OMID 2017 Team Description

Ali Mollajafai[1], Mohammad hossein Zahedi[1], Matin Farshchi[1], Mohammad hossein Rezaie[1], Rasoul Aboutalebi[1], Erfan Jahanshiri[1], Javad Rahmani[1], Omid Mahdizadeh[1], Maryam Akhoundian[2]

[1] Department of Electrical Engineering of Shahed University of Tehran, Iran

[2] Department of Computer Science of Shahed University of Tehran, Iran

http://www.omidrobotics.ir

omid.robotics.ssl@gmail.com

Abstract. This paper is an explanation of OMID 2017, Robocop small size team, technical estate and robots technical improvement which generally divided into three part: Mechanical part, Electronic part and Software part.

1 Introduction

Omid Robotics Team has begun small size league activities since summer of 2007, as a branch of robotics society of department of Electrical Engineering of Shahed University. Omid robotics team endeavor to confirm new member's substitution as a custom due to create a suitable situation for talented and enthusiastic students. Now 4th generation of Omid robotics team is working hard to achieve more glory.

According to last competition (Iran Open 2016) and the problems we faced, some improvements has done:

- Replacing Altera with Xilinx FPGA.
- Communication module data rate problem solved.
- Improving robot's inner controlling system.
- Optimizing robot's FPGA codes.
- Design and setting new boards.
- Changing ball detection system.

More details added in related part.

2 Mechanical Design

According to Small-Size League rules, the robots must have specific dimensions. Our robots' diameter is 178mm and it's height is 148mm. Also each robot covers less than 20% of ball. The whole robot is about 2 kilograms weight. 3D simulation models shown in Fig. 2 are created with Solid Works

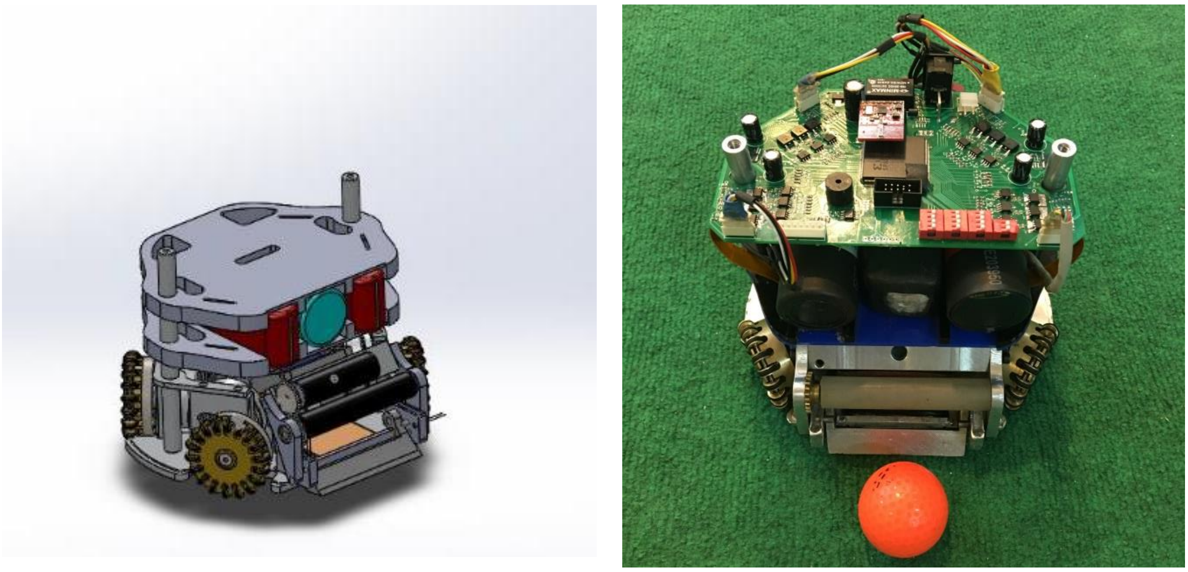

Fig. 1. Robot's mechanical plan design

2.1 Driving System

The main chassis of robot is made of Aluminum. 4 Omni-directional wheels carry the main body. Each wheel is coupled to an EC-45-Flat brushless 30 watt motor via an inverse gear with a transmission ratio of 1:5. These Wheels are fully designed in one piece and no screw has been used in the structure of wheels. This feature causes more efficiency, more wheel life time and simplicity in design.

