

V-REP and LabVIEW in the Service of Education

Marek Gawryszewski, Piotr Kmieciak and Grzegorz Granosik

Abstract The following paper exposes an effective approach of teaching robotics by applying very efficient and popular set of tools. It demonstrates a combination of applications such as V-REP and LabVIEW in order to be later utilized for rapid prototyping, algorithm design and revival of both simple as well as fairly advanced robotic environments. The described teaching methodology has been successfully applied to high-school students during their facultative robotics courses which have been taking place at the Lodz University of Technology. Detailed summary of the results of this exertion is also provided.

Keywords Educational robotics · V-REP · LabVIEW · Simulation environment · Rapid prototyping

1 Introduction

According to various statistics [1, 2], robotics market has been exponentially growing for the last few years and nothing indicates that this could change in the nearest future. Becoming more and more specialized, robots take over a significant number of tasks performed by the human so far. Particular growth can be seen not only in the industrial process automation area, but also, and especially, in consumer robotics.

Even though the prophecies of robots taking over most people's jobs still seem to be a fantasy, the transition is clearly visible.

M. Gawryszewski (✉) · P. Kmieciak · G. Granosik
Institute of Automatic Control, Lodz University of Technology,
B. Stefanowskiego 18/22, 90-924, Lodz, Poland
e-mail: marek.gawryszewski@dokt.p.lodz.pl
URL: <http://robotyka.p.lodz.pl>

P. Kmieciak
e-mail: piotr.kmieciak@asero.pl

G. Granosik
e-mail: granosik@p.lodz.pl

At the moment, this means that we have to get used to the presence of robotic devices in our lives and learn to cooperate with them. Even though some of the job opportunities have already expired, a huge amount of new ones is just opening. However, the transformation we are witnessing must enforce the change in the way we perceive robots and in the direction in which potential workforce should be trained. The conclusion is that robotics should be introduced to the vast majority of students at possibly early level of education—to get them accustomed to the new technologies and in the way that is not overwhelming—to let them develop interest in it.

In the following paragraphs we provide an evaluation of two powerful pieces of software, which when combined together, have a rich potential to significantly facilitate the learning process and may also be very useful in further development. Subsequently, we include a set of instructions of how this can be achieved and eventually we deliver the results of our attempt.

2 Related Work

The demand for having flexible training tools in order to meet the needs of the emerging robotics market and the potential of virtual reality has been recognized quite a long time ago. In [3], KUKA's universal simulation environment, targeted especially for educational purposes, has been presented in contrast to the other, device-specific modelling software available at that time. Another virtual visualization environment, which includes three robotic models and is suitable for teaching robotics at the university level has been presented in [4]. A more contemporary approach, based on Gazebo simulator has been demonstrated in [5]. The summary of the usage of similar tools can be found in [6, 7], where the authors share their experiences gained during the several years of educational activities. The reported results seem to be satisfactory enough to still take on this subject.

3 Motivation and Background

As an interdisciplinary field of science, robotics requires from its adepts to acquire an extremely wide range of knowledge in many differed areas in order to be able to freely express their creativity. It is also worth mentioning, that robotic systems are usually fairly complex, susceptible to damage or even dangerous when handled improperly. They are rather expensive as well, which can often result in a limited access to such devices. Those are only a few reasons why to look for many alternative ways of targeting the challenges thrown by the world of modern robotics. A short description of the tools of our choice is presented as follows.

3.1 V-REP

V-REP (*Virtual Robot Experimentation Platform*) is a versatile, cross-platform, 3D robotic simulation environment made by Coppelia Robotics [8, 9]. It can be used for modelling simple robotic components such as sensors and actuators as well as complex fully functional robotic systems.

The application provides forward and inverse kinematics calculations along with the four real-time physics engines dedicated for dynamic calculations and simulation of full objects interactions.

It supports various interchangeable CAD formats such as URDF, COLLADA, DXF, 3DS, OBJ, STL for easier integration and can be bridged with various external applications through several remote API's (C/C++, Python, Java, Matlab, Octave, Lua, Urbi) and the ROS interface. With its well designed, user friendly and customizable interface it is a very good choice for presenting the basic principles of robotics as well as designing and testing any robotic device.

