# A Thousand Robots for Each Student: Using Cloud Robot Simulations to Teach Robotics

**Ricardo Tellez**

**Abstract** One of the main problems when teaching robotics is the lack of robots for all the students in the class due to its costs and difficulty to maintain. In this paper, we analyze the advantages and drawbacks of using simulations for teaching robotics instead of real robots, as a solution to that problem. We describe a simulation tool based in the cloud that allows the simulation of complex robots off the shelf by using only a web browser as the base system for teaching robotics. Finally we provide the description of a teaching protocol for minimizing costs and troubles while maximizing students experience, which makes a combined use of simulations and real robots.

**Keywords** Teaching robotics · Robot simulations · Web application

## 1 Introduction

Robotics is a fashion subject, that is in an exponential growth all over the world. New companies about robotics are created every week and the forecast indicates that it is going to increase even more [1]. Hence, there is an increasing need of more prepared engineers in this subject.

Teaching robotics requires to teach in the most optimal way in order to transfer a knowledge that advances quickly, and to engage students into it. Fortunately, at this moment in time, we have access to a large base of commercial robots, schematics of open source robots [2], and free software [3]. All of that makes learning robotics easier than ever. However, the own existence of such large number of robotics options, makes very difficult to teach it, because making the learning experience work for all the students of the class can be, first, very expensive, and second, very complex.

R. Tellez (✉)
The Construct Sim, San Francisco, USA
e-mail: rtellez@theconstructsim.com

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent
Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_14

We identify the main problems of teaching robotics as the following:

1. Robots and components for robots are expensive. Hence it can be very expensive for the school to provide those robots or parts to each of the students.
2. Even if the school has access to the robots or parts, it takes a long time to correctly make work a given robotic setup. Robotics is a difficult subject, and requires the proper functioning of a large amount of different parts. If the student has to build all those parts and make them work it is going to take time. Furthermore, the number of problems exponentially increase with the number of students in the class.
3. Finally, setting up a robotics development environment requires a huge effort for the students. Teachers can design the best experiments with robotics ever, but making those run in the students own environments (or even in the environments the school provides) is very tricky. Students have access to different types of computers and operating systems. In the same line, schools have different computer configurations configurations, old computers, etc. Hence, even if the teacher achieves to make the exercises work in her system, making the exercises work for all the students is a different and more complex task.

In order to solve the first and second problems we propose the use of robot simulators for teaching robotics. In order to solve the third problem we propose the use of web based simulations in the cloud.

## 2 Using Simulations for Teaching Robotics

The use of robot simulations for teaching means that, instead of using a real robot to make students learn about robotics, the teacher uses a computer program (the simulator) that shows the robots on the computer screen as if it were the real robot. The simulated robot is called a model of the robot. There are many different simulators available, each one with its own characteristics and working flow. Depending on the simulator used and of the model quality, the model can look and behave more or less close to the behavior of the real robot (see Fig. 1 with real Nao robot and Nao simulation model for Webots simulator).

In order to start simulating a robot, the simulator program must be installed in the computer of the user previously to start working with it. It usually requires specific setups or configurations (for example, a given operating system in the computer, some specific libraries installed, etc.) prior to the installation. Those requirements make the installation process a little complex in most of the cases.

Once a simulator has been installed in a computer, the teacher must create the model of the robot that she wants to use in her classes. Another option is to get the model from an already created source. There are different repositories on internet providing models already made for very famous robots, even if in most of the cases, make those models work on the students computers represents a challenge in itself.

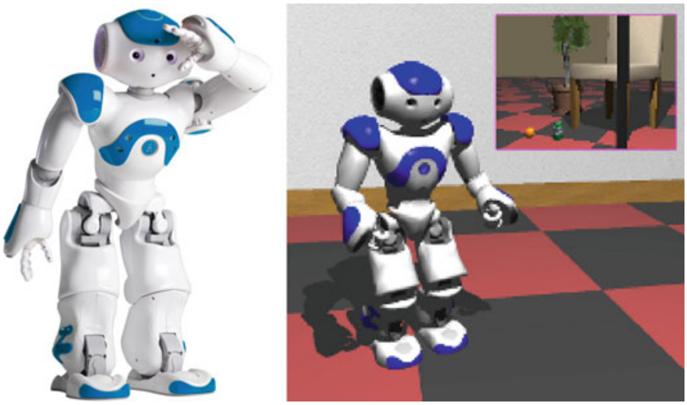

**Fig. 1** The real Nao robot and its simulation model of Webots simulator

Users always have the option of building their own models for existing robots, or for other robots that they invent.

