Development and Deployment of a New Robotics Toolbox for Education

ARTURO GIL, OSCAR REINOSO, JOSE MARIA MARIN, LUIS PAYA, JAVIER RUIZ

Universidad Miguel Hernandez de Elche (UMH), *Avda. de la Universidad s/n*, 03202 *Elche* (*Alicante*), *Spain*

Received 14 January 2014; accepted 7 October 2014

ABSTRACT: This paper presents a new toolbox focussed on the teaching of robotic manipulators. The library works under Matlab and has been designed to strengthen the theoretical concepts explained during the theory lectures. The educational approach is focussed on teaching the main concepts through developing math modeling and simulation. In order to do this, the toolbox aims at the fulfillment of a set of practical sessions that allow the students to test most of the concepts of an introductory course in robotic manipulators. In addition, the library possesses features that typically needed the usage of proprietary software, such as the visualization of a realistic 3D representation of commercial robotic arms and the programming of those arms in an industrial language. The practices include the concepts of direct and inverse kinematics, inverse and direct dynamics, path planning and robot programming. As a transversal practice, during the sessions, the student is asked to choose and integrate a new robotic arm in the library, proposing a particular solution to the direct and inverse kinematic problem, as well as the inclusion of other important parameters. The library has been deployed during the last year in bachelor and master studies and has received a nice acceptance. Finally, the library has been assessed in terms of usefulness, design and usage by means of a student survey. In addition, the surveys were designed to establish a relation between the student perception of the system, the time spent on the tool and their learning achievements. © 2014 Wiley Periodicals, Inc. Comput Appl Eng Educ 9999: 1–12, 2014; View this article online at wileyonlinelibrary.com/journal/cae; DOI 10.1002/cae.21615

Keywords: robotic manipulators; teaching; simulation

INTRODUCTION

This paper presents a new library addressed to the teaching of robotic manipulators. The tool is addressed to the teaching of serial manipulators in the last year of Bachelor studies or in Master studies. The toolbox works under Matlab and consists of a list of functions that solve general problems, such as direct kinematics, as well as graphics functions that allow to obtain a realistic representation of robotic manipulators. The library includes a set of manuals, laboratory sessions and exercises that guide the students and helps them understand the main concepts that are usually taught in an introductory course in robotics manipulators. We consider that these kind of toolboxes are of great help in the area of robotics where it is of paramount importance the execution of practical activities in order to understand and practice the knowledge acquired during theory lectures [1–3]. Moreover, in order to obtain a deep understanding of the concepts acquired during theory lessons, the students should develop their own mathematical models and simulate them. The choose of a high-level package such as MATLAB greatly eases the development of more advanced robotic models [4,5]. In the approach presented here, both modeling and simulation are developed under Matlab. The toolbox has been named ARTE (A Robotics Toolbox for Education) and is published under GNU LGPL license (it can be freely downloaded from http://arvc.umh.es/arte).

Experimentation in an engineering discipline is of paramount importance during undergraduate courses because it allows the students to observe real applications and obtain a deep comprehension of the theoretical concepts. In engineering, this experimentation has to be carried out by means of the utilization of real equipment, such is the case of industrial robotic manipulators. We can distinguish different ways to interact with this real equipment: first, classical laboratories allow the students to interact with real equipment and are of great importance, since they offer the possibility to observe the actual behavior and response of the systems being deployed. Second, we can mention virtual and remote laboratories (VRLs) that allow the students to

Correspondence to A. Gil (arturo.gil@umh.es).

© 2014 Wiley Periodicals, Inc.

interact with real equipment by means of a web interface. An example of this kind of laboratories is presented in [6,7] where the users can simulate and test positioning commands for a robot by means of a virtual environment as well as execute high level motion commands on a real robotic manipulator that is remotely observed by means of a camera.

During the last decade, the development of Internet applications and the use of virtual and remote laboratories have become popular. These laboratories allow the access to Web-based environments from any place and at a flexible schedule. These remote or virtual laboratories [6–8] provide realistic hands-on experiences and allow the student a fast understanding of concepts. However, in the field of automation and robotics, the educational methodologies should not be based merely on the use of VRLs. First, VRLs are usually designed to be used individually, and this fact may cause feelings of isolation in the student, thus reducing his motivation [9]. In addition, these tools are normally closed from the point of view of the mathematical models exposed. Thus, new models cannot be added and simulated by students. Typically, in the case of remote laboratories devoted for serial manipulators, a single robot is modeled and simulated. In addition, some authors consider that VRLs laboratories should be used as an initial contact to observe the studied phenomena for the student [7,8]. In the case of robotic manipulators, we consider that a deep understanding of the equations that model the kinematics and dynamics of these machines can only be achieved when the student writes their own models and test their own equations. In this sense, we believe that Matlab [10] is a well-known tool that allows the students to test their models easily and in a fast way. Matlab toolboxes are used frequently for educational purposes in different fields [11–13], generally with nice classroom experiences and nice results.

The rest of the paper is organized as follows: next, Motivation explains the main reasons that motivated the development of this new robotic toolbox. Next, Main Features of the Toolbox presents the main features of the toolbox. Following, Course Sequence and Student Background presents the course sequence and students' background. The software architecture is described in Software Architecture. Next, Practical Sessions and Deployment of the Tool presents the deployment of the tool by means of a set of laboratory sessions. Finally, an assessment of the tool is presented in Evaluation of the Tool Effectiveness, whereas the main conclusions in Conclusions.

MOTIVATION

In this section, we present the reasons that originated the idea of implementing a new robotics toolbox for educational purposes. First, we should cite some related work in this field, being the "Robotics Toolbox for Matlab" [11] an important contribution. This toolbox is based on Matlab and offers a very complete tool that includes a great quantity of basic and advanced algorithms for robotic manipulators. However, if we consider the usage of the tool for educational purposes some features may be missing, mainly, other toolboxes [11,14] do not consider a realistic representation of the robots and only use a simplistic wire representation. We consider that the student is able to learn faster when they can observe a realistic representation of the robot. The effects of 3D simulation in the motivation and learning of students has been proved before [15]. Thus textbook instruction and hands-on lessons do not suffice and cause that students often lack conceptual understanding [16]. It is also worth mentioning the contribution presented in [3], where the SnAM simulation software is presented, which consists of a public domain package written in C^{++}. The graphical representation is handled by OpenGL libraries. As a result, SnAM provides a tool to simulate with flexibility the kinematics of any kind of serial robot. A similar tool is presented in [17]. Also based on C^{++} and developed under Microsoft Visual C^{++}, RobotScene allows to simulate the kinematics of several industrial robots in typical applications.

In addition, an introductory course in robotics manipulators should introduce a robotic programming language to the students, such as, for example, RAPID or MELFA [18,19]. In addition, the Robotics toolbox for Matlab [11] does not consider the programming of the robot in any language, nor the usage of a teach pendant to program the robot. Finally, the last version of the toolbox is object oriented and we have observed that this fact causes that the students experiences some difficulties when using the library to solve exercises.

There exist other general tools that can be used to teach robotics. For example, SimMechanics [20] is a general Matlab toolbox oriented to the simulation of multi-body systems. This tool can be applied to solve a broad set of different applications, and thus, it is not specifically oriented to the field of robotic manipulators. According to our experience, students find difficulties when adapting a general tool to simulate a particular robotic application. For example, in SimMechanics the Denavit–Hartenberg [21] conventions are not employed to model the robot links.