3.2 LabVIEW

LabVIEW is a widely spread engineering platform from National Instruments [10]. It comes with a visual programming language named *G*, which basically means that the program structure consists of a number of graphical blocks representing individual instructions, all connected with wires to ensure a suitable work-flow. Such an approach guarantees even inexperienced users, unfamiliar with the programming principles to efficiently learn and operate through the environment. The application supports a wide range of data acquisition and instrument control devices which makes it a great tool for working with the real hardware.

4 Course Description

The main objective of the classes is to enable a group of high school students to design, simulate and build a factual mobile robot: a Mars rover. By working on this project, its participants familiarize themselves with a wide range of aspects of the development process: from making the first design choices to the final testing of a working machine.

As a part of the introduction, various instruments were presented to them as the possible ways to solve the whole task: modelling tools, simulators, programming environments, prototype boards and many others. In this article we focus on the two aspects of the whole work: simulations and algorithms development.

One of the most important things in the whole process is the fact, that the course is led by example and try. Students were provided with knowledge of how to begin.

The rest is all about achieving their goals on their own with only a minimal assistance and supervision coming from a teacher.

4.1 The Role of Selected Software

Harnessing V-REP and LabVIEW enables students to benefit from various, well-established tools like inverse kinematics solver right out of the box. This means, that they do not have to understand deeply all aspects of the given problem, e.g. inverse kinematics. Considering less advanced high school attendees this might be crucial to preserve motivation, as some problems might be worked around without decreasing overall complexity of a project.

Up till now, V-REP has been used to demonstrate, how to model and simulate a robot. The application does not only allow to conduct simulations of a designed device, but also of the entire surrounding environment.

During the hands-on sessions, students have been creating models of mobile robots to understand the possibilities and limitations of the whole process. Thanks to built-in modules, they were able to easily simulate dynamics of their creations and plan a path to follow, which included taking into account existing obstacles.

Two methods of implementing robot's algorithms were demonstrated. The first one was scripting in Lua language [11], which is an internal mechanism of V-REP.

Additionally, the originators of V-REP have established APIs that allow users to create applications of any kind, which then are able to communicate with the simulation and control its elements. Thanks to that, it is now possible to create control algorithm in any modern programming language, but C++, Python, Java, Matlab and LabVIEW are the preferred ones.

As for the second method, we have chosen V-REP's LabVIEW interface, which was originally designed by Peter Mačička [12]. Utilizing his work, we asked the students to create their own programs to control simulated robots.

4.2 Modelling Basics

Modelling with V-REP involves dealing with three basic groups of elements [13]:

1. *Scene objects*—twelve types of construction blocks (e.g. shapes, joints, proximity, force/torque or vision sensors), that can be combined with each other to create more complex designs
2. *Calculation modules*—five basic embedded algorithms: forward and inverse kinematics, path planning, collision detection, physics and dynamics, minimum distance calculation

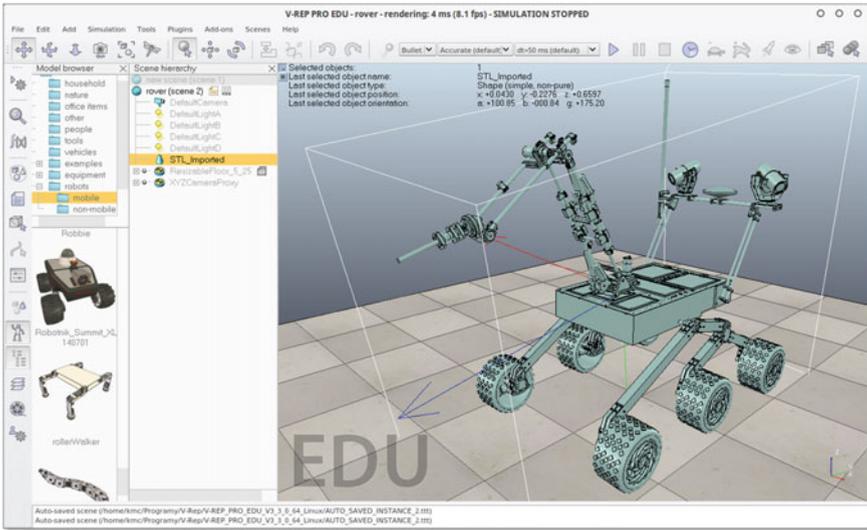


Fig. 1 V-REP’s default user interface running under Linux operating system (with a model of an authentic Mars rover, built by the students at the Lodz University of Technology [14], imported to the scene)