Once the robot model is available, it is required to create a simulated environment for it. For example, the Nao robot of Fig. 1, is in a main room environment. The whole thing including the robot model and the environment is called a simulation. The simulation, is the program is executed inside the simulator and describes what must be shown on the screen by the simulator (that is, the robot in its environment).

The robot model must have a programming interface that allows users to command the simulated robot, reading its sensors and sending commands to its actuators. For example, in the case of Nao robot, the programming interface allows users to capture data from the (simulated) camera that the robot has in its head, or send commands to the motors of the joints and make the robot walk.

Once a model and its environment are created inside the simulator, teacher and students can program it as if it were the real robot using the programming interface. When the simulator executes the created program, the resulting behavior of the robot is shown on the screen.

Latest simulators tend to present to the user the same kind of programming interface that the real robot has. This fact represents a huge improvement from old robotics systems, since a program created for the simulated robot can be used later in the real robot without modification, and make the real robot behave (more or less) the same as the simulated robot.

Finally, to say that there exist a lot of different simulators available, each one with its own specifics. Some simulators are mainly focused on industrial robots, others on service robots. Some are based on standards and some have their own protocol. Deciding which one to use may be key for the teaching at hands. Table 1 contains a table of the most used simulators for robotics.

**Table 1** List of most popular simulators, their intended scope and their availability for different operating systems

| Simulators | Operating system | Scope |
|---|---|---|
| Gazebo | Linux, MacOS | Service and industrial robots |
| Webots | Windows, Linux, MacOS | Service and industrial robots |
| V-rep | Windows, Linux, MacOS | Service and industrial robots |
| UsarSim | Windows | Service and industrial robots |
| Morse | Linux | Service robots |
| RobotStudio | Windows | Industrial robots |
| Virtual robotics toolkit | Windows, MacOS | LEGO and VEX robots |
| Visual components | Windows | Industrial robots |
| Robot virtual worlds | Windows | LEGO, VEX and TETRIX robots |

### 2.1 Advantages of Using Simulators

Using a simulation in front of a real robot has several advantages:

1. High reduction of the cost. Real robots tend to costs between hundreds and thousands of euros. Simulations instead can start from zero to a hundred, depending on the requirements and quality of the simulator.
2. Each student has access to her own (simulated) robot. Having a real robot for each student is impossible even in the best schools, because of the prices. The result is that the students must group in order to access the real robot and test their algorithms. This results in a very poor learning experience for the student. Instead, when using a simulator, each student can have to the simulated robot by using only a computer.
3. Students cannot break the robot by doing errors. If a real robot is used, a simple mistake in the code can make the robot crash or fall, breaking its parts. With simulations, student's mistakes cause no harm.
4. Access to the best robots of the world (see Fig. 2). The best robots of the world, the ones being used now in the best research labs of the world, have a simulation model available. By using a simulation, students can access those robots and learn from the results of the best labs of the world.
5. Access to many different environments for the robot. Even if the school manages to have a real robot for each student, it is very unlikely that will be able to provide interesting environments for those robots (basically because you have to build those environments). If you use a simulator, you can place the robot in any environment you can think of. For instance it is possible to have a robot on space, deep water or nuclear reactor, and test the students programs for the robots in those environments.

