PARSIAN
Team Description for RoboCup 2015

Alireza Zolanvari, Mohammad Mahdi Shirazi, Seyede Parisa Dajkhosh, Amir Mohammad Naderi , Maziar Arfaee, Mohammad Behbooei, Hamidreza Kazemi Khoshkijari, Erfan Tazimi, Mohammad Mahdi Rahimi and Alireza Saeidi Shahrivar.

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

Abstract. This is the team description paper of the Small Size Soccer Robot team “PARSIAN” for entering the RoboCup 2015 competitions in China. In this paper we will represent our robots’ current design in mechanical, electrical, control and software parts. Improvements and developments like new mechanical design, control system, visual planner and enhancements in predefined plays.

1 Introduction

“PARSIAN” small size soccer 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.

“PARSIAN” team consists of ten active members from electrical, mechanical and computer science backgrounds. We have been qualified for nine consequent years for RoboCup SSL. We participated in 2008, 2009, 2010, 2011, 2012, 2013, and 2014 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.

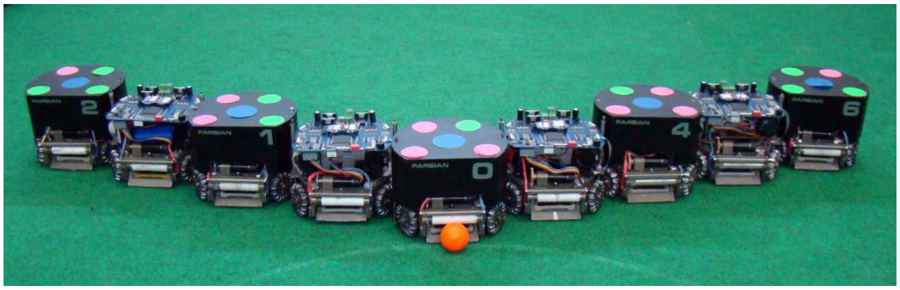


Fig. 1. Our Robots

2 Mechanical Design

In this part an introduction is going to be rendered on the mechanical structure of the new robot designed for RoboCup 2015. In this design the focus has been on three parts: the dribbler part, the center of mass, and easy access to the different parts of the robot which will be explained in detail in the following sections.

2.1. Dribbler structure

The dribbler of the previous robot had some problems with spinning, controlling, and detecting the ball, which created some obstacles to achieve an efficient shoot. To remove these obstacles and detect the ball in all situations we've decided to keep the sensor fixed and put the axis of the dribbler motion in a lower state in comparison to the preceding case. In Fig.2 the previous dribbler with the unfixed sensor (because of being placed on the moving part of the dribbler) is indicated.

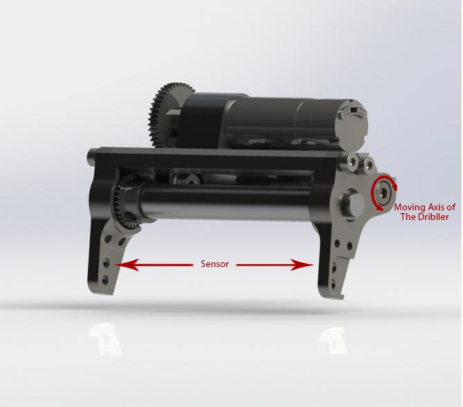


Fig.2. Preceding Dribbler

2.2. Center of mass

To have a high performance in deceleration state in the small size robots, it's far more efficient to keep the center of mass close to the back side of the robot. In order to exert this modification in our new robot we've changed the position of the capacitor and batteries, which are heavier than the other parts, and located them at the back of the robot, as indicated in Fig.3.

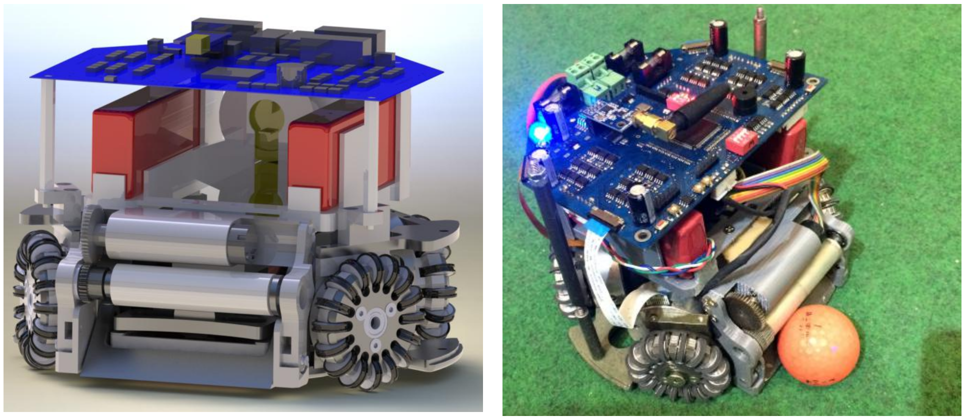


Fig.3. New mechanical design (SolidWorks design Left, Real robot Right)

