APPLICATION OF SIMULATION MODELS FOR PROGRAMMING OF ROBOTS

Beata Jakubiec
Czestochowa University of Technology, Faculty of Electrical Engineering, Poland

Abstract. *In recent years, there has been an increasing robotization of many areas of life. This requires knowledge of issues related to robots and their programming. Devices of this type, especially large industrial or medical robots, are very expensive, or they may be hardly affordable for educational purposes. Simulation models are helpful in such situation. The aim of the article is to present the possibilities of using simulation models of robots and robotic stations in the educational and research process. Specialized software packages make it easier not only to understand how robots work, but they also allow to test different operating conditions of the designed computer model of the real robotized process.*

Keywords: *programming, robot, simulation model.*

Introduction

In the last several years, a significant increase in robotization can be observed in many areas of life. Robots are used primarily in various branches of industry. Due to the benefits of their implementation in the industrial process, such as precision, repeatability, speed of operation, reliability and the ability to work in difficult conditions, they are being installed in plants not only by large corporations, but more and more often by medium and small manufacturing enterprises. These include, for example, palletizing, welding, assembly, painting and gluing robots. However, the range of robot applications is much wider. We can find them, among others, in medicine (medical telemanipulators, rehabilitation robots), uniformed services (mobile inspection robots, unmanned military vehicles), but also in everyday life as intelligent vacuum cleaners, lawnmowers, or even interactive toys. Such a dynamic development of robotics requires the inclusion of issues related to robots and their programming in the education of students, especially of technical faculties, to a lesser or greater extent, depending on their field of study. In the case of industrial robots, their programming is traditionally carried out using Teach Pendant. It is a device equipped with a screen and buttons connected to the robot controller. With its help, the operator can remotely control robot drives, change parameters and, primarily, program the sequence of robot movements using the appropriate commands (i.e. programming language). In each of these cases, however, it is

http://dx.doi.org/10.17770/sie2018vol1.3214

necessary to remove (exclude) the robot from the original process and to pay special attention when programming the robot movement, so as not to enter incorrect settings that could result in a malfunction of the robot. Moreover, devices of this type, especially large industrial or medical robots, are quite expensive or their availability for educational purposes is limited. This is where the simulation programs, in which the actual robotized system can be modeled, are helpful (Figure 1).

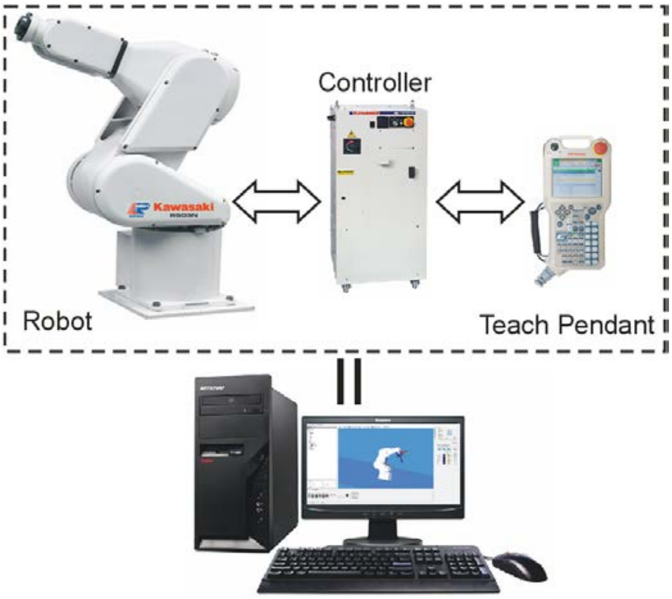


Fig. 1. **Robot workstation and computer with simulation software**

The aim of the article is presenting the purpose of using simulation models of robots and robotic stations in the design, teaching and research process. Specialized software packages make it not only easier to understand how robots work. They allow to prepare off-line control program for the analyzed robotized station, as well as to test various working conditions of the designed model of the actual process. In most cases, on-line feedback between the software and the robot is also possible. The usefulness of computer modeling and simulation, has already been confirmed in the teaching and research process in other engineering fields, and not just, as presented in publications, e.g. (Anderson, 2014; Grabust, 2017; Olesiak, 2017; Prauzner, 2016; Ptak, 2017).