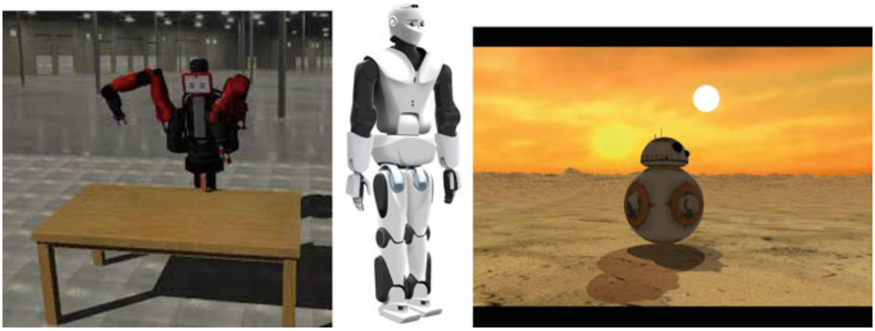

**Fig. 2** A simulation of Baxter (from Rethink robotics), Reem-C (from Pal robotics), and BB-8 (from Disney) robots, some of the most advanced robots in the world

## *2.2 Drawbacks of Using Simulators*

Using a simulator for teaching, has a series of drawbacks in front of real robots:

1. No learning about real hardware. By concentrating the student on the simulation, she learns all the inner workings about simulations, a thing that is very different from the inner works of real robots. Students don't get this special learning that is obtained by working with real stuff, and all the problems that are associated to real robots.
2. Learning with simulators is mostly about software, with almost no learning about mechanics nor electronics. 90 % of simulators are concentrated on learning how to program a robot to do something. There is no learning about how to build the mechanical parts of the robots or the electronic systems inside the robot that run the control software.
3. Difference between the behavior of real and simulated robot. Even for the best simulators, the models of the robots are not perfect. Hence, the behavior of the simulated robot varies from the behavior of the real robot. How this variation is important or not depends on the level of programming one wants the students to concentrate at. If the level of programming is at that of the controller, then there is usually a large difference between real and simulation. If instead the focus at the level of functionality, the behaviors are very similar to real robots.
4. Simulation are less engaging for students. Making a simulated robot grasp an orange from a table is not as exciting as making the real robot do it. Hence, simulations do not have the same appealing as real robots. However, this engagement highly depends on the implication of the teacher and her ability to create really engaging simulation examples or exercises.
5. Many of current robot simulators are tricky to install, set up and have them working in different equipments. When using a simulator, there is a previous work that needs to be done by the teacher, which requires installing the simulator in the computers that are going to be used, if they belong to the school. If students'

computers are going to be used, all kinds of errors and problems appear just trying to install, for which the teacher must provide some kind of support.

6. Creating the robot model at present is not a simple task and requires a lot of previous work to create it. This situation is at present quickly improving, because there is an emerging market for web pages that provide already made complex robot simulations to be used off the shelf [4, 5].
7. If several simulators are needed (because of different levels of proficiency of the students, or because teaching different functionalities are required) then the setting up work multiplies and becomes more complex. Of course, teachers must provide the simulation for each of the setups which is not straight.

## 3 Using Web Based Simulations for Teaching

Given some of the drawbacks presented in the previous section, we suggest the use of web based simulators for teaching instead of desktop based simulators. The difference between desktop and web simulators is that the former requires to be installed in a specific computer (with all the drawbacks indicated above), and the later requires no installation at all. While desktop simulators require the user to install the simulator in the user desktop or laptop computer, the web simulator is just a web page that the user can access with her account using only a web browser. Users can use any browser, which means that they can simulate from any computer at any location, including from home, school of internet cafe.

### *3.1 Advantages of Web Based Simulators*

These are the advantages of using a web based simulator:

1. Do not need installation nor maintenance. Installation and maintenance of the simulator is performed by the web portal that provides the simulator. Hence, neither the school nor the students have to deal with this unrelated task. This is specially convenient for some schools where the policy of installation of software is very strict and requires a long bureaucracy process.
2. Teachers can be sure that their exercises are going to work as they have designed them. Since all students and teachers are using the exact same simulation system, the results that are going to be obtained are exactly the same. Teacher can prepare exercises and show steps to follow that, if correctly performed, will produce the expected results in the simulated robot.
3. Students cannot break the simulation program by making errors or miss use of the simulation. If students make mistakes and something goes wrong crashing the simulation, they just can relaunch the web simulation again and start again from the initial conditions. Performing that error in desktop computers is one of the errors that consumes most of the time when teaching robotics, since students

perform errors that destroy the simulation system in their computers. This may require even to reinstall the whole operating system. With web simulators students cannot break any system, what makes students more prone to experiment and try new programming ideas.

