iLabit Robotica: A Photo-Realistic Extension for Robotics Toolbox

Ivan Kholodilin1,2, Nikita Savosteenko2, Yuri Abramov2 and Luis A Lago1
1 Beijing Institute of Technology, Beijing 100081, China
2 South Ural State University, Chelyabinsk 454080, Russia

E-mail: kholodilin@bit.edu.cn

Abstract. This paper presents a photo-realistic extension for Robotics Toolbox (iLabit Robotica). Built using Unity, the proposed simulator integrates the 6-axis robot arm, sensors, and elements for interactions. iLabit Robotica is aimed to increase the quality of the engineering education and motivation of students towards the educational process, by providing a new beginning for the project-based learning (PBL) approach with the aid of virtual technologies. We strongly believe that by simulating real technological processes and integrating them into the educational process, students will not only achieve a valuable practical experience by solving “real-life” existing problems, but also a substantial interest towards the subject. We demonstrate the compatibility between our simulator and Robotics Toolbox by interacting with objects placed inside the virtual environment through Matlab via TCP/IP. Experiments demonstrate that the operating process is simple and it provides robust results and a higher measurement accuracy. The proposed simulator and supporting materials are available online: http://www.ilabit.org.

1. Introduction

Specialized software products capable of providing three-dimensional visualization, e.g. a technological process or high-precision mathematical modelling such as Robot Studio, SOLIDWORKS, and ANSYS, were developed and widely used at the end of the last century. This kind of software is implemented in almost all segments of the engineering industry, ranging from the defence industry, heavy industry, and aerospace engineering to microelectronics, medicine, and simulators. Modelling and analysis in several industries may avoid costly and time-consuming design-build-test cycles. As a result, the widespread use of different types of software in industry, educational institutions started to integrate these software products into the educational process. Results showed that teaching courses based on specialized software products had a significant impact on the level of knowledge acquisition, competencies, and even motivation of science and engineering students [1]. The research on the quality of education received by students in higher educational institutions shows that the introduction of specialized software had a significant impact on the level of education and competencies of graduates, especially in engineering-related specialities [2].

Nowadays one of the fastest growing areas in engineering is industrial robotics, namely robotic manipulators. Robot manipulators are complex and expensive technical objects that have found wide applications in medical [3], and industrial fields [4]. In order to work with robot manipulators specific programming skills are required, understanding of the technological process based on these robots, as well as deep knowledge of the theory of electric drives and automation. However, sometimes it can be
difficult or even impossible to recreate real technological processes in a classroom environment. Moreover, in many cases the resources used as a necessary means to successfully carry out some of these activities are very expensive, e.g., industrial robots. At the same time, project-based learning (PBL) approach is an important tool that can be successfully applied in a variety of scenarios. It has proven to be effective in different situations, while increasing the interest of students towards them not only in higher education [5], but in the middle school education as well [6]. Therefore, the finding of new interesting and cost-effective ways to implement PBL in engineering education remains as a relevant topic. Secondly, during the 2020 year we have witnessed how the educational process has been interrupted to some extent. Engineering students have required laboratories to test their theories and conduct practical experiments and they have lost this possibility given this difficult situation. Therefore, new ideas should be explored in online education. Last but not least, for college graduates it can take a long time to get accustomed to the industrial processes, which, at the same time, can be very time- and cost-consuming for employers. Consequently, new ways of interactions are required on early stages (before graduation from universities), aimed at preparing students to the “real life” in a variety of working environments. It is worthwhile to note that it would be better if these new methods do not distract students from the main study.

