Minho MSL - A New Generation of soccer robots

Fernando Ribeiro, Gil Lopes, João Costa, João Pedro Rodrigues, Bruno Pereira, João Silva, Sérgio Silva, Paulo Ribeiro, Paulo Trigueiros

Grupo de Automação e Robótica, Departamento de Electrónica Industrial,
Universidade do Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
fernando@dei.uminho.pt, gil@dei.uminho.pt, joaovazcosta@gmail.com,
pedro50073@gmail.com, brunomiguel1987@gmail.com, jsilva86@gmail.com,
sergio.fil@gmail.com, paulorogeriocp@gmail.com, paulo@trigueiros.com

Abstract. This paper describes the new generation of Minho Team robots since the last RoboCup participation. Robots have been rebuilt from top to bottom, keeping the essence of the previous generation, but lighter, faster, improved sight, new algorithms and new technology. The demands of RoboCup MSL are very high and a new breed of robots was necessary to cope with it. This paper describes a bottom-up view of the robot, the hardware used, the vision head, the high level software and some conclusions.

1. Introduction

Minho Team has been participating in RoboCup almost from the beginning, improving their robots year after year and in these last few years the team was forced to a halt due to several internal and personal reasons.

The previous generation of these robots was started in 2003, and now a new generation is born. The robots were almost completely built in the University of Minho Robotics Laboratory (mechanics, electronics and software), having the main motor controller board being developed by a spin-off of the University of Minho made by previous MSL team members. The following descriptions on this paper are related to just a single robot since the others are a copy of it. The robot's full structure and parts were first drawn in CAD software (Fig.1) in order to test if all components would fit physically.

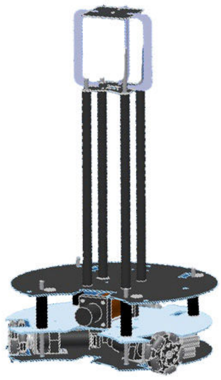


Fig. 1 - Robot 3D CAD model

Similarly with the previous generation, layers of circular aluminium boards with 500 mm diameter were used creating platforms where the hardware could be laid and fixed. Driving and steering is achieved with three omnidirectional wheels phasing 120º from each other. The ball kicker is coil based with very high power discharge for a full kick taken from a battery of capacitors. The robot runs on LiPo batteries for lighter and powerful energy. Driving, kicking, power, control boards and processing are all fixed to the base platforms. For an improved sight a tower rises in the robot's centre to enable and positioning the catadioptric vision system at the robot far top, just below the 800 mm. A compass and the network connection reside right underneath the camera.

The robots are constantly being tested at Minho Team Laboratory, and the tests have proven that the hardware is feasible and stable. This paper follows with the description of the structural arrangement and parts displacement, and by the newly developed servo-motor driver and the technologies used to improve the robot's vision. The last part is dedicated to the software development in object oriented programming.

2. Hardware

Hardware changes have been deeply made since the last participation in RoboCup. Although the base platforms are similar to the former version, major changes are visible from hardware arrangement to electronics. The tower and the head has a new design to support a camera and a custom made convex mirror.

2.1 Hardware Layout

The Hardware layout of the new MSL Minho robots is shown in Fig.2.

