

A NOVEL DESIGNED INTERACTIVE TRAINING PLATFORM FOR INDUSTRIAL ROBOT OFFLINE PROGRAMMING AND ROBOTICS EDUCATION

Cengiz Deniz* and Mustafa Çakır**

Abstract

The people interested in industrial robots at a beginner level, such as engineers, operators, or technicians, as well as robotics lecturers typically have difficulties during their first interactions with a robot. The first difficulty is to create a relation between the robot and the mathematical equations of its kinematics. One of the other barriers is to learn the robot workspace and the robot manipulator's reachability. After achieving these two steps, it possible to start robot programming and physical robot interaction appropriate for industrial robot applications. In this study, a novel interactive training platform is presented. This platform is designed for the offline programming of industrial robots and its goals are to support robot education, practical robot study, and industrial robot science. In this interactive robot platform, all commercial robot brands' manipulator geometry configuration and new manipulator geometry designs can be created. All functions such as programming, basic movement type (linear, joint, reorient), manipulator workspaces definition, tool centre point and axis velocity, acceleration are available. The platform presents some solid objects that can be inserted in the workspace and can make demonstrations for programming. There are also real programming logic controller and robot connections for offline programming. This platform is designed and developed based on open-source libraries, is small in size, and is able to work on all operating systems (*e.g.*, Windows, Mac, and Linux). This software may not be a satisfactory tool for robot specialists but is very effective for beginners in robot programming, such as students and robot operators in the plant.

Key Words

Interactive software, visualization in education, robot education, engineering education, robot workspace, robot cad

* Ford Otosan, Body Construction Engineering Department, Golcuk Plant, 41670, Golcuk, Kocaeli, Turkey; e-mail: cdenizden@gmail.com, cdeniz1@ford.com.tr

** Department of Electronics and Communication Engineering, Kocaeli University, Kocaeli, Turkey; e-mail: mcakir@kocaeli.edu.tr

1. Introduction

In engineering applications and education, physical equipment and training sets are crucial for a project's success. To achieve the targeted success, the related application system needs to be created and tested in a virtual software platform before their physical installation. In addition, these virtual training software tools are the key element in educational opportunities in case there is no physical training sets, such as in school, laboratory, or academy. These tools have also a big impact on trial and failure of activities to improve an application and to increase experience of engineers, students, and robot technicians. These software platforms itself are adequate for an experienced engineer who has practical and theoretical knowledge. However, beginners at the engineering education or vocation face a steep learning curve. All theoretical knowledge in engineering science, unlike social and medical science, can be tested as a practical application. But, it is very difficult to keep in mind the effects of mathematical calculations, parameters, and their changes. Thus, calculations must be shown on graphs. Animated illustrations instead of static graphics in the computer environment enable a more effective understanding of the subjects. In addition, if an address is defined as a sketch instead of a verbal description, it will make easier to reach the address. Visualization of clear expressions on computer screens makes the subject much easier to understand. Visually enhanced interactive software with integrated realistic models for robot kinematics makes it easier to understand of robotic science [1]. This type of the software allows the user to create their own manipulator designs.

In classical education, the Denavit–Hartenberg (DH) notation in the robotic science is one of the most difficult topics to be presented in classroom. Also, students cannot be expected to create 3D animations by combining the drawings on the plane. As presented in [1], the rate of response to questions in the visual material assisted education increased by up to 10%. This further emphasizes the importance of visually interactive educational software.

The interface presented in this study has been developed to investigate special topics in robotics. Robotics field is a combination of different disciplines such as math, mechanics, physics, control system, and circuits [2]. Also, it is stated that, in addition to traditional theoretical classroom activities, the use of interactive software has increased student motivation and increased the efficiency of the course. The most important thing that authors have achieved with the feedback they collect from the participants after the course is that the interactive software tools has to interact with the real robot. This requirement is presented in [3]. Virtual laboratories have indicated that the training efficiency has been increased and the inadequacies of the adjustable in basic robot parameters of the used software and hardware have been evaluated. In this publication, industrial robot software and commercial robot programming packages are discussed. However, while software libraries in the form of a Matlab toolbox are adequate for academic students, they are not useful for practitioners. This situation is presented in [4] as well. The software developed for master science students offers quite comprehensive features. The authors pointed out that robotic is an interdisciplinary science and training efficiency is enhanced by active learning and project-based work. They also state that this situation is valid for participants with a base knowledge of robotics.