Proprietary software of robot manufacturers

Currently, virtually every industrial robot supplier offers the possibility of programming and testing their robots off-line (i.e., outside of the production environment). Dedicated computer programs are used for this purpose. The offered environments are designed to create virtual 3D models of robotic stations and to simulate their operation. Virtual robot programming environments are a great engineering tool used to modernize or design new robotic stations, including planning the robot motion trajectory, creating control programs using programming languages compatible with programming languages of real robots (also using virtual Teach Pendant), as well as optimizing the production process. Engineering staff as well as learners of robotics have the ability to quickly create a robotic station model on a PC and analyze its operation. Three-dimensional models of production cells are built using the included libraries with models of robots and their equipment (e.g. grippers, paint nozzles), and objects surrounding the robot (e.g. conveyors, tables, platforms, fences). Components of a robotic station can also be designed by the user himself/herself or imported from CAD environments. A list of selected manufacturers of robots, programming languages and programming and simulation environments is presented in Table 1.

Table 1 **Selected robot manufacturers, off-line programming softwares and robot programming languages**

Robot Brands	**Software**	**Programming Language**
Comau	RoboSim Pro	PDL2
Fanuc	RoboGuide	TP/Karel
Yaskawa-Motoman	MotoSim	Inform
ABB	ABB RobotStudio	RAPID
Kawasaki	K-Roset	AS
Nachi Fujikoshi	FD On Desk	Slim
Denso	WINCAPS III	PAC
Kuka	Kuka.Sim Pro	KRL
Mitsubishi	MELFA WORKS	MELFA BASIC

From the point of view of people learning how to use, program and create robotic production cells, access to libraries with sample workstation models, offered in some simulation packages, is helpful in the education. Figures 2 and 3 show examples of virtual robotic stations made using Kawasaki and Kuka software. A useful function offered in the software is also visualizing the displacement of the effector by drawing a trajectory of motion in the form of points or lines (Figure 2 and Figure 4).

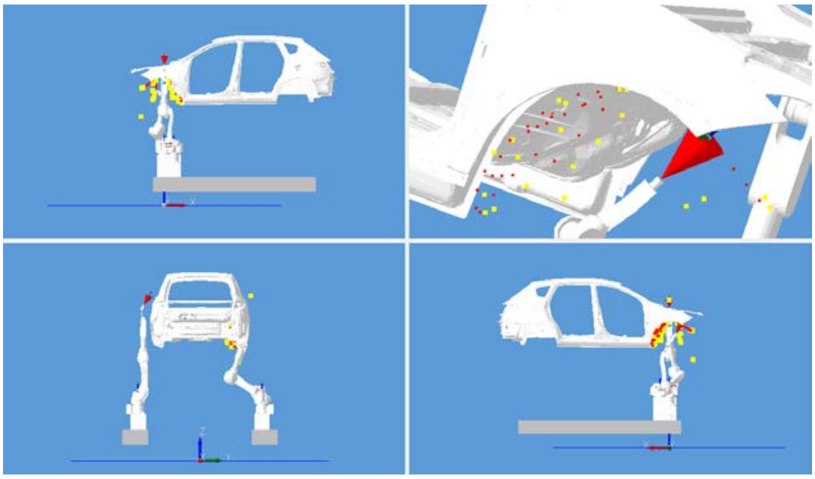


Fig. 2. **Virtual model of workstation realized in K-Roset** (Kawasaki, 2014)