2.3. Robot’s specifications

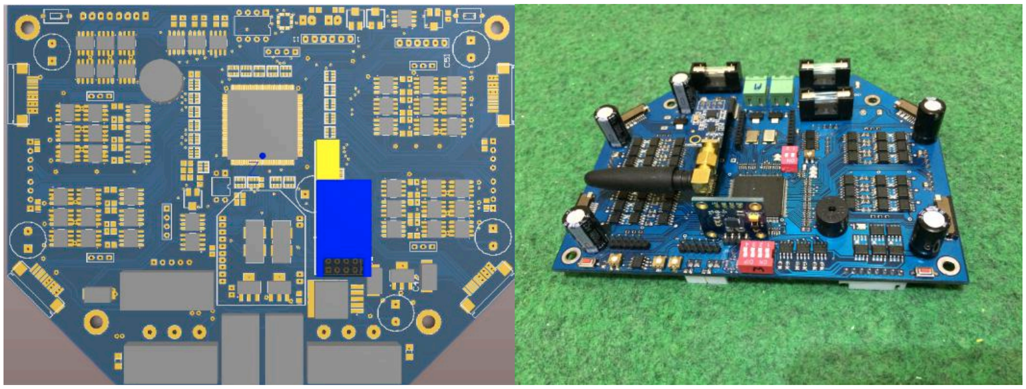
Robot Diameter	179 mm
Robot Height	140 mm
Ball Coverage	18 %
Max Linear Velocity	4.1 m/s
Weight	2.1 kg
Maximum kick speed	12 m/s
Limited kick speed	7.8 m/s
Maximum chip kick distance	6 m
Maximum ball speed catching	6.5 m/s

Table 1. Robot’s specification

3 Electrical Design

3.1 New Main Board

After testing the test bed we found out optimal design for our main board (fig 4) that has been discussed in Parsian’s 2014 TDP. Here some minor changes in kicking system and encoder will be discussed.



(a) **(b)**

Fig. 4. (a) 3D design (b) Real hardware

3.2. Kicker Board

Some minor modifications are applied to our last year's kicker board to optimize size and order of the kicker board parts. Another change is direction of the IGBTs on the board from horizontal to vertical for better heat transfer of the IGBTs.

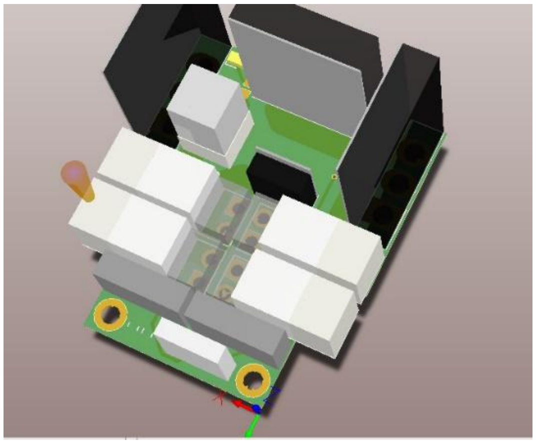


Fig. 5. Kicker board

3.3 New encoders

The motor's encoders just as other parts, damage and need to be replaced. But since we cannot afford the price, we decided to design and build our own ones. We use an AS5045 IC that is a 12 Bit programmable magnetic rotary encoder. This IC uses an array of hall sensors. Cylindrical magnet as depicted, locates above the IC's surface in an appropriate distance, connected to the rotating part of the motor.

We design our PCB just like the original Maxon encoders, to avoid the unnecessary changes to the current housings.

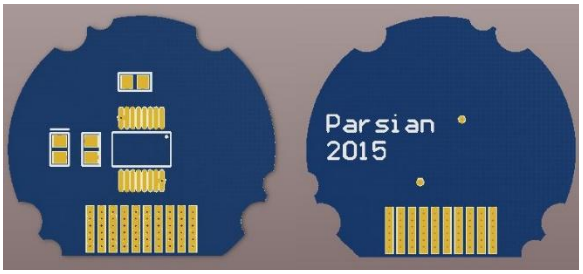


Fig.6. Sensor's board

3.4 Kicker calculations

Since we reviewed our mechanical design for this year, we decided to improve the kicker performance by using a mathematical model. So we use the electromagnetic and mechanic laws as described below:

