STEM Education
STEM Education, 1 (1): 32–46
DOI: 10.3934/steme.2021003
Received: October 2020
Revised: January 2021

https://www.aimsciences.org/journal/A0000-0006

Research article

Interactive MATLAB based project learning in a robotics course: Challenges and achievements

Lotfi Romdhane[1,*] and Mohammad A. Jaradat[1]

[1] Mechatronics Engineering, Department of Mechanical Engineering, American University of Sharjah, P.O Box 26666, Sharjah, UAE; mjaradat@aus.edu (M.A.J.)
* **Correspondence:** lromdhane@aus.edu; Tel: +971-6-5152497

Academic Editor: Giuseppe Carbone

Abstract: This paper illustrates the conducted efforts for deploying an interactive project-based learning for robotics course using MATLAB. This project is part of a first course on robotics at the graduate level. The course combines both the theoretical and practical aspects to achieve its goals. The course consists of a set of laboratory sessions ends with a class project, these labs experimentally illustrate the modeling, simulation, path-planning and control of the Robot, using the robotics toolbox under MATLAB tools as well as physical interaction with the different robot platforms. The interaction between the student and the physical robot platforms is finally addressed in the class project; in this project, two tasks are considered. The first one is to control a 5DoF robot manipulator to perform a pick and place task. Initially the task is simulated under MATLAB robotics toolbox; the robot is commended to pick objects from initially known poses and stacks them in target poses. Furthermore, the robot manipulator in the second part of the project, with the aid of a vision system, is commended to work as an autonomous robotic arm that picks up colored objects, and then places them in different poses, based on their identified colors. The demonstrated results from the course evolution and assessment tools reflect the benefits of high-level deployment of robot platform in interactive project based learning to increase the students' performance in the course, about 100% and 75% of the student groups successfully completed the required tasks in the project first part and second part respectively.

Keywords: robotic arm, inverse kinematics, color detection

1. Introduction

Educational robotics, because of its multidisciplinary nature, is an excellent tool for the practical, hands-on application of various concepts encompassing a wide spectrum of topics from engineering and science [1]. Teaching robotics at the graduate or the undergraduate levels is usually challenging and exciting endeavor. Most of the graduate courses are limited to only lectures. However, having a laboratory component in the course can be highly beneficial for the learning of the students. The graduate course at hand is called Robotic Systems and it is part of the Mechatronic Engineering Master program at the American University of Sharjah, UAE. The main objective of the course is to initiate the students in the modeling and simulation of serial robots. However, with this theoretical background comes a practical component, where the students practice the use of these models in controlling serial robots in the lab.

In the literature, several undergraduate and even high school courses involved robotics as tool to cultivate the fundamental spirit of students for technology in general [1-4, 6-9], to increase the student's motivation and retention, and give students an overview about more advanced topics they will encounter if they elect to go for graduate studies [6].

Educational robotics is a popular topic and it was tackled in the literature from different perspectives. Along with the growing attention to STEM (Science, Technology, Engineering, Mathematics) education, robotics has been suggested as an innovative solution [2, 3]. In [3], the authors highlighted some problematic observations and some weaknesses of the current situation related to robotics in education in Europe. In [6], the authors introduced a first-year educational course offered under the General Education category. They tried to assess the use of robotics as an educational tool to improve the students' teamwork capabilities and communication skills.

In all these courses robotics as an engineering topic was not the main objective of the course, but it is rather used as a mean to improve the student's skills in communication, teamwork, and creativity [1-3].

Another category of courses is more focused on the topics of robotics and targeted improving the students' technical skills. The change from having a robotics course as a collection Mechanical, Electrical and Computer topics to a standalone and integrated course of robotics started in the mid 2000 [4]. Robotics courses were introduced where the main goal is to teach students how to model, simulate, and control robots [4, 11-13]. In [11], the authors presented a course open for diverse disciplines ranging from Neuroscience to Mechanical, where they introduced a programming course based on MATLAB. This multidisciplinary in the body of the students enriched the course. The authors of [12] presented a robotics course in the mechanical engineering curriculum, which is based on MATLAB. This course aims to improve the student's skills in programming automated manufacturing processes. The course was based on an educational robot. The students used MATLAB to simulate the system and to program it for different contoured paths. The authors concluded that MATLAB and the educational robot can enhance the student experience and give them greater insight into robotic applications. In [4], the authors presented a competition-based course project and the benefits of this type of activity in enhancing the learning outcomes of the robotics course.

The authors of [13] developed models under MATLAB® for the robots and sensors used in their laboratory. They proposed an integrated environment for the design and testing of advanced robot control schemes, under MATLAB®. In [22] the authors provided the results from a pilot project about evaluation the process of using MATLAB® in introductory robotics courses for engineers. A new MATLAB toolbox for teaching practical robotics class for new students in presented in [23, 24] using Lego Mindstorms NXT platform. The active learning experience in robotics course was reported in [25], the projects reflect the interaction process between the students and the physical components using robot platform to design and implement manufacturing systems. The results from using a reliable training robot platform for teaching real-time control courses for undergraduate students was demonstrated in [26]. Other researchers deployed the robot platform in teaching control systems theory for higher level students, the robot platform components were used with the MATLAB platform to model, design and analyze different control approach performance including classical or modern control approaches [27], while other researchers provide the control concept in teaching using the complete robot platform to perform a specific task such as line following [28]. Using the updated Lego robot platform (EV3) is introduced in [29] for teaching robotics, real-time MATLAB® /Simulink tools is presented to teach robotics subject. A review about the role of deploying robotics in teaching is summarized in [30], the survey provides the result for combining the theoretical and practical aspects in teaching using robotics. the importance of hardware based robotics class is discussed in [32], as reported the overall students' performance in the course was improved. The idea of teaching the robot kinematics using the integration between hardware and simulation tool is introduced in [33] using Lego Mindstorms NXT platform for undergraduate level experience. In [34] MATLAB® toolboxes were demonstrated for high level teaching especially for mechatronics and robotics courses. In [35] new platform was introduced for teaching purposes; the mobile robot platform is generalized using MATLAB® and high-level Robot Operating System (ROS).