3. *Control mechanisms*—local and remote interfaces like: embedded scripts, plugins, add-ons, remote APIs, ROS nodes and custom (client/server model based) solutions, that can work simultaneously and cooperate with each other

The main program interface (Fig. 1) is clear and simple. By default, it consists of standard toolbars offering the ability to access most of the features. A scene pane which can display multiple views as well as custom user interfaces is available on the right. On the left, there is a browser with thumbnails of all available models that can be dragged into the scene and finally a tree view called *Scene hierarchy*, which represents dependencies between all objects of the project.

When all elements are finally placed on the scene, they can be easily controlled via the internal scripting language or provided remote APIs. To illustrate the functionality more accurately, we can imagine creating a rotary joint on the scene and then setting its velocity by referencing it via the Lua script.

Lua is described as very fast, powerful, yet simple and lightweight scripting language ideal for rapid prototyping. In addition to procedural syntax, it provides meta-mechanisms for implementing elements typical for object-oriented programming languages, like classes and inheritance [15].

There are several types of embedded scripts that can be applied to the project. The most important are those used when the simulation is running which divide into four categories: *Main*, *Child*, *Joint Control* and *Contact Callbacks*. Then we can also use the *Customization*, *General callback* and finally *Dialog* and *Editor* scripts. Below we

provide a short example of a *Child* script, which demonstrates referencing objects in V-REP.

Lua script—referencing simulation objects in V-REP

```
-- Handle of the left motor
leftMotor=simGetObjectHandle('`bubbleRob_leftMotor`')
-- Handle of the right motor
rightMotor=simGetObjectHandle('`bubbleRob_rightMotor`')

speed=minMaxSpeed[1]+(minMaxSpeed[2]-minMaxSpeed[1])

if (backUntilTime<simGetSimulationTime()) then
    -- When in forward mode,
    -- move forward at the desired speed
    simSetJointTargetVelocity(leftMotor,speed)
    simSetJointTargetVelocity(rightMotor,speed)
else
    -- When in backward mode,
    -- backup in a curve at reduced speed
    simSetJointTargetVelocity(leftMotor,-speed/2)
    simSetJointTargetVelocity(rightMotor,-speed/8)
end
```

(based on: Line following BubbleRob tutorial)

4.3 *Interfacing V-REP with LabVIEW*

There are many aspects in which the embedded scripting excels. The ease of the integration, inherent scalability, robustness and compatibility are just only a few of them. However, considering the general purpose of the course, we decided to present the remote APIs as well. Among others, LabVIEW seemed to be the best compromise between the number of available features and intuitiveness.

The interfacing is extremely simple and poses absolutely no problem. On the server side (in our case this means V-REP simulation), we just have to edit the internal script to set the available port, include a proper plug-in called *v_repExtRemoteApi.dll* (the name may vary depending on the platform) and enable remote APIs by calling *simExtRemoteApiStart()*. From the clients perspective, we use a universal *Call Library Function Node* (indicated by arrow in Fig. 2). We just have to make sure, that proper Dynamic-Link Library (*remoteApi.dll*) is set in the block properties and select the adequate parameters (indicated by arrow in Fig. 3). A short tutorial with a wide range of examples has been provided by the author of the interface [12].

