

# A Study about the Transition of Simulated Algorithms to Real Robotics and its Application in Distance Learning

Clairton A. Siebra, Wanderson G. Souza

Informatics Center  
Federal University of Paraíba (UFPB) – Joao Pessoa, PB – Brazil  
clairton@di.ufpb.br, wandersonslash@gmail.com

***Abstract.** The use of virtual environments to simulate experiments of technical disciplines, such as Robotics, is a strategy that is being used in some distance learning courses. However, students may become frustrated when they discover that their simulated algorithms do not present the same behavior when running in real robots. This study, which is part of the preparation of an Educational Robotics distance learning module, intends to investigate the main reasons for this different behavior, so that such reasons can be presented and discussed with students. The idea is to encourage students in developing their algorithms considering solutions for this divergence. Thus, a practical and known problem could be transformed in an opportunity for learning.*

## 1. Introduction

Distance learning has undertaken a significant growth and change over the last decade to a point that it is now widely accepted as a viable and flexibly form of education. The use of this technology has, for example, allowed engineering courses to be delivered to locations and populations that have historically not been afforded opportunities for involvement [Schaefer et al, 2008]. However, the implementation of the distance learning in disciplines, such as Electrical and Mechanical Engineering, is still far behind the mature practice in narrative disciplines such as Economy or History [Jong et al, 2013]. The principal reason is that several of the engineering disciplines inevitably involve laboratory exercises and this fact drastically increases the complexity of creating an appropriate virtual environment that supports the learning process of students.

Facing this complexity, several approaches are using the idea of virtual laboratories that support remotely operated exercises. For example, the work of Casini et al [2003] presents a remote laboratory of automatic control whose target is to enable students to interact with a set of physical processes through the Internet, so that they are able to run experiments, change control parameters and remotely analyze the results. Another interesting example in [Popovic et al, 2013] describes a LabVIEW-based implementation of remote control of laboratory equipment for basic Electronics courses.

Remote labs are also being used for teaching topics in robotics. A teaching experience in this area [Casini et al, 2011] uses a web page that enables users to interact with the lab, which is based on the LEGO Mindstorms technology and the Matlab environment. In [Chaos et al, 2013], a virtual laboratory was implemented to enable students to develop signal processing and control algorithms for simulated robots and then deploy such algorithms on real robots for testing purposes.

According to these related works, there is a trend that remote robotic labs use real robots that are available to students to evaluate their algorithms. However, this approach is not appropriate to distance learning courses, once they normally attend more than 100 students at each semester. Thus, it is not easy to maintain real robots remotely available to students and manage their interactions. Previous approaches [Andular and Mateo, 2010; Wang et al, 2010] use to apply a booking system integrated within the learning environment to ensure an exclusive access control of one use at any time. However, the analysis of Brazilian learning environments shows that students use to mainly access the environment on weekends and this behavior increases the management complexity. Consequently the use of simulators may offer a more convenient approach to support the learning process of these students.

The problem appears when algorithms created in simulated platforms are used in real robots. In fact, it is not ensured that an algorithm, which is validated in simulation platforms, works as expected when such algorithm is loaded to real robots. This happens because robots input data and the action of their effectors are affected by several factors, such as robot hardware, physical constants and general conditions of the environment. This latter factor requires a special attention once the environment provides the perceptions about the world to robots via sensorial devices and this operational scenario presents several features that are changing in real-time. Examples of aspects, which compose the environment and are complex to deal with are: variations in different types of landscapes, sources of noising, interference of solar rays and odometry errors. The occurrence of isolated and infrequent errors is not generally a problem. However, the accumulation of such errors can lead robots to unacceptable levels of deviations, [Kyriacou et al, 2008]. Thus, this situation has a huge potential to frustrate the expectations of students and negatively affect their learning process.

This paper investigates the main reasons for this different behavior, so that such reasons can be presented and discussed with students. To that end, several experiments are carried out on virtual and real environments, so that their results can be compared and the reasons for divergences can be characterized. This study is part of the preparation of the *Educational Robotics* module that composes the *Intelligent Agent in Education* discipline of the Licentiate Degree in Computing course (distance modality, Federal University of Paraiba). The aim of this module is to present how concepts of robotics can be efficiently used to support the learning of several topics from disciplines such as Logics, Mathematics and Physics.