Lately, the online experience is being introduced in educational robotics. In [13], the authors developed two massive open online courses (MOOCs) using MATLAB® as the main environment for delivering the material and the assessment. The issue with the online courses is their lack of hands-on experience, which cannot be delivered online. However, with the COVID19 pandemic, new concepts are being introduced on the market to try to virtualize the labs. These products are usually called Vlabs, for virtual labs. However, these VLabs are still at early stages and they are simply simulations, which cannot replace the practical experience of the students when dealing with the real system.

The practical and hands-on component is undeniably the most important part in fulfilling the learning outcomes of the course. Going online with a robotics course might not be recommended as it takes the practical part of the course away.

The paper is organized as follows, after the introduction in section one the courses overview is covered in section two, the course project is demonstrated in section three. Finally, the concluded remarks are summarized in section four.

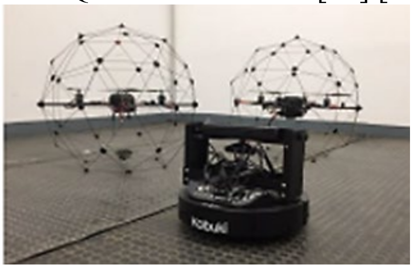
2. Course overview

This paper, besides presenting the main topics of the course, focuses on the practical experience of the students during this course. Two practical components are considered, the first one is the related to the laboratory experiments, and the second one is related to the course project.

The topics covered in the course deal mainly with modeling and simulation of serial robots. However, the last chapter deals with mobile robots. The course starts by introducing the mathematical tools needed for building the different models. 3D geometry with the different representations used for the position and orientation of a rigid body, are introduced. The Denavitt-Hartenberg parameters [15] along with the concept of homogeneous matrices, were used to model the kinematics of serial manipulators. Symbolic MATLAB® examples are used where possible to give the students an insight in the mathematical complexity of the different models. The symbolic tool is a valuable one as the use of industrial robotic software does not give the students any idea on the complexity of the model they are solving when simulating a robot motion. Then, the different kinematic and dynamic models were covered with practical examples and case studies. These case studies use existing robots in the lab.

Different robots, with different sizes are used in the Mechatronics Lab. Table 1 shows a brief presentation of these robots.

Table 1. List of Robots used in the Labs.

3 DOF OMNI robot, by Quanser [16],	4 DOF Mico robot by Kinova [17]
	
Quanser QBot mobile robot [18] [31]	5-DOF industrial robot by ABB [19].
	

Except for the ABB robot, these robots are didactic ones and the student can interface with them through MATLAB/Simulink routines and toolboxes. The ABB robot is a recent acquisition and it comes with its own dedicated software. The first three robots are used by the students in their experiments to illustrate different concepts introduced during the lectures. Experiments, using the ABB robot, are planned for this current semester, Fall 2020.

Besides the hardware platforms, in this course, the students are initiated in using different software packages:

• The first one is the symbolic toolbox of MATLAB, where the students derive symbolically several models with increasing complexity. This exercise is a valuable tool to give the students an insight of the complexity of the models used in robotics and its limitation, especially when it comes solving the inverse kinematic problem.

• The second software is more specific to robotics, where the students are initiated in using two robotic toolboxes under MATLAB, i.e., Robotic Toolbox, by Corke [15] and Robotic Systems Toolbox, by Mathworks [20].

• The third type is an industrial software, called Robot Studio, by ABB [19]. This software comes with the newly acquired ABB robot.

The concepts introduced in the lectures are applied and illustrated in the labs. The student is first familiarized with the robot structure then more advanced tasks are introduced. The list of labs used introduced in the course are:

• Model serial manipulators: This first lab requires the students to extract the Denavitt-Hartenberg parameters of the three existing serial manipulators.

• Solve the inverse kinematic problem for the Mico robot.

• Derive the Manipulator Jacobian of the Mico robot

• Trajectory planning for the Mico robot

• Trajectory planning for the Mobile robot

• This Fall 2020 a new experiment will be introduced using the industrial robot by ABB.

During these experiments, the student is asked to manipulate closely the robot and to be acquainted with the different risks associated with the use of robots. Even though these robots are all didactic and do not present any risk for the students, the concept of safety is stressed to make them aware of the risks when manipulating industrial robots. The newly experiment to be introduced addresses this issue. Indeed, the robot setup has a safety fence, which shows the students the importance of safety while manipulating the robot.

The last component of the course is the project. This important part will be detailed in the next section.

The grading of the course reflected this practical component of the course. The adopted scheme for grading accounts for more 30% of the grade is based on labs and the project.

The rest of the paper will focus on the project and the different steps followed to achieve the goals.