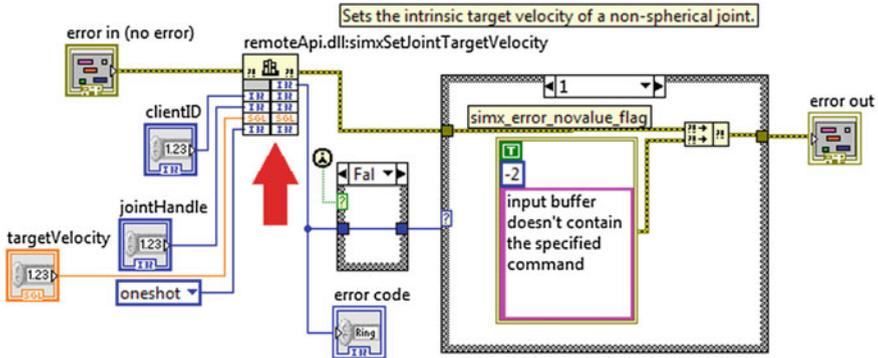


Fig. 2 SubVI handling joint velocity

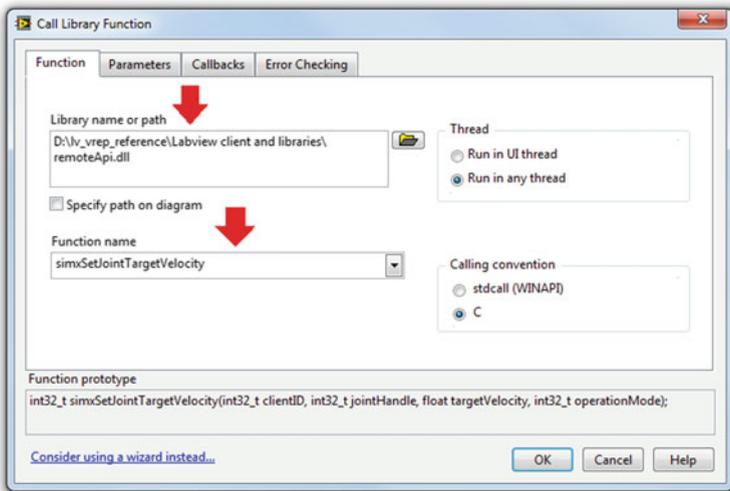


Fig. 3 Call Library Function Node properties

4.4 Workflow and Educational Values

At the very beginning of the course a modelling software has been presented to students. It is also worth mentioning, that in addition to V-REP, the presentation included Autodesk Inventor, which will later be used to create their own models.

The next step was the introduction to LabVIEW. The lecture included environment basics such as: the distinction between graphs and front panels, basic data types, kinds of controls, indicators and constants as well as execution structures i.e. Loops and Cases. Participants learned how to create simple programs, according to paradigms and patterns used in LabVIEW.

The following four meetings, which are the main scope of this article, were all about modelling in V-REP and programming in LabVIEW.

During the first one, students were shown the capabilities of V-REP, the basic concepts, and how to build simulated objects and elements of the environment. They gained knowledge about its distributed control architecture and the advantages guaranteed by this kind of solution (e.g. portability and scalability). Finally, all of them learned how to associate scripts with individual objects. The possibility of importing prototypes created in the above mentioned Autodesk Inventor has also been demonstrated as shown in Fig. 1.

The second and the third meeting were focused on modelling the BubbleRob robot, which is a part of the official V-REP’s tutorial [16]. All the students, divided into the groups of two members each, had to go through the whole process of creating a simulation of a simple mobile robot step-by-step, which definitely allowed them to improve their practical skills. As a result, they were able to see for themselves, how robotic models are being designed and how to resolve typical problems.

During the fourth meeting, two previously introduced tools were connected together: we used LabVIEW to create control application for the robot simulated in V-REP. This has happened in the two stages. The first one was to set up an example control application, which shows how to exchange data in both directions. The difference between all available interfaces has been outlined emphasizing regular and remote APIs. The second one involved students in creating their own programs, to control robot in V-REP (Fig. 4).