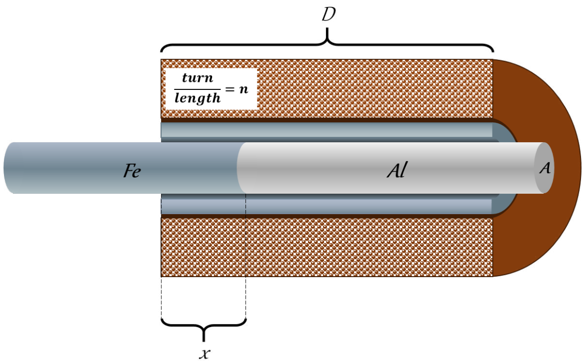


Fig.7. kicker's material

Our kicker solenoid, as depicted, has a 2-parted shaft. So we model it as to two solenoid in series.

$$L = \mu_{Fe} n x A + \mu_{Al} n (D - x) A$$
$$W = \frac{1}{2} L i^2$$
$$F = \frac{\partial W}{\partial x} = \frac{nA}{2} (\mu_{Fe} - \mu_{Al}) i^2$$

Where i is the current that the capacitors discharge through solenoid with. To simplify the math, we assume that $i(t)$ is linear between 0 and $\frac{V_f}{R_{solenoid}}$. Where V_f is the measured voltage of capacitors after kick.

So it yields:

$$v(T) = \int_0^T \frac{F}{m_p} dt = \frac{nA}{2m_p} (\mu_{Fe} - \mu_{Al}) \int_0^T \frac{V_f}{T R_{solenoid}} t.dt = \frac{nA}{2m_p} (\mu_{Fe} - \mu_{Al}) \frac{T.V_f}{2.R_{solenoid}}$$
$$v(T) = \frac{\mu_{Fe} n A V_f}{4R_{solenoid}.m_p} T$$

Where m_p is the plunger’s mass. So the ball’s speed after the kick is yield from the conservation law of momentum:

$$m_p V_p = m_b V_b \Rightarrow V_b = \frac{m_p}{m_b} V_p = \frac{\mu_{Fe} n A V_f}{4R_{solenoid}.m_b} T$$

4 Control System

4.1. Low Level

A fuzzy Logic PID controller has been used for low-level speed control (Fig.8) that has been discussed in Parsian’s 2014 TDP.

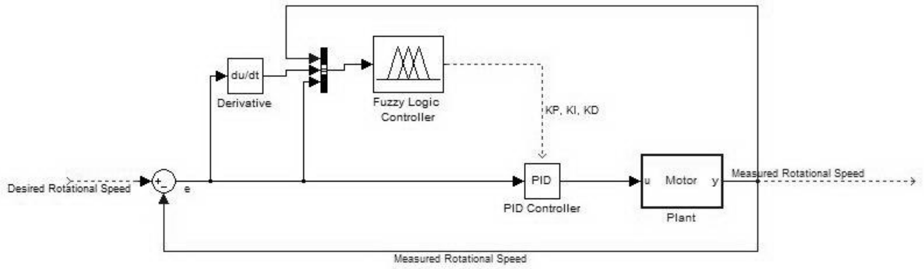


Fig.8. Low-Level Motor Controller Structure

4.2. High Level

One of the most common and big problems in control of the robots is existence of some errors in sensors data, which sometimes lead to big mistakes in localization of the robots. In order to reduce these errors, we fuse the information from the sensors and SSL vision. Through this action, we can read and use more reliable data from the sensors in control loop. [18]

The Kalman filter is an optimal and efficient sensor fusion technique. Application of the Kalman filter to localization requires posing the robot localization problem as a sensor fusion problem. The basic probabilistic update of robot belief state can be segmented into two phases, *perception update* and *action update.* [18]

The first step is action update or *position prediction*, the straightforward application of a Gaussian error motion model to the robot’s measured encoder and Gyro travel. The robot then collects actual sensor data and extracts appropriate features in the *observation* step. At the same time, based on its predicted position in the map, the robot generates a *measurement prediction, which identifies the features that the robot expects to find*, and the positions of those features. In *matching* the robot identifies the best pairings between the features actually extracted during observation and the expected features due to measurement prediction. Finally, the Kalman filter can fuse the information provided by all of these matches to update the robot belief state in *estimation*. Fig.9. Depicts the particular schematic for Kalman filter localization. [18]