3. Course project

The main objective of this project is to illustrate the concepts of forward and inverse kinematics problems when defining a task for the serial robot. Visual feedback is also utilized to automate the detection of the objects to be manipulated.

Each team will have its own Lynxmotion Robot platform [21]. The final objective is to real-time control the robot through a program written in MATLAB® for the pick-and-place of 6 blocs between the initial and final poses. Each two teams had different pose. Team 1 and 2 had the initial and final poses shown in Figure 1. Whereas team 3 and 4 had the poses shown in Figure 2.

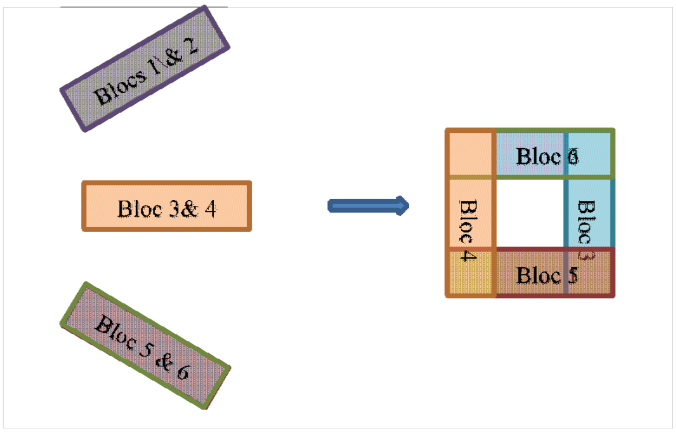


Figure 1. Initial and final poses of the blocs for Groups 1 and 2.

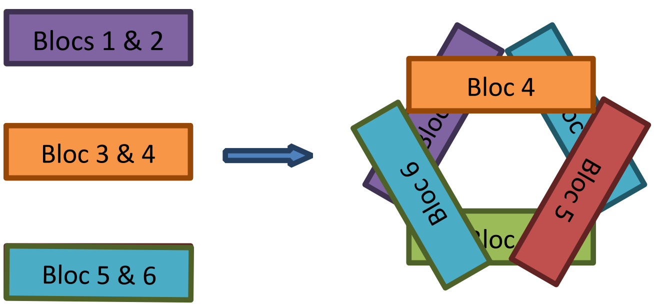


Figure 2. Initial and final poses of the blocs for Groups 3 and 4.

The program should use the inverse and the forward kinematics for a pick and place task and the control is performed in open loop. A plate with the poses of the blocks is available in the lab.

In addition, the required speed of each servo should be calculated in order to complete a motion in the shortest time in a smooth way and without any jerky motion. This project uses a platform robot kit from Lynxmotion, which is a 5-axis robot using servomotors for independent joint control. The student has to assemble the robot kit and test its mechanics.

The task should first be simulated using MATLAB® and an animation showing the full task has to be approved by the Instructor before connecting the robot to the computer. The main objective of this part is to make sure that the student can generate the full model of the robot and plan its trajectory.

A vision-feedback based task was added, as a bonus, to the project to motivate the students to push the limits, since this part was assigned for life-long learning part of the class. This task consists in picking several colored blocks and scattered in a given area and place them in 3 different locations corresponding to the 3 available colors. The vision part the students did their own search and learning task, the objective was achieved successfully by 3 out of the 4 teams completed this vision- control based task. The assessment of the project considers 4 rubrics as shown in Table 2.

Table 2. Assessment rubrics

Rubric 1	Report and presentation: quality, scientific content, Matlab code…	30%
Rubric 2	Accuracy of the final pose of the blocs	35%
Rubric 3	Speed of execution of the task	20%
Rubric 4	Smoothness of the motion	15%
Rubric 5 (Bonus)	Vision-based task	10%

3.1. Methodology

The DH parameters were used to construct 5-axis robot using the MATLAB® robotics toolbox [15]. The inverse kinematics was applied using the pre-defined objects' poses to determine the joints' angles. The dimensions shown in the below table are based on actual measures by the students. A small error was noticed in the measurements, which affected the final poses of the blocs.

Table 3. DH parameter of the robot

i	Links	θ_i(rad)	d_i(cm)	α_i(rad)	a_i(cm)	offset
1	Base	θ_1	6.5 ± 0.5	-90	0	-90
2	Shoulder	θ_2	0	0	14.5 ± 0.5	-90
3	Elbow	θ_3	0	0	18.5 ± 0.5	90
4	Wrist	θ_4	0	-90	0	0
5	Gripper	θ_5	12.5	0	0	-90

This problem was subdivided into a position problem, using the three first axes, and an orientation problem, using the last two axes. Figure 3 shows how the three joints are simulated in MATLAB®.