4. Different simulators available which allows entrance at different level of complexity. A web simulation system can provide different simulators at the same portal, each one concentrated in teaching a specific subject (for instance, simulators concentrated in robotics arms, or simulators for humanoids or simulators for evolutionary robotics), or different levels of proficiency (for high school students, for university students, etc.).
5. Students can use any type of computer. In a class, there are all kinds of students, each one using their preferred operating system. As can be seen in Table 1, not all the simulators can work in all the operating systems. With a web simulator, this problem fades away since the only required thing is a web browser, which is available for all types of operating systems.
6. Students and teachers can work from anywhere. Since the only requirement is to have access to internet, teachers and students can actually be anywhere while doing their simulations. This opens a huge door for remote teachings.
7. Students can cooperate with their mates while working on the simulation. Web simulation is by nature a collaborative thing, at the contrary of desktop simulations. Hence, differently from desktop simulators, web simulators allow the work of different people at the same time on the same simulation. This is very useful when students need to cooperate to program the different parts of a robot, or when the teacher needs to see where the student is stuck and help her get the solution.

### *3.2 Drawbacks*

As any system, web simulations have also drawbacks:

1. Users need a fast enough internet connection. Web simulations are run through an internet connection, and since the simulations tend to be complex, it is required to have an internet connection with fast speed.
2. By being on the cloud, web simulation inherits all the security problems that any cloud system has. It is very difficult to ensure that the data stored in the cloud will not be accessed by unauthorized people.

## 4 A Case Study: The Construct Web Simulation System

As a solution for web simulation we propose The Construct. The Construct is a web platform in the cloud that provides a large list of simulators ready to be used by means of a web browser. It is a web based simulator with all the advantages (and drawbacks) explained in the previous section.

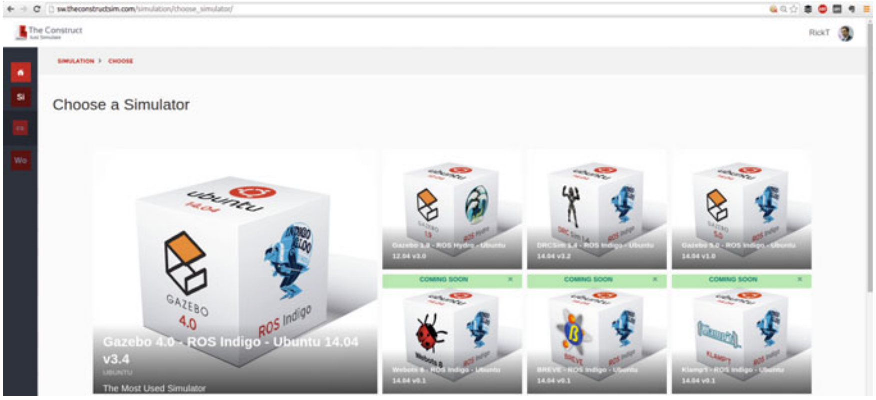
sw.theconstructsim.com/simulation/choose_simulator/
The Construct
RickT
SIMULATION > CHOOSE
Choose a Simulator
Gazebo 4.0 - ROS Indigo - Ubuntu 14.04
v3.4
UBUNTU
The Most Used Simulator
Gazebo 1.9 - ROS Hydro - Ubuntu
12.04 v3.0
DRCSim 1.4 - ROS Indigo - Ubuntu
14.04 v2.2
Gazebo 5.0 - ROS Indigo - Ubuntu
14.04 v1.0
COMING SOON
COMING SOON
COMING SOON
Webots 8 - ROS Indigo - Ubuntu
14.04 v0.1
BREVE - ROS Indigo - Ubuntu
14.04 v0.1
KlampT - ROS Indigo - Ubuntu
14.04 v0.1

**Fig. 3** A view of the simulators available at The Construct

The Construct allows to execute simulations off the shelf by using WebGL visualization. Users do not have to install anything, not even in their browsers, since all modern browsers already support WebGL based code. When the user access the platform using her preferred web browser, she has access to a list of different pre-configured robot simulators (Fig. 3).

