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, Amir Maleki, Mohammad Amin Najafgholian, Maryam Mehraban, Nastaran Khodabandeloo, 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 from mechanical, electrical aspect and the team's recent work on the offensive and defensive tactics in software section and low level skills in control segment. Among the offensive tactics, we introduce new features in our visual planner in play on and play off then we present new defense mark system that increase reliability and finally, we describe improvements in algorithms which estimate robots state by profiling robots.

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 respectively.

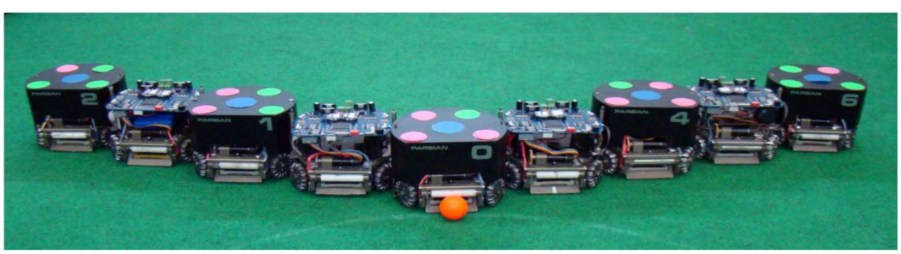

Fig. 1. Our Robots

2 Mechanical Design

In this part an introduction is going to be presented on the mechanical structure of the new robot designed for RoboCup 2016.

In this design the focus was on two parts: the dribbler part and easy access and repair part which will be explained in detail in the following sections:

2.1. Dribbler

Adding a new part to dribbler to covers and routes shot sensor cables in order to avoid collision with wheels or other robots for preventing damage to cable.

2.2. Easy access and repair

This year we assembled all components (electrical boards, batteries and capacitors) together which makes repairing and maintenance so faster and easier, specially when separation motors from electrical boards is needed.

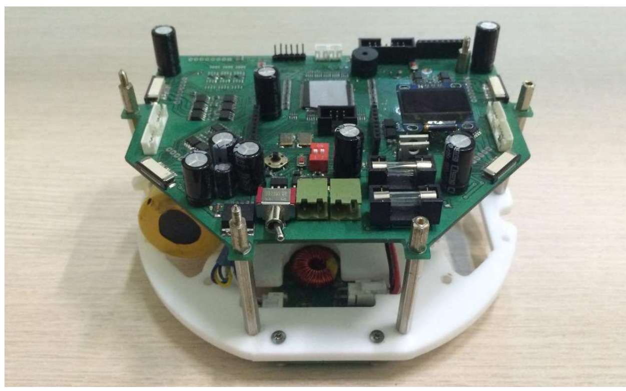

Fig.2. New integrated mechanical design

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.

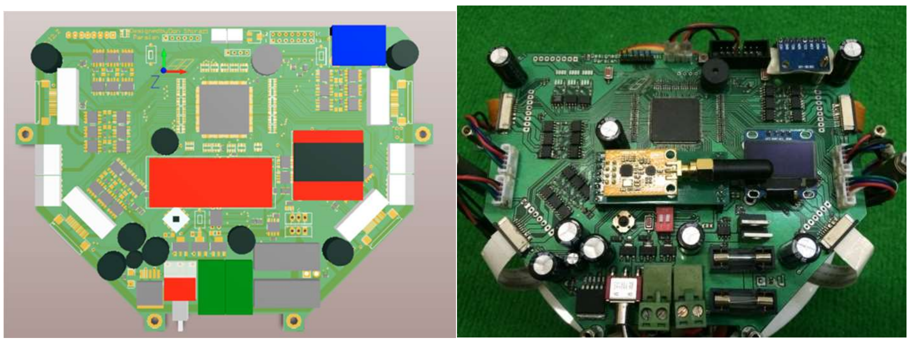

Fig3, 4. (Left) 3D design & (Right) real hardware

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.