By taking the aforementioned problems into account, we can state that the issue of developing a specialized virtual laboratory, which will provide the possibility to carry out a wide range of experiments, is of particular importance today. One of the possible existing solutions is the one developed by ABB “Robot Studio” software product, which not only supports a wide range of scenarios, but also allows to interact with industrial robots, and at the same time, immerse the user inside the technological process through realistic three-dimensional graphics. The main disadvantage of this software is that users do not have the ability to entirely interact with an industrial robot, namely with electric drives belonging to the corresponding joints of the robot. This drawback can be eliminated with the use of the “Robotics Toolbox” software package for MATLAB [7]. The Robotics Toolbox is a software package that enables MATLAB user to readily create and manipulate datatypes fundamental to robotics such as homogeneous transformations, quaternions, and trajectories. Besides the programming algorithms for manipulating with the robot manipulator, this Toolbox allows to use mathematical models of electric drives. The limitation of this program is the lack of realistic graphics and interaction with objects. However, by using modern game engine platforms, e.g. Unity, it is possible not only to visualize the technological process, but also to integrate physics into it. In the view of the current problems and trends in engineering education, this paper proposes a photo-realistic virtual laboratory, which includes a robot manipulator fully compatible with the Robotics Toolbox.

2. Simulator overview

There are existing robotics toolboxes [7-11] which enable students and teachers to better understand the theoretical concepts behind classical robotics and computer vision through easy and intuitive simulation and visualization. It is worthwhile mentioning that the most popular one was developed by P. Corke and called “Robotics Toolbox” [7]. The toolbox is based on a very general method of representing the kinematics and dynamics of serial-link manipulators using objects. Each link is represented by a link object whose attributes are the standard or modified Denavit and Hartenberg (DH) parameters, as well as the link and motor inertial properties, friction, and gear ratio. Functions in this toolbox are written in a textbook manner, which helps to better understand theory by practice. However, the Robotics Toolbox has particular limitations as well. For example, Matlab environment is not really suitable for visualization tasks as it suffers from the photorealism. In contrast, more flexibility towards visualization can be provided by game engines such as Unity, UNIGINE, CRYENGINE, and Unreal Engine 4. Moreover, it was found earlier that virtual technologies are more attractive for students than traditional educational materials [2]. By taking this information into consideration we decided to provide new insights towards engineering education by creating an extension for the Robotics Toolbox. Our simulator is built by Unity, the main screen of the simulator is presented in Fig. 1. Main parts of the scene consist of the industrial robot and the conveyor with
colored objects. The manipulator includes the camera attached to the end effector (EE); this camera can be activated/hidden by the corresponding button located on the panel. Next to the camera button is located the button that activates the coordinates of the joints of the manipulator. This information can be useful for describing the manipulator in terms of DH parameters [12] or modified DH parameters [13].

![Figure 1](image.png)

**Figure 1.** Main screen of the proposed simulator. The left blocks of the menu are in charge of the interaction with the manipulator and objects on the conveyer; the right block shows the streaming of the camera attached to the end effector.

The proposed simulator can be simply installed and configured on Windows, macOS, and Linux operating systems. The simulator provides three main features: (1) interaction with scenes and objects included into the virtual environment, (2) communication with other programs by TCP/IP, (3) a detailed user guide and technical support. All these features allow to configure experimental conditions for testing theories and improving operating algorithms of the system. During the development of the simulator, a particular attention was dedicated to creating photorealistic scenes, by using rendering capabilities such as light sources, reflection from the surfaces, shadows, and so on. Fig. 1 shows a snapshot taken by our simulator which reveals rendering capabilities. The simulator supports two modes: manual mode and through communication with other programs by TCP/IP (see Table 1).

| Interaction with the robot | Interaction with the objects* | Activating new objects |
|----------------------------|-------------------------------|------------------------|
| Manual mode from the user interface (UI) of the simulator | Communication with other programs by TCP/IP | Interaction with the robot |
| The angles of the joints can be controlled by sliders or input fields (see Fig. 1) | The angles of the joints can be sent through TCP/IP separately, e.g. forward kinematics or in a loop, e.g. inverse kinematics. | The distance between the EE and the object can be controlled from zero (for precise operating tasks) to bigger values (when high precision operating is not required). For example, if this distance is set up equal to zero and there is any misalignment between the EE and the object, consequently, in this particular case, it is not possible to grab the object. |
| EE can grab and release objects by toggle “Grab”, generating “Green Cube”, “Red Cube”, “Random Cube” or removing Cube from the conveyer with the corresponding buttons (see Fig. 1). | EE can grab and release objects by discrete parameter “Grab”, generating “Green Cube”, “Red Cube” or “Random Cube” or removing Cube from the conveyer with the corresponding combination of bits. | |

