Parsian

(Amirkabir Univ. Of Technology Robocup Small Size Team)

Team Description for Robocup 2012

Vahid Mehrabi, Ali Koochakzadeh, Seyed Saeed Poorjandaghi, Alireza Saeidi, S.Mehdi Mohaimanian Pour, Erfan Sheikhi, Pourya Kaviani, Sina Saharkhiz, Abbas Razzaghpanah, Ali Pahlavani, and Arash Bahmand

Electrical Engineering Department
Amirkabir Univ. Of Technology (Tehran Polytechnic)
424 Hafez Ave. Tehran, Iran
small-size@parsianrobotic.ir

Abstract. This is the team description paper of the Robocup Small Size Soccer Robot team "Parsian" for entering the Robocup 2012 competitions in Mexico. In this paper we will represent our robots' current hardware design, as well as the software architecture in detail with focus on new improvements that have been made since last year. Improvements and developments like new mechanical design, improvements on planinng structure and enhancements in predefined plays, a high speed positioning evaluator will be discussed in detail.

1 Introduction

''Parsian" small size soccer robots team, founded in 2005, is organized by electrical engineering department of Amirkabir University of Technology. The purpose of this team is to design, build and program a small-size soccer robots team compatible with International Robocup competition rules as a student based project. "Parsian" team is a group of eleven active members with electrical, mechanical and computer science/engineering backgrounds. We have been qualifed for six consequent years for the international RoboCup SSL. We participated in 2008, 2009, 2010 and 2011 RoboCup competitions. Our most notable achievement is being awarded second place in RoboCup 2010 SSL's technical challenges.

In this paper we first introduce our robots' hardware (section 2). Our new mechanical design will be discussed In section 2.1 and our electrical design will be covered in section 2.2. Our vision system will be discussed briefly in section 3. Section 4 explains our software framework including high level planning algorithm, low level control algorithms and our 3D simulator.

2 The Robot's Hardware

2.1 The Robot's Mechanical Design

In this section we introduce our robot's *new* mechanical design and some of the improvements we have been working on since RoboCup 2010. Our 2010 team

Fig. 1. Our Robots

description paper describes in detail, the design we previously used on our robots [5].

After RoboCup 2010 we decided to design and build more robust and powerful robots. Therefore, we've started to work on the optimization of our mechanical design in order to build more reliable, more powerful, faster and more accurate robots. After achieving the goals set as improvements in design, we replaced all robots with the new platform model for RoboCup 2011. During 2011 competitions we realized that our design works well and we have reached our goal. So our new goal was to make some minor changes and take what we had and improve it even further. Therefore, we have made the robots lighter, modified the size of the wheels, optimized the dribbling mechanism and changed manufacturing tolerances of the wheels for the better. Figure 2 makes a comparison between the initial design and the optimized one.

The new design's characteristics are as follows:

Robot Diameter	178 mm
Robot Height	138 mm
Ball Coverage	19 %
Max Linear Velocity	3.5 m/s
Weight	2.0 kg
Maximum kick speed	14m/s
Maximum chip kick distance	7.5 m
Maximum passing ball speed catching	6m/s

Main Structure and Driving System Our robots main structure is made of aluminum alloy 2024-T351 to keep it robust and light. To prevent short circuit between electrical components and to reduce erosion, all parts are harden anodized. Our design is a four wheel omni directional robot. Each wheel is driven by 30 watt Maxon EC45 motors. The motor power is transmitted via two gears

