Citation: G. López-Nicolás, A. Romeo, J. J. Guerrero. Active learning in robotics based on simulation tools. Computer Applications in Engineering Education, vol. 22, no. 3, pp, 509-515, 2014.

Active learning in robotics based on simulation tools

G. López-Nicolás, A. Romeo and J. J. Guerrero

Dept. de Informática e Ingeniería de Sistemas, Universidad de Zaragoza
C/ María de Luna 1, E-50018 Zaragoza, Spain
{gonlopez, romeo, jguerrer}@unizar.es

Abstract

Simulation is a powerful tool that has a very important role in the different areas of research, development and education. In particular, simulation tools are widely used in robotics. In this paper, we present an active learning experience based on simulation tools in the framework of robotics courses taught at engineering degrees. It is well known that the field of robotics involves many disciplines taught in higher education, as mechanics, electronics or computer science. In the context of a robotics course for an Industrial Engineering Master Degree, the different topics addressed can be highly benefited by the use of simulation tools. Additionally, the benefits of active learning activities have been widely acknowledged and discussed. Here, we describe the methodology and two simulation tools, RobotScene and SGRobot, developed for this project-based learning activity. This activity has been carried out successfully during several years; and our experience is that it motivates the students and improves their understanding of the theoretical concepts involved. We also present the academic results that support the benefits of the activity presented.

Keywords:
Computer simulation, Simulation-based learning, Learning support, Robotics learning

1. Introduction

In a broad perspective Industrial Robotics is an inherently cross-disciplinary subject whose in-depth knowledge involves a variety of tasks such as modeling, design, simulation, control, optimization, and performance evaluation.

Preprint submitted to Computers & Education *December 21, 2009*

In addition, successful application of industrial robotics involves the integration of different tools from related disciplines. The multidisciplinary issues involved, like kinematics, dynamics, mechanics, electronics, control engineering or computer science, as well as the complexity of the theoretical concepts required, make this discipline difficult for students to get deep understanding with classical teaching methodologies.

A new education model started in Europe with the signing in 1999 of the Bologna declaration by Ministers of Education from European countries. The purpose of this process is to create the European higher education area by making academic degree standards and quality assurance standards more comparable and compatible throughout Europe. The new model comes closer to the North American and Japanese systems. It gives greater weight to practical training and to intensive research projects [1], [2], [3]. Thus, the student has to play the leading role of his learning, basing the learning process in active methodologies and student autonomous work to the detriment of traditional lecture methods, in which professors talk and students listen. In essence, students should do more than just listen. They should read, write, discuss, or be engaged in solving problems. Specially, students must engage in such higher-order thinking tasks as analysis, synthesis, and evaluation to be actively involved. Strategies promoting active learning are defined as instructional activities involving students in doing things and thinking about what they are doing [4], [5]. This is the framework of the project-based learning activity proposed here.

In this paper we describe an active learning experience in the field of robotics in the context of a Master Degree on Industrial Engineering. Two practical aspects are mainly treated in a course of Industrial Robotics for an engineering degree: The modeling of the robot for the design of its control system, and the practice to obtain programming skills using specific robot languages. Both complementary learning aspects can be benefited by simulation tools. The advantages of simulation-based learning have been studied in [6], [7]. The robot modeling has a high geometric and mathematical complexity, which is added to the difficulties in terms of time and budget required for programming industrial tasks with real robots. The proposed activity has been conducted in two similar courses: *Control and Programming of Robots* and *Industrial Robotics*.

For several years we have proposed this learning project-based experience in which students must model and solve kinematics for a commercial model of a robot manipulator, design some aspects of its control system and program the robot in an industrial application. Simulation tools, developed for this specific project, are essential for student's motivation and understanding of the problem. The interest and the success of this experience is endorsed by the results obtained along these years and presented here. The simulation tools developed and results described have been partially presented in conferences [8] and [9].

The paper is organized as follows. Section 2 briefly describes the courses in which the learning activities have been conducted. The project-based learning activity is presented in Section 3, describing also the main features of the simulation tools in this context. The performance and results of the proposal in the courses are analyzed in Section 4. Statistic academic results as well as student's polls are also presented. Conclusions are discussed in Section 5.

2. Framework: Robotics courses

The first commercial industrial robots appeared just only around forty years ago. Nowadays, the challenge is still the same that promoted its development, to approach the robot to the humankind for improving its existence. The evolution of robotics is related with the technology available each time of the history. In any case, it seems that the future of the humankind is becoming more and more linked to robotics. Hence the proliferation of conferences, courses and, in general, program studies of universities that, now more often, include courses related directly or indirectly with robotics. Probably, a near future will bring specific degrees in this subject with higher specialization. In this context, the experience described here can be valuable.