There exist lots of proprietary tools provided by the main robot models manufacturers that are oriented to the simulation of their own robot models [18,19]. Usually, these tools allow the programming of the robot in an industrial robot programming language and provide a nice and intuitive interface that presents a realistic 3D view of the robot. However, those tools are not truly useful for education, since it is not possible to observe or modify the kinematic or dynamic parameters of the robots. For example, it is important that the students observe the evolution of the coordinates of the end effector, while, at the same time, views the joint coordinates at each time step. It is worth mentioning the tool RobotStudio [18] focussed on the simulation of ABB GmbH robots. The software allows to obtain a very realistic simulation of the arms in its workspace along with a set of end tools such as soldering torches or grippers and auxiliary equipment. The user may include target points and create a RAPID program interactively. Alternatively, any RAPID program can be simulated to test its integrity and obtain process times. However, though complete, the software is not oriented for educational purposes and does not easily clarify concepts such as direct or inverse kinematics. The algorithms employed in the simulation or the robot parameters cannot be changed, whereas new robot models cannot be added.

Other commercial tools such as Easy-Rob [22] allows the simulation of robotic arms of different manufacturers, model any robotic cell or simulate any robotic language [23]. However, new robots cannot be easily included. In addition, the student cannot observe the evolution of the joint coordinates as a function of the end effector's position and orientation nor can be observed the forces or torques at each joint.

Due to the reasons stated before, we considered the idea of developing an educational tool that improved some of the deficiencies of the before mentioned tools. The first differentiating feature of ARTE compared to other Matlab-based similar tools, such as [11], is the capacity to observe the robots in three dimensions with a detailed representation. We have observed that, in this way, the student obtains a better comprehension of most of the concepts, such as the utility of the Denavit–Hartenberg reference systems attached to each link. For example, the different solutions of the inverse kinematic can be easily observed by the student with a solid representation of the robot. Although there exist other tools that allow a 3D realistic representation of the robot arms [11], we consider that using several different software tools requires a great amount of time until the student is able to use the properly, thus reducing the really useful time used to comprehend the concepts. In our case, Matlab is used throughout different subjects and thus the students know it deeply. The plotting capabilities of Matlab can be used to easily represent and visualize the variables of the arm. The second differentiating element of the library is the clearness in the implemented code. The Matlab code has been written so that it can be easily read and understood. In addition, the toolbox has been designed to allow students to add new robots to the library while solving the main kinematic and dynamic problems. We have realized that, in this way, the students are more motivated when carrying out the practices, since they obtain tangible results for their effort. In addition, the toolbox is accompanied with a set of documents that sketch 6 different practical sessions. These documents guide the student and allow him to go into the subject in deep autonomously.

MAIN FEATURES OF THE TOOLBOX

The toolbox presents the following main features:

(a) Denavit–Hartenberg's representation of the robotic manipulator.
(b) Capacity to visualize the position, velocity and acceleration of the joint variables of the robot when it performs a movement. In addition, the capacity to represent the velocity of the end effector during the simulation.
(c) Capacity to visualize joint forces and torques during the performance of any movement.
(d) Realistic 3D representation of the robot link as solid objects.
(e) Path planning of joint trajectories.
(f) Programming the robot in an industrial language. Step-by-step simulation of the program.
(g) Programming of the robot using a virtual teach pendant. Creation of target points and way points using a graphical interface.
(h) Easy inclusion of new robots.
(i) Realistic representation of the robot in a robotic cell with auxiliar equipment.

In Table 1, we evaluate some of the software tools mentioned before from the point of view of these features (a, b, c, d, etc.). It is important to highlight that except for ARTE, none of the other evaluated tools possesses all these capabilities. The combination of several software tools may allow the accomplishment of all these attributes, but it may force the student to learn different programs, thus reducing the useful time to learn the important concepts. In the Bachelor and Master studies in Industrial Engineering, where the tool is currently being deployed, Matlab is a common program used in several subjects, thus the adaptation time to ARTE is generally short.

One of the key features of the toolbox is the 3D realistic representation of the toolbox. To the best of our knowledge, ARTE is the only Matlab-based robotic toolbox that offers this capability. As will be shown in the toolbox assessment, this feature is nicely received by students. In addition, a very important feature of the toolbox is the possibility to add new robots in an easy manner. In this way, the students, as a part of their laboratory sessions, include a new robot in the library. In order to do so, the student uses standard graphical files (in Stereo Litography file format, STL), solves the direct and inverse kinematic problem and includes the dynamic parameters of the robot. The educational outcomes of this activity are very interesting: We have observed that it increases the motivation of the student, since he obtains a tangible result that can be checked by the realistic simulation of the robot. Moreover, the students contributes to a continuously growing project and this encourages him in the fulfillment of his task. An example of the realistic 3D representation of the robot is presented in Figure 1 where the robot is presented in a robotic cell and can be programmed in a realistic way.

The ARTE tool has been deployed during the last two academic years. It contains a significant number of robotic manipulators, most of them corresponding to industrial models. Table 2 presents a list of the currently supported robots, as well as the features implemented in each one. It is worth noting that the list includes industrial robots from different manufacturers. In this sense, we believe that this is positive from an educational point of view, since the study is not focussed on a particular manufacturer. A total of 42 robots are available in the library, since several arm models have versions and subversions. As well, classical manipulators, such as the Unimate Puma 560 or the Stanford arm have been included, since their kinematic and dynamic parameters have been well studied and determined. Thus, they are very well suited for educational purposes. Finally, the library includes other elements that are of great importance such as robotic manufacturing cells, parallel or angular grippers, vacuum grippers, conveyor belts and manufacturing tables.

Table 1 Comparison of Software Tools for the Simulation of Robotic Manipulators

Name	Features	Comment	License
Robotics Toolbox for Matlab	a b c e h	www.petercorke.com	LGPL
SimMechanics toolbox	b c d e i	www.mathworks.es	commercial
RobotStudio	d f g i	www.abb.com	commercial
KUKA, Sim	d f g i	www.kuka.com	commercial
Solutions in Motion	d f g i	www.motoman.com	commercial
Easy-Rob 3D	d f g	www.easy-rob.de	commercial
ARTE	a b c d e f g h i	arvc.umh.es/arte	LGPL

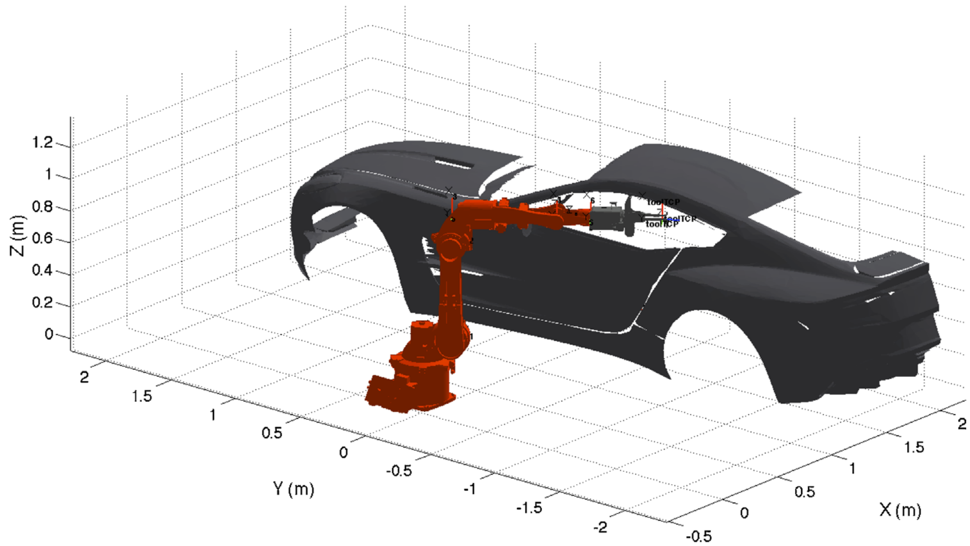


Figure 1 The figure presents a KUKA robot in a manufacturing cell in a welding application. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

