Team Description Paper: Team Emerotecos

Abílio Marcos Coelho de Azevedo, André Seidel Oliveira, Ivan Seidel Gomes, Yan Victor Ribeiro Marim, Matheus Pimentel Canejo Pinheiro da Cunha, Hudson Cássio Gomes Oliveira, Felipe Nascimento Martins

IFES – Federal Institute of Education, Science and Technology of Espirito Santo, Brazil
NERA – Núcleo de Estudos em Robótica e Automação

***Abstract*— This paper presents a description of an omnidirectional robot to be used on the RoboCupJunior Rescue B competition. The main robot controller was written in C++ and runs in an Arduino DUE board. The Arduino board is responsible for the distance and temperature sensors readings, and is connected to an MBed board that controls the motor drivers. There are eight infra-red distance sensors mounted on the top of the robot, four of them measuring the distance from the robot to the side walls, two measuring distance to obstacles or walls in front of the robot and the other two measuring distance to obstacles or walls in the back. The robot mechanical structure was designed using a 3D modeling software and was built using acrylic. It has four omnidirectional wheels that allow the robot to move in any direction without the need of turning itself. The robot was entirely designed, built and programmed by the team students.**

1. Introduction

This is to show team Emerotecos' strategies to solve the challenge proposed by the RoboCupJunior Rescue B competition.

To build the robot, we didn't use any building kit, as we designed the whole robot from sketch, using Dassault Systemes' Solid Works software [1]. The robot was built basically with 5mm thick acrylic and carbon fiber pieces, that we found are tough enough for the application.

As the main processor, we use an Arduino DUE board [2], which is programmed using C++. We choose the Arduino DUE board because, comparing to the Android mobile phone that we have used last year, it communicates faster with the motors and sensors as the board has many analogic and digital inputs. It also has support for I^2C protocol, which we are using.

2. Objective

The objective of this project is to build a smart robot that can cross a maze build with wood walls, and identify the electrically heated victims that are placed along the maze's walls.

3. Ambient

The Challenge happens in a modular arena, made with wood, which has 2 floors, and 4 main rooms. The location of the walls are always unknown by the time the robot starts running, so that it runs in a real maze. The only constant information about the arena is its total size, which makes it easier to the robot to find its way on the place. An illustration of the arena is shown in Figure 1.

Some heated "victims" are randomly positioned on some walls in the arena, and the robot has to identify them in order to score.

In some places, there are some "dead ends", which are a black mat on the ground. The robot can run over these black mats, but it has to leave it on the same side it came from, it can't cross the black area.

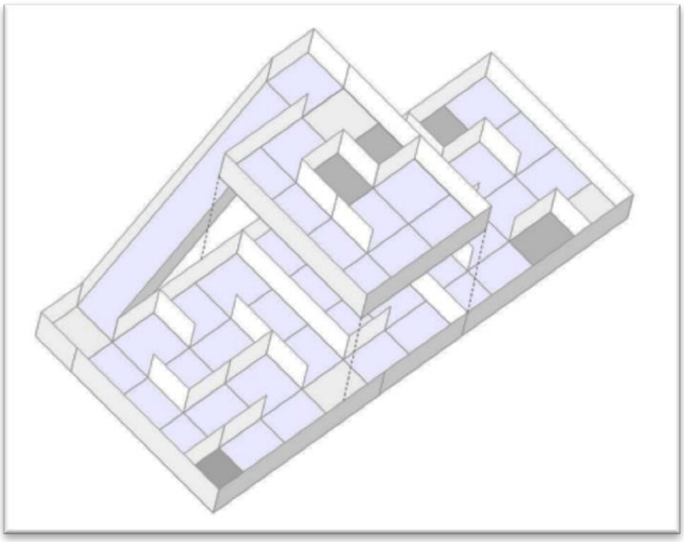

Fig. 1. Rescue B Arena illustration.