* The distance between the EE and the object can be controlled from zero (for precise operating tasks) to bigger values (when high precision operating is not required). For example, if this distance is set up equal to zero and there is any misalignment between the EE and the object, consequently, in this particular case, it is not possible to grab the object.
3. Experiment
We strongly believe that the proposed simulator is able to bring new insights towards robotics course by interacting with objects and modern graphics. Thus, the main goal of the experiment was to investigate the compatibility between the Robotics Toolbox in Matlab and our simulator. It was decided to demonstrate this compatibility by means of practical tasks, namely by sorting six objects (three green and three red) with the known coordinates of them and joints of the robot manipulator. The simulation model of the robot was based on the real parameters of the manipulator Motoman MH50II (see Fig. 2). More details about the experiment are provided below.

Figure 2. Simulation of the manipulator. From left to right: parameters of the manipulator, model built by Robotics Toolbox, model inside the simulator.

3.1. Experiment setup
The proposed simulator is able to interact with objects included into the virtual environment, such as objects on the conveyor and pallets. Consequently, in order to operate with these objects through Matlab it was necessary to provide an understanding of their corresponding positions and sizes. Once the information about the objects is entered into the simulator it can be verified by carrying out the experiment with Matlab. This experiment was based on sorting the green and red objects to the corresponding pallets from the initial position on the conveyer with the coordinates (1.733; 0.000; 0.369). Red objects were sorted on to the centre of the right pallet. Calculations were based on given coordinates for the center of the right pallet (0.000; 1.350; -0.400), and the height of the objects was set at 0.158 meters. Green objects were sorted onto the left pallet in a more complicated way, namely to the corner. Calculations were based on given coordinates for the center of the right pallet (0.015; 1.220), and the size of objects was set at (0.400; -0.400; 0.158). The above-mentioned calculations are depicted for the first objects in Fig. 3, the remaining objects should be placed above the corresponding green and red objects.

Figure 3. Calculating coordinates of objects placed onto pallets.
Finally, once all of the coordinates computed, the calculated Robotics Toolbox function “ikine” was used, which calculates inverse kinematics based on the given points and function “jtraj” calculating trajectory between given points. Angles of every joint were transmitted to the simulator through TCP/IP. Results are presented in the next subsection.

3.2. Results
Visual results of the experiment are shown in Fig. 4. From this figure it can be seen that the objects take a desired location without any visual misalignments between objects. More accurate comparison analysis is presented in Table 2. From the values in Table 2 it can be seen that the coordinates of the experiment are similar to the coordinates from the experiment setup, the difference between them is very small, in the range of a few millimetres.

![Figure 4. Results of the experiment.](image)

| Objects | No. | Calculated coordinates, meters | Experiment coordinates, meters | Absolute error, meters |
|---------|-----|--------------------------------|--------------------------------|------------------------|
| Red     | 1st | (0.000; 1.350; -0.242)         | (0.000; 1.350; -0.244)         | (0.000; 0.000; 0.002)   |
|         | 2nd | (0.000; 1.350; -0.84)          | (0.000; 1.351; -0.83)          | (0.000; 0.001; 0.001)   |
|         | 3rd | (0.000; 1.350; 0.740)          | (0.005; 1.350; 0.780)          | (0.005; 0.000; 0.002)   |
| Green   | 1st | (0.452; -1.042; -0.202)        | (0.460; -1.041; -0.201)        | (0.008; 0.001; 0.001)   |
|         | 2nd | (0.452; -1.042; 0.202)         | (0.456; -1.042; 0.197)         | (0.004; 0.000; 0.005)   |
|         | 3rd | (0.452; -1.042; 0.602)         | (0.458; -1.041; 0.595)         | (0.006; 0.001; 0.007)   |