As mentioned, the proposed activity has been conducted in two similar courses in engineering degrees at the Computer Science and System Engineering Department at the University of Zaragoza: *Control and Programming of Robots* (four-month long course) and *Industrial Robotics* (full year course). The main difference between these courses is that the former has further theoretical background whereas the other is more focused to practice. In order to be brief, we focus on one of the courses involved. Thus, we now briefly describe the course *Control and Programming of Robots*.

The ECTS subject form of the course *Control and Programming of Robots* is presented as example in Table 1. This form is available for all students before they enroll for the course at the beginning of the academic year. The basics of the course are described in this form, pointing out the goals, skills and competencies that the student will develop during the course. The contents of the course are also enumerated and practical information is given. *Control and Programming of Robots* is a course for Master degree which can be taken at the Industrial Engineering Master degree and at the Computer Science Engineering Master degree. The course is designed with 4.8 ECTS, where ECTS stands for European Credit Transfer and accumulation System. One ECTS is considered to be around 25 hours of student's work, this includes not only teaching classes or practice in laboratory sessions but also the autonomous work of the student and the rest of activities involved in the course. The course of *Control and Programming of Robots* is designed with 120 hours of student's work. This means that it is expected that the student will spend a total of 120 hour to pass the course. This total time breaks down into 36 hours of teaching classes, 21 laboratory hours, and 63 hours of personal work and other activities.

The general goals of the courses are: 1) the transmission of the basics that allows future engineers to tackle the system design of the control and programming of a robot. 2) We aim that the student will use correctly in his work general concepts of industrial robotics like mechanical structure, modeling, motion generation and industrial implantation. 3) We also present the concepts required to robotize a production process. 4) Develop the abilities to model an program an industrial robot. 5) Introduce the main evolution lines of the robotics. These courses assume that the students have a prior knowledge, acquired in previous courses, about algebra tools like vectors and matrix calculus and basic knowledge in programming. Knowledge in general physics, mechanics, systems theory or automation systems are also advisable.

The course of *Control and Programming of Robots* is structured in four modules:

1. Components and subsystems of the industrial robot (2 weeks).
2. Modeling, control and programming of robots (8 weeks).
3. Applications and implantation of robots (2 weeks).
4. Advanced robotics (2 weeks).

The laboratory sessions are conducted simultaneously and coordinated with the different modules, in such a way that the student carry out them with the required previous knowledge. Before each laboratory session the student has to prepare a previous work solving a set of questions related with the session and, after the session, a report has to be submitted and presented

Table 1: ECTS Subject form of the course *Control and Programming of Robots*.

University of Zaragoza **ECTS SUBJECT FORM**

Degree: Industrial Engineering	Type: Optative
Faculty: Centro Politécnico Superior	Year: Fifth
Dept.: Computer Science and System Engineering	ECTS credits: 4.8
Knowledge area: System and automatic engineering	Total hours student work: 120
Subject name: Control and programming of robots	Teaching classes hours: 36
Teachers: Josechu Guerrero and Gonzalo López	Laboratory hours: 21
Length: Four-month period	Hours of personal work: 63
Teaching room: A25, L0.05 and L0.06	Teaching language: Spanish

Goals, skills and competencies: The student will use correctly in his work the basic concepts of industrial robotics: mechanical structure, modelling, programming, motion generation, robotics control, as well as industrial implantation. During laboratory hours the student will carry out modeling tasks of the robot and its control as well as learning of the different programming techniques of robots. The student will face successfully the resolution of the kinematics problem of a robot in an autonomous way.

Contents (brief description of the subject):
1. Introduction to Robotics. 2. Morphology of the robot: mechanical structure, Terminal devices, actuators, transmissions and sensors. 3. Spatial localization. 4. Geometric and kinematic modelling. 5. C kinematic control. Trajectory generation. 6. Modeling and dynamic control. 7. Programming of industrial robots. 8. Perception and adaptive control: proximetry, tact, forces. 9. Industrial application. Selection and implantation.

Scheduling of Laboratory classes:
D1. Spatial localization. D2. Geometric and kinematic modeling of manipulators. D3. Control and trajectory generation D4. Dynamic model and robot control. R. Programming and operation of a industrial robot. S1. Programming of robots with simulator. S2. Textual programming.

Bibliography: Refer to `http://add.unizar.es`

Teaching methodology: Expository and active methodologies. Laboratory practices and final project.

Assessment: Attendance to laboratory classes is compulsory. The student will present a report of the laboratory classes and the practical task in a presentation.