The overall information in the literature is that the training efficiency of visual enhancement software is higher than classical classroom presentations. There are many software based on this aim. Most of these are related to specific subject of robotics. Some of them are designed for participants who have a mechanics, programming, and control know-how. The designed software should be simple to use, understandable, and should provide the least amount of information needed. To increase participant motivation and to strengthen their perception, visuality should be supported with real models and the software should be able to connect with real industrial applications.

In this study, we developed an industrial robotics human-machine interface with highly interactive monitoring capability that visualizes constraints that need to be considered during design and operation in robotic processes. The capabilities of the professional offline robot platforms are intended to be presented in this software. The workspace and arm volume visualization, joint kinematics tracking, original manipulator design, programming logic controller (PLC) supervision, and control developed in IRT are not present yet in any commercial software. We also recommend that these features be included in professional offline software packages.

2. Motivation

Today, desires and methods are changing rapidly. Existing software and tools should be flexible to these changes. To provide this flexibility, sometimes the modular structure is preferred and sometimes it can be provided by multi-purpose and reconfigurable tools. The widespread of flexible automation systems in industry has begun to reduce hardware costs and the importance of programming cost

and efficiency has been increased. Autonomy and heuristic approaches which are frequently mentioned in academic studies are not used, because reliability is based on industry. Conventional programming is preferred in industry. Robotics manufacturing is the primary solution when it comes to flexible automation. This result shows that in a growing need for personnel with a background in robotics and with deep robotic manufacturing knowledge. However, robot design requires high expertise and also it is a subject that can be achieved together by academics and engineers from different disciplines. Robot programming is a field in which college or undergraduate students could have experience with fast courses. In this study, we present the software that we use as trainers for robot programming education as well.

There are two complementary methods for robot programming. To reduce the installation time, the feasibility and accessibility studies of the robots are first done with offline CAD software. The robot program framework is created offline. During the installation, the robot operators perform position and trajectory control of the manipulator by the hand terminals. Today, the usage of offline programming software is widespread in industry. Different robot programming options have been compared in [5]. In this study, the easiest way for inline installation/teaching method for system integration has been described by researchers as costly and including high safety risk. It has been stated that the preferred robot programming methods rapidly exchanged to offline programming. As the generic software and the commercial software provided by the robot manufacturer reduce the field work, the advantages and disadvantages of these software are revealed in working according to brands and models. Robot Studio (ABB), KukaSim (KUKA), and Roboguide (FANUC) are some of these robot manufacturer's software and generic software ones such as Delmia, RobCad, and RoboMaster and Robsim can be presented as sample [1]–[5].

A real industrial robot and related conditions in the industrial environment are needed for the industrial robot lecture process about robot structure, programming, kinematics, dynamics, and working space in engineering education. However, it may not always be feasible to create field conditions of a real industrial environment or to physically have a robot in the laboratory. Even if a robot is provided, the subject of the robotics technique is limited to the robot's brand and model.

In this article, a novel industrial robot software called "IRT" (Industrial Robot Training Tool) is presented (see Fig. 1). This software includes a robot workspace, supports limits of working configuration, angle limits of each arm, basic robot movements (joint, linear, and circular), and programming. This software is also used as a visual material in engineering education. IRT can be downloaded from http://ehm.kocaeli.edu.tr/dersnotlari_data/?dir=mcakir/IRT.

The software is designed based on the C++ and the Fox toolkit which is open source. It also includes Libnodave for PLC communication and Ptgrey FlyCapture SDK for Stereo camera access. This software is platform independent and can also be compiled on the Linux

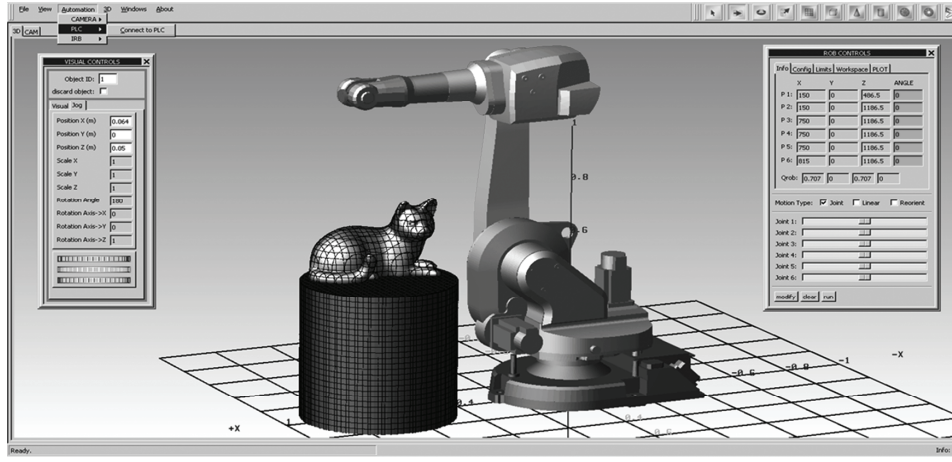


Figure 1. Industrial robot trainer (IRT).