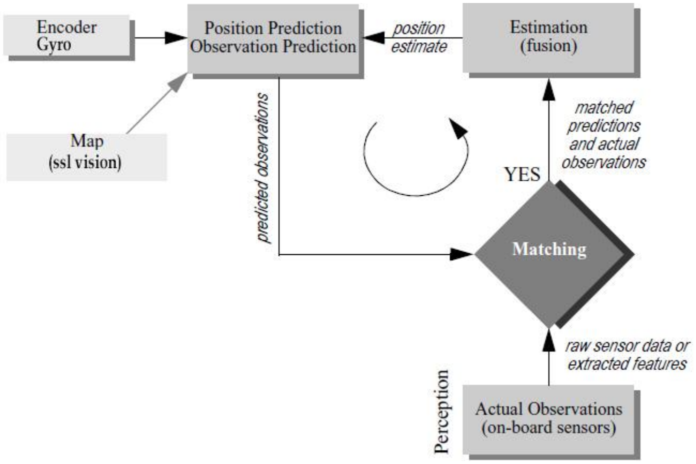


Fig.9. Schematic for Kalman filter robot localization

In future, we want to apply a Fuzzy-extended Kalman filter (Fuzzy-EKF) method to avoid the robot using the large error data to update the position continuously in order to improve the accuracy of localization. A weight scalar will be designed to change the noise covariance by inputting the robot rotation angle, innovation and the measurement data variation into the fuzzy system. Therefore, the proportion of the system and measurement value changed, which decreases the robot state errors indirectly. [20]

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).

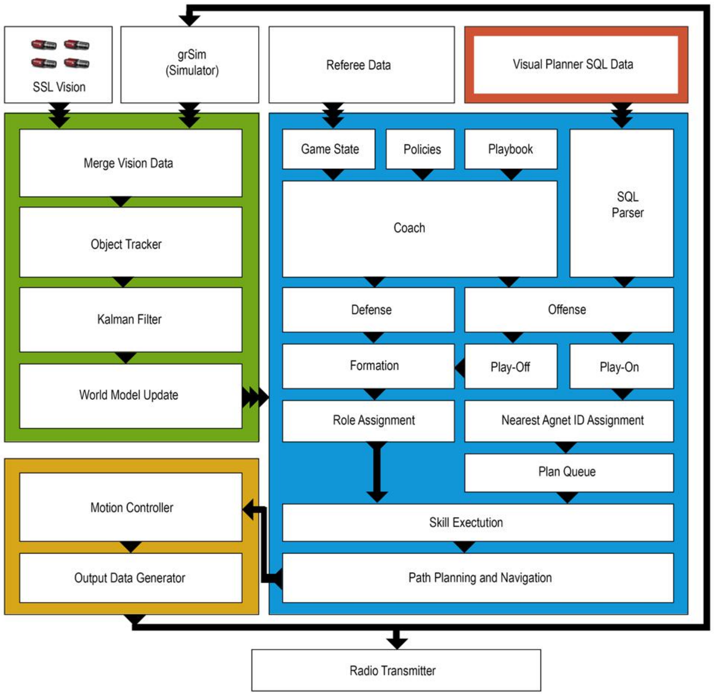


Fig.10. Software chart

5.2. Visual Planner

Since RoboCup SSL competition will be held in double sized field, it is very important to have offensive strategy to score more points and play in multi states. In normal fields it was very difficult to perform a plan completely due to small area of movement and speeds of new robots, but new fields are way easier to create plan for that actually works.

For reasons mentioned, we developed a semi static visual planner (Visual Planner). It's high level software for arranging agents according to ball position and agents conditions.

In each plan first of all we set a ball position then according to that we decide each robot task.

Tasks are consists of Pass, Chip, Kick, Move, and One-Touch skills and each has two offensive and defensive mode. Each agent can have up to 4 tasks and these tasks will be executed upon possibility.

The most challenging problem was to assigning ID to each robot, if we used an exact ID for each agent it would be very difficult to arrange plans for them. Best solution is to use a DFS Tree [17] to find the nearest agent to the current plan dynamically. For example assume there are two robots in area 5 and 9, and plan is to move an agent from 6 to 10, program assigns nearest robot to it and it would be the former robot.

Filed is divided to 6 sections for ball position and 10 sections for agents in order to minimal the plans.

So far it was static plan in dynamic plan, it will decide what points are best positions for starting and ending a plan loop and it will be processed in several threads in order to make it faster and more real time.

And the final step is to end the plan, each plan can be ended by two condition, time and desired conditions (for example when an exact agent receives a pass it immediately goes to next plan, and doesn't wait for all robots to complete their tasks). Advantage of this method is single state plans and it does not need to create multi-level plans that could be disturbed by opponent agents.

A#1	1: NA A:2 B:6	2: NA A:2 B:6	3: NA A:2 B:6	4: NA A:2 B:6
A#2	1: NA A:6 B:NA	2: NA A:6 B:NA	3: NA A:6 B:NA	4: NA A:6 B:NA
A#3	1: NA A:8 B:6	2: NA A:8 B:6	3: NA A:8 B:6	4: NA A:8 B:6
A#4	1: NA A:5 B:10	2: NA A:5 B:10	3: NA A:5 B:10	4: NA A:5 B:10
A#5	1: NA A:8 B:7	2: NA A:8 B:7	3: NA A:8 B:7	4: NA A:8 B:7

Fig.11. Visual planner

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