4. Strategy

A. Robot Structure

We have participated in the Rescue B category last year. As our robot had a satisfactory performance, we have decided to build our new robot basically like our first version. Both robots are shown in Figure 2. Although the structure is the same, we have made some important improvements that we hope will ensure a better performance is this year's competition. The robot's basic structure is built with an acrylic layer and a carbon fiber layer, with some acrylic beams. We use a Mecanum omnidirectional wheels system, which makes our robot able to move in every direction. Four strong motors, with encoders, drive the robot very precisely. We rely on 4 motors, one for each wheel, because the MECANUM omnidirectional system need independent control for each wheel. Each motor has a torque of 6kgfm, and its encoder can read up to 3,592 pulses per turn because we have a 75:1 speed reduction.

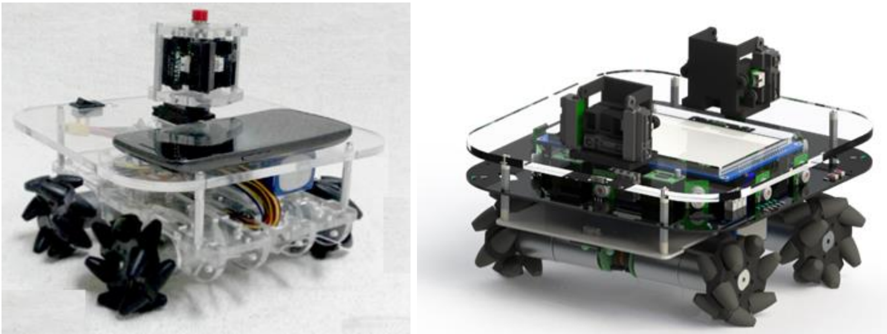

Fig.2. Last year’s robot (left) and this year´s robot (right).

The Mecanum system works as a regular omni system, and now we’re going to try a new suspension system to hold the motors. This is made of Kevlar and its design is illustrated in Figure 3.

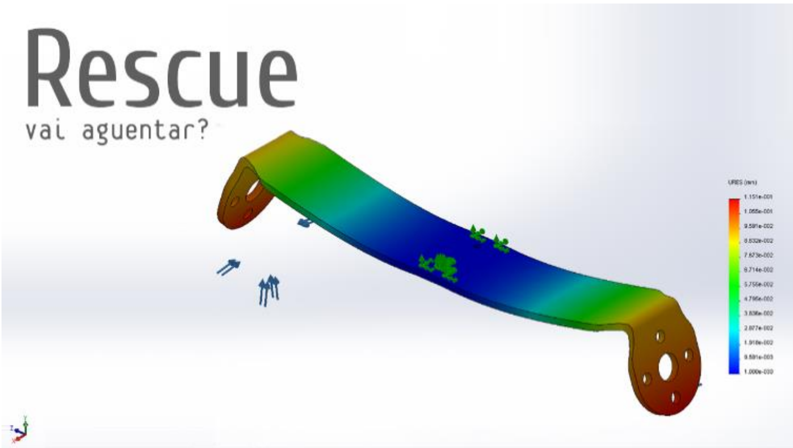

Fig. 3. Suspension bending simulation.

Figure 4 shows the robot electronics Printed Circuit Board (PCB) design. To process the encoder pulses, we have a MBED board [3], with ARM Cortex M3 processor, at 100MHz. This board is also responsible for controlling the motors speed. The communication protocol used to read the encoders from the interface, and to set the motors speeds, is the I^2C. The MBED board only has 3 encoder inputs. So, we had to program another encoder counter.

To make good measurements with the infrared distance sensors, we’ve made two supports that will accommodate them in the front and back of the robot, to get measurements in every direction. Each support will carry four sensors.