The remainder of this paper is structured as follows: Section 2 summarizes some projects that try to characterize this virtual to real transition. Section 3 presents our research method and experiments setup. Section 4 details the experiments and their results. Finally, Section 5 concludes this work with the main remarks and research directions.

## **2. From Virtual to Real Robots**

Some works in robotics has focused their efforts on analyzing the difference between the simulated and real robots behavior. The principal aim of such works is to enable an easy transition of algorithms that were originality developed in simulated platforms to real robots [Balakirsky et al, 2009]. The work of Xu et al. (2010), for example, analyses the

behavior of robots via a hierarchical state machine, which models the problem using soccer game rules once this project was applied to the Robocup soccer scenario. In this model, there are four basic states: goal keeper, defender, midfielder and attacker. The decision of changing between the states depends on the features of the robot and current game scenario. The robots use learning machine to improve their abilities of running, kicking and passing the opponents, so that the actions are adjusted along the time. At the end, the robots in both virtual and real implementations tend to perform the same actions. This approach was used by the *Humboldt* soccer team, which was the first Robocup team to use the same algorithm in both simulated and real leagues. This work, however, does not stress the reasons for different behaviors, but uses a machine learning strategy to adjust such differences along the real performance of the algorithm. Furthermore, it is domain dependent, so that it needs a different model for each different problem.

The work of Kyriacou et al (2008) argues that a robot's behaviour is affected by three aspects: the robot hardware, the implemented algorithm and the environment where this robot operates. According to Kyriacou et al, these three aspects generate a complex non-linear system that does not ensure that results of simulation experiments can be replicated in real environments. Based on this idea, the authors developed a new method to create simulations, which enables to predict the behaviour of robots in real scenarios and how such scenarios could affect the robots performance. This theoretical proposal, which relates robot and its environment, is described via a mathematical model called NARMAX polynomial. As the previous work, this approach does not stress the specific reasons for different behaviours, once the mathematical treatment of the input information is considered a black-box method.

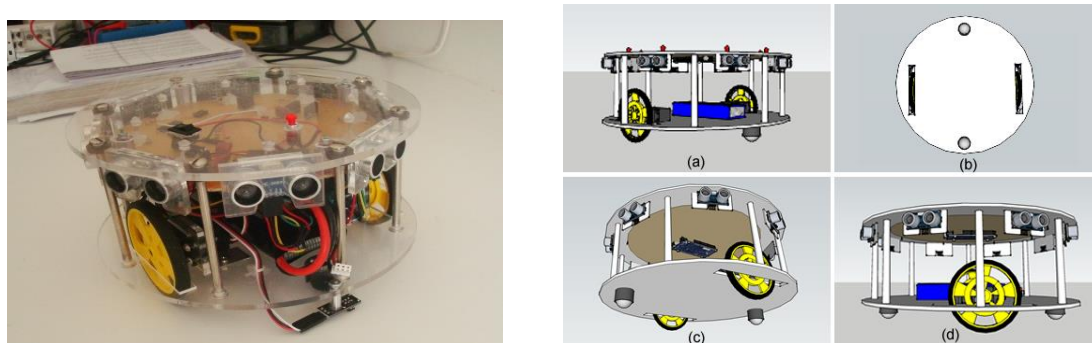
Chen et al (2009) proposed the concept of *mixed-reality* where a unique simulation uses real and virtual elements. This work is interesting because a robot that is navigating in a real environment must be able to avoid both real and virtual obstacles that are introduced in these environments. However, this work is more focused on synchronization problems for perceptions and updates of scenarios.

Independently of using robots in both simulated and real scenarios, any of these works clearly demonstrate which aspects generate deviations on the use of a same algorithm to control a robot in such scenarios. This kind of knowledge is very important to students that are developing their projects, so that they might be aware of future problems and can be able to develop algorithms that deal with this limitation in a preventive way.

### 3. Experiment Setup

This experiment intends to use an algorithm that resolves a typical problem of robotics (moving avoiding obstacles), which is presented in our Educational Robotics module, and use this algorithm to control a real robot and its virtual replica. Then the differences of behavior are analyzed and their causes are identified. The robot for this experiment is a mobile circular robot with 200mm of diameter. It has two sets wheels/servomotors and a third passive wheel that assists in maintaining the equilibrium of the robot. This robot is based on an Arduino Mega microcontroller and other components used in its development were a home-made omnidirectional sensor that used 8 ultrasonic sensors, a

compass Arduino shield for orientation and a transceiver for communication with the base computer. The virtual replica of this robot was modelled using the *Google SketchUp* software, which supports the definition of 3D elements. The real and virtual robots are illustrated in Figure 1.



