

# Immortals 2016 Team Description

Mohammad Reza Niknezhad<sup>1</sup>, Ali Salehi<sup>2</sup>, SeyedAli GhaziMirSaeed<sup>3</sup>, Ali Amouzandeh<sup>3</sup>, MohammadHossein Fazeli<sup>4</sup>, Mohammad Tabasi<sup>4</sup>, Mustafa Talaezadeh<sup>5</sup>

<sup>1</sup> Department of Computer Engineering of Iran University of Science and Technology

<sup>2</sup> Department of Electrical Engineering of Sharif University

<sup>3</sup> Department of Mechanical Engineering of Tehran University

<sup>4</sup> Department of Mechanical Engineering of Iran University of Science and Technology

<sup>5</sup> Department of Computer Engineering of Tehran University

**Abstract.** Below, we have classified information in 3 major parts: mechanics, electronics and software. Pictures of available instruments are attached.

## 1 Introduction

“Immortals” is a robotic team consisting of Sharif and Tehran university students. The team was formed in 2003 to attend junior soccer league competitions. The small size project started in summer 2007 and simple-structured robots were made by summer 2008 and have continually improved since then. After participating Robocup 2009 Graz and Robocup 2010 Singapore competitions, it was decided to gear up for the next competitions with revising the whole Mechanical system and equipping the robot with a sophisticated electronic system to extract the maximum efficiency and maneuverability from the robots. Inheriting all preceding robots’ strengths while decreasing their weaknesses, a brand new generation of robots was set to participate in the Robocup 2011 competitions. Those robots were granted the 2<sup>nd</sup> place at Robocup 2011 Turkey.

Now with some minor modifications and adjustments in hardware aspects, the main focus of the team is more sophisticated AI and improvements in the software-side, that are described briefly in this document.

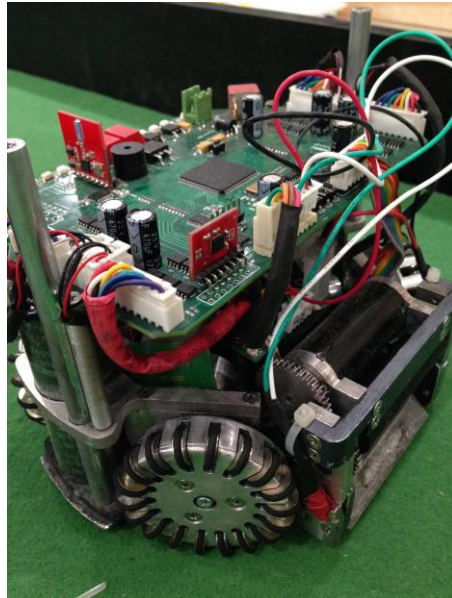
## 2 Mechanics

The mechanical design of robots has no major change since 2012 and just some part of kicking system and dribbling system optimized for better performance. So by this point of view, we explain the mechanics details.

The main chassis consists of a 5mm hard anodized 7075 aluminum alloy plate on which motors, batteries, encoders, capacitors, and kicker systems are mounted. Robots are covered by a carbon-fiber compound that is light as it is solid.

To stabilize movements and kicks mechanically, the altitude of the center of mass is lowered by arranging rather heavy parts on the plate, that level first in the whole robot structure.

The overall height of the robot is 148mm and its diameter is 178mm.



**Fig. 1.** Current Immortals Robots

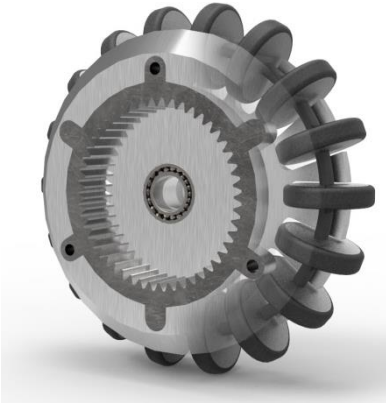
## 2.1 Wheels

Robots are designed omni-directional by using four omni-wheels. Each wheel is 61mm in diameters and has 18 rollers covered by double seal O-rings for better grip. Both wheels and rollers are made of 7075 aluminum alloy.

