International Journal of Technology and Human Interaction
Volume 13 • Issue 4 • October-December 2017

Simulator for Teaching Robotics, ROS and Autonomous Driving in a Competitive Mindset

Valter Costa, Faculty of Engineering University of Porto, Porto, Portugal
Rosaldo Rossetti, LIACC-DEI/FEUP, University of Porto, Porto, Portugal
Armando Sousa, INESC TEC and Faculty of Engineering University of Porto, Porto Portugal

ABSTRACT

Interest in robotics field as a teaching tool to promote the STEM areas has grown in the past years. The search for solutions to promote robotics is a major challenge and the use of real robots always increases costs. An alternative is the use of a simulator. The construction of a simulator related with the Portuguese Autonomous Driving Competition using Gazebo as 3D simulator and ROS as a middleware connection to promote, attract, and enthusiasm university students to the mobile robotics challenges is presented. It is intended to take advantage of a competitive mindset to overcome some obstacles that appear to students when designing a real system. The proposed simulator focus on the autonomous driving competition task, such as semaphore recognition, localization, and motion control. An evaluation of the simulator is also performed, leading to an absolute error of 5.11% and a relative error of 2.76% on best case scenarios relating to the odometry tests, an accuracy of 99.37% regarding to the semaphore recognition tests, and an average error of 1.8 pixels for the FOV tests performed.

KEYWORDS

Autonomous Driving Competition, Educational Robotics, Gazebo, Hardware in-the-loop Simulation, Mobile Robotics, ROS – Robotics Operating System, STEM – Science Technology Engineering Mathematics

INTRODUCTION

Interest in autonomous vehicles (AVs) have grown in the past years and represent an important innovation for the automotive industry. Self-driving cars pave the way for myriad relevant applications across multiple fields. In such a dynamic context, work and research in this area has grown greatly over the last years (Carvalho & Borrelli, 2015). This topic has even raised interest within the robotics community and was the target focus of many robotic competitions around the world. An autonomous intelligent vehicle has to perform a number of tasks, sometimes in a limited amount of time. The most critical task is the perception and mapping of the surrounding environment. This involves being capable of identifying and tracking road lanes, being able to process traffic lights and road signs, and being consistent at identifying and avoiding obstacles (Häne, Sattler, & Pollefeys, 2015). The interest in robotics field as a teaching tool to promote the STEM areas - Science, Technology, Engineering and Mathematics has grown in the past years. Some initiatives like RoboCup contest (RoboCup, 2016) that have its first edition 1993, RoboParty (RoboParty, 2016) going in the 10th edition in 2016, the CEABOT (CEABOT, 2016) since 2006, ISTROBOT (ISTROBOT 2016, 2016) since 2000, "Micro-Rato" (Micro-Rato, 2015) since 1995, Micromouse Portuguese Contest (Micromouse, 2016) since

DOI: 10.4018/IJTHI.2017100102

2011 focus on promote robotics challenges. In 2015, the car brand Audi has proposed a challenge called Audi Autonomous Driving Cup (Audi Autonomous Driving Cup, 2016) aiming to promote the autonomous driving challenges. This competition is orientated to students of computer science, electrical engineering, mechanical engineering or similar disciplines. The participants are asked to develop fully automated driving functions and the necessary software architectures. For this task, a hardware platform model vehicle, with a scale of 1:8, is provided. This vehicle was developed specially for the competition. Giving the last year success, Audi has repeated the competition in 2016, offering a cash prize of 10,000 euros. As such, the autonomous driving paradigm is well-suited to incentive robotics students in those competitions, such as the Portuguese National Robotics Festival (PNRF) - "Festival Nacional de Robótica."

LITERATURE REVIEW

In order to properly prepare the contestants for a successful participation, some authors have presented strategies to teach and enthusiasm students into the robotics area using hardware based platforms. A major point to support this solution is the integration of Robotics classes' exercises with real-world problems, with the output of the working project interacting directly with a real setup, thus motivating students (Cardeira & da Costa, 2005; Lenskiy, Junho, Dongyun, & Junsu, 2014). To reduce complexity, they can be based on off-the-shelf components. By skipping the complications of hardware development, which students in their first graduate years do not have, development in autonomous driving robotics is much more accessible (Cardeira & da Costa, 2005). They can also permit online development, making possible to create online classes, giving greater accessibility to the platform (Lenskiy et al., 2014). Usage of this manner of platform as a learning tool can prove to be costly and potentially difficult to distribute between potential students. Being a physical solution, it is prone to malfunctions, thus requiring maintenance and it is harder to share, unlike software. Unless it is a modular setup, the robots will be tailored to specific tasks, limiting their usage in different scenarios (Yusof & Hassan, 2012). Another disadvantage of using real platforms is the associated monetary cost, being sometimes inaccessible to fund. A solution to this problem is the creation of almost inexpensive platforms to introduce the same robotic tasks: a possible substitution is the use of 3D printed models to replace parts such as the wheels or the chassis (Gonçalves, Silva, Costa, & Sousa, 2015); the printed circuit board that hosts the electronic components can be designed to also serve as the robots' structure (Valente, Salgado, & Boaventura-Cunha, 2014). Although these solutions are much less expensive, they always have an associated monetary cost. A workaround is the use of a simulator or a hybrid solution – simulator and a real platform – thus minimizing aforementioned costs.