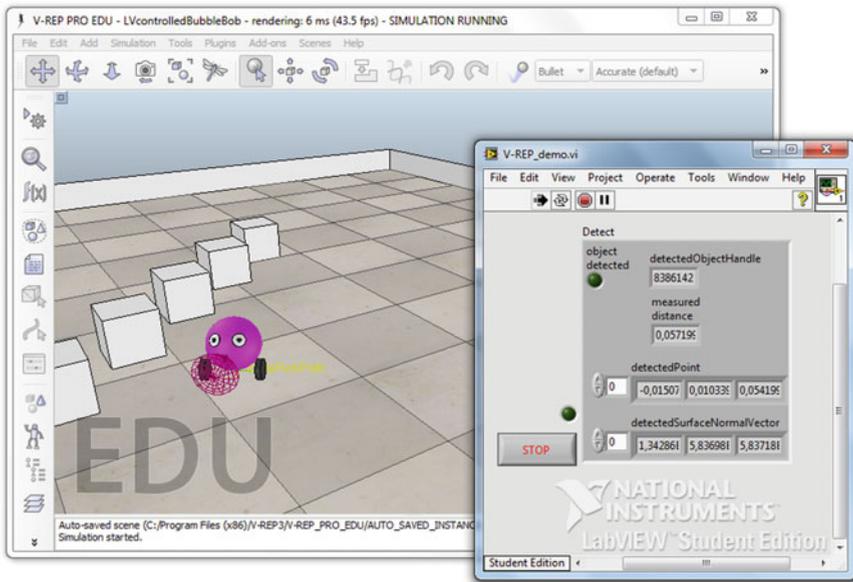


Fig. 4 “BubbleRob” robot (back window), controlled by LabVIEW application (front window)

What is important, as LabVIEW was selected as an environment to create algorithms for the real robot, the students were able to test their programs with the simulation and use it in a real-world application afterwards. They also had a chance to exploit a variety of *Toolkits* available for LabVIEW, including those dedicated especially to robotic applications.

Given the knowledge, that the only thing that differs is a low level communication layer (i.e. function calls) and that it should be replaced by the one appropriate for the selected hardware, they were finally enabled to use the same program with both, the simulated and the actual robot.

The next step of our course is to show the basics of electronics with Arduino [17], which serves as a controller of robots intended for the course. After it is finished, students will be ready to start making real robots, as they gain enough skills to know how to start and they have already seen on simulation that their models can work.

4.5 Survey

A couple of weeks after the last meeting focused on LabVIEW and V-REP, we asked our students to provide feedback about the workshops. Our goal was to understand, the students' opinion about the form of the classes—if they are satisfied with it and the provided content as well as if they noticed any increase in the level of their knowledge.

Eleven participants completed the questionnaire. There were 9 statements (collected in Table 1) describing the workshop and the responses were formatted in a 5-point Likert scale: answer 1 means: I strongly disagree and 5: I strongly agree.

In general, our work as tutors has been well evaluated, as all the students indicated that tutors were helpful and willing to give explanations if something was unclear. They also agreed that the knowledge was presented in an explicit and understandable manner as shown in Fig. 5, plot: “Questions 1, 2”.

On the other hand, the presented content did not fit their needs perfectly, as 4 students marked answer 3 which might be considered as a slightly negative opinion (Fig. 5, plot: “Questions 3, 4”).

For the vast majority of the students our workshops were interesting (9 out of 11) and most of them declared to take part in more advanced courses, as an extension of this one (Fig. 6, plot: “Questions 5, 6, 7”).

We also asked about the programming skills, and 9 participants claimed that they have improved their skills as a result of the course. Unfortunately, the other two disagreed with this statement.

The question about the further use of the acquired knowledge seems to be unresolved: there are 6 students which claim that they see benefits from the workshop as a way to apply their knowledge in their own projects (Fig. 6, plot: “Question 8, 9”), yet still 5 people has no clear idea or do not see any profit.

Table 1 Questions

No.	Topic	Question
1	Tutors	Tutors were willing to answer questions
2		Knowledge was presented in an explicit and understandable manner
3	Materials	Course materials used were useful
4		Course materials were easy to understand
5	Course form	Course was interesting
6		Course form allows me to take part actively
7		My programming skills raised through participation in the course
8	Future	I would like to take part in more advanced edition of the course in the future
9		Knowledge gained during course will allow me to implement my own ideas

This was the first iteration of workshops and this survey showed us some space for improvements. We indicated three crucial areas which should be reworked before the next edition:

- *the content*—the course was based on step-by-step, written instructions, which were generally considered as useful, but there is some space for refinement such as dividing the tutorial into smaller parts or introducing better balanced level of difficulty.
- *programming skills*—part that introduces LabVIEW shall be reworked to another form.
- *students' ideas*—the main goal of the entire project is to build a Mars rover. It is also very important to create a strong the base to develop students' own ideas on basis of this project, and this needs to be emphasized.