COURSE SEQUENCE AND STUDENT BACKGROUND

The ARTE toolbox described here was designed for an introductory course for serial robotics. The tool is used in the courses that involve the teaching of robotics at Miguel Hernandez University (Spain), which is taught by the System Engineering and Automation department in two different studies: the degree in Electronic Engineering and Industrial Automation, and in Industrial Engineering (please refer to http://en.umh.es for more information). The Robotics course receives different names in the studies mentioned, as can be seen in Table 3. The main topics covered include direct and inverse kinematics for serial manipulators, dynamics and path planning.

The tool is deployed in the different courses and occupies the number of credits described in Table 3. Depending on the number of hours available at each course the students are able to carry out all the practicals or just a subset of them.

The robotics courses in which ARTE is used have no prerequisites. It is taught in the fourth year of their studies. The course requires only a basic mathematical background that the students acquire during their first and second year in engineering studies. In addition, the students possess a basic formation in physics that they have also acquired during the first and second year in engineering.

The introduction to serial robotics in which the ARTE tool is used deals with the following educational goals:

- Direct and inverse kinematics.
- Direct and inverse dynamics.
- Path Planning.
- Control and simulation.
- Robot programming.

SOFTWARE ARCHITECTURE

Figure 2 presents the software architecture implemented in ARTE. The tool- box is designed around the robot structure depicted in the figure. The robot structure stores the kinematic and dynamic parameters of the robot, as well as the graphic representation. This data representation was chosen for its clarity and easy usage. As depicted in Figure 2, the robot structure is passed as an argument for the Matlab functions that solve the kinematic and dynamic problems. In addition, the structure is also used by the Matlab

Table 2 List of Robots Supported by the Library

Manufacturer	Model
ABB	IRB52, IRB140, IRB1600, IRB1600iD, IRB2400, IRB4400, IRB6620, IRB6620LX, IRB6650, IRB7600
ADEPT	VIPER S 1700D
EPSON	PROSIX C3 A601C KR5 arc
FANUC	LR MATE 200iC
KUKA	KR16 arc HW, KR30 L16 2, KR1000-1300 Titan, KR5 2 arc HW, KR5 arc, KR5 scara R350 Z200, KR5 sixx R650, KR5 sixx R850, KR6 2, KR30 jet, KR90 R2700 pro, KR90 R3100 extra
MITSUBISHI	PA-10 6DOF, RV 6S
STAUBLI	RX160L
CLASSIC ROBOTS	UNIMATE PUMA 560, Stanford arm
EXAMPLE ROBOTS	2 DOF PLANAR ROBOT, 3 DOF PLANAR ROBOT, PRISMATIC ROBOT, GENERAL SCARA ROBOT
ROBOTIC END TOOLS	Angular grippers, parallel grippers, vacuum grippers, spot welding tools
ROBOTIC CELLS AND AUXILIAR EQUIPMENT	Bumper cutting cell, car bodywork cell, conveyor belts, tables

Table 3 Deployment of the Tool by Subjects

Studies	Course	Credits	#students
Degree in Electronic Engineering and Industrial Automation	1770 ROBOTICS	7.5	45
Industrial Engineering	4747 ROBOT CONTROL AND SENSORIAL SYSTEMS	12[6]	63

functions called by the RAPID interpreter. Finally, the robot structure shares its parameters with the Simulink environment and the "teach" GUI. The graphical properties of the structure are plotted on a standard Matlab figure.

PRACTICAL SESSIONS AND DEPLOYMENT OF THE TOOL

In this section, we present a series of activities that can be carried out using the presented tool. The sessions are designed to introduce the topics gradually in combination with the necessary theory provided in the lectures. Each of the practical sessions is described in a document, available at http://arvc.umh.es/arte, where the theoretical bases are summarized. As well, the documents pose a series of activities and exercises that use the library. Each of the sessions is described next in detail. In addition, there exist a transversal practice that suggests the students to include a new robot to the library, thus solving the direct and inverse kinematics, dynamics and including the necessary graphic files. Next, we present the proposed sessions:

- Session 1: Direct kinematics (2 h).
- Session 2: Inverse kinematics (4 h).
- Session 3: Inverse dynamics (2 h).
- Session 4: Direct dynamics (2 h).
- Session 5: RAPID programming (4 h).
- Session 6: PID control in simulation (4 h).
- Session 7: Adding a new robot to the library (6 h).

The duration of each practical session considers the total time spent by the student using the tool. This time may be spent by the student at the laboratory and at his own home. Table 3 presents