In robotics, a simulator plays a very important role, aiding the development of prototypes. By emulating reality within a certain degree of rigor, it allows inexpensive and less consuming tests to developed architectures. Using this strategy, countless number of tests in various well-defined environments can be done with no fear of causing real damage to the system under test: for example, one can test developed algorithms in the simulator before implementing in the physical robot, thus guaranteeing a certain degree of dependability in those algorithms (Kuc, Jackson, & Kuc, 2004). It is also possible to repeat experiments with the same initial conditions and parameters, or change them in the simulator, helping in understanding the impact of each in the result (Abiyev, Ibrahim, & Erin, 2010). Simulation results can be improved by adding a mathematical representation of all related dynamic systems or by using samples or datasets from components at runtime. This approach is normally called "hardware-in-the-loop" simulation (Jong, Park, & Kim, 2012; Lee, 2008; Temeltas, Gokasan, Bogosyan, & Kilic, 2002), and is normally employed to develop complex embedded systems, greatly shortening the research and development process of control systems. With the implementation of sensors and actuators in a simulation, some processes can be simplified, also making them more close to reality (Lee, 2008). It should be noted that this solution, when compared to hardware based ones, brings up disadvantages, such as the insertion of errors which are hard to estimate or the high amount of necessary computational resources to simulate complex systems. All these can be avoided

by choosing a good compromise between the complexity of the model and the accuracy of the physics associated with the real process. "Hardware-in-the-loop" simulation can also help in the integration of students on Robotics: by using real hardware, students can see their work in a system close to reality, thus motivating them (Jong et al., 2012).

RESEARCH ISSUES

From the literature review and the presented context, the following issues were identified:

- It is quite hard to attract students to the STEM area
- Even if robotic is visually appealing, it is very hard and expensive to tackle with the complexities and running costs of real world robotics applications
- The Portuguese National Robotics Festival is a well-known, long running event that aims for attractiveness taking advantage of the natural competitive mindset of Higher Education students but also tries to attract general public to technological events
- The PNRF - Autonomous Driving Robotic Competition (PNRF-ADRC) has been decreasing in number of competitors (most certainly due to economic pressure and the complexities of the challenge)
- The PNRF-ADRC has a close resemblance to real world autonomous driving cars that are now and will be in the near future a big issue in the generalist communication mass media and this proximity is expected to bring even more attractiveness to non-expert audiences and again attract students
- The mentioned competition is a real world competition that occupies large amounts of space for the track (roughly 7m by 17m), needs road signals, monitors as part of the setup and many other resources hard to keep allocated to such competition; the typical robot for the competition needs a laptop for image processing coming from several sources (track, semaphores, traffic signals, etc.)

Next it is presented the proposal and research goals in order to address the supra-mentioned questions.

PROPOSAL

The work presented focus on the construction of an autonomous driving simulator, using the Portuguese autonomous driving competition (XVI Portuguese robotics open, 2016) as the simulator's world model. This is a work in progress (Costa, Rossetti, & Sousa, 2016) and some improvements to the previous simulator were already made. It is intended to use this simulator on a fifth-year course of Integrated Master in Informatics and Computation Engineering in Faculty of Engineering of University of Porto (FEUP) called "Intelligent robotics". The developed simulator introduces the motion control challenges of a mobile robot and/or the image processing challenges applied to the same robot. One of the major goals is the creation of a micro competition, challenging the students to form teams (two or three students per team) and compete with each other. It was demonstrated by previous work, (Costa, Cunha, Oliveira, Sobreira, & Sousa, 2015) that the use of a competitive mindset provides enthusiasm to students and propels them to overcome complex issues that may appear on autonomous driving competitions. The main advantage of using a simulator is portability. Making arrangements to prepare tracks for real robots is an expensive and time consuming task. Another issue is the space needed to build and maintain the track. The simulator development includes the design of a track, Figure 1, the competition semaphores, Figure 2 and robot model, Figure 3b. One important aspect to have into account is, in the context of the Portuguese autonomous driving competition, a semaphore is a sign projected on two monitors and not the common three color sign. An evaluation of the simulator performance is also described.

Research Goals

The goals of the presented research are:

- To keep the proximity to the Portuguese National Robotics Festival – Autonomous Driving Robotic Competition
- Try to attract new participants
- To make development easier on the existing platform
- To have a versatile but realistic test platform for all parts of the competition (track and robot), preferably including synthetized perception, physics, etc.
- To take advantage of a healthy, extendible architecture by promoting code reuse maybe from an initial starting point (allowing developers to target limited problems)

Figure 1. Track used on the 2016 Portuguese autonomous driving competition

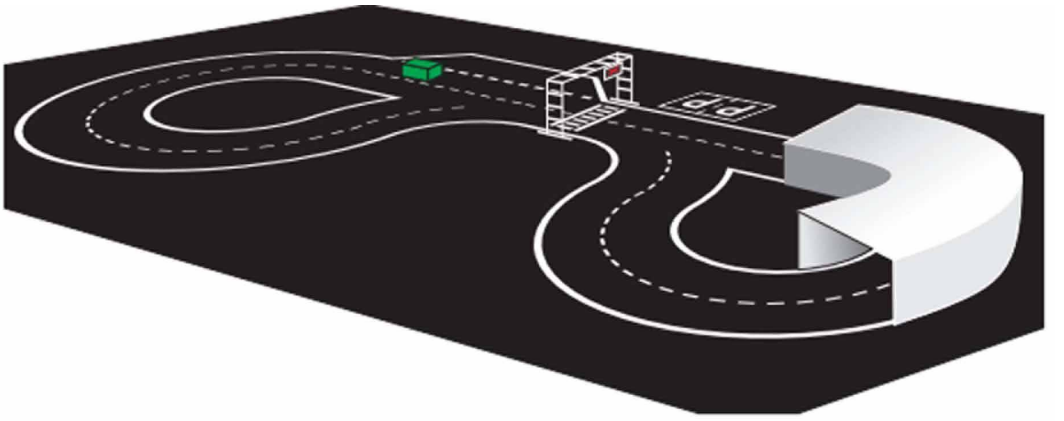
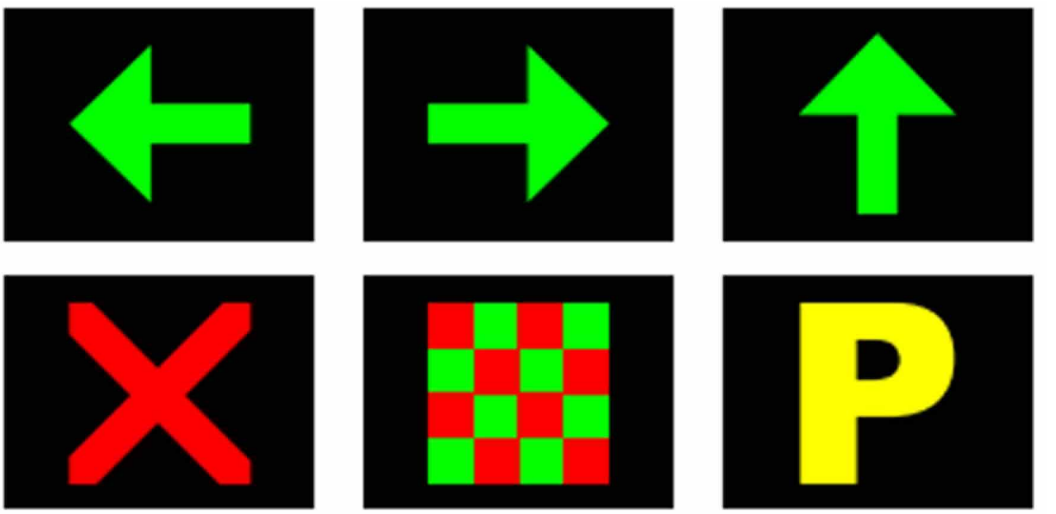