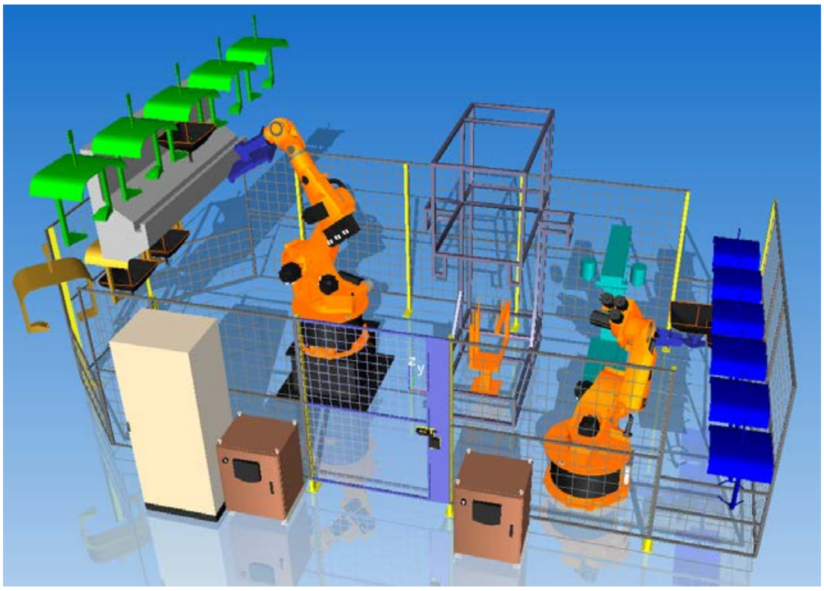


Fig. 3. **Virtual model of workstation realized in Kuka. Sim Pro** (KUKA, 2015)

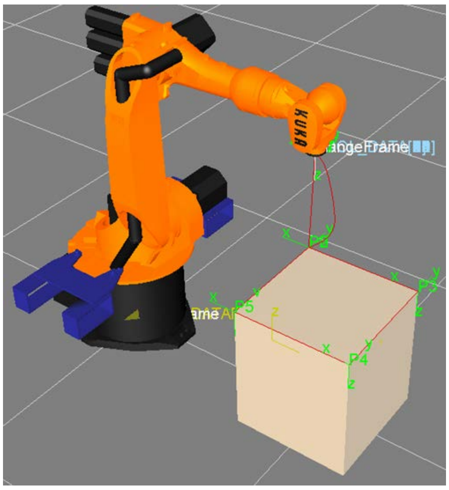
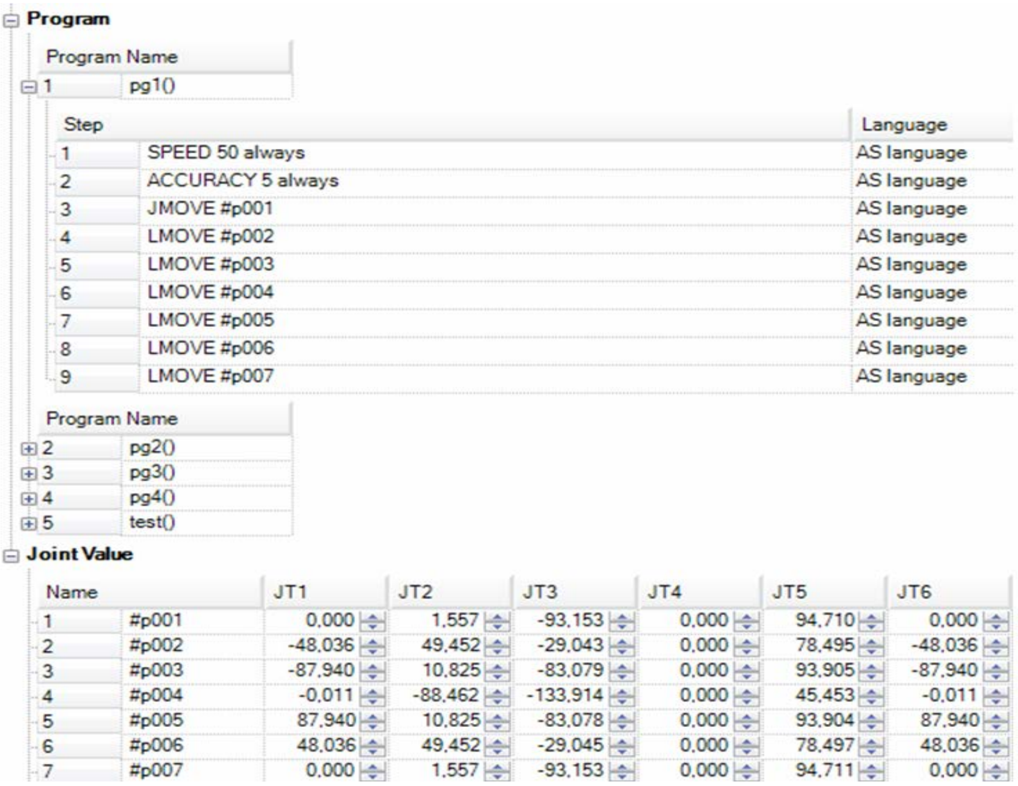