5 Final Remarks

The most important aim of our project was to introduce fairly complex robotic problems to young, inexperienced people and additionally solve these problems in an

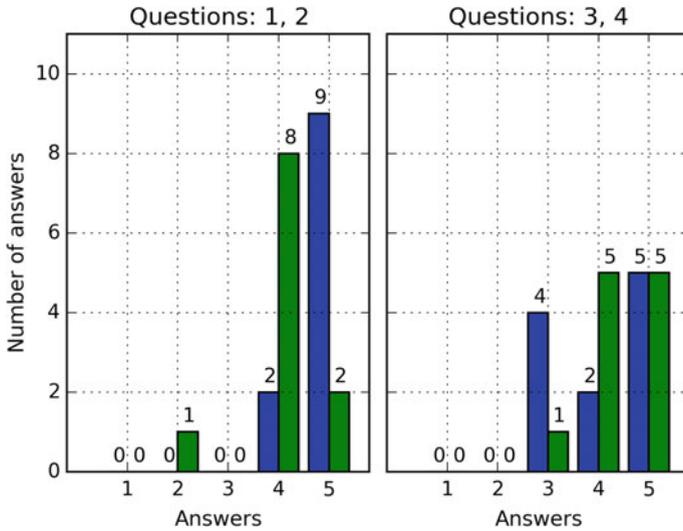


Fig. 5 Histogram of answers the the questions: 1—“Tutors were willing to answer questions.” (left plot, blue), 2—“Knowledge was presented in an explicit and understandable manner.” (left plot, green), 3—“Course materials used were useful.” (right plot, blue), 4—“Course materials were easy to understand.” (right plot, green)

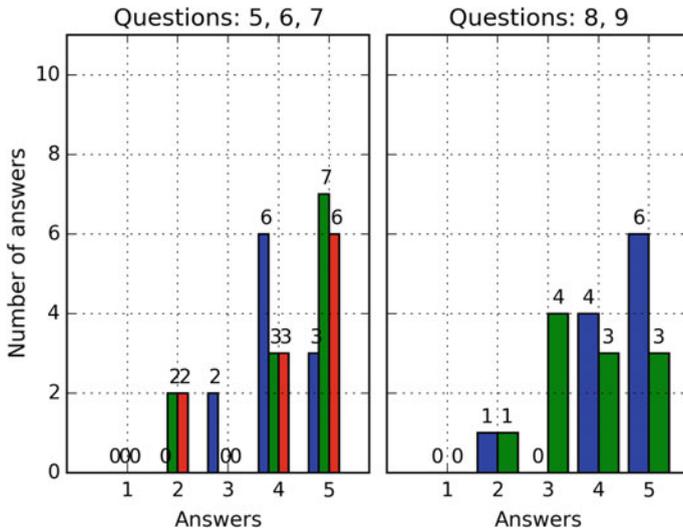


Fig. 6 Histogram of answers the the questions: 5—“Course was interesting.” (left plot, blue), 6—“Course form allows me to take part actively.” (left plot, green), 7—“My programming skills raised through participation in the course.” (left plot, red), 8—“I would like to take part in more advanced edition of the course in the future.” (right plot, blue), 9—“Knowledge gained during course will allow me to implemnet my own ideas.” (right plot, green)

interesting manner with the available (possibly free) tools. These issues are present in most projects dealing with kids and teenagers [18, 19].

Among several simulation environments like Webots [20] or Gazebo [21] we have chosen V-REP as the most engaging and accessible. There are at least five great premises in favour of this decision:

1. *Usability*—representation of the scene is done in a similar way as in any other modelling software, which students are already familiar with way before they get introduced to V-REP
2. *Efficiency*—the vast majority of work can be done by simply using drag-and-drop functionality which considerably improves productivity
3. *Scalability and portability*—a great variety of ways to add logic to the scene: by internal scripts or external applications over standardized communication channels
4. *Compatibility*—possibility to import/export objects in various CAD formats, so the elements in simulation can be based on ones created in modelling software
5. *Simulation capabilities*—it is important to see how the mass and inertia of the robot influences its ability to move or perform other tasks and V-REP provides four different models of dynamics, all of them available free of charge for educational purposes

Our goal was to deliver tools and methods that are fun, easy to use and can be utilized with limited training, as our final goal is to build a robot, not to learn new programming environments.