Figure 2. Semaphores used on the 2016 Portuguese autonomous driving competition. Left to right, indication to turn left on green, indication to turn right on green, indication to move forward on green, indication to stop on red, end of trial on Red and green checkers flag colors are presented in an alternate succession and indication to follow to the parking area



- To allow future multi-robot development
- To reduce running costs for the setup and the robot
- To reduce the gap from simulation to reality
- To ease the study of new developments for the track and the platform and rule set improvement
- To try to make improvements in the scientific area of Mobile Robotics and Autonomous Driving.

To accomplish these goals a 3D simulator respecting the Portuguese Autonomous Driving Competition was designed and evaluated.

The remainder of this paper is organized as follows: in Section II all the steps and methods followed in the designing of simulator are laid out; Section III displays the results of the evaluation made to the simulator; Section IV draws conclusions from the obtained data; finally, Section V presents proposals to improve it in future work.

METHODOLOGIES

All the tasks ranging from the development of the robot model till the construction of the simulation world representing the autonomous driving competition are presented in this section. The simulation was based on a 3D model representing a real robot, Figure 3a and Figure 3b, and the world where the robot will move on.

The developments presented follow a commonly well accepted trail of designing a 3D simulator for the active track setup and the robot using a common use 3D simulator. Recent robotics most frequently take advantage of ROS as a robotic middleware for standard communications (for example inside a robot) and thus naturally promote healthy architecture, separation of issues and code reuse. Beyond standard communications, ROS also allows some degree of hardware abstraction and that in turn, makes indifferent hardware or simulation (or even partial hardware, example hardware in the loop).

The Gazebo platform is the only totally free, open source, generalist 3D simulator with physics engine in the ROS normal distribution (indigo version). This software is thus the natural solution, still leaving room for future developments. The drawback of this platform (and many other 3D simulators) is that they require heavy 3D workload.

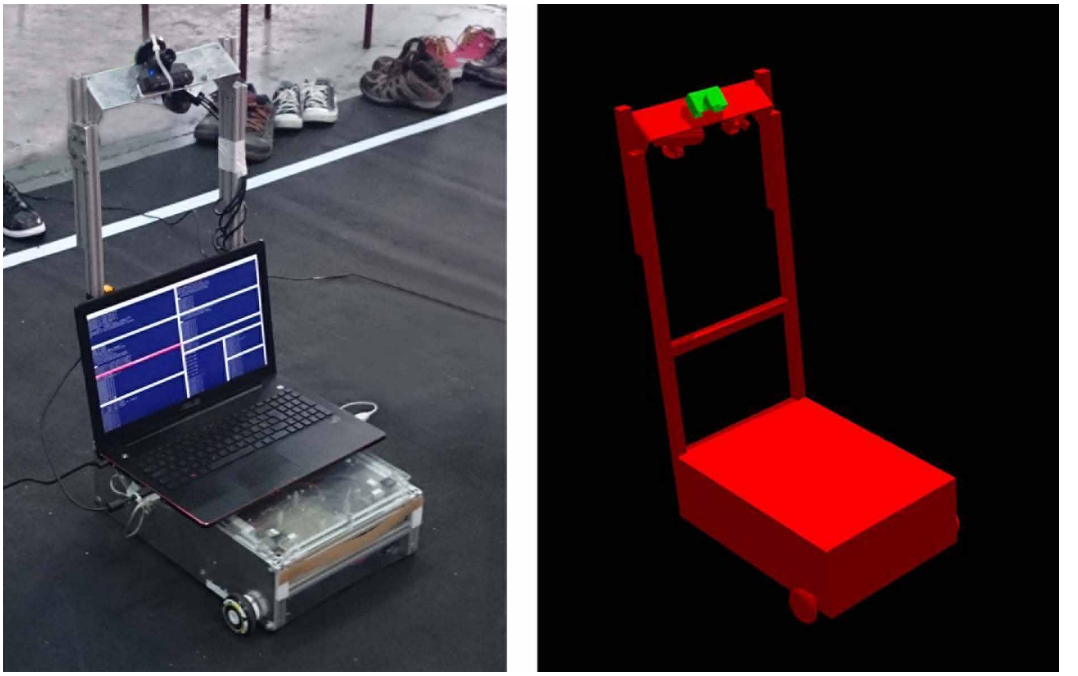
Designing the Robot Model

The robot model was improved and all the parts (wheels, chassis, cameras, castor-wheel) were designed using a CAD software (Computer-Aided Design). Then, it was exported to .stl files that are imported to the Gazebo simulator using a .sdf file. SDF is a XML format that describes objects and environments for robot simulators, visualization, and control. This type of file is particular to Gazebo and cannot be used in other simulator. The advantage of designing all the robot parts using a 3D modelling software is the possible portability; these parts can be used in other simulation software. For instance, these parts can be exported to an .urdf file, Unified Robot Description Format, which is an XML format for representing a robot models.