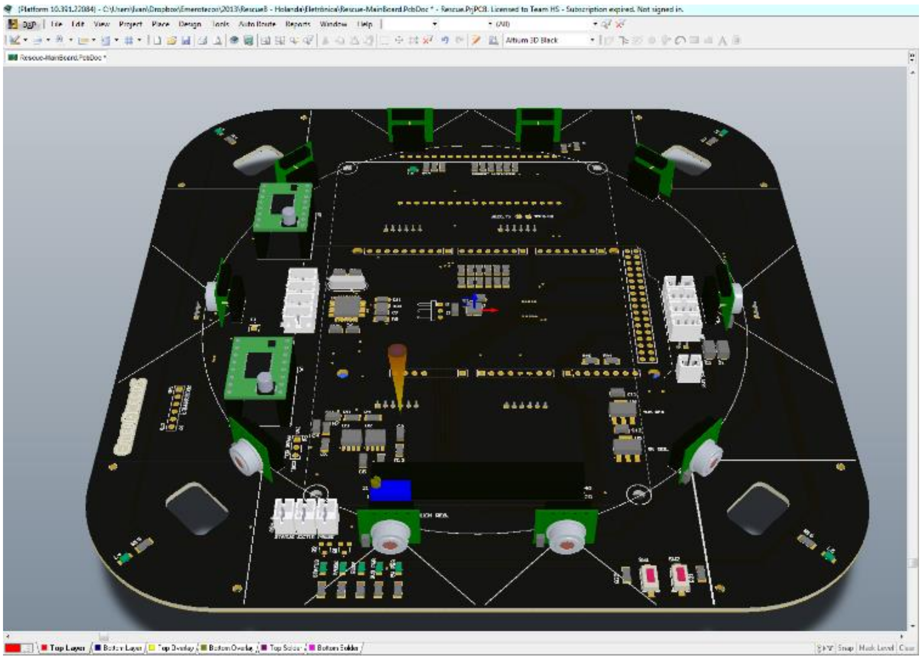

Fig. 4. Eletronics PCB design.

As we use complex algorithms (described latter), we need a very complex processing unit. For this reason we have decided to use an Arduino Due instead of the Android phone used on our last robot. We chose Arduino because its communication with the other controlling systems will be faster. Also, we have an independent LCD touchscreen and several sensors, like IMU, so we can replace the Android facilities. Our sensors give the robot a very precise way to know it's position and orientation, on all axles, so that we can execute our mapping device with precision and reliability.

B. Programming

The main controller of our robot is an Arduino Due, witch process the main code and receives information from the distance sensors so that it can send commands to the MBED, which controls the motors.

Last year, an Android phone were used to send and receive data to/from an IOIO board [4], which reads the sensors, and communicates with the MBED board, the one responsible for controlling the motors. We have noticed that this configuration was overloading the communication system. To overcome this problem we are suing an Arduino Due instead of the Android phone, and we don't need the IOIO board, which makes it faster to communicate with sensors and motors.

Our robot uses two types of infrared distance sensors: short distance sensors and long distance ones. Both types are mounted in front and in the back of the robot, pointing in the same direction. This was implemented to improve precision of the distance measurement in these directions (see Figure 6).

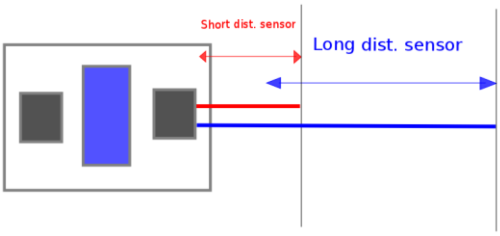

Fig. 6. Front/back sensors displacement

We also have heat sensors to detect victims all along the robot borders. The reason of choosing the front/back formation for the sensors supports is because that way the robot will know more accurately the end of any set of walls (figure 7), so they need to be as far as they can be from each other.