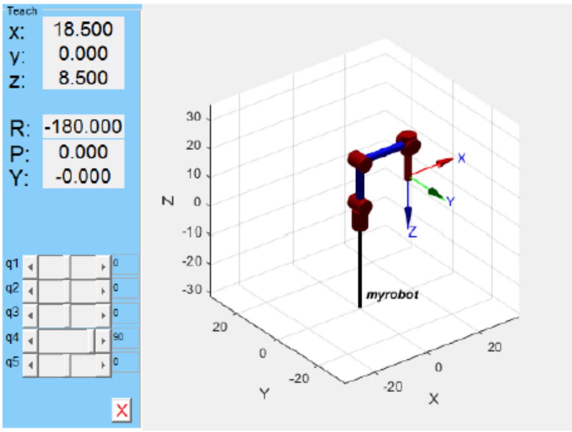


Figure 3. Robot simulation in MATLAB®

After obtaining the first three joints' angles, the fourth joint's angle was obtained using the below polygon shape as shown in Figure 4, as follows:

$$q_4 = 360 - (90 - q_2) - 90 - (90 - q_3) \quad (1)$$

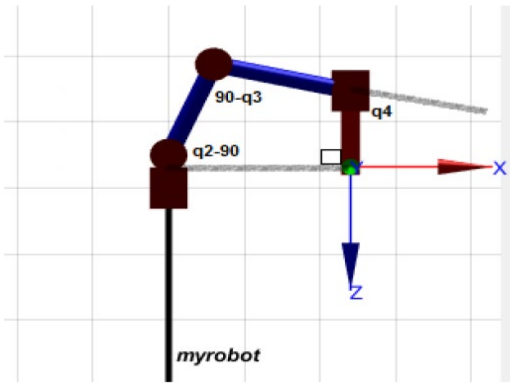


Figure 4. Fourth angle calculation method

The fifth joint's angle was calculated using the known object's orientation. The students found that building the model was a relatively easy task, especially when using the Robotic Toolbox. However, most of them were not successful in solving the inverse kinematics problem, using the toolbox. This difficulty is caused by the fact that the robot has only 5 DOFs. The function in the toolbox is made for 6 DOF serial robots with a spherical wrist. All the teams preferred to solve the geometric problem manually for this specific structure. Some numerical issues were encountered due to the multiplicity of possible solutions. The selection of the right solution was based on the observation of the real system. An example of poses representing the pick and place are given as follows:

P_1(:,:,1) =

1.0000	0	0	0
0	-1.0000	-0.0000	150.0000
0	0.0000	-1.0000	70.5000
0	0	0	1.0000

```
P_2(:,:,2) =     1.0000         0         0         0
                      0   -1.0000   -0.0000  150.0000
                      0    0.0000   -1.0000  110.5000
                      0         0         0    1.0000

P_3(:,:,3) =     0.7071    0.7071         0  106.0660
                 0.7071   -0.7071   -0.0000 -106.0660
                -0.0000    0.0000   -1.0000  111.0000
                      0         0         0    1.0000

P_4(:,:,4) =     0.7071    0.7071         0  106.0660
                 0.7071   -0.7071   -0.0000 -106.0660
                -0.0000    0.0000   -1.0000   41.0000
                      0         0         0    1.0000
```

P_1 represents the first pose, where the first bloc was picked. P_4 is the final pose, where the bloc is placed. These poses change depending on the location of the different blocs to be picked and placed. The two intermediate poses P_2 and P_3 do not change for all the blocs.

The joint angles along the path are generated using the command "q = ikine(robot, T)", where q is an array containing the 5 joint angles corresponding to the pose T. 30 intermediate poses were selected to smooth the path. Figure 5 shows the results of the joint angles variations between the pick and place poses.

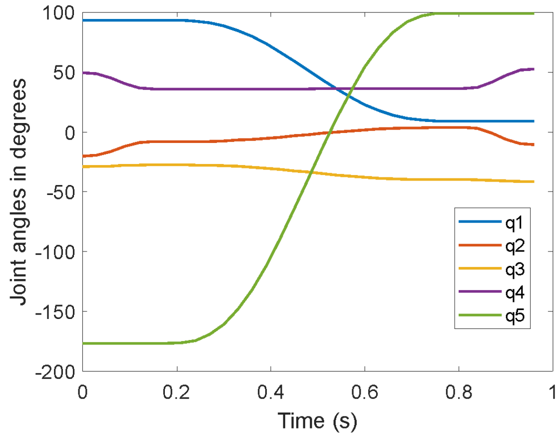


Figure 5. Joint angles variation between the pick and place poses

3.2. Results

1) Task 1: palletization

Each of the 4 teams had 15 minutes to showcase his project. Moving the 6 blocs normally takes around 3 minutes or less. More than one attempt is allowed within the 15 minutes allocated to each team. This task was performed successfully by the 4 different teams. The time ranged from 2.5 minutes to 4 minutes. The slowest team was very conservative and elected to go slowly to make sure of the accuracy of the final pose. Due to the differences in the measured dimensions of the robot, some differences appeared while executing the tasks. 3 out of the 4 teams succeeded in obtaining the correct final poses and the stacking was very close to the desired one. For one team the model had a lot of error and this resulted in an error in the final pose, which caused the stacking not to be successful. This same team took the longest as its members elected to go slowly to ensure the accuracy of the final pose. However, the issue was not the speed but rather the values of the parameters injected in the model. This exercise clearly showed the importance of the model and how it influences the use of the robot. Since the open-loop control was used and there are no sensors to feedback the real pose of the end-effector, it was not possible to correct the error online. The students must stop the robot revise their model and test it again. To motivate the students, the project was set up as a competition. Measures of the quality of the staking operation were the main criterion used to evaluate the success of the task. Speed and quality of the motion were also used. The percentages assigned to the different assessment rubrics (Table 2) reflect this evaluation. Rubrics 3 and 4 in the same table are somehow related. Indeed, if the speed of the servomotors is selected high, the motion becomes jerky and the robot can overshoot when reaching the final pose.

Table 4 shows the scores obtained by the different teams. The grading scheme allows for a grade higher than 100 due to the bonus part.