This robot uses a differential steering locomotion system, in which its movement changes by varying the relative rate of rotation of its wheels. A castor wheel placed at the end of the robot is needed in order to balance the structure. Another improvement on this work was the modelling of a castor wheel, instead of using a sphere to balance the robot. Although it is a valid simplification, it doesn't represent some of the traits present in a real caster wheel, such as errors derived from abrupt direction changes from the movement of the robot.

After designing the parts, the next task is the linkage between them. This step was accomplished using the .sdf file. The second step was to connect the wheels and motors and link them into the chassis, in which a plugin called libgazebo_ros_diff_drive was used and the connection to ROS was made using a library called libgazebo_ros_diff_drive.so. This linkage is very important, allowing the control of the motion of the robot by using a standard message type named geometry_msgs/

Figure 3. Autonomous Driving Robot - Picture on the left real robot "Conde", on the right the robot model



Twist from ROS, thus publishing into the velocity topic given from Gazebo. The next step was the integration of the cameras. As referred before, the robot runs on a track with lanes (that require to be tracked) and needs to identify semaphores. In order to complete both, three cameras are installed on the robot. The first two are pointed to the ground to detect the track and the last one is pointed up, to detect the semaphores. The reason to use two cameras pointed down instead of one is due the fact that using only one is not possible to see the entirety of the track. These cameras are added to the robot model by using sensor tag of type camera. Once again, to connect the camera to ROS a plugin is needed. The plugin used was libgazebo_ros_camera. Using this plugin it is possible to define camera frame rate, resolution, intrinsic camera parameters, and the topic in which the camera will publish the captured image.

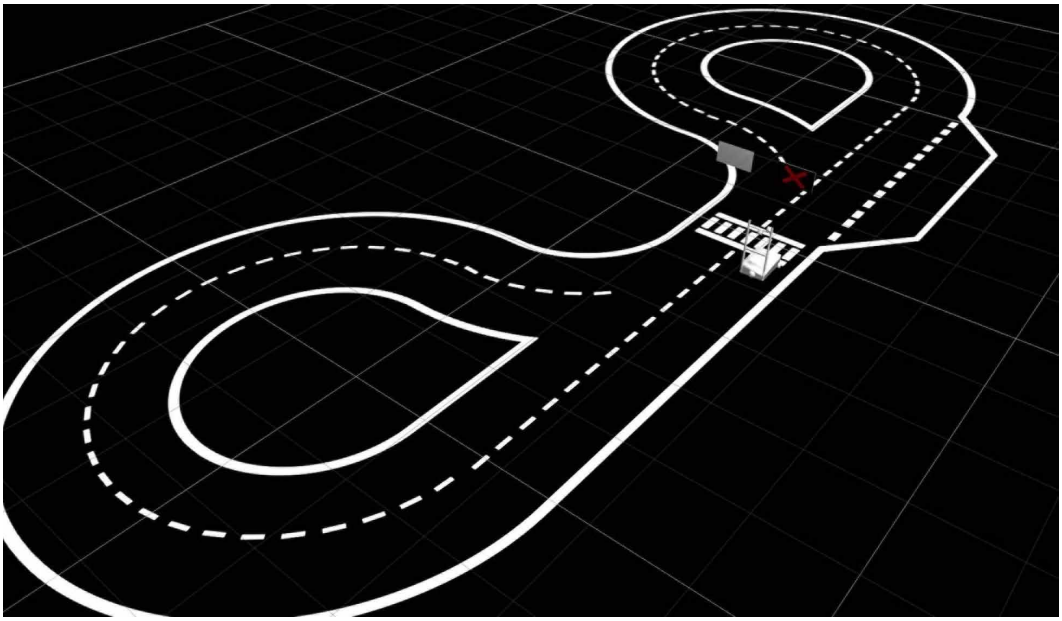
Designing the Robot World

After designing the robot model, the next stage is building the world where the robot is going to move. The world created is a representation of the 2015 Portuguese Autonomous Driving Competition track held in Vila Real. The track was made by importing a .png image, also adding two monitors to display the semaphores. These semaphores use another plugin from ROS called libgazebo_ros_video and an additional node was made to control the displays on the simulator. The competition world built is presented in Figure 4.

The track shown in Figure 4 was built in flat ground, with well-defined lighting conditions. However, with the Gazebo simulator, it is possible to define various scenarios, with different lighting conditions (e.g.: by adding spotlights, or changing lighting parameters) or even model the track with uneven ground.

To control the semaphores a ROS node was created. This node displays a simple menu allowing for the user to change the semaphore to exhibit in running time.

Figure 4. Representation of the autonomous driving competition track with semaphore panels



System Architecture

As mentioned before, all applications were built using ROS. The use of this middleware allows a simple swap of connection between the simulator and the real robot. Each ROS-Node created corresponds to an application, and the communication between nodes is made through topics using the publisher-subscriber topology.

The connections between nodes using ROS for real the robot is presented on Figure 5, and on Figure 6 is showed the connections on the simulator architecture. The first noticeable difference is the node that controls the semaphores on the simulator called semaphore_controller. This node is only needed on the simulator and it is used to change the semaphores projected on the monitors. The common elements are the conde_tracking node on which runs the application vision for the track detection, the conde_semaphore node responsible for the semaphore recognition algorithm, and the conde_decision node, in which runs a state-machine responsible for the decisions to take along the competition. The conde_tracking and conde_semaphore nodes can either subscribe to real USB cameras, conde_tracking_left, conde_tracking_right and conde_semaphore or the simulated ones, from the gazebo node. The communication with the conde_decision node is made by sending the measured distance and angle, crosswalk detection and the semaphore information. Then a decision is taken, and a message is sent to the conde_control node. This message complies the actual distance and angle, the distance and angle references to follow and the linear velocity on which the robot will move. Taking this data, the conde_control node does some computation and calculates the velocities to apply on each wheel. This call is made by publishing the velocity, on each wheel of the robot through the topic /cmd_vel. In the case of the real robot this information goes the real drives, otherwise goes to the simulated robot gazebo.