Fig. 4. **Visualization of trajectory in KUKA. Sim Pro**

Simulation programs have functions, either built-in or obtained by installing additional packages, thanks to which it is possible to reproduce the actual robotic manufacturing station, e.g. [Comau, 2018; Kawasaki, 2014; KUKA, 2015; Fanuc, 2016]:

- measurement of work cycles corresponding to the actual controller (controller emulator),
- collision detection,
- reproduction of the actual kinematics and dynamics of the robot,
- robot limit zones,
- tracking of conveyors and sensors,
- robot load identification.

Simulation tests of the analyzed system can be carried out not only virtually but also in real time. Thanks to this, after a robot program corresponding to the user's requirements is designed, it can be transferred to a physical robot controller, with the certainty of working properly in real production conditions. Figure 5 presents an example of a program for a virtual robot written in the K-Roset environment.



⊟ **Program**

Program Name	
⊟ 1	pg1()

Step		Language
1	SPEED 50 always	AS language
2	ACCURACY 5 always	AS language
3	JMOVE #p001	AS language
4	LMOVE #p002	AS language
5	LMOVE #p003	AS language
6	LMOVE #p004	AS language
7	LMOVE #p005	AS language
8	LMOVE #p006	AS language
9	LMOVE #p007	AS language

Program Name	
⊞ 2	pg2()
⊞ 3	pg3()
⊞ 4	pg4()
⊞ 5	test()

⊟ **Joint Value**

Name		JT1	JT2	JT3	JT4	JT5	JT6
1	#p001	0,000	1,557	-93,153	0,000	94,710	0,000
2	#p002	-48,036	49,452	-29,043	0,000	78,495	-48,036
3	#p003	-87,940	10,825	-83,079	0,000	93,905	-87,940
4	#p004	-0,011	-88,462	-133,914	0,000	45,453	-0,011
5	#p005	87,940	10,825	-83,078	0,000	93,904	87,940
6	#p006	48,036	49,452	-29,045	0,000	78,497	48,036
7	#p007	0,000	1,557	-93,153	0,000	94,711	0,000

Fig. 5. **Robot program written in AS Language developed with K-Roset**

Universal robot simulation and off-line programming environments

The second group of software are universal programs for modeling, simulating and analyzing the behavior of robots. Table 2 presents a list of several of them. Their advantage is that the user can build and test virtual models with robots of different manufacturers in the same environment. Some programs have libraries, which include not only industrial manipulators, but also driving, walking or flying robots – Figure 6. Ponadto z programów CAD mogą być importowane autorskie modele i komponenty. In addition, proprietary models and components can be imported from CAD programs. Unlike dedicated simulators, the virtual model is controlled using a script written in one of the languages, e.g. C/C++, LUA, Matlab, Octave, Python, depending on the environment. For programming industrial robots, prepared off-line programs, thanks to post processors implemented in virtual environments, are compiled to the basic robot programming languages. Thus, they are output into the form that is ready to be uploaded, thanks to the right drivers, to the controllers of real robots (Coppelia, 2016; Corke, 2016; Cyberbotics, 2018; MathWorks, 2017; RoboDK, 2017).

