

Development of a prototype robot for transportation within industrial environments

Tiago Ribeiro, Inês Garcia, Dylan Pereira, João Ribeiro, Gil Lopes, A.Fernando Ribeiro

Department of Industrial Electronics

University of Minho

Guimarães, Portugal

a71157@alunos.uminho.pt, a70557@alunos.uminho.pt, a71038@alunos.uminho.pt, a71698@alunos.uminho.pt,
gil@dei.uminho.pt, fernando@dei.uminho.pt

Abstract - This paper describes the design and development of an autonomous robot for the Robot@Factory league at “Festival Nacional de Robótica 2016”, held in Bragança, Portugal. This paper describes all the hardware and software components developed for a localization and performance of the robot according to the rules. The challenge consists of a table setup that recreates an industrial environment where a robot has to successfully transport boxes from an initial warehouse to the final warehouse. The destination to which the robot has to carry each box, depends on the state of the box, i.e., depending on the box LED color, even though in some cases the robot has to leave the box temporarily in the called processing machines (which are intermediate stations). The most significant innovation feature of this robot prototype consists of the possibility of carrying up to three boxes simultaneously while being able to select which box to drop. This project was developed with great success, since the team managed to reach the 3rd place in the competition.

Keywords—“Festival Nacional de Robótica 2016”, Robot@Factory, Factory, Robot Transportation, Industrial environment, MinhoTeam, Universidade do Minho

I. INTRODUCTION

This paper describes an autonomous robot developed to participate on the Robot@Factory competition of “Festival Nacional de Robótica 2016” in Bragança, Portugal (Fig. 1). This robot was designed and developed by students from the Integrated Masters in Industrial Electronic Engineering and Computers, of the University of Minho, being an extracurricular work developed in the Laboratory of Automation and Robotics.

The competition consists of recreating a factory environment, where a robot has to overcome challenges in this context. The factory is made up of a supply warehouse, a final product warehouse and eight processing machines. The objective of this competition is to transport boxes between initial warehouse and final warehouse in the shortest possible time, and the boxes must be collected, transported and delivered correctly. The boxes might need to be temporarily placed in the intermediate processing machines.

For better performance, the robot needs to localize itself correctly within the factory environment, and in addition it must be able to navigate avoiding collisions with obstacles.

The University of Minho has been participating in this league since the start of this competition in 2011. However, the robot developed in this work brings new features from previous participating teams, not just from University of Minho but also from other institutions. One of the main characteristics important to point out is the ability to carry more than one box (more precisely three boxes) with the advantage that all the boxes are ready to be unloaded at any time, making the whole process faster and more efficient.

This paper focus all different robot development systems. This work is divided in 5 main sections, namely introduction, methodology, results, discussion and conclusions.

A. Robot@Factory description

The competition consists of an 3.5x2.5 meter environment that resembles a factory. The robot is expected to move around the track avoiding obstacles and transporting boxes, which simulate goods from a factory, from initial to final warehouses [5].

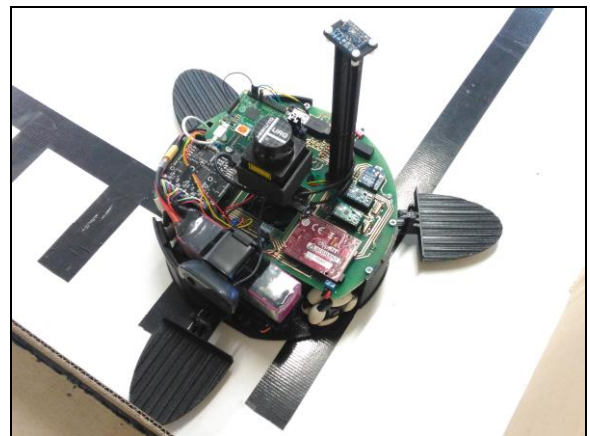


Figure 1 – Robot developed with three shovels

Each box has a destination according to the box LED colour, where it can be the final product warehouse or the intermediate processing machines. In order to place the box in the correct slot, the robot must be able to identify the box state and to locate both the warehouses /processing machines. Should the box require to pass through a processing machine, a successful placement will be considered by the jury, only when the box is placed on the final product warehouse.

In order to simplify robot localization, the factory environment contains lines that indicate possible routes for each warehouse and each processing machine.