Using these packages, the students can program and test their own applications without worrying about the system's communication and its related configuration or what kind of information is needed to transmit between applications. The nodes that can be edited by university students in order to program their own applications are conde_tracking node, conde_semaphore node, conde_decision node and conde_control node. The coding of these tasks introduces some important areas like motion control and image processing that are very useful when designing real robotics applications.

Figure 5. ROS architecture for the real robot

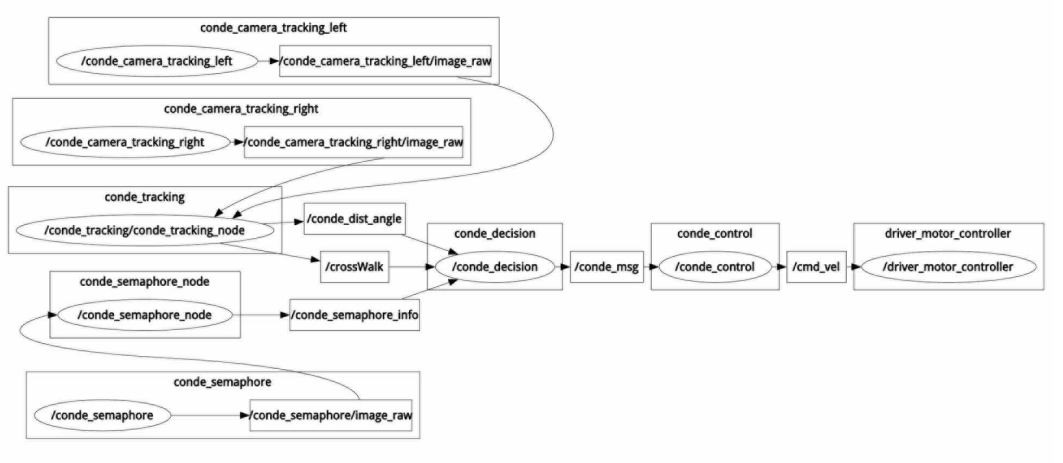
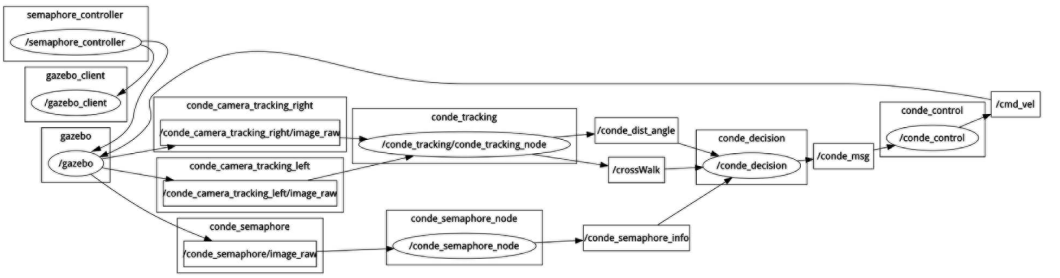


Figure 6. ROS architecture for the simulated robot



The gazebo node and the gazebo_client refer to the simulation world where the robot moves.

System Requirements

To run the simulator on the student's computer there are some requirements to have into account:

1. Operating system – Ubuntu 14.04.02;
2. ROS:
   a. ROS distro: indigo;
   b. ROS version: 1.11.16
3. Gazebo version: 2.2.3

There are the minimal requirements to run the simulator. All the platforms used are free and open source.

RESULTS

This section presents the results obtained from the evaluation of the simulator. Four tests were performed: the first two are related to the locomotion system, while the second set of tests were

performed by testing the image recognition program with real semaphore dataset and also by testing the Field Of View (FOV) of the real cameras and the simulated cameras. The first locomotion test was performed by taking the real robot and a test code that puts the robot moving straight forward given a distance reference and by measuring what was the real travelled distance made by the robot. This test was made on the real robot and the simulator using the same test code. The results are showed in Table 1.

Observing Table 1 makes clear that the precision of the obtained results are close to the real ones, with the maximum relative error value in the simulation of 4.65%.

The second test to the locomotion system was the square test. The objective is to put the real robot and the simulated one moving on the floor, with the path forming a square, and measure the translation error. This error is calculated as the Euclidean error (offset) between the starting point and the finishing point. This test was repeated four times at two different linear velocities, 0.3m/s and 0.5m/s, and with an angular velocity of 1rad/s.

The results show that the modulation for the locomotion system on the simulator is between the expected values and the parameters are well adjusted. The first column of Table 2 presents the linear velocities used when the robot is moving straight forward (the edges of the square), the second column shows the angular velocities used to make the robot turn (the vertices) in order to form a square, and the last column exposes the translation error in meters between the real robot and the simulated one. The translation errors obtained derive from the real robot motor controller and some nonlinearities in real world which are not included in the simulation.

Regarding the vision system, the first test to the cameras was performed by running the developed algorithm with a real semaphore dataset, analyzing the percentage of correctly classified semaphores, as shown in Figure 7. The dataset used is constituted by 186 semaphore park images, 111 for the stop

Table 1. Odometry results for the real robot and the simulated robot

	Odometry Results				
	Real Robot Measures (m)	**Simulated Robot Measures (m)**	**Real Robot Average Absolute Error (%)**	**Simulated Robot Average Absolute Error (%)**	**Average Relative Error Simulator vs Reality (%)**
1 meter tests	1.097	1.150	10.14	15.26	4.65
	1.103	1.153			
	1.107	1.153			
	1.102	1.154			
	1.098	1.153			
2 meter tests	2.085	2.153	4.44	7.68	3.10
	2.095	2.154			
	2.085	2.154			
	2.084	2.154			
	2.095	2.153			
3 meter tests	3.060	3.152	2.28	5.11	2.76
	3.055	3.155			
	3.063	3.153			
	3.080	3.154			
	3.085	3.152			

Table 2. Translation error for the square test

Linear Velocity [m/s]	Angular Velocity [rad/s]	Translation Error [m]
0.3	1	0.030
0.3	1	0.040
0.5	1	0.050
0.5	1	0.042

semaphore, 141 go forward semaphore, 104 for the left semaphore, and 158 for the right semaphore, adding all to the total of 700 test images.