Table 4: Grading of the different teams

	Rubric 1 (30%)	Rubric 2 (35%)	Rubric 3 (20%)	Rubric 4 (15%)	Rubric 5 bonus (10%)	Score (110%)
Team 1	5	5	5	5	5	110
Team 2	4	4	3	3	-	73
Team 3	3	3	3	3	3	66
Team 4	4	4	3	4	2	80

Figure 6 shows the results obtained by the winning team. The model was highly accurate, which explains the close to the target final poses. Moreover, this team succeeded in controlling the motion in a smooth way and got the shortest time to perform the task.

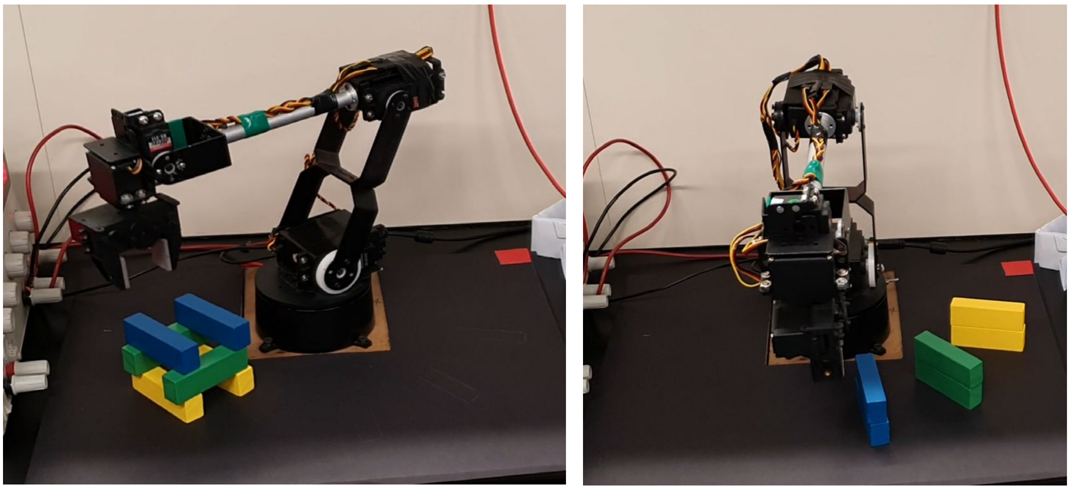


Figure 6. Results of the winning team (Team 1)

Figure 7 shows the results obtained by Team 4. The task turned out to be a bit more challenging than the first task.

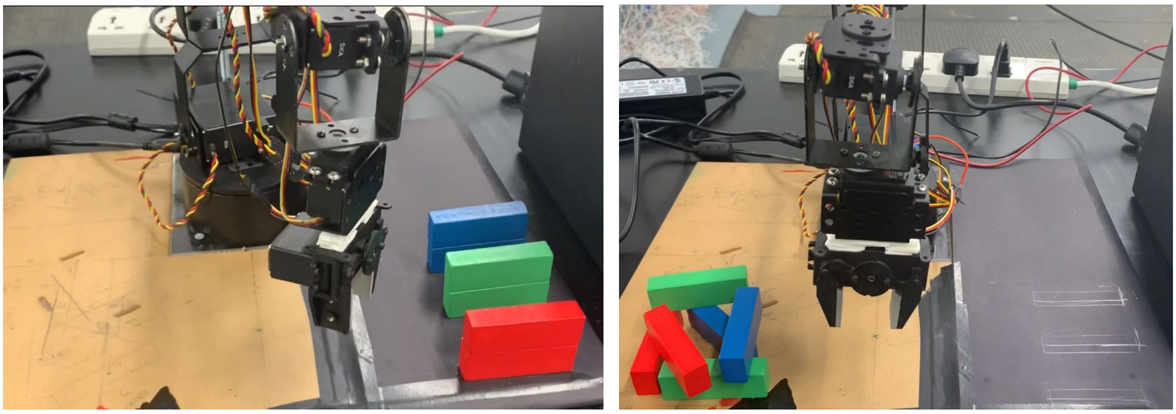


Figure 7. Results obtained by Team 4

3.3. Vision-feedback based task

The second objective of the project was to design an autonomous robotic arm to pick objects with different colors using the feedback from a camera. In this section, the process of color detection is explained. The robot uses this data to locate the objects and pick them, then place them in a known pose based on their color.

3 out the 4 teams, elected to use a webcam to identify the blocs, whereas Team 1 elected to use the camera of a smartphone to detect the object. The algorithm of the object's detection was programmed under MATLAB. The MATLAB Support Package for USB Webcams was used to interface with the webcam. The fourth team (Team 1) still used MATLAB with a Support Package for IP Cameras. First, a snapshot is acquired from the webcam (I_{RGB}) , which consists of 3 layers of RGB colors. Each layer was separated and detected, separately:

$$I_{RGB}(x,y) = \{R_{red}(x,y), G_{green}(x,y), B_{blue}(x,y)\} \tag{2}$$

Then, the colored image layers are then combined to a single gray image (I_{Gray}) using the weighted sum function [36]:

$$I_{Gray}(x,y) = \alpha R_{red}(x,y) + \beta G_{green}(x,y) + \gamma B_{blue}(x,y) \tag{3}$$

where α, β, and γ are constants. The gray image is converted to binary image (I_B (x,y) = 0 or 1) using image thresholding (μ):