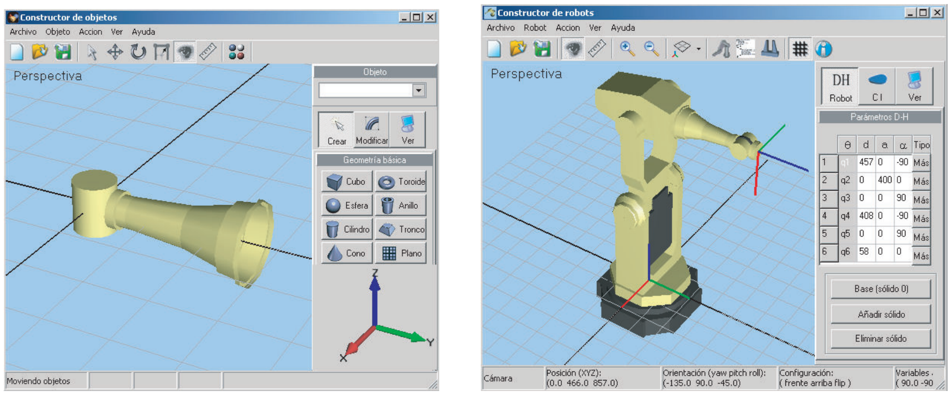
Previous requirements: Preferably for fourth year students.

for assessment. The course is also supported by the ADD (Anillo Digital Docente: `http://add.unizar.es`). This is an intranet or virtual campus based on Moodle and it provides the material used during the course like user's guide, the simulation tools (RobotScene and SGRobot) and class material. Moodle (Modular Object-Oriented Dynamic Learning Environment) is a free learning software platform designed to create online courses and allow interaction with the students. This platform is becoming very popular in the educational community [10], [11], [12].

The assessment has two related goals. The first is to know in each moment the compliance with the proposed formative objectives and the other is to assign the student's grades. The final grade is mainly obtained through the assessment of the fulfilment of the practical task with its presentation and the evaluation of the laboratory sessions.

3. Learning activity and simulation tools

In this section, we describe the activity proposed using the simulation tools developed for this project-based course: RobotScene and SGRobot (SGRobot stands for Graphics Simulator of manipulator Robots). RobotScene [13] is a specific software tool that provides a graphical interface for solving the kinematics of a robot, following the Denavit-Hartemberg procedure [14]. RobotScene also provides a framework for programming the previously created robots. It is composed by three modules specialized on solid, robot and scene creation respectively. SGRobot [15] is a graphical simulator for manipulators where robotic tasks can be programmed using a VAL II-based language. The simulation tools have been designed with Object-Oriented Programming using C++, and the 3D visualization of the robots and surrounding scene is based on OpenGL (Open Graphics Library). There are many robotic platforms that provide simulation frameworks in which users can develop robotic applications [16], [17], but they are not specifically oriented to robot manipulator design. Other platforms provide tools useful for robot kinematics and dynamics simulation, for example the Robotic Toolbox for Matlab [18] or Spacelib [19], but they do not provide graphical interfaces or robot programming tools. Some projects as OROCOS [20] and ROBOOP [21] allow covering this empty space, but they do not provide an easy graphical interface and so, their use requires important learning efforts. Furthermore, they need an external compiler in order to perform any simulation. In addition, there are several simulation platforms as ROBOGUIDE [22] and



		θ	d	a	α	Tipo
1	q1	457	0	-90	Más	
2	q2	0	400	0	Más	
3	q3	0	0	90	Más	
4	q4	408	0	-90	Más	
5	q5	0	0	90	Más	
6	q6	58	0	0	Más	

Posición (XYZ): (0.0 466.0 857.0)
Orientación (yaw pitch roll): (-135.0 90.0 -45.0)
Configuración: (frente arriba flip)
Variables . (90.0 -90

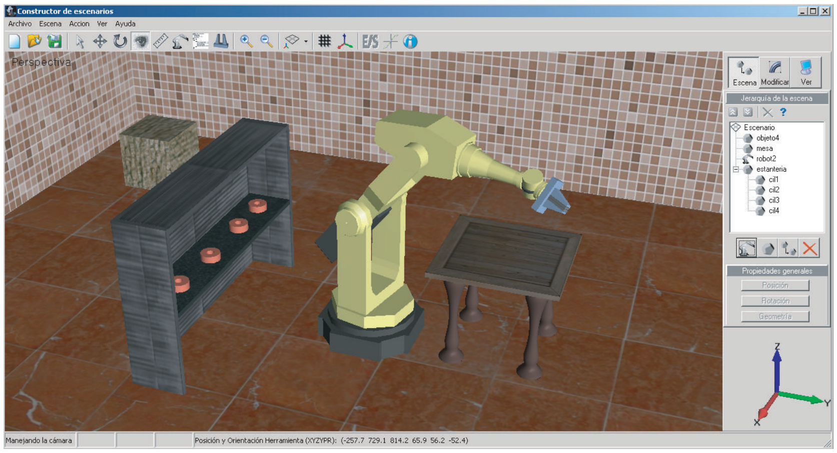
Figure 1: Example of the definition of a solid using RobotScene (left) and the robot obtained after assembling all the robot solids (right).