4. Conclusion
The paper has demonstrated a publicly available extension for the Robotics Toolbox in MATLAB, the principle features, and its capabilities. The software product presented in this article showed the possibility of implementing virtual laboratories for studying robot manipulators, whereas the experiment results demonstrated its full compatibility with the Robotics Toolbox. By simulating real technological processes, it was possible to determine new ways for solving the problems defined earlier. We strongly believe that virtual technologies will not only influence on the motivation of students, but at the same time, it will save economic resources for educational centers, as virtual complexes are substantially less costly and the resources involved in the maintenance process are less demanding than those required by the real ones. The proposed simulator provides an opportunity to
conduct experiments from any place, only a computer is required. Thus, even in a difficult situation students are able to combine theory with practice, which is of particular importance to each of them in their training as future engineers.

The literature review related with the process of creating virtual laboratories showed that this direction is young and is just emerging. In fact, in order to facilitate teaching and learning of the fundamental concepts in the subject, pedagogy of teaching the subject should take into consideration the latest technological tools and move with the times. An example of how the robotics course can be improved was shown in this paper. In the future, we aim to further work on the improvements of the simulator by including new scenarios and also consider new directions in which virtual laboratories can take place.

Acknowledgments
The work was supported by Foundation for Assistance to Small Innovative Enterprises, contract No. 477GUCES8-D3/61968 from 05.10.2020.

References
[1] Rodrigues C S and Rocha A M 2014 Analysing Students' Attitudes Towards the Learning of Specialized Software Proc. Int. Conf. on Computational Science and Its Applications (Guimarães) 264-267
[2] Bazarov S E, Kholidin I Y, Nesterov A S and Sokhina A V 2017 Applying Augmented Reality in practical classes for engineering students Proc. of the IOP Conf. Ser.: Earth Environ. Sci. 87
[3] Kucuk S and Gungor B D 2016 Inverse kinematics solution of a new hybrid robot manipulator proposed for medical purposes Medical Technologies National Congress (Antalya) p 1-4
[4] Zeng Q, Liu X, Qiu C and Li A 2019 Kinematic Characteristics Analysis of Cooperative Welding Robot with Multiple Manipulators 2019 Proc. Int. Conf. on Intelligent Robot Systems (Nagoya, Japan) p 1-5
[5] Mones A, Sanchez E, Dimitriadis Y A, Abellan I M, Avi B and Gorgojo G 2005 Multiple Case Studies to Enhance Project-Based Learning in a Computer Architecture Course J IEEE Transactions on Education 48 482-489
[6] Timothy D W 2017 Teach Like Finland (W.W. Norton, Incorporated)
[7] Corke P I 1996 A Robotics Toolbox for Matlab IEEE Robotics and Automation Magazine.
[8] Vila-Rosado D N and Dominguez-Lopez J A 2005 A Matlab toolbox for robotic manipulators Proc. Int. Conf. on Computer Science (Puebla, Mexico) p 256-263
[9] Righettini P, Strada R, Olama E K and Valilou S 2015 Symbolic kinematic and dynamic modelling toolbox for Multi-DOF robotic manipulators Proc. Int. Conf. on Automation and Computing (Glasgow) p 1-7
[10] Elshatarat H L, Baniyounis M and Biesenbach R 2016 Design and implementation of a RoBO-2L MATLAB toolbox for a motion control of a robotic manipulator Proc. Int. Conf. on Systems, Signals & Devices (Leipzig) p 89-95
[11] Liu J and Luo Q 2019 Modeling and Simulation of Robotic Arm in MATLAB for Industrial Applications Proc. Int. Conf. on Intelligent Human-Machine Systems and Cybernetics (Hangzhou, China) p 346-349
[12] Paul R P 1981 Robot Manipulators: Mathematics, Programming, and Control (Cambridge, Massachusetts: MIT Press)
[13] Craig J 1986 Introduction to Robotics (Addison Wesley)