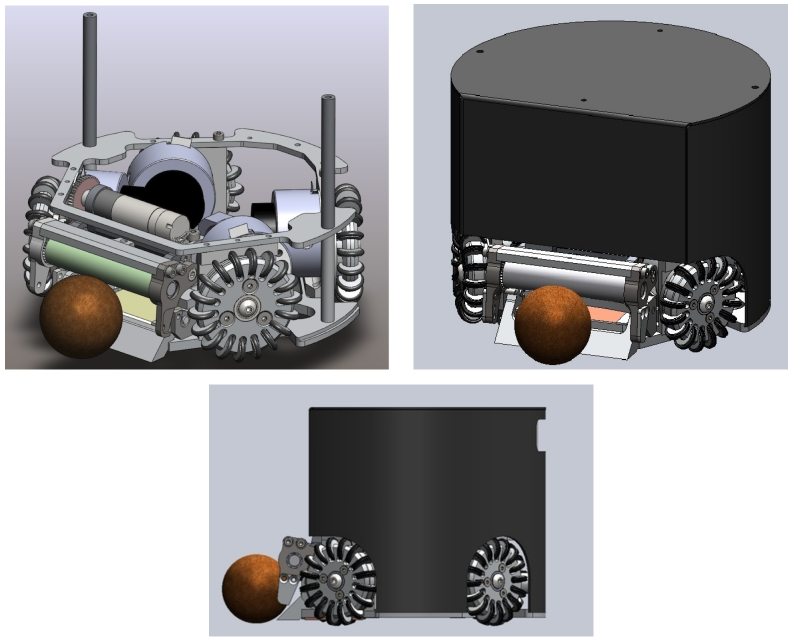

Fig. 2. Initial design of Parsian's Robots 2011 (left) and final Mechanical Design 2011-2012 (right) side view (down)

with ratio of 3.6:1 (76:21) to wheels. The wheels are 52mm in diameter with 16 sub-wheels. They are made of aluminum alloy 7075. Figure 3 shows the driving system in detail.

Direct/Chip Kick Our new robots use two solenoid systems in order to move plungers and kick the ball. For direct kicks, a cylindrical solenoid with length of 55mm is used with a 23AWG enameled wire. We optimized our direct kicking system to consume less space without losing efficiency. Kicker bar (plunger) is made of 3 parts with diameter of 13mm and total length of 130mm which are thread fastened to each other. The end part of this component is made of titanium alloy to endure high impact caused by kicker bar. The chip kick system is similar to direct kick, however its solenoid shape is flat. The size of the flat plunger in new design has been increased by 150

Dribbling and Suspension System In our new design we use Maxon EC16 DC motors (15 Watts) with GP22 1:3.8 gear heads for spinning the ball. With help of two external gears with ratio of 2.5:1 the spinner can reach the speed of 10000 RPM. This actuator is powerful enough for a successful ball manipulation. The whole system is depicted in Figure 4(c).

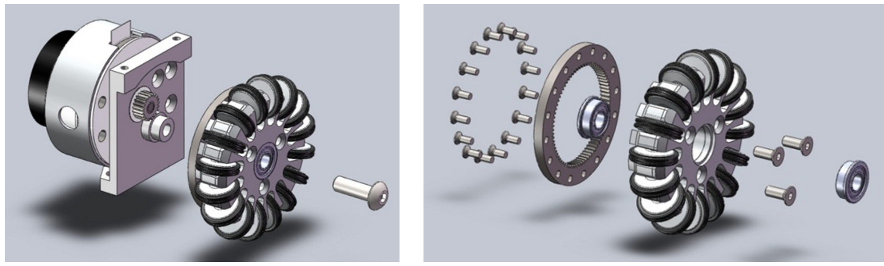

Fig. 3. The Driving System

2.2 Electrical Design

The electrical system hasn't been changed much since last year. We dont mean to make vast alterations to the current design. This years modifications include reduction in size and weight of the electronic boards, removal of unnecessary elements and improvements in wireless communication system design. The main electronics design consists of a main board and a kicker board. Different parts of the main board are in charge of carrying out tasks such as driving BLDC[1] motors, wireless communication, decoding sensors readout, execution of the control loop and sending control commands to the kicker board. The kicker board is designed to recharge the capacitors in the shortest time possible and release the energy stored into the solenoid over a controlled discharge time. More information can be found at [7]. A block diagram regarding the electronics systems design and behavior is shown in Fig. 5.

Wireless Communication In 2010, there was one 2.4 GHz XBee PRO wireless module for receiving control commands from the remote host PC and sending monitoring data on each robot. The XBee is half-duplex so it cannot both send and receive data simultaneously. This results in loss of data. This year we use two XBee modules on each robot, one dedicated to receiving desired robot velocities, kick/chip desired speed and permissions and another XBee module to send battery levels, ball detection status and other monitoring data.

Kicker Board To decrease the size and weight of the kicker board, this year we have redesigned the kicker board by means of using new electrical components. The kicker board continuously charges two 2200 μF 100V capacitors connected in parallels. The current design is based on DC to DC boost convertor circuit which utilizes a power MOSFET to discharge the stored energy into two solenoids. The resulting kick power is controlled by the discharge time according to a look-up table managed by an ATMEGA8 micro controller.