RobotStudio [23] developed by robot manufacturers in order to provide off-line programming tools specifically designed for their robots. RobotScene provides modules for creating solids, robots and robotic scenarios in an easy manner, and it is used in the first part of the project.

3.1. Robot modeling and kinematics: RobotScene

The first stage of the learning project is to model and solve the kinematics of a commercial robot. To achieve this, students follow the Denavit-Hartenberg (DH) convention [14]. J. Denavit and R. S. Hartenberg proposed in 1955 a procedure to systematically define a coordinate system attached to each link of an articulated chain, allowing to determine the forward kinematics of the complete chain in a systematic way. A different commercial robot is assigned to each student. Once the students have analyzed the morphology of their assigned manipulator, identifying its joints and its solids, and assigning all their reference frames according to the DH rules, they can begin to model their robot using RobotScene. Figure 1(left) shows an example of solid definition of a FANUC robot required for the subsequent assembly into a cinematic chain.

Once all the manipulator solids have been created, students can assemble their robots using the Robot Constructor Module. At this point, students need to have previously determined both the DH parameters as well as the equations that solve the inverse kinematics of their robot. It is important to note at this point the role of the Denavit-Hartenberg convention in the robot modeling process with RobotScene: it determines not only the main

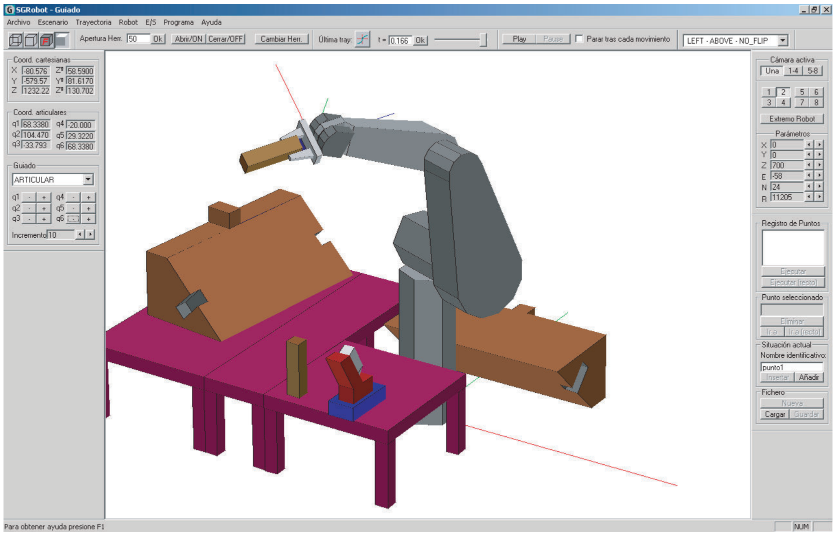


Manejando la cámara

Posición y Orientación Herramienta (XYZYPR): (-257.7 729.1 814.2 65.9 56.2 -52.4)

Figure 2: Example of a scene created with RobotScene consisting of floor, walls and several objects with a robot preciously defined by the student in the first part of the task.

part of the joint modeling, but the form in which each solid has been created as well. Once students have finished the robot assembly process, see Fig. 1(right) as example, they can use a robot guidance tool that allows them to move the robot by dragging the joint-associated cursors or by specifying robot destinations in joint coordinates. In order to complete the robot modeling stage, students must implement the equations that solve the inverse kinematics of their robot. These equations must have been previously derived by using either geometric or algebraic approaches. For making possible the mentioned implementation, the Robot Constructor Module provides a specific programming tool which allows editing, performing syntactical checking and compiling source code. During this programming phase, students must pay attention to several problems inherent to inverse kinematics, as multiplicity of solutions, ill-conditioned equations and singularities detection and their treatment. Once inverse kinematics is implemented, students can easily check its correctness by using the robot guidance tool, that allows them to improve their performance in this learning step. Next, a realistic environment can be defined using RobotScene to place the robot created by the student and the objects to be manipulated by the implemented robot. An example of a scene is shown in Fig. 2.