As the field size has been increased, higher speeds are inevitable; so the gear ratio is optimized to 45:12 and 50 Watts Maxon EC-45 brushless motors are used to overcome the max-speed/max-acceleration trade off. The gearbox is merged into wheels to minimize the volume occupied by driving system. Its design is unique so that the gear used for transition is carved into the wheel itself.



**Fig. 2.** Merged Gear in Wheel



**Fig. 3.** Faded view

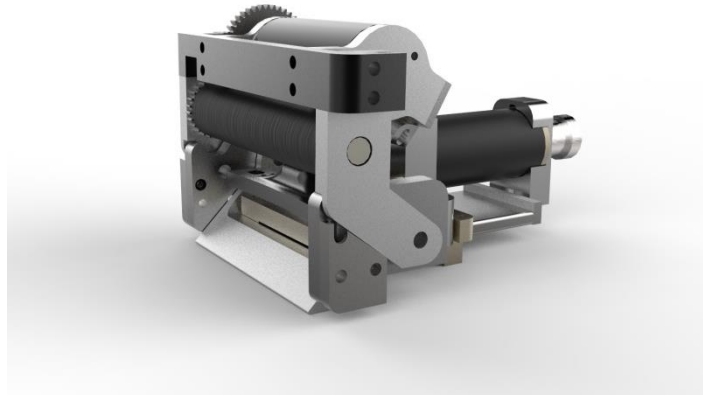
One of the major issues in the robots was inaccurate low-level navigation. After many experiments, measurements and inspections on the control system, it came out that non-identical mechanical properties of each gear are the biggest issue, which was caused by manufacturing inaccuracies. To have smoother and more accurate movement, gear module is redesigned with smaller module size this year. Also there are a little change in sub wheels design and tolerance to reduce unsteadiness and to stick better to the field during fast accelerations.

## 2.2 Casing

Although safe navigation has been a must for years in the league, highly dynamic nature of the SSL causes unavoidable collisions that lead to damages ranging from minor to severe. To avoid damages to internal parts, especially electronics which are fragile, making a robust, solid casing that can efficiently damp impacts is a necessity. Thus, a case made up of 2mm thick carbon-fiber compound is made which is as solid as it is light to cover the whole body but kicking devices and the dribbler.

## 2.3 Kicking & Dribbling System

To account for the increase in field size, the kicking system is redesigned and to have more power and accuracy in booth chip and direct shoots. Also because of some high voltage short circuit caused by magnet solenoids, in our new version whole kick system is isolated from body and there isn't any direct metal to metal contact by using resin as an isolator material.



**Fig. 4.** Kicking and Dribbling System

### 3 Electronics

The electronics design of robots has no major change since 2012 and our kicking board is just optimized for better performance. So by this point of view, we explain the electronic details.

A Xilinx XC3S400 chip functions as the only processor on the main board and operates telecommunication, decodes IR sensors data, drives motors and executes PID controller. This chip was chosen because of its low power consumption and its huge logic gate numbers in comparison with other similar products.

A TSK-3000A soft processor is implemented in the FPGA and operates as the main processor. The following features are implemented inside the TSK3000 processor:

- An error detection system, that detects many errors, including low-voltage situations, voltage spikes, short circuits, over-temperature, daughter boards' malfunction, packet drops, and sensor reading errors, motor stuck situations and much more. Then these errors are written to the flash memory for later diagnostics. But some of them are categorized as critical, and the processor halts the function of the malfunctioned part, and report the situation by making an error-specific sound, turning the STOP led on, and sending it over to the main

computer.

These errors should be cleared by using the diagnostics software, and the person erasing it should solve the problem before clearing it.

- Acting as an odometer, calculating the usage of each part, including BLDC motors, voltage booster unit, servo motor, rotary encoders and the wheels. These data then are used by the team, to know when to change or check each unit.
- Providing real-time diagnostics data over wireless link, such as each motor speed, voltages of some critical points, state of debugging switches, IMU data and voltage booster state.
- PID control loop with torque converter for each motor. The control loop frequency is 1.2 kHz, and delay time is 122  $\mu$ S. The loop is so time-critical, and so is written in pure assembly.