B. State of the Art

To date, many teams have participated on the Robot@Factory competition with different methodologies and ideas. Many teams remain in the concept of carrying just one box at a time [10]. However, some have already advanced to the method of transporting more than one box simultaneously. Also, some teams use robotic simulation with a program called SimTwo that was adapted specifically for the competition [10].

II. METHODOLOGY

A. Hardware

1) Mechanics

The robot consists of an omnidirectional platform with three motors [3], each with an omnidirectional wheel, offset by 120 degrees between each other (Fig. 1). The box transporting mechanism uses three shovels driven by servomotors developed by the team. They are also offset by 120 degrees (Fig. 2).

The robot uses a LIDAR physically placed so that it does not detect the warehouses and the machines. Therefore, a mechanism has been developed which allows raising and lowering the sensor easily.

It should be noted that the entire robot structure and shovels were designed and printed in a 3D printer in PLA material, which makes the robot lightweight and easy to adapt to different circumstances.

The robot also contains an IMU which is placed well above the main PCB, in order to avoid magnetic influences from the motors, electronics circuits and eventual floor wiring. The IMU standing support was placed in the blind area of the LIDAR so that it would not interfere with the final readings.

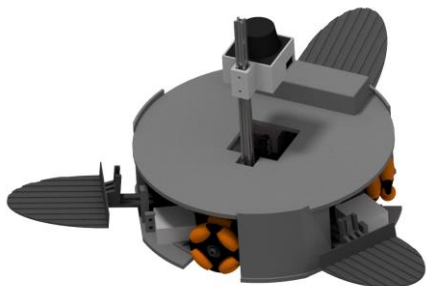


Figure 2 - 3D Modeling of the robot

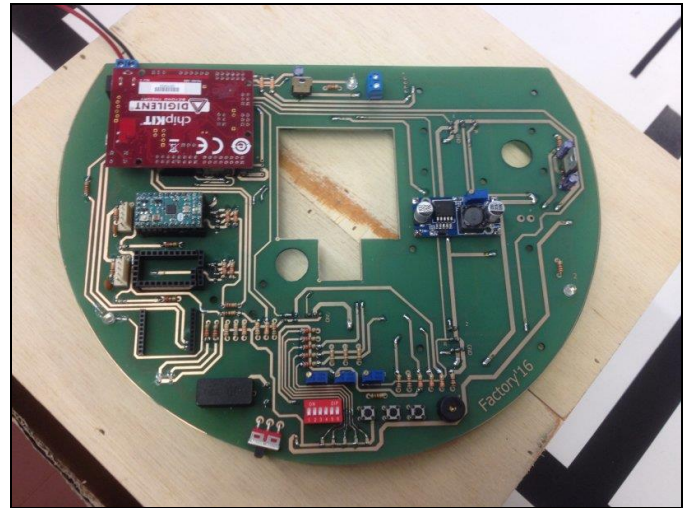


Figure 3 – Robot Main PCB developed

2) Electronics

At the electronic level a PCB board was designed and built (Fig. 3), with the software CadSoft Eagle [2], which contains all the electronic circuits necessary for the robot. This board contains several processors, namely: two Arduino Mini [6], one Raspberry Pi 2 Model B [8], one chipKIT Uno32 [7], one OMINI 3MD [9]. The PCB contains also the sensors and actuators necessary for the competition.

Regarding the sensory system, the robot contains one IMU (Adafruit 10-DOF IMU BREAKOUT - L3GD20H + LSM303 + BMP180), one LIDAR (HOKUYO URG-04IX-UG01), one array of infrared sensors (developed by the team specifically for this purpose), one infrared sensors on each shovel (to detect the presence of a box), and one USB camera (Logitech HD Webcam C270), one XBee [4] used to communicate with the API (Application Programming Interface) mostly for debugging. Debugging can also be done without the API feature through the use of several LEDs and a buzzer. Several buttons and dip-switches are also included in the circuit for easier parameter configuration.

Figure 4 describes the complete schematics of the robot hardware.

Due to some sensor connections restrictions, different methodologies for communicating between boards had to be implemented. The initial idea was to connect all the processors to the chipKIT Uno32 via I2C. This processor is the Master and sends all data packets to all other processors, and if necessary they would respond with their respective data packet. However, both Arduino Mini already have a sensor that requires an I2C connection, meaning they must be Masters to the sensors they are connected to, making communication via I2C impossible between these two processors and the central processor. As an alternative solution, the SPI data transmission method was used between the two Arduino Mini and the central processor.