Apertura Herr. 50

t = 0.166

□ Parar tras cada movimiento

LEFT - ABOVE - NO_FLIP

Coord. cartesianas

X -80.576 Zª 58.5900
Y -579.57 Yª 81.6170
Z 1232.22 Zª 130.702

Coord. articulares

q1 68.3380 q4 -20.000
q2 104.470 q5 29.3220
q3 -33.793 q6 68.3380

Guiado

ARTICULAR

q1 - + q4 - +
q2 - + q5 - +
q3 - + q6 - +

Incremento 10

Parámetros

X 0
Y 0
Z 700
E -58
N 24
R 11205

Situación actual

Nombre identificativo: punto1

Figure 3: Example of a PUMA robot arm in a scenario performing an assembly task with SGRobot.

3.2. Robot task programming: SGRobot

The second stage of the learning project-based experience consists in designing and programming an industrial robotic task. For this part of the activity, a scene containing the robot arm and the objects involved in the task is given (Fig. 3).

The assigned task the students have to achieve consists of several steps: First, the robot has to take the tool, which is a glue gun. Then, the tool needs to be calibrated and the objects of the scene located. The glue is spread around the edge surface of the base object following particular specifications of velocity, continuity or relative position. Finally a second object is mounted on the first fastened with two pins and the glue. Students have to program the robot by using the provided language. For this purpose the SGRobot tool is used. SGRobot is a graphical simulation tool that allows the programming of robot arms in a graphical way (guided) and textual way using a defined programming language similar to the commercial language VAL II.

The SGRobot provides the editor-compiler of the robotic programming language CVAL2 developed for SGRobot. An example of code in CVAL2 is given in Table 2 to show the general structure and syntax of a program. In that simple program, the hand of the robot just performs a circular motion. The SGRobot also provides the guidance tool, which is for programming the

Table 2: Example of program in CVAL2.

```
//*****************************************************************
//          PROGRAM DEMOCIRCLE
// The robot follows a circular trajectory
//*****************************************************************
void main() {
 // Circle center:
  T_TRANSF transfCenter;
 // Auxiliary variables:
  T_FLOAT radio, angle, incAngle, rad;
  T_TRANSF transf;
 // Circle parameters:
  transfCenter = TRANSF_EULER(400,400,400,0,0,0);
  radio = 200;
  incAngle = 20;
 // Perform the motion:
  for (angle = 0; angle<=360; angle = angle + incAngle) {
    rad = TO_RADIANS(angle);
    transf = SHIFT_BY(transfCenter,radio*cos(rad),radio*sin(rad),0);
    MOVE(transf);
  }
}
```

robot by guidance, graphical representation of trajectories, object manipulation, etc. The learning objectives of the robot task programming are that students achieve the competencies of planning a robotic task in a simulation environment, implementation of the task by means of guidance and explicit programming language, and solve successfully the usual problems of obstacle avoidance, object localization, detection and avoidance of robot singularities and robot configuration selection. The concept of the singularity is complex to understand but, thanks to the help of the simulation tool, the student can learn this concept intuitively. The simulation tool gives the option to compile and execute a program in CVAL2 and additionally, it allows traducing automatically the code to VAL II, the standard language that can be directly executed in the real robot. The code that can be generated from the simulation tool is used in a practicing activity with the real robots (Fig. 4). The goal of this part is that students work with real robots learning the basic elements of an industrial robot and its operation. They also learn and practice the different ways of robot programming while designing the robotic task by guidance and textual programming.

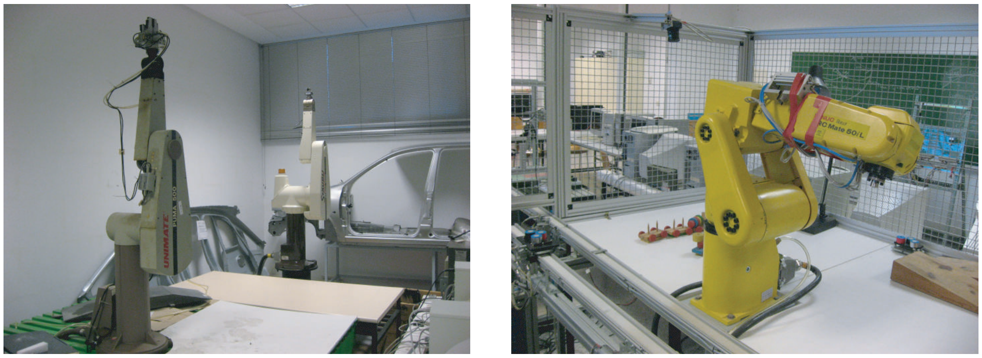


Figure 4: PUMA robots (left) and FANUC robot (right) used in the practicing activity.