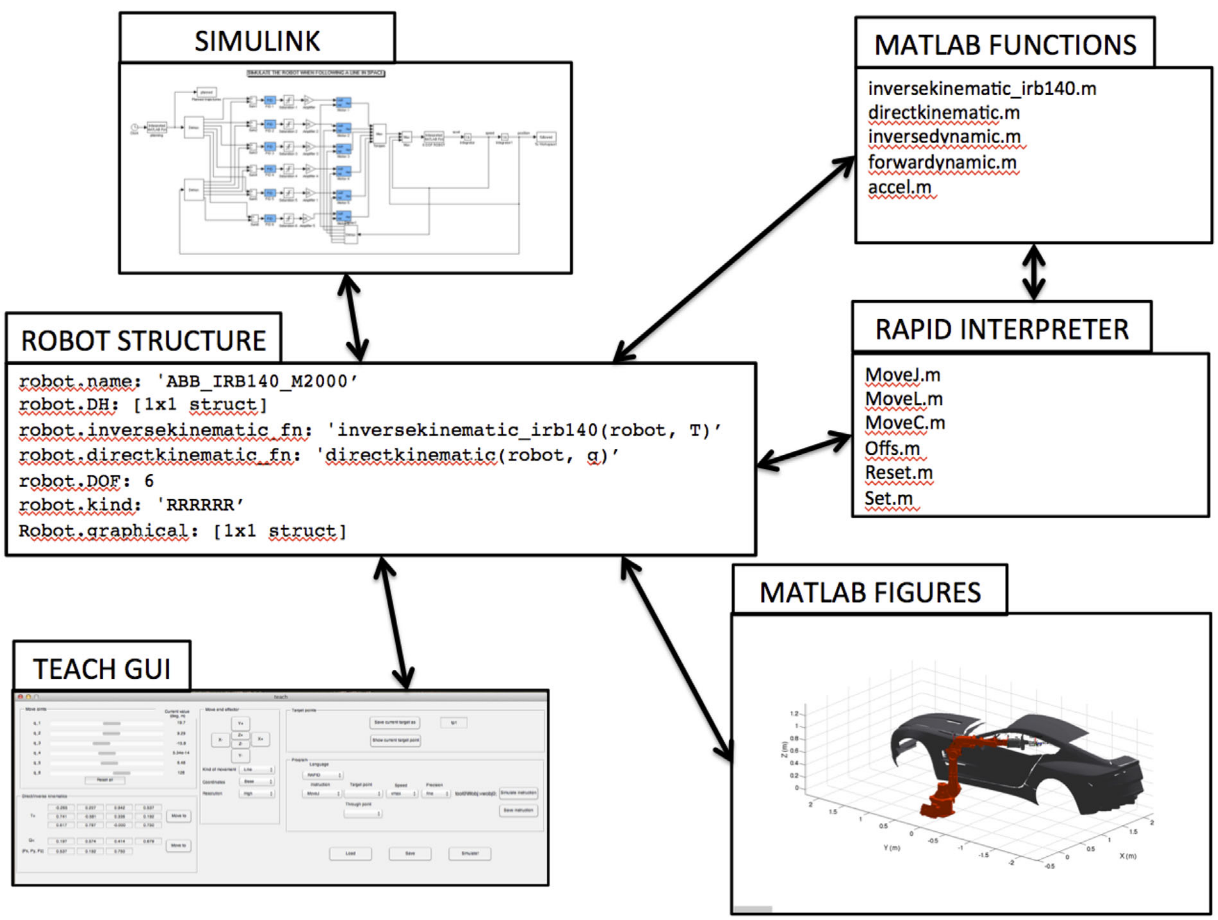


Figure 2 The figure presents the software architecture implemented in ARTE. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

the number of credits and registered students for each of the subjects where the toolbox has been deployed. Carrying out the 6 sessions requires a minimum of 24 h whereas the theoretical lectures require approximately 40 h (please, refer to http://en.umh.es for more information about the studies).

Session 1: Direct Kinematics

This practice allows the student to understand the basic concepts on homogeneous transformation matrices and the standard D-H parameters of a robot. The practice poses different exercises to explain these concepts. The student solves the D-H parameters for different robots as an example. The next commands exemplify a particular activity of this practice:

```
>> robot=load robot('abb','IRB6620');
>> T = directkinematic(robot, [pi/2 pi/4 pi/4 0 0 0])
>> drawrobot3d(robot, [pi/2 pi/4 pi/4 0 0 0])
```

where the matrix T represents the position and orientation of the robot's end effector and the last command is used to make a 3D representation of the arm as presented in Figure 3. The D-H parameters of each arm are stored in a parameters file. Next, we show a fragment of code that defines the D-H parameters for the ABB IRB 140 robot arm:

```
function robot = parameters()
robot.name= ' IRB140 M2000';
robot.DH.theta= '[q(1) q(2)-pi/2 q(3) q(4) q(5) q(6)]';
robot.DH.d='[0.352 0 0 0.380 0 0.065]';
robot.DH.a='[0.070 0.360 0 0 0]';
robot.DH.alpha= '[-pi/2 0 -pi/2 pi/2 -pi/2 0]';
```

Session 2: Inverse Kinematics

During this practice, the student must solve the inverse kinematics of an example robot. Next, the student has to solve the inverse kinematic problem for the new robot that is being included in the library. Each robot has a specific Matlab function that solves the inverse kinematic problem and is stored in the own directory of the robot. The next commands exemplify one of the activities of this practice:

```
robot = load robot('abb','IRB140');
T = directkinematic(robot, [pi/2 pi/4 pi/4 0 0 0])
drawrobot3d(robot, [pi/2 pi/4 pi/4 0 0 0])
qinv = inversekinematic(robot, T)
```

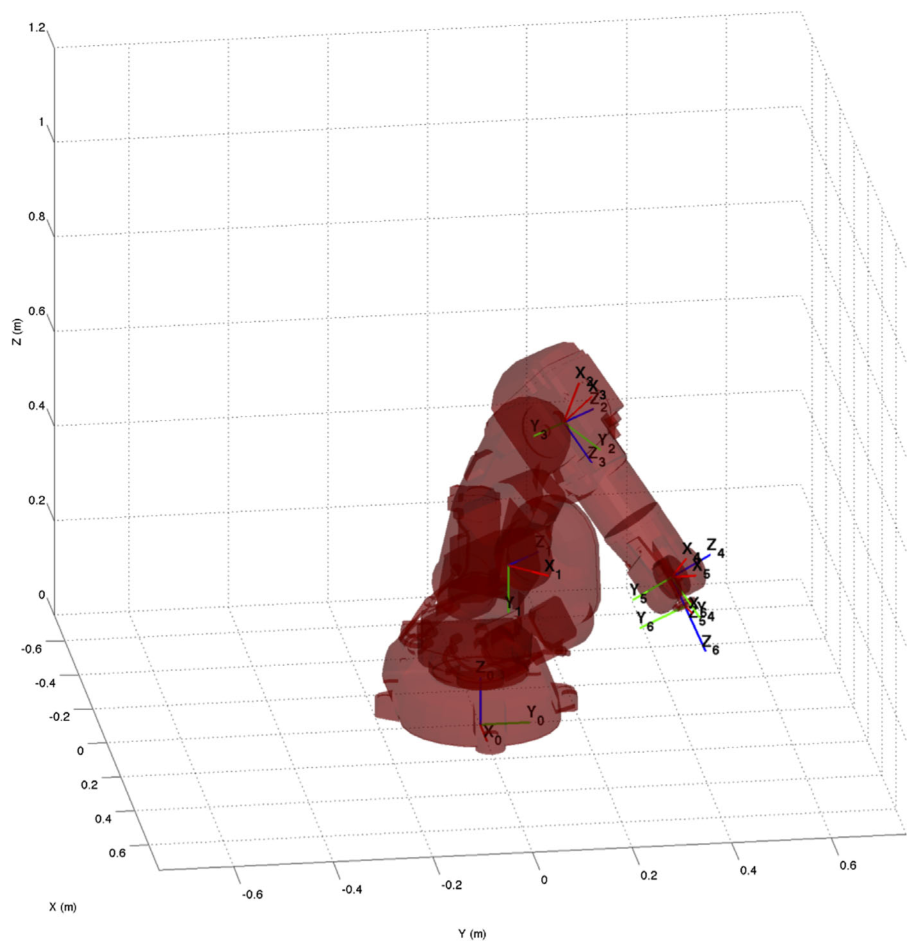


Figure 3 The figure presents one of the robots available in the library. In this case, the links are drawn in transparent mode, so that the D-H reference systems can be clearly observed. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

where the variable qinv stores, in this case, the 8 eight possible solutions that bring the arm to the same position and orientation. The student is encouraged to test whether all the solutions are valid and he must check that the achieved position and orientation is the same for all of them.

Session 3: Inverse Dynamics

During the practice, the toolbox allows the student to clarify the concept of inverse dynamics: given the movement state of the arm, the library computes the torque or force at each joint. The student is able to test the variation of the forces and torques when, for example, the robot carries different loads.