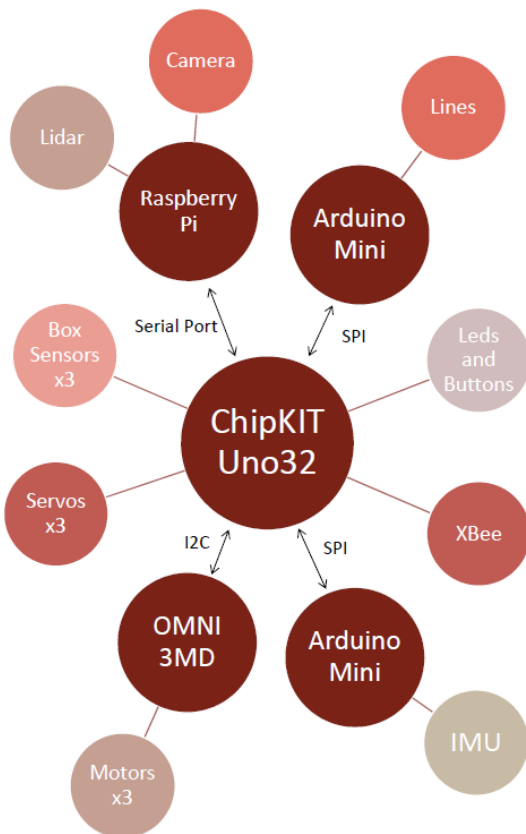


Figure 4 – Robot Schematics

The OMNI 3MD board transmits data via I2C to the central processor. As for Raspberry Pi 2 Model B, it uses the serial port protocol for the sake of data transmission ease. The XBee transmitting data to the API is connected to the central processor by serial port also.

B. Software

For modularity purposes this robot uses 5 processors, as already mentioned in the previous section, which allow faster computing, ease of replacing modules individually (should a break down occur) and in the eventual case of a new module being developed. This method also allows the robot to perform several tasks in parallel. The CPU assigns tasks to itself and other processors achieving better system management and better performance of all processors.

1) Software on each module

- A Raspberry Pi 2 Model B, with "Ubuntu Mate" operating system manages the camera and the LIDAR. The LIDAR is used for robot localization (further down in this paper the localization process will be described in more detail). The camera is used to scan the Box LED colours.
- One Arduino Mini to control the IMU. Its main task is to compute the robot heading direction, which is one of the parameters for the localization

(the localization is described further down). A Kalman filter is applied to increase direction precision, and it combines the values of the three sensors: accelerometer, gyroscope and magnetometer. The implementation of the Kalman filter causes the direction values to have a considerably lower error than compared to the results without the Kalman filter.

- One OMNI-3MD - Motor Driver responsible for the commands sent to the motors and velocity control (this includes acceleration ramp). This board also reads the encoders values (coupled on each motor) as well as the battery voltage.
- One Arduino mini to control a 7x7 array of infrared emitter/receiver pairs. Each infrared emitter/receiver pair has a capacitor in series with the receiver's phototransistor, which makes possible to acquire an analogue value from the time the capacitor takes to discharge. This time is calculated from the current the photo-transistor requires. For different colours, different capacitor discharge time values are obtained (due to emissivity coefficients), thus allowing an easy detection between black and white, which are the two colours present on the factory floor. The objective of this array of sensors consists of identifying the different line situations, for example, a vertical line, horizontal line, a corner, etc. (Fig. 5). To develop this algorithm, the robot direction was taken into account, since the robot does not move always in the same direction, but it moves in the direction that best suits for better performance.
- One chipKIT Uno32 runs the software to control several devices: *debug LED's*, dipswitch for different robot configurations, buttons used to configure parameters, one buzzer, 3 sensors of the shovels and 3 LEDs (to indicate box detected), 3 servomotors responsible for lifting the shovels, one XBee to communicate with a remote computer. This processor is also responsible for instructing the other processors to return the data read.

2) Box picking up system

When the robot is required to pick up a box, it performs a set of tasks. First, the robot slowly moves towards the box with the selected shovel down. When the shovel sensor detects the box presence the Box Detected LED is activated and the shovel starts to raise. Afterwards, the robot moves slowly backwards until it reaches the white line on the floor, knowing it is safe to turn, and continues its way.