4. Results and learning improvements

In the framework of new methodologies, laboratory and project-oriented courses are encouraged. In our case, two main learning aspects can be benefited from the use of simulation tools in the context of an industrial robotics related subject: the robot control and modeling in-depth comprehension, and the acquisition of robot programming skills. The main learning improvements provided by the use of RobotScene during the robot modeling phase, are: the better comprehension of DH convention, and the in-depth understanding of the inverse kinematics problem. RobotScene provides a framework in which students can check in a visual way their DH parameters, because any error will be reflected as a wrong solid assembly. On the other hand, the inverse kinematics programming allows understanding its authentic complexity, because students must take into account some different problems related to its nature and implementation. On the other hand, SGRobot is useful to know and appreciate the utility of a graphical robotic simulator. Students can learn and practice with the different ways of programming robots. The activity requires that students implement under the simulator an assembling task using guidance and explicit textual language. They also need to document their final application including explanations about the reference assignment and the motion planning design. In order to perform the robotic task successfully the students need to solve usual problems that appears in programming robotic tasks, like obstacle avoidance, object localization, singularities avoidance, robot configuration selection, etc.

In general, it should be noted that traditional lecture methods are less effective and less efficient than instructional approaches that place a strong

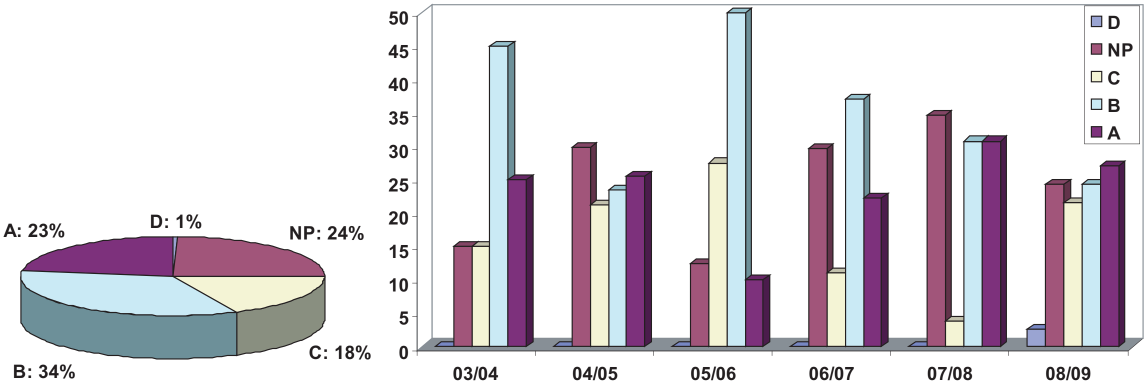


Figure 5: Grades of the course *Control and Programming of Robots*. Mean grades from years 2003 to 2009 (left) and evolution of the percentage of the grades along the years (right). The grades are from minor to major: NP, D, C, B, A where NP, D mean fail to pass.

emphasis on guidance of the student learning process [24]. However, the advantage of guidance and active learning begins to recede when students do not have sufficiently high prior knowledge for autonomous learning. Therefore, in order to perform these activities with guarantees of success, a prior knowledge is required from the students.

The improvements observed because of the activity proposed are supported by the rate of success and the good reception of the courses involved. The mean values of the grades obtained by students from the last 6 years (from 2003 to 2009) in *Control and Programming of Robots* and *Industrial Robotics* are shown in Fig. 5 and Fig. 6, respectively. The grades are defined in a scale from 0 to 10 with A between 9 and 10, B between 7 and 9, C between 5 and 7, D failure to pass, and NP if the student does not apply for the evaluation (the presentation of the project). Note that there are very few D grades in the results. This is because, given the project-based focus of the course, the student's work is all up to date. Moreover, each student himself knows before the end of the course if he has acquired the competencies necessary to pass. Otherwise, they are advised during the course how to improve they performance in order to complete the course successfully. If not, the student usually choose not to apply for the evaluation.

We have polled 50% of the students during the academic year 08-09 about their dedication to the course of *Control and Programming of robots* and the results are shown in Fig 7. The mean result is that they spent 50 hours (35