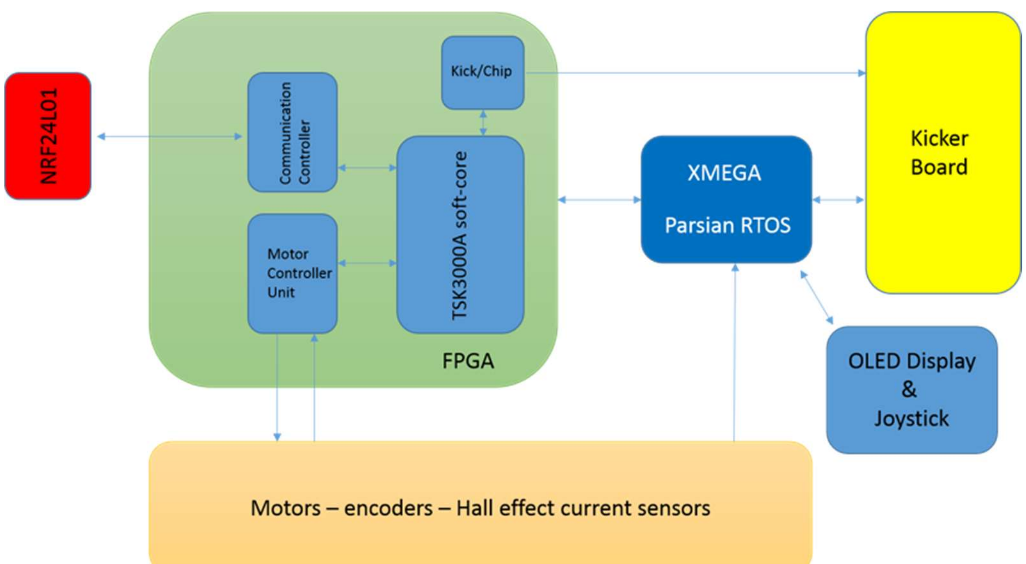

Fig. 5. New Main-Board Block Diagram

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 communication module, but this year two NRFs are used in each board to provide full duplex two way communication.

Also a new case designed for base communication module to prevent damages from communication board.

Fig. 6. New case that designed for protecting 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. 7. 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 a force to the robot in order to move in the first direction and after a while apply another 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 will be resulted and the robot will roll over.

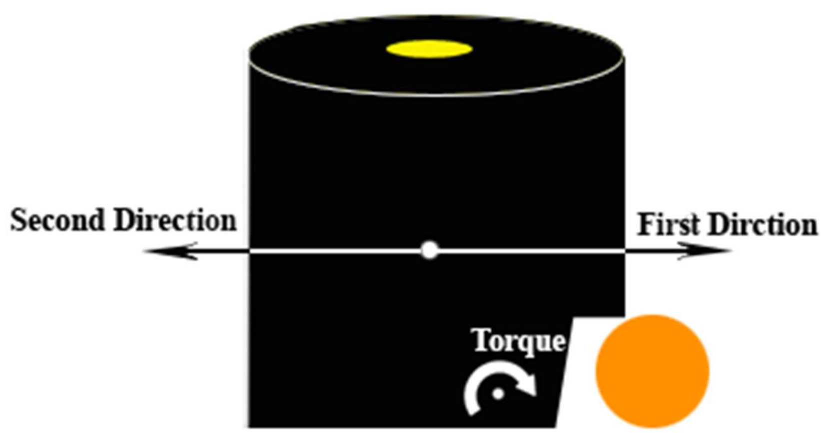

Fig.8. Robot and applied forces

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

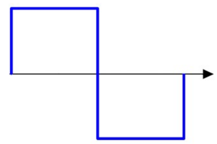

Fig.9. Applied forces diagram

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:

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

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.[20] In the following figure, a square wave before and after applying this control rule is shown.

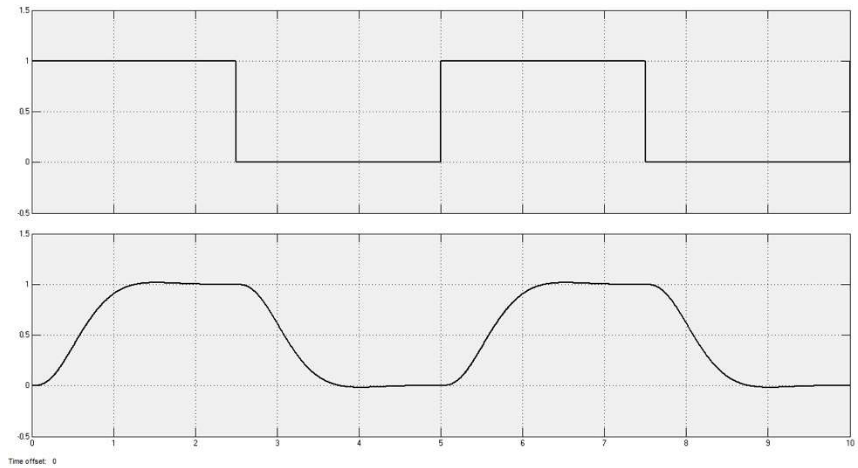

Fig.10. Applied forces diagram before and after applying minimum jerk trajectory control rule

5 software

5.1. Architecture

This year the software architecture has some minor changes that will be discussed in the next part.[17] Here is The Parsian Software architecture chart (Fig.10).