$$I_B\,(x,y) = \{1\; I_{\,Gray} \geq \mu\;\; 0\, else \tag{4}$$

Since the size of the object is known, the image is cleaned by deleting any small objects, which could appear because of errors in the detection phase, later on the holes in the binary image are processed and filled, the remaining object in the image are connected and labeled. The cleaned image is then used to determine the position and orientation of the object with respect to the camera. The center of each object (xc_i , yc_i) and the object orientation (θc_i) with respect to the image frame are extracted using the moments utilizing the MATLAB® image processing toolbox [37]:

$$xc_i = \frac{M_{10}}{M_{00}} \tag{5}$$

$$yc_i = \frac{M_{01}}{M_{00}} \tag{6}$$

where

$$M_{ij} = \sum_{x}^{n} \sum_{y}^{m} x^{i} y^{j} I_{B}(x, y) \tag{7}$$

$$\theta c_i = 0.5 \left(\frac{\underline{M}_{11}}{\underline{M}_{20} - \underline{M}_{02}}\right) \tag{8}$$

where

$$\underline{M}_{11} = M_{10} - \frac{M_{10}M_{01}}{M_{00}} ; \underline{M}_{20} = M_{20} - \frac{M_{10}M_{10}}{M_{00}} ; \underline{M}_{02} = M_{02} - \frac{M_{01}M_{01}}{M_{00}} \tag{9}$$

The object center is used to detect the color of the respective object, by using the original image RGB value at this center to specify the object color:

$$Color = max \left[I_{RGB} \left(x_{ci}, y_{ci}\right)\right] \tag{10}$$

Up to this point, the location points obtained are in pixels with respect to the image frame and size. A mapping function is used to convert the pixels to an actual distance from the environment based on the size of the robot workspace. As shown in Figure 8 the positions and orientations from the image local frame is transformed to the robot frame. The block diagram for the of vision system is shown in Figure 9 [38, 39].

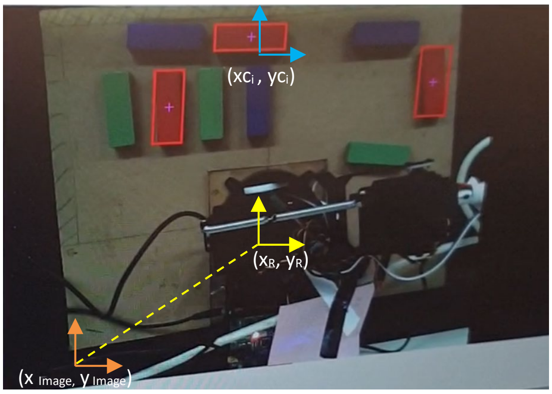


Figure 8. Frames transformation from image space (orange) to robot workspace (yellow).

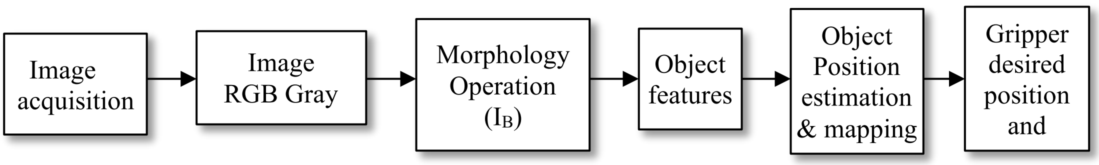


Figure 9. Block diagram for the of vision system.

For the orientation, the webcam/IP cameras toolboxes calculate the orientation angle based on the object local frame and it gives the lowest angle from the x-axis in either direction positive or negative:

• If the orientation angle is positive, then the '+x' axis is used for the object local frame and to transform the orientation angle to the robot frame a value of 'π' is added.

• If the orientation angle is negative, then in this case the '-x' axis is used in the object local frame and to transform to robot frame '$-\pi$' is added.

Once the center of the object and its orientation have been defined with respect to robot frame, the robot uses the inverse kinematic routine to move the robot to the center of the object as explained in the pick and place task.

Only 3 out of the 4 teams have succeeded in completing this task. Figure 9 shows three tests made by the three teams, where the objects were randomly scattered. In each case, the robot was able to detect each object based on its color. However, for one of the teams the localization was not successful and the robot was unable to complete the task.

