IRSS Deluxe Team Description Paper

Gabriel Ahumada, Sebastián Arriagada, Cristóbal Nettle, Marcela Polanco,
Pablo Riquelme, and Tomás Rodenas

Centro de Robótica
Universidad Técnica Federico Santa Maria, Chile

Abstract. This paper describes the current development status of the recently formed SSL team, IRSS Deluxe, with the purpose to register for RoboCup SSL Open at LARC 2013. We present the initial design and implementation we have got so far in order to achieve this goal, showing the electrical, mechanical and software topics involved in our work, which were designed according to satisfy the RoboCup rules.

1 Introduction

This RoboCup team was founded at the ending of 2011, it consists of students from different disciplines such as mechanical engineering, informatics, and electronics engineering where we can found students specialized on computer science, control and automation, or power electronics. As a new emerging robocup team, and specially as the first SSL team on our country, we have earned enough financial support in order to build our first robot prototypes.

This document describes our design and the implementation we have got so far, showing all the work made in the different areas involved at this category. In particular, we describe mechanical design, electronics design for different devices and algorithms implementation for the (robotic) team coordination.

2 Mechanical Design

This corresponds to the first generation of robots of our team, satisfying the restrictions imposed on the rules:

- Height: 146 mm
- Diameter: 178 mm
- Maximum coverage of the ball: 18 %

Figure 1 shows the robot (with a provisional external case).

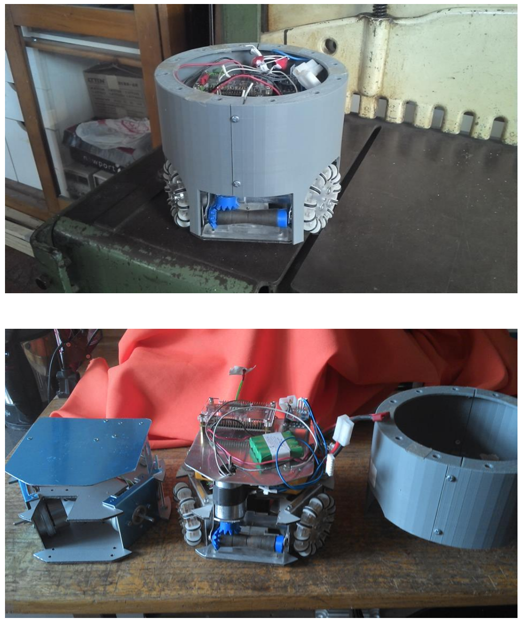

Fig. 1. Real robot

2.1 Drive system

As can be seen in Figure 1, each robot has 4 omnidirectional wheels, each one with 61 mm of diameter and 13 sub-wheels of 15.5 mm of diameter, so the robot can move in all directions.

Each omnidirectional wheel is driven by a motor Maxon EC 45, 30 Watts and integrated electronics.

This wheel is shown in Figure 2.

Fig. 2. Omnidirectional wheel

3 Hardware

Each robot is controlled by an Altera Cyclone IV FPGA, using DE0-Nano development board. This model was chosen because of its reduced size (given that it doesn't have so many peripherals, display, switches and VGA as other models), and it does include an accelerometer and ADC wich simplifies the electronic work.

We choose the FPGA more than a microcontroller because the simplicity for generating PWM and synchronic signals, and we also could achieve a multitask approach as movement, data receiving, sensor data processing wich leads to a faster action response.

3.1 Kicker

The kicker device consist of a boost circuit that elevates the voltage coming from a 12 [V] battery to 200 [V] charging a capacitor, saving 80 [J] that are discharged to the kicker solenoid. This circuit uses the L inductor characteristics injecting the current created switching the left mosfet of the diagram, closing the source-inductance path. This current remains and is passed through the upper diode to the C capacitor. The circuit implemented allows to control the voltage, and therefore the energy, in the capacitor. This makes controllable the strength of the hit to the ball. If the voltage is too high than the desired stroke, the mosfet controlled by the DS signal is fed connecting the resistance and discharging the capacitor through that path.

When the robot is ready to stroke, the R1 relay connects the solenoid to the circuit, discharging almost instantaneously the capacitor and finally shooting the ball.

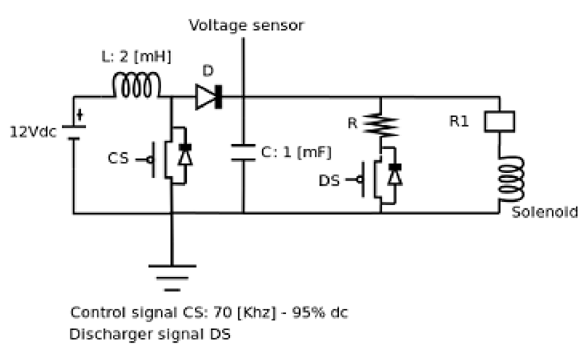

Fig. 3. Circuit for kicker device

3.2 Dribbler

According to RoboCup rules, the robot is allowed to cover up to 20% of the ball. Experimentally, it has been proved that it's easier to catch the ball when the dribbler has a slightly curve to center the ball on its own. So, this design involves two diameters, D_1 and D_2 and based on this information, maximum height possible is calculated and we obtain this expression:

$$H = \sqrt{\frac{1}{4}\left(D_2\left(2d + D_2\right) + D_1\left(4pd - 2d - D_1\right) + 4pd^2\left(1 - p\right)\right)} + \frac{d}{2}, \qquad (1)$$

where d and p corresponds to the ball's diameter and maximum coverage of the ball respectively. Figure 4 shows the relation between those variables.