Once the user access the portal, she can select which simulator to start simulating with. Selection of the simulator depends on the type of work she is going to do, and the level of complexity of the simulations. For instance, for beginners, it is easier to start using Webots simulator [6] because it is more simple in terms of interface and tools. However, if you want to learn about the latest robotics technologies like ROS [3] you may need to use the Gazebo simulator [7]. Finally, if you want to develop robotics with deep learning you should use the DRC+CUDA simulator [8].

The simulators available in The Construct are in fact existing desktop simulators. This implies that simulations created with the desktop versions of the simulator are compatible with the web version, and vice versa. Hence, teachers and students can upload to The Construct their already existing simulations and run them in the web. The opposite is also possible.

The Construct only needs a WebGL enabled browser. Most modern browsers are. Officially supported are Safari, Chrome, Firefox. This means that it can be used with any device that has one of those browsers, including Linux, Windows, MacOS, or even tablets and smartphones.

Apart from showing the simulator on the web browser, The Construct incorporates a series of very useful features for teaching robotics, that make the system go far beyond its desktop versions.

### 4.1 Sending Simulation Files to Students

When a teacher has created a simulation for their students, she can directly share the files that compose the simulation with all her students with a single click, without having to send files through email, or storing the files in some place of the internet. With a single click, the file selected is sent to a list of students provided by the teacher. From that moment, the students of the list can run the exact simulation that the teacher created, each student in her own environment free to experiment without interfering with the other students simulations. The opposite is also true: the students can send their simulations to the teacher from within The Construct and be sure that their simulations will work for the teacher as they expected.

### 4.2 Sharing Simulation with Teacher

If a student has a problem and does not know how to continue, she can request help from the teacher, by sharing her running simulation with the teacher. When a running simulation is shared, both the student and the teacher are seeing the same running simulation. This implies that the teacher can see what is failing in the simulation and programs of the student, and help her understand the error. The teacher has complete access to the simulation of the student and can modify it or suggest new ways of continuing.

This same mechanism of sharing can be used between students to collaborate between them in the programming of a simulated robot. In the same sense a Google Docs allows different people to write over the same document, sharing a simulation allows different people to work over the same simulation. It is just a matter of selecting the option of sharing and specifying the list of users that the owner of the simulation wants to share with.

After the collaboration is finished, the owner of the simulation has the right at any time to close the sharing session and forbid the access to her simulation.

### 4.3 Integrated Development Environment

Neither students nor teachers need extra programs than the web browser to simulate the robots and create the control programs for the simulated robot. A full environment is provided including IDE, web console and also a Python web environment. Hence, the full learning can be done using only a web browser.

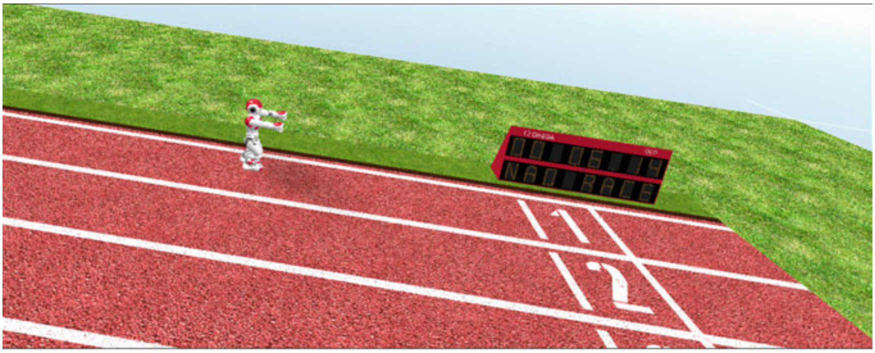

**Fig. 4** Simulation contest of Nao robot running the 10 m

### 4.4 Engage Students Through Simulation Contests

One way to encourage students and robotics research is by doing contests where the participants have to compete against each other. For example one of the most important robotics simulation competition was the Virtual Robotics Challenge [9] where the simulation was web based and involved the best robotics teams of the world to control a human size humanoid robot on a nuclear disaster area.