The results showed a good performance using the developed algorithm to recognize the semaphores, with an accuracy of 99.37%.

The final test to the vision system was measuring the horizontal and vertical field of view. The setup used is shown in Figure 8. The camera is placed in front of a wall (in this case, in pointing to a chessboard) in a distance x, both in real world and simulator. FOV is the angle (α) which corresponds to the length 2y. The metric length 2y is proportional to the number of pixels on a captured image. The error between the real camera and the simulated one is calculated as the difference in the number of horizontal pixels between the chessboard edges. Vertical FOV was tested using the same technique. The results are presented in Table 3.

From Table 3, it is visible that the FOV parameter was well adjusted on the simulator. The error increases towards the edges due to the real camera lens distortion.

Figure 7. Semaphore detection results for the real and simulated robot

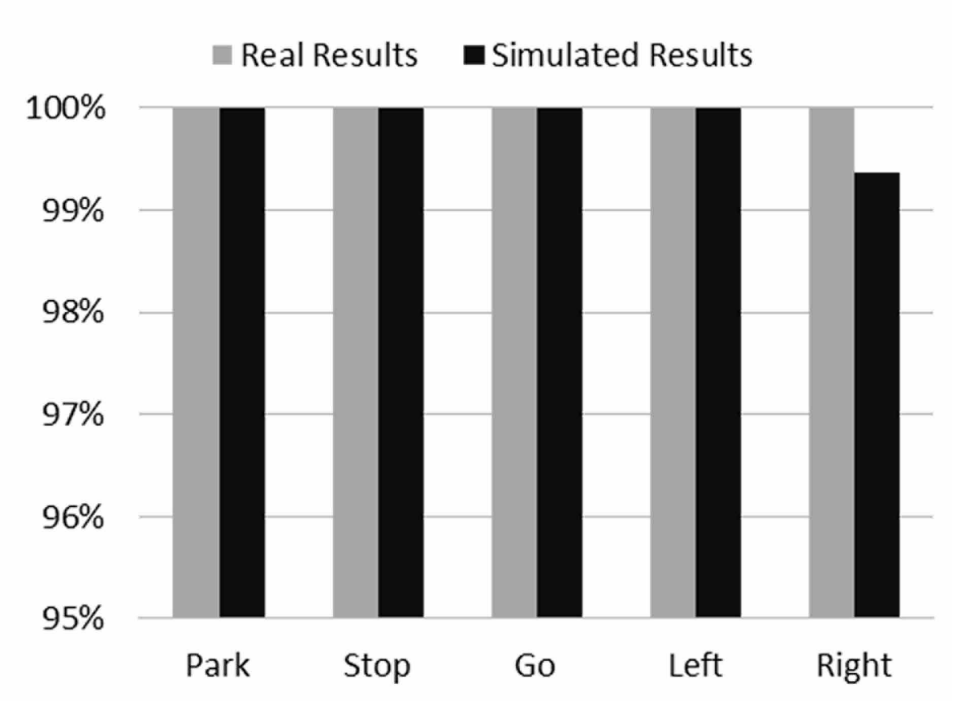
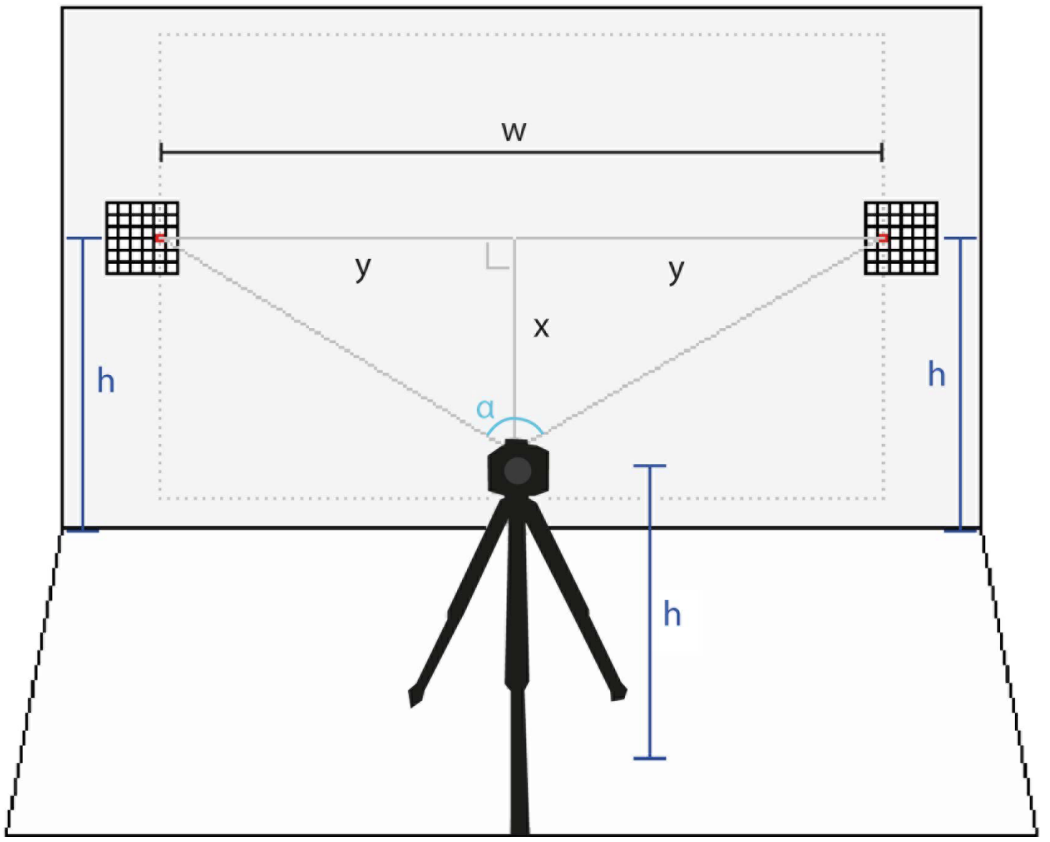