We believe, that our approach will finally succeed, as participants of the course are now able to build simulations and control algorithms for their own robots with a very limited supervision.

We also hope, that our experience will be helpful for other tutors that may face similar issues.

Acknowledgments The authors would like to thank Mr Kacper Andrzejczak for his effort during the lessons.

References

1. Industrial Robotics Market by Type (Articulated, Cartesian, SCARA, Cylindrical, Parallel), Application (Automotive, Electrical and Electronics, and Metal and Machinery), Component (Controller, Sensors, Drive), and Geography—Analysis & Forecast to 2022. <http://marketsandmarkets.com>, <http://www.marketsandmarkets.com/Market-Reports/Industrial-Robotics-Market-643.html>
2. Grand View Research: Service robotics market analysis by application (Professional, Personal) and segment forecasts To 2020. <http://www.grandviewresearch.com/industry-analysis/service-robotics-industry>
3. Sett, A., Vollmann, K.: Computer based robot training in a virtual environment. In: 2002 IEEE International Conference on Industrial Technology, Bangkok, Thailand (2002)

4. Yang, X., Zhao, Y., Wu, W., Wang, H.: Virtual reality based robotics learning system. In: Proceedings of the IEEE International Conference on Automation and Logistics, Qingdao, China, Sept 2008
5. Canas, J.M., Martn, L., Vega, J.: Innovating in robotics education with Gazebo simulator and JdeRobot framework. XII Congreso Universitario de Innovacion Educativa en las Enseñanzas Tecnicas (2014)
6. Lopez-Nicolas, G., Romeo, A., Guerrero, J.J.: Simulation tools for active learning in robot control and programming. In: EAEEIE Annual Conference (2009)
7. Jung, S.: Experiences in developing an experimental robotics course program for undergraduate education. *IEEE Trans. Educ.* **56**(1) (2013)
8. V-REP project website. <http://www.coppeliarobotics.com>
9. Rohmer, E., Singh, S.P.N., Freese, M.: V-REP: a versatile and scalable robot simulation framework. In: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on Robots and Systems (2013)
10. LabVIEW product website. <http://www.ni.com/labview>
11. Lua project website. <http://www.lua.org>
12. LabVIEW interface for V-REP. <http://www.coppeliarobotics.com/contributions/labview.zip>
13. V-REP's overview presentation. <http://www.coppeliarobotics.com/v-repOverviewPresentation.pdf>
14. Raptors (a Mars Rover project) website. <http://raptors.p.lodz.pl/>
15. Lerasalimschy, R., de Figueiredo, L.H., Celes Filho, W.: Lua an extensible extension language. *Softw. Pract. Exp.* **26**(6), 635–652 (1996)
16. BubbleRob tutorial. <http://www.coppeliarobotics.com/helpFiles/en/bubbleRobTutorial.htm>
17. Arduino Mega 2560 product website. <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>
18. Petrovic, P.: Having Fun with Learning Robots. In: Proceedings of the 3rd International Conference on Robotics in Education, RiE2012. September 2012, pp. 105–112. MatfyzPress, Czech Republic (2012)
19. Swenson, J., Danahy, E.: Examining influences on the evolution of design ideas in a first-year robotics project. In: Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education, Padova (Italy), 18 Jul 2014
20. Webots project website. <https://www.cyberbotics.com/overview>
21. Gazebo project website. <http://wiki.ros.org/gazebo>



<http://www.springer.com/978-3-319-42974-8>

Robotics in Education

Research and Practices for Robotics in STEM Education

Merdan, M.; Lepuschitz, W.; Koppensteiner, G.; Balogh,
R. (Eds.)

2017, XI, 289 p. 106 illus., 96 illus. in color., Softcover

ISBN: 978-3-319-42974-8