environment. Robot kinematics basis are presented for a 6 DOF (degree of freedom) industrial robot in the next section. In Section 4, we introduce our program, which we use for our undergraduate students and also for operator trainings in support of on-the-job training in the manufacturing industry. Our findings and proposals were finalized at the end of the article, which we present with a sample application.

3. 6 DOF Articulated Industrial Robot Modelling

Robot kinematics science define the relationship between the position, velocity, and acceleration of the link arms and end effectors of the mechanical units. These mechanical units may have different geometric shapes and designs. In the robot kinematics (R Joints), the arms are assumed as a fixed body and the joints are assumed to perform the rotation [6]–[8]. The robot's forward and inverse kinematic models need to be formulated to find the relationship between the robot arms and end effector position and orientation in the workspace. Forward kinematics calculates the position and orientation of the robot end effector in the workspace using the axis angles, whereas inverse kinematics provides the angles of the axis of the robot using the position and orientation information of the end effector in the working space [9]–[13]. In the kinematic calculations of robot manipulators, there are three different methods: the iterative approach, the geometric model, and the analytical method. In the iterative method, the estimation method for each axis angle is used to send the robot tool centre point to the desired point. However, this process needs a long calculation time. In the geometric method, robot mechanics are analysed with trigonometric relations and the robot is modelled by basic geometry formulas. The model presented in this method is valid only for the relevant mechanical arm and must be re-modelled when the arm structure changes. In the analytical method, kinematic equations can be formulated independent of robot mechanical structure and kinematic solutions are provided dynamically for the required robot arm dimensions.

Generating the DH parameter table is the starting point for almost everyone when the forward and inverse kinematic models are needed. Homogeneous

transformation matrices are generated by using the tables [12], [13]. In the DH method, the most important point is to associate two coordinate systems with the minimum number of parameters. It is a complicated and difficult approach because of the minimum number of parameter concerns. At the same time, there are weaknesses in definition of rotations with homogeneous transformation matrices. In this study, quaternion algebra was used in the kinematic model to facilitate the students' understanding, to reduce the duration of the solution, and to achieve the result with fewer operations. Thanks to the Quaternion model, it is easy to extract the forward and inverse kinematic equations of the robot and the solution time is shorter. Quaternion was firstly founded in 1843 by Irish mathematician William Rowan Hamilton. A quaternion allows three-dimensional space rotation to be defined as a four-element vector [14]–[16]. The composite coordinate transformation with quaternion, which can be easily derived from the angular axis representation is shown in Fig. 2.

$$q = (\theta, \vec{N}) = (q_s, \vec{q}_v) = (q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k}) = (q_0, q_1, q_2, q_3) = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right) \cdot \frac{\vec{N}}{\|\vec{N}\|} \quad (1)$$

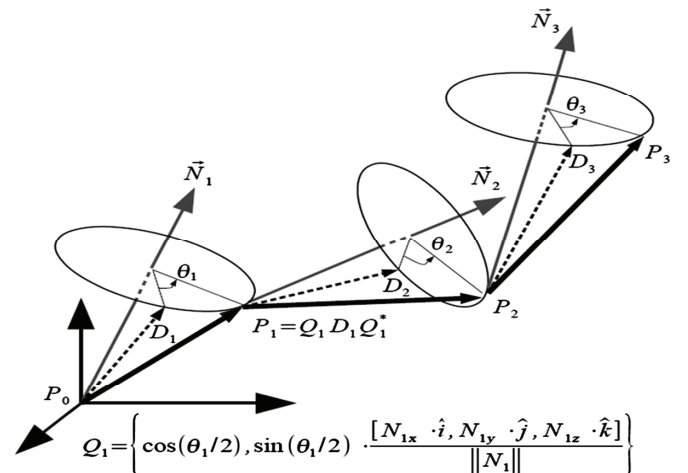


Figure 2. Link Coupling with quaternion (axis angle) representation.

