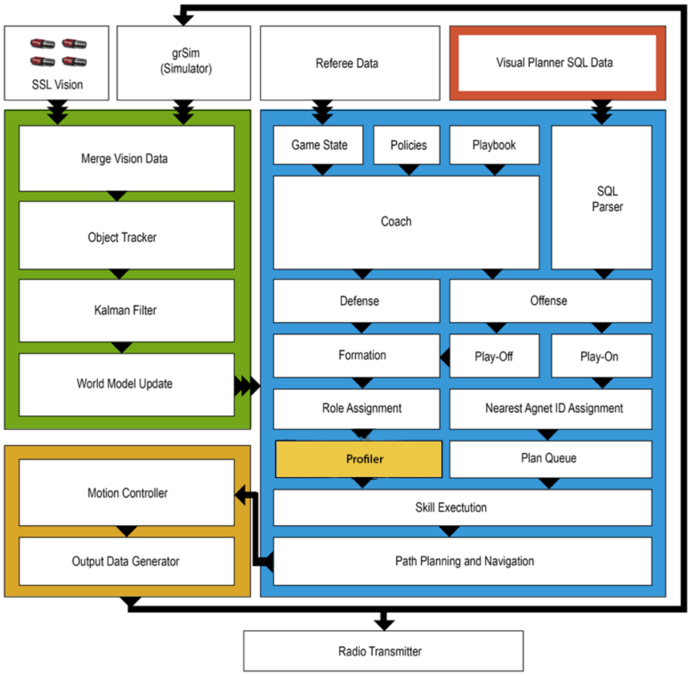

Fig.10. Software chart

5.2. New Visual Planner

According to Parsian 2015 TDP the visual planner is a graphical and very user friendly planner that you can make one level plan for game on. since then we added some new features and made a bit change in game on region dividing that presented in below:

5.2.1 Play off and multi-level plans:

Since the play offs (when ball is stopped for a free-kick) are not so dynamic and looks like a repetitive situation we decide to make some semi-static plays that contain all of the different situations and then we run them completely not step by step like when the game on till we reach to the situation that we want(mostly is till goal happen) or one step doesn't happen the way it supposes to be then we deny remains of plan and choose the best plan for that situation from game on Playbook;

5.2.2 NewPlayOn Planner:

In last version of visual planner we have 6 static region for ball and 10 static region for robots that was so hard to make plans for all of the situation of course if you don't make a plan for a situation, AI automatically assigns another play to that situation that is closer that others.

In new planner we decrease the quantity of regions and usage of semi-statics play just for very likely situation and increase using of dynamic and AI algorithm for have more unpredictable play.

5.2.3 New features and options:

As we make some new skills such as passing a ball in front of another robot when the passer robot is in back so we should modify planner in order to use new skills and for make plays more unpredictable with don't give a robot a static order in same situation we give some weight to each possible work on that situation run a random function to choose the skill that should execute (each skill that have more weight have more chance to be selected).

5.2.4 Open sourcing:

The latest release version that we use in Robocup 2015 are now open sourced and you can download it from here : http://parsianrobotics.aut.ac.ir/

5.3 Profiler:

Profile is a JSON database the include any kind of data for robots and different fields that they use mostly for movement and shooting the ball. Profiler is a tool that we use to collect this data and fill them in database or parse them whenever we want. We can use profiler simultaneously during the game or we can give it our log of the game with log of the orders that we sent to robots or we can make a play that contains continuously direct passing or chipping.

The Profiler collect the speed of the ball that cause by the shot from our robots and also the order that we sent to robot to use how much of your power for that shot and make a lookup table in order to whenever we want to have shot with certain speed it's search in table and give us the best usage of power for that shot and we sent it to robot.

Actually whenever we want to have a direct or chip, pass or shot we use the location and final location that ball is spouse to go there and calculate the friction of path so depends on the friction of pass and passer robot we find best usage of power and sent it to robot to make that kick.