Figure 8. Setup used to measure the field of view



CONCLUSION

In this paper, it was presented a simulator that replicates the Portuguese autonomous driving competition as a tool to introduce, attract, promote enthusiasm and teach university students various relevant themes in the robotics field. The simulator was made using Gazebo as a 3D simulator and was tested by modeling a real robot, called "Conde", that uses a differential steering locomotion, but

Table 3. Horizontal and vertical field of view results

Horizontal FOV			Vertical FOV		
Real Line Length [pixels]	Simulated Line Length [pixels]	Absolute Error [pixels]	Real Line Length [pixels]	Simulated Line Length [pixels]	Absolute Error [pixels]
50	51	1	25	25	0
62	61	1	58	56	2
105	102	3	62	65	3
128	129	1	76	75	1
147	144	3	91	89	2

many others can be implemented using the same simulation world, making the proposed simulator modular. The different simulator's components communicate using the ROS framework through the available gazebo_ros_pkg, making the building of a simple architecture possible, which facilitates the users' coding task and allows reuse of developed code.

An evaluation of the simulator was performed, regarding the odometry of the modulated robot and the cameras used. The odometry results have shown an acceptable error, given the context of this work. By carefully inspecting the achieved results, we can see that the discrepancy between the reality and the simulation rounds the 0.05 meters. An improvement to reduce this error would be a simple calibration on the robot model. This discrepancy would also decrease with a correct calibration of the modulated robot in parameters such as inertia moment, for example. In the chosen robot model, the default values from Gazebo were used. Regarding the test of the cameras used in the robot model, it was performed by (i) running a semaphore recognition algorithm with a real semaphore dataset and analyzing the percentage of correctly classified semaphores and (ii) by measuring the FOV both in real robot and simulation. The results achieved have shown that the implemented algorithm are precise, with an accuracy of 99.37% regarding the dataset used, and an average error of 1.8 pixels regarding the horizontal FOV measures and 1.6 pixels on the vertical FOV. From this evaluation, it can be concluded that this simulator is versatile and realistic enough in this context, allowing the possibility of multi-robot development in the future, while also reducing the gap between reality and simulation. This modularity is a key feature of the proposed simulator, simplifying the task of importing existing robot models into the simulated world. The usage of the ROS framework allows the code-maintenance task to be simpler and modular.

The main purpose of this work is to use the developed simulator in a 2016 fifth year course and formulate an in-course competition to help students to overcome the obstacles that may appear during the resolution of autonomous driving challenges. As demonstrated on previous work, using a competitive mindset will provide motivation for students to overcome problems and improve their engineering knowledge and social skills, leading them to a more interesting learning process.

FUTURE WORK

The proposed objective of this paper was successfully achieved and a simulator regarding to the Portuguese Autonomous Driving Competition was created. Being a work in progress, there are some improvements that can be implemented. These include:

1. Add more features to the simulator, like the motion control with a joystick and/or keypad, obstacles to detect and avoid, objects such as traffic signs and the tunnel;
2. Add a node to evaluate the response of the robots for a given challenge. By having this application it would be possible to score and award the robots performance.
3. Formulate the in-course competition to be held in 2016 edition of Intelligent Robotics.

ACKNOWLEDGMENT

This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme, and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project «POCI-01-0145-FEDER-006961»

REFERENCES

Abiyev, R., Ibrahim, D., & Erin, B. (2010). EDURobot: An educational computer simulation program for navigation of mobile robots in the presence of obstacles. *International Journal of Engineering Education*, *26*(1), 18–29. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-77949517300&partnerID=40&md5=fc3bc277cfc61e7b3608a43b2a28892b

Audi Autonomous Driving Cup. (2016). Audi Autonomous Driving Cup. Retrieved from https://www.audi-autonomous-driving-cup.com/

Cardeira, C., & da Costa, J. S. (2005). A low cost mobile robot for engineering education. *Proceedings of the 31st Annual Conference of IEEE Industrial Electronics Society IECON '05* doi:10.1109/IECON.2005.1569239

Carvalho, A., & Borrelli, F. (2015). A Learning-Based Framework for Velocity Control in Autonomous Driving. *IEEE Transactions on Automation Science and Engineering*, *13*(1), 1–11. doi:10.1109/TASE.2015.2498192

CEABOT. (2016). CEABOT. Retrieved from http://www.ceautomatica.es/sites/default/files/upload/10/CEABOT/index.htm

Costa, V., Cunha, T., Oliveira, M., Sobreira, H., & Sousa, A. (2015). Robotics: Using a competition mindset as a tool for learning ROS. *Proceedings of the Second Iberian Robotics Conference Robot '15* (pp. 757–766). Doi:10.1007/978-3-319-27146-0_58

Costa, V., Rossetti, R. J. F., & Sousa, A. (2016). Autonomous Driving Simulator for Educational Purposes. *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*. Doi:10.1109/CISTI.2016.7521461

Gonçalves, J., Silva, M., Costa, P., & Sousa, A. (2015). Proposal of a low cost educational mobile robotics experiment: an approach based on hardware and simulation. *Proceedings of the 6th International Conference on Robotics in Education*.

