FURGbot 2014 Team Description Paper

Sidnei Carlos da Silva Filho, Eduardo do Amaral, Gerson Urban, Matheus Machado, Joel de Oliveira, Márcio Gonçalves, Floricio Teixeira, Alexandre Horn, Luan Silveira, Pedro Ballester, Vagner Rosa, Emanuel Estrada, Paulo Drews, and Silvia Botelho
sidneifilho123@gmail.com, eduardo.doal@gmail.com
gersonurb@gmail.com, matheusbg8@gmail.com, joelfelipe94@gmail.com, marcio-pel@ibest.com.br, floricio@ymail.com, alexandredecamposhorn@gmail.com, luansilveira.dp@gmail.com, pedballester@gmail.com, vsrosa@gmail.com, emanuelestrada@gmail.com, dudopel@gmail.com and silviacb@furg.br

Universidade Federal do Rio Grande, Rio Grande, RS, Brasil

Abstract. This paper describes an overview of the FURGbot team, a Small-Size league robot soccer team. The system is designed to participate in Robocup 2014. The hardware, software and mechanical components are described in this paper.

Keywords: Robotics, Artificial Inteligence, Control Theory

1 Introduction

The FURGbot Small-Size Team is a project of the Center of Computational Science from Federal University of Rio Grande, Brazil. Our objective is to stimulate robotic research and education in applications in the field of robotics, artificial intelligence, embedded systems and control theory. The FURGbot Team was created in 2006 by undergraduate students. We have won the Brazilian Championship six times and the Latin American Competition one time.

A more robust embedded system and a new off board software will be available for RoboCup 2014. This paper details our approach for this competition. Section two describes our mechanical structure. Section three exposes the new hardware modules. Finally, the new off-board system, based on the STP Architecture [1], is described on section four.

2 Mechanical Structure

The mechanical structure of the robot is made of an aluminium base and four omnidirectional wheels, see Figure 1. Each wheel is drives by an motor, with the coupling made by a gear to reduce the angular velocity. The base still has a dribbler device, made by a rubber cylinder coupled to the motor and kick devices, which uses a solenoid.

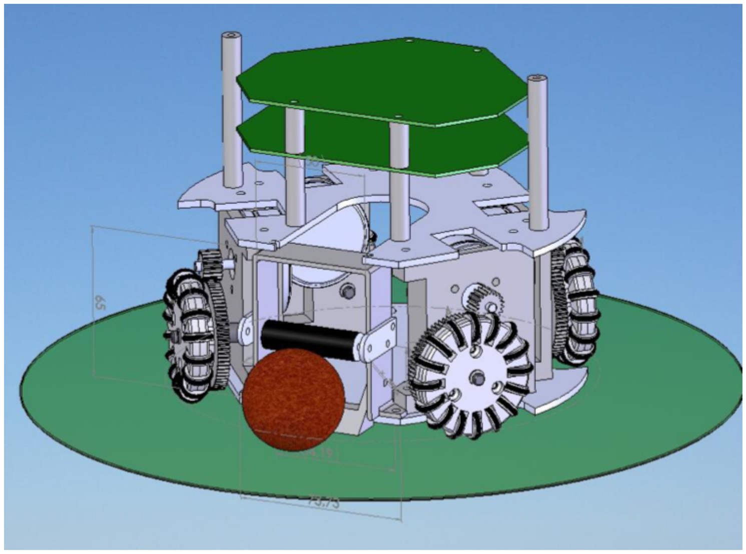

Fig. 1. Mechanical Structure.

3 Embedded Eletronics

The robot must be able to run, kick, pass and dribble, and moreover it needs to be compact and robust. Furthermore, due to the highly dynamic environment of the game, the robot needs to be fast, accurate and have low power consumption. Figure 2 shows an overview of our architecture, detailed in the next subsections.

3.1 Processing Unit

The robot presents a dribbling device and a high/low kick, allowing the performance of more elaborate moves. Furthermore, there is a motor speed control system that uses Hall-effect sensors, battery load control sensor, ball detection sensor and others. We use a Arduino Mega 2560 micro-controller to address the needs associated with the number of I/O pins, channels of PWM (Pulse Width Modulation), serial ports and analogue to digital converters.