---

[1] Brushless DC Motor

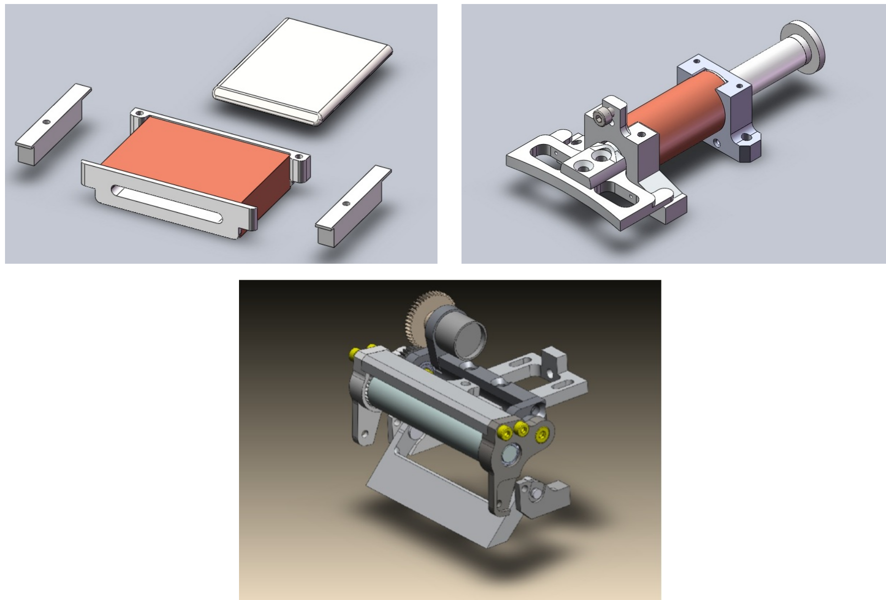

Fig. 4. (a) The new chip kick solenoid and plunger (b) Direct kick system (c) Dribbling and Suspension System

3 Vision System

Last year, we have replaced our cameras from AVT Marlin F046C (1394a) to AVT Stingray F046C IRF (1394b) which are capable of capturing 780px×580px frames at 65fps. Using these cameras in YUV color space with ROI enabled , we make use of SSL shared vision system [9] software. For a wider view of the field we also replaced our fixed 4.5mm lenses with 4-12mm wide zoom lenses.

3.1 Object Tracker

Processing of each camera's output is independent within the SSL shared vision system. The resulting package includes data of all detectable objects for each camera. In this manner there can be any number of different objects. i.e. the package may contain numerous ball positions inside it.

In order to have a unified view of the whole soccer field and to avoid mis recognition of noisy objects (e.g. the hands of referee which may be detected as ball), the output of SSL-Vision's data should be merged and filtered.

In order to merge frames and track objects within them, the two raw output data packets of SSL-Vision in each time step are passed through a *Bayesian filter*. The filtering is based on the euclidian distance between various objects within two consequent time steps. This process is done in a separate thread so the planning system can access the most reliable data at anytime. There is

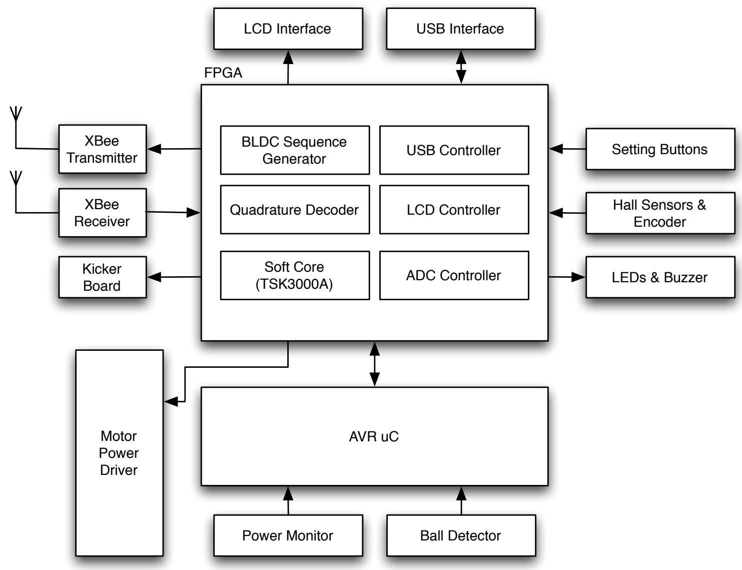

Fig. 5. The electrical system's diagram

another level of filtering in this thread, which uses a Kalman filter in order to reduce noise, estimate velocities and compensate the loop delay.