Quaternions are defined by very different notations as can be represented by quaternion researchers (1). It is easy to define a quaternion when the pivot vector which consists of the rotation θ and the rotation axis \vec{N} data are known. The unit quaternion is used to indicate rotations. To calculate the composite rotation, the multiplication defined in (2) must be done. If we assume that the manipulator connections are vectors that can be defined by \vec{D} , then the unit quaternion q , the new position of \vec{D} at the end of the rotation is called as P and can be calculated by \vec{P} (3) equation.

$$q_1 * q_2 = (q_{1s}q_{2s} - \vec{q}_{1v} \cdot \vec{q}_{2v}) + (q_{1s}\vec{q}_{2v} + q_{2s}\vec{q}_{1v} + \vec{q}_{1v} \times \vec{q}_{2v}) \quad (2)$$

$$\vec{P} = q\vec{D}q^* = (q_s^2 - \vec{q}_v \cdot \vec{D})\vec{D} + 2q_s(\vec{q}_v \times \vec{D}) + 2\vec{q}_v(\vec{q}_v \cdot \vec{D}) \quad (3)$$

In our software, the ABB IRB 6650S 125/3.5m manipulator model was used to check the accuracy of kinematic calculations. In Fig. 3, it can be seen that the calculations give the same results as Robot Studio, which is the company's proprietary software. In Fig. 3, as it is shown in the technical drawing, the joint lengths \vec{D}_i and rotation axis \vec{N}_i are defined by determining pivot vector (4). It has been determined by our experience that the use of the notation (4) is more understandable than the DH table.

$$\begin{aligned} D_0 &= [0000 & 0000 & 0000] & N_0 &= [0 & 0 & 1] \\ D_1 &= [6000 & 0000 & 6300] & N_1 &= [0 & 0 & 1] \\ D_2 &= [0000 & 0000 & 1280] & N_2 &= [0 & 1 & 0] \\ D_3 &= [1592 & 0000 & 2000] & N_3 &= [0 & 1 & 0] \\ D_4 &= [0000 & 0000 & 0000] & N_4 &= [1 & 0 & 0] \\ D_5 &= [0000 & 0000 & 0000] & N_5 &= [0 & 1 & 0] \\ D_6 &= [200,0 & 0000 & 0000] & N_6 &= [1 & 0 & 0] \\ D_7 &= [0000 & 0000 & 0000] & N_7 &= [0 & 1 & 0] \end{aligned} \quad (4)$$

$$q_{(0)} = (\theta_{(0)}, N_{(0)}), \quad q_{(i)} = q_{(i-1)} * (\theta_{(i)}, N_{(i)}) \quad (5)$$

$$P_{(0)} = q_{(0)}D_{(0)}q_{(0)}^*, \quad P_{(i)} = q_{(i)}D_{(i)}q_{(i)}^* + P_{(i-1)} \quad (6)$$

The composite transformations of the joints are calculated by the rotation (5) which connects the base plane to the corresponding link coordinate plane. Finding the end point of each link (6) is now very easy. q_7 is used to move the tool coordinate system of this manipulator with 6 DOF to the same position as that given in the booklet. Inverse kinematic solution is too easy due to spherical wrist structure of the robot. After the end point found using \vec{D}_6 and \vec{P}_3 calculated for rotation, the first three angle of axis are found. The notations given by q_3 obtained by using these angles are processed and the angles of the wrist group are found. These calculations are done quickly and easily by using only four terms of the quaternions instead of 4×4 matrices which define the homogeneity coordinate transformation.

4. Industrial Robot Training Platform

The basic criteria we recommend to get the most efficiency of the software designed for robot training are as follows:

It should be easy to use: it should be available to students as well as to trainers and operators. It must require minimal background information. It should not be forgotten that training is necessary not only for the students but also for the beginners in the industry. *Access should be easy:* the support of commercial software should not be needed as much as possible. The mobility of the program must be high. The use of additional libraries should be avoided to achieve this. Providing open-source code is a great advantage for users. Commercial closed box software, hardware restrictions cannot be attached to this number. The requested change should be made easily by the trainees/students.