Table 2 **Exemplary of robot simulation and offline programming softwares**

Developers	**Software**	**Programming Language**
Coppelia Robotics	V-Rep	LUA
Cyberbotics	Webots	C++
MathWorks/ Peter Corke	Robotic Systems Toolbox/ Robotics Toolbox	MATLAB scripting language/ Octave GNU
OpenRAVE Community	OpenRAVE	C++, Python
Open Source Robotics Foundation	Gazebo	C++
RoboDK	RoboDK	Python

An interesting proposal is RoboDK, a software for off-line programming and simulation of industrial robots that is free to use for educational purposes. Its advantage is a large number of models of robots from various brands (currently over 200), detection of singularities and collisions, as well as being able to observe the position and orientation values of the tool and manipulator units during the simulation – Figures 7.

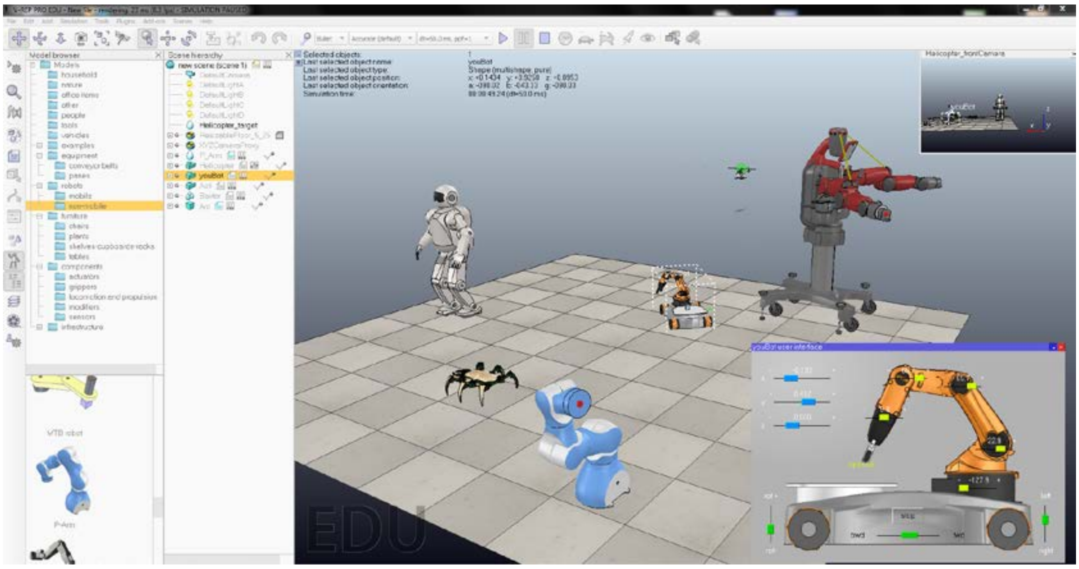
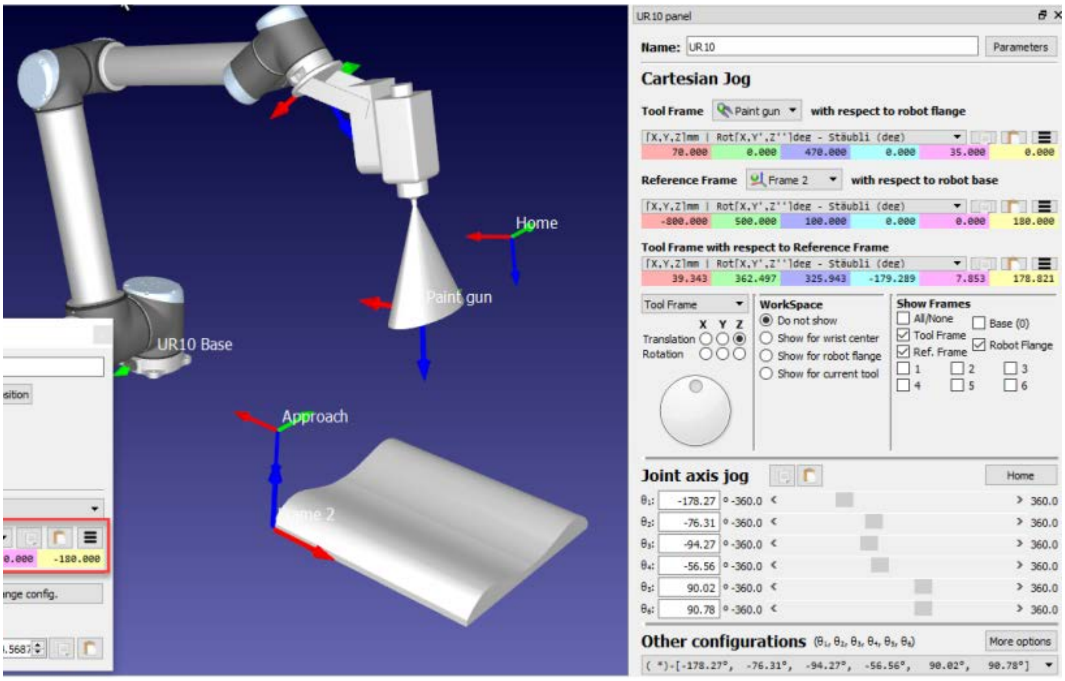