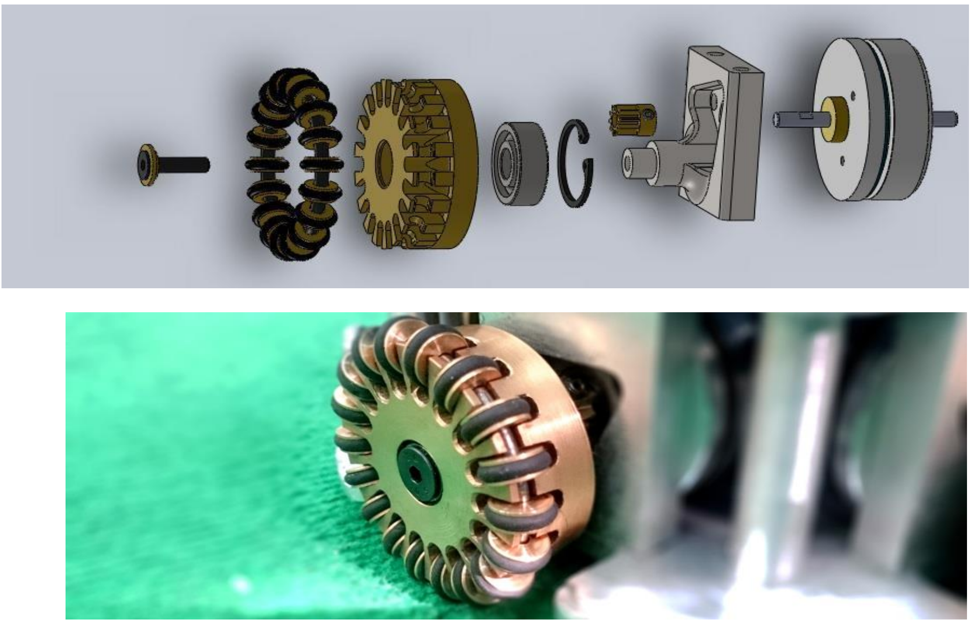

Fig. 2. Omni-directional wheel structure.

2.2 Kicking System

There are two solenoids that help to kick the ball, both for direct and chip kick. A flat plunger is made of steel, 4mm thickness. Direct kick plunger is made of two materials. The first part has been made of steel which is magnetic, and the second one has been made of a material with no magnetic property such as Aluminum. This feature causes a powerful kicking system.

3 Electrical System

Robots consist of three general part: Communication, main board and shoot board. In the main board we have FPGA Xilinx Spartan 6 XC6SLX9-2TQG144[1], motor drivers and etc. A 4-cell li-polymer battery (2200 mAh) used as power supply. Block diagram of robot's electrical hardware and software is shown in Fig.3.

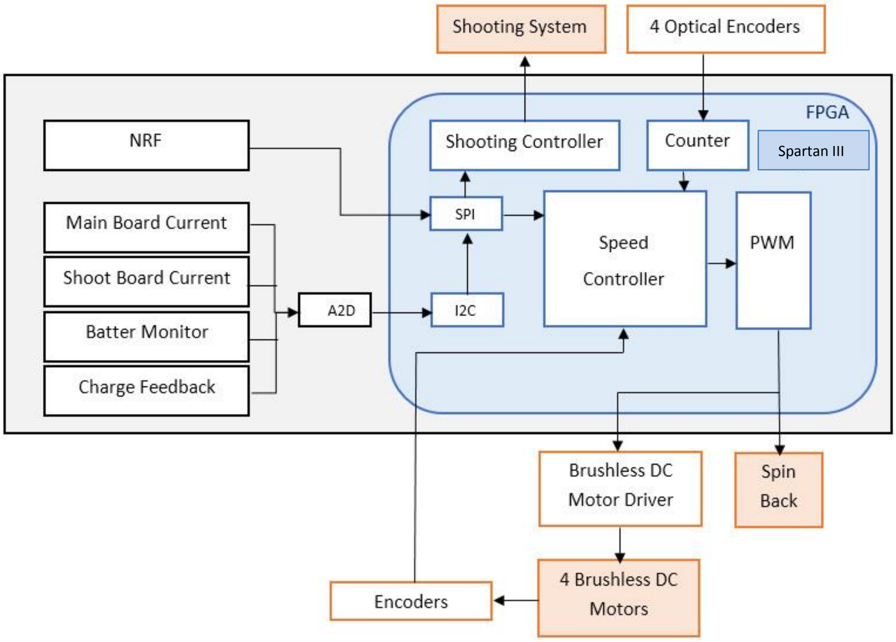

Fig. 3. Robots Hardware Block Diagram.