4 Planner

An overview of our planner system is demonstrated in figure 6. The data flow starts from vision part, in which SSL-Vision packets are received and processed. After this process the world model and its history are updated and the decision making loop is executed. The result of total processing cycle is the generated velocity commands for robots, which are sent to radio transmission module. The planner framework is written in C++ using Qt Framework[6] under Ubuntu Linux OS.

4.1 High Level Planner

The Coach layer is the first step in the high level planning (decision making) loop. Chosing a formation for the team is done prior to any other decisions. According to policies, that are a mixture of manual configurations and game-state dependant updated values, each cycle the coach layer decides the team's formation. Therefore, each agent takes part in one of the main plans: defense, midfield and offense. Defense plan consists of agents which are near the friendly penalty area, including goalie and some blocker agents. Middle plan agents intend

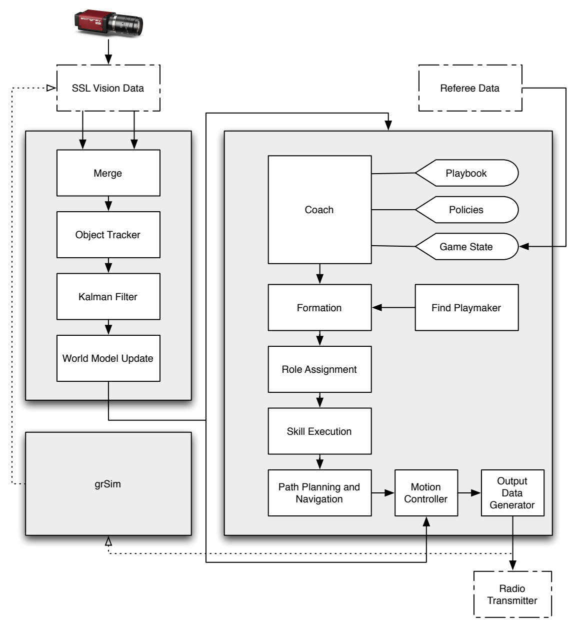

Fig. 6. Parsian's Software Architecture

to possess the ball owned by opponent and diminish their attacking opportunities with marking, blocking, ball interception and etc. Offense plan includes agents that are going to create attacking chances to score. One agent always takes the role of the "playmaker" (the agent that possesses the ball), other offense agents should take suitable positions. After running the plans, a set of roles are assigned to agents in an optimized way, so that minimum movement is needed for agents to execute their roles. To perform a role, each agent may use a different set of basic skills. For example "marker" itself is a role but it uses the "gotopoint" skill to reach its target. The hierarchy of the coach structure is shown in figure 7.

As a matter of fact, in a small-size game, most of the time the game is in stop mode (i.e. ball is moved out and the game should be started either by a direct or an indirect kick). Thus, having a knowledgeable game play when the game starts (direct or indirect kicks) may result in more scores. Kickoff, indirect kick, direct kick and penalty kick are the main "non-play-on" plays in a small-size robotic game. To have more diverse "non-play-on" game plans, we have implemented a script language. There is a simple kickoff plan written in our game script in figure 8.

Defense Plan The main goal of defenders is to take suitable positions to cover the goal from the ball's predicted position as much as possible. The main idea is to find places for defender robots on the bisector angles from ball to empty