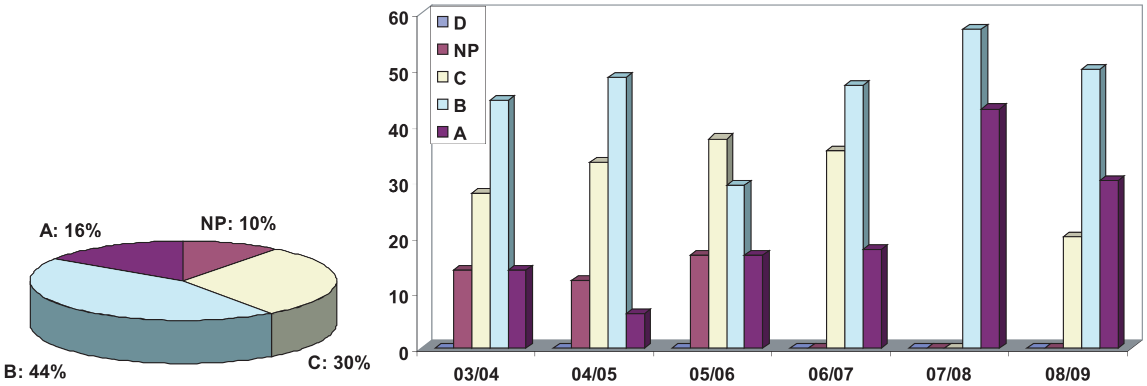


Figure 6: Grades of the course *Industrial Robotics*. Mean grades from years 2003 to 2009 (left) and evolution of the percentage of the grades along the years (right). The grades are from minor to major: NP, D, C, B, A where NP, D mean fail to pass.

%) in studying the theory and assistance to class hours. They spent 43 hours (30 %) in practice sessions, including previous preparation, assistance, and elaboration of the corresponding reports. And they spent 51 hours (35 %) in their work projects. Notice from these mean values that students estimate that 65 % of the time dedicated to the course corresponds to practice, and relevance of practice is a characteristic of the competencies-based learning. As previously commented, the course is designed with 4.8 ECTS, approximately a total 120 hours of student work (Table 1). Comparing with the poll results in Fig 7, students estimate that they spent in mean 24 hours of personal work over the corresponding designed time of the course. In order to reduce this difference we are going to introduce more active learning methodologies in teaching classes to improve acquisition of theoretical concepts required to fulfill the project-based activities.

We have also represented the dedication hours of the students in the first half of the course and the second half in Fig 7 (left). It can be seen that the dedication to theory decreases along the course. The reason is that the theoretical concepts require higher effort at the beginning, but once the concepts are comprehended less dedication is needed. The total hours dedicated to practice also decreases between the first to second part of the course, this is because the scheduling of the laboratory sessions makes the students work up to date, avoiding a peak of work when the laboratory sessions end. It can be seen in Fig 7 (left) that the total hours dedicated to the project is approximately constant during the course. This agree with the

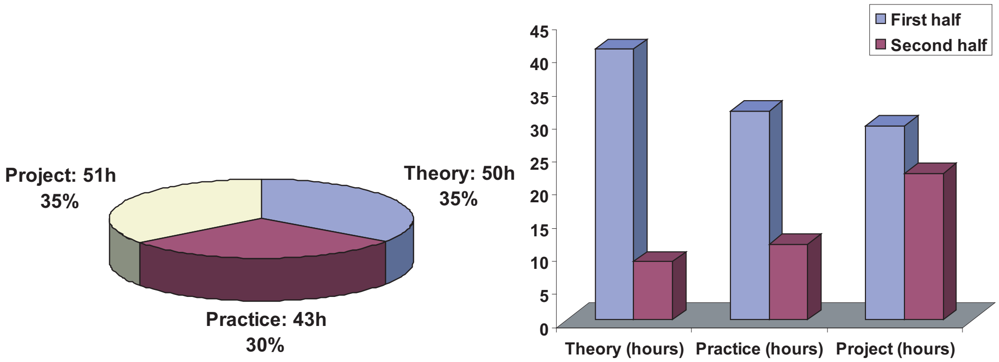


Figure 7: Dedication hours of the students polled in academic year 2008-2009. (Left) Mean values of the hours dedicated to theory (classes and personal study), practice (laboratory sessions with previous and posterior work) and project (the work about the robot assigned to each student). (Right) Mean values of the hours dedicated to theory, practice and project divided into the first half and second half of the course.

design line of the course focused to a project-based orientation.

Additional advantage of using the simulation tools is the lower budget, which allows to increase the number of workstations keeping an affordable number of real robots. This allows the students to work and develop their code in a realistic environment, finally trying their programs with the real robot. Moreover, the use of simulation tools is useful in any robotic activity for the reduction of danger with the robots. This is specially advisable in the initial stage of the activity, when, given the the insufficient knowledge of the students, working with the real robot would be risky.

5. Conclusions

The benefits of active learning have been acknowledged and, in the framework of new methodologies, laboratory and project-oriented courses are encouraged. In this work, we describe a project-based activity that takes advantage of simulation tools covering these issues. Active learning and guidance of the student learning process is more effective and more efficient than instructional approaches, but only if the students have sufficiently high prior knowledge for autonomous learning. Therefore, in order to perform these activities with guarantees of success, a prior knowledge is required from the students. The improvements observed because of the activity proposed are supported by the rate of success and the good reception of the courses involved. We found that one disadvantage of this type of project-base learning activity is that it can get involved students so much. Then, they could spend too much time to complete the activity, exceeding the advisable maximum to pass the course to the detriment of other courses. So, this kind of activities needs to be designed carefully.