3.1 Communication module

The communication between robots and computer is performed with nRF24L01+ module [2]. The nRF24L01+ is a single chip 2.4GHz transceiver with an embedded baseband protocol engine, suitable for ultra-low power wireless applications. The nRF24L01+ is designed for operation in the world wide ISM frequency band at 2.400 - 2.4835GHz.

In the recent competitions we faced with connection lost between Robots and Computer and low data rate. In order to solve the problem we used Logic analyzer to check nRF24L01+ more evident. The result was that a human body between computer and robots makes data lost which happens a lot in the field. Therefore we have used nRF24L01+ with an amplifier for each robot, which reduces our data lost and makes reliable communication in double size field.

3.2 Main Board

In the main board we use FPGA (Spartan 6) as central controller that controls motors and shooting system by producing PWM. In addition we use an ARM microcontroller as co-processor to monitor significant voltages and currents, which is our recent improvement. New board doesn't have much difference from the previous one in schematic design but the appearance of the board has changed and some PCB designing points has been applied.

In FPGA we do some code optimization and change velocity controller [3] with a torque controller [6] for each wheel.

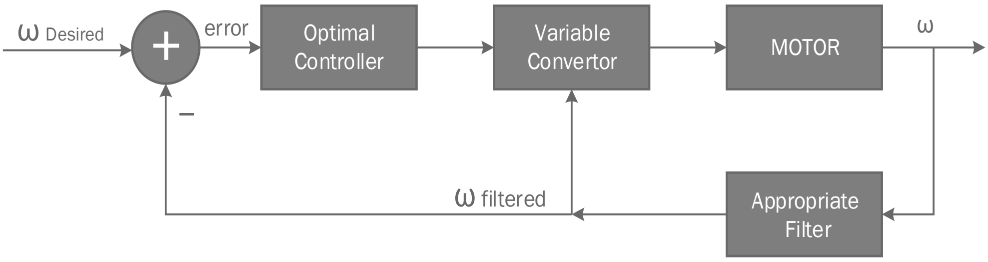

Fig. 4. Torque controller [6] block diagram

Breaking down of MOSFETs and MOSFET drivers was our problem in driving system. we have changed our MOSFETs and MOSFET drivers model due to the shortage of input capacitors and less current.

3.3 Shooting System

Because of free space limitation on robot we compelled to design new shooting board, Which is smaller and more efficient than the previous one.

There are two kickers, a direct kicker and a chip kicker. We have developed the flat kick system to kick in maximum velocity, approximately 10 m/s. The kicker board can charge two 2200μF capacitors from 0V to 170V in about 4 seconds with 1A average current. In the charging moment a PWM is used, that PWM and Duty Cycle change proportional with capacitor voltage's percentage. In order to avoid IGBTs breaking down at the shooting moment, RCD-SNUBBER [5] as secondary Circuit has been used, that damp extra current.