**Figure 1. Robot used during our experiments and its virtual replica.**

The Unity3D was defined as the simulation platform. One of main reasons was its support to import models created via *Google SketchUp*, rather than only CAD models. Furthermore, Unity3D has a very decent free version for educational purposes and it accepts the definition of scripts using C# and JavaScript languages.

An anti-collision moving algorithm was used to this experiment, once moving avoiding obstacles is a basic problem for any mobile robotic system and its principles must be presented to students during an Education Robotics course. The pseudo code of the algorithm used along our experiments is detailed as follows:

---

**Prodecure RobotMoving**

---

**Repeat**

**if**  $\neg(\text{obstacle blocks the path})$  **then**

**Move a step ahead;**

**else if**  $(S_7 < D_{\min} \wedge S_0 < D_{\min}) \vee (S_0 < D_{\min} \wedge S_1 < D_{\min})$  **then**

**Turn left around its axis; // clockwise**

**else if**  $(S_1 > D_{\min}) \wedge (S_2 < D_{\min})$  **then**

**Turn right around its axis; //clockwise**

**else if**  $(S_7 < D_{\min} \wedge S_0 < D_{\min}) \vee (S_7 < D_{\min} \wedge S_6 < D_{\min})$  **then**

**Turn right around its axis; // anticlockwise**

**else if**  $(S_6 > D_{\min} \wedge S_5 < D_{\min})$  **then**

**Turn left around its axis; // anticlockwise**

**endif**

**End**

---

The variable  $D_{\min}$  represents the minimal distance that the robot should maintain to obstacles. If any distance returned by one of the sensors  $S_i$  is less than  $D_{th}$ , this means that the side covered by  $S_i$  is very close to the obstacle. Figure 2 shows the result of this algorithm when it is used to avoid a rectangular and a triangular object.

Finally, this same algorithm was used by the real robot and its virtual replica. The next section discusses the results of this experiment.

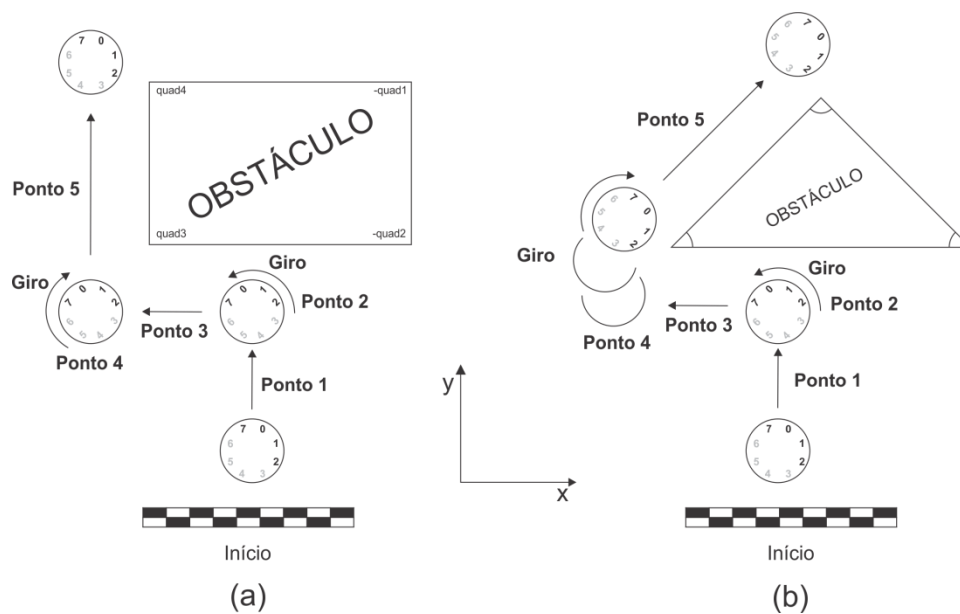


Figure 2. Schemas that show the performance of the anti-collision algorithm.

#### 4. Experiments and Results

The main aim of the experiments is to show which features account for different behavior between real and virtual robots, so that students can understand such features and develop algorithms that could decrease possible deviations. Next subsections detail these experiments.