This kind of competition can be easily done using web simulations at The Construct, like for example the Robto Race to Hawaii contest based on Nao robot race, where students around the world should make a Nao robot walk for 10 meters as fast as possible (see Fig. 4).

## 5 Proposing an Optimized Robotics Teaching Environment

One of the main criticisms to using simulations to teach robotics instead of real robots is that students get a distorted vision of the field and separate themselves from the real robots, by concentrating too much on the simulations instead of real robots. One way to solve this problem without incurring into too much costs is to combine simulations with real robots.

Modern simulators work based on a hardware abstraction layer (HAL) [10]. A HAL is a framework for a robot in which the commands sent to the motors or sensors of the robot do not differentiate between the real robot or the simulated robot. Usually, the HAL is provided by the company that builds the robot. Sometimes, the robotics community creates this HAL for interesting robots that do not come with it.

Other times, when a robot does not provide such HAL, the simulator may provide instead a cross compilation facility [11]. This is the case for example of some of the robots provided by Webots simulator. In the case, for instance of Aibo robot, it does not have such abstraction layer due to its low CPU computer and being an old robot.

Webots provides a mechanism that allows the translation of the code you use in the simulation of Aibo robot to code to be executed in the real robot. The result is that, the same things that the robot does in the simulation with the user program will be done by the real robot when using the cross-compilated code [12].

Having this framework available, either through HAL or cross-compilation, a new teaching environment that optimises costs and maximizes students experience can be devised. We call this environment the Optimized Robotics Teaching Environment (ORTE). It works as follows:

1. The teacher selects the real robot to use and buys it. Only one or two units of the real robot may be necessary (depending on the amount of students per class and the resources available).
2. The teacher creates a simulation of the robot which mimics the way the real robot will be used for the class. It is very likely that a simulation for that robot and environment already exists in the net, hence it is suggested to perform a search for it. Getting an already made model of the robot can save many days of work.
3. The teacher distributes the simulation among students and specifies what it is required to do (for instance, make the robot recognize an orange, make it walk up to the table, etc.).
4. Students create their control programs for the simulated robot in order to accomplish the assigned task.
5. Once a student has achieved a program in the simulation that performs the task, she can transfer her program to the real robot making use of the HAL or the cross-compilation, and check if it works the same way in the real robot. Chances are not. Hence the student returns back to simulation to analyze why not and devise a new strategy.
6. Keep repeating fourth and fifth points until fifth is accomplished.

One example of successful use of the ORTE is the MIT course MIT 2.166 of 2016 on autonomous vehicles (class named Duckietown [13]), where students have to build their real robots using arduino boards and use the simulation of those robots in The Construct to build and test control programs (Figs. 5 and 6 show the real and simulated Duckiebot model and real and simulated environment).

## 6 Conclusion

Simulations are a powerful tool for teaching robotics, since allow the creation of cheap robotics environments, and access to the latest (simulated) robotics technologies at a very low cost.

Simulations can be simplified for teachers and students by making use of web based simulators. Web based simulators universalize the access to such software because allow anybody use them with any device with a fraction of the cost of the desktop simulator.