The activity proposed is supported by two simulation tools specifically developed for this purpose. These simulation tools can be freely distributed and, to our knowledge, they are currently used for teaching in the context of Industrial Engineering Degrees at several universities as well as the University of Zaragoza. These tools are also used for teaching at secondary school. In this case, the activity involves the learning of the types of robots and their main features, like degrees of freedom, types of sensors and actuators and control system. This particular activity takes advantage of the graphical and intuitive interface of the software and it is designed for students at $14 - 15$ years old. Their broad use shows the versatility and usefulness of these simulation tools for educational purposes at different levels.

6. References

[1] K. De Wit, The consequences of european integration for higher education, International Association of Universities (IAU), in Higher Education Policy 16 (2) (2003) 161–178.

[2] http://www.bologna2009benelux.org (2009).

[3] T. Halvorsen, G. Mathisen, T. Skauge, Identity Formation or Knowledge Shopping: Education and research in the new globality, Norwegian Centre for International Cooperation in Higher Education (SIU), Bergen, 2005.

[4] A. W. Chickering, Z. F. Gamson, Seven principles for good practice, The American Association for Higher Education Bulletin 39 (7) (1987) 3–7.

[5] C. H. J. Davies, Student engagement with simulations: a case study, Computers & Education 39 (3) (2002) 271–282.

[6] K.-E. Chang, Y.-L. Chen, H.-Y. Lin, Y.-T. Sung, Effects of learning support in simulation-based physics learning, Computers & Education 51 (4) (2008) 1486–1498.

[7] P. Forcheri, M. T. Molfino, Software tools for the learning of programming: A proposal, Computers & Education 23 (4) (1994) 269–276.

[8] G. López-Nicolás, A. Romeo, J. J. Guerrero, Simulation tools for active learning in robot control and programming, in: European Association for Education in Electrical and Information Engineering (EAEEIE), Valencia, Spain, 2009.

[9] G. López-Nicolás, A. Romeo, J. J. Guerrero, Project based learning of robot control and programming, in: International Conference on Engineering Education. ICEE/ICEER, Korea, 2009.

[10] K. J. Crippen, B. L. Earl, The impact of web-based worked examples and self-explanation on performance, problem solving, and self-efficacy, Computers & Education 49 (3) (2007) 809 – 821.

[11] C. Romero, S. Ventura, E. García, Data mining in course management systems: Moodle case study and tutorial, Computers & Education 51 (1) (2008) 368–384.

[12] T. Martín-Blas, A. Serrano-Fernández, The role of new technologies in the learning process: Moodle as a teaching tool in physics, Computers & Education 52 (2009) 35–44.

[13] A. Romeo, The role of simulation tools in the teaching of robot control and programming, in: International Symposium on Robotics, Barcelona. Spain, 2009, pp. 157–162.

[14] J. Denavit, R. S. Hartenberg, Kinematic notation for lower-pair mechanisms based on matrices, Transactions of the ASME, Journal of Applied Mechanics 23.

[15] J. Guerrero, Prácticas de control y programación de robots, dep. legal Z-178-2003. Universidad de Zaragoza. It includes Simulator SGRobot 2.0. Email to jguerrer@unizar.es to obtain a free copy for teaching (2003).

[16] `http://msdn.microsoft.com/robotics` (2009).

[17] M. Mellado, C. Correcher, J. V. Catret, D. Puig, Virtualrobot: an open general-purpose simulation tool for robotics, Naples, Italy, 2003, pp. 271–350.

[18] P. I. Corke, A robotics toolbox for matlab, IEEE Robotics and Automation Magazine 3 (1) (1996) 24–32.

[19] G. Legnani, Spacelib: a software library for the kinematic and dynamic analysis of systems of rigid bodies, `http://bsing.ing.unibs.it/~glegnani` (2006).

[20] H. Bruyninckx, Open robot control software: the orocos project, in: IEEE International Conference on Robotics and Automation (ICRA), 2001, pp. 2523–2528.

[21] R. Gourdeau, Object oriented programming for robotic manipulators simulation, IEEE Robotics and Automation Magazine 4 (3).

[22] Roboguide: a family of offline robot simulation software, `http://www.fanucrobotics.com` (2009).

[23] Robotstudio: of offline robot programming for abb robots, `http://www.abb.com` (2009).

[24] P. A. Kirschner, J. Sweller, R. E. Clark, Why minimal guidance during instruction does not work: an analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching, Educational Psychologist 41 (2) (2006) 75–86.