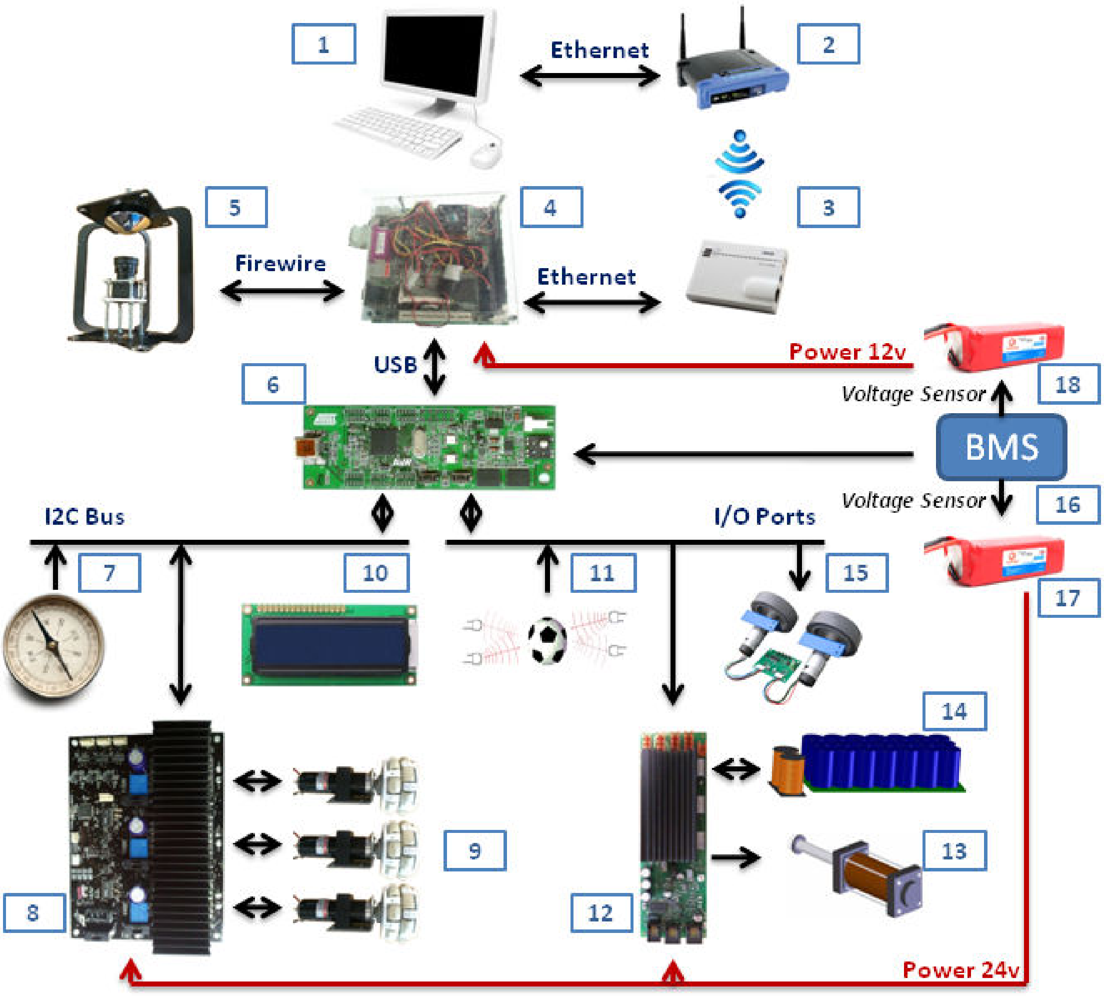
At the highest level, a base station defines and controls the game strategy, receiving and transmitting information to and from the robots via wireless. Each robot has a MB890 Mini-ITX motherboard 12VDC 5A with 1.7 GHz processor that receives images from a PointGrey Flea 2 Firewire camera and processes them. After processing a captured image and virtual sensors information, it sends high level action commands to the low level control system via USB.

The low level control system is an Atmel AT90 USBKey microcontroller board, working as a *Hardware Abstract Layer.* It receives high level commands and interprets them according to the specifications of the Hardware used devices. The control of these devices can be made in two ways: serial communication TWI (Philips I2C compatible) or Digital I/O ports.

Through TWI communication several devices are controlled such as a 500W 24VDC Omni3D-MAX motor control board which controls three 150W DC Maxon motors in closed loop, an electronic compass, a battery management system that reports the battery level and charges them when the robot is connected to mains and an LCD for debugging purposes. Through Digital I/O ports, the microcontroller board reads the state of infrared sensors, then sends a pulse width modulation signal to the kick control board to control the power of the kick and controls the dribbler motors via PWM. The status of these devices is sent at a fixed rate to the top-level also via USB.

The motor driver OMNI3D-MAX was recently developed and it is commercially available. It was made to drive three 500 W DC motors with quadrature encoders for omnidirectional three wheeled platforms. This high power small board generates real throughput to the motor demands in difficult conditions. Communication with the board is carried out via an I2C connection. A watchdog protects the board from running when there is no speed updates.

Should something happen to the robot's brain and no speed messages are sent to the board, the robot stops immediately preventing any collisions. This board controls three DC motors of 150 W each. Fig. 3 shows the motor driver electronic board.