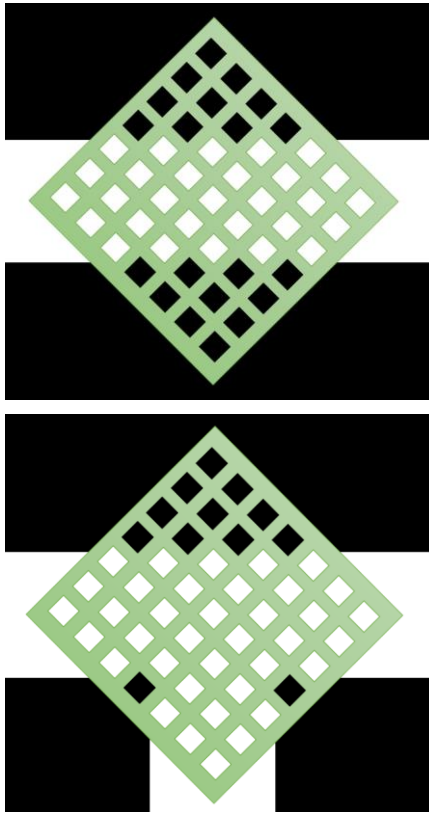


Figure 5 – Two examples of Line situations where the matrix of 7x7 IR sensors detects black and white differences

3) Localization

The robot needs to know with precision where it is on the factory environment (using its global localization system) in order to navigate avoiding obstacles and to collect and place the boxes in the expected slots. It is important to point out that the slots are barely larger than the boxes, leaving little scope for inaccuracies.

For the robot localization, a bi-dimensional Cartesian coordinate system is used, with the centimetre as its unit. The localization algorithm developed takes into account four different sensors:

- The IMU is used to give the robot direction.
- The encoders give the robot distance travelled since the last iteration, but it adds up accumulated errors due to rounding and eventual wheels drift, making encoders a good solution only for short term.
- The array of bi-dimensional infrared sensors can updates one or both of the Cartesian coordinates, depending on the perceiving line situation. This depends on the actual robot position and similar line situations described on the competition map.
- The LIDAR outputs distance to obstacles within a range of 270° with steps of 0,36°. The priority when developing the localization software with

the LIDAR was given the lowest computational time to achieved the highest frequency response (Fig. 6 and 7).

Keeping this in mind a simple algorithm was developed consisting of having only the distance to one of the walls in the X axis and the Y axis. Taking into account the actual robot direction, it is chosen the positions from the LIDAR vector which represent the distance to each axis. Considering the robot direction it is possible to know if the chosen angle is pointing towards the wall in the positive direction of the chosen axis or in the negative direction of the chosen axis (this works for both axis). The computing time is low, however, should the distance of the desired angle be wrong, the returned value would also be wrong, but since there is no dynamic obstacle other than the robot on the field, these kind of errors will not happen.

With the sensorial fusion of these 4 sensors it was possible to develop a precise localization algorithm. Since the encoders on the wheels are reliable only for short term (because they accumulate errors for long term), for long term the system uses the lines situation and the LIDAR.

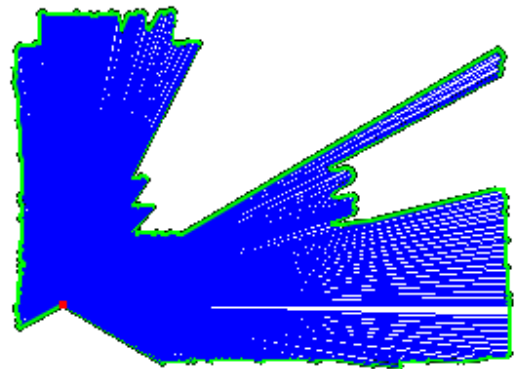


Figure 6 - LIDAR output with wrong height adjustment, where processing machines show up