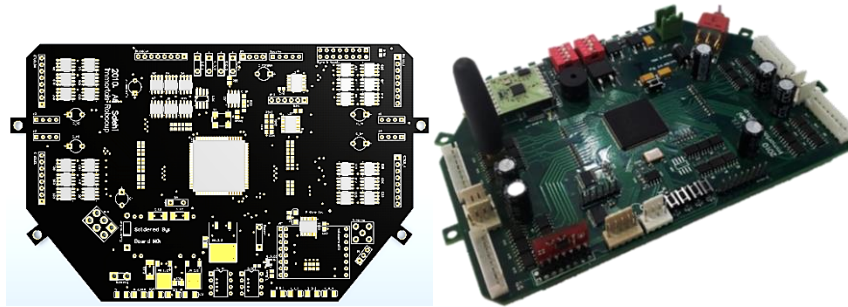


Fig. 5. Main electronic board

### 3.1 Wireless Communication

Robots use nrF 24L01 multi-channel radio transceiver, operating within the 2.4 GHz band. The module operates in user selectable channels.

## 4 Software

The overall software architecture is to take information of the field from SSL-Vision over network, filter this information and pass them to the AI. The AI calculates a target for each robot. Finally, it transmits the processed data to the robots, in each frame.

#### **4.1 Calculating World State**

Kalman filter is used for reducing noises, and predicting the future state. There are two types of input for the filter. For our robots, we use the data that is sent to the robots in previous frames, and for the ball and opponent robots, vision-calculated data is used.

#### **4.2 Decision Making**

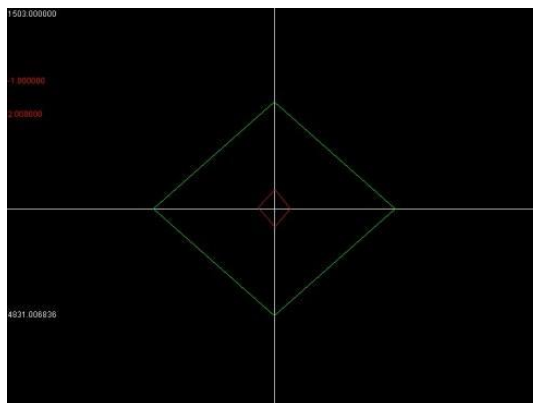
The main method for this part is STP, developed by CMDragons and is tested since 2003, and the results show the success of this method. STP consists of Skills for executing the low-level actions that make up robot behavior and tactics for determining which skills to execute, and Plays for coordinating synchronized activity amongst team members. In our system, skills and tactics are script files, and plays are generated by either script files or the visual Strategy Maker software.

#### **4.3 Dynamic Safety Search**

Dynamic Safety Search (DSS) is a multi-agent sampling-based dynamic path planning method. In contrast to positional planners (e.g. ERRT) it considers each agent kinematic parameters and current velocities in planning. DSS is proposed by J. R. Bruce<sup>[8]</sup>, and is an improvement over the well-known Dynamic Window method. The main improvements are:

- Replacing grid-base sampling with random sampling
- Being multi-agent, meaning it can generate a safe path for any number of agents

The set of possible accelerations used in DSS calculation is also recalculated, using our actual robot model which is based on BLDC motor equation. The author proposed that acceleration space plot of the set is likely to be a partial ellipse, but based on our computations, it actually is a diamond (fig. 6).



**Fig. 6.** Acceleration space plot of the set based on BLDC equations

In our AI software, the output of the ERRT planner is fed into the motion planner, and an acceleration vector is generated for the robot. Then this vector is passed to the DSS as the input, and a safe acceleration vector  $f$  is generated.

On the implementation side, the DSS algorithm generates the result for all robots simultaneously, so it cannot be parallelized. But because of its anytime basis, it can be terminated whenever required, and it will give the best found result. In our experiments, it almost always gives good enough results in the given 0.2 mS time.

#### 4.4 Strategy Maker

Because it is harder to code strategies even with scripting, so a tool had to be built to make it easy for everyone to propose strategies so we develop a visual editor tool for making the strategies and send the final to AI as a message containing strategies. The messaging is implemented using Google protobuf [12]. This message can be delivered to the AI using either UDP or a file. Then the AI software use these strategies in every free kick to assign roles to the robots. So with this tool we can choose when the robot can start the play and the other robots role. In each game the AI have to select one of the strategies depends on their probability which is assigned manually. In this year we let the AI change the probabilities automatically based on the scoring and result of running that strategies to do the best. The final roles in this software can choose between the wait for pass on the position, wait for pass in best position, attract the defenders, run away, go to position, shoot directly to goal and etc. Fig 5 is a screenshot of the strategy maker software that shows a

strategy for the corner kick. In this strategy if all the robots reach their targets, the purple robot pass directly to the red robot to pass-shoot to the goal.