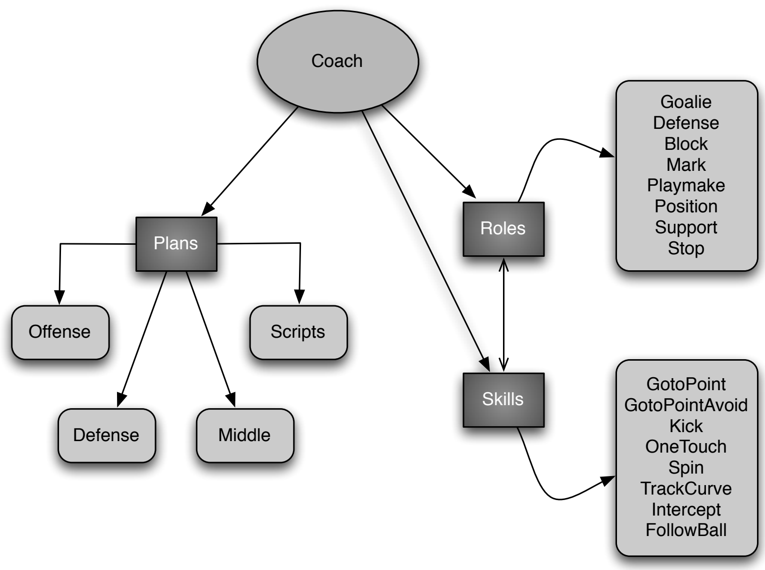

Fig. 7. The hierarchy of coach stucture

```
fav=1
$ourkickoff
>always
<start
<ballmoved
{
        >count(this,3)
        ~kickoff3
        playmake(stop,wing)
        position(ourmidfieldtopwing,@onetouch)
        position(ourmidfieldbottomwing,@onetouch)
}
{
        >count(this,2)
        ~kickoff2
        playmake(kickoff,stop)
        position(ourmidfieldtopwing,@onetouch)
}
{
        >count(this,1)
        ~kickoff1
        playmake(stop,kickoff)
}
```

Fig. 8. A sample of OurKickOff script

areas of the goal. Although there is a strong need to clear the ball in some cases to avoid danger.

Midfield Plan When the team has no possession on the ball, it has to prevent the opponent from passing and block the passes and mark pass receiving agents of the opponent team. This is done using an optimized agent marking.

Offense Plan

1. Playmaker
   Playmaker is the agent that owns the ball and plans to make appropriate pass/shoot commands to create scoring opportunities. Playmaker can choose an action between passing, shooting, one touch kicking and spinning the ball. There are some evaluation functions that predict success rate of each one of the above-mentioned actions. Then playmaker chooses the best action using these probabilities and some pre-defined constants that predict priority of each one of the actions. For example most of the times playmaker should find a way to kick the ball, so kicking has the highest priority, thus has the highest constant. The set of all constants creates the attacking behaviour of the team.
2. Positioning
   The other agents in the attack plan are only searching for suitable positions for catching probable passes or blocking opponent agents. For each point of the field some features like goal-visibilty, angle to shoot, openness, distance to goal, cornerness and distance from opponents are evaluated and mixed using a power weighted multipication. Because the process is done for each point of the field, it is necessary to find a way to accelarate this compution. We used CUDA and OpenCL for this purpose to benefit from the parallel processing power of the modern GPUs. A sample of our positioning output is depicted in figure 9(b). The process could be done with a rate of 10 times/second.
   Almost all skills make use of the *Navigation* module. First in this module, a safe path is obtained using our developed version [5] of ERRT algorithm [3][4]. The aforementioned enhancements not only help robots to find an obstacle-free path, but also to find a path which is far enough from moving dynamics objects. Next, a motion planning algorithm is used to generate a trajectory. This algorithm employs a binary search on Velocity Space to plan a trajectory for the desired path. Afterward, a nonlinear motion controller is applied to enhance the tracking precision of the designed trajectory. And finally generated commands are sent to each robot.

4.2 3D Simulator

This year we've made some improvement to our *Open Dynamics Engine* [8] and *OpenGL* [1] based 3D SSL simulator software, *"grSim"* which we introduced in our TDP 2010 [5]. The robot simulation layer made more realistic, in addition to lots of improvement to the user interface. A screenshot of our software environment, the planner and the simulator is depicted in figure 10.

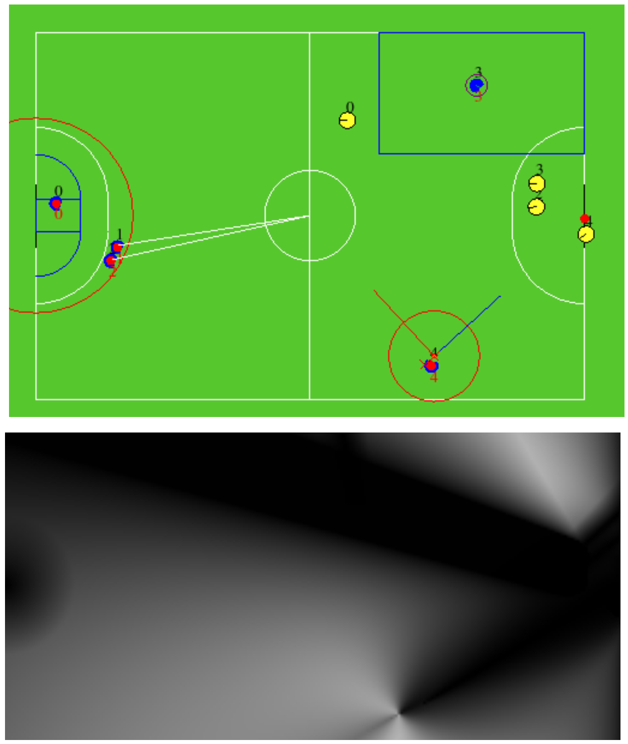

Fig. 9. (a) A screenshot of monitor (b) Output of positioning evaluation system.