Session 4: Direct Dynamics

This practice allows to understand the concept of direct dynamic model. The ARTE library is able to compute the acceleration at each joint given the movement state of the arm and the torques or forces applied by means of the "method 1" described in [24]. The accelerations computed in this way are integrated by means of the ode45 Matlab function. The results are presented to the student in the form of plots showing the evolution of the joint positions, velocities and accelerations, as well as a 3D simulation of the robot movement.

Session 5: RAPID Programming

As mentioned before, the GUI application described in Course Sequence and Student Background allows to create and simulate programs written in RAPID. The programs are translated to Matlab using the ARTE library functions, so that the student can simulate step by step the trajectories, velocities and accelerations of each joint during the movements. After the debugging and simulation process, the student is able to test the program in a real ABB IRB140 real placed at our laboratory. In this way, the student corroborates the validness of his programming task.

Session 6: PID Control in Simulation

In this session the student learns the basic concepts related to the independent joint control using a basic PID (Proportional Integral Differential) controller. The whole arm is simulated using the tool. The simulation includes the mechanical dynamic model and a dynamic model of the electric motors as well. The students are encouraged to tune the PID parameters of the controller and, in addition, test the trajectories performed by the robot under different conditions when changing the parameters.

Session 7: Adding a New Robot to ARTE

As a transversal activity during the six practical sessions, the student must include a robot of his election to the library. This activity is made in teams of two or three people. In order to add a new robot, each group has to solve the following tasks: first, during the first practical session, they have to choose an industrial robot and obtain the graphic files from the manufacturer's site. Next, the students have to solve the direct kinematic problem and adapt the graphic files to the particular D-H parameters of the arm. In addition, the students solve the inverse kinematic problem. In some cases, the dynamic parameters are also included. Figure 4 presents some of the robots included by students. In this way, we have observed that the students are highly motivated since they believe that their work will be used by other students.

Graphical User Interface–Teach Pendant

The toolbox has a Graphical User Interface (GUI) that simulates a typical robotic teach pendant and is named "teach". It allows the student to move the robot in an intuitive way with most of the options included in industrial teach pendants. The robot can be moved joint by joint by using a set of sliders. In addition, the robot can be moved in inverse kinematics by using a set of buttons named X−, X+, Y−, Y+, Z− and Z.

Robot Programming RAPID Language

The GUI application is capable of translating the RAPID instructions to Matlab instructions. In this way, any RAPID program can be translated to a Matlab script that can be easily edited and simulated step by step. Next, as an example, we present a small RAPID program:

```
MODULE Module1
  VAR extjoint xt:
  =[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09];
  CONST robtarget tp1:
  =[[0.5,0,0.7], [0,0.707,0,0.707],[0,0,0,0],xt];
  CONST robtarget tp2:
  =[[0.6,0.1,0.7],[0,0.707,0,0.707],[0,-1,0,0],xt];
  CONST robtarget tp3:
  =[[0.6,0,0.4],[0,0.707,0,0.707],[0,-2,1,0],xt];
  CONST robtarget tp4:
  =[[0.8,0,0.2],[0,0.707,0,0.707],[0,-2,1,0],xt];
  MoveL tp3,vmax,fine,tool0\Wobj:=wobj0;
  MoveJ tp2, vmax , fine, tool0\Wobj:=wobj0;
  MoveJ tp1, vmax , fine, tool0\Wobj:=wobj0;
  MoveC tp2, tp3, vmax, fine, tool0\Wobj:=wobj0;
ENDMODULE
```

The GUI application translates the afore program to a series of Matlab instructions and functions of the library:

```
%BEGINMODULE
%BEGINTARGETPOINTS
  xt=[9E+9,9E+9,9E+9,9E+9,9E+9,9E+9];
  tp1=[[0.5,0,0.7],[0.0,0.707,0,0.707],[0,0,0,0],xt];
  tp2=[[0.6,0.1,0.7],[0.0,0.707,0,0.707],[0,-1,0,0],xt];
  tp3=[[0.6,0,0.4],[0.0,0.707,0,0.707],[0,-2,1,0],xt];
  tp4=[[0.8,0,0.2],[0.0,0.707,0,0.707],[0,-2,1,0],xt];
%ENDTARGETPOINTS
  robot=MoveL(robot,tp3,'vmax','fine',robot.
    tool0,robot.wobj0);
  robot=MoveJ(robot,tp2,'vmax','fine',robot.
    tool0,robot.wobj0);
  robot=MoveJ(robot,tp1,'vmax','fine',robot.
    tool0,robot.wobj0);
  robot=MoveC(robot,tp2,tp3,'vmax','fine',robot.
    tool0,robot.wobj0);
%ENDMODULE
```

Once this action is done, the program can be edited using the Matlab editor. This capability of the library allows the students to modify the program, debug and execute it easily.

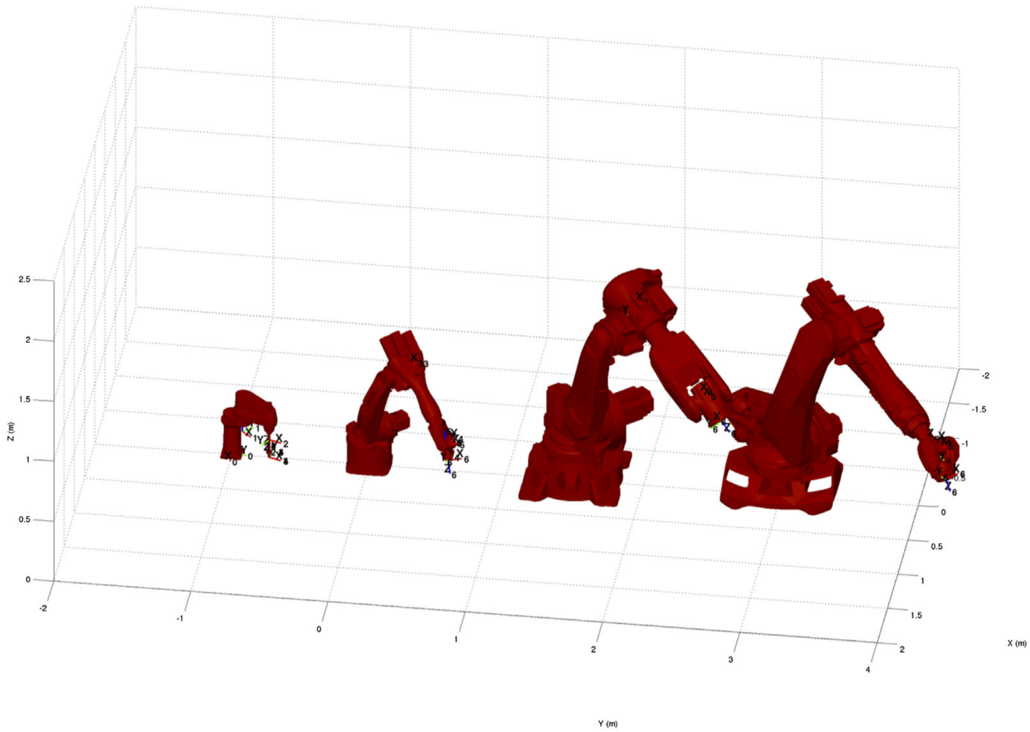


Figure 4 The figure presents other robots already included in the library. From left to right: KUKA R350 Z200, KUKA KR5 arc, ABB IRB6620 and KUKA KR90 R2700 pro. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