Legend:

1. Base Station
2. Base Station's Router
3. Robot Computer's Router
4. Robot Computer
5. Robot's Head (with camera and mirror)
6. Atmel AT90 USBKey (low level control system)
7. Compass
8. OMNI3D-MAX Motor Driver Board
9. Three DC Motors (with Wheels and Encoders)
10. LCD
11. Infra Red System
12. Kick Control Board
13. Kick Coil
14. Battery of Capacitors
15. Dribbler System
16. Battery Management System
17. Battery 24V
18. Battery 24V

Fig. 2 - Layout of the hardware connections

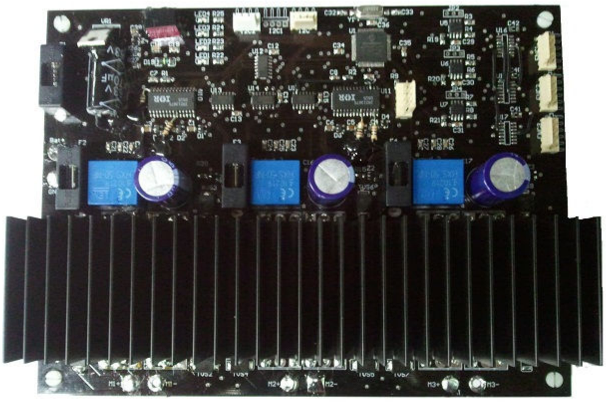


Fig. 3 - OMNI3D-Max: Triple motor controller with quadrature encoder input

2.2 The kicker system

The kicker is based on a controller board, a battery of capacitors (4.3 mF) and a magnetic coil that impels the metallic rod to kick the ball. Fig. 4 shows the different modules. The kicker controller is a board that charges up the battery of capacitors with a potential of 344 Joules (@400V) and waits for a command to discharge its energy through the magnetic coil of 3.8 mH through a small pulse of just a few milliseconds. The pulse width defines the amount of power for the kick, and it can be very low to perform just a ball pass. A discharge pulse of 15 ms generates the maximum kick power.

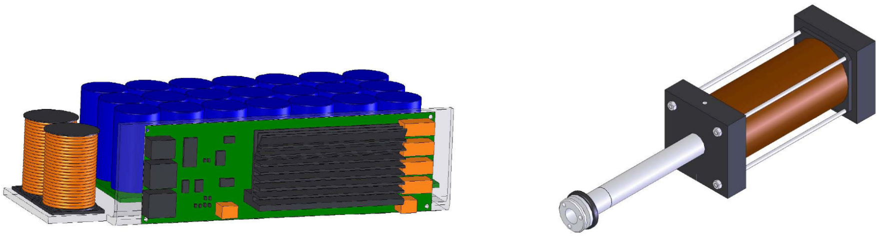


Fig. 4 - Controller board and battery of capacitors (left) and magnetic coil kicker (right)

2.3 Battery management system

The battery management system was developed to charge and monitor the state of all batteries on-board the robot. There are two LiPo independent sets: one with 24V is used to give power to the motor drive board and one with 12V is used to give power to the central computer. This system informs the central computer of the battery state enabling it to adjust the robot tactics according to the power needs and availability. The battery management system has also the ability to charge the batteries when connected to mains.

2.4 Compass

The robot orientation is gathered by a CMPS03 Electronic Compass (Fig. 5). The compass uses a Philips KMZ51 magnetic field sensor, which is sensitive enough to detect the Earth’s magnetic field. The output from two of them mounted at right angles to each other is used to compute the direction of the horizontal component of the Earth’s magnetic field. This device has an I2C bus communication and a resolution of 0.1 degrees.

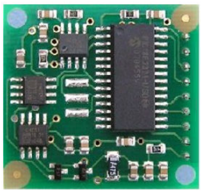


Fig. 5 - CMPS03 Compass

2.5 Infrared sensors and dribblers

There are two infrared devices placed at the robot’s bottom front in order to detect when the ball is in the position to kick. The board signal conditioning and the device supporting the infrared are shown in Fig.6.

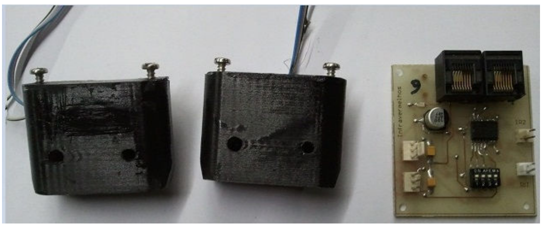


Fig. 6 - Infrared sensors and signal conditioning board

For better ball control there are two DC 12V motors, controlled by the microcontroller board via PWM according to the information of the high level control. These two motors have small rubber wheels to pull the ball against the robot for dribbling it.

2.6 Low level control system

The Atmel AT90 USBkey development board (Fig. 7) is responsible for managing low-level hardware, working as *Hardware Abstract Layer*. This USB evolution kit is designed around Atmel’s AT90USB1287 microcontroller, offering full speed OTG Host and Device USB interface in addition to a broad set of general microcontroller peripherals and serial communication interfaces like TWI (Philips I2C compatible), SPI and UART.