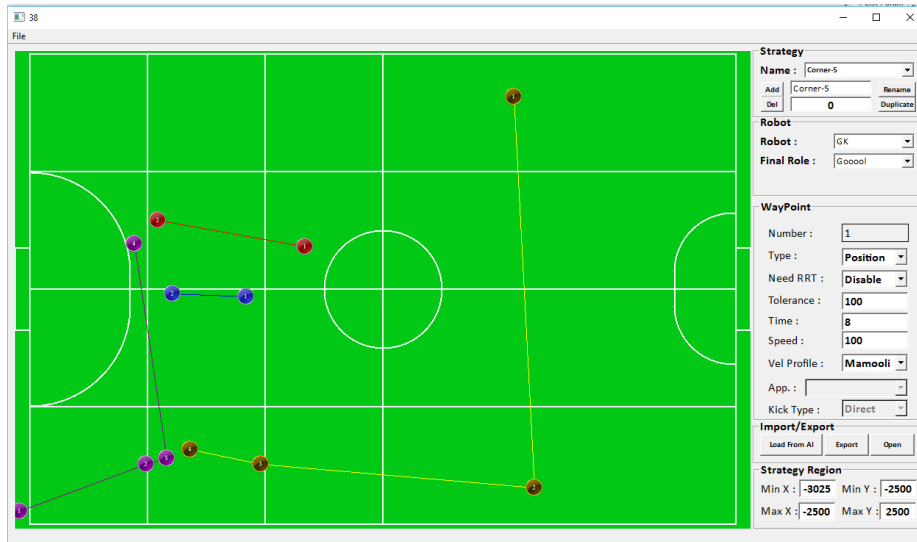


Fig. 7. Strategy Maker Software

## References

1. Zickler, S., Vail, D., Levi, G., Wasserman, P., Bruce, J., Licitra, M., Veloso, M.: *CMDragons 2008 Team Description*. In: Proceedings of Robocup 2009.
2. Srisabye, J., Hoonsuwan, P., Bowarnkitiwong, S., Onman, C., Wasuntapichaikul, P., Signhakarn, A., et al.: *Skuba 2009 Team Description*. In: proceedings of Robocup 2009.
3. Van Goch, B.P.T.: *Optimizing a solenoid for a Robocup kicker*, University of Technology Eindhoven.
4. Ruud Tilburgs, *Design and realization of a solenoid for a Robocup kicking device*.
5. *Lenses selection and setup*, [www.theimagingsource.com/en](http://www.theimagingsource.com/en).
6. Bruce, J., Zickler, S., Licitra, M., Veloso, M.: *CMDragons 2007 Team Description. Technical report, Tech Report CM -CS-07-173*, Carnegie Mellon University, School of Computer Science.



7. Browning, B., Bruce, J.R., Bowling, M., Veloso, M.: *STP: Skills tactics and plans for multi-robot control in adversarial environments*. In: *Journal of System and Control Engineering*.
8. Bruce, J.R.: *Real-Time Motion Planning and Safe Navigation in Dynamic Multi-Robot Environments*. PhD thesis, Carnegie Mellon University
9. Bruce, J., Veloso, M.: *Real-time randomized path planning for robot navigation*. In: *Proceedings of IROS-2002*, Computer Science Department
10. Browning, B., Kaminka, G., Veloso M.: *Principled Monitoring of Distributed Agents for Detection of Coordination Failure*. In: *Proceedings of DARS-2002, the Seventh International Symposium on Distributed Autonomous Robotic Systems*, Fukuoka, Japan, June 2002.
11. Behnke, S., Egorova, A., Gloye, A., Rojas, R., Simon, M.: *Predicting away the Delay*. In N.N. (editors): *RoboCup-2003: Robot Soccer World Cup VII*, Springer, 2004.
12. *protobuf - Protocol Buffers - Google's data interchange format*, [code.google.com/p/protobuf/](http://code.google.com/p/protobuf/).