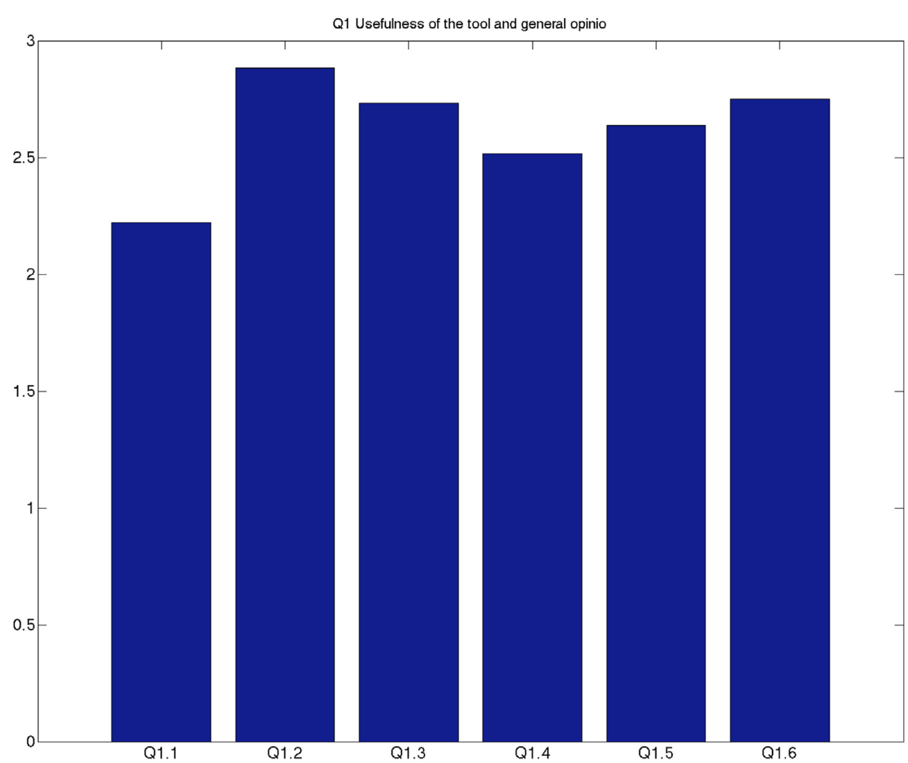


Figure 5 The figure presents the results for the questions named Q1. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

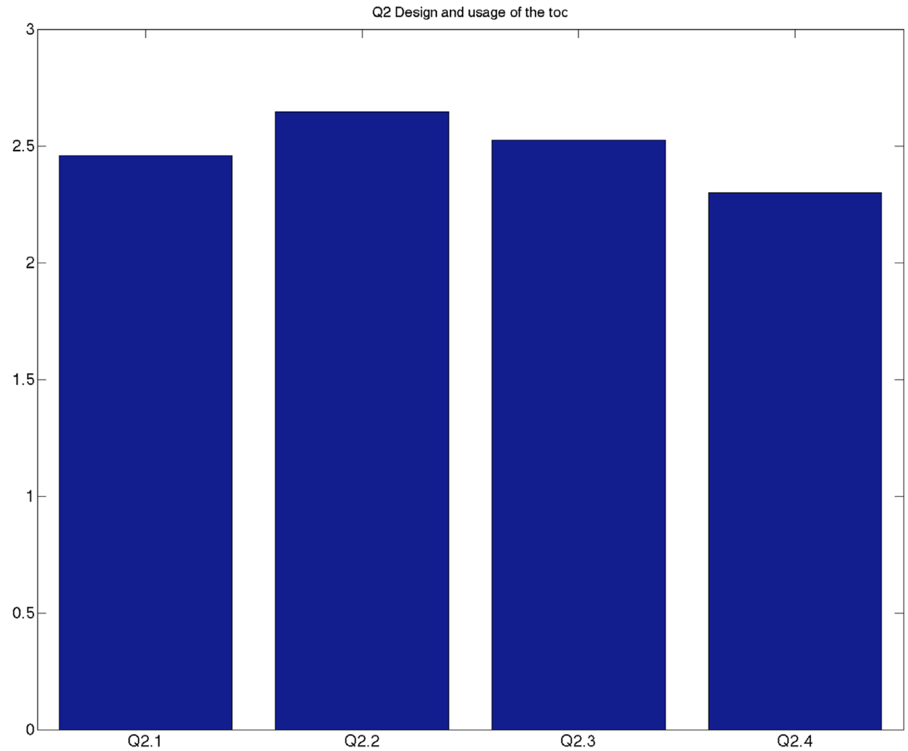


Figure 6 The figure presents the results for the questions named Q2. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

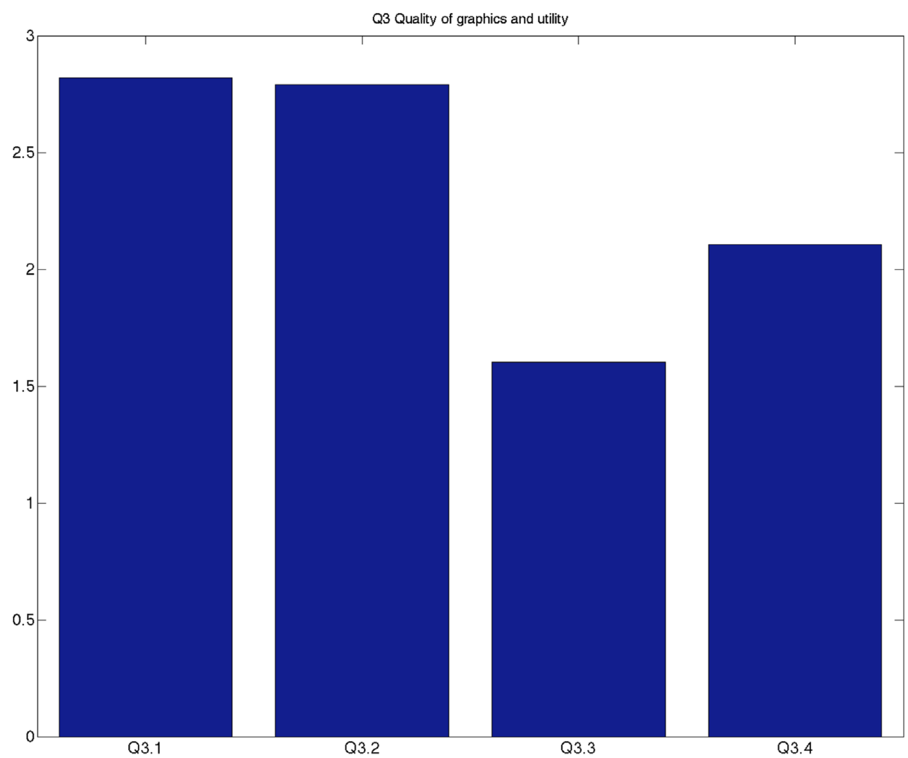


Figure 7 The figure presents the results for the questions named Q3. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

EVALUATION OF THE TOOL EFFECTIVENESS

We conducted a personal survey to obtain the opinion of the student about the presented software tool. On the one hand, the library was analyzed in terms of usefulness, design and usage as well as the quality of the graphics presented. In addition, the subjective difficulty of the laboratory sessions was also analyzed. In addition, we asked the students about the time deployed by the student using the tool. The surveys were designed to establish the relation of the student perception of the system, the time spent on the tool and their learning achievements. The topics of the survey follow the guidelines presented in [17] and deployed in [13,25].