3.2 Motors and Drivers

The motors used to drive the wheels are the model EC-45 - 30W from Maxon company. Unlike brushed DC motors, these motors have no physical contact and internal motor abrasion. However their drivers are more complex [2]. This engine model has internal Hall-effect sensors that makes possible to do a wheel speed control even with a low-resolution encoder.

To drive the dribbler we used another type of motor, the EC-16 - 15W. This model is smaller and faster than the other, but it has the same electrical features of the wheel motors.

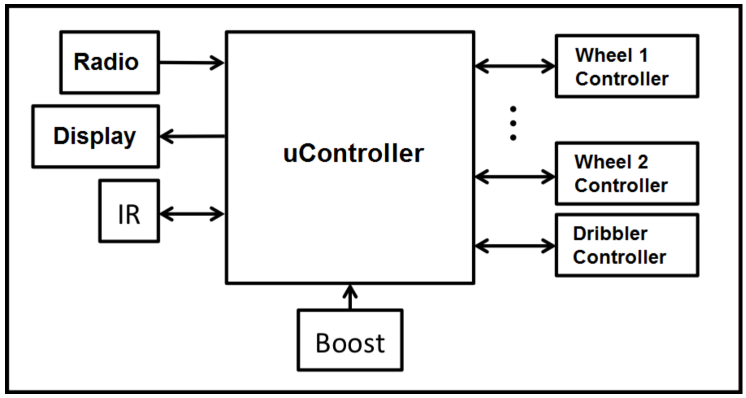

Fig. 2. Overview of Embedded Architecture.

For conversion of the PWM signal from the micro-controller to motors, the L6235 controller from ST Microelectronics is used.

3.3 Passing and Kicking

The difference between a pass and a kick is obtained by controlling the discharge of a capacitor bank. These capacitors of 470uF each are grouped by six. To charge the capacitors a DC-DC converter circuit is used [3], built on a separate board and isolated from other components in order to avoid noise that might cause damage.

Furthermore, to control charging of the capacitors, not allowing the charge voltage to exceed the maximum of 250V or reach a lower limit of 200V, voltage comparators were used. The circuit also allows, in accordance with the signal from the micro-controller, the way of the kick (high or low) and speed are selected, allowing more elaborate moves and passes.

3.4 Ball Detection

Infrared sensors are responsible for detecting the ball and allowing the kick. A signal is sent to the controller indicating the presence or absence of the ball. In previous versions, due to interference from ambient light, the sensors did not work correctly. The solution to the problem was the use of an integrated circuit capable of comparing the signal emitted by the infrared LED and received by the photo-transistor. Image 3 shows the operation of the designed circuit.

4 Software

With an significant increase of the embedded system performance, it was possible to invest in a more complex and robust off-board software architecture. This

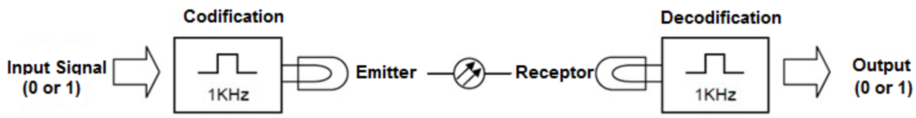

Fig. 3. Infrared Sensor.

system is divided in three main modules: the Tracker, the Data Manager and the AI Server. The components of the system are shown on figure 4.