##### 4.1. Scenario 1: omnidirectional sensors analysis

The first scenario intends to analyze the performance of the omnidirectional sensor and the navigation system when they face a rectangular obstacle and try to avoid it. A simple schema of this experiment can be seen in Figure 2 (right-hand side), which shows the points (from 1 to 5) where the measures were captured to construct the graphs. The total execution of both robots, real and virtual, spent about 25 seconds, the sampling rate was 2 measures per second and the next graphs show the distance calculated from the real (Figure 3.a) and virtual (Figure 3.b) robots to the obstacle.

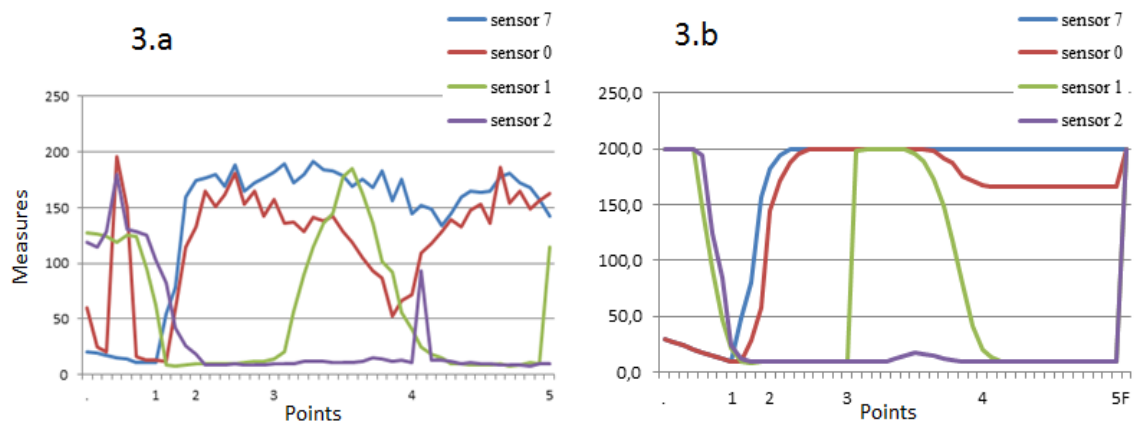


Figure 3. Distance measures returned from four sensors, considering the real (left-hand side) and virtual (right-hand side) robots.

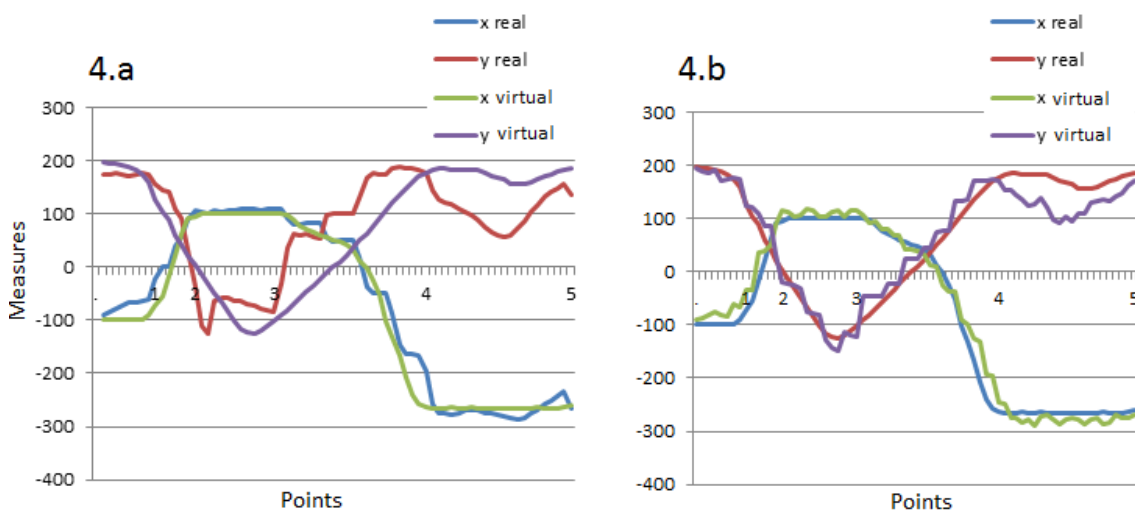
These measured distances correspond to the four sensors that become active with the presence of the rectangular obstacle. The range of the sensor is 200 centimeter. Thus, if the measure is 200 centimeters this possible means that the area covered by a sensor is free of obstacles.