The tests were conducted by students of two different subjects corresponding to two different studies (Degree in Electronic Engineering and Industrial Automation and Industrial Engineering). The number of registered students in each subject is found in Table 3, as well as the number of credits. The tool was analyzed to obtain information regarding the following items:

- Q1) Usefulness of the tool in their learning process and general opinion of the student.
- Q2) The design and usage of the tool.
- Q3) The quality of the graphics and their utility in the educational process.
- Q4) The difficulty of the laboratory sessions.
- Q5) How many hours did you spend using the tool?
- M1) Marks obtained by students.

The item Q5 gathered the time spent by the students using the tool for each of the laboratory sessions. This time includes the time spent by the student at home. The last item M1 is a result of the correctness of the answers presented by the students after the laboratory sessions, as well as the marks obtained by the students in the final exam.

First, the usefulness of the tool was analyzed in order to find whether the tool helped the students understand the concepts seen during theory lessons. With this purpose, the students should answer the following questions:

- Q1.1) In general, do you think that ARTE has helped you to understand the concepts explained during theory lessons?
- Q1.2) Do you regard the toolbox as a useful tool to understand the direct/inverse kinematics of serial robotic mechanisms?
- Q1.3) Do you consider that the toolbox is useful to understand the direct/inverse dynamics of serial robotic mechanisms?
- Q1.4) Do you think that the toolbox is useful to program and simulate a robotic arm in RAPID language?
- Q1.5) Did you find interesting to add your own robot to the ARTE toolbox?
- Q1.6) In general, I have found the practical sessions carried out with ARTE interesting and useful.

Each student gave a mark to each of the mentioned items, from 0 (fully disagree), 1 (disagree), 2 (agree) and 3 (fully agree). The mean result obtained at each of the questions is presented in Figure 5. In general, all the items received an acceptable result. In general, it can be said that, according to the results, the tool was found useful by students. It can be observed that question Q1.3 received a low mark. This result can be explained by the high complexity of the forward dynamic problem. Next, the design of the tool was analyzed from the students' point of view:

- Q2.1) Did you find the tool easy to use?
- Q2.2) Did you find the toolbox well organized and structured?
- Q2.3) Is the representation of the variables involved in the kinematic analysis of the arms clear?
- Q2.4) Did you find the tool well documented?

Each student gave a mark to each of the mentioned items, from 0 (fully disagree), 1 (disagree), 2 (agree) and 3 (fully agree). The mean result obtained at each of the questions is presented in Figure 6. The students consider, in general, that the tool is easy to use and well organized.

In addition, we would like to measure the usefulness of the 3D graphic files of the robots, as well as the multimedia files.

- Q3.1) The graphical capabilities of the toolbox are useful to understand the direct and inverse kinematic problem.
- Q3.2) The "teach" application is easy to use and allows to move the robot intuitively.
- Q3.3) The videos enclosed in the laboratory sessions where helpful to accomplish the tasks.
- Q3.4) The documents that come along each laboratory session where helpful to understand the practical sessions.

These questions where considered from 0 (fully disagree), 1 (disagree), 2 (agree) to 3 (fully agree). Figure 7 presents the mean result at each of the items. It can be said, according to the results, that the tool was found useful by students. The items named Q3.1, Q3.2 and Q3.4 received nice results. However, the video tutorials for the laboratory sessions (Q3.3) did not receive a nice acceptance.

In addition, a measure of the difficulty of the laboratory sessions and the time spent by students was carried out. Thus, for each of the 6 laboratory sessions, the following questions were asked:

- Q4) Did you find the sessions with ARTE difficult?

Table 4 presents the difficulty perceived by the student from 3 (easy) to 0 (difficult). Table 4 presents the mean results. It can be said that, in general, the difficulty of the practicals is low.

Finally, the personal survey allowed us to find the time spent by the students using the tool for each of the laboratory sessions:

- Q5) How many hours did you spend using ARTE for each session?

The total time spent by students, both at laboratory and at home is presented in Table 5. In general, we can be observe that the time is very close to the planned time for each laboratory session. Sessions 2, 5 and 6 last slightly longer than the planned time and will be shortened in the future. The knowledge of this time is of

Table 4 Evaluation of the Tool Difficulty Perceived by Students (Q4; 0, Easy, 3 Difficult) for Each of the Laboratory Sessions

Session	Q4: difficulty (0–3)
Session 1	0.45
Session 2	2.50
Session 3	0.82
Session 4	2.30
Session 5	1.31
Session 6	0.51
Session 7	0.53

Table 5 Evaluation of the Tool Q5. Time Spent by Students (hours) for Each of the Laboratory Sessions

Session	Q5: mean time (hours)
Session 1	1.53
Session 2	4.22
Session 3	1.77
Session 4	3.05
Session 5	3.68
Session 6	4.41
Session 7	6.79

Table 6 Evaluation of the Tool (M1): Correctness of the Answers

Session	Mean qualification (0–10)	Standard deviation
Session 1	9.32	0.27
Session 2	6.15	0.82
Session 3	7.50	1.33
Session 4	6.29	2.57
Session 5	9.32	0.98
Session 6	9.83	0.67
Session 7	9.68	0.45

paramount importance, since it permits us to foresee the time needed by the student at home to fulfill their work and thus configure correctly the ECTS (European Credit Transfer and Accumulation System) credits.

Finally, an evaluation of the knowledge acquired by the students was carried out (M1). For this purpose, once the students have finished all the practical sessions they must present a document answering a list of questions and problems. In particular, each session has, at least, 5 questions, and the teacher rates the answers from 0 to 10, being 10 a perfect answer. The correctness of the answers for the different concepts was tested. Table 6 presents the results in terms of correctness. A mean value and standard deviation are shown. Sessions 1, 5, 6 and 7 were easier to understand by the students and, thus, obtained higher marks. However, sessions 2 (inverse kinematics) and 4 (forward dynamics) obtained lower qualifications, since the concepts involved in them are more difficult to comprehend.

CONCLUSIONS

We have presented a new toolbox addressed to the teaching of robotic manipulators in bachelor and master studies. The library has been implemented as a Matlab toolbox and possesses some differentiating features compared to other previous tools. The most important features consist of the possibility of making a realistic representation of the robot in three dimensions, the capability of simulating programs written in RAPID language and the ease in the inclusion of new robots to the library.

We have described the tool and a set of practical sessions that are based on it. The toolbox was deployed in two different subjects corresponding to bachelor and master studies during the last year.

Moreover, the tool has been evaluated by the students by means of an anonymous survey that included different items, such as the usefulness of the tool, its design, the quality of the graphics included and the difficulty of the laboratory sessions. The evaluation demonstrated that the students consider that the tool helps them understand the theory concepts, and, at the same time, the practicals are not very difficult to complete.

In general, we consider that the tool is useful in the teaching-learning process for robotics. We have found that the idea of adding robots to the library by part of the students is interesting. In general, the students feel highly motivated when they observe that his effort is translated to a tangible result that will be used by others. Finally, as a future work, we plan to include models of parallel robots to the toolbox.