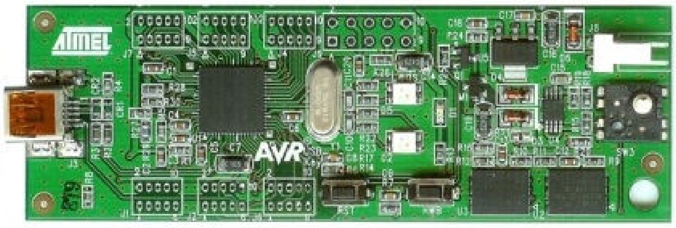


Fig. 7 - Atmel AT90 USBKey

The use of this microcontroller has two main goals: abstraction from the hardware high level usage and reducing the processing time of the main computer.

The high level processing is independent of the type of Hardware used. The high level always sends the same command without knowing the type of Hardware that it controls, e.g., the command kick: The high level processing only have to send the command *Horizontal_kick (intensity[0 .. 100])* and the low-level microcontroller is responsible for interpreting the command and control the device used to kick. This method also reduces the processing time of the high-level processor, leaving more time available for image processing and localization and strategy algorithms.

In general, the microcontroller board receives from the high level processing the locomotion and kicker commands at a rate set by the cycle of image processing thread. The microcontroller sends to the high-level processing the robot's orientation and the state of the Infrared to a fixed pace. Other information, such as debug information, hardware tests or log files can also be sent in both directions.

3. Artificial vision system

The vision head uses a catadioptric omnidirectional system as shown in Fig. 8 consisting of an upright Firewire PointGrey Flea 2 camera facing the centre of a convex mirror, making it possible to see all around the robot.

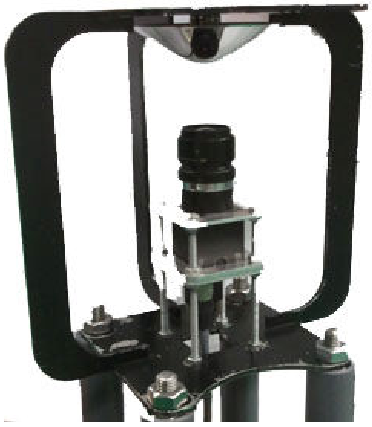


Fig.8 - Catadioptric vision system

The image processing is performed in two parts, one made offline and the other in real time. In offline mode a look up table is created where the pixels addresses are stored to be later on analyzed. In real time an image segmentation is performed in order to detect the colors on the surrounding areas during the game. This segmentation defines boundaries in HSV colour space, since it is more immune to variations of luminosity than the RGB colour space.

4. High Level Software

The high level software is the brain of the robot. It commands the whole robot performing its localization, motion behaviour and tactics.

4.1 Localization algorithm

The algorithm for localization uses radial lines to search for the field's white lines and it is performed in 5 degree angles as shown in Fig. 9 to a maximum of 216 points per line. When the number of detected white line points is low (around 20), an axial search is performed, as it is shown in Fig. 10 which allows more points of the white lines to be detected. The points are stored online for later use for the localization process.

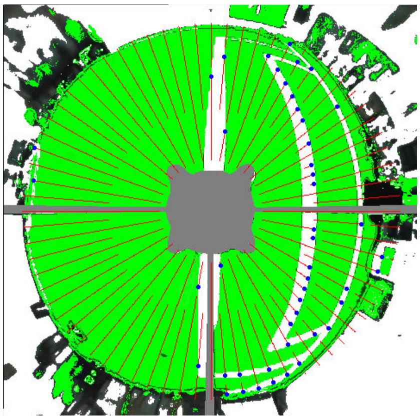


Fig. 9 - Radial Search

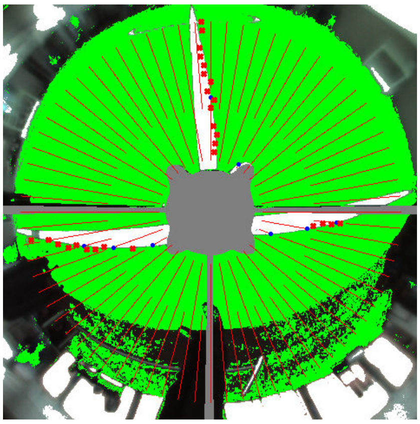
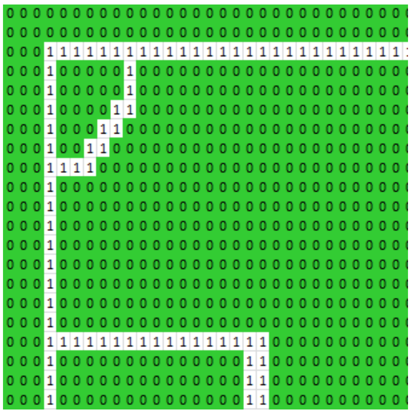