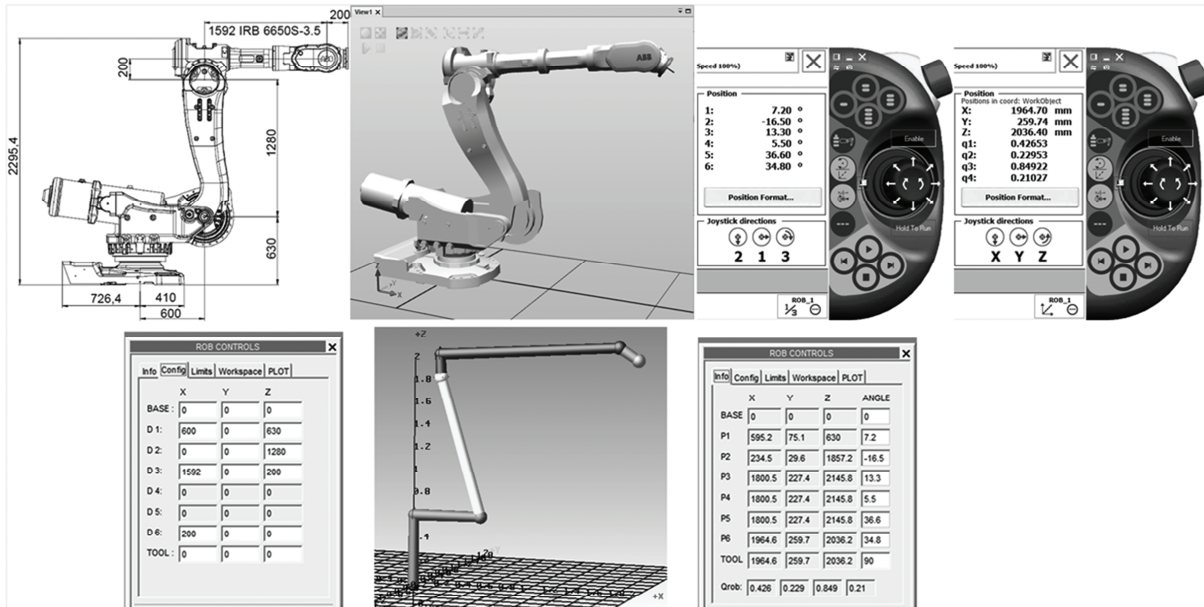


Figure 3. Modelling market robots and reliability check with commercial programs (ABB Robot Studio).



Figure 4. Rob control panel widgets.

The reality must be enhanced: Solid 3D models should be used instead of symbolic drawings to attract the attention of users. Interactive manipulation of the visual examination should be possible.

It must be able to communicate with physical equipment: This feature must be brand and model independent. Communication should be preferred over universal standards. Special communication interface channels and software development libraries should be avoided. It must be influential to add innovations to existing applications and designs.

In this section, we aim to outline the features of the program that are not found in academic nor commercial programs. The program has met the above criteria because of the features we have added. However, we had to use a library to access physical equipment. At this point we have open source and operating system independent libraries that we prefer to use. Because we mentioned mathematical basis in the previous section we do not include it in this section. We do not put a lot of emphasis on the program usage and menus, as it was an extra feature in the article that would increase students' understanding of what we wanted to do. Interested readers can download the program from http://ehm.kocaeli.edu.tr/dersnotlari_data/?dir=mcakir/IRT and discover the features we have not mentioned in the article. Particularly, there is a difficulty in visualizing the working space of articulated robot arms. However, when the robot's workspace is visualized, the access tests can be made more comfortable with collision zone analysis. Through this screen (Fig. 5), the robot can be placed in the optimum position by evaluating the access and collision zone. Robot control panel in Fig. 4 can be used to create the working area given in Fig. 5. Within the workspace tab on this control panel, the access area can be shown by entering the individual working range and scanning step for each joint. It is possible to add models in 3ds format to the main screen with related toolbar. As shown in Fig. 1, the scanning lines can be visualized on the added models or the cells including the model as in Fig. 5. The manipulator design with different geometric measurements on the robot control panel in Fig. 5 can be done as in [1].

5. Practical Application Examples

It is not enough to say that the IRT software only has application areas in education. It can also be used to develop practical applications. As shown in Fig. 1, there are scanning lines on the model placed in front of the manipulator. Point or position teaching can be done by moving the manipulator on these lines. This approach is the basis of robotic milling applications. Students could add these additional modules to the program so that they can be automatically transferred to the robot program in the project and their thesis studies. For example, they are able to make simple models with the manipulator and milling tool. In another study, the process of creating point clouds which is the starting point of reverse engineering applications is implemented by adding a stereo camera module. With the stereo control widget shown in Fig. 6, parameter effects are detected by live observation.