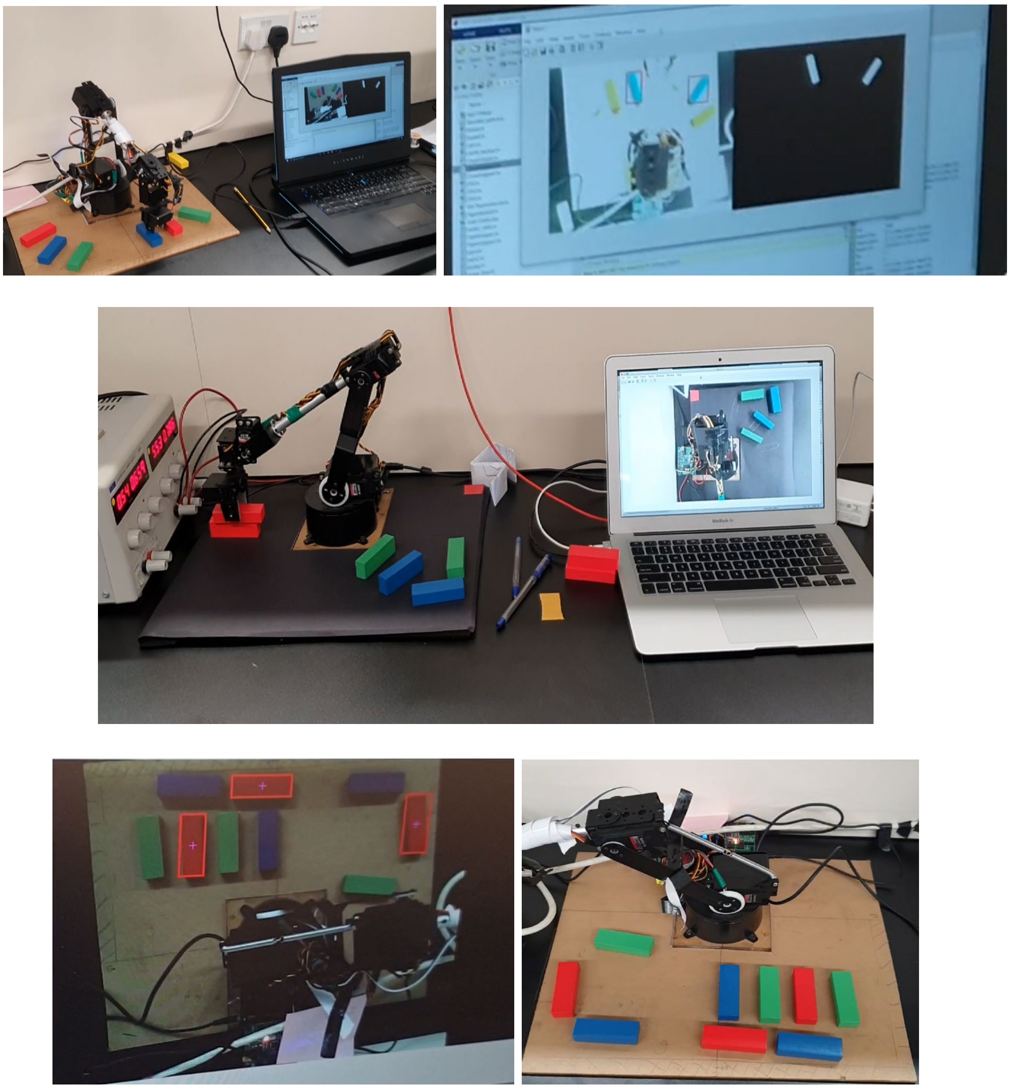


Figure 9. 3 experiments where the blocs were detected based on their colors

Based on these experiments, the feedback of the students was collected in the course evaluation process. In this feedback some students confirmed the benefits of this type of interactive projects in a robotics course. One student stated "…it also contained plenty of hands–on work in both the lab and the project" another student mentioned "…in addition adding a well–rounded project that really reinforces the material."

4. Conclusion

In this paper, the development of interactive project-based learning for robotics courses is described. The practical interactive feature of the course enabled the students to gain the required hands-on experience through a set of laboratory experiment and a group project. The developed curriculum allowed the students to practically apply advanced concepts related to robotics under MATLAB® platform. The course achieved its integration objectives by giving the students the required hands-on skills for real time control of robot manipulator with the theoretical part by successfully completing the project. The 5 DoF robot manipulator platform was successfully able to perform pick and place task utilizing the control commands from MATLAB program developed by the students. Another task was also deployed with MATLAB® in the course, in this task visual feedback task was successfully implemented to pick different colored objects (Red/Green/Blue), and then to place them in different places, based on their colors. The demonstrated results from the course evolution and assessment tools reflect the benefits of high-level deployment of robotics in interactive project-based learning to increase the students' performance in the course.

Acknowledgments

We are grateful for the constructive feedback provided by the reviewers.

References

1. Ronald, R., Bloom, D.S., Carpinelli, J., Burr-Alexander, L., Hirsch, L.S., Kimmel, H. (2010) Advancing the "E" in K-12 STEM education. *The Journal of Technology Studies* 36(1): 53-64.
2. Jung, S.E., Won, E.S. (2018) Systematic review of research trends in robotics education for young children. *Sustainability* 10(4): 905.
3. Bredenfeld, A., Hofmann, A., Steinbauer, G. (2010) Robotics in education initiatives in Europe - Status, shortcomings and open questions. *Proceedings of International Conference on Simulation, Modeling and Programming for Autonomous Robots*, November 15-16, 2010, Darmstadt (Germany).
4. Rus, D. (2006), Teaching robotics everywhere. *IEEE Robotics & Automation Magazine* 13: 15-94.
5. Calnon, M., Gifford, C.M., Agah, A. (2012) Robotics competitions in the classroom: Enriching graduate-level education in computer science and engineering. *Global Journal of Engineering Education* 14(1): 6-13.
6. Alimisis, D. (2013), Educational robotics: Open questions and new challenges. *Themes in Science & Technology Education* 6: 63-71.
7. Eguchi, A. (2014) Educational robotics for promoting 21st century skills. *Journal of Automation, Mobile Robotics & Intelligent Systems* 8(1): 5-11.
8. Amy, E. (2015) *Educational robotics as a learning tool for promoting rich environments for active learning (REALs)*. Handbook of Research on Educational Technology Integration and Active Learning, edited by Jared Keengwe, IGI Global, 19-47.
9. Zeidler, D.L. (2016) STEM education: A deficit framework for the twenty first century? A sociocultural socioscientific response. *Cultural Studies of Science Education* 11: 11–26.
10. Aroca, R.V., Watanabe, F.Y., De Ávila M.T., Hernandes, A. C. (2016) *Mobile robotics integration in introductory undergraduate engineering courses*. The XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium, 139-144.
11. Joshi, S.S. (2004) Development and implementation of a MATLAB simulation project for a multidisciplinary graduate course in autonomous robotics. *Computer Applications in Engineering Education* 12: 54–64.
12. Hamilton, C. (2007) Using MATLAB to advance the robotics laboratory. *Computer Applications in Engineering Education* 14: 205–213.