Fig. 6. **Simulation of robots in V-Rep software** (Coppelia, 2016)



UR 10 panel

Name: UR 10 | Parameters

Cartesian Jog

Tool Frame Paint gun **with respect to robot flange**

[X,Y,Z]mm | Rot[X,Y',Z'']deg - Stäubli (deg)

70.000	0.000	470.000	0.000	35.000	0.000

Reference Frame Frame 2 **with respect to robot base**

[X,Y,Z]mm | Rot[X,Y',Z'']deg - Stäubli (deg)

-800.000	500.000	100.000	0.000	0.000	180.000

Tool Frame with respect to Reference Frame

[X,Y,Z]mm | Rot[X,Y',Z'']deg - Stäubli (deg)

39.343	362.497	325.943	-179.289	7.853	178.821

Tool Frame

	X	Y	Z
Translation	○	○	●
Rotation	○	○	○

WorkSpace
● Do not show
○ Show for wrist center
○ Show for robot flange
○ Show for current tool

Show Frames
□ All/None □ Base (0)
☑ Tool Frame ☑ Robot Flange
☑ Ref. Frame
□ 1 □ 2 □ 3
□ 4 □ 5 □ 6

Joint axis jog Home

θ1:	-178.27	° -360.0	360.0
θ2:	-76.31	° -360.0	360.0
θ3:	-94.27	° -360.0	360.0
θ4:	-56.56	° -360.0	360.0
θ5:	90.02	° -360.0	360.0
θ6:	90.78	° -360.0	360.0

Other configurations (θ1, θ2, θ3, θ4, θ5, θ6) More options

(*)-[-178.27°, -76.31°, -94.27°, -56.56°, 90.02°, 90.78°]

Fig. 7. **Simulation of industrial robot in RoboDK software** (RoboDK, 2017)

An interesting program, more in terms of research and science, is the Matlab software with two robotics toolboxes. The proprietary Robotic Systems Toolbox contains, among others, functions that enable conversion between various forms of rotation and translation, path planning algorithms, path tracking, map representations. It is designed for creating applications of autonomous robotic ground vehicles. It features communication interfaces for Matlab/Simulink and robots that support ROS or process visualization in graphic robot simulators, e.g. Gazebo (MathWorks, 2017). Robotics Toolbox developed by Peter Corke is a tool with functions that show the mathematical foundations of robotics in the field of kinematics, dynamics and trajectory generation, including different versions of Denavit-Hartenberg parameters, reverse kinematics, motion equations, creation of trajectories in Cartesian space (Corke, 2016).