This figure shows that sensors are affected by external interferences in several points along the real simulation (Figure 3.a). Sensor 0, for example, presents the most problematic measures between the points 0 and 1. In this case, the sensor sent an ultrasonic wave and it did not receive an answer during the expected time. Thus, the sensor considered the inexistence of obstacles on its covered area and set the maximum value (200 cm) to this measure. A similar situation happens with Sensor 2 between the points 4 and 5. Differently, as the virtual modelling is a “perfect world”, we see that such interferences from external factors do not exist and the measures (lines) are totally clean and correct (Figure 3.b).

When students are only working with simulations, they do not need to be worried about these external interferences on sensors. However, as such interferences have the potential to affect the information reading of sensors, real robots may have a different behavior when compared to their virtual versions. In this way, this kind of experiment is very important as a resource to clearly show sensor interference problems to students.

#### 4.2. Scenario 2: navigation system with compass

The schema of this experiment can be seen in Figure 2 (right-hand side), which also shows the points (from 1 to 5) where the measures were captured to construct the next graphs (Figure 4). The time of execution and sampling rate were the same of the previous experiment. The values represented by the lines mean the x and y Cartesian distance from the position where the measure is taken to a reference, considering both virtual and real robots.



**Figure 4. Measures of four sensors, considering real (left-hand side) and virtual (right-hand side) robots that use a navigation system based on a compass.**

Graph 4.a shows the tracks of virtual and real robots when they are not using a navigation system. In this case, the omnidirectional sensor accounts for leading the robot around the obstacles. However, it is possible to see the considerable difference between

the track of real and virtual robots. The main reasons are the sensor errors, as discussed in the previous scenario, and the different rotation/power of the servomotors. Even using the same configuration for both servomotors (left and right wheels), it is almost impossible to have two commercial servomotors that ensure the same physical constants, as better discussed in the next scenario (Section 4.3).

After running this experiment, we repeated the same steps but now using a navigation system based on a compass. This system enables that the robot revises its route along the track, decreasing the difference to the optimal case that is represented by the values of the virtual robot. Graph 4.b shows that the track of the virtual robot is still the same. However the lines of the real robot present an undulated appearance, once the robot is continually revising its route to maintain a straight line when an obstacle is not blocking its path. This is possible due to the information provided by the compass, which is used by the navigation system to control the servomotors. Note that the tracks between virtual and real robots are still different, however the total error was drastically reduced.

This lack of orientation is one of the most common problems when simulated algorithms are loaded into real robots, so that it is also a basic topic to be presented to students of a Robotic course. Traditional approach to fix this problem are based on lines and grids [Buschmann et al, 2004], where light sensor are used to correct the position of robots. However this approach is a quite artificial once we need to modify the environment using marks along the scenario. Thus the use of resources, such as the compass sensor, must be presented to students as a form to provide an autonomous ability to robots independently of modifications in the environment.

### **4.3. Scenario 3: performance of effectors (servomotors)**

As discussed in the previous scenario, differences between servomotors can affect the behavior of robots. The students' comprehension of this problem is important because servomotors are the most common type of effector used in robotics. To that and, it was initially developed a RPM (Rotation per Minute) meter and fixed to the two wheel of the robot (Figure 4). Then, the robot was placed on a straight line and moved itself along this line during 6 seconds. After that, the number of spins of each wheel was identified and multiplied by 10 to calculate the rotation of such wheels in one minute interval. This test was repeated to different rotation speeds and directions (clockwise and anticlockwise). Speed and direction are controlled by the weight of pulses in milliseconds that are sent to the servomotors.

The pulses used along this experiment are presented in Table 1. According to this table, when pulses of 1500ms are sent to motors, their rotation is null. The rotation, and consequently the robot speed, increases in a non-linear rate when the values change from 1500ms to 1700ms (clockwise direction) and when the values change from 1500ms to 1300ms (anticlockwise direction). As the motor are places in opposite sides of the robot chassis (mirror effect, see Figure 1), we need to send tuples of pulses that generate the same RPM in the wheels (Motors 1 and 2), however in opposite directions. For example the parallel application of 1300ms (Motor 1) and 1700ms (Motor2) pulses theoretically ensure that the robot will move ahead at maximum speed.