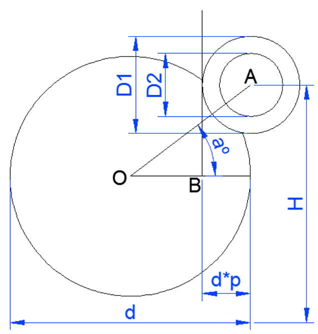

Fig. 4. Relation between variables involved in dribbler design

Then, on the implementation (which is shown in Figure 5) we use a DC Motor, Maxon RE-max 29, 22 watts and the dribbler is driven by a gear system, noting that those gears were built on a 3D printer.

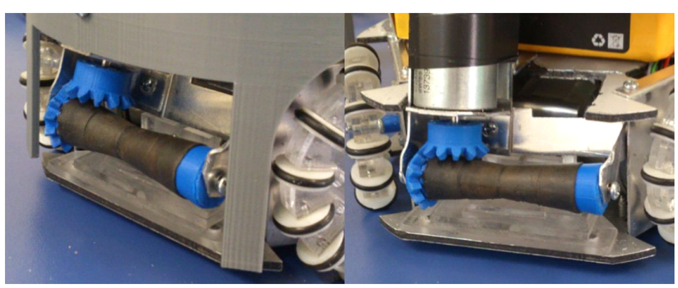

Fig. 5. Dribbler device

4 Communication

For wireless communication between robots and the main computer, we use Xbee Series 2 devices, selected by their transmission features, transmission range and power consumption. The network consists on a coordinator (computer), which sends sequential instructions to all robots in broadcast mode, where each robot (equipped with an Xbee END Point) codes only the first part of messages, recognizing if this message is intended to be for that robot or discarding in case it is for another one. Instructions protocol consists on 13 bytes per package, formed by:

- 1 byte: Pattern wich indicates that this is the first byte of the package and id of the selected robot. This byte also has the kicker and dribbler instructions.
- 1 byte: id of the selected wheel of the robot, this wheel will have a velocity (and direction) given by the next bytes.
- 2 bytes: This contains the velocity within a 12 bits resolution
- 9 bytes: These bytes are used for the other wheels (3 bytes per wheel).

This transmission is set to a 115200 baudes rate.

5 Software

An important aspect of a RoboCup team intelligence is Path Planning. We have tested different methods looking for a suitable algorithm which gives good results at the moment of avoiding obstacles.

The first method tested was Potential Field algorithms [2]. This proposes a potential field representation for obstacles and target, using sources for the prior and sink for the latter. In this way, vector trajectories are generated avoiding obstacles and leading the agent to the target, as we let a ball fall down. A disadvantage of this method, is that we could obtain local minimums without reaching the target.

The second method tested was Rapidly Exploring Random Trees (RRTs) [3], which consists on expanding a tree on the target zone, avoiding to add nodes that could produce collitions with targets. The added points to the tree are randomly chosen with probability p in a straight line to the target, and with probability $1 - p$ selecting a random point on the space, making more exploration and avoiding to get stucked on a different location to the target.

For improving its performance, we have implemented and tested some of the algorithms based on RRT, waypoints, smoothing and some extensions like RRT* presented on 2011 [1].

In order to simulate the robotic team coordination, and test different multi-agent algorithms, we make use of GrSim [4], software that has been very helpful to test game strategies.

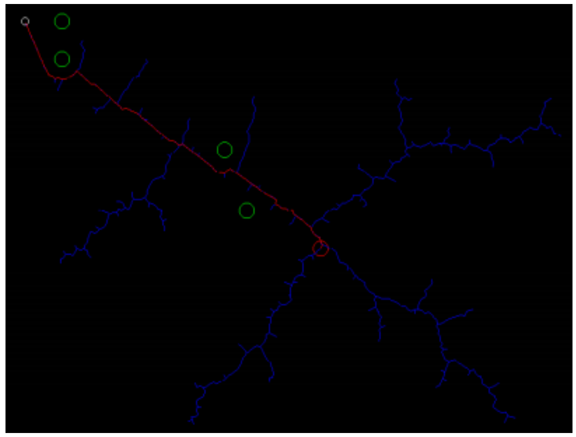

Fig. 6. Path planning simulation

6 Conclusions

This paper describes the current development of IRSS Deluxe team, which is a multidisciplinary team involving mechanical, electronics and computer science faculties. We have started the robots replication process and we expect to have an acceptable performance at competition.

References

1. Adam Bry and Nicholas Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 723–730. IEEE, 2011.
2. Shuzhi Sam Ge and Yan Juan Cui. New potential functions for mobile robot path planning. *Robotics and Automation, IEEE Transactions on*, 16(5):615–620, 2000.
3. Steven M LaValle. Rapidly-exploring random trees a ew tool for path planning. 1998.
4. Valiallah Monajjemi, Ali Koochakzadeh, and Saeed Shiry Ghidary. grsim–robocup small size robot soccer simulator. In *RoboCup 2011: Robot Soccer World Cup XV*, pages 450–460. Springer, 2012.