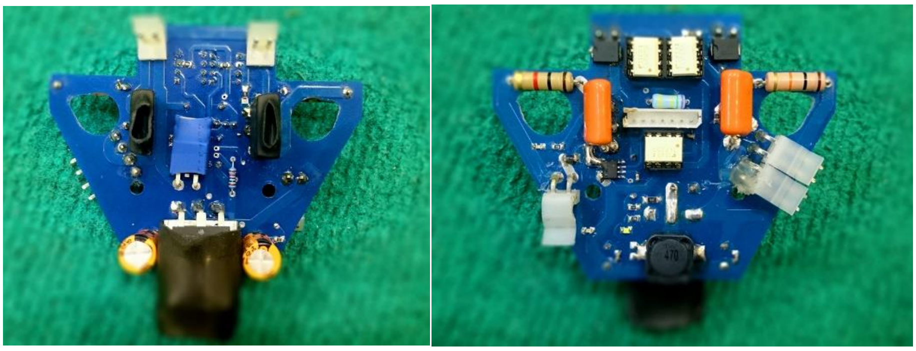

Fig. 5. New Shooting board

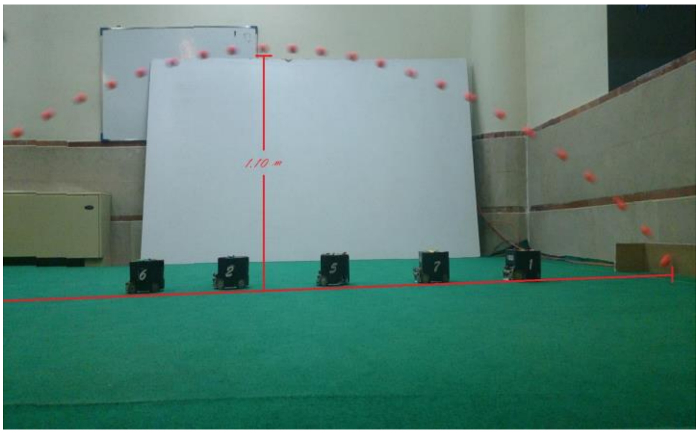

Fig. 6. Chip kick test in laboratory

4 Computer Software

The computer software is separated into two main sections. Algorithms & User Interface and controlling system.

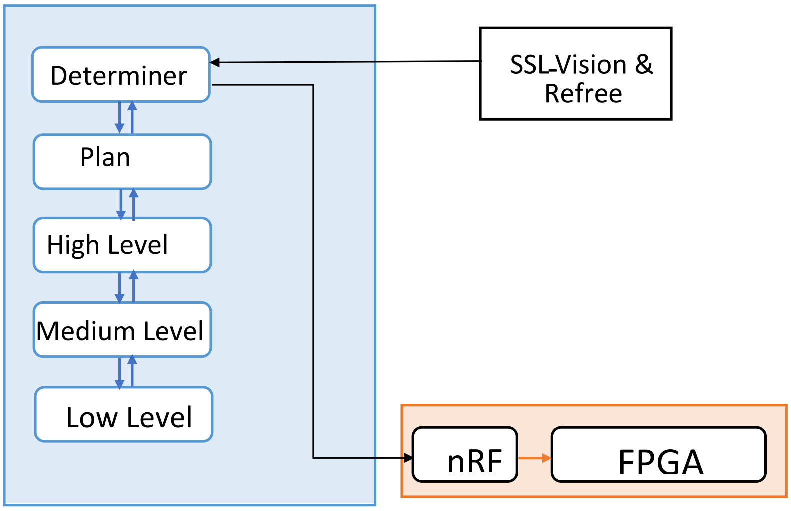

Fig. 7. Block diagram of computer software

4.1 Algorithms & User Interface

Algorithms structure contains four basic parts: 1) Plans 2) High Level 3) Medium Level 4) Low-Level

Entirely we classify functions in these four parts in order to make the program more simple & flexible.

To monitor the status of each robot and control all robots in the game field, Monitoring Software is programmed that will be installed on the off board controlling system. The robots location and ID on the game field is received from the visioning software and simulated. In other words this software manages the game play.

In Algorithms part we suffice to code optimization in order to reduce calculation loops and to keep old algorithms and layer structure of software. In addition, in the same way for user interface, we have just fixed some problems.

Fig. 8. Software based on windows (Visual Studio is used as our compiler)

4.2 Navigation

For navigation of robots we use random rapidly tree (RRT) as a path planning with some customization that limit random trees in order to reduce calculation time.
In time optimization part we use Bang-Bang trajectory and PI coefficient in some cases (when high accuracy is needed like direct kick or penalty).
Nevertheless, observation results wasn't eligible so we have revisioned whole system and system's delays. We found out an important delay which is not covered (around 0.3s). This delay caused our AI software observe the game 0.3s later than real-time. To cover this delay and put AI system in real-time we chose to add robot's position (ΔX) to their observed position (X) by each wheel velocity feedback from robots to AI.