In this application, it is necessary to transfer the points obtained in the camera coordinate plane to the reference coordinate plane. The positions that the camera detects from different angles can only be combined on a common reference plane. The relationship between the camera position and the manipulator endpoint must be known to do this process. In other words, the camera needs to be introduced as an additional tool. The displacement and rotation information to introduce the posture can be realized by solving the equation system in the form $AX = XB$ at the end of the process known as hand-eye calibration [17]–[20].

After accessing the camera with the modules developed by the students in the project lesson, communication of the software with the manipulator shown in Fig. 7 is provided. The communication between the PLC and the robot control system was established through the Libnodave library. Figure 7 presents the environment image formed by combining information obtained from different positions. In the images shown in the horizontal profile, the dot cloud is a thin line, which is an indication of the sensitivity of the calibration process.

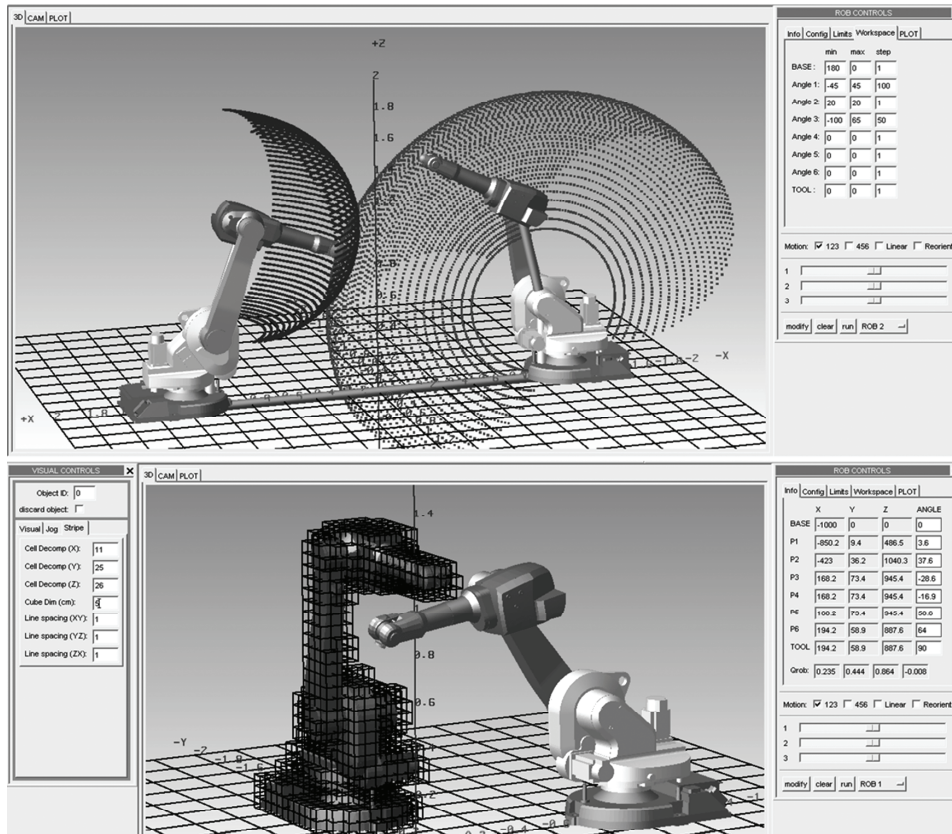


Figure 5. Workspace visualization and collision checking.