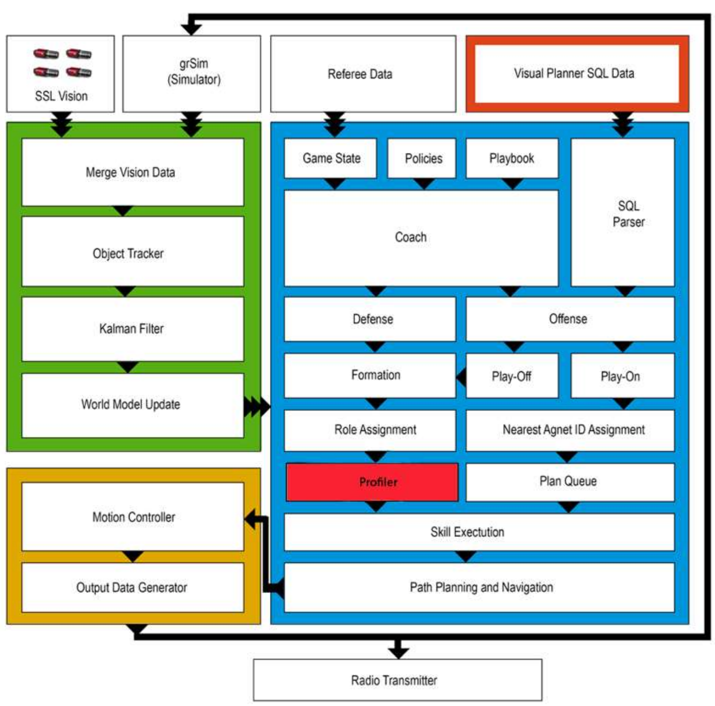

Fig.11. Software chart, the part in red is added this year

5.2. New Visual Planner

According to Parsian 2015 TDP the visual planner is a graphical and user friendly planner that can make one level plan for play on.
Also this app is write with QT5.3.4 so you can run this code in any OS(linux/osx/win).

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 off (free-kick) are not too 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.

When we reach to the desire situation or when one step fails, we deny rest of the plan and choose the best plan for that situation from play on playbook.

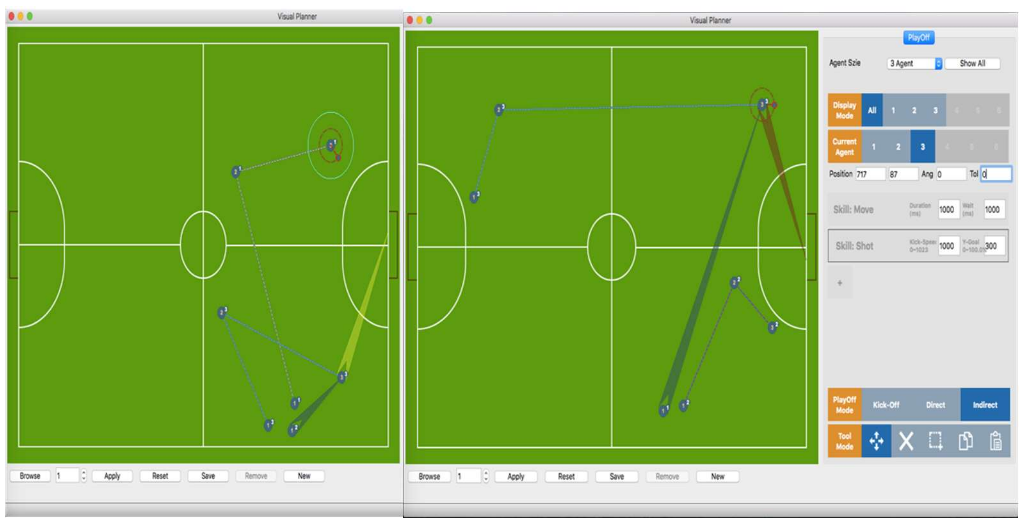

fig.12 two sample plan with visual planner play off section

5.2.2 New play on planner

In last version of visual planner we had 6 static region for ball and 10 static region for robots that was so hard to make plans for all of the states, however if you didn't make a plan for a state, AI automatically assign closet plan to that state.

In new planner we decrease the quantity of regions and use semi-statics plays just for situation that are so close to our plans and for rest of the game we are using of dynamic and AI algorithm for have more unpredictable play.

5.2.3 New features and options

As we made 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.

For making plays more unpredictable we don't give robots static tasks in same situation. Actually we give some weight to each possible work on that situation and then 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 revision that we used 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 that include any kind of data for robots and different fields that they use mostly for movement and shooting.

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 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 to collect data.

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 its power for that shot, and make a lookup table in order to whenever it have to shot with certain speed, program search in table and find the best usage of power for that shot and sent it to robot.

actually whenever we want to have a direct or chip, pass or shot we use the location of ball and final location that ball is suppose 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 (Zone Mark)

For this year we create a new algorithm for staying in the best point at field to be prepared for any situation that's probable happen.

For finding this point we should find all dangerous situation that possible, so any robot that are able to receive a pass or make a shot to goal are some kind of this dangerous factors.
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 and direction of robots and their velocity vector with ball position and velocity we can make some vectors like P_i that get different weight(W_i) 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] Part of Helios team released materials, available at: http://sourceforge.jp/projects/rctools/

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

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

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

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

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

[10] 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)

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

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

[13] 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

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

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

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

[17] Alireza Zolamvari, Mohammad Mahdi Shirazi, Seyede Parisa Dajkhosh, Maziar Arfaee, Amir Mohammad Naderi, Hamidreza Kazemi, Mohammad Mahdi Rahimi, Amirhossein Abbasi and Alireza Saeidi.: PARSIAN team description for RoboCup 2015

[18] 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

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

[20] R. Shadmehr, S. P. Wise, Computational Neurobiology of Reaching and Pointing