4.3 Motor Controller tuner

Due to important problems in controlling robots we spent more time on it and re-designed this part.
To make this part better than before we programed a software in MATLAB to receive, save and analyze motor velocity. With the velocity of motor which is transfer by nRF, in first step, open-loop transfer function of each motor will be available. Then with MATLAB's own pi tuner, coefficient is calculated.
But obviously for effete motors the best transfer function (optimized for time rise) is different and cause an error in robots global movement. So we need that motors have same closed loop transfer function, when motors have the same functions they behave same together, therefore we add the ability to change the coefficients manually and see the results in diagrams. Curves should be coincident two by two (because wheel's angle).
With this ability we reduce most of the global movement errors in the robot.

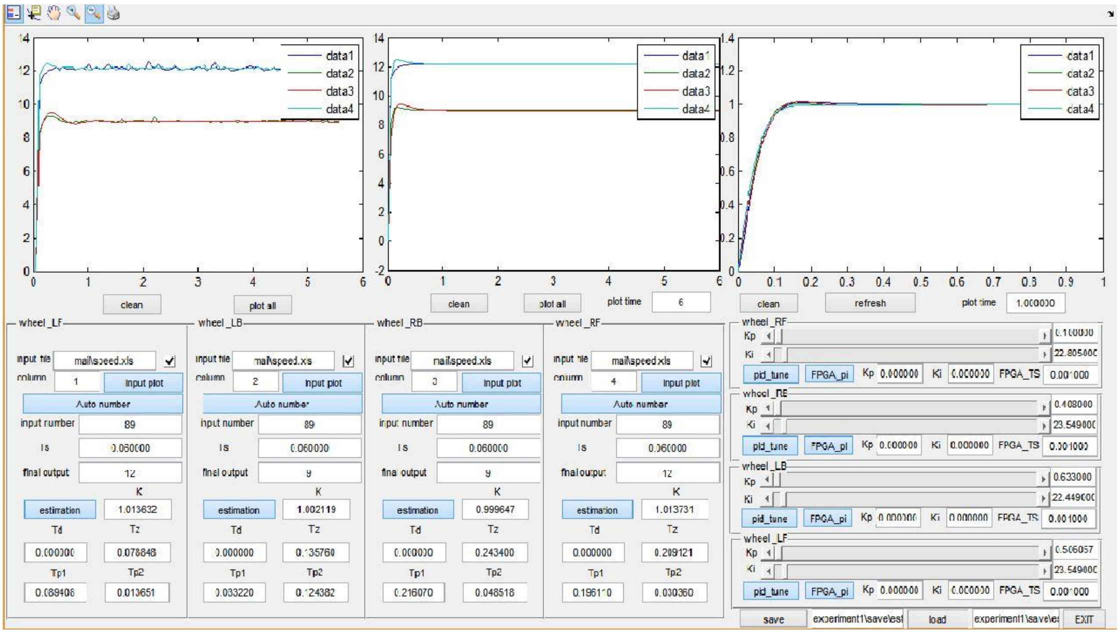

Fig. 9. Screenshot of Motor sampler software in MATLAB

References

1. Xilinx : Spartan-6 FPGA Family data sheet.
2. MathWorks. : Learning MATLAB.
3. Nordic Semiconductor. : nRF24L01+ Single Chip 2.4GHz Transceiver Product Specification v1.0 (2008).
4. Sebastian THRUN, Wolfram BURGARD, Dieter FOX. : PROBABILISTIC ROBOTICS.
5. Rudy Severns. : DESIGN OF SNUBBERS FOR POWER CIRCUITS.
6. Kanjanapan Sukvichai, Piyamate Wasuntapichaikul, Yodyium Tipsuwan. : IMPLEMENTATION OF TORQUE CONTROLLER FOR BRUSHLESS MOTORS ON THE OMNI-DIRECTIONAL WHEELED MOBILE ROBOT
7. Li-Chun Lai, Chia-Nan Ko, Tsong-Li Lee, Chia-Ju Wu. : Time-Optimal Control of an Omni-Directional Mobile Robot.
8. Omid Robotics Team. : OMID 2016 Team Description paper. Technical report, ECE Department, Shahed University of Tehran, 2016.