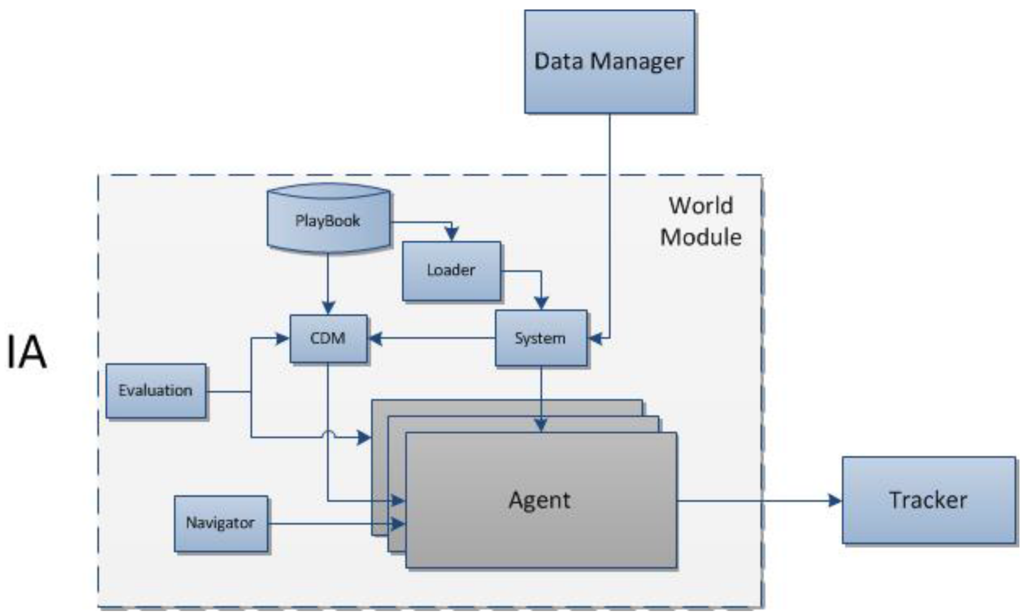

Fig. 4. Software Design.

The data manager module has to manage the input data of the system. It does the communication with the SSL Vision System and the Referee Box. After receiving the vision data, we need some extra computing in order to minimize the noises and delays in the information. In order to estimate more accurate positions for the robots and the ball, we choosed to implement an Kalman Filter [4] on the raw data received from the camera.

The position of the team robots are estimated with the last speed sent to the robots. For the ball and the opponents, the speed used is calculated processing the camera frames. After processing, the data is sent to the AI module. The AI process is constituted of an alternative implementation of the STP architecture. The STP architecture is a multilayer hybrid approach directed to dynamic and adversarial environments. With this architecture we have a mechanism that deals with long term goals, but is also able to react to fast changes in the environment. This makes the architecture well suited to the Robocup Small-Size category.

One of the main problems in implementing a more complex system is the increased computational cost. This creates the possibility of loosing information in case where the system has a slower response time than the SSL vision transmission speed. So, considering the multi-core processors we were able to made a change in comparison to STP architecture: the division between a sequential and a parallel stage.

On the Sequential stage there is the loading of the plays used in the game by reading files on the disk. The CDM module is responsible for making the most suited play for the current state of the World Model. Also in the sequential part there is the System module who communicates with the Data Manager and organizes the execution of the agents in the AI module.

On the parallel scope, each robot is represented by a execution ow called Agent. Each agent acts interpreting the tactics. From each tactic a state machine is generated. The states represent the reactive skills which depend on the robot perception on a world instance. Some of the main skills make targets to be reached. The navigator module provides a service for determining an obstacle free path to the goal. The algorithm used to resolve the possible paths is the ERRT [5] using the KDTREE structure [6].

When we decided to use program units running concurrently, it was obvious that the agents would have to wait some mutual synchrony moment for making cooperative tactics. To solve this problem the CDM module is also used to coordinate cooperative plays.

The evaluation module encapsulates a significant part of the computations made in the world by the agents and the CDM modules in order to make better decisions. The computations use mainly geometric calculations provided by the CGAL library [7].

One external part of the AI system is the Tracker process, responsible for visualizing the ongoing match. While the monitor is showing the match on the visualization screen, it saves all the speeds, strategies and positions of the match players, making it possible to reproduce the match later.

5 Conclusion

This paper presents an overview of the FURGbot 2014 project. The team consists of undergraduate students at the FURG. This year we have a more robust embedded electronic and a new off-board software. An overview of our architecture, mechanical structure and software control and strategy was presented.

As future work we intend to test and validate all modules, aiming to deal with uncertainties, noise and unanticipated failures.

References

1. Browing, B., Bruce, J.B.M., Veloso, M.N.: Stp: Skills, tactics and plays for multi-robot control in adversarial environments. (2005)
2. Kenjo, T., Nagamori, S.: Permanent- magnet and brushless dc motors. (1985)
3. Mohan, N., U.T.M., Robbins, W.P.: Power electronics: Converters, applications, and design. (2002)
4. Meinhold, R.J., Singpurwalla, N.D.: Understanding the kalman
lter, the american statistician. (1983)
5. Bruce, J., V.M.: Real-time randomized path planning for robot navigation. (2002)
6. Atramentov, A., L.S.: Efficient nearest neighbor searching for motion planning. (2002)
7. CGAL: Computational geometry algorithms li brary (n.d.). http://www.cgal.org. (2013)