13. Žlajpah, L., Nemec, B., Omrþen, D. (2010) MATLAB-based robot control design environment for research and education. *Simulation Notes Europe* 20: 55-66.
14. Corke, P., Greener, E., Philip, R. (2016) An innovative educational change: Massive open online courses in robotics and robotic vision. *IEEE Robotics & Automation Magazine* 23: 81-89.
15. Corke, P. (2013) *Robotics, Vision and Control: Fundamental Algorithms in MATLAB* (1st ed.). Springer.
16. https://www.quanser.com/products/omni-bundle/ [accessed September 2020]
17. https://www.kinovarobotics.com/en/knowledge-hub/mico-robotic-arm [accessed September 2020]
18. https://yujinrobot.com/ [accessed September 2020]
19. https://global.abb/group/en [accessed September 2020]
20. https://www.mathworks.com/ [accessed September 2020]
21. http://www.lynxmotion.com/ [accessed September 2020]
22. Azemi, A., Pauley, L.L. (2008). *Teaching the introductory computer programming course for engineers using Matlab*. The 38th IEEE Annual Frontiers in Education Conference.
23. Behrens, A., Atorf, L., Aach, T. (2010). *Teaching practical engineering for freshman students using the RWTH-Mindstorms NXT Toolbox for MATLAB*. Matlab-Modelling, Programming and Simulations, 41-65.
24. Behrens, A., Atorf, L., Schwann, R., Neumann, B., Schnitzler, R., Balle, J., Herold, T., Telle, A., Noll, T.G., Hameyer, K. and Aach, T. (2009). MATLAB meets LEGO Mindstorms - A freshman introduction course into practical engineering. *IEEE Transactions on Education 53*(2): 306-317.
25. Lugaresi, G., Lin, Z., Frigerio, N., Zhang, M., Matta, A. (2019). *Active learning experience in simulation class using a LEGO®-based manufacturing system*. IEEE Winter Simulation Conference (WSC), pp.3307-3318.
26. Cruz-Martín, A., Fernández-Madrigal, J.A., Galindo, C., González-Jiménez, J., Stockmans-Daou, C., Blanco-Claraco, J.L. (2012). A LEGO Mindstorms NXT approach for teaching at data acquisition, control systems engineering and real-time systems undergraduate courses. *Computers & Education 59*(3): 974-988.
27. Kim, Y. (2010). Control systems lab using a LEGO Mindstorms NXT motor system. *IEEE Transactions on Education 54*(3): 452-461.
28. Ding, J., Li, Z., Pan, T. (2017). Control system teaching and experiment using LEGO Mindstorms NXT robot. *International Journal of Information and Education Technology 7*(4): 309.
29. Montés, N., Rosillo, N., Mora, M.C., Hilario, L. (2018). *Real-time Matlab-Simulink-Lego EV3 framework for teaching robotics subjects*. International Conference on Robotics and Education. Springer.
30. Spolaôr, N., Benitti, F.B.V. (2017). Robotics applications grounded in learning theories on tertiary education: A systematic review. *Computers & Education* 112: 97-107.
31. https://www.quanser.com/products/qbot-2e/. [accessed September 2020]
32. Ackovska, N., Kirandziska, V. (2017). *The importance of hands-on experiences in robotics courses*. The 17th IEEE International Conference on Smart Technologies.
33. González-García, S., Rodríguez-Arce, J., Loreto-Gómez, G., Montaño-Serrano, V.M. (2020). Teaching forward kinematics in a robotics course using simulations: Transfer to a real-world context using LEGO mindstorms™. *International Journal on Interactive Design and Manufacturing (IJIDeM)* 14(3): 773-787.
34. Kobzev, A., Monakhov, Y., Lekareva, A. (2019). *Using Matlab software package in the curriculum of Mechatronics and Robotics*. The IEEE International Conference on Industrial Engineering, Applications and Manufacturing, pp. 1-6.

35. Ferreira, N.M.F. (2019). A generalized Matlab/ROS/Robotic platform framework for teaching robotics. *Robotics in Education: Current Research and Innovations* 1023:159.
36. Matlab Image Processing Toolbox, https://www.mathworks.com/help/matlab. [accessed September 2020]
37. Spong, M.W., Hutchinson, S., Vidyasagar, M. (2006). *Robot Modeling and Control*. John Wiley and Sons.
38. Jaradat, M.A.K., Al-Fandi, M., Nasir, M.T. (2012). Automatic control for a miniature manipulator based on 3D vision servo of soft objects. *Mechatronics* 22(4): 468-480.
39. Al-Fandi, M., Jaradat, M.A., Abusaif, A., Yih, T.C. (2010). *A real time vision feedback system for automation of a nano-assembly manipulator inside scanning electron microscope*. The 7th IEEE International Multi-Conference on Systems, Signals and Devices, pp. 1-5.