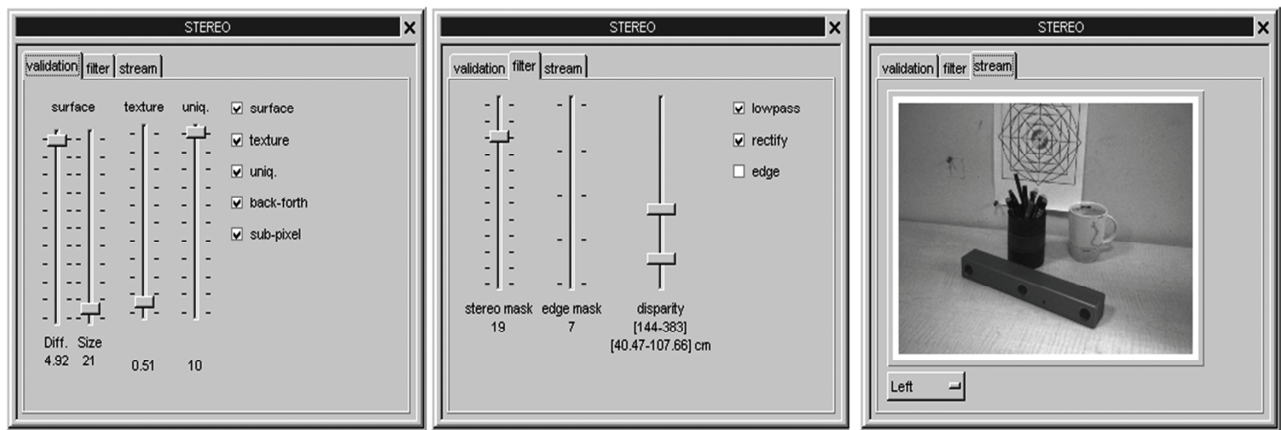


Figure 6. Stereo vision toolbar widgets.

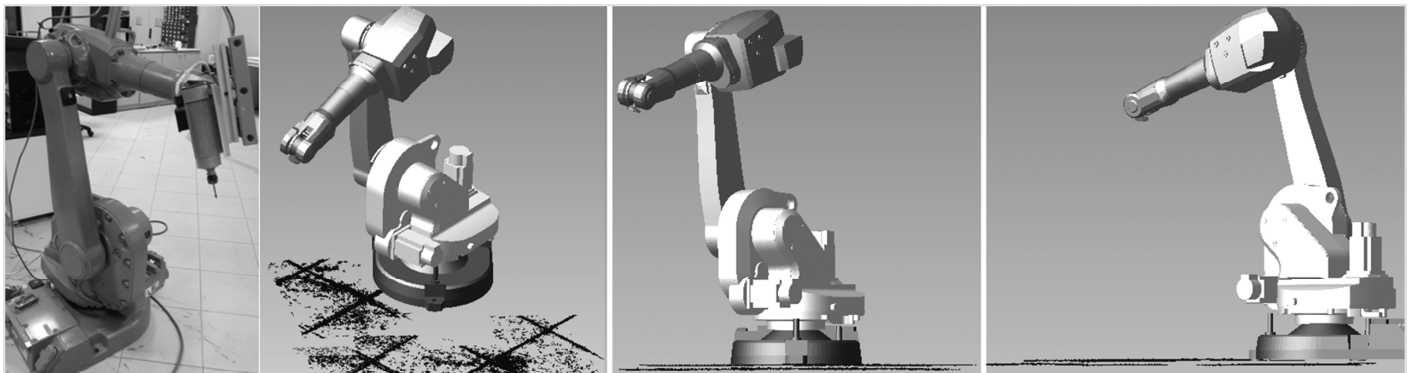


Figure 7. Real rob. Practice: Point cloud generation. Camera-robot transformation known as hand-eye calibration should be determined. PLC and Camera connection should be established.

As a result of the applications described above, the participants have seen the use of theoretical information. While students are learning code development more easily, it has been observed that business operators have difficulty doing this. However, they have been able to make observations about how the process works for them through this program.

6. Conclusion

In this study, a new software developed for offline robot platforms which is a basic necessity in robot education and applications is presented. It is not always possible to physically have a robot in the lab and recreate the conditions of an industrial environment. Even if these were possible, the robotic technique would be limited to a single robot brand and model. This training material is designed for industrial robots which include all robot brands and model configurations based on the robotic science. In this study, a software platform is developed that includes subjects related to industrial robots in engineering education such as programming, kinematics, workspace, speed-acceleration, and robot dynamic parameters. The platform is developed in the C++ programming language and the Fox ToolKit library and it is designed to work with every brand and model robot arm configurations. In this platform, it is possible to visualize the limits of working range, angle limits of each arm, basic robot movements (joint, linear and circular), and programming by means of any robot arms with desired lengths.

We have observed that this robot training platform enhances perception and persistence of knowledge in on-the-job training, in robotic training courses at universities and colleges, in special robot training courses, and in manufacturing shops that include robots. It has been seen that users can develop additional applications easily in project and thesis studies due to the platform being open source. It will be a very useful material for robotic and robotics lessons, special robot training courses, robotic manufacturing plants, and robotic field.