Fig. 10 - Axial Search

A binary game field map is stored in file and consists of 0s and 1s where 0 means the green field and 1 the white line as shown in Fig. 11. The algorithm seeks the file for each position and keeps the distance to the closest line in a matrix corresponding to the size of the field, with a resolution of 0.1 m.



```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
```

Fig. 11 - Binary game field file loaded in memory

The points obtained from the radial search are converted to real distances and a search is made in the distance matrix and the sum of the distance errors is calculated for all the points. The location with the smallest error sum represents the location of the robot on the field.

4.2 Robot dynamics and the rule based game behaviour

Nonlinear dynamics is used to control the robot's motion defining attractors (ball) and repulsors (opponents) thus enabling the robot to move in the field, being attracted by the game ball and repelled by other robots. These algorithms follow a new concept of rule based game behaviour where the robot performs different conditions depending on its state, field position, ball position, opponent positions and partners positions. The conjugation of different factors makes the rules valid and therefore, the robot behaviour will be defined accordingly.

4.3 External monitoring - the coach

The coach is one of the centre pieces of a robot soccer team. It has a set of tools to control and monitor the state of the robots during the game and allows the creation and selection of tactics for the robots. The entire process of creating tactics is performed according to the type of game situation. For the kick-off, throw-in, goal-kick, corner-kick and penalty kick, some positions are defined on the field for ball repositioning and command the robot to the desired positions. A linear interpolation is calculated between the two closest points previously defined in order to know what the positions of the robots in a particular situation. For free-kick, the implementation is similar to the previous one. That is, the points defined in the field to free-kick are grouped into triangles, and during the game an interpolation between the triangle points that belongs to the real ball position is calculated (Fig. 12).

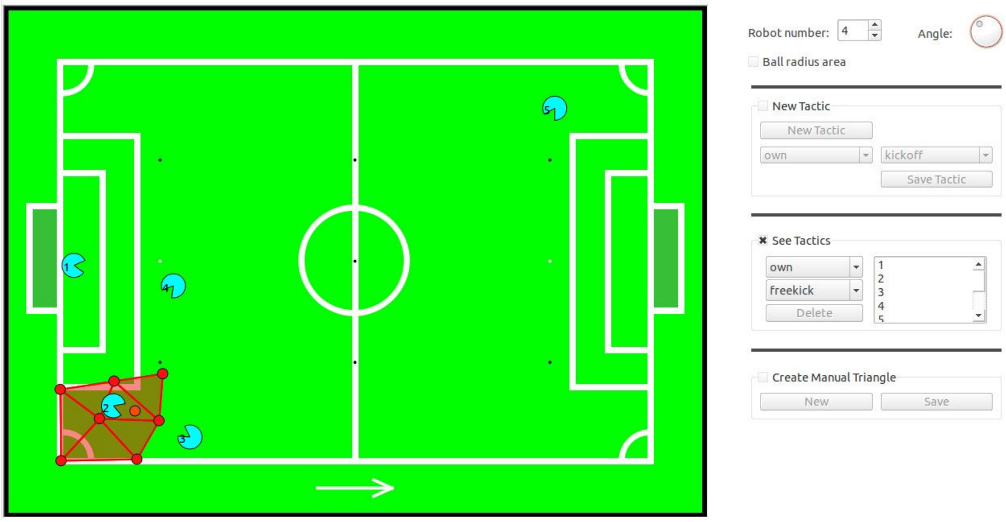

Robot number: 4

Fig. 12 - External monitor (coach) screen

During the game, in addition to the above calculation, the robots position in the set pieces is also defined by the coach that receives the commands from the RefBox and sends them to the robots.

The monitor also shows the battery level of the robots, the role and behaviour that is being played by each robot, the distance that relative positions each robot sees the ball and a field where each robot is designed according to the position and orientation obtained from the localization algorithm. Any communication errors between the robot and the coach are logged.

As just described, the coach is considered a key element in a robot soccer team, both in control and monitoring the robots during the game as the creation of tactics for the set pieces.

5. Conclusions

Four years later and after participating in other league, Minho Team is back with a new set of MSL robots just built from scratch. The experience from previous generations allowed to take some shortcuts on decisions avoiding major problems. New hardware was developed, to make the robots more reliable, faster and optimizes. A completely new set of software was created with a different programming tool with new ideas. Localization is achieved and working properly. The image processing is even faster than in previous generations making control a little more easy to tackle allowing more promising results.