REFERENCES

[1] S. Dormido, Control learning: present and future, Ann Rev Control 28 (2004), 115–136.

[2] F. A. Candelas, S. R. Puente, F. Torres, F. Ortiz, P. Gil and J. Pomares, A virtual laboratory for teaching robotics, Int J Eng Educ 19 (2003), 363–370.

[3] M. A. González-Palacios, E. A. González-Barbosa and L. A. Aguilera-Cortés, SnAM: A simulation software on serial manipulators, Engineering with Computers Springer 29 (2013), 87–94.

[4] P. R. Turner, Teaching scientific computing through projects, J Eng Educ 90 (2001), 79–83.

[5] S. S. Joshi, Development and implementation of a Matlab simulation project for a multidisciplinary graduate course in autonomous robotics, Comput Appl Eng Educ 12 (2004), 54–64.

[6] C. A. Jara, F. A. Candelas, S. T. Puente and F. Torres, Hands-on experiences of undergraduate students in automatics and robotics using a virtual and remote laboratory, Comput Educ 57 (2011), 2451–2461.

[7] C. A. Jara, F. A. Candelas, J. Pomares and F. Torres, Java software platform for the development of advanced robotic virtual laboratories, Comput Appl Eng Educ 21 (2011), E14–E30.

[8] S. Dormido, H. Vargas, J. Sanchez, R. Dormido, N. Duro, S. Dormido-Canto and F. Morilla, Developing and implementing virtual and remote labs for control education: The UNED pilot experience. In: Proceedings of the 17th IFAC World Congress, Elsevier, Seoul, Korea, 2008.

[9] V. Potkonjak, M. Vukobratovíc, K. Jovanovíc and M. Medenica, Virtual mechatronic/robotic laboratory? A step further in distance learning, Comput Educ 55 (2010), 465–475.

[10] M. Kamel, A. Taylor and A. Breton, A synchronous communication experiment within an on-line distance learning program: A case study, Telemed J e-Health 1 (2005), 583–593.

[11] P. Corke, A robotics toolbox for Matlab, IEEE Robotics and Automation Magazine 3 (1996), 24–32.

[12] J. Francés, M. Pérez-Molina, S. Bleda, E. Fernandez, C. Neipp and A. Beléndez, Educational software for interference and optical diffraction analysis in fresnel and fraunhofer regions based on matlab GUIS and the FDTD method, IEEE Transactions on Education 55 (2012), 118–125.

[13] G. Jovanovic Dolecek, Matlab-based program for teaching autocorrelation function and noise concepts. Education, IEEE Transactions on 55 (2012), 349–356.

[14] M. Cakir and E. Butun, An educational tool for 6-DOF industrial robots with quaternion algebra, Comput Appl Eng Educ 15 (2007), 143–154.

[15] C. Kohl, H. Tan, K. Tan, L. Fang, F. Fong, D. Kan, S. Lye and M. Wee, Investigating the effect of 3D simulation-based learning on the motivation and performance of engineering students, J Eng Educ 99 (2010).

[16] B. Kolloeffel and T. de Jong, Conceptual understanding of electrical circuits in secondary vocational engineering education: Combining traditional instruction with inquiry learning in a virtual lab, J Eng Educ 102 (2013), 375–393.

[17] O. A. Iglesias, C. N. Paniagua and R. A. Pessacq, Evaluation of university educational software, Comput Appl Eng Educ 5 (1997), 181–188.

[18] ABB GmbH. http://www.abb.com/.

[19] Mitsubishi Electric. http://mitsubishirobotics.com.

[20] The MathWorks. http://www.mathworks.com.

[21] R. S. Hartenberg and J. Denavit, A kinematic notation for lower pair mechanisms based on matrices, J Appl Mech 77 (1955), 215–221.

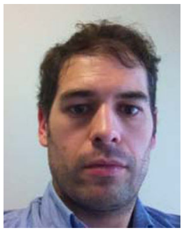
[22] Easy-Rob. 3D Robot Simulation Tool ver. 6.0. www.easy-rob.de.

[23] L. Zlajpah, Simulation in robotics, Math Comput Simulation 79 (2008), 879–897.

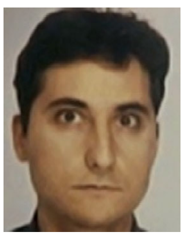
[24] M. W. Walker and D. E. Orin, Efficient dynamic computer simulation of robotic mechanisms, ASME J Dyn Syst Meas Control 104 (1982), 205–211.

[25] G. López-Nicolás, A. Romeo and J. J. Guerrero, Active learning in robotics based on simulation tools, Comput Appl Eng Educ 22 (2011), 509–515.

BIOGRAPHIES



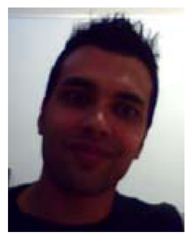
Arturo Gil Aparicio received the MEng degree in Industrial Engineering from Miguel Hernández University (UMH), Elche, Spain, in 2002, receiving also the Best Student Academic award in Industrial Engineering by the UMH. He obtained the PhD degree in 2008, entitled: "Cooperative construction of visual maps by means of a robot team". Since 2003, he works as a lecturer and researcher at the UMH, teaching subjects related to Control and Computer Vision. His research interests are focused on mobile robotics, visual SLAM and cooperative robotics. He is currently working on techniques to build visual maps using teams of mobile robots.



Óscar Reinoso García received the MEng degree from the Polytechnical University of Madrid (UPM), Madrid, Spain, in 1991. Later, he obtained the PhD degree in 1996. He worked at Protos Desarrollo S.A. company in the development and research of artificial vision systems from 1994 to 1997. Since 1997, he works as a professor at Miguel Hernández University (UMH), teaching subjects related to Control, Robotics and Computer Vision. His research interests are in mobile robotics, climbing robots and visual inspection systems. He is member of CEA–IFAC and IEEE.

José María Marín obtanined a degree in Mechanical Engineering from U.N.E.D., Madrid, in 2000, and the Ph.D. degree in Mechanical Engineering from Carlos III de Madrid University, in 2009. Since 2001, he has been at the Miguel Hernandez University, as professor in Theory of Mechanisms. He is interested in mechanic vibrations, roller bearing and gears.

Luis Payá Castelló holds a MEng degree in Industrial Engineering (2002) and a PhD at Miguel Hernández University (UMH), Elche, Spain (2014). He works currently as a lecturer and researcher at UMH, where he teaches some subjects about automatic control of dynamic systems. His current research interests include map creation and navigation of mobile robots using computer vision and appearance-based techniques and development and test of remote laboratories. He has authored several papers and communications in the cited topics, presented in international scientific conferences and peer-reviewed journals.



Javier Ruiz Ramírez received the MEng degree in Industrial Engineering from Miguel Hernández University (UMH), Elche, Spain, in 2007, receiving also the Best Student Academic award in Industrial Engineering by the UMH. He obtained a PhD in 2014, entitled: "Experimental characterization and modelling of the binomial distribution system-drift eliminator in cooling towers". Since 2008, he works as a lecturer and researcher at the UMH, teaching subjects related to Fluid Mechanics and Heat Transfer. His research interests are focused on heat dissipation systems, CFD and renewable energies. He is currently working on improving the efficiency of air conditioning systems.