References

- [1] M. Flanders and R.C. Kavanagh, Build-a-robot: Using virtual reality to visualize the Denavit–Hartenberg parameters *Computer Applications in Engineering Education*, 23(6), 2015, 846–853. doi:10.1002/cae.21656
- [2] M. Berenguel, F. Rodríguez, J.C. Moreno, J. L. Guzmán, and R. González, Tools and methodologies for teaching robotics in computer science & engineering studies, *Computer Applications in Engineering Education*, 24(2), 2016, 202–214. doi:10.1002/cae.21698
- [3] A. Gil, O. Reinoso, J.M. Marin, L. Paya, and J. Ruiz, Development and deployment of a new robotics toolbox for education, *Computer Applications in Engineering Education*, 23(3), 2015, 443–454. doi:10.1002/cae.21615
- [4] G López-Nicolás, A. Romeo, and J.J. Guerrero, Active learning in robotics based on simulation tools, *Computer Applications in Engineering Education*, 22(3), 2014, 509–515. doi:10.1002/cae.20576
- [5] Z. Pan, J. Polden, N. Larkin, S.V. Duin, and J. Norrish, Recent progress on programming methods for industrial robots,

Robotics and Computer-Integrated Manufacturing, 28, 2012, 87–94.

- [6] Hodges and Smith, *Letter on quaternion* (Cambridge: Macmillan & Co), 1853, 60.
- [7] S. Cubero, *Industrial robotics: Theory, modelling and control*. ISBN 3-86611-285-8.
- [8] P. Richard, *Robot manipulators: Mathematics, programming, and control: The computer control of robot manipulators* (Cambridge, MA: MIT Press, 1981). ISBN 978-0-262-16082-7.
- [9] M.M. Fateh, Fuzzy task-space control of a welding robot, *International Journal of Robotics and Automation*, Vol. 25, 2010, 206–3437.
- [10] C.Q. Huang, S.J. Shi, X.G. Wang, and W.K. Chung, “Parallel force/position controls for robot manipulators with uncertain kinematics”, *International Journal of Robotics and Automation*, Vol. 20, 2005, 206–2721.
- [11] J.H. Kim, D.P. Roberts, J. Meusch, and S. Rahmatalla, Kinematic and kinetic evaluation of a throwing model using motion primitives, *International Journal of Robotics and Automation*, Vol. 30, 2015, 206–4280.
- [12] J.M. McCarthy, *Introduction to theoretical kinematics* (Cambridge, MA: MIT Press, 1990).
- [13] J. Denavit and R.S. Hartenberg, A kinematic notation for lower-pair mechanisms based on matrices, *Transactions ASME Journal of Applied Mechanics*, 23, 1955, 215–221.
- [14] R.S. Hartenberg and J. Denavit, *Kinematic synthesis of linkages* (New York: McGraw-Hill, 1964).
- [15] *On quaternions; or on a new system of imaginaries on algebra* (Letter To John T. Graves, Dated October 17, 1843), 1843.
- [16] B.A. Rosenfeld, *The history of non-Euclidean geometry: Evolution of the concept of a geometric space* (1988). ISBN 978-1-4419-8680-1.
- [17] H. Wang, X. Lu, Z. Hu, and Y. Li, A vision-based fully-automatic calibration method for hand-eye serial robot, *Industrial Robot: An International Journal*, 42(1), 2015, 64–73.
- [18] H. Pan, N.L. Wang, and Y.S. Qin, A closed-form solution to eye-to-hand calibration towards visual grasping, *Industrial Robot: An International Journal*, 41(6), 2014, 567–574.
- [19] J.-S. Hu and Y.-J. Chang, Eye-hand-workspace calibration using laser pointer projection on plane surface, *Industrial Robot: An International Journal*, 39(2), 2012, 197–207.
- [20] P.S. Ramaiah and K.N. Rao, A model-based approach to robot hand-eye coordination system, *Proc. Modelling and Simulation*, May 13–15, Marina Del Rey, USA, 2002.

Biographies



Cengiz Deniz received his B.Sc. degree in Electrical Science and M.Sc. and Ph.D degrees in Electronics and Communication Engineering from Kocaeli University, Kocaeli, Turkey, in 2007, 2011 and 2017 respectively. He has been working on industrial robots and applications at Ford Otosan Company in Turkey since 2007. Currently, he is working as Plant Technical Leader and responsible

for all robotics applications and new automation projects in the Kocaeli plants. His research interests are automation systems and industrial robotic applications.



Mustafa Çakır received his B.S., M.S., and Ph.D. degrees in Electronics and Communication Engineering from Kocaeli University in Turkey in 1996, 1998 and 2006, respectively. He worked as automation engineer at Ford Otosan Company in Turkey for 5 years. He is currently Ass. Prof. Dr. in Electronics and Communication Engineering department at Kocaeli University. He has lectured on control system and robotic science at the department. His research interests are automation systems, control, robotics, and power electronics.