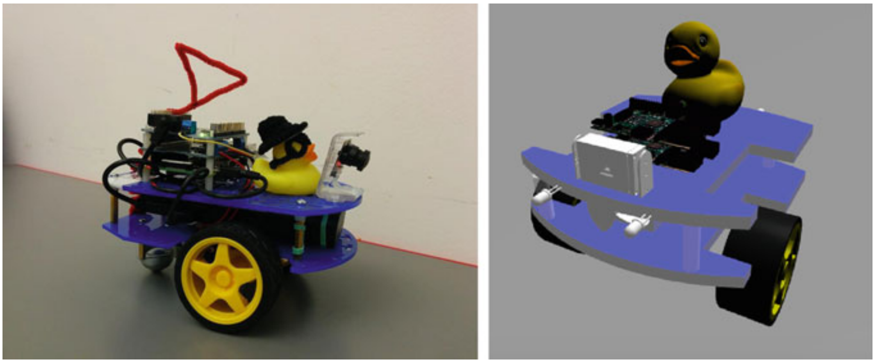

**Fig. 5** Real Duckiebot and its simulation model

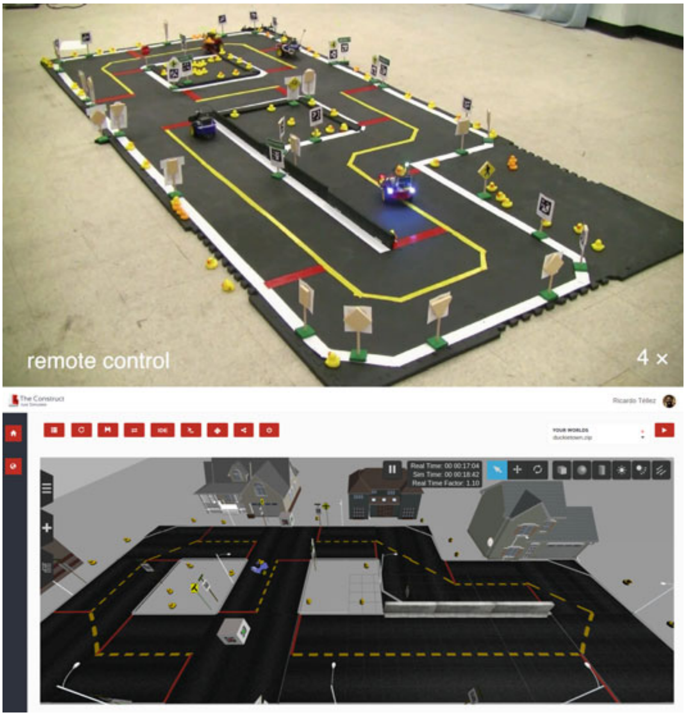

**Fig. 6** Real Duckietown environment and simulated environment using The Construct

Finally, simulations can be integrated in a complete learning system that includes real robots in the path, minimizing costs and maximizing usage and students experience (ORTE method).

**Acknowledgments** We would like to thank all the Duckietown team for making it possible and specially Andrea Censi for making us part of the project.

## References

1. Silicon Valley Robotics: Service robotics case studies in silicon valley. Tech Report (2015)
2. Poppy humanoid robot. http://www.poppy-project.org
3. Robotics operating system. http://www.ros.org
4. Simulating Husky robot: the easy way. http://www.theconstructsim.com/simulating-husky-robot-the-easy-way
5. Simulating Tiago robot: the easy way. http://www.theconstructsim.com/simulating-tiago-robot-the-easy-way
6. Michel, O.: Webots: professional mobile robot simulation. J. Adv. Robot. Syst. (2004)
7. Koenig, N., Hsu, J.: The many faces of simulation: Use cases for a general purpose simulator. IEEE International Conference on Robotics and Automation (2013)
8. Hsu, J.M., Peters, S.C.: Extending open dynamics engine for the DARPA virtual robotics challenge. In: 4th International Conference, SIMPAR (2014)
9. Agero, C.E., Koenig, N., Chen, I., Boyer, H., Peters, S., Hsu, J., Gerkey, B., Paepcke, S., Rivero, J.L., Manzo, J., Krotkov, E., Pratt, G.: Inside the virtual robotics challenge: simulating real-time robotic disaster response. IEEE Trans. Autom. Sci. Eng. **12**(2) (2015)
10. Jorg, S., Tully, J., Albu-Scheffer, A.: The hardware abstraction layer—supporting control design by tackling the complexity of humanoid robot hardware. IEEE International Conference on Robotics and Automation (2014)
11. Michel, O., Rohrer, F.: The Rats Life Benchmark: Competing Cognitive Robots RSS (2008)
12. Holh, L., Tllez, R., Michel, O., Ijspeert, A.: Aibo and Webots: simulation, wireless remote control and controller transfer. Robot. Auton. Syst. **54**(6), 472–485 (2006)
13. Duckietown MIT 2.16. http://duckietown.mit.edu/