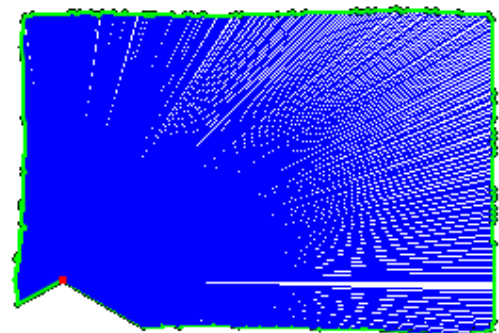


Figure 7 - LIDAR output with proper height adjustment

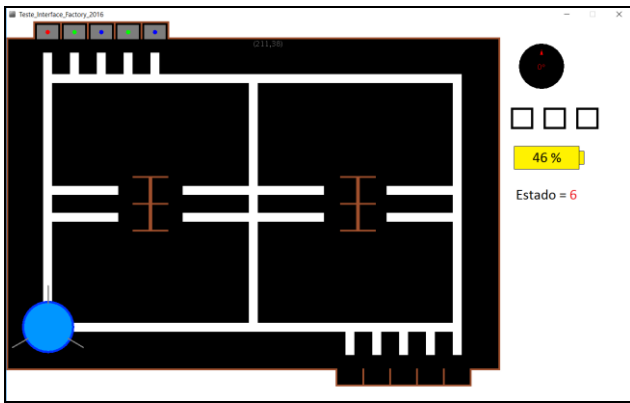


Figure 8 – Graphical Environment API Developed

4) State machine

The robot main algorithm is based on a finite state machine. The idea is quite simple and contains 4 commands:

- Go to the X,Y coordinates
- Rotate W degrees
- Pick up box
- Drop box

With these 4 commands, it is possible to complete all the required tasks. The success of this algorithm depends significantly on the robot localization precision.

5) Colour tracking

The boxes colour screening is carried out at the beginning of the each leg. The robot starts by advancing to the correct distance to see the first box colour in the predefined conditions (close to the box), and after recording the colour it moves laterally parallel to the wall where all the boxes are placed, perceiving and saving them all in memory. After storing the last LED colour, the finite state machine is set up for minimum time performance and the decision is taken about which boxes should go into which shovel and slot. Then, the robot enters the first state of the finite state machine and begins to run the whole code.

6) Processing developed API

An API (Fig. 8) has been developed in PROCESSING [1] to see in real time all the robot sensors/actuators values. A graphical environment was created to picture the competition field. It contains not only the white lines on the black background but also the boxes in their positions, with their respective LED colours. The robot was also drawn and, in real time, its current position is sent to the API, so that it is possible to see the location given by the localization algorithm. This tool was of extreme importance for a better understanding of what the localization algorithm does, as well as a better development of the robot software.

Besides the localization, the robot also sends to the API its direction, the boxes it is transporting, the LEDs colours, the buttons states, the battery voltage, the robot speed, the current state within the finite state machine algorithm, the lines situation, and the distance of the angles chosen by the

algorithm that deals with the LIDAR. These parameters are drawn in the real-time graphical environment, and only then can developers be aware of all robot inputs/outputs and the way the software manages them.

III. RESULTS

A noticeable improvement is observed when comparing the performance of the localization before the sensorial fusion and after it. The robot always moves to the desired coordinates avoiding obstacles correctly.

Regarding moving and transporting boxes, the robot moves in a very smooth and controlled manner. However, some alternative measures had to be taken regarding the boxes pick up order due to the robot mechanical and structural characteristics, such as the centre of mass offset from the robot centre, the short distance from the robot's bottom to the floor, and the raw material used for the base platform (PLA) being too flexible.

These measures were only taken for a better performance, but they would not compromise the good development of the robot in the competition.

The whole process of picking up and dropping the three boxes was performed quite successfully (Fig. 9). The robot could always lift and drop the boxes without ever losing any. As for its transport, the robot has always been able to keep the box on top of the shovel regardless of the direction in which it had to navigate, the angular velocity, the linear velocity, and the peaks of acceleration and brakes that the state machine instructed. All these parameters were tested for values well above those used in the competition to ensure the whole process was reliable.

IV. DISCUSSION

The localization is a key issue in this project, because the robot needs to know precisely where it is to avoid hitting obstacles, and when placing down the boxes the robot must be perfectly aligned with the slot. In the robot localization the direction is one of the parameters that can be improved, and to achieve that the influence of magnetic fields has to be filtered. Should the robot calculates a wrong direction (due to eventual interference), the distance value output from the LIDAR will no longer be the real distance.