5.4 New marking system for defense:

For this year we create a new algorithm for stay in best point in field for be prepared for any situation that's probable happen. For finding this point we should find all dangerous situation that likely so any robot that are able to receive a pass or make a shot to goal are some kind of this dangerous point that we should stand in other word we find a place in field where are most efficient place for block any pass or shot and set target of robot go to point skill to that point. then after situation changes by pass, shot or changing ball possession we start to block ball if is needed or back off defense robot to defense area.

With combine position of opponent's robot, ball, direction of robots and their velocity vector and ball velocity vector we can make some vectors like P(I) that get different weight and we find best point by this equation :

$$\frac{\sum_{i=1}^{i=n} W_i . P_i}{n} = \bar{P}$$

W(i) is the weight of P(i) that calculated according to P(i) parameters and by using a decision making engine. Although we have wide range of choices in selecting our decision engine we decided to use Mamdani decision engine as simple one.

References

[1] Hesam T. Dashti, Shahin Kamali and Nima Aghaeepour (2007). Positioning in Robots Soccer, Robotic Soccer, Pedro Lima (Ed.), ISBN: 978-3-902613-21-9, InTech, Available from: http://www.intechopen.com/books/robotic_soccer/positioning_in_robots_soccer

[2] Akiyama, H., Noda, I.: Multi-agent positioning mechanism in the dynamic environment. In: RoboCup 2007: Robot Soccer World Cup XI. (2008)

[3] H.Akiyama, I. Noda, H.Shimora, Helios 2008 Team Description Paper, RoboCup 2008.

[4] H.Akiyama, Helios 2007 Team Description Paper, RoboCup 2007.

[5] Part of Helios team released materials, available at: http://sourceforge.jp/projects/rctools/

[6] Lecture notes from Michael Goemans class on Combinatorial Optimization. http://math.mit.edu/~goemans/18433S09/matching-notes.pdf, 2009.

[7] M. Paul. Algorithmen für das Maximum Weight Matching Problem in bipartiten Graphen. Master’s thesis, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, 1989.

[8] M.Norouzitallab, A.Javari, A.Noroozi, S.M.A. Salehizadeh, K.Meshgi, Nemesis Team Description Paper 2010, RoboCup 2010.

[9] M.Malmir, M.Simchi, S.Boluki, AUT Team Description Paper 2012, Robocup 2012.

[10] - the industry standard for high performance graphics (2011), http:// www.opengl.org/, [accessed February, 2011]

[11] OpenGL 2. Browning, B., Bruce, J. Bowling, and M. Veloso.: STP: Skills, tactics, and plays for multi-robot control in adversarial environments. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 219(1), 33{52 (2005)

[12] Bruce, J., Veloso, M.: Real-time randomized path planning for robot navigation. Lecture Notes in Computer Science pp. 288{295 (2003)

[13] Bruce, J., Veloso, M.: Safe multi robot navigation within dynamics constraints. Proceedings-IEEE 94(7), 1398 (2006)

[14] S.Mehdi Mohaimanian Pour, Vahid Mehrabi, Alireza Saeidi, Erfan Sheikhi, Masoud Kazemi, Ali Pahlavani, Mohammad Behbooei, and Parnian Ghanbari.: PARSIAN extended team description for RoboCup 2013

[15] Inc.: Qt - A cross-platform application and UI framework (2011), http:// qt.nokia.com/, [accessed February, 2011]

[16] Nokia 7. Smith, R.: ODE - Open Dynamics Engine (2011), http://www.ode.org/, [accessed February, 2011]

[17] http://en.wikipedia.org/wiki/Depth-first_search

[18] Roland Siegwart and Illah R. Nourbakhsh (2004), Introduction to Autonomous Mobile Robots, A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England, ISBN: 0-262-19502-X

[19] Alireza Saeidi, Mohammadhossein Malmir, Mohammad Mahdi Shirazi, Mohammed Behbooei, , Shahin Boluki, Masoud Kazemi, Seyed Mehdi Mohaimanian Pour, Parnian Ghanbari, Saeed Jamshidiha, Parisa Dajkhosh and Amin Zahedi.: PARSIAN team description for RoboCup 2014

[20] Hai-Yun Wang, Jong-Hun Park, and Uk-You I Huh, Fuzzy-EKF for the Mobile Robot Localization Using Ultrasonic Satellite, ICCAS 2014