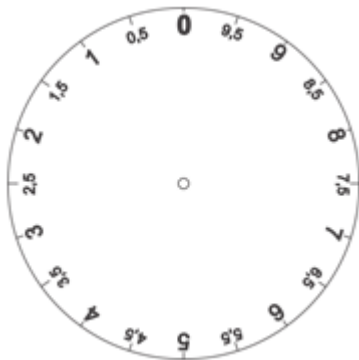


Figure 5. RPM meter

Table 1. Relation between pulse weight (PW) and RPM of two different motors

Motor 1		Motor 2	
PW	RPM	PW	RPM
1300	48	1700	46,2
1350	47,1	1650	45,5
1400	43,1	1600	41,2
1450	29,5	1550	26,5
1500	0	1500	0
1550	-28,3	1450	-26,9
1600	-43,8	1400	-40,9
1650	-47,3	1350	-46,6
1700	-47,6	1300	-47

Figure 6.a presents the relation between RPM and pulse weight in milliseconds to the two servomotors of our real robot. In this case, the robot is lift (no contact with the soil), so that we can apply the same pulse weight to both servomotors. The divergent lines show that, to the same pulse weight signal, the motors present different RPM values and, consequently, different angular velocity. This is the reason that leads the robot to slightly turn to the side of the motor that has the slower rotation.

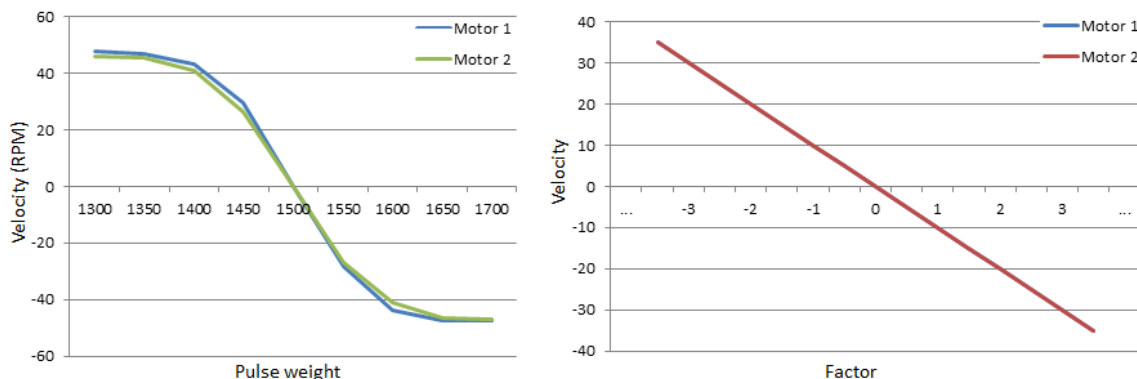


Figure 6. Comparison of the velocity of two motors used in the real (6.a) and virtual (6.b) robots to different pulse weight values.

Measures of velocity work differently in simulators. In real robots, velocity is measured in terms of meters per second (m/s) or RPM. In simulated environments, the smaller time unit is called *Frame*. Then, to calculate the velocity, we just need to capture the time spent for a simulation frame and use a factor to represent changes of velocity where negative factors mean anticlockwise direction, while positive factors mean clockwise direction. Then, motors velocity can be calculated as: “*motorspeed = factor \* frametime*”. Each simulator has its own frame time constant and it needs to be captured via some command available in the simulator API. In our case, the command was “Time.deltaTime”. Using this idea, Figure 6.b presents a similar test, however it was carried out in a simulated environment. We can observe two differences in this graph regarding Figure 6.a. First, the relation between pulse weight and RPM is linear. Second,



there are not synchronism divergences between the measures of motors 1 and 2. Thus, the robot always goes ahead in a straight line.

#### 4.3. Scenario 4: battery charge and its effects

The aim of this experiment was to show how different levels of the battery affect the robotic system and, in special, its effectors such as servomotors. When students work in simulated environment, this problem of the battery discharging is completely forgotten, once the simulation only works considering a full energy source. This experiment used a Lithium battery 7.4 Volts, which was modified to different charging levels.

The first cycle of tests was carried out using the maximum charging (100%) of the battery and servomotors configured to their maximum rotation, which correspond to the use of 1300ms and 1700ms pulse weight values to motor 1 and 2 respectively. Then, using a charging balance device called IMAX B6AC, we could gradually reduce the charging to 50%, 25%, 10% e 5% of the maximum charge and run again the experiment with each of these values. The results are shown in Table 2. It is possible to observe the sudden decrease in performance between the 25% and 10% charging levels. This happens because the torque of motors is directly related to the battery charging. The higher this charging, the stronger is the applied torque. When the level is inferior than 5%, the servomotors do not generate enough torque to move the robot. This kind of experiment is not possible to be carried out in a simulation mode because any of the simulation environments that were investigated along our research support this charging feature.