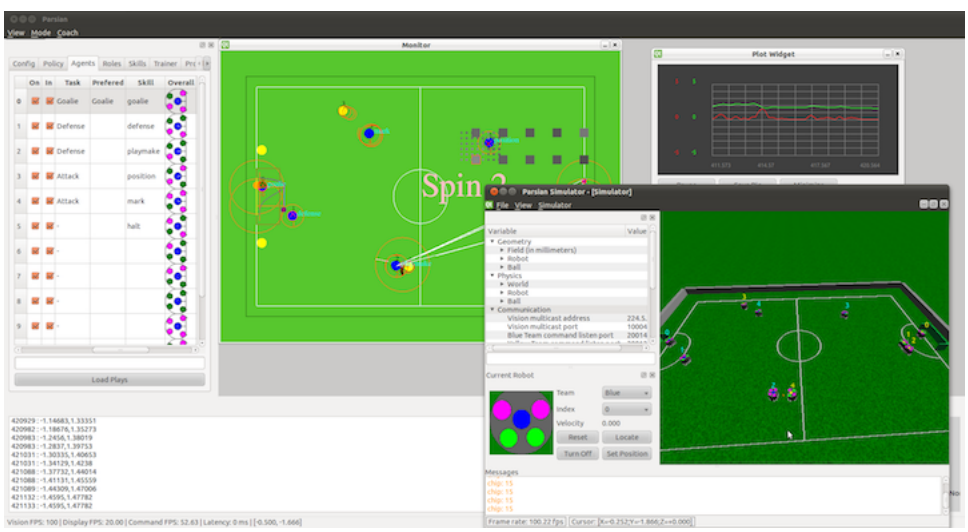

Fig. 10. The Software Environment

4.3 Future Plans

The list of our current research is given bellow. The main attitude of the mentioned researches is concentrated on improving the artificial intelligent methods utilized in the software architecture.

1. Designing and implementing a new system for pass evaluation based on research done in our RoboCup soccer simulation team.
2. Utilizing continuous-state reinforcement learning methods (such as Fuzzy Reinforcement Learning) to improve individual robots' skills and tuning the team's policy.
3. Identification of robot's dynamics model to improve the navigation technique and path planning algorithms.

References

1. OpenGL - the industry standard for high performance graphics (2011), `http://www.opengl.org/`, [accessed February, 2011]
2. Browning, B., Bruce, J., Bowling, M., Veloso, M.: 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)
3. Bruce, J., Veloso, M.: Real-time randomized path planning for robot navigation. Lecture Notes in Computer Science pp. 288–295 (2003)
4. Bruce, J., Veloso, M.: Safe multirobot navigation within dynamics constraints. Proceedings-IEEE 94(7), 1398 (2006)
5. Monajjemi, V., Atashzar, S.F., Mehrabi, V., Nabi, M.M., Omidi, E., Pahlavani, A., Poorjandaghi, S.S., Sheikhi, E., Koochakzadeh, A., Ghaednia, H., Pour, S.M.M., Behmand, A., Rastgar, H., Arabi, M., Nouredanesh, M.: Parsian - team description for robocup 2010 ssl. RoboCup 2010
6. Nokia Inc.: Qt - A cross-platform application and UI framework (2011), `http://qt.nokia.com/`, [accessed February, 2011]
7. Poorjandaghi, S.S., Monajjemi, V., Mehrabi, V., Nabi, M.M., Koochakzadeh, A., Atashzar, S.F., Omidi, E., Pahlavani, A., Sheikhi, E., Behmand, A., Pour, S.M.M., Saeidi, A., Shamipour, S., Karkon, R.: Parsian - team description for robocup 2011 ssl. RoboCup 2011
8. Smith, R.: ODE - Open Dynamics Engine (2011), `http://www.ode.org/`, [accessed February, 2011]
9. Zickler, S., Laue, T., Birbach, O., Wongphati, M., Veloso, M.: SSL-vision: The shared vision system for the RoboCup Small Size League. RoboCup 2009: Robot Soccer World Cup XIII pp. 425–436 (2010)