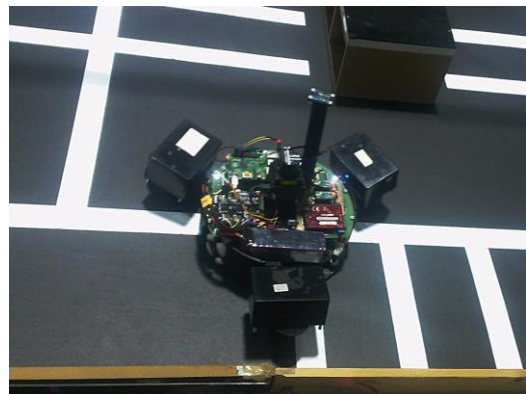


Figure 9 – Real robot transporting three boxes

The value of the LIDAR is of extreme importance for the compensation of the encoder error. The LIDAR algorithm has space for improvement. After some study it was found that, the low computational time is not such a big advantage when compared to an accurate output. The encoders as well as the line situations are solutions that worked very well and helped to compensate for errors of the other two solutions.

The process of selecting the shovel to pick up each box had to be changed due to some robot structural and mechanical characteristics. When boxes are picked up, the robot centre of mass changes, and its change is very different according to the number of boxes the robot carries and which shovel carries the boxes.

The process of lifting/lowering boxes reliably is also extremely important since it is the competition central requirement. This system was tested with various types of shovels (built up in a 3D printer) as well as for various lifting processes (springs, coils, different servomotors). The method here described proved to be the most efficient of all, being able to perform this task with utmost reliability. The new concept introduces a transporting robot with three shovels in different directions and it proved to be extremely effective, improving the time performance in only two trips to transport the five boxes to the final warehouse. With the three shovels, it was also achieved better management about which boxes to take first and to what place, since in some trials the boxes would have to stay 30 seconds inside processing machines until they were again moved to their next destination.

V. CONCLUSION

The development of a completely autonomous and independent robot is of great interest in the world of Control, Automation and Robotics. This project has become an even more compelling challenge given that it is applied to a simulated manufacturing environment, which enables the development and future application in real manufacturing.

As mentioned in the Discussion section, there are many components in the robot that have been very successful, but there are still some others that require improvements to make the robot behaviour more reliable and cohesive. Among them it is the localization algorithm, since in this kind of robots which operate and transport objects from point A to point B, it is essential that they can be located accurately in any environment they are in.

One of the aspects that can be ascertained is the robot structure and organization, since some characteristics of the robot, only perceptible after its construction, have brought unwanted responses, such as the height of the base relative to the ground.

Regarding the robot interface developed, it is important to mention that it proved to be an added value during the development of the prototype, since it gave the opportunity to analyze in real time the system behaviour.

It should be noted that the balance of this project is extremely positive. The University of Minho team managed to win the 3rd place in the Robot@Factory competition at the National Robotics Festival of 2016 in Bragança, Portugal.

VI. REFERENCES

- [1] Processing, [Online]. Available: <https://processing.org/>.
- [2] CadSoft, "Eagles PCB Design," [Online]. Available: <https://cadsoft.io/>.
- [3] "Pololu - Metal Gearmotor 37Dx57L mm," [Online]. Available: <https://www.pololu.com/product/1446>.
- [4] Digi International Inc., "XBee®/XBee-PRO® RF Modules," Minnetonka, 2009.
- [5] J. Lima, P. Costa e A. P. Moreira, "Robot@Factory - Competition rules and technical specifications," 8 02 2016. [Online]. Available: <http://robotica2016.ipb.pt/docs/regras-robot-factory.pdf>. [Acedido em 29 01 2017].
- [6] Arduino, "Arduino Mini," [Online]. Available: <https://www.arduino.cc/en/Main/arduinoBoardMini>. [Acedido em 27 01 2017].
- [7] ChipKIT, "ChipKIT Uno32," [Online]. Available: <http://chipkit.net/wpcproduct/chipkit-uno32/>. [Acedido em 30 01 2017].
- [8] Raspberry Pi Foundation, "Raspberry Pi 2 Model B," [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>. [Acedido em 29 01 2017].
- [9] Bot'n Roll, "Bot'n Roll OMNI-3MD," [Online]. Available: <http://botnroll.com/omni3md/>. [Acedido em 30 01 2017].
- [10] J. Gonçalves, J. Lima, P. Costa e A. Moreira, "Manufacturing education and training resorting to a new mobile robot competition," em *Conference on Flexible Automation and Intelligent Manufacturing FAIM 2012*, Ferry Cruise Conference Helsinki-Stockholm-Helsinki, 2012.