Educational robotics

Students, teenagers and children can also get in touch with the problems of robotics through play and entertainment. LEGO Mindstorm and VEX Robotics sets are perfect for this. These are sets of so-called smart bricks, which are an excellent tool for modern education in STEM disciplines (Science, Technology, Engineering, Mathematics). It allows building didactic robots, e.g. driving or walking robots, or programmable automatic machines. An interesting opportunity for learning how to create robotic systems using LEGO bricks is the Virtual Robotics Toolkit for LEGO Mindstorms simulation software (Cogmation, 2018). An example of a physical robot and a computer LEGO model is presented in Figure 8. This environment allows designing a robot thanks to the 3D model import, e.g. from LEGO Digital Designer, and then its programming and simulation. Thanks to the simulation, you can check the operation of the model, and possibly improve the functionality of the robot-toy before it is physically built. What is important is that programming of the virtual robot, that is its virtual intelligent brick, is done in the same language as the LEGO Mindstorm (NXT-G, EV3, LabVIEW for LEGO).

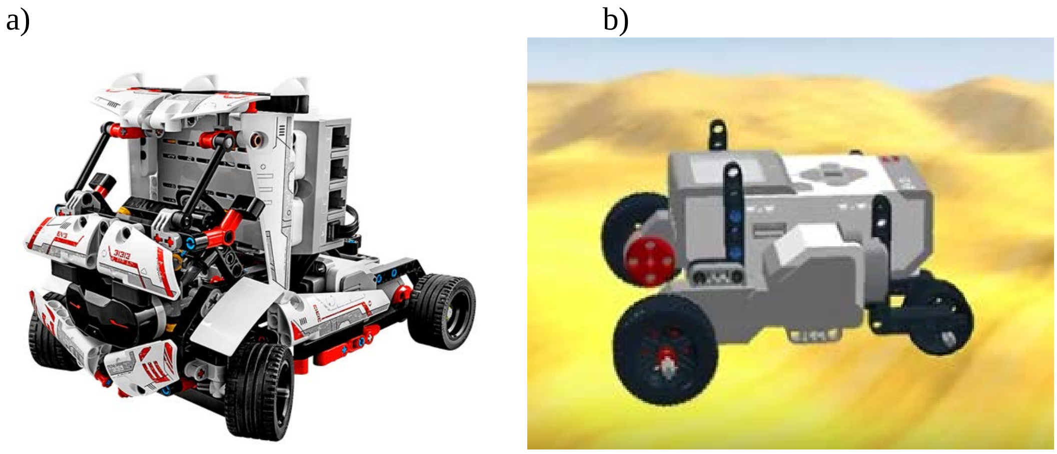


Fig. 8. **LEGO Robot: a) real construct RAC3 Truck** (LEGO, 2017),
b) virtual model realized in Virtual Robotics Toolkit (Cogmation, 2018)

Another example of a simulation environment created for educational sets is Robot Virtual Worlds. This software enables building, programming and testing virtual robots, not only LEGO, but also VEX – Figure 9. The programming of models is done in the ROBOTC language. It is a C-based programming language with a real-time debugger. It is interesting because it allows the young user, in an accessible and pleasant way, to gain the ability to use programming language, as used in professional programming tools or for programming industrial devices by IT professionals and engineers. Robot Virtual Worlds also offers the possibility of computer programming of LEGO models in their own languages (Robomatter, 2017).