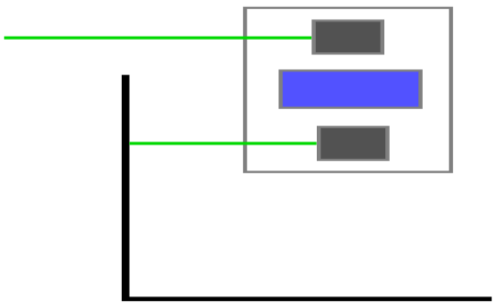

Fig. 7. Lateral sensors displacement

To move the robot, we use a simplified kind of SLAM (Simultaneous Localization And Mapping) algorithm, so that we can map the arena. We know that the walls are oriented in only 2 directions (0º and 90º), and we also know that each part of the maze is formed by a 30 x 30 cm square. With this knowledge and using the distance measures obtained via the infrared sensors, we can find out the walls positions and build a map of the maze. The SLAM is programmed in C++, and it runs on the Arduino DUE while the robot is moving in the arena.

We intend to change the algorithm to implement obstacle avoidance and to improve motion control. The idea is to make the robot move continually, with no stops. Last year´s robot succeeded to map, as shown in Figure 8, but on each block it had to stop to take distance measurements. We are trying to avoid this now. We also intend to attach more heat sensors. Those improvements will be completed by June 2013 for the RoboCupJunior 2013, in Holland.

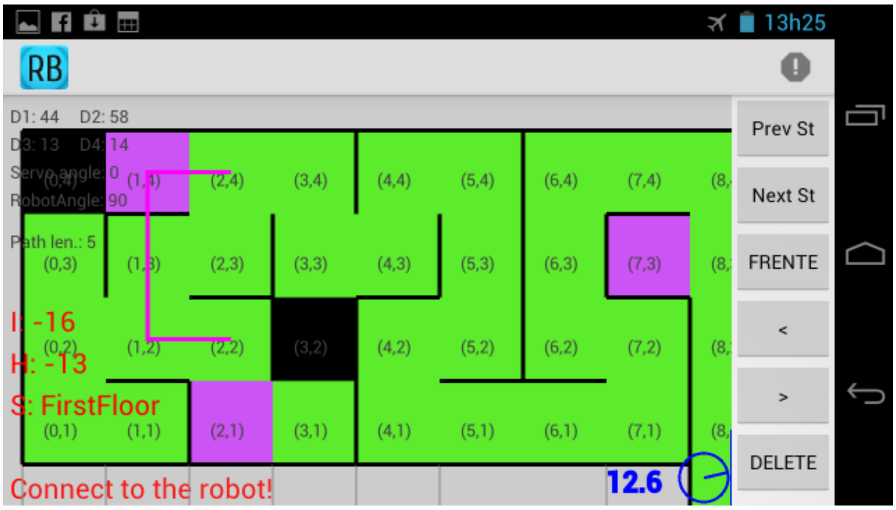

Fig. 8. On-line mapping from last year´s robot.

5. Conclusion

Mapping is the best way to solve this challenge, because it makes it easy to visit every module of the arena, and consequently, find all victims. We have successfully used this robot in the RoboCupJunior 2012 Rescue B competition. A presentation video about the robot is available in [5]. In RoboCupJunior 2013 Rescue B we intend to succeed in mapping the arena, but now with the SLAM algorithm!

6. Acknowledgement

The authors thank the Federal Institute from Espírito Santo (IFES) for all the support given in this project.

7. References

[1] 3D CAD Design Software SolidWorks. Available at: http://www.solidworks.com/. Access: Sep. 5, 2012.

[2] Arduino Due. Available at: http://arduino.cc/en/Main/ArduinoBoardDue . Access: Mar. 29, 2013.

[3] MBED - Rapid Prototyping for Microcontrollers. Available at: http://mbed.org/. Access: Sep. 5, 2012.

[4] IOIO Documentation. Available at: https://github.com/ytai/ioio/wiki. Access: Sep 5, 2012.

[5] NERA. Prêmio de Melhor Robô do mundo é de alunos do IFES! (com vídeo). Available at http://nera.sr.ifes.edu.br/?p=143. Access: Sep. 5, 2012.