**Table 2. Values of RPM for different levels of battery charging**

	<b>RPM for 100% of battery charge</b>	<b>RPM for 50% of battery charge</b>	<b>RPM for 25% of battery charge</b>	<b>RPM for 10% of battery charge</b>	<b>RPM for 5% of battery charge</b>
Motor 1	48	42,4	40,2	32,9	30,6
Motor 2	46,2	41,3	38,8	31,7	29,7

## 5. Conclusions and Research Directions

Robotics has an important role as a motivational and practical tool to support the learning process of disciplines such as Logic, Physics and Mathematics. Thus, it is important to show to students of Licentiate degree courses the advantages of this resource. The principal contribution to this work, which is focused on the learning of Educational Robotics principles, is to make clear to students the fundamental differences that exist between simulated and real environments. In fact, distance learning students need inevitably to use simulated environment to complement their formation. Thus, they must understand the difference from such simulation platforms to real world applications.

This work presented basic scenarios that could be included in an Educational Robotics module. However, other further scenarios could be discussed such as the effects of the battery discharging on the range and precision of sensors. The investigation of additional scenarios and use of other typical robotics algorithms (e.g. surface covering, pathfinder, etc.) are possible themes for future researches.

## References

- Andujar, J. and Mateo, T. (2010) “Diseno de laboratorios virtuales y/o remotos. Un caso practico”. *Revista Iberoamericana de Automatica e Informatica*, vol. 1, pp. 67–72.
- Balakirsky, S. et al (2009) “From Simulation to Real Robots with Predictable Results: Methods and Examples”. *Performance Evaluation and Benchmarking of Intelligent Systems*, pp. 113-137.
- Buschmann, C., Müller, F. and Fischer, S. (2004) “Grid Based Navigation for Autonomous Mobile Robots”. *Proceedings of the Workshop on Positioning, Navigation and Communication (WPNC 2004)*, pp. 157–162.
- Casini, M., Prattichizzo, D. and Vicino, A. (2003) “The Automatic Control Telelab: A User-Friendly Interface for Distance Learning”. *IEEE Transactions on Education*, 46(2):252-257.
- Casini, M., Garulli, A., Giannitrapani, A. and Vicino, A. (2011) “A LEGO Mindstorms multi-robot setup in the Automatic Control Telelab”. *Proceedings of the 18th IFAC World Congress*, pp. 9812–9817, Milan, Italy.
- Chen, I., Macdonald, B. and Wunsche, B. (2009) “Mixed Reality Simulation for Mobile Robots”. *IEEE International Conference on Robotics and Automation*. Kobe, Japan.
- Jong, T., Linn, M. and Zacharia, Z. (2013) “Physical and Virtual Laboratories in Science and Engineering Education”. *Science* 19, 340(6130): 305-308.
- Kyriacou, T. et al. (2008) “Accurate robot simulation through system identificaiton”. *Proceedings of the 10th British Conference on Mobile Robotics - Towards Autonomous Robotic Systems*. United Kingdom, pp. 1082-1093.
- Popovic, B., Popovic, N., Mijic, D., Stankovski, S. and Ostojic, G. (2013) “Remote control of laboratory equipment for basic electronics courses: A LabVIEW-based implementation”. *Computer Applications in Engineering Education*, 21(S1):E110–E120.
- Schaefer, D., Scott, D.W., Molina, G. J., Al-Kalaani, Y., Murphy, T., Johnson, W., and Thamburaj Goeser, P. (2008) “Integration of Distance Learning Technology into Traditional Engineering Physical Laboratory Exercises”. *ASEE Southeast Section Conference*.
- Wang, Z., Dai, Y., and Yao, Y. (2010) “An Internet-Based e-Experiment System for Automatic Control Education and Research”. *Proceedings of Second International Conference on Communication Systems, Networks and Applications*, (ICCSNA), Hong Kong, China, vol. 2, pp. 304–307.
- Xu, Y., Mellmann, H., and Burkhard, H. (2010) “An Approach to Close the Gap between Simulation and Real Robots”. *Proceedings of the Second international conference on Simulation, modeling, and programming for autonomous robots (SIMPAN'10)*. Heidelberg: Springer-Verlag. pp. 533-544.