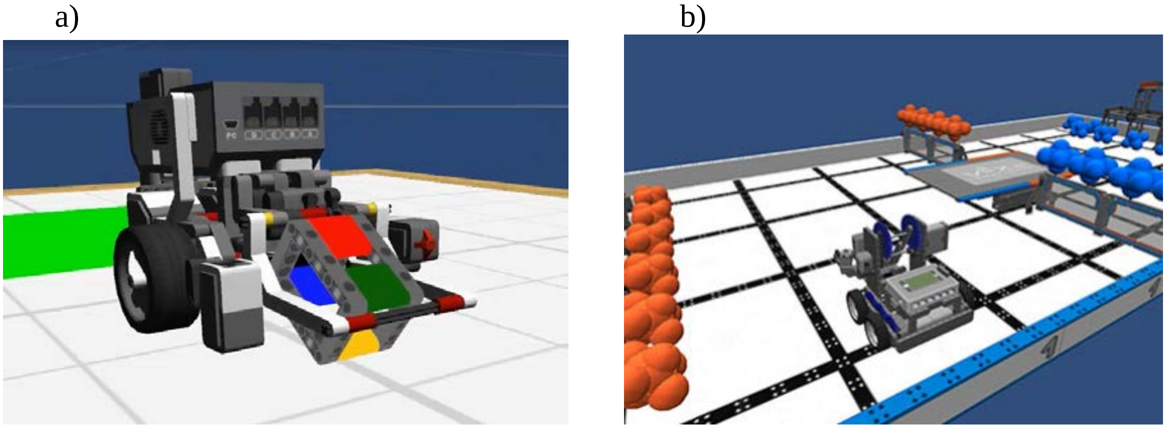


Fig. 9. **Robot Virtual Worlds software: a) LEGO robot virtual model**, **b) VEX robot virtual model** (Robomatter, 2017)

Conclusions

The high dynamics of robotics development affects the need to educate a larger number of students and specialists in this field. The use of virtual environments for programming of robots and simulation of their operation is ideal in the teaching and research process. It provides a safe way for conducting training on robot programming, learning the principles of designing robotic systems, even for inexperienced persons. Off-line programming, i.e. programming outside the production environment, eliminates the production downtime caused by the robot being taken out from the process. The functionality of the simulators allows for quick development of the production station model, optimization of the solution and enables to eliminate most errors occurring at the design stage and in practical implementation. The wide range of simulation environments also includes programs that enable conducting advanced scientific research, modeling and testing of complex robot control algorithms. An early technical education for children and adolescents through their contact with software for designing and simulating toy-robots is certainly a very good stimulus for logical thinking and it encourages further development of interests in this direction.

References

Anderson, K. (2014). *Design Energy Simulation for Architects: Guide to 3D Graphics*. Routledge.

Cogmation (2018). *Virtual Robotics Toolkit for LEGO Mindstorms* Retrieved from https://www.virtualroboticstoolkit.com

Comau (2018). *RoboSim Pro software* Retrieved from www.comau.com

Coppelia (2016). *V-REP PRO EDU* Retrieved from www.coppeliarobotics.com

Corke, P. (2016). *Robotics Toolbox* Retrieved from petercorke.com/wordpress/

Cyberbotics (2018). *Webots software* Retrieved from https://www.cyberbotics.com

Fanuc (2016). *ROBOGUIDE software* Retrieved from www.fanuc.eu

Grabusts, P. (2017). Simulation modelling possibilities in teaching economic processes. *Society, Integration, Education, Proceedings of the International Scientific Conference. Volume III, May 26th-27th 2017*, 463-471.

Robomatter (2017). *Robot Virtual Worlds software* Retrieved from www.robotvirtual worlds.com

Kawasaki (2014). *K-ROSET software* Retrieved from https://robotics.kawasaki.com

KUKA (2015). *KUKA.Sim Pro software* Retrieved from https://www.kuka.com

LEGO (2017). Retrieved from https://www.lego.com

MathWorks (2017). *Robotic Systems Toolbox* Retrieved from https://www.mathworks.com

Olesiak, K. (2017). Application of fuzzy logic toolbox for modelling fuzzy logic controllers. *Society, Integration, Education, Proceedings of the International Scientific Conference. Volume III, May 26th-27th 2017*, 539-546.

Prauzner, T. (2016). Interactive computer simulation as a response to contemporary problems of technical education. *Society. Integration. Education, Proceedings of the International Scientific Conference. Volume II, May 27th – 28th 201*6, 579-588.

Ptak, P. (2017). Application of computer programmes in research projects and teaching. *Society, Integration, Education, Proceedings of the International Scientific Conference. Volume III, May 26th-27th 2017*, 565-574.

Robo, D. K. (2017). *RoboDK software* Retrieved from https://robodk.com