Häne, C., Sattler, T., & Pollefeys, M. (2015). Obstacle Detection for Self-Driving Cars Using Only Monocular Cameras and Wheel Odometry. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)* (pp. 5101–5108). http://doi.org/ doi:10.1109/IROS.2015.7354095

ISTROBOT. 2016. (2016). ISTROBOT 2016. Retrieved from http://www.robotika.sk/contest/2016/index.php

Jong, J. J., Park, H., & Kim, C. (2012). Development of a Vehicle Simulator Based on a Real Car for Research and Education Purposes. *Proceedings of the FISITA 2012 World Automotive Congress* (Vol. 196).

Kuc, R., Jackson, E. W., & Kuc, A. (2004). Teaching introductory autonomous robotics with JavaScript simulations and actual robots. *IEEE Transactions on Education*, *47*(1), 74–82. doi:10.1109/TE.2003.817619

Lee, Y. H. (2008). Hardware-in-the-loop simulation for autonomous driving. *Proceedings of the 2008 34th Annual Conference of IEEE Industrial Electronics* (pp. 1742–1747). http://doi.org/ doi:<ALIGNMENT.qj></ALIGNMENT>10.1109/IECON.2008.4758217

Lenskiy, A., Junho, H., Dongyun, K., & Junsu, P. (2014). Educational platform for learning programming via controlling mobile robots. *Proceedings of the 2014 International Conference on Data and Software Engineering (ICODSE)*. Doi:10.1109/ICODSE.2014.7062695

Micro-Rato. (2015). Micro-Rato. Retrieved from http://microrato.ua.pt/

Micromouse. (2016). Micromouse Portuguese Contest – Official site of Micromouse Portuguese Contest. Retrieved from http://www.micromouse.utad.pt/

RoboCup. (2016). RoboCup. Retrieved from http://www.robocup.org/

RoboParty. (2016). RoboParty. Retrieved from http://www.roboparty.org/

Temeltas, H., Gokasan, M., Bogosyan, S., & Kilic, A. (2002). Hardware in the loop simulation of robot manipulators through Internet in mechatronics education. *Proceedings of the 28th Annual Conference of the Industrial Electronics Society IEEE '02* (pp. 2617–2622). Doi:10.1109/IECON.2002.1182806

Valente, A., Salgado, P., & Boaventura-Cunha, J. (2014). Grigora S: A low-cost, high performance micromouse kit. *Proceedings of the 11th Portuguese Conference on Automatic Control CONTROLO '14* (pp. 535–544).

XVI Portuguese robotics open. (2016). Retrieved from http://robotica2016.ipb.pt/indexen.html

Yusof, Y., & Hassan, M. (2012). Development of an Educational Virtual Mobile Robot Simulation. *Proceedings of the World Congress on Engineering WCE '12* (Vol. 2). Retrieved from http://www.iaeng.org/publication/WCE2012/WCE2012_pp864-868.pdf

Born in 1991 in the city of Porto, Portugal, with a master degree in Electrical and Computers Engineering on Faculty of Engineering of University of Porto (FEUP), Valter Costa is a Researcher Fellowship on INEGI - driving science & innovation and is a PhD student of the Doctoral Program in Informatics Engineering (ProDEI), also on FEUP. During its master's degree, he was a teaching assistant in two programming course units. His research interests are in computer vision and mobile robotics. Beyond that he is an enthusiastic for autonomous driving/self-driving cars and on development of platforms for teaching robotics. Also, he has participated in several robotic competitions, and in 2016 he conquered the third place in the XVI Portuguese Robotics Open.

Rosaldo Rossetti is currently a Senior Research Fellow and a member of the Directive Board of the Artificial Intelligence and Computer Science Laboratory, and an Assistant Professor with the Department of Informatics Engineering, University of Porto, in Portugal. His main professional activity is engineering education and research. More specifically, his research interests include complex systems analysis, behavioural modelling, social simulation, multi-agent systems, and spatio-temporal data analytics and machine learning. He focuses on applications of multi-agent systems as a modelling metaphor to address issues in artificial transportation systems, future mobility paradigms and urban smartification, industrial applications, behaviour modelling, potential uses of serious games, and gamification in transportation and mobility systems. Rosaldo Rossetti received the degree in civil engineering from UFC, Fortaleza, Brazil, in 1995, and the M.Sc. and Ph.D. degrees in computer science from UFRGS, Porto Alegre, Brazil, in 1998 and 2002, respectively. He carried out his doctoral thesis as a Graduate Research Student with the Network Modelling Group, Leeds University's Institute for Transport Studies, Leeds, UK. Dr. Rossetti served as a Board of Governors Member of the IEEE ITS Society from 2011 to 2013. He is currently the Chair of the ATS and Simulation Technical Activities Sub-Committee, an Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, a member of the ITS Podcast Editorial Board, an ITS Department Editor of the IEEE Intelligent System Magazine, and a member of the Steering Committee of IEEE Smart Cities Initiative, where he serves as the Editor-in-Chief of the Smart Cities News Bulletin and an Editor of the Readings on Smart Cities. He is also a member of the Technical Committee on Intelligent Industrial Systems of the IEEE Systems, Man, and Cybernetics Society. He will be the General Chair for the 2016 IEEE 19th International Intelligent Transportation Systems Conference, the flagship conference of IEEE ITS Society. He is also a member of ACM, APPIA (Portuguese AI Society), and a Founder Member of the Portuguese-Brazilian Society for Modelling and Simulation.

Armando Jorge Sousa has received his PhD in the area of Robotics in 2004 at the Faculty of Engineering of the University of Porto (FEUP), Portugal. He is currently a researcher at Inesc Tec and an Auxiliary Professor at FEUP. Main research interests include Automation, Robotics and Education. He is a member of the Portuguese Robotics Society and has been engaged in the organization of the National Festival of Robotics. He has received several RoboCup robotic soccer awards (2006, 2nd place world competition, etc) and was given the Pedagogic Excellency award in 2015 for the University of Porto. He currently participates in international research projects in robotics and in education. He has co-authored 6 indexed journal articles, 9 book chapters and more than 